

LATVIJAS UNIVERSITĀTE  
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE  
MATEMĀTIKAS NODAĻA

**NESTRIKTA LINEĀRĀ PROGRAMMĒŠANA PREČU OPTIMĀLĀS  
PLŪSMAS NOTEIKŠANAI**

BAKALAURA DARBS

Autors: **Viktorija Čapa**

Stud. apl. vc13023

Darba vadītājs: prof. Dr. math. Svetlana Asmuss

RĪGA 2017

## **Anotācija**

Darbs ir veltīts uzdevumam par preču optimālo plūsmu transporta tīklā, kuru modelē, lietojot grafu ar indeksētiem lokiem. Problēma ir apskatīta ar nestriktiem parametriem un risināta, lietojot nestriktās lineārās programmēšanas metodi. Optimālās plūsmas problēmai atbilstošais nestriktās programmēšanas uzdevums ir reducēts uz klasisko diskrētās programmēšanas uzdevumu ar papildus lēmuma pieņemšanas parametriem, kura risināšanai tiek pielietots sazarosšanās un robežu algoritms. Aprakstītā metode ir aprobēta, analizējot skaitliskus piemērus un veicot iegūto rezultātu analīzi atkarībā no lēmumu pieņemšanas parametriem.

Atslēgas vārdi: vairāku veidu preču plūsma, nestrikts lielums, iespējamības un nepieciešamības sakarības, nestrikta lineārā programmēšana, sazarosšanās un robežu algoritms

## **Abstract**

This paper is devoted to the multicommodity flow problem in a transportation network, which is modeled with a weighted graph. The problem is inspected with fuzzy parameters and solved using fuzzy linear programming. A fuzzy linear programming problem corresponding to the optimal flow problem is reduced to a discrete programming problem with extra decision making parameters and solved using the branch and bound algorithm. The proposed technique is approved by using numerical examples and analyzing obtained results that depend on decision making parameters.

**Keywords:** multicommodity flow, fuzzy number, possibility and necessity relations, fuzzy linear programming, branch and bound algorithm

# Saturs

<b>Apzīmējumu saraksts</b>	<b>2</b>
<b>Ievads</b>	<b>5</b>
<b>1 Problēmas nostādne</b>	<b>7</b>
1.1 Grafs kā transporta tīkla modelis . . . . .	7
1.2 Optimālās plūsmas problēma viena veida precēm . . . . .	9
1.3 Optimālās plūsmas problēma vairāku veidu precēm . . . . .	10
1.4 Optimālās plūsmas problēma ar nestriktiem parametriem . . . . .	13
<b>2 Nestrikti lielumi un to lietojumi optimālās plūsmas problēmā</b>	<b>14</b>
2.1 Nestriktas kopas . . . . .	14
2.2 Nestrikti lielumi un to veidi . . . . .	15
2.3 Darbības ar nestriktiem lielumiem . . . . .	17
2.4 Optimālās plūsmas problēmas piemēri . . . . .	18
<b>3 Nestriktā lineārā programmēšana</b>	<b>24</b>
3.1 Nestrikti parametri lineārās programmēšanas uzdevuma nosacījumos . . . . .	24
3.2 No nestriktiem parametriem atkarīgas funkcijas optimizēšana . . . . .	26
<b>4 Problēmas risināšana, lietojot diskrēto programmēšanu</b>	<b>31</b>
4.1 Diskrētā programmēšanas uzdevuma ierobežojošo parametru izvēle . . . . .	31
4.2 Diskrētā programmēšanas uzdevuma mērķa funkcijas parametru izvēle . . . . .	32
4.3 Sazarošanās un robežu algoritms diskrētās programmēšanas uzdevuma risināšanai un tā praktiskā realizācija . . . . .	33
<b>5 Preču optimālās plūsmas noteikšanas piemēri</b>	<b>38</b>
5.1 Pirmais uzdevums . . . . .	38
5.2 Otrais uzdevums . . . . .	43
<b>Nobeigums</b>	<b>47</b>
<b>Izmantotā literatūra un avoti</b>	<b>48</b>
<b>1 Pielikums</b>	<b>49</b>
1.1 Pirmā uzdevuma atrisinājums . . . . .	49
1.2 Otrā uzdevuma atrisinājums . . . . .	54
<b>2 Pielikums</b>	<b>57</b>
2.1 Programmas kods . . . . .	57

## Apzīmējumu saraksts

$G$  grafs,

$V$  grafa virsotņu kopa,

$A$  grafa loku kopa,

$n$  grafa virsotņu skaits,

$(i, j)$  grafa loks jeb posms,

$U$  svara funkcija,

$u_{ij}$  loka  $(i, j)$  caurlaides spēja,

$s$  sākuma virsotne,

$w$  beigu virsotne,

$c_{ij}$  vienas vienības transportēšanas izmaksas pa loku  $(i, j)$ ,

$P$  grafa maršrutu kopa,

$p$  maršruts grafā,

$x(p)$  plūsma pa maršrutu  $p$ ,

$d$  pieprasītais preces daudzums,

$\delta_{ij}(p)$  indikatormainīgais, kas parāda, vai loks  $(i, j)$  pieder maršrutam  $p$ ,

$K$  preču veidu skaits,

$s^k$   $k$ -tā veida preces transportēšanas sākuma virsotne,

$w^k$   $k$ -tā veida preces transportēšanas beigu virsotne,

$c_{ij}^k$  vienas  $k$ -tā veida preces vienības transportēšanas izmaksas pa loku  $(i, j)$

$P^k$  visu no virsotnes  $s^k$  uz virsotni  $w^k$  vērsto maršrutu kopa  $k$ -tā veida preces transportēšanai,

$c(p)$  kopējās  $k$ -tā veida preces transportēšanas izmaksas pa maršrutu  $p \in P^k$ ,

$d^k$  pieprasītais  $k$ -tā veida preces daudzums,

$\tilde{c}_{ij}^k$  vienas  $k$ -tā veida preces vienības transportēšanas izmaksas pa loku  $(i, j)$  nestriktos skaitļos,  
 $\tilde{c}(p)$   $k$ -tā veida preces transportēšanas izmaksas pa maršrutu  $p \in P^k$  nestriktos skaitļos,  
 $\tilde{u}_{ij}$  loka  $(i, j)$  caurlaides spēja nestriktos skaitļos,  
 $\tilde{d}^k$   $k$ -tā veida preces pieprasītais daudzums nestriktos skaitļos,  
 $\mathbb{R}$  reālo skaitļu kopa,  
 $L$  universāla kopa,  
 $\mathbb{Z}$  veselo skaitļu kopa,  
 $\tilde{T}, \tilde{H}, \tilde{D}$  nestrikta kopas,  
 $\mu$  piederības funkcija,  
 $\alpha, \beta$  lēmumu pieņemšanas parametri,  
 $\tilde{T}_\alpha$  kopas  $\tilde{T}$   $\alpha$ -griezums,  
 $\tilde{\tau}, \tilde{h}, \tilde{d}$  nestrikta lielumi,  
 $\Pi, \preceq^\Pi$  iespējamības sakarība,  
 $\mu^\Pi$  iespējamības sakarības piederības funkcija,  
 $N, \preceq^N$  nepieciešamības sakarība,  
 $\mu^N$  nepieciešamības sakarības piederības funkcija,  
 $(\tilde{h})_\alpha^L$  nestrikta lieluma  $\tilde{h}$   $\alpha$ -griezuma infīms,  
 $(\tilde{h})_\alpha^R$  nestrikta lieluma  $\tilde{h}$   $\alpha$ -griezuma suprēms,  
 $z$  mērķa funkcija,  
 $\tilde{a}_{ij}, \tilde{b}_i, \tilde{c}_j$  nestrikta lielumi,  $i = 1, \dots, m, j = 1, \dots, g$ ,  
 $\tilde{E}$  klasiskās binārās sakarības " $\leq$ " turpinājums,  
 $\tilde{E}_\alpha$   $\alpha$ -sakarība, kas ir saistīta ar  $\tilde{E}$ ,  
 $\tilde{E}_\alpha^*$  stingra  $\alpha$ -sakarība, kas ir saistīta ar  $\tilde{E}$ ,  
 $y(p)$   $k$ -tā veida preces transportēšanas izmaksas maršrutā  $p \in P^k$ ,

$D$  plānu kopa,

$t$  iterāciju indekss,  $t = 0, 1, 2, \dots$ ,

$M_t$  diskrētās programmēšanas uzdevumu kopa iterācijā  $t$ ,

$x_t^*$  diskrētās programmēšanas uzdevuma atrisinājuma aproksimācija pēc  $t$  iterācijām,

$f_t^*$  mērķa funkcijas maksimālās vērtības aproksimācija pēc  $t$  iterācijām,

$LPU_t$  lineārās programmēšanas uzdevums iterācijā  $t$ ,

$DPU_t$  diskrētās programmēšanas uzdevums iterācijā  $t$ .

# Ievads

Preču optimālās plūsmas problēma jeb minimālo izmaksu preču plūsmas problēma tiek uzskatīta par vienu no pamata problēmām grafu teorijā. Tās mērķis ir atrast visefektīvāko maršrutu no sākuma vietas, ražotnes, uz mērķa vietu, noliktavu vai tirdzniecības vietu, un plūsmas lielumu pa maršruta posmiem, nodrošinot vismazākās izmaksas un nepieciešamā plāna izpildi. Problēma ir universāla un var tikt izmantota vai modificēta citu līdzīgu problēmu (piemēram, īsākā ceļa problēmas) risināšanai.

Darbā tiek apskatīts gadījums, kad vienlaikus ir jāpārvadā vairāku veidu preces, katra veida preces izcelsmes vieta un galamērķis var būt atšķirīgi. Ir pieņemts, ka maršruta posma jeb loka caurlaides spēja attiecas uz visām precēm, citiem vārdiem, pa katru maršruta posmu ir iespējams nogādāt noteiktu vienību daudzumu. Pietuvojoties reālām dzīves situācijām, ir pieņemts, ka transportēšanas izmaksas, caurlaides spēja un preces pieprasītais daudzums nav konkrēti zināmi un ir izteikti, lietojot nestriktus parametrus. Problēmas risināšanai tiek izmantota nestriktā lineārā programmēšana, kuras ietvaros optimālo atrisinājumu meklē atkarībā no lēmumu pieņemšanas parametra  $\alpha$  un  $\beta$ . Problēma tiešā mērā saistās ar loģistikas nozari, kur ikdienas galvenie uzdevumi ietver gan finansiālā, gan laika ziņā izdevīgāko maršrutu plānošanu.

Darba mērķis ir aprakstīt vairāku veidu preču optimālās plūsmas problēmas risināšanas metodes, lietojot nestrikto lineāro programmēšanu, un ar skaitliskiem piemēriem parādīt, kā iegūtos rezultātus izmantot lēmumu pieņemšanā. Darbs pamatā balstās uz diviem nozīmīgiem zinātniskajiem rakstiem [4] un [3].

Darba uzdevumi ir:

- iepazīties ar nestrikto lielumu teorijas pamatnostādņiem;
- iepazīties ar nestrikto lineārās programmēšanas problēmu risināšanas metodēm;
- apgūtās metodes pielietot preču optimālās plūsmas problēmas risināšanai;
- izveidot programmas kodu programmēšanas valodā  $R$ ;
- aprobēt metodes, analizējot skaitliskus piemērus;
- veikt iegūto rezultātu analīzi atkarībā no lēmumu pieņemšanas parametriem  $\alpha$  un  $\beta$ .

Pirmā nodaļā ietver izmantoto grafu teorijas jēdzienu skaidrojumu, galvenās problēmas nostādni un tās vispārinājumus. Otrā nodaļa satur informāciju par nestriktām kopām, nestriktiem

lielumiem, darbībām ar nestriktiem lielumiem. Nodaļas noslēgumā ir aprakstīti divi preču optimālās plūsmas problēmas piemēri ar parametriem nestriktu lielumu veidā. Trešā nodaļa tiek veltīta iespējamības un nepieciešamības sakarību lietojumiem nestriktās lineārās programmēšanas uzdevumu analīzei. Ceturtā nodaļa ietver galvenās problēmas reducēšanu uz diskrētās programmēšanas uzdevumu un sazarošanās un robežu algoritma aprakstu. Piektā nodaļa veltīta divu optimālās plūsmas problēmas piemēru risināšanai un iegūto rezultātu analīzei. Pielikums satur piemēru risinājumu starprezultātus un izmantotās programmas kodu.

Attēlu un tabulu numerācija sākas ar katru jaunu apakšnodaļu. Pirmie divi skaitļi norāda apakšnodaļas numuru, kurā tabula vai attēls atrodas, un pēdējais skaitlis norāda attēla vai tabulas kārtas numuru. Darbā iekļautajiem piemēriem numerācija piešķirta tikai diviem galvenajiem aplūkotajiem, praktiski risinātajiem piemēriem. Numerācija darbā iekļautajām definīcijām un apgalvojumiem ir atsevišķa un vienkārša – neatzīmējot nodaļu vai apakšnodaļu, kurā formulējums atrodas. Numerācija darbā iekļautajām lemmām un sekām arī ir vienkārša un kopīga.

# 1 Problēmas nostādne

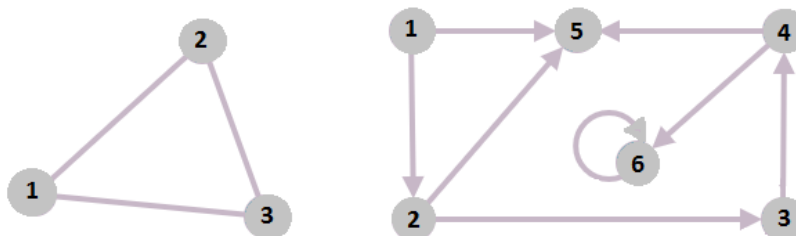
## 1.1. Grafs kā transporta tīkla modelis

Matemātikā transporta tīklu modelēšanai izmanto grafus. Transporta tīkla mezglus sauc par virsotnēm, taču ceļus, kas šos mezglus savieno, uzskata par lokiem. Bakalaura darbā aplūkotās problēmas tiek skatītas transporta tīklos, līdz ar to problēmu uzdošanai tiek izmantoti grafi. Šajā nodaļā tiek aprakstīti ar grafiem saistītie jēdzieni.

**1. definīcija.** [7] Pāri  $(V, A)$ , kur  $V$  ir virsotņu kopa un  $A \subseteq V \times V$  ir loku kopa, sauc par grafu un apzīmē ar  $G$ .

**1. piezīme.** [7] Grafs  $G = (V, A)$  ir galīgs, ja grafa virsotņu skaits ir galīgs, t.i. eksistē naturāls skaitlis  $n$  tāds, ka  $V = \{1, \dots, n\}$ .

**Piemērs.** Aplūko 1.1.1. attēlu ar diviem grafiem – neorientētu (pa kreisi) un orientētu (pa labi). Neorientētā grafā loki  $(i, j)$ , kur  $i \in V$  un  $j \in V$ , nav orientēti, t.i., katram lokam  $(i, j) \in A$  arī loks  $(j, i) \in A$ . Attēlā var redzēt neorientētu grafu ar 3 virsotnēm, attiecīgi grafa loku kopa  $A$  satur lokus-  $\{(1, 2), (2, 1), (2, 3), (3, 2), (1, 3), (3, 1)\}$ . Attēla labajā pusē attēlots orientēts grafs ar 6 virsotnēm un 8 orientētiem lokiem. Bultiņas ilustrācijā norāda virzienu, uz kurieni loks tiek vērsts. Orientēta grafa loku kopa  $A$  satur lokus-  $\{(1, 2), (1, 5), (2, 3), (2, 5), (3, 4), (4, 5), (4, 6), (6, 6)\}$ . Loku, kura sākuma virsotne sakrīt ar tā beigu virsotni, sauc par cilpu, šeit attiecīgi novērojama cilpa virsotnei 6.



Att. 1.1.1: Neorientēts (pa kreisi) un orientēts (pa labi) grafs

**2. definīcija.** Par maršrutu no virsotnes  $i$  uz virsotni  $j$  ( $i \in V$  un  $j \in V$ ) grafā  $G = (V, A)$  sauc

secīgu loku kopu  $\{(i, i_1), (i_1, i_2), \dots, (i_{t-1}, i_t), (i_t, j)\}$ . Maršrutu apzīmē ar  $p$  un lieto pierakstu  $p = (i, i_1, i_2, \dots, i_{t-1}, i_t, j)$  vai  $p : i \rightarrow i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_{t-1} \rightarrow i_t \rightarrow j$ .

Aplūkojot 1.1.1. attēla orientēto grafu, no virsotnes 1 uz virsotni 5 virzās trīs maršruti:  $p = (1, 5)$ ,  $p = (1, 2, 5)$ ,  $p = (1, 2, 3, 4, 5)$ . Turpmāk maršrutu pierakstīsim veidā  $1 \rightarrow 2 \rightarrow 5$ , kas ekvivalents maršrutam  $p = (1, 2, 5)$ . Bakalaura darbā tiks aplūkots plūsmas lielums maršrutā  $p$ , ko apzīmē ar  $x(p)$  (t.i., preču daudzums, ko transportē pa katru no maršruta lokiem).

**3. definīcija.** [7] Grafu  $G$  sauc par grafu ar indeksētiem lokiem vai grafu ar svariem, ja ir uzdota funkcija  $U : A \rightarrow \mathbb{R}$ , kas katram lokam  $(i, j) \in A$  piekārto nenegatīvu vērtību  $U(i, j)$ .

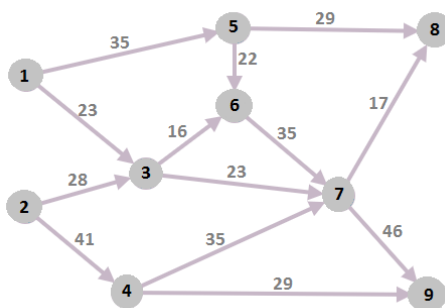
Grafu ar indeksētiem lokiem bieži pieraksta kā trijnieku  $G = (V, A, U)$ . Šīs funkcijas vērtības tiek izmantotas uzdevuma nostādņē. Vērtību  $U(i, j)$  bakalaura darbā interpretē kā loka caurlaidi un apzīmē ar  $u_{ij}$ .

Turpmākajā darbā tiek aplūkoti galīgi grafi bez cilpām ar svaru funkciju  $U$ , kas pieņem tikai nenegatīvas vērtības.

**4. definīcija.** [7] Virsotni  $i \in V$  sauc par izteku tad un tikai tad, ja katram  $j \in V$   $(j, i) \notin A$ .

**5. definīcija.** [7] Virsotni  $i \in V$  sauc par ieteku tad un tikai tad, ja katram  $j \in V$   $(i, j) \notin A$ .

**Piemērs.** Aplūko dzelzceļa tīklu, kas sastāv no 9 stacijām. Stacijas savā starpā ir savienotas ar sliedēm, kā tas ir redzams 1.1.2. attēlā, kopā veidojot 14 posmus. Pa šiem posmiem kursē preču vilcieni. Vilciena kustības virzienu attēlā norāda bultiņas. Pa katru maršruta posmu iespējams transportēt noteiktu preču daudzumu, kas attēlā norādīts tonnās. No stacijas 1 uz stacijām 8 un 9 ir jānogādā akmeņogļu kravas, no stacijas 2 uz staciju 9 ir jānogādā stikla rūpniecības smilšu krava. Šajā gadījumā stacijas 1 un 2 tiek dēvētas par transporta tīkla iztekām, stacijas 8 un 9 – transporta tīkla ietekām.



Att. 1.1.2: Transporta tīkla ilustrācija

## 1.2. Optimālās plūsmas problēma viena veida precēm

Dots transporta tīkls, kas satur  $n$  virsotnes. Tiek aplūkota optimālās plūsmas problēma viena veida preču transportēšanas gadījumā. Ir zināmi ar transportēšanu saistītie raksturlielumi.

Prece jānogādā no tās sākuma virsotnes  $s$  uz beigu virsotni  $w$ . Katram lokam  $(i, j) \in A$  ir definētas vienas vienības transportēšanai nepieciešamās izmaksas  $c_{i,j}$  un tā caurlaides spēja  $u_{i,j}$ .  $P$  apzīmē visu maršrutu, kas iet no virsotnes  $s$  uz virsotni  $w$ , kopu, taču  $x(p)$  apzīmē plūsmu pa maršrutu  $p$ . Modelējot situāciju, uzskata, ka  $s$  ir grafa izteka un  $w$  ir grafa ieteka, t.i., var neņemt vērā lokus, kas ir vērsti uz virsotni  $s$ , un lokus, kas ir vērsti no virsotnes  $w$ .

