RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Informācijas tehnoloģijas institūts

Aleksandrs VALIŠEVSKIS
Vadības informācijas tehnoloģijas doktora programmas students

LĒMUMU PIEŅEMŠANA VARBŪTISKĀS NENOTEIKTĪBAS UN IZPLŪDUMA APSTĀKĻOS

Promocijas darbs

Zinātniskais vadītājs
Dr.habil.sc.comp., profesors
A. BORISOVS

Rīga 2006

RĪGAS TEHNISKĀ UNIVERSITĀTE
DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS FAKULTĀTE
Informācijas tehnoloģijas institūts

LĒMUMU PIEŅEMŠANA VARBŪTISKĀS NENOTEIKTĪBAS UN IZPLŪDUMA APSTĀKĻOS

Aleksandrs Vališevskis

Anotācija

Promocijas darbs ir veltīts lēmumu pieņemšanas metodes izstrādei divu nenoteiktības avotu apstākļos. Proti, gadījumam, kad pieejamā informācija ir gan nedeterminēta, gan izplūdusī. Piedāvātā metode ir balstīta uz izplūdušo granulu jēdziena.

Veikta pētījuma ietvaros izstrādātā metode iekļauj sevī šādus posmus: alternatīvu un kritēriju ģenerēšana; alternatīvu aprakstīšana ar izplūdušo granulu palīdzību; varbūtību, ka alternatīvu kritēriju vērtības būs vienādas ar vēlamajām vērtībām, noteikšana; alternatīvu informatīvuma novērtēšana; preferenču modelēšana; alternatīvu sakārtojuma iegūšana; izveidota modeļa jutīguma analīze; iegūto rezultātu analīze.

Darbā ir izanalizētas eksistējošas lēmumu pieņemšanas metodes kā viena, tā arī vairāku nenoteiktības avotu apstākļos.

Izstrādājot metodi alternatīvu informatīvuma novērtēšanai tika atrisināts entropijas vispārināšanas uzdevums intervālu varbūtību gadījumam. Turklāt, ir pierādīts, ka vispārinātā entropija ir aditīva.

Darbā izstrādātās metodes praktiskā pielietojamība ir pierādīta ar tādu lemšanas uzdevumu risināšanu, kuros uz nenoteiktās sākuma informācijas pamata tiek lemts par celulozes kombināta būvēšanas projekta izvēli un tiek prognozēts jaunā produkta dzīves cikla posms.

Eksperimentu veikšanai *Microsoft Visual C++ 6.0* vidē ir izveidota objektorientēta programmatūra.

Darbs sastāv no 156 lappusēm, 55 formulām, 45 attēliem un 18 tabulām.

RIGA TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
Information Technology Institute

DECISION-MAKING UNDER PROBABILISTIC UNCERTAINTY AND FUZZINESS

Aleksandrs Vališevskis

Abstract

The Ph.D. Thesis is devoted to the development of decision-making method that can be used under two sources of uncertainty – non-determinism and fuzziness. The proposed method is based on the fuzzy granule notion.

The research has yielded a decision support method that includes the following stages: generation of alternatives and criteria; description of the alternatives with the help of fuzzy granules; evaluating probability that the alternatives' criteria values will be equal to the desired value; ranging of the alternatives; sensitivity analysis of the designed model and analysis of the results obtained.

Existing methods for decision-making under one source and several sources of uncertainty have been analysed.

During the development of the method for alternatives' informativeness evaluation, the entropy generalisation problem to the case of interval probabilities has been solved. Besides that, it is shown that the entropy obtained is additive as well.

The method developed has practical applications, which is demonstrated by the examples considered. One of the examples considers project selection problem. The example analyses pulp mill construction projects based on uncertain input data. The second example considers forecasting of a product's life cycle stage.

Object-oriented software has been developed in *Microsoft Visual C++ 6.0* environment in order to carry out the experiments.

The Thesis consists of 156 pages, 55 formulae, 45 figures and 18 tables.

РИЖСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ
Институт информационной технологии

ПРИНЯТИЕ РЕШЕНИЙ В УСЛОВИЯХ ВЕРОЯТНОСТНОЙ НЕОПРЕДЕЛЕННОСТИ И НЕЧЕТКОСТИ

Александр Валишевский

Аннотация

Диссертация посвящена разработке метода принятия решений в условиях двух источников неопределенности – недетерминированности и нечеткости. Предлагаемый метод основан на понятии нечетких гранул.