Balstoties uz ieviestajiem apzīmējumiem, optimālās plūsmas problēmu viena veida preču transportēšanas gadījumā ir sekojoša:

$$\min \sum_{p \in P} c(p)x(p) \quad (1.2.1)$$

$$\begin{cases} \sum_{p \in P} \delta_{i,j}(p)x(p) \leq u_{i,j}, & (i, j) \in A, \\ \sum_{p \in P} x(p) \geq d, \\ x(p) \geq 0, & p \in P, \end{cases} \quad (1.2.2)$$

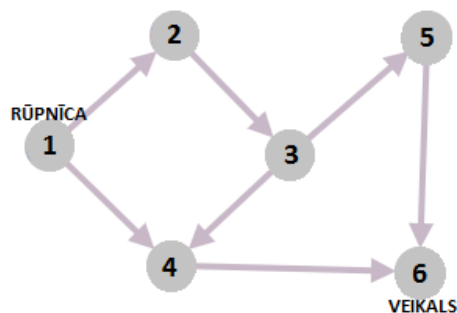
kur  $d$  ir pieprasījums pēc preces, kas jāapmierina, un

$$c(p) = \sum_{(i,j) \in A} c_{i,j} \delta_{i,j}(p) = \sum_{(i,j) \in p} c_{i,j}$$

ir izmaksas, kas veidojas transportējot pieprasīto preces daudzumu pa visiem maršrutiem. Indikatormainīgais  $\delta_{i,j}$  parāda to, vai loks pieder maršrutam  $p$  (vērtība 1) vai nepieder maršrutam  $p$  (vērtība 0).

Mērķis ir iegūt vismazākās iespējamās transportēšanas izmaksas, kas rodas, transportējot pieprasīto preces daudzumu. Pirmais lineārās vienādojumu sistēmas ierobežojumus nodrošina to, ka preces plūsma uz loka  $(i, j)$  nepārsniedz tā caurlaides spēju, taču otrais ieviests, lai tiktu apmierināts pieprasījums pēc preces. Uzdevums ir atrast tādu preču plūsmu, kas apmierinātu pieprasījumu, iekļautos caurlaides spējas nosacījumos un tajā pašā laikā veidotu vismazākās izmaksas.

**Piemērs.** Dots transporta tīkls ar 6 virsotnēm,  $V = \{1, 2, 3, 4, 5, 6\}$ . Nepieciešams nogādāt 10 preces vienības,  $d = 10$ , no rūpnīcas uz veikalu, patērējot pēc iespējas mazāk līdzekļu. Transporta tīkla ilustrācija dota 1.2.1. attēlā.



Att. 1.2.1: Transporta tīkla ilustrācija viena veida preču optimālās plūsmas problēmas piemēram

Loka  $(i, j) \in A$  caurlaides spēja  $u_{i,j}$  un izmaksas par vienu transportējamo vienību  $c_{i,j}$  dotas 1.2.1. tabulā.

Tabula 1.2.1: Transportēšanas izmaksas un loku caurlaides spēja viena veida preču optimālās plūsmas problēmas piemēram

$(i, j)$	$c_{i,j}$	$u_{i,j}$
(1, 2)	1	15
(1, 4)	5	15
(2, 3)	2	1
(3, 4)	15	1
(3, 5)	3	8
(4, 6)	4	15
(5, 6)	15	1

### 1.3. Optimālās plūsmas problēma vairāku veidu precēm

Tiek apskatīta vairāku veidu preču optimālās plūsmas problēma. Transporta tīklā eksistē vairākas iztekas, no kurienes sākas noteikta veida preces transportēšana, un vairākas ietēkas, uz kurieni noteikta veida preces tiek transportētas. Vairums transportēšanas parametru ir atkarīgi no preces veida.

Doti  $K$  preču veidi, kur katra veida precei  $k$ ,  $k = 1, \dots, K$ , ir definēta sava izteka  $s^k$  un ietēka  $w^k$ . Katram lokam  $(i, j) \in A$  ir dotas transportēšanas izmaksas atkarībā no preces veida  $c_{i,j}^k$ ,  $k = 1, \dots, K$ , un loka kopējā caurlaides spēja  $u_{i,j}$ .  $P^k$  apzīmē visu no virsotnes  $s^k$  uz virsotni  $w^k$  vērsto maršrutu kopu, savukārt  $x(p)$  apzīmē plūsmu pa maršrutu  $p$ , kur  $p \in P^k$ .

Vairāku veidu preču optimālās plūsmas problēmu formulē sekojoši:

$$\min \sum_{k=1}^K \sum_{p \in P^k} c(p)x(p), \quad (1.3.1)$$

$$\begin{cases} \sum_{k=1}^K \sum_{p \in P^k} \delta_{i,j}(p)x(p) \leq u_{i,j}, & (i,j) \in A, \\ \sum_{p \in P^k} x(p) \geq d^k, & k = 1, \dots, K, \\ x(p) \geq 0, & k = 1, \dots, K, p \in P^k, \end{cases} \quad (1.3.2)$$

kur  $d^k$  ir pieprasījums pēc  $k$ -tā veida preces, kas jāizpilda, un

$$c(p) = \sum_{(i,j) \in A} c_{i,j}^k \delta_{i,j}(p) = \sum_{(i,j) \in p} c_{i,j}^k, \quad k = 1, \dots, K,$$

ir izmaksas  $k$ -tā veida preces transportēšanai pa maršrutu  $p$ .

Mērķis ir atrast tādu atrisinājumu, pie kura mērķa funkcijas vērtība sasniedz savu minimumu, jeb, transportējot visa veida preču pieprasīto daudzumu, tiek radītas vismazākās izmaksas. Pirmais nosacījums nosacījumu sistēmā (1.3.2) nodrošina to, ka visu veida preču plūsmu summa uz loka  $(i, j)$  ir mazāka kā attiecīgā loka caurlaides spēja. Nosacījumu sistēmas (1.3.2) otrais nosacījums ir veidots, lai apmierinātu katras patēriņa preces pieprasījumu.

**Piemērs.** Atbilstošās problēmas ilustrācijai 1.3.1. attēlā dots transporta tīkls, kas sastāv no 13 virsotnēm un 31 orientēta loka.



*Att. 1.3.1: Transporta tīkla ilustrācija vairāku veidu preču optimālās plūsmas problēmas piemēram*

Uzdevums ir nogādāt divu veidu preces,  $K = 2$ , no to sākuma virsotnes uz beigu virsotni-  $s^1 = 1, w^1 = 13$  un  $s^2 = 1, w^2 = 12$ , radot vismazākās izmaksas. Loku caurlaides spēja un izmaksas par katras preces vienības transportēšanu pa loku dotas 1.3.1. tabulā.

**Tabula 1.3.1: Loku caurlaides spēja un transportēšanas izmaksas vairāku veidu preču optimālās plūsmas problēmas piemēram**

$(i, j)$	$c_{i,j}^1$	$c_{i,j}^2$	$u_{i,j}$
(1, 3)	20	35	75
(1, 4)	30	22	45
(1, 5)	66	55	93
(1, 6)	28	40	47
(2, 3)	47	40	42
(2, 4)	6	60	85
(2,5)	99	75	53
(2, 6)	58	60	20
(3, 7)	42	39	67
(3, 8)	52	65	84
(3, 9)	33	25	2
(4, 7)	43	50	68
(4, 8)	23	41	38
(4, 9)	58	50	83
(4, 10)	76	80	50
(4, 11)	53	40	71
(5, 7)	64	50	43
(5, 9)	21	40	30
(6, 8)	38	71	19
(6, 10)	78	51	19
(6, 11)	68	66	68
(7, 12)	46	40	30
(7, 13)	57	50	54
(8, 12)	79	70	15
(8, 13)	6	2	70
(9, 12)	60	65	38
(9, 13)	5	10	86
(10, 12)	42	40	85
(10, 13)	30	25	59
(11, 12)	87	80	50

## 1.4. Optimālās plūsmas problēma ar nestriktiem parametriem

Tiek aplūkota vairāku veidu preču optimālās plūsmas problēma ar nestriktiem parametriem. Tā ir 1.3. apakšnodaļā definētās problēmas modifikācija.

Gadījumā, kad transportēšanas izmaksas pa lokiem, loku caurlaides spēja un preču pieprasījums nav konkrēti zināmi, minētie lielumi tiek izteikti nestriktos skaitļos. Balstoties uz 1.3. apakšnodaļā izveidoto lineārās programmēšanas uzdevumu, vairāku veidu preču plūsmas problēma ar nestriktiem parametriem definē sekojoši:

$$\min \sum_{k=1}^K \sum_{p \in P^k} \tilde{c}(p)x(p), \quad (1.4.1)$$

$$\begin{cases} \sum_{k=1}^K \sum_{p \in P^k} \delta_{i,j}(p)x(p) \preceq \tilde{u}_{i,j}, & (i, j) \in A, \\ \sum_{p \in P^k} x(p) \succeq \tilde{d}^k, & k = 1, \dots, K, \\ x(p) \geq 0, & k = 1, \dots, K, p \in P^k, \end{cases} \quad (1.4.2)$$

kur  $\tilde{c}(p)$ ,  $\tilde{u}_{i,j}$  un  $\tilde{d}^k$  ir nestrikta lielumi.

Lai rastos lielāka izpratne par uzdevumu specifiku nestriktu parametru gadījumā, nākamā nodaļa tiks veltīta nestriktu kopu un lielumu teorijai. Nodaļa saturēs arī problēmas nostādni ilustrējošus piemērus.

## 2 Nestrikti lielumi un to lietojumi optimālās plūsmas problēmā

Darbā tiek apskatīta vairāku veidu preču optimālās plūsmas problēma, kur parametri ir uzdoti nestriktos skaitļos. Lai radītu izpratni par nestriktiem skaitļiem, turpmāk tiks aplūkoti nestriktas kopas un nestrikta lieluma jēdzieni.

### 2.1. Nestriktas kopas

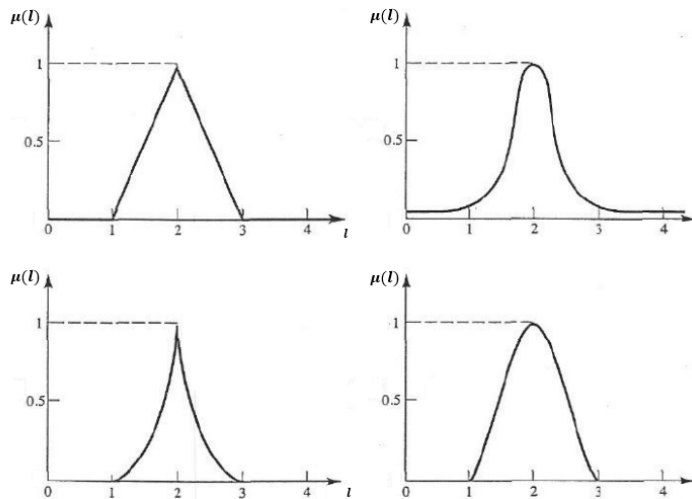
Pieņem, ka  $L$  ir reāla kopa. Aplūkojot klasisku reālas kopas  $L$  apakškopu  $T$ , katra punkta piederību kopai var raksturot, lietojot piederības funkciju:

$$\mu_T(l) = \begin{cases} 1, & \text{ja } l \in T, \\ 0, & \text{ja } l \notin T. \end{cases}$$

Funkcija pieņem vērtību 1, ja kopas punkts  $l$  pieder apakškopai  $T$ , un 0, ja attiecīgais kopas punkts nepieder apakškopai. Klasiskās apakškopas jēdzienu var uzskatīt par nestriktas apakškopas jēdziena vispārinājumu.

Par kopas  $L$  nestriktu apakškopu sauc kopu  $\tilde{T}$ , kura ir uzdota ar piederības funkciju  $\mu_{\tilde{T}} : L \rightarrow [0, 1]$ . Piederības funkcija nestriktai kopai pieņem vērtībās intervālā  $[0, 1]$ , šī vērtība tiek interpretēta kā piederības pakāpe, ar kādu punkts  $l$  pieder kopai  $\tilde{T}$ .

Grafiski attēlojot piederības funkciju nestriktai kopai, horizontālā ass  $0l$  raksturo kopas punktus, turpretim vertikālā ass attēlo piederības pakāpi, ar kādu punkts  $l$  pieder kopai  $\tilde{T}$  katram  $l \in L$ . [8]. Daži no piederības funkciju grafikiem attēloti 2.1.1. attēlā.



Att. 2.1.1: Piederības funkciju piemēri

**6. definīcija.** [8] Nestriktu kopu  $\tilde{T}$  sauc par normālu, ja

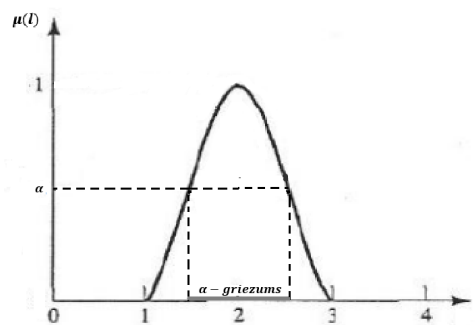
$$\sup_{l \in L} \mu_{\tilde{T}}(l) = 1.$$

Nestrikta kopa ir tukša tad un tikai tad, ja tās piederības funkcija ir vienāda ar 0 kopā  $L$ . Divas nestrikta kopas  $\tilde{H}$  un  $\tilde{D}$  ir vienādas (pieraksta  $\tilde{H} = \tilde{D}$ ) tad un tikai tad, ja to piederības funkcijas  $\mu_{\tilde{H}}(l) = \mu_{\tilde{D}}(l)$  katram  $l \in L$ . [8]

Apakškopas  $\tilde{T}$   $\alpha$ -griezumu apzīmē ar  $[\tilde{T}]_{\alpha}$  un definē sekojoši [4]:

$$[\tilde{T}]_{\alpha} = \begin{cases} \{l \in L | \mu_{\tilde{T}}(l) \geq \alpha\} & \alpha \in (0, 1], \\ cl\{l \in L | \mu_{\tilde{T}}(l) > 0\} & \alpha = 0, \end{cases}$$

kur  $cl$  ir attiecīgās kopas slēgums. Apakškopas  $\tilde{T}$   $\alpha$ -griezuma ilustrāciju var aplūkot 2.1.2. attēlā.



Att. 2.1.2: Kopas  $\tilde{T}$   $\alpha$ -griezums

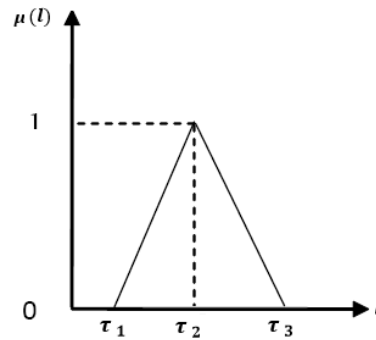
## 2.2. Nestrikta lielumi un to veidi

Turpmāk tiks aplūkoti nestrikta lielumi, to speciālgadījumi, ar ko autors sastopas, risinot konkrētus vairāku veidu preču optimālās plūsmas problēmas piemērus.

**7. definīcija.** [4] Par nestriktu lielumu sauc nestriktu kopu  $\tilde{T}$ , kura ir normāla un kuras  $\alpha$ -griezums  $[\tilde{T}]_\alpha$  ir ierobežots, slēgts un izliekts katram  $\alpha \in (0, 1]$ .

**8. definīcija.** [6] Pieņem, ka  $\tau_1, \tau_2, \tau_3 \in \mathbb{R}$  un  $-\infty < \tau_1 < \tau_2 < \tau_3 < \infty$ , tad par trijstūrveida nestriktu lielumu sauc nestriktu lielumu  $\tilde{\tau} = (\tau_1, \tau_2, \tau_3)$ , kura piederības funkciju ir formā:

$$\mu_{(\tau_1, \tau_2, \tau_3)}(l) = \begin{cases} \frac{l - \tau_1}{\tau_2 - \tau_1}, & l \in [\tau_1, \tau_2], \\ \frac{\tau_3 - l}{\tau_3 - \tau_2}, & l \in [\tau_2, \tau_3], \\ 0, & l < \tau_1 \text{ un } l > \tau_3. \end{cases}$$



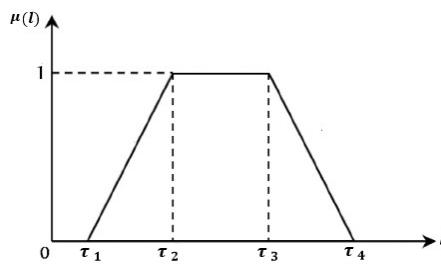
Att. 2.2.1: Trijstūrveida nestrikta lieluma piederības funkcija

Trijstūrveida nestriktu lielumu var interpretēt, kā skaitli, kas ir *aptuveni vienāds ar*  $\tau_2$ , vai *skaitlis*  $\tau_2$  *atrodas intervālā*  $[\tau_1, \tau_3]$ . Piederības funkcijas grafiku var aplūkot 2.2.3. attēlā. Intervālā  $[\tau_1, \tau_3]$  sastāv no divām taisnām līnijām kopā veidojot trijstūri ar  $0l$  asi, kamēr tas ir vienāds ar 0 ārpus intervāla  $[\tau_1, \tau_3]$ . [6]

Pastāv gadījumi, kad trijstūrveida nestrikta lieluma vērtības sakrīt, t.i.,  $\tau_1 = \tau_2$  vai  $\tau_2 = \tau_3$ , līdz ar to piederības funkcija ir citādākā izskatā. Šāds gadījums darbā aplūkots netiek.

**9. definīcija.** [6] Pieņem, ka  $\tau_1, \tau_2, \tau_3, \tau_4 \in \mathbb{R}$  un  $-\infty < \tau_1 < \tau_2 < \tau_3 < \tau_4 < \infty$ , tad trapecveida nestriktu lielumu sauc nestriktu lielumu  $\tilde{\tau} = (\tau_1, \tau_2, \tau_3, \tau_4)$ , kura piederības funkciju ir formā:

$$\mu_{(\tau_1, \tau_2, \tau_3, \tau_4)}(l) = \begin{cases} \frac{l - \tau_1}{\tau_2 - \tau_1}, & l \in [\tau_1, \tau_2], \\ 1, & l \in [\tau_2, \tau_3], \\ \frac{\tau_4 - l}{\tau_4 - \tau_3}, & l \in [\tau_3, \tau_4], \\ 0, & l < \tau_1 \text{ un } l > \tau_4 \end{cases}$$

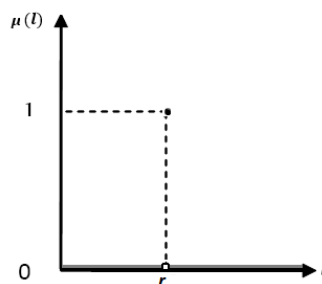


Att. 2.2.2: Trapecveida nestrikta lieluma piederības funkcija

Trapecveida nestrikts lielums ir trijstūrveida nestrikta lieluma vispārinājums, turklāt, trijstūrveida nestrikts lielums  $\tilde{\tau} = (\tau_1, \tau_2, \tau_3)$  var tikt uzskatīts, kā trapecveida nestrikta lieluma speciālgadījums, ja trapecveida nestrikta lieluma vērtības  $\tau_2 = \tau_3$ . [6]

Tā kā problēmas nostādnē nestrikti lielumi tiek salīdzināti ar reāliem skaitļiem, definēsim reālu skaitli kā nestrikta skaitļa speciālgadījumu. Jebkurš reāls skaitlis  $r \in \mathbb{R}$  tiek uzskatīts par nestriktu lielumu ar piederības funkciju formā

$$\mu_r(l) = \begin{cases} 1, & l = r, \\ 0, & l \neq r. \end{cases}$$



Att. 2.2.3: Reāla skaitļa piederības funkcija

## 2.3. Darbības ar nestriktiem lielumiem

Aplūkotā vairāku veidu preču optimālās plūsmas problēma ar nestriktiem parametriem iekļauj tādas algebriskas darbības kā pozitīvu nestriktu skaitļu saskaitīšanu, nestrikta lieluma reizināšanu ar naturālu skaitli un nestriktu lielumu salīdzināšanu. Turpinājumā definēsim šīs algebriskās darbības.

**10. definīcija.** [5] Pieņem, ka  $\tilde{h} = (h_1, h_2, h_3, h_4)$  un  $\tilde{d} = (d_1, d_2, d_3, d_4)$  ir nestrikti lielumi, kur  $h_i, d_i > 0, i = 1, 2, 3, 4$ . Tad nestriktu lielumu saskaitīšanu apzīmē ar  $\tilde{+}$  un definē sekojoši:

$$\tilde{h} \tilde{+} \tilde{d} = (h_1 + d_1, h_2 + d_2, h_3 + d_3, h_4 + d_4).$$

**11. definīcija.** [5] Pieņem, ka  $\tilde{h} = (h_1, h_2, h_3, h_4)$  ir nestrikts lielums un  $r \in \mathbb{N}$ . Tad nestrika lieluma reizināšanu ar naturālu skaitli definē sekojoši:

$$r \cdot \tilde{h} = (r \cdot h_1, r \cdot h_2, r \cdot h_3, r \cdot h_4).$$

Lai varētu salīdzināt nestriktus lielumus  $\tilde{h}$  un  $\tilde{d}$ , tiek izmantotas nestrikta sakarības. Šīs sakarības uzskata par klasiskās lineārās sakarības "≤" nestriktu turpinājumu.

**12. definīcija.** [3] Pieņem, ka  $\tilde{h}$  un  $\tilde{d}$  ir nestrikta lielumi ar piederības funkcijām  $\mu_{\tilde{h}}$  un  $\mu_{\tilde{d}}$  attiecīgi. Tad varbūtības sakarību un nepieciešamības sakarību definē sekojoši:

$$\Pi(\tilde{h} \preceq \tilde{d}) = \sup_{x \leq y} \{ \min(\mu_{\tilde{h}}(x), \mu_{\tilde{d}}(y)) \},$$

$$N(\tilde{h} \prec \tilde{d}) = \inf_{x > y} \{ \max(1 - \mu_{\tilde{h}}(x), 1 - \mu_{\tilde{d}}(y)) \}.$$

Ņemot vērā to, ka, strādājot ar nestriktām sakarībām, lieto piederības funkcijas, piederības funkcijām izmantosim šādus apzīmējumus:

$$\mu_{\Pi}(\tilde{h}, \tilde{d}) = \Pi(\tilde{h} \preceq \tilde{d}) = (\tilde{h} \preceq^{\Pi} \tilde{d}), \quad (2.3.1)$$

$$\mu_N(\tilde{h}, \tilde{d}) = N(\tilde{h} \prec \tilde{d}) = (\tilde{h} \prec^N \tilde{d}). \quad (2.3.2)$$

**2. apgalvojums.** [3] Pieņem, ka  $\tilde{h}, \tilde{d}$  ir nestrikta lielumi un  $\alpha \in (0, 1]$ . Tad

$$\begin{aligned} \Pi(\tilde{h} \preceq \tilde{d}) \geq \alpha &\Leftrightarrow \inf[\tilde{h}]_{\alpha} \leq \sup[\tilde{d}]_{\alpha}, \\ N(\tilde{h} \prec \tilde{d}) \geq \alpha &\Leftrightarrow \sup[\tilde{h}]_{1-\alpha} \leq \inf[\tilde{d}]_{1-\alpha}, \\ \Pi(\tilde{h}, \tilde{d}) < \alpha &\Leftrightarrow \sup[\tilde{d}]_{\alpha} < \inf[\tilde{h}]_{\alpha}, \\ N(\tilde{h}, \tilde{d}) < \alpha &\Leftrightarrow \inf[\tilde{d}]_{1-\alpha} < \sup[\tilde{h}]_{1-\alpha}, \end{aligned}$$

kur  $\inf[\tilde{h}]_{\alpha} = (\tilde{h})_{\alpha}^L$  un  $\sup[\tilde{h}]_{\alpha} = (\tilde{h})_{\alpha}^R$ . (Skat. 3.2.1. attēlu)

## 2.4. Optimālās plūsmas problēmas piemēri

Lai uzskatāmi parādītu nestriktu lielumu izmantošanu darbā, turpinājumā tiek definēti piemēri galvenajai problēmai.

**1. piemērs.** Tiek apskatīts transports tīkls ar 13 virsotnēm un 31 loku. [1]



Att. 2.4.1: Transporta tīkla ilustrācija 1. piemēram

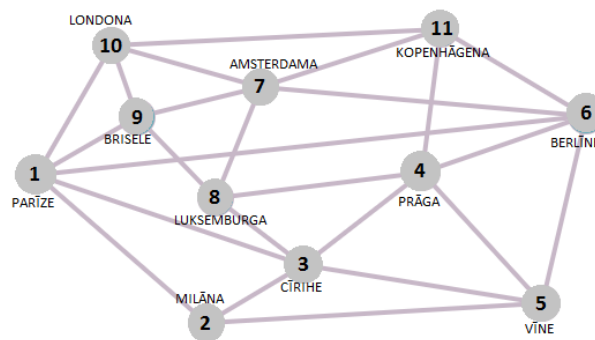
Mērķis ir nogādāt 2 veidu preces,  $K = 2$ , no virsotnes 1 uz virsotni 13 un no virsotnes 1 uz virsotni 12, attiecīgi, radot vismazākās izmaksas. Preču pieprasījums ir dots nestriktos skaitļos- $\tilde{d}^1 = (100, 120, 135, 150)$  un  $\tilde{d}^2 = (60, 61, 75, 100)$ . Transporta tīklu raksturojošie lielumi, tas ir, loku caurlaides spēja un  $k$ -tās,  $k = 1, 2$ , preces vienības transportēšanas izmaksas pa loku  $(i, j)$  nestriktos skaitļos dotas 2.4.1. tabulā.

**Tabula 2.4.1: Loku caurlaides spēja un transportēšanas izmaksas nestriktos skaitļos 1. piemēram**

$(i, j)$	$\tilde{c}_1$	$\tilde{c}_2$	$\tilde{u}_{i,j}$
(1, 3)	(20, 30, 37, 45)	(35, 40, 41, 45)	(75, 85, 93, 95)
(1, 4)	(30, 32, 36, 42)	(22, 25, 40, 50)	(45, 47, 53, 56)
(1, 5)	(66, 72, 81, 89)	(55, 60, 70, 95)	(93, 99, 107, 114)
(1, 6)	(28, 33, 38, 45)	(40, 41, 55, 70)	(47, 52, 59, 62)
(2, 3)	(47, 56, 60, 63)	(40, 45, 60, 70)	(42, 51, 54, 59)
(2, 4)	(6, 14, 22, 25)	(60, 62, 80, 98)	(85, 93, 96, 97)
(2,5)	(99, 104, 109, 112)	(75, 77, 80, 97)	(53, 58, 61, 71)
(2, 6)	(58, 58, 60, 65)	(60, 61, 79, 85)	(20, 20, 25, 31)
(3, 7)	(42, 50, 57, 64)	(39, 40, 55, 60)	(67, 75, 82, 86)
(3, 8)	(52, 56, 64, 67)	(65, 70, 85, 92)	(84, 88, 91, 96)
(3, 9)	(33, 39, 39, 47)	(25, 30, 42, 55)	(2, 8, 16, 19)
(4, 7)	(43, 51, 58, 64)	(50, 58, 62, 80)	(68, 76, 82, 86)
(4, 8)	(23, 32, 36, 40)	(41, 43, 51, 67)	(38, 47, 51, 53)
(4, 9)	(58, 65, 73, 80)	(50, 51, 60, 67)	(83, 90, 97, 103)
(4, 10)	(76, 78, 83, 88)	(80, 83, 91, 97)	(50, 52, 57, 65)
(4, 11)	(53, 57, 64, 68)	(40, 43, 53, 60)	(71, 75, 79, 84)
(5, 7)	(64, 73, 77, 84)	(50, 60, 71, 77)	(43, 52, 59, 65)
(5, 9)	(21, 30, 33, 39)	(40, 43, 54, 61)	(30, 39, 45, 47)
(6, 8)	(38, 42, 44, 52)	(71, 81, 88, 90)	(19, 23, 31, 35)
(6, 10)	(78, 87, 89, 99)	(51, 60, 72, 80)	(19, 28, 38, 46)
(6, 11)	(68, 69, 76, 81)	(66, 70, 71, 78)	(68, 69, 74, 81)
(7, 12)	(46, 50, 53, 62)	(40, 42, 51, 60)	(30, 34, 43, 48)
(7, 13)	(57, 65, 70, 72)	(50, 57, 60, 60)	(54, 62, 64, 70)
(8, 12)	(79, 79, 81, 91)	(70, 71, 76, 80)	(15, 15, 25, 33)
(8, 13)	(6, 7, 14, 17)	(2, 2, 12, 30)	(70, 71, 74, 75)
(9, 12)	(60, 62, 66, 69)	(65, 70, 72, 86)	(38, 40, 43, 49)

(9, 13)	(5, 7, 16, 25)	(10, 13, 20, 25)	(86, 88, 97, 98)
(10, 12)	(42, 48, 57, 64)	(40, 41, 45, 50)	(85, 91, 98, 102)
(10, 13)	(30, 33, 39, 40)	(25, 33, 36, 39)	(59, 62, 63, 66)
(11, 12)	(87, 89, 94, 94)	(80, 87, 89, 90)	(50, 52, 52, 61)
(11, 13)	(2, 2, 11, 20)	(23, 24, 25, 30)	(90, 90, 99, 99)

**2. piemērs.** Ir dots preču transportēšanas tīkls *COST 239 European Optical Network*, kas sastāv no 11 virsotnēm un 25 abos virzienos vēršiem lokiem. [1]



Att. 2.4.2: Transporta tīkla ilustrācija 2. piemēram

Attēlā 2.4.2. ilustrēts telekomunikācijai nepieciešamās informācijas transporta tīkls, kas transportē tādas informācijas nesējus kā datus, audio, vizuālos failus un citus. Tiek aplūkotas četras dažāda veida preces,  $K = 4$ . Tabulā 2.4.2. dots  $\tilde{d}^k$  – pieprasījums pēc katra veida preces nestriktos skaitļos.

Tabula 2.4.2: Pieprasījums pēc precēm nestriktos skaitļos 2. piemēram

Prece	Izteka	Ieteka	Pieprasījums
1	1	11	(15, 17, 18, 19)
2	10	5	(13, 16, 17, 19)
3	1	5	(2, 3, 4, 6)
4	10	6	(20, 23, 25, 27)

Prece 1 ir jānogādā no *Parīzes* uz *Kopenhāgenu*, prece 2- no *Londonas* uz *Vīni*, prece 3- no *Parīzes* uz *Vīni*, prece 4- no *Londonas* uz *Berlīni*.

Katra loka caurlaides spēja  $\tilde{u}_{i,j}$  un katra veida preces vienības transportēšanas izmaksas uz loka  $\tilde{c}_{i,j}^k$ ,  $k = 1, 2, 3, 4$ , dotas 2.4.3. un 2.4.4. tabulās.

Tabula 2.4.3: Loku caurlaides spējas nestriktos skaitļos 2. piemēram

Loka numurs	Sākuma virsotne	Beigu virsotne	Caurlaides spēja
1	1	2	(18, 20, 20 ,23)
2	1	3	(17, 19, 21, 24)
3	1	6	(37, 40, 43, 44)
4	1	9	(16, 18, 20, 23)
5	1	10	(27, 30, 31, 34)
6	2	3	(17, 20, 22, 25)
7	2	5	(8, 10, 13, 14)
8	2	9	(9,10, 14, 16)
9	3	4	(2, 2.5, 2.5 4)
10	3	5	(7, 9, 10, 13)
11	3	8	(1.5, 2, 2.5, 4)
12	4	5	(2, 3, 3.5, 5)
13	4	6	(9, 10, 12 13)
14	4	8	(1.5, 2, 2, 4)
15	4	11	(2, 3, 3.5, 4)
16	5	6	(27, 30, 33, 34)
17	6	7	(18, 20, 21, 23)
18	6	11	(9, 10, 10, 13)
19	7	8	(1.5, 2, 3, 3)
20	7	9	(8, 10, 10, 12)
21	7	10	(17, 19, 20, 21)
22	7	11	(2, 2.5, 2.5, 3.5)
23	8	9	(2, 2, 3, 4)
24	9	10	(8, 9, 10, 11)
25	10	11	(7, 9, 9, 10)
26	2	1	(18, 21, 22, 23)
27	3	1	(20, 21, 23, 23)
28	6	1	(35, 37, 40, 42)
29	9	1	(17, 19, 22, 24)
30	10	1	(28, 31, 31, 33)
31	3	2	(17, 20 21, 22)

32	5	2	(7, 9, 10, 11)
33	9	2	(9, 10, 11, 13)
34	4	3	(2, 2, 2.5, 3)
35	5	3	(8, 9, 10, 12)
36	8	3	(2, 2.5, 3, 4)
37	5	4	(1.5, 2, 2.5, 3)
38	6	4	(18, 10, 12, 13)
39	8	4	(2, 2.5, 3.5, 4)
40	11	4	(2, 2.5, 3.5, 4)
41	6	5	(27, 30, 32, 33)
42	7	6	(18, 20, 21, 23)
43	11	6	(9, 10, 12, 13)
44	8	7	(2, 2.5, 3.5, 4)
45	9	7	(9, 11, 11, 13)
46	10	7	(18, 19, 21, 22)
47	11	7	(2, 2.5, 3, 4)
48	9	8	(2, 2, 3.5, 4)
49	10	9	(8, 9, 10, 12)
50	11	10	(9, 11, 12, 14)

Tabula 2.4.4: Preču transportēšanas izmaksas 2. piemēram

(i, j)	(j, i)	$\tilde{c}^1$	$\tilde{c}^2$	$\tilde{c}^3$	$\tilde{c}^4$
1	26	(190, 210, 210, 230)	(290, 300, 300, 390)	(261, 270, 270, 351)	(319, 330, 330, 429)
2	27	(269, 280, 280, 289)	(174.5, 270, 270, 274.5)	(157.05, 243, 243, 247.05)	(191.95, 297, 297, 292.05)
3	28	(463, 490, 490, 496)	(296.5, 310, 310, 313)	(266.85, 279, 279, 281.7)	(325.15, 341, 341, 344.3)
4	29	(180, 190, 190, 240)	(250, 255, 255, 280)	(255, 229.5, 229.5, 252)	(275, 280.05, 20.05, 308)
5	30	(940, 970, 970, 990)	(935, 950, 950, 960)	(841.5, 855, 855, 864)	(88, 104.5, 104.05, 115.5)
6	31	(224, 320, 320, 327)	(42, 55, 55, 108.5)	(37.8, 49.5, 49.5, 97.65)	(46.2, 60.5, 60.5, 119.35)
7	32	(300, 315, 315, 330)	(252.5, 260, 260, 267.5)	(227.25, 234, 234, 240.75)	(277.75, 286, 286, 294.25)
8	33	(246, 276, 276, 306)	(251, 266, 266, 281)	(222.9, 239.4, 239.4, 255.9)	(276.1, 292.6, 292.6, 309.1)
9	34	(140, 157, 157, 253)	(248.5, 257, 257, 355)	(228.95, 231.3, 231.3, 319.5)	(283.35, 282.7, 282.7, 390.5)
10	35	(227, 245, 245, 267)	(79, 88, 88, 99)	(71.1, 79.2, 79.2, 89.1)	(87.9, 96.8, 96.8, 108.9)
11	36	(215, 233, 233, 255)	(234, 243, 243, 254)	(210.6, 218.7, 218.7, 228.6)	(257.4, 267.3, 267.3, 279.4)
12	37	(82, 91, 91, 102)	(126.5, 131, 131, 136.5)	(113.85, 117.9, 117.9, 122.85)	(139.15, 144.1, 144.1, 150.15)
13	38	(110, 140, 140, 160)	(105, 120, 120, 130)	(94.5, 108, 108, 117)	(115.5, 132, 132, 143)
14	39	(310, 330, 330, 335)	(300, 310, 310, 312.5)	(270, 279, 279, 281.25)	(330, 341, 341, 343.75)
15	40	(210, 240, 240, 270)	(235, 250, 250, 265)	(211.5, 225, 255, 238.5)	(258.5, 275, 275, 291.5)
16	41	(110, 160, 160, 190)	(535, 560, 560, 575)	(481.5, 504, 504, 517.5)	(588.5, 616, 616, 632.5)
17	42	(264, 284, 284, 314)	(276, 274, 274, 289)	(237.6, 246.6, 246.6, 260.1)	(290.4, 301.4, 301.4, 317.9)
18	43	(255, 267, 267, 285)	(241, 247, 247, 256)	(216.9, 222.3, 222.3, 230.4)	(265.1, 271.7, 271.7, 281.6)
19	44	(330, 342, 342, 355)	(346, 352, 352, 358.5)	(311.4, 316.8, 316.8, 322.65)	(380.6, 387.2, 387.2, 394.35)
20	45	(203, 213, 213, 233)	(190, 195, 195, 205)	(171, 175.5, 175.5, 184.5)	(209, 214.5, 214.5, 255.5)
21	46	(144, 156, 156, 164)	(150, 156, 156, 160)	(135, 140.4, 140.4, 144)	(165, 171.6, 171.6, 176)
22	47	(248, 270, 270, 288)	(261, 272, 272, 281)	(234.9, 244.8, 244.8, 252.9)	(287.1, 299.2, 299.2, 309.1)
23	48	(240, 247, 247, 255)	(243.5, 247, 247, 251)	(219.15, 222.3, 222.3, 225.9)	(267.85, 271.7, 271.7, 276.1)
24	49	(130, 142, 142, 160)	(146, 152, 152, 161)	(131.4, 136.8, 136.8, 144.9)	(160.6, 167.2, 167.2, 177.1)
25	50	(150, 210, 210, 330)	(120, 150, 150, 210)	(108, 135, 135, 189)	(132, 165, 165, 231)

## 3 Nestriktā lineārā programmēšana

Vairāku veidu preču optimālās plūsmas problēma ar nestriktiem parametriem tiek izteikta ar lineārās programmēšanas uzdevuma palīdzību. Tā kā problēma ietver nestriktus parametrus, nepieciešams izprast, kā ar šiem lielumiem strādāt. Šim nolūkam aplūkosim nestrikto lineāro programmēšanu – lineārās programmēšanas modifikāciju. Šajā nodaļā aplūkotā nestriktās lineārās programmēšanas teorija balstās uz *J.Ramík* rakstu [4].

### 3.1. Nestrikti parametri lineārās programmēšanas uzdevuma nosacījumos

Pieņem, ka  $m, g \in \mathbb{N}$ . Apskata lineārās programmēšanas uzdevumu formā:

$$\begin{cases} \max (c_1x_1 + \dots + c_gx_g) \\ a_{i1}x_1 + \dots + a_{ig}x_g \leq b_i, & i = 1, \dots, m, \\ x_j \geq 0, & j = 1, \dots, g. \end{cases} \quad (3.1.1)$$

**3. apgalvojums.** Pieņem, ka  $\tilde{c}_j, \tilde{a}_{ij}, \tilde{b}_i$  ir nestrikti lielumi ar piederības funkcijām, attiecīgi,  $\mu_{\tilde{c}_j}, \mu_{\tilde{a}_{ij}}, \mu_{\tilde{b}_i}, i = 1, \dots, m, j = 1, \dots, g$ . Tad  $\tilde{c}_1x_1 + \dots + \tilde{c}_gx_g$  un  $\tilde{a}_{i1}x_1 + \dots + \tilde{a}_{ig}x_g$  arī ir nestrikti lielumi.