В рамках проведенного исследования разработан метод, включающий в себя следующие этапы: генерация альтернатив и критериев; описание альтернатив при помощи нечетких гранул; оценка вероятности того, что параметры альтернатив примут желаемое значение; оценка информированности об альтернативах; моделирование предпочтений; упорядочивание альтернатив; анализ чувствительности построенной модели; анализ полученных результатов.

Проведен анализ существующих методов поддержки принятия решений как в условиях одного источника, так и в условиях нескольких источников неопределенности.

В рамках разработки метода оценки информированности об альтернативах решена задача обобщения энтропии на случай интервальных вероятностей. Кроме того, показано, что обобщенная энтропия также является аддитивной.

Разработанный метод имеет практическое применение, о чем свидетельствуют приведенные примеры. В одном из примеров рассматривается задача выбора проектов, в которой на основании неопределенной исходной информации анализируются проекты строительства целлюлозной фабрики. Во втором примере на основании неопределенной исходной информации прогнозируется этап жизненного цикла продукта.

Для проведения экспериментов в среде *Microsoft Visual C++ 6.0* разработано объектно-ориентированное программное обеспечение.

Работа состоит из 156 страниц, 55 формул, 45 рисунков и 18 таблиц.

Darbā izmantojamie saīsinājumi

Saīsinājums	**Atšifrējums**
AHP	Analytic Hierarchy Process
ANFIS	Adaptive Network-Based Fuzzy Inference System
ANGIE	Adaptive Network for Granular Information and Evidence Processing
CW	Computing with Words
EC	Expected Certainty
ELECTRE	ÉLimination Et Choix Traduisant la Réalité
EΠ	Expected Possibility
F-GRANULA	Izplūdusī granula
MS	Microsoft
PROMETHEE	Preference Ranking Organization Method for Enrichment Evaluations
PRUF	Possibility Relational Universal Fuzzy
ROI	Return on Investment
SDK	Sākuma datu kopa
SMART	Simple Multi-Attribute Rating Technique

Saturs

IEVADS

Lēmumu analīzes mērķis ir palīdzēt lēmumu pieņemošai personai labāk izprast vidi, kurā tiek pieņemts lēmums un sekmēt informatīvāka lēmuma pieņemšanai. Tas nozīmē, ka analīzes rezultātus var izmantot, lai noskaidrotu dažādu alternatīvu lēmumu stabilitāti, jūtīgumu pret nelielām izmaiņām, kā arī lai strukturētu lēmumu un padarītu to par pārskatāmāku. It īpaši tas attiecās uz problēmām ar vairākām alternatīvām un kritērijiem. Lēmumu pieņemšanas metode piedāvā algoritmisku pieeju lēmumu strukturēšanai un preferenču modelēšanai un tā rezultātā – alternatīvu ranžējuma iegūšanai, kas ir atkarīgs no lēmumu pieņemošas personas sniegtas informācijas.

Novērojot lēmumu pieņemšanas metožu attīstību laika gaitā var redzēt tieksmi tuvināt alternatīvu aprakstīšanu reālās pasaules apstākļiem, jeb paplašināt metožu aprakstīšanas spējas, lai būtu iespējams analizēt lēmumus, kuri ir jāpieņem nenoteiktajā vidē, kuras aprakstīšanai nepietiek ar agrākiem līdzekļiem. Piemēram, grāmatā [Keeney&Raiffa, 1976] ir aprakstīta lēmumu analīzes metode, kas balstīta uz varbūtību teorijas un lietderīguma teorijas. Šajā pieejā tiek izvirzīta hipotēze, ka cilvēka attieksmi pret risku var aprakstīt ar lietderīguma funkcijas palīdzību, bet nenoteiktības aprakstīšanai pietiek ar varbūtību teorijas līdzekļiem. Viena no šīs pieejas acīmredzamākām pielietošanas jomām ir apdrošināšana, kurā riska prēmija ir pa tiešo saistīta ar riska gadījuma varbūtību. Savukārt, lietderīguma teorijas līdzekļus var izmantot, lai riska prēmijas lieluma aprēķinā varētu iekļaut attieksmi pret risku. Ranžēšana šajā pieejā ir atkarīga no sagaidāmās lietderības.