Pieņem, ka  $\tilde{E}$  ir klasiskās binārās sakarības "≤" nestrikts turpinājums. Uzdevuma (3.1.1) parametru  $a_{ij}, b_j, c_j$  vietā lietosim nestriktus lielumus  $\tilde{a}_{ij}, \tilde{b}_i, \tilde{c}_j, i = 1, \dots, m, j = 1, \dots, g$ . Tad nestrikta lineārās programmēšanas problēma, kas ir saistīta ar klasisko lineārās programmēšanas problēmu (3.1.1), izskatās sekojoši:

$$\begin{cases} \text{"max"} (\tilde{c}_1x_1 + \dots + \tilde{c}_gx_g) \\ (\tilde{a}_{i1}x_1 + \dots + \tilde{a}_{ig}x_g) \tilde{E} \tilde{b}_i, & i = 1, \dots, m \\ x_j \geq 0, & j = 1, \dots, g, \end{cases} \quad (3.1.2)$$

kur vērtība  $\tilde{a}_{i1}x_1 + \dots + \tilde{a}_{ig}x_g$  tiek salīdzināta ar nestrikto lielumu  $\tilde{b}_i$ , izmantojot nestrikto sakarību  $\tilde{E}$ . Uzdevuma nostādnē tiek lietots pieraksts "max", jo pagaidām nav definēts, kādu no mērķa funkcijas vērtībām uzskata par maksimālo nestrikto parametru gadījumā. Darbā ne-

strikta sakarība  $\tilde{E}$  vietā tiks izmantota iespējamības sakarība  $\Pi$  un nepieciešamības sakarība  $N$ .

**13. definīcija.** Pieņem, ka  $\mu_{\tilde{a}_{ij}} : \mathbb{R} \rightarrow [0, 1]$  un  $\mu_{\tilde{b}_i} : \mathbb{R} \rightarrow [0, 1]$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, g$ , ir piederības funkcijas nestriktajiem lielumiem  $\tilde{a}_{ij}$  un  $\tilde{b}_i$ , attiecīgi, un  $\tilde{E}$  ir nestrikts binārās sakarības turpinājums. Nestrikta kopa  $\tilde{X}$ , kuras piederības funkcija katram  $x \in \mathbb{R}^g$  tiek definēta sekojoši:

$$\mu_{\tilde{X}}(x) = \begin{cases} \min\{\mu_{\tilde{E}}(\tilde{a}_{11}x_1 \tilde{+} \dots \tilde{+} \tilde{a}_{1g}x_g, \tilde{b}_1), \dots, \mu_{\tilde{E}}(\tilde{a}_{m1}x_1 \tilde{+} \dots \tilde{+} \tilde{a}_{mg}x_g, \tilde{b}_m)\}, & \text{ja } x_j \geq 0, j = 1, \dots, g, \\ 0, & \text{pretējā gadījumā,} \end{cases} \quad (3.1.3)$$

tiek saukta par iespējamo apgabalu nestrikta lineārās programmēšanas problēmai (3.1.2).

Katram  $\beta \in (0, 1]$  vektors  $x \in [\tilde{X}]_\beta$  tiek saukts par  $\beta$ -iespējamu problēmas (3.1.2) atrisinājumu.

**4. apgalvojums.** Pieņem, ka  $\tilde{a}_{ij}$ ,  $\tilde{b}_i$  ir nestrikti lielumi,  $x_j \geq 0$ ,  $j = 1, \dots, g$ ,  $i = 1, \dots, m$ , un  $\beta \in (0, 1)$ . Pietam  $\preceq^\Pi$ ,  $\prec^N$  ir binārās sakarības " $\leq$ " nestrikti turpinājumi, kas ir definēti (2.3.1)-(2.3.2). Tad katram  $i = 1, \dots, m$  izpildās

$$\mu_{\preceq^\Pi}(\tilde{a}_{i1}x_1 \tilde{+} \dots \tilde{+} \tilde{a}_{ig}x_g, \tilde{b}_i) \geq \beta \Leftrightarrow \sum_{j=1}^g (\tilde{a}_{ij})_\beta^L x_j \leq (\tilde{b}_i)_\beta^R, \quad (3.1.4)$$

$$\mu_{\prec^N}(\tilde{a}_{i1}x_1 \tilde{+} \dots \tilde{+} \tilde{a}_{ig}x_g, \tilde{b}_i) \geq \beta \Leftrightarrow \sum_{j=1}^g (\tilde{a}_{ij})_{1-\beta}^R x_j \leq (\tilde{b}_i)_{1-\beta}^L. \quad (3.1.5)$$

Pierādījums izriet no 2. apgalvojuma pirmajiem diviem punktiem.

**5. sekas.** Pieņem, ka  $\tilde{E} = \preceq^\Pi$ . Vektors  $x = (x_1, \dots, x_g)$  ir  $\beta$ -iespējams atrisinājums problēmai (3.1.2) tad un tikai tad, ja tas ir atrisinājums nevienādību sistēmai

$$\begin{cases} \sum_{j=1}^g (\tilde{a}_{ij})_\beta^L x_j \leq (\tilde{b}_i)_\beta^R, & i = 1, \dots, m, \\ x_j \geq 0, & j = 1, \dots, g. \end{cases}$$

**6. sekas.** Pieņem, ka  $\tilde{E} = \prec^N$ . Vektors  $x = (x_1, \dots, x_g)$  ir  $\beta$ -iespējams atrisinājums problēmai (3.1.2) tad un tikai tad, ja tas ir atrisinājums nevienādību sistēmai

$$\begin{cases} \sum_{j=1}^g (\tilde{a}_{ij})_{1-\beta}^R x_j \leq (\tilde{b}_i)_{1-\beta}^L, & i = 1, \dots, m, \\ x_j \geq 0, & j = 1, \dots, g. \end{cases}$$

## 3.2. No nestriktiem parametriem atkarīgas funkcijas optimizēšana

Šajā nodaļā tiks aplūkota problēmas (3.1.2) mērķa funkcijas maksimizēšana. Lai tiktu galā ar funkcijas  $\tilde{c}_1 x_1 + \dots + \tilde{c}_g x_g = \tilde{z}$  maksimizēšanu, meklē vislabāko nestriktu lielumu  $\tilde{z}$  attiecībā uz nestriktajiem nosacījumiem.

**14. definīcija.** Pieņem, ka  $\tilde{E}$  ir nestrikta sakarība kopā  $\mathbb{R}$ ,  $\alpha \in (0, 1]$  un  $\tilde{h}$  un  $\tilde{d}$  ir nestrikti lielumi, tad

$$\tilde{h}\tilde{E}_\alpha\tilde{d}, \text{ ja } \mu_{\tilde{E}}(\tilde{h}, \tilde{d}) \geq \alpha,$$

kur  $\tilde{E}_\alpha$  sauc par  $\alpha$ -sakarību telpā  $\mathbb{R}$ , kas ir saistīta ar  $\tilde{E}$ ,

$$\tilde{h}\tilde{E}_\alpha^*\tilde{d}, \text{ ja } \mu_{\tilde{E}}(\tilde{h}, \tilde{d}) \geq \alpha \text{ un } \mu_{\tilde{E}}(\tilde{d}, \tilde{h}) < \alpha,$$

kur  $\tilde{E}_\alpha^*$  sauc par striktu  $\alpha$ -sakarību telpā  $\mathbb{R}$ , kas ir saistīta ar  $\tilde{E}$ .

Jāņem vērā, ka  $\tilde{E}_\alpha^*$  un  $\tilde{E}_\alpha$  ir bināras sakarības nestrikto lielumu kopā un ir veidojušās no nestrikta sakarības  $\tilde{E}$  pie piederības pakāpes  $\alpha \in (0, 1]$ . Ja nestrikta sakarība  $\tilde{E}$  ir binārās sakarības  $\leq$  turpinājums un  $h, d \in \mathbb{R}$ , tad  $h\tilde{E}_\alpha d \geq \alpha$  tad un tikai tad, ja  $h \leq d$ , un  $h\tilde{E}_\alpha^* d \geq \alpha$  tad un tikai tad, ja  $h < d$ . [3]

**7. apgalvojums.** Pieņem, ka  $\tilde{h}$  un  $\tilde{d}$  ir nestrikti lielumi,  $\alpha \in (0, 1]$  un  $\tilde{E} = \preceq^\Pi$  ir nestrikta sakarība, kas definēta (2.3.1), tad

$$\begin{aligned} \tilde{h}\tilde{E}_\alpha\tilde{d}, \text{ ja } (\tilde{h})_\alpha^L &\leq (\tilde{d})_\alpha^R, \\ \tilde{h}\tilde{E}_\alpha^*\tilde{d}, \text{ ja } (\tilde{h})_\alpha^R &< (\tilde{d})_\alpha^L. \end{aligned} \quad (3.2.1)$$

**8. apgalvojums.** Pieņem, ka  $\tilde{h}$  un  $\tilde{d}$  ir nestrikti lielumi,  $\alpha \in (0, 1]$  un  $\tilde{E} = \prec^N$  ir nestrikta sakarība, kas definēta (2.3.2), tad

$$\begin{aligned} \tilde{h}\tilde{E}_\alpha\tilde{d}, \text{ ja } (\tilde{h})_{1-\alpha}^R &\leq (\tilde{d})_{1-\alpha}^L, \\ \tilde{h}\tilde{E}_\alpha^*\tilde{d}, \text{ ja } (\tilde{h})_{1-\alpha}^R &\leq (\tilde{d})_{1-\alpha}^L \text{ un } (\tilde{h})_{1-\alpha}^L < (\tilde{h})_{1-\alpha}^R. \end{aligned} \quad (3.2.2)$$

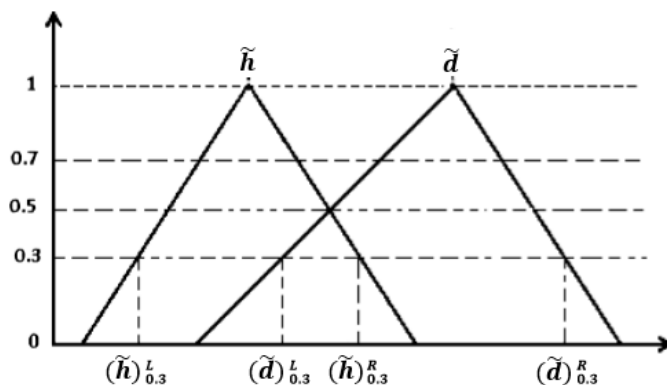
**Piemērs.** Tiek aplūkota divu nestriktu lielumu  $\tilde{h}$  un  $\tilde{d}$  salīdzināšana, lietojot nestrikta sakarības  $\tilde{E}(\tilde{h}, \tilde{d}) \geq \alpha$  un  $\tilde{h}\tilde{E}_\alpha^*\tilde{d}$ .

Pie piederības pakāpes  $\alpha \in (0, 1]$ ,  $\tilde{h}$  nav lielāks par  $\tilde{d}$  ar iespējamības mēru  $\Pi(\tilde{h}, \tilde{d}) \geq \alpha$ , ja minimālā  $[\tilde{h}]_\alpha$  vērtība ir mazāka vai vienāda ar  $[\tilde{d}]_\alpha$  maksimālo vērtību. Citiem vārdiem, izvēlas vērtības  $h$  un  $d$ , kur  $\mu_{\tilde{h}}(h) \geq \alpha$  un  $\mu_{\tilde{d}}(d) \geq \alpha$ , tā, ka  $h \leq d$ . Turklāt, sakarība  $\tilde{h}\Pi_\alpha^*\tilde{d}$  izpildās, ja augstākā  $[\tilde{h}]_\alpha$  vērtība ir stingri mazāka par  $[\tilde{d}]_\alpha$  zemāko vērtību.

Attēlā 3.2.1. diviem nestriktiem lielumiem  $\tilde{h}$  un  $\tilde{d}$  sakarības  $\Pi(\tilde{h}, \tilde{d}) \geq 0.7$ ,  $\tilde{h}\Pi_{0.7}^*\tilde{d}$  un  $\Pi(\tilde{h}, \tilde{d}) \geq 0.5$  izpildās, taču sakarība  $\tilde{h}\Pi_{0.5}^*\tilde{d}$  nav spēkā, jo  $(\tilde{h})_{0.5}^R \not\leq (\tilde{d})_{0.5}^L$ .

Nepieciešamības sakarība  $N(\tilde{h}, \tilde{d}) \geq \alpha$  ir patiesa, ja maksimālā  $[\tilde{h}]_{1-\alpha}$  vērtība ir mazāka vai vienāda ar minimālo  $[\tilde{d}]_{1-\alpha}$  vērtību. Turklāt,  $\tilde{h}N_{\alpha}^*\tilde{d}$  izpildās, ja  $N(\tilde{h}, \tilde{d}) \geq \alpha$  un minimālā  $[\tilde{h}]_{1-\alpha}$  ir stingri mazāka par  $[\tilde{d}]_{1-\alpha}$  maksimālo vērtību.

Attēlā 3.2.1. sakarības  $N(\tilde{h}, \tilde{d}) \geq 0.5$  un  $\tilde{h}N_{0.5}^*\tilde{d}$  izpildās, taču  $N(\tilde{h}, \tilde{d}) \geq 0.7$  un  $\tilde{h}N_{0.7}^*\tilde{d}$  nav spēkā, jo  $(\tilde{h})_{0.3}^R > (\tilde{d})_{0.3}^L$ . [3]



Att. 3.2.1: Nestrikti lielumi  $\tilde{h}$  un  $\tilde{d}$

Definējot problēmas (3.1.2) mērķa funkcijas maksimizēšanu, pieņem, ka nestrikta sakarība  $\tilde{E}$  ir klasiskās bināras sakarības " $\leq$ " turpinājums. Sakarību  $\tilde{E}$  izmanto gan mērķa funkcijai, gan ierobežojumiem, taču izšķir dažādas piederības pakāpes:  $\alpha$ - mērķa funkcijai un  $\beta$ - ierobežojumiem,  $\alpha, \beta \in (0, 1]$ . Ērtībai problēmas (3.1.2) mērķa funkciju pieraksta arī vektoru formā:

$$\tilde{c}_1 x_1 + \dots + \tilde{c}_g x_g = \tilde{c}^T x.$$

**15. definīcija.** Pieņem, ka  $\tilde{c}_j$ ,  $\tilde{a}_{ij}$  un  $\tilde{b}_i$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, g$ , ir nestrikti lielumi,  $\tilde{E}$  ir klasiskās binārās sakarības " $\leq$ " nestrikts turpinājums un  $\alpha, \beta \in (0, 1]$ . Tad  $\beta$ -iespējams atrisinājums problēmai (3.1.2)  $x \in [\tilde{X}]_{\beta}$  tiek saukts par  $(\alpha, \beta)$ -optimālu atrisinājumu problēmai (3.1.2), ja neeksistē tāds  $x' \in [\tilde{X}]_{\beta}$ , ka  $x \neq x'$  un  $\tilde{c}^T x \tilde{E}_{\alpha}^* \tilde{c}^T x'$ , kur  $\tilde{E}_{\alpha}^*$  ir stingra  $\alpha$ -sakarība saistīta ar  $\tilde{E}$ .

**9. lemma.** Pieņem, ka  $\tilde{c}_j$ ,  $j = 1, \dots, g$ , ir nestrikti lielumi,  $\alpha \in (0, 1)$ , nestrikta sakarība  $\tilde{E} = \preceq^{\Pi}$  un  $\tilde{E}_{\alpha}^*$  ir stingra  $\alpha$ -sakarība, kas saistīta ar  $\tilde{E}$ . Vektori  $x = (x_1, \dots, x_g)$ ,  $x' = (x'_1, \dots, x'_g)$ ,  $x_j, x'_j \geq 0$ ,  $j = 1, \dots, g$ , apmierina  $\tilde{c}^T x \tilde{E}_{\alpha}^* \tilde{c}^T x'$  tad un tikai tad, ja

$$\sum_{j=1}^g (\tilde{c}_{\alpha}^R)_j x_j < \sum_{j=1}^g (\tilde{c}_{\alpha}^L)_j x'_j. \quad (3.2.3)$$

*Pierādījums.* Izmantojot rezultātus (3.2.1), iegūst

$$\sup[\tilde{c}^T x]_\alpha = \sum_{j=1}^g (\tilde{c}_j)_\alpha^R x_j < \sum_{j=1}^g (\tilde{c}_j)_\alpha^L x'_j = \inf[\tilde{c}^T x']_\alpha.$$

□

**10. sekas.** Ja nevienādība (3.2.3) izpildās, tad

$$\begin{aligned} \sum_{j=1}^g (\tilde{c}_j)_\alpha^L x_j &< \sum_{j=1}^g (\tilde{c}_j)_\alpha^L x'_j, \\ \sum_{j=1}^g (\tilde{c}_j)_\alpha^R x_j &< \sum_{j=1}^g (\tilde{c}_j)_\alpha^R x'_j. \end{aligned} \quad (3.2.4)$$

**11. lemma.** Pieņem, ka  $\tilde{c}_j$ ,  $j = 1, \dots, g$ , ir nestrikta lielumi,  $\alpha \in (0, 1)$ , nestrikta sakarība  $\tilde{E} = \preceq^N$  un  $\tilde{E}_\alpha^*$  ir stingra  $\alpha$ -sakarība, kas saistīta ar  $\tilde{E}$ . Vektori  $x = (x_1, \dots, x_g)$ ,  $x' = (x'_1, \dots, x'_g)$ ,  $x_j, x'_j \geq 0$ ,  $j = 1, \dots, g$ , apmierina  $\tilde{c}^T x \tilde{E}_\alpha^* \tilde{c}^T x'$  tad un tikai tad, ja

$$\sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x_j \leq \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^L x'_j \text{ un } \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^L x_j < \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x'_j. \quad (3.2.5)$$

*Pierādījums.* Izmantojot rezultātus (3.2.2), iegūst

$$\sup[\tilde{c}^T x]_{1-\alpha} \leq \inf[\tilde{c}^T x']_{1-\alpha} \text{ un } \inf[\tilde{c}^T x]_{1-\alpha} < \sup[\tilde{c}^T x']_{1-\alpha},$$

kas ir ekvivalenti nevienādībām (3.2.5). □

Turpmākie divi apgalvojumi dos pietiekamus nosacījumus  $x^*$  kā nestrikta lineārās programmēšanas problēmas (3.1.2) atrisinājumam.

**12. apgalvojums.** Pieņem, ka  $\tilde{c}_j$ ,  $\tilde{a}_{ij}$ ,  $\tilde{b}_i$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, g$ , ir nestrikta lielumi un  $\alpha, \beta \in (0, 1)$ . Pieņem, ka  $\tilde{X}$  ir iespējamais problēmas (3.1.2) apgabals attiecībā uz  $\tilde{E} = \preceq^\Pi$ , un  $c_j$  ir tāds, ka  $(\tilde{c}_j)_\alpha^L \leq c_j \leq (\tilde{c}_j)_\alpha^R$ ,  $j = 1, \dots, g$ . Ja  $x^* = (x_1^*, \dots, x_g^*)$  ir optimālais atrisinājums lineārās programmēšanas problēmai

$$\begin{aligned} &\max \sum_{j=1}^g c_j x_j, \\ &\begin{cases} \sum_{j=1}^g (\tilde{a}_{ij})_\beta^L x_j \leq (\tilde{b}_i)_\beta^R, & i = 1, \dots, m, \\ x_j \geq 0, & j = 1, \dots, g, \end{cases} \end{aligned} \quad (3.2.6)$$

tad  $x^*$  ir  $(\alpha, \beta)$ -optimāls atrisinājums problēmai (3.1.2).

*Pierādījums.* Pieņem, ka  $x^*$  ir optimāls atrisinājums problēmai (3.2.6). Pierādīsim to, ka  $x^*$  ir  $(\alpha, \beta)$ -optimāls atrisinājums problēmai (3.1.2) attiecībā uz  $\tilde{E} = \preceq^\Pi$ . Balstoties uz rezultātu (3.1.4),  $x^*$  ir  $\beta$ -iespējams atrisinājums problēmai (3.1.2). Pieņem, ka  $x^*$  nav  $(\alpha, \beta)$ -optimāls

atrisinājums problēmai (3.1.2). Tad eksistē tāds  $\beta$ -iespējams atrisinājums  $x' = (x'_1, \dots, x'_g)$ , ka  $\tilde{c}^T x^* \tilde{E}_\alpha^* \tilde{c}^T x'$ . Pēc 2. sekām iegūst

$$\sum_{j=1}^g (\tilde{c}_j)_\alpha^L x_j^* \leq \sum_{j=1}^g (\tilde{c}_j)_\alpha^R x_j^* < \sum_{j=1}^g (\tilde{c}_j)_\alpha^L x'_j \leq \sum_{j=1}^g (\tilde{c}_j)_\alpha^R x'_j. \quad (3.2.7)$$

Turklāt, pēc pieņēmumiem, ka

$$(\tilde{c}_j)_\alpha^L \leq c_j \leq (\tilde{c}_j)_\alpha^R, \text{ un } x_j^* \geq 0, x'_j \geq 0, j = 1, \dots, g,$$

un (3.2.7) iegūst

$$\sum_{j=1}^g (\tilde{c}_j)_\alpha^L x_j^* \leq \sum_{j=1}^g c_j x_j^* \leq \sum_{j=1}^g (\tilde{c}_j)_\alpha^R x_j^* \text{ un } \sum_{j=1}^g (\tilde{c}_j)_\alpha^L x'_j \leq \sum_{j=1}^g c_j x'_j \leq \sum_{j=1}^g (\tilde{c}_j)_\alpha^R x'_j. \quad (3.2.8)$$

Apvienojot (3.2.7) un (3.2.8), iegūst rezultātu

$$\sum_{j=1}^g c_j x_j^* < \sum_{j=1}^g c_j x'_j,$$

kas ir pretruna ar pieņēmumu, ka  $x^*$  ir optimāls atrisinājums problēmai (3.2.6).  $\square$

**13. apgalvojums.** Pieņem, ka  $\tilde{c}_j, \tilde{a}_{ij}, \tilde{b}_i, i = 1, \dots, m, j = 1, \dots, g$ , ir nestrikti lielumi un  $\alpha, \beta \in (0, 1)$ . Pieņem, ka  $\tilde{X}$  ir iespējamais problēmas (3.1.2) apgabals attiecībā uz  $\tilde{E} = \prec^N$ , un  $c_j$  ir tāds, ka  $(\tilde{c}_j)_{1-\alpha}^L \leq c_j \leq (\tilde{c}_j)_{1-\alpha}^R, j = 1, \dots, g$ . Ja  $x^* = (x_1^*, \dots, x_g^*)$  ir optimālais atrisinājums lineārās programmēšanas problēmai

$$\begin{aligned} & \max \sum_{j=1}^g c_j x_j, \\ & \begin{cases} \sum_{j=1}^g (\tilde{a}_{ij})_{1-\beta}^R \leq (\tilde{b}_i)_{1-\beta}^L, & i = 1, \dots, m, \\ x_j \geq 0, & j = 1, \dots, g, \end{cases} \end{aligned} \quad (3.2.9)$$

tad  $x^*$  ir  $(\alpha, \beta)$ -optimāls atrisinājums problēmai (3.1.2).

*Pierādījums.* Pieņem, ka  $x^*$  ir optimāls problēmas (3.2.9) atrisinājums. Pierādīsim to, ka  $x^*$  ir  $(\alpha, \beta)$ -optimāls problēmas (3.1.2) atrisinājums ar  $\tilde{E} = \prec^N$ . Balstoties uz rezultātu (3.1.5),  $x^*$  ir  $(\beta)$ -iespējams problēmas (3.1.2) atrisinājums. Pieņem, ka  $x^*$  nav  $(\alpha, \beta)$ -optimāls problēmas (3.1.2) atrisinājums. Tad eksistē tāds  $\beta$ -iespējams problēmas (3.1.2) atrisinājums  $x' = (x'_1, \dots, x'_g)$ , ka  $\tilde{c}^T x^* \tilde{E}_{1-\alpha}^* \tilde{c}^T x'$ . Pēc 11. lemmas iegūst

$$\sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x_j^* \leq \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^L x'_j \text{ un } \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^L x_j^* < \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x'_j. \quad (3.2.10)$$

(a) Pieņem, ka  $(\tilde{c}_j)_{1-\alpha}^L \leq c_j < (\tilde{c}_j)_{1-\alpha}^R, x_j^*, j = 1, \dots, g$ , un  $x^* \neq 0$ , tad eksistē vismaz viens  $x_j > 0, j = 1, \dots, n$ . Tā kā ir iegūts, ka

$$\sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^L x_j^* \leq \sum_{j=1}^g c_j x_j^* < \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x_j^* \text{ un } \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^L x'_j \leq \sum_{j=1}^g c_j x'_j < \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x'_j \quad (3.2.11)$$

Apvienojot (3.2.10) un (3.2.11), iegūst

$$\sum_{j=1}^g c_j x_j^* < \sum_{j=1}^g c_j x'_j,$$

kas ir pretruna tam, ka  $x^*$  ir optimālais problēmas (3.2.9) atrisinājums.

(b) Pieņem, ka  $c_j = (\tilde{c}_j)_{1-\alpha}^R$  un  $x_j^* = 0$ ,  $j = 1, \dots, g$ . Balstoties uz (3.2.5) un (3.2.10), iegūst

$$0 = \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^L x_j^* < \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x'_j \text{ un } 0 = \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x_j^* < \sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x'_j,$$

kas atkal ir pretruna ar pieņēmumu, ka  $x^*$  ir optimālais problēmas (3.2.9) atrisinājums ar mērķa funkciju

$$\sum_{j=1}^g (\tilde{c}_j)_{1-\alpha}^R x_j.$$

Rezultātā  $x^*$  ir  $(\alpha, \beta)$ -optimāls atrisinājums problēmai (3.1.2). □

## 4 Problēmas risināšana, lietojot diskreto programmēšanu

Lineārās programmēšanas uzdevumu ar papildus nosacījumu uz atrisinājumu veselos skaitļos sauc par diskrētās programmēšanas uzdevumu. Jāņem vērā, ka par diskrētās programmēšanas uzdevumu tiek uzskatīts arī tāds lineārās programmēšanas uzdevums, kam vismaz viena no problēmas atrisinājuma vērtībām tiek sagaidīta veselos skaitļos. Problēmu (1.4.2) var reducēt uz diskreto programmēšanas problēmu, uzliekot papildus nosacījumu veselos skaitļos nezināmajam  $x(p)$  ( $x(p) \in \mathbb{Z}$  katram  $p \in P^k$ ,  $k = 1, \dots, K$ ).

### 4.1. Diskrētā programmēšanas uzdevuma ierobežojošo parametru izvēle

Balstoties uz lineārās programmēšanas problēmu ar nestriktiem parametriem (1.4.2) un 3. nodaļā iegūtajiem rezultātiem, diskrētās programmēšanas problēmu, ievērojot iespējamības sakarību, definē sekojoši:

$$\begin{aligned} & \min \sum_{k=1}^K \sum_{p \in P^k} \tilde{c}(p)x(p) \\ & \left\{ \begin{array}{ll} \sum_{k=1}^K \sum_{p \in P^k} \delta_{i,j}(p)x(p) \leq (\tilde{u}_{i,j})_{\beta}^R, & \forall (i, j) \in A, \\ \sum_{p \in P^k} x(p) \geq (\tilde{d}^k)_{\beta}^L, & k = 1, \dots, K, \\ x(p) \geq 0, & k = 1, \dots, K, p \in P^k, \\ x(p) \in \mathbb{Z}, & k = 1, \dots, K, p \in P^k, \end{array} \right. \end{aligned} \quad (4.1.1)$$

kur  $(\tilde{c}(p))_{\alpha}^L \leq c(p) \leq (\tilde{c}(p))_{\alpha}^R$ ,  $p \in P^k$ ,  $k = 1, \dots, K$ .

Balstoties uz lineārās programmēšanas problēmu ar nestriktiem parametriem (1.4.2) un 3. nodaļā iegūtajiem rezultātiem, diskrētās programmēšanas problēmu, ievērojot nepieciešamības sakarību, pārveido sekojoši:

$$\min \sum_{k=1}^K \sum_{p \in P^k} \tilde{c}(p)x(p)$$

$$\begin{cases} \sum_{k=1}^K \sum_{p \in P^k} \delta_{i,j}(p)x(p) \leq (\tilde{u}_{i,j})_{1-\beta}^L, & \forall (i,j) \in A, \\ \sum_{p \in P^k} x(p) \geq (\tilde{d}^k)_{1-\beta}^R, & k = 1, \dots, K, \\ x(p) \geq 0, & k = 1, \dots, K, p \in P^k, \\ x(p) \in \mathbb{Z}, & k = 1, \dots, K, p \in P^k, \end{cases} \quad (4.1.2)$$

kur  $(\tilde{c}(p))_{1-\alpha}^L \leq c(p) \leq (\tilde{c}(p))_{1-\alpha}^R, p \in P^k, k = 1, \dots, K$ .

## 4.2. Diskrētā programmēšanas uzdevuma mērķa funkcijas parametru izvēle

Tā kā preču transportēšanas izmaksas nav noteiktas, taču ir zināms intervāls, kādā vērtības atrodas, izmaksas pieņem par nezināmo lielumu. Līdz ar to problēmā iekļauj papildus nosacījumus uz izmaksām un paplašina to, pievienojot papildus nezināmos.

Apzīmē:  $y(p) = \tilde{c}(p)x(p), k = 1, \dots, K, p \in P^k$ .

Apskata diskrētās programmēšanas uzdevumu (4.1.1) pie iespējamības mēra. Tā kā

$$(\tilde{c}(p))_{\alpha}^L \leq c(p) \leq (\tilde{c}(p))_{\alpha}^R$$

un ir ieviests apzīmējums  $y(p)$ , veidojas papildus nevienādības

$$(\tilde{c}(p))_{\alpha}^L x(p) \leq y(p) \leq (\tilde{c}(p))_{\alpha}^R x(p).$$

Diskrētās programmēšanas uzdevumu (4.1.1) pārveido sekojoši:

$$\min \sum_{k=1}^K \sum_{p \in P^k} y(p)$$

$$\begin{cases} \sum_{k=1}^K \sum_{p \in P^k} \delta_{i,j}(p)x(p) \leq (\tilde{u}_{i,j})_{\beta}^R, & (i,j) \in A, \\ \sum_{p \in P^k} x(p) \geq (\tilde{d}^k)_{\beta}^L, & p \in P^k, k = 1, \dots, K, \\ y(p) - (\tilde{c}(p))_{\alpha}^L x(p) \geq 0, & p \in P^k, k = 1, \dots, K, \\ (\tilde{c}(p))_{\alpha}^R x(p) - y(p) \geq 0, & p \in P^k, k = 1, \dots, K, \\ x(p) \geq 0, & p \in P^k, k = 1, \dots, K, \\ x(p) \in \mathbb{Z}, & p \in P^k, k = 1, \dots, K. \end{cases} \quad (4.2.1)$$

Kā redzams, plūsmas lielumu  $x(p)$  joprojām meklē veselos skaitļos, taču kopējām izmaksām  $y(p)$  šāds ierobežojums nav nepieciešams.

Diskrētās programmēšanas uzdevumu (4.1.2) pie nepieciešamības mēra pārveido tādā pašā veidā un rezultātā iegūst:

$$\min \sum_{k=1}^K \sum_{p \in P^k} y(p)$$

$$\left\{ \begin{array}{ll} \sum_{k=1}^K \sum_{p \in P^k} \delta_{i,j}(p)x(p) \leq (\tilde{u}_{i,j})_{1-\beta}^L, & (i, j) \in A, \\ \sum_{p \in P^k} x(p) \geq (\tilde{d}^k)_{1-\beta}^R, & p \in P^k, k = 1, \dots, K, \\ y(p) - (\tilde{c}(p))_{1-\alpha}^L x(p) \geq 0, & p \in P^k, k = 1, \dots, K, \\ (\tilde{c}(p))_{1-\alpha}^R x(p) - y(p) \geq 0, & p \in P^k, k = 1, \dots, K, \\ x(p) \geq 0, & p \in P^k, k = 1, \dots, K, \\ x(p) \in \mathbb{Z}, & p \in P^k, k = 1, \dots, K. \end{array} \right. \quad (4.2.2)$$

### 4.3. Sazarošanās un robežu algoritms diskrētās programmēšanas uzdevuma risināšanai un tā praktiskā realizācija

Tiek aprakstīts sazarošanās un robežu algoritms – viena no diskrētās programmēšanas uzdevuma risinājuma metodēm.

Pamatā algoritms balstīts uz divām procedūrām – sazarošanās procedūras un novērtēšanas procedūras. Sazarošanās procedūra satur rekursīvu plānu kopas sadali mazākās apakškopās, plānu kopās. Iegūtās apakškopas veido izvēles koku, kura sakne ir visu plānu kopa, taču beigu galotnes – konkrēti problēmas atrisinājumi. Algoritms ietver uz priekšu un atpakaļ ejošus soļus pa izvēles koku.

Uz priekšu ejošajā solī pa izvēles koku notiek virzīšanās uz leju, turklāt, katru reizi tiek izvēlēts perspektīvākais zars. Zara perspektīvu nosaka ar novērtēšanas procedūras palīdzību, kurai jādod novērtējums uz augšu, ja tiek meklēts maksimums, vai uz leju, ja tiek meklēts minimums, mērķa funkcijai pie attiecīgā plāna. Solis izbeidzas pēdējā virsotnē – noteiktā uzdevuma plānā.

Atpakaļejošajā solī algoritmā notiek virzīšanās uz augšu izvēles kokā, turklāt, katrai koka virsotnei, plānu apakškopai, tiek noteikts ekstrēms, t.i., faktiski sasniedzamā mērķa funkcijas vērtība šajā apakškopā. [9]

Aplūko diskrētās programmēšanas uzdevumu:

$$\max \sum_{j=1}^g c_j x_j$$

$$\begin{cases} \sum_{j=1}^g a_{ij} x_j \leq b_i, & i = 1, \dots, m, \\ x_j \geq 0, & j = 1, \dots, g, \\ x_j \in \mathbb{Z}, & j \in J, \end{cases}$$

kur  $J \subset \{1, \dots, g\}$ ,  $J \neq \emptyset$ .

Algoritma darbības ieskatam ievieš apzīmējumus [2]:

$D$  plānu kopa,

$t = 0, 1, 2, \dots$  iterāciju indekss,

$M_t$  diskrētās programmēšanas uzdevuma kopa,

$x_t^*$  diskrētās programmēšanas uzdevuma atrisinājuma aproksimācija pēc  $t$  iterācijām,

$f_t^*$  mērķa funkcijas maksimālās vērtības aproksimācija pēc  $t$  iterācijām.

Algoritma izpildes sākuma nosacījumi ( $t = 0$ ):  $M_0 = \{DPU\}$ ,  $x_0^* \in \emptyset$ ,  $f_0^* = -\infty$ .

**Iterācijas  $t = 0 \rightsquigarrow t = 1$  apraksts.**

Uzdevumam  $DPU_0 \in M_0$  lieto apzīmējumu  $DPU_0$ . Uzdevumam  $DPU_0$  atbilstošais lineārās programmēšanas uzdevums tiek apzīmēts ar  $LPU_0$ . Tiek meklēts uzdevuma  $LPU_0$  atrisinājums  $x^*(LPU_0)$ . Tālāk ir iespējami divi gadījumi.

1)  $\neg \exists x^*(LPU_0) \Rightarrow \neg \exists x^*(DPU_0)$ . Ja lineārās programmēšanas uzdevumam atrisinājums neeksistē, tad atrisinājums neeksistē arī sākotnējam diskrētās programmēšanas uzdevumam un algoritms savu darbu beidz. Lai to panāktu, lieto  $M_1 = M_0 \setminus \{DPU_0\}$ .

2)  $\exists x^*(LPU_0)$ . Ja lineārās programmēšanas uzdevumam atrisinājums eksistē, tad aplūko divus gadījumus:

2a)  $x_j^*(LPU_0) \in \mathbb{Z}$ ,  $j \in J$ . Ja iegūtais lineārās programmēšanas uzdevuma atrisinājums apmierina diskrētās programmēšanas uzdevuma nosacījumus, tad iegūtais atrisinājums ir atrisinājums arī sākotnējam diskrētās programmēšanas uzdevumam:

$$x^*(DPU_0) = x^*(LPU_0);$$

$$x_1^* = x^*(DPU_0), f_1^* = f(x_1^*);$$

$$M_1 = M_0 \setminus \{DPU_0\}.$$

2b) Taču, ja kāda no atrisinājuma vērtībām ar indeksiem no kopas  $J$  nav vesels skaitlis (t.i., eksistē  $j \in J$ , :  $x_j^*(LPU_0) \notin \mathbb{Z}$ ), tad diskrētās programmēšanas uzdevumam izdala jaunas palīgproblēmas, balstoties uz atrasto atrisinājuma vērtību:

$$DPU'_0 : DPU_0 \oplus (x_j \leq [x_j^*(LPU_0)]);$$

$$DPU''_0 : DPU_0 \oplus (x_j \geq [x_j^*(LPU_0)] + 1);$$

$$M'_0 = M_0 \cup \{DPU'_0, DPU''_0\};$$

$$M_1 = M'_0 \setminus \{DPU_0\}.$$

Šeit zīme  $\oplus$  nozīmē to, ka uzdevumam tiek pievienots papildus nosacījums, un  $[x]$  ir skaitļa  $x$  veselā daļa.

Pēc šī soļa pāriet uz nākamo iterāciju, kur tādā pat veidā aplūko izveidotās palīgproblēmas.

### Iterācijas $t \rightsquigarrow t + 1$ apraksts.

$M_t, f_t^*, x_t^*$  –?

1) Ja aplūkotā uzdevumu kopa  $M_t$  ir tukša, tad apskata divus gadījumus:

$$1a) f_t^* = -\infty \Rightarrow \neg \exists x^*(DPU);$$

$$1b) f_t^* \in \mathbb{R} \Rightarrow x^*(DPU) = x_t^*, f(x^*(DPU)) = f(x_t^*).$$

2) Ja  $M_t \neq \emptyset$ , tad brīvi izvēlas uzdevumu no uzdevumu kopas  $M_t$  un apzīmē ar  $DPU_t$ .

Uzdevumam  $DPU_t$  atbilstošais lineārās programmēšanas uzdevums tiek apzīmēts ar  $LPU_t$  un tiek meklēts tā atrisinājums. Iespējami divi gadījumi:

$$2a) \neg \exists x^*(LPU_t) \Rightarrow f_{t+1}^* = f_t^*, x_{t+1}^* = x_t^*, M_{t+1} = M_t \setminus \{DPU_t\}.$$

2b) Ja  $\exists x^*(LPU_t)$ , tad analogi sākotnējai iterācijai izšķir divus gadījumus:

2b<sub>1</sub>) Ir iegūts palīgproblēmas atrisinājums, kas apmierina diskrētās programmēšanas uzdevuma nosacījumus:

$$x_j^*(LPU_t) \in \mathbb{Z}, j \in J, x^*(DPU_t) = x^*(LPU_t);$$

$$f(x^*(DPU_t)) > f_t^* \Rightarrow f_{t+1}^* = f(x^*(DPU_t)), x_{t+1}^* = x^*(DPU_t);$$

$$f(x^*(DPU_t)) \leq f_t^* \Rightarrow f_{t+1}^* = f_t^*, x_{t+1}^* = x_t^*;$$

$$M_{t+1} = M_t \setminus \{DPU_t\},$$

2b<sub>2</sub>) eksistē  $j \in J : x_j^*(LPU_t) \notin \mathbb{Z}$  :

Ja  $f(x^*(LPU_t)) \leq f_t^*$ , tad  $x_{t+1}^* = x_t^*, f_{t+1}^* = f_t^*$ , virzību pa šo zaru pātrauc un atlikušās palīgproblēmas neapskata.  $M_{t+1} = M_t \setminus \{DPU_t\}$ .

Ja  $f(x^*(LPU_t)) > f_t^*$ , tad izdala palīgproblēmas, balstoties uz atrasto atrisinājumu:

$$DPU'_t : DPU_t \oplus x_j \leq [x_j(LPU_t)], DPU''_t : DPU_t \oplus x_j \geq [x_j(LPU_t)] + 1;$$

$$M'_t \cup \{DPU'_t, DPU''_t\};$$

$$x_{t+1}^* = x_t^*, f_{t+1}^* = f_t^*;$$

$$M_{t+1} = M'_t \setminus \{DPU_t\}.$$

**Piemērs.**

$$\begin{aligned} & \max (x_1 + x_2) \\ & \begin{cases} 2x_1 + 3x_2 \leq 6 \\ 5x_1 + 2x_2 \leq 10 \\ x_1 \geq 0, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z} \end{cases} \end{aligned}$$

Sākotnēji:  $M_0 = \{DPU\}$ ,  $x_0^*$  - nav noteikts,  $f_0^* = -\infty$

**Iterācija  $t = 0 \rightsquigarrow t = 1$ .**

$DPU_0 \in M_0$ ,  $DPU_0 = DPU$  (sākotnējais uzdevums)

$LPU_0 = LPU$ , meklē  $x^*(LPU_0)$ , iegūst  $x^*(LPU_0) = (\frac{18}{11}, \frac{10}{11})^T$  -  $x_j^* \notin \mathbb{Z}$ ,  $j = 1, 2$

Patvaļīgi izvēlas kādu no atrisinājuma vērtībām, kas nav vesels skaitlis, šeit-  $x_1$ , veido palīgproblēmas:

$DPU'_0 = DPU_0 \oplus (x_1 \leq 1)$ ,  $DPU''_0 = DPU_0 \oplus (x_1 \geq 2)$

$D(DPU'_0) \cup D(DPU''_0) = D(DPU_0)$

$M_1 = M_0 \setminus \{DPU_0\} \cup \{DPU'_0, DPU''_0\}$  jeb  $M_1 = \{DPU_0 \oplus (x_1 \leq 1), DPU_0 \oplus (x_1 \geq 2)\}$

$x_1^*$ - nav noteikts,  $f_1^* = f_0^* = -\infty$

**Iterācija  $t = 1 \rightsquigarrow t = 2$ .**

$DPU_1 \in M_1$  (brīvi izvēlēts),  $DPU_1 = DPU \oplus (x_1 \leq 1)$ ,  $LPU_1 = LPU \oplus (x_1 \leq 1)$

Risina  $LPU_1$ , iegūst  $x^*(LPU_1) = (1, \frac{4}{3})^T$ ,  $x_2^*(LPU_1) \notin \mathbb{Z}$ , līdz ar to veido palīgproblēmas:

$DPU'_1 = DPU_1 \oplus (x_2 \leq 1)$ ,  $DPU''_1 = DPU_1 \oplus (x_2 \geq 2)$

$D(DPU'_1) \cup D(DPU''_1) = D(DPU_1)$

$M_2 = M_1 \setminus \{DPU_1\} \cup \{DPU'_1, DPU''_1\}$  jeb

$M_2 = \{DPU \oplus (x_1 \leq 1, x_2 \leq 1), DPU \oplus (x_1 \leq 1, x_2 \geq 2), DPU \oplus (x_1 \geq 2)\}$

$x_2^*$ - nav noteikts,  $f_2^* = f_1^* = -\infty$

**Iterācija  $t = 2 \rightsquigarrow t = 3$ .**

$DPU_2 \in M_2$ ,  $DPU_2 = DPU \oplus (x_1 \leq 1, x_2 \leq 1)$ ,  $LPU_2 = LPU \oplus (x_1 \leq 1, x_2 \leq 1)$

Risina  $LPU_2$ , iegūst  $x^*(LPU_2) = x^*(DPU_2) = (1, 1)^T$ ,  $f^*(LPU_2) = f^*(DPU_2) = 2$

$M_3 = M_2 \setminus \{DPU_2\} = \{DPU \oplus (x_1 \leq 1, x_2 \geq 2), DPU \oplus (x_1 \geq 2)\}$

$x_3^* = (1, 1)^T$ ,  $f_3^* = 2$

**Iterācija  $t = 3 \rightsquigarrow t = 4$ .**

$DPU_3 \in M_3$ ,  $DPU_3 = DPU \oplus (x_1 \geq 2)$ ,  $LPU_3 = LPU \oplus (x_1 \geq 2)$

Risina  $LPU_3$ , iegūst  $x^*(LPU_3) = x^*(DPU_3) = (2, 0)$ ,  $f^*(LPU_3) = f^*(DPU_3) = 2$

$M_4 = M_3 \setminus \{DPU_3\} = \{DPU \oplus (x_1 \geq 2)\}$

$$x_4^* = (2, 0)^T, f_4^* = 2$$

**Iterācija  $t = 4 \rightsquigarrow t = 5$ .**

$$DPU_4 \in M_4, DPU_4 = DPU \oplus (x_1 \leq 1, x_2 \geq 2), LPU_4 = LPU \oplus (x_1 \leq 1, x_2 \geq 2)$$

$$\text{Risina } LPU_4, \text{ iegūst } x^*(LPU_4) = x^*(DPU_4) = (0, 2), f^*(LPU_4) = f^*(DPU_4) = 2$$

$$M_5 = M_4 \setminus \{DPU_4\} = \emptyset$$

$$x_5^* = (2, 0)^T, f_5^* = 2$$

Rezultātā ir iegūti 3 dažādi optimālie atrisinājumi, kam ir vienāda mērķa funkcijas vērtība,

$$x^* = (1, 1)^T, x^* = (2, 0)^T, x^* = (0, 2)^T \text{ un } f_{max}^* = 2.$$

# 5 Preču optimālās plūsmas noteikšanas piemēri

## 5.1. Pirmais uzdevums

Tiek aplūkoti 1. piemēra rezultāti. Piemērā apskatīts orientēts grafs, uzdevums ir transportēt 2 dažādu veidu preces uz dažādiem galamērķiem no kopīgas izcelsmes vietas. Atrisinājums iegūts gan pie iespējamības, gan pie nepieciešamības mēra, lēmumu pieņemšanas parametriem mainoties.

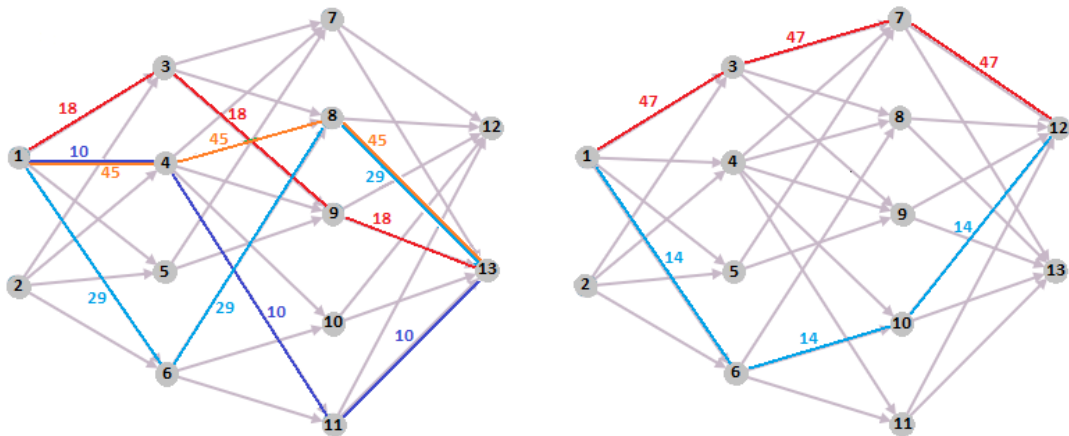
Vispirms aplūko gadījumu, kad tiek fiksēta piederības pakāpe  $\alpha = 0.8$ , taču piederības pakāpe  $\beta$  tiek mainīta ar soli 0.1. Rezultātu detalizēts apkopojums dots 1.1. pielikumā. Ieskatam 5.1.1. tabulā dots piemēra atrisinājums pie iespējamības mēra. Pie  $\alpha = 0.8$  un  $\beta = 0.1$  mērķa funkcija  $z$  sasniedz vērtību 15224.2. Pa maršrutu  $1 \rightarrow 4 \rightarrow 8 \rightarrow 13$  tiek transportētas 18 pirmā veida preces vienības, pa maršrutu  $1 \rightarrow 4 \rightarrow 11 \rightarrow 13$  tiek transportētas 45 pirmā veida preces vienības, pa maršrutu  $1 \rightarrow 3 \rightarrow 7 \rightarrow 12$  tiek transportētas 47 otrā veida preces vienības u.t.t. Uz virsotni 13 tiek transportētas 102 pirmā veida preces vienības un uz virsotni 12 tiek transportēta 61 otrā veida preces vienības, tādējādi piegādājot pieprasīto preču daudzumu.

Tabula 5.1.1: Pirmā uzdevuma atrisinājums pie iespējamības mēra,  $\alpha = 0.8$

$\beta$	Prece	Maršruts	Plūsma	$z$
0.1	1	$1 \rightarrow 3 \rightarrow 9 \rightarrow 13$	18	15224.2
		$1 \rightarrow 4 \rightarrow 8 \rightarrow 13$	45	
		$1 \rightarrow 4 \rightarrow 11 \rightarrow 13$	10	
		$1 \rightarrow 6 \rightarrow 8 \rightarrow 13$	29	
	2	$1 \rightarrow 3 \rightarrow 7 \rightarrow 12$	47	
		$1 \rightarrow 6 \rightarrow 10 \rightarrow 12$	14	

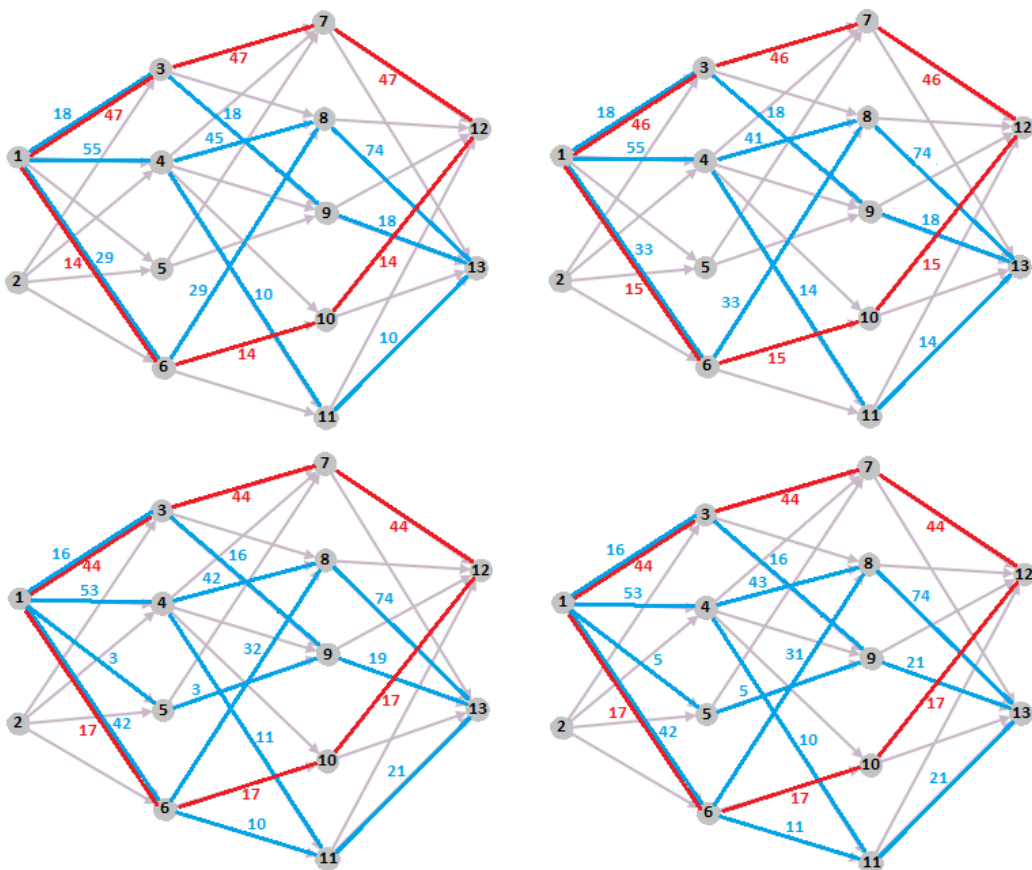
Attēlā 5.1.1. redzami visi atrisinājuma maršruti pirmā un otrā veida precēm. Katrs maršruts iezīmēts atšķirīgā krāsā. Pie lokiem atzīmēts transportētais preces daudzums pa loku, skaitļa krāsa parāda piederību noteiktam maršrutam. Pa loku (1, 4) tiek transportētas 55 pirmā veida preces vienības, tālāk preces tiek sadalītas transportēšanai uz virsotnēm 8 un 11, uz kuriem tiek trans-

portētas attiecīgi 45 un 10 preces vienības. Maršrutu skaits pirmā veida preču transportēšanai ir 4, taču otrā veida precī transportē pa diviem maršrutiem.



Att. 5.1.1: Tabulas 5.1.1. atrisinājuma ilustrācija pirmā un otrā veida precī

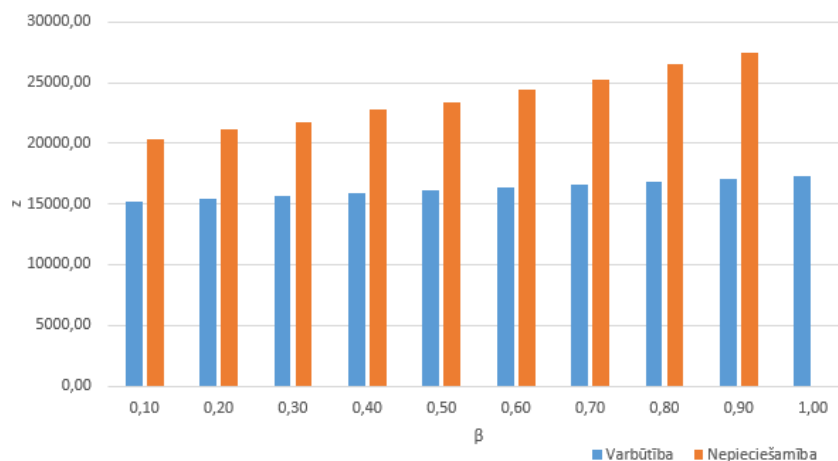
Tālāk aplūko 5.1.2. attēlu, kurā attēloti četri 1. piemēra atrisinājumi pie iespējamības sakarības, piederības pakāpei  $\beta$  mainoties.



Att. 5.1.2: Pirmā uzdevuma atrisinājuma ilustrācija pie iespējamības mēra,  $\beta = 0.1, 0.3, 0.7$  un  $0.8$

Šeit attiecīgi ilustrēti rezultāti, kad  $\beta$  vērtība sasniedz 0.1, 0.2, 0.7 un 0.8 atzīmi. Vieglākai pār-

redzamībai pirmā veida preces transportēšanas maršruti attēloti zilā krāsā, taču otrā veida preces transportēšanas maršruti – sarkanā krāsā. Redzams, ka pie  $\beta = 0.1$  un  $\beta = 0.3$  maršruti, pa kuriem transportē preces, nemainās, mainās tikai transportētais daudzums (pirmā veida precei) un transportētā daudzuma sadalījums pa maršrutiem (pirmā un otrā veida precēm). Sākot ar  $\beta = 0.7$  pirmā veida preces transportēšanai tiek izmantoti papildus maršruti –  $1 \rightarrow 5 \rightarrow 9 \rightarrow 13$  un  $1 \rightarrow 6 \rightarrow 11 \rightarrow 13$ , pa kuriem attiecīgi tiek transportētas 5 un 10 pirmā veida preces vienības.



**Att. 5.1.3: Pirmā uzdevuma mērķa funkcijas vērtības pie dažādām piederības pakāpes  $\beta$  vērtībām,  $\alpha = 0.8$**

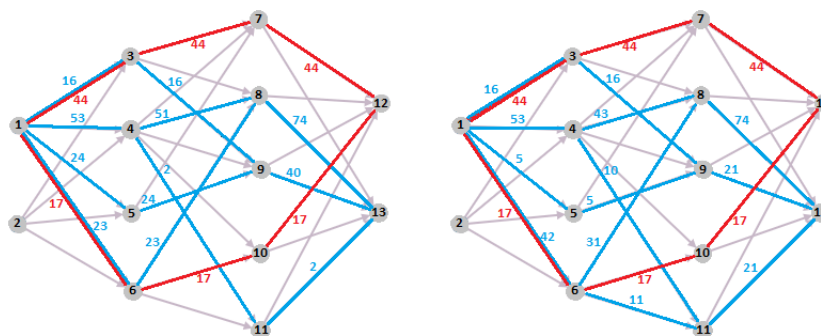
Attēls 5.1.3. parāda iespējamā un nepieciešamā  $(\alpha, \beta)$ -optimālā atrisinājuma mērķa funkcijas vērtību uzvedību atkarībā no dažādām piederības pakāpes  $\beta$  vērtībām. Palielinot parametru  $\beta$ , vērtība, balstoties uz iespējamības sakarību, palielinās. Palielinot parametru  $\beta$ , vērtība, balstoties uz nepieciešamības sakarību, palielinās. Pie piederības pakāpes 1 nepieciešamais  $(\alpha, \beta)$ -optimālais atrisinājums aplūkotajai problēmai neeksistē. Salīdzinot rezultātus, kur izmantota iespējamības vai nepieciešamības sakarība, pie fiksētām piederības pakāpes vērtībām, optimālais atrisinājums problēmai balstoties uz nepieciešamības sakarību ir ievērojami lielāks nekā problēmai, kas balstās uz iespējamības sakarību.

Tabula 5.1.2: Pirmā uzdevuma mērķa funkcijas vērtības pie iespējamības un nepieciešamības mēra,  $\alpha = 0.8$

$\beta$	Prece	$d, \Pi$	$d, N$	$z, \Pi$	$z, N$
0.1	1	102	135	15224.2	20333.6
	2	61	75		
0.2	1	104	137	15426.6	21179.4
	2	61	78		
0.3	1	106	138	15648.4	21707.2
	2	61	80		
0.4	1	108	140	15897.4	22737.6
	2	61	81		
0.5	1	110	141	16122.4	23398.6
	2	61	85		
0.6	1	112	143	16329.6	24386.2
	2	61	88		
0.7	1	114	144	16606.4	25180
	2	61	90		
0.8	1	116	146	16819.2	26529.2
	2	61	93		
0.9	1	118	147	17052.6	27487.2
	2	61	95		
1	1	120	—	17263.8	—
	2	61	—		

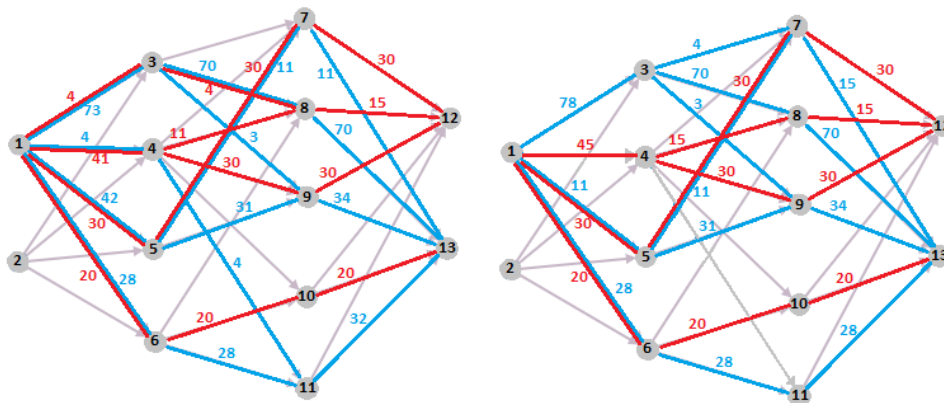
Iespējamā un nepieciešamā  $(\alpha, \beta)$ -optimālā atrisinājuma mērķa funkciju vērtības un transportēto pirmā un otrā veida preces daudzumu atkarībā no piederības pakāpes  $\beta$  pie fiksētas  $\alpha$  vērtības detalizēti var aplūkot 5.1.2. tabulā.

Tālāk aplūko rezultātus, kad tiek fiksēta vērtība  $\beta = 0.8$  un vērtība  $\alpha$  tiek mainīta ar soli 0.1. Jāpiemin, ka pēc problēmas nostādnes  $\alpha$  vērtības maiņa ietekmē tikai transportēšanas izmaksas. Atrisinājuma vērtības būtiski nemainās. Attēlā 5.1.4. var aplūkot divas atrisinājuma pie iespējamības mēra atkarībā no lēmumu pieņemšanas parametra  $\alpha$  ilustrācijas. Atrisinājums atbilst gadījumiem, kad attiecīgi  $\alpha = 0.1, 0.2, 0.3, 0.4, 0.5$  un  $\alpha = 0.6, 0.7, 0.8, 0.9, 1$ .

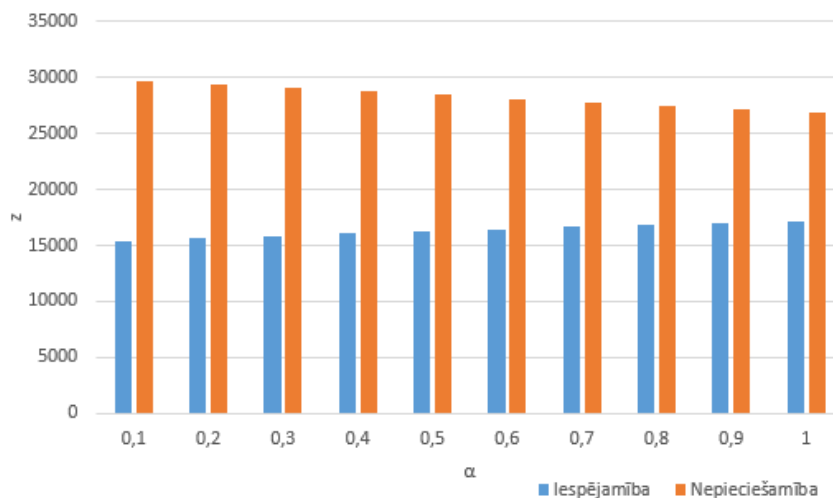


Att. 5.1.4: Atrisinājuma ilustrācija pirmajam uzdevumam pie iespējamības mēra,  $\beta = 0.8$

Attēlā 5.1.5. var aplūkot divas atrisinājuma pie nepieciešamības mēra ilustrācijas atkarībā no lēmumu pieņemšanas parametra  $\alpha$ . Atrisinājums atbilst gadījumiem, kad  $\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7$  un  $\alpha = 0.8, 0.9, 1$ , attiecīgi. Izmaksu parametru izmaiņas maršrutu izvēli būtiski nemaina. Abos gadījumos izmaksu izmaiņu ietekmē, mainās maršruts un transportētais daudzums pa kādu no lokiem, sākot ar noteiktu soli. Pieprasījums pēc preces un loku caurlaides spēja paliek nemainīga, kas atbilst problēmas nostādnei.



Att. 5.1.5: Atrisinājuma ilustrācija pirmajam uzdevumam pie nepieciešamības mēra,  $\beta = 0.8$

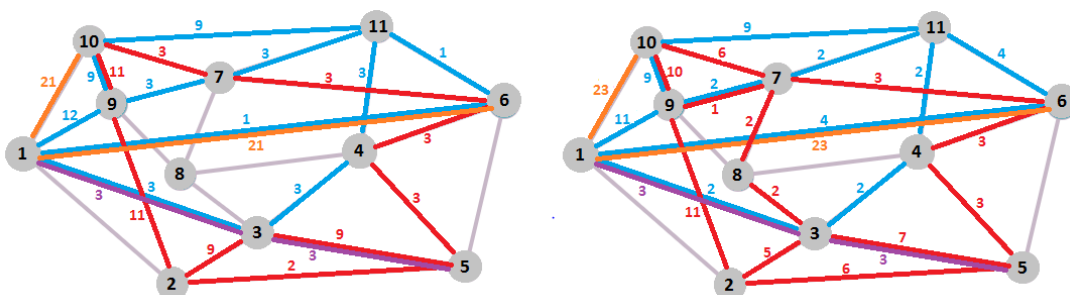


Att. 5.1.6: Pirmā uzdevuma mērķa funkcijas vērtības pie dažādām piederības pakāpes  $\alpha$  vērtībām,  $\beta = 0.8$

Attēls 5.1.6. parāda iespējamā un nepieciešamā  $(\alpha, \beta)$ -optimālā atrisinājuma mērķa funkcijas vērtības uzvedību atkarībā no  $\alpha$  vērtības pie fiksēta  $\beta = 0.8$ . Palielinot parametru  $\alpha$ , vērtība, balstoties uz iespējamības sakarību, palielinās. Palielinot parametru  $\alpha$ , vērtība, balstoties uz nepieciešamības sakarību, samazinās. Salīdzinot rezultātus starp izmantotajiem mēriem, pie fiksētām piederības pakāpes vērtībām, optimālais atrisinājums problēmai ar nepieciešamības mēru ir ievērojami lielāks nekā problēmai ar iespējamības mēru.

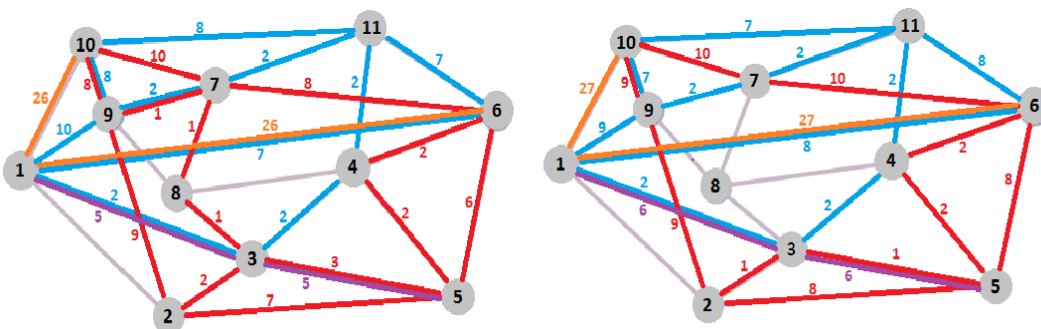
## 5.2. Otrais uzdevums

Turpinājumā aplūkosim 2. piemēra iegūtos rezultātus, izmantojot gan iespējamības, gan nepieciešamības mēru un mainot lēmumu pieņemšanas parametru vērtības. Aplūkotā piemēra nostādņē definētais grafs ir neorientēts, nepieciešams transportēt 4 dažāda veida preces un šo preču izcelsmes vietas un transportēšanas galamērķi ir dažādi. Piemērā aplūko sakaru informācijas plūsmu, kur nepieciešams efektīvi transportēt datus saturošus elementus no izcelsmes vietas uz informācijas saņemšanas vietu. Visus 2. piemēra starprezultātus var aplūkot 1.2. pielikumā.



Att. 5.2.1: Atrisinājuma ilustrācija otrajam uzdevumam pie iespējamības mēra,  $\alpha = 0.7$

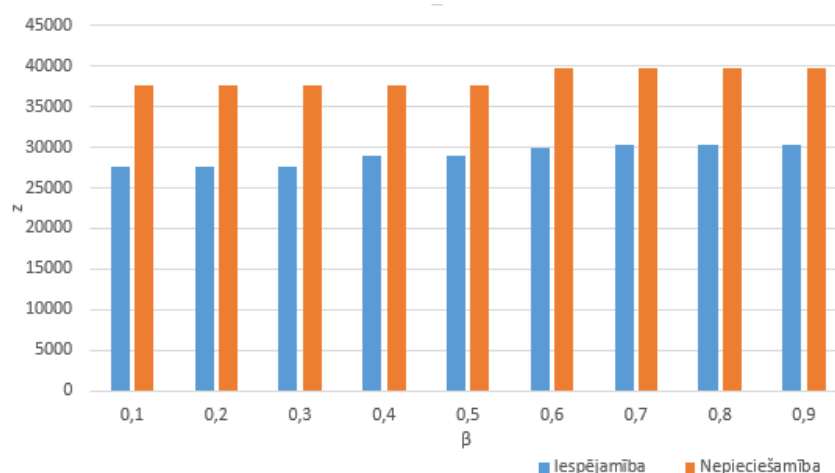
Attēlā 5.2.1. ilustrēts atrisinājums pie iespējamības mēra, kur vērtība  $\alpha = 0.7$  (fiksēta) un vērtība  $\beta$  mainās. Šeit attiecīgi ilustrēti atrisinājumi, kad  $\beta$  vērtība ir 0.1 un 0.7. Maršrutu izmaiņas transporta tīklā ir novērojamas otrā veida precei (attēlā maršruti ilustrēti ar sarkanu krāsu). Palielinoties  $\beta$  vērtībai, ir atrasti papildus divi otrā veida preces transportēšanas maršruti. Pārējām precēm, salīdzinot atrisinājumu pie  $\beta = 0.1$ , maršruti nemainās, taču ir izmainījies transportētais preču daudzums.



Att. 5.2.2: Atrisinājuma ilustrācija otrajam uzdevumam pie nepieciešamības mēra,  $\alpha = 0.7$

Apskatot rezultātus, kas iegūti apskatot problēmu pie iespējamības mēra,  $\beta$  vērtībai mainoties ar soli 0.1 un  $\alpha$  vērtībai paliekot nemainīgai ( $\alpha = 0.7$ ), nav novērojamas būtiskas atrisinājuma izmaiņas. Ir iegūti divi dažādi atrisinājumi, kas sakrīt vairākām  $\beta$  vērtībām. Attēlā

5.2.2. var aplūkot iegūto atrisinājumu ilustrāciju. Pirmais grafs ilustrē problēmas atrisinājumu pie  $\beta$  vērtībām 0.1, 0.2, 0.3, 0.4, 0.5, otrais grafs ilustrē problēmas atrisinājumu pie  $\beta$  vērtībām 0.6, 0.7, 0.8 un 0.9.



Att. 5.2.3: Otrā uzdevuma mērķa funkcijas vērtības pie dažādām piederības pakāpes  $\beta$  vērtībām,  $\alpha = 0.7$

Attēlā 5.2.3. parādītas mērķa funkciju vērtības problēmai pie iespējamības un nepieciešamības mēra,  $\beta$  vērtībai mainoties ar soli 0.1. Kā minēts iepriekš, rezultāti problēmai pie nepieciešamības mēra būtiski nemainās, mainot vērtību  $\beta$ . Daļai apskatīto gadījumu, kad  $\beta = 0.1, 0.2, 0.3, 0.4, 0.5$ , atrisinājums ir vienāds un līdz ar to mērķa funkcijas vērtības ir vienādas. Tāda pati tendence novērojama problēmai pie iespējamības mēra, mērķa funkcijas vērtības paliek nemainīgas vai pamazām pieaug,  $\beta$  vērtībai mainoties. Salīdzinot mērķa funkcijas vērtības atkarībā no problēmas risināšanā izmantotās sakarības, mērķa funkcijas vērtība problēmai pie nepieciešamības mēra ir lielāka nekā problēmai pie iespējamības mēra.

Aplūkosim rezultātus piemēram pie fiksētas  $\beta$  vērtības ( $\beta = 0.6$ ),  $\alpha$  vērtībai mainoties. Piemēram pie iespējamības sakarības, mainot  $\alpha$  vērtību ar soli 0.1, atrisinājums nemainās, mainās tikai mērķa funkcijas vērtība. Rezultātu var izskaidrot ar to, ka  $\alpha$  vērtība problēmas nostādnē ietekmē tikai izmaksu lieluma izvēli, pārējais, pieprasījums pēc precēm un caurlaides spēja, paliek nemainīgi un izmaksu izmaiņas nav radījušas nepieciešamību maršruta izmaiņām. Tabulā 5.2.1. parādīts atrisinājums, kas atbilst problēmai pie visām aplūkotajām  $\alpha$  vērtībām.

Tabula 5.2.1:  $(\alpha, \beta)$ -optimālais atrisinājums otrajam uzdevumam pie iespējamības mēra,  $\beta = 0.6$

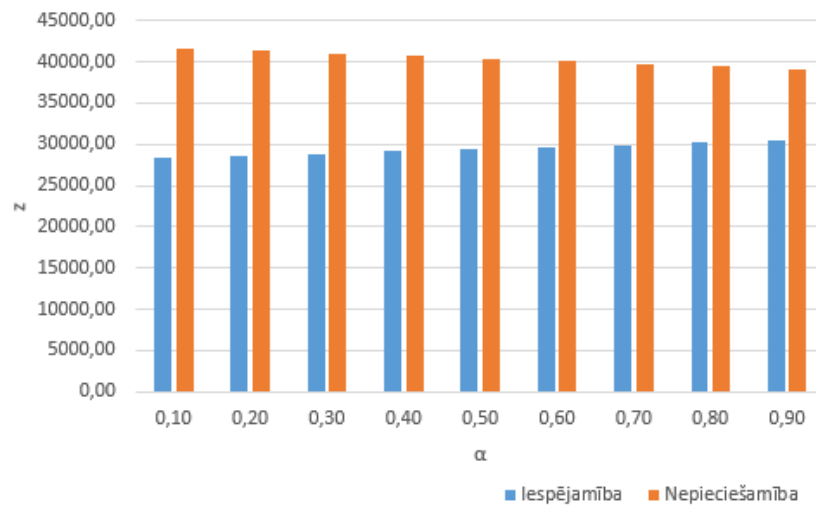
Prece	Maršruts	Plūsma
1	1 → 3 → 4 → 11	3
	1 → 6 → 11	3
	1 → 9 → 7 → 11	2
	1 → 9 → 10 → 11	9
2	10 → 7 → 6 → 4 → 5	4
	10 → 9 → 2 → 3 → 5	8
	10 → 9 → 2 → 5	3
3	1 → 3 → 5	3
4	10 → 1 → 6	22

Tādus pat rezultātus iegūst, aplūkojot problēmu pie nepieciešamības mēra –  $\alpha$  vērtībai mainoties, mainās mērķa funkcijas vērtība, taču maršruts un transportētais preces daudzums pa lokiem paliek nemainīgs. Iegūto atrisinājumu, maršrutu var aplūkot 5.2.4. tabulā.

Tabula 5.2.2:  $(\alpha, \beta)$ -optimālais atrisinājums otrajam uzdevumam pie nepieciešamības mēra,  $\beta = 0.6$

Prece	Maršruts	Plūsma
1	1 → 3 → 4 → 11	2
	1 → 6 → 11	8
	1 → 9 → 7 → 11	2
	1 → 9 → 10 → 11	7
2	10 → 7 → 6 → 4 → 5	2
	10 → 7 → 6 → 5	8
	10 → 7 → 9 → 2 → 3 → 5	1
	10 → 9 → 2 → 5	8
3	1 → 3 → 5	6
4	10 → 1 → 6	27

Attēls 5.2.4. parāda iespējamā un nepieciešamā  $(\alpha, \beta)$ -optimālā atrisinājuma mērķa funkcijas vērtības uzvedību atkarībā no  $\alpha$ . Palielinot parametru  $\alpha$ , vērtība, balstoties uz iespējamības sakarību, palielinās. Palielinot parametru  $\alpha$ , vērtība, balstoties uz nepieciešamības sakarību, samazinās. Salīdzinot rezultātus starp izmantotajiem mēriem, pie fiksētām piederības pakāpes vērtībām, optimālais atrisinājums problēmai ar nepieciešamības mēru ir ievērojami lielāks nekā problēmai ar iespējamības mēru.



*Att. 5.2.4: Otrā uzdevuma mērķa funkcijas vērtības pie iespējamības un nepieciešamības mēra,  
 $\beta = 0.6$*

## Nobeigums

Šajā darbā tika aplūkota nestriktās lineārās programmēšanas metode, kas risina lineārās programmēšanas uzdevumus ar nestriktiem parametriem, un risināta preču optimālās plūsmas problēma. Lai rastos lielāka izpratne par uzdevuma specifiku, tika aplūkota nestriktu kopu un lielumu teorija. Preču optimālās plūsmas problēma tika ilustrēta ar uzskatāmiem piemēriem. Aprakstītā metode nestriktās lineārās programmēšanas uzdevuma reducēšanai uz diskrēto programmēšanas uzdevumu ir labi aprobēta. Metodes aprobēšanai izmantotie skaitliskie piemēri sniedz sagaidāmos rezultātus.

Metodes īstenošanai tika izveidots programmas kods programmēšanas valodā *R*. Koda izstrādē lielākās grūtības rada grafa uzdošanai nepieciešamo nosacījumu izveide un ievade. Papildus problēmas nostādnē esošajiem nosacījumiem bija nepieciešams izveidot nosacījumus, kas nodrošina to, ka risināšanas gaitā ir aplūkoti visi iespējamie maršruti un plūsma grafā definēta korekti. Programmas kodu ir iespējams uzlabot, ja ir vajadzība aplūkot pēc uzbūves sarežģītākus grafus. Darbā tika aplūkoti divi piemēri. Viens no piemēriem balstās uz orientētu grafu, otrs – uz neorientētu grafu. Piemērā, kur tika apskatīts neorientēts grafs, bija vajadzība ieviest papildus nosacījumus norientēta grafa struktūras un īpašību dēļ. Risināšanas rīku var attīstīt, aplūkojot problēmas, kuru pamatā ir sarežģītākas uzbūves grafi – virsotņu skaita ziņā lielāki grafi, grafi ar cilpām, grafi ar kādiem īpašiem transportēšanas nosacījumiem.

# Izmantotā literatūra un avoti

- [1] M. Ghatee, S.M. Hashemi. Some concepts of the fuzzy multicommodity flow problem and their applications in fuzzy network design. *Mathematical and Computer Modelling*, 2009, N 49, 1030-1043
- [2] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, E. C. Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching and pruning. *Discrete Optimization*, 2016, N 19, 79-102
- [3] M. Niksirat, S. M. Mashemi, M. Ghatee. Branch-and-price algorithm for fuzzy integer programming problems with block angular structure. *Fuzzy Sets and Systems*, 2016, N 296, 70-79
- [4] J. Ramik. Duality in fuzzy linear programming with possibility and necessity relations. *Fuzzy Sets and Systems*, 2006, N 157, 1283-1302
- [5] J. Vahidi, S. Rezvani. Arithmetic operations on trapezoidal fuzzy numbers. *Nonlinear Analysis and Applications*, 2013, N 2013, 1-8
- [6] M. G. Voskoglou. Use of the triangular fuzzy numbers for student assessment. *American Journal of Applied Mathematics and Statistics*, 2015, N 4, 146-150
- [7] R. J. Wilson. *Introduction to Graph Theory*. England, Essex: Addison Wesley Longman Limited, 1996
- [8] L. A. Zadeh. Fuzzy sets. *Information and Control*, 1965, N 8, 338-353
- [9] Б. А. Гладких. *Методы оптимизации и исследование операций для бакалавров информатики*. Томск: НТЛ, 2009

# 1 Pielikums

## 1.1. Pirmā uzdevuma atrisinājums

Tabula 1.1.1:  $(\alpha, \beta)$ -optimālais atrisinājums pirmajam uzdevumam pie iespējamības mēra,  
 $\alpha = 0.8$

$\beta$	Prece	Maršruts	Plūsma	$z$
0.1	1	1 →3 →9 →13	18	15224.2
		1 →4 →8 →13	45	
		1 →4 →11 →13	10	
		1 →6 →8 →13	29	
	2	1 →3 →7 →12	47	
		1 →6 →10 →12	14	
0.2	1	1 →3 →9 →13	18	15426.6
		1 →4 →8 →13	43	
		1 →4 →11 →13	12	
		1 →6 →8 →13	31	
	2	1 →3 →7 →12	47	
		1 →6 →10 →12	14	
0.3	1	1 →3 →9 →13	18	15648.4
		1 →4 →8 →13	41	
		1 →4 →11 →13	14	
		1 →6 →8 →13	33	
	2	1 →3 →7 →12	46	
		1 →6 →10 →12	15	
0.4	1	1 →3 →9 →13	17	15897.4
		1 →4 →8 →13	41	
		1 →4 →11 →13	13	
		1 →6 →8 →13	33	

		1 →6 →11 →13	4	
	2	1 →3 →7 →12	46	
		1 →6 →10 →12	15	
0.5	1	1 →3 →9 →13	17	16122.4
		1 →4 →8 →13	41	
		1 →4 →11 →13	13	
		1 →6 →8 →13	33	
		1 →6 →11 →13	6	
	2	1 →3 →7 →12	45	
		1 →6 →10 →12	16	
0.6	1	1 →3 →9 →13	17	16329.6
		1 →4 →8 →13	42	
		1 →4 →11 →13	12	
		1 →6 →8 →13	32	
		1 →6 →11 →13	9	
	2	1 →3 →7 →12	45	
		1 →6 →10 →12	16	
0.7	1	1 →3 →9 →13	16	16606.4
		1 →4 →8 →13	42	
		1 →4 →11 →13	11	
		1 →5 →9 →13	3	
		1 →6 →8 →13	32	
	1 →6 →11 →13	10		
2	1 →3 →7 →12	44		
		1 →6 →10 →12	17	
0.8	1	1 →3 →9 →13	16	16819.2
		1 →4 →8 →13	43	
		1 →4 →11 →13	10	
		1 →5 →9 →13	5	
		1 →6 →8 →13	31	
	1 →6 →11 →13	11		
2	1 →3 →7 →12	44		
		1 →6 →10 →12	17	

0.9	1	1 →3 →9 →13	16	17052.6
		1 →4 →8 →13	43	
		1 →4 →11 →13	10	
		1 →5 →9 →13	8	
		1 →6 →8 →13	31	
		1 →6 →11 →13	10	
	2	1 →3 →7 →12	43	
		1 →6 →10 →12	18	
1	1	1 →3 →9 →13	16	17263.8
		1 →4 →8 →13	43	
		1 →4 →11 →13	10	
		1 →5 →9 →13	10	
		1 →6 →8 →13	31	
		1 →6 →11 →13	10	
	2	1 →3 →7 →12	43	
		1 →6 →10 →12	18	

Tabula 1.1.2:  $(\alpha, \beta)$ -optimālais atrisinājums pirmajam uzdevumam pie nepieciešamības mēra,  $\alpha = 0.8$

$\beta$	Prece	Maršruts	Plūsma	z
0.1	1	1 →3 →8 →13	40	21179.4
		1 →3 →9 →13	7	
		1 →4 →8 →13	10	
		1 →4 →11 →13	18	
		1 →5 →9 →13	38	
		1 →6 →8 →13	20	
		1 →6 →11 →13	4	
	2	1 →3 →7 →12	33	
		1 →4 →8 →12	15	
		1 →4 →9 →12	3	
		1 →6 →10 →12	18	
		1 →3 →8 →13	45	
		1 →3 →9 →13	6	

0.2	1	1 →4 →8 →13	3	21707.2
		1 →4 →11 →13	22	
		1 →5 →9 →13	37	
		1 →6 →8 →13	22	
		1 →6 →11 →13	3	
	2	1 →3 →7 →12	32	
		1 →4 →8 →12	15	
		1 →4 →9 →12	6	
		1 →5 →7 →12	1	
		1 →6 →10 →12	26	
0.3	1	1 →3 →8 →13	53	21707.2
		1 →3 →9 →13	6	
		1 →4 →8 →13	2	
		1 →4 →11 →13	18	
		1 →5 →9 →13	36	
		1 →6 →8 →13	15	
	1 →6 →11 →13	10		
	2	1 →3 →7 →12	23	
		1 →4 →8 →12	15	
		1 →4 →9 →12	9	
1 →5 →7 →12		9		
1 →6 →10 →12	25			
0.4	1	1 →3 →8 →13	58	23398.6
		1 →3 →9 →13	5	
		1 →4 →11 →13	17	
		1 →5 →9 →13	35	
		1 →6 →8 →13	12	
		1 →6 →11 →13	14	
	2	1 →3 →7 →12	18	
		1 →4 →8 →12	15	
		1 →4 →9 →12	14	
		1 →5 →7 →12	14	
1 →6 →10 →12	24			

0.5	1	1 →3 →8 →13 1 →3 →9 →13 1 →4 →11 →13 1 →5 →9 →13 1 →6 →8 →13 1 →6 →11 →13	65 5 13 34 5 21	24386.2
	2	1 →3 →7 →12 1 →4 →8 →12 1 →4 →9 →12 1 →5 →7 →12 1 →6 →10 →12	10 15 18 22 23	
0.6	1	1 →3 →7 →13 1 →3 →8 →13 1 →3 →9 →13 1 →4 →11 →13 1 →5 →9 →13 1 →6 →11 →13	2 70 4 8 33 27	24386.2
	2	1 →3 →7 →12 1 →4 →8 →12 1 →4 →9 →12 1 →5 →7 →12 1 →6 →10 →12	3 15 22 28 22	
0.7	1	1 →3 →7 →13 1 →3 →8 →13 1 →3 →9 →13 1 →4 →11 →13 1 →5 →7 →13 1 →5 →9 →13 1 →6 →11 →13	5 70 3 4 5 32 27	26529.2
	2	1 →4 →8 →12 1 →4 →9 →12 1 →5 →7 →12 1 →6 →10 →12	15 26 31 21	

0.8	1	1 → 3 → 7 → 13	4	27487.2
		1 → 3 → 8 → 13	70	
		1 → 3 → 9 → 13	3	
		1 → 5 → 7 → 13	11	
		1 → 5 → 9 → 13	31	
		1 → 6 → 11 → 13	28	
	2	1 → 4 → 8 → 12	15	
		1 → 4 → 9 → 12	30	
		1 → 5 → 7 → 12	30	
		1 → 6 → 10 → 12	20	

## 1.2. Otrā uzdevuma atrisinājums

Tabula 1.2.1:  $(\alpha, \beta)$ -optimālais atrisinājums otrajam uzdevumam pie iespējamības mēra,  $\alpha = 0.7$

$\beta$	Prece	Maršruts	Plūsma	z		
0.1	1	1 → 3 → 4 → 11	3	27678.4		
		1 → 6 → 11	1			
		1 → 9 → 7 → 11	3			
		1 → 9 → 10 → 11	9			
	0.2	2	10 → 7 → 6 → 4 → 5		3	
	0.3		10 → 9 → 2 → 3 → 5		9	
			10 → 9 → 2 → 5		2	
			3		1 → 3 → 5	3
			4		10 → 1 → 6	21
0.3	1	1 → 3 → 4 → 11	3	28905.9		
		1 → 6 → 11	1			
		1 → 9 → 7 → 11	3			
		1 → 9 → 10 → 11	9			
	0.4	2	10 → 7 → 6 → 4 → 5		4	
			10 → 9 → 2 → 3 → 5		8	
			10 → 9 → 2 → 5		3	
			3		1 → 3 → 5	3
			4		10 → 1 → 6	22

0.6	1	1 →3 →4 →11	3	29933.6	
		1 →6 →11	3		
		1 →9 →7 →11	2		
		1 →9 →10 →11	9		
2		10 →7 →6 →4 →5	4		
		10 →9 →2 →3 →5	8		
		10 →9 →2 →5	3		
3		1 →3 →5	3		
4		10 →1 →6	22		
0.7	1	1 →3 →4 →11	2		30279.4
		1 →6 →11	4		
		1 →9 →7 →11	2		
		1 →9 →10 →11	9		
0.8		10 →7 →6 →4 →5	3		
		10 →7 →8 →3 →5	2		
0.9	2	10 →7 →9 →2 →3 →5	1		
		10 →9 →2 →3 →5	4		
		10 →9 →2 →5	6		
3		1 →3 →5	3		
4		10 →1 →6	23		

Tabula 1.2.2:  $(\alpha, \beta)$ -optimālais atrisinājums otrajam uzdevumam pie nepieciešamības mēra,  $\alpha = 0.7$

$\beta$	Prece	Maršruts	Plūsma	z
0.1	1	1 →3 →4 →11	2	37582.5
		1 →6 →11	7	
		1 →9 →7 →11	2	
		1 →9 →10 →11	8	
0.2	2	10 →7 →6 →4 →5	2	
0.3		10 →7 →6 →5	6	
0.4		10 →7 →8 →3 →5	1	
0.5		10 →7 →9 →2 →5	1	
		10 →9 →2 →5	6	

		10 → 9 → 2 → 3 → 5	2	
	3	1 → 3 → 5	5	
	4	10 → 1 → 6	26	
0.6	1	1 → 3 → 4 → 11	2	39738.57
		1 → 6 → 11	8	
		1 → 9 → 7 → 11	2	
		1 → 9 → 10 → 11	7	
0.7		10 → 7 → 6 → 4 → 5	2	
0.8	2	10 → 7 → 6 → 5	8	
0.9		10 → 9 → 2 → 5	8	
		10 → 9 → 2 → 3 → 5	1	
	3	1 → 3 → 5	6	
	4	10 → 1 → 6	27	

Tabula 1.2.3: Otrā uzdevuma mērķa funkcijas vērtības pie dažādām  $\alpha$  parametra vērtībām,  $\beta = 0.6$

$\alpha$	$z, \Pi$	$z, N$
0.1	28323.58	41657.82
0.2	28591.92	41337.95
0.3	28860.26	41018.07
0.4	29128.59	40698.2
0.5	29396.92	40373.32
0.6	29665.26	40058.45
0.7	29933.6	39738.57
0.8	30201.93	39418.7
0.9	30470.26	39098.82

## 2 Pielikums

### 2.1. Programmas kods

```
library ( lpSolve )

m <- 31 #rindas
n <- 62 #kolonnas
nodes <- 13 #virsoņu skaits
k <- 2 #patēriņa preču skaits
alpha <- 0.8 #mainīgs
beta <- 0.7 #mainīgs

#funkcija,kas veidota, lai nolasītu vērtību no trapeces, augoša
up.line <- function(a, b, pak1)
{
pak1*(b-a)+a
}

#funkcija,kas veidota, lai nolasītu vērtību no trapeces, dilstoša
down.line <- function(c, d, pak2)
{
pak2*(c-d)+d
}

iztekas <- c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5,
6, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11)
ietekas <- c(3, 4, 5, 6, 3, 4, 5, 6, 7, 8, 9, 7, 8, 9, 10, 11, 7, 9,
8, 10, 11, 12, 13, 12, 13, 12, 13, 12, 13, 12, 13)

source <- c(1, 1) #transportēšanas sākums -> mērķis
destination <- c(13, 12)
nodes <- 13 #virsoņu skaits

v1 <- read.table("fuzzycost1.txt")

fuzzy.cost <- matrix(c(v1), nrow=n, ncol=4, byrow = TRUE)

v2 <- read.table("fuzzycapacity1.txt")

fuzzy.capacity <- matrix(c(v2), nrow=m, ncol=4, byrow = TRUE)
```

```

fuzzy.demand <- matrix(c(100, 120, 135, 150, 60, 61, 75, 100),
nrow=k, ncol=4, byrow = TRUE)

#maina atkarībā no izmantotās sakarības
capacity <- rep(0, m)
for ( i in 1: m )
{
capacity[i] <- down.line(fuzzy.capacity[i, 3], fuzzy.capacity[i, 4], beta)
}

obj.fun <- rep(1, 2*n)
for ( i in 1: n )
{
obj.fun[i] <- 0
}

#maina atkarībā no izmantotās sakarības
demand <-rep(0, k)
for ( i in 1: k )
{
demand[i] <- up.line(fuzzy.demand[i, 1], fuzzy.demand[i, 2], beta)
}

#####MCFP#####

capacity_constr <- matrix (0 , m , 2*n ) #matrica caurlaidei
dem_constr <- matrix (0 , k , 2*n ) #matrica pieprasījumam
node_constr <- matrix (0, 2*nodes, 2*n) #matrica grafa uzdošanai

#capacity
for ( i in 1: m )
{
capacity_constr [i , i ] <- 1
capacity_constr [i , i+m ] <-1
}

#demand
for ( j in 1: k ) {
for ( i in 1: m ) {
if (destination[j] == ietekas[i] ) {
dem_constr[j, i+(j-1)*m] <- 1
}
}
}

#graph
r <- 1
for ( h in 1: nodes) {
for ( i in 1: m ) {
if (ietekas[i] == h ) {

```

```

node_constr[r, i] <- 1
} else if(iztekas[i] == h ) {
node_constr[r, i] <- -1
}
}
r <- r+1
}

for ( h in 1: nodes) {
for ( i in 1: m ) {
node_constr[h+nodes, i+m] <- node_constr[h, i]
}
}

#noņem tīrās iztekas un tīrās ietekas
node_constr <- node_constr[-c(1, 2, 12, 13, 14, 15, 25, 26),]

#**
#pirmās izmaksas,  $y(p)-cL(p)f(p) \geq 0$ 
cost1.y <-matrix(0, n, 2*n)
for ( i in 1: n )
{
cost1.y[i, i] <- -up.line(fuzzy.cost[i, 1], fuzzy.cost[i, 2], alpha)
cost1.y[i, i+n] <- -1
}

#otrās izmaksas,  $cR(p)f(p)-y(p) \geq 0$ 
cost2.y <-matrix(0, n, 2*n)
for ( i in 1: n )
{
cost2.y[i, i] <- down.line(fuzzy.cost[i, 3], fuzzy.cost[i, 4], alpha)
cost2.y[i, i+n] <- -1
}

#manuāls skaits- 8 2*2 iztekas 2*2 ietiekas
zer <-rep(0, 2*n+4*nodes-16)

constr <-rbind(capacity_constr, dem_constr, node_constr,
node_constr, cost1.y, cost2.y)
#savieno kopā divas matricas

constr.dir <- c(rep("<=", m ) , rep(">=", k ),
rep(">=", k*nodes-8 ), rep("<=", k*nodes-8 ), rep(">=", 2*n ) )

rhs <-c(capacity, demand, zer)

#risina DPU
y <- lp("min", obj.fun , constr , constr.dir , rhs , int.vec=1:62,
compute.sens = TRUE )

#####KODS 2#####

```

```

library( Rglpk ) #bibliotēkā atrodas funkcija, kas risina masīvu LPU

m <- 50 #rindas
n <- 100 #kolonnas
k <- 4
n2 <- m*k
n3 <- 2*n2
beta <- 0.7 #mainīgs
alpha <- 0.7 #mainīgs
nodes <- 11 #virsošnes

#funkcija,kas veidota, lai nolasītu vērtību no trapeces, augoša
up.line <- function(a, b, pak1)
{
pak1*(b-a)+a
}

#funkcija,kas veidota, lai nolasītu vērtību no trapeces, dilstoša
down.line <- function(c, d, pak2)
{
pak2*(c-d)+d
}

iztekas <- c(1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 4,
5, 6, 6, 7, 7, 7, 7, 8, 9, 10, 2, 3, 6, 9, 10, 3, 5, 9, 4,
5, 8, 5, 6, 8, 11, 6, 7, 11, 8, 9, 10, 11, 9, 10, 11)
ietekas <- c(2, 3, 6, 9, 10, 3, 5, 9, 4, 5, 8, 5, 6, 8,
11, 6, 7, 11, 8, 9, 10, 11, 9, 10, 11, 1, 1, 1, 1, 1, 2,
2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 6, 6, 7, 7, 7, 7, 8, 9, 10)

source <- c(1, 10, 1, 10) #transportēšanas sākums -> mērķis
destination <- c(11, 5, 5, 6)

v1 <- read.table("fuzzycost2.txt")

fuzzy.cost <- matrix(c(v1), nrow=n2, ncol=4, byrow=TRUE)

v2 <- read.table("fuzzycapacity2.txt")

fuzzy.capacity <- matrix(c(v2), nrow=m, ncol=4, byrow = TRUE)

fuzzy.demand <- matrix(c(15, 17, 18, 19, 13, 16, 17, 19
, 2, 3, 4, 6, 20, 23, 25, 27), nrow=k, ncol=4, byrow = TRUE)

capacity <- rep(0, m)
for ( i in 1: m )
{
capacity[i] <- down.line(fuzzy.capacity[i, 3], fuzzy.capacity[i, 4], beta)
}

```

```

l <- n2+1
obj.fun <- rep(0, n3)
for ( i in 1: n3 )
{
obj.fun[i]<- 1
}

demand <-rep(0, k)
for ( i in 1: k )
{
demand[i] <- up.line(fuzzy.demand[i, 1], fuzzy.demand[i, 2], beta)
}

#####MCLP#####

capacity_constr <- matrix (0 , m , n3 ) #matrica caurlaidei
dem_constr <- matrix (0 , k , n3 ) #matrica pieprasījūmam
node_constr <- matrix (0, k*nodes, n3)

#capacity
for ( i in 1: m ) {
capacity_constr [i , i ] <- 1
capacity_constr [i , i+m ] <-1
capacity_constr [i , i+m+m ] <-1
capacity_constr [i , i+m+m+m ] <-1
}

#matrica pieprasījuma apmierināšanai
for ( j in 1: k ) {
for ( i in 1: m ) {
if (destination[j] == ietekas[i] ) {
dem_constr[j, i+(j-1)*m] <- 1
}
}
}

#PLŪSMAS KĀ LPU DEFINEŠANA (1 prece)
r <- 1
for ( h in 1: nodes) {
for ( i in 1: m ) {
if (ietekas[i] == h ) {
node_constr[r, i] <- 1
} else if(iztekas[i] == h ) {
node_constr[r, i] <- -1
}
}
}
r <- r+1
}

#plūsmas definēšana visām precēm

```

```

for ( h in 1: nodes) {
for ( i in 1: m ) {
node_constr[h+nodes, i+m] <- node_constr[h, i]
node_constr[h+(2*nodes), i+(2*m)] <- node_constr[h, i]
node_constr[h+(3*nodes), i+(3*m)] <- node_constr[h, i]
}
}

#noņem tīrās iztekas un tīrās ietekas
node_constr <- node_constr[-c(1, 11, 16, 21, 23, 27, 39, 43),]

#pirmās izmaksas,  $y(p)-cL(p)f(p) \geq 0$ 
cost1.y <-matrix(0, n2, n3)
for ( i in 1: n2 )
{
cost1.y[i, i] <- -up.line(fuzzy.cost[i, 1], fuzzy.cost[i, 2], alpha)
cost1.y[i, i+n2] <- 1
}

#otrās izmaksas,  $cR(p)f(p)-y(p) \geq 0$ 
cost2.y <-matrix(0, n2, n3)
for ( i in 1: n2 )
{
cost2.y[i, i] <- down.line(fuzzy.cost[i, 3], fuzzy.cost[i, 4], alpha)
cost2.y[i, i+n2] <- -1
}

#papildus nosacījumi,jo grafs neorientēts
notdir <- matrix(0, 35, n3)
r <- 1
for ( j in 1: k )
{
for ( i in 1: m )
{
if (ietekas[i] == source[j])
{
notdir[r, i+m*(j-1)] <- 1
r <- r+1
}
else if ( iztekas[i] == destination[j])
{
notdir[r, i+m*(j-1)] <- 1
r <- r+1
}
}
}
}

zer <-rep(0, 2*k*nodes-16+n3+70)

constr <-rbind(capacity_constr, dem_constr, node_constr, node_constr,

```

```

cost1.y, cost2.y, notdir, notdir)

constr.dir <- c(rep("<=" , m ) , rep(">=" , k ), rep(">=" , k*nodes-8 ),
  rep("<=" , k*nodes-8 ), rep(">=" , n3 ), rep("<=" , 35 ), rep(">=" , 35 ) )

rhs <-c(capacity, demand, zer)

#atzīmē, kuru vērtību meklē veselos skaitļos
types <- rep("C", n3)
for ( i in 1: n2)
{
types[i] <- "I"
}

#risina DPU
y <- Rglpk_solve_LP(obj.fun, constr, constr.dir, rhs, bounds = NULL,
  types = types, max = FALSE)

answer <- y$solution
answer1 <- matrix (0 , n2, 4)
for ( j in 1: k)
{
for ( i in 1: m)
{
if (answer[i+(j-1)*m] > 0 )
{
answer1[i+(j-1)*m, 1] <- iztekas[i]
answer1[i+(j-1)*m, 2] <- ietekas[i]
answer1[i+(j-1)*m, 3] <- answer[i+(j-1)*m]
answer1[i+(j-1)*m, 4] <- answer[i+n2+(j-1)*m]/answer[i+(j-1)*m]
}
}
}
}

```

Bakalaura darbs “Nestrikta lineārā programmēšana preču optimālās plūsmas noteikšanai”  
izstrādāts LU Fizikas un Matemātikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Viktorija Capa

\_\_\_\_\_  
(paraksts) (datums)

Rekomendēju darbu aizstāvēšanai.

Vadītāja: prof. Dr. math. Svetlana Asmuss

\_\_\_\_\_  
(paraksts) (datums)

Recenzente: Dr. math. Inese Bula

\_\_\_\_\_  
(paraksts) (datums)

Darbs iesniegts Matemātikas nodaļā \_\_\_\_\_  
(datums)

Dekāna pilnvarotā persona: vecākā metodiķe Dzintra Holsta

Darbs aizstāvēts Valsts pārbaudījuma komisijas sēdē

\_\_\_\_\_ prot. Nr. \_\_\_\_\_, vērtējums \_\_\_\_\_  
(datums)

Komisijas sekretāre: asociētā profesore Ingrīda Uljāne \_\_\_\_\_  
(paraksts)