Par atkāpšanos no šīs pieejas, kuru nosacīti var nosaukt par formālu amerikāņu lēmumu analīzes skolu, var uzskatīt eiropiešu piedāvātas heiristiskās metodes PROMETHEE [Brans u.c., 1986] un ELECTRE [Roy, 1991], kuras upurē formālismu par labu tuvināšanai cilvēka domāšanas veidam. PROMETHEE metodē ir ievests vispārināta kritērija jēdziens, kas nosāka vairākus pārspēšanas (*outranking* – angļu val.) veidus, sīkāk tas ir aprakstīts 2.1.3. paragrāfā. Savukārt, ELECTRE metodē tiek ievesta heiristiska pārspēšanas relācija, kura izmanto konkordances un diskordances vērtības, lai noteiktu dominējušo alternatīvu. Šīs metodes nav tik formālas, tādējādi ar to palīdzību var ērtāk aprakstīt dažādas lēmumu problēmas, kuras vai nu neprasa formālā modeļa izveidi, vai tas prasītu nepieņemami daudz līdzekļu. Viena no šo metožu iespējamām pielietošanas jomām ir personāla atlase, kura ir raksturota ar to, ka saistība starp kritērijiem šajā uzdevumā nav precīzi noteikta, novērtējumi var būt skaitliskie vai kvalitatīvi (pēc noteiktās skalas) un saistība starp kritērijiem nav lineāra, piemēram, ir iespējama situācija, kad viena kritērija zemu vērtību nevar kompensēt ar cita kritērija augstu vērtību. Piemēram, ja tiek rīkots konkurss uz sekretāres amatu un tiek ņemtas vērā viņas drukāšanas ātrums un precizitāte, kas savā starpā noteikti ir saistīti, tad zemu drukāšanas ātrumu nevar kompensēt ar lielu precizitāti.

Nākamais solis lēmumu pieņemšanas metožu attīstībā tika sperts uzreiz pēc izplūdušo kopu teorijas izveides [Zadeh, 1965], jo tā ļāva iekļaut alternatīvu aprakstā jauno nenoteiktības veidu, proti, izplūdumu. Ar šo metožu palīdzību kļuva iespējams alternatīvu aprakstā izmantot izplūdušās vērtības. Šīs metodes ļauj izmantot izplūdušas vērtības problēmas aprakstā, kā arī nosakot kritēriju svarus. Uz izplūdušās loģikas balstītās metodes var izmantot kad pieejamā informācija ir izplūdusī un kritērijiem nav iespējams uzdot precīzās vērtības, vai ja precīzo vērtību iegūšana ir pārāk dārga. Galvenais šo metožu trūkums ir tas, ka tās prasa ļoti daudz skaitļošanas resursu. Profesors L. Zadē norāda uz šo problēmu un pat piedāvā tās risināšanai izmantot ģenētiskos algoritmus [Zadeh, 1996]. Šīs metodes var izmantot tādās jomās kā projektu izvēle, ja to raksturojumi nav precīzi zināmi.

Pēc uz izplūdušās loģikas balstīto lēmumu pieņemšanas metožu izstrādes parādījās metodes, kurās vienlaicīgi ir iespējams izmantot dažādus nenoteiktības tipus. Proti, alternatīvu aprakstā ir iespējams izmantot gan precīzās, gan izplūdušās vērtības, gan varbūtības. Šādu metožu piemēri ir aprakstīti rakstos [Ben Amor u.c., 2004; Zadeh, 1996; Zadeh, 1978]. To galvenā īpašība ir tāda, ka tie ļauj alternatīvu aprakstā izmantot dažāda veida nenoteikto informāciju un līdz ar to aprakstīt vairākus nenoteiktības avotus, kuri ir lielākā daļā reālās pasaules uzdevumu. Par šo metožu pielietošanas jomu var uzskatīt uzdevumus ar īpaši augstu nenoteiktības pakāpi, kad objekti tiek raksturoti ar izplūdušām vērtībām, pie kā šīs vērtības tiek pakļautas varbūtību sadalījumam. Piemēram, ja ir jāizvēlas viens no projektiem pie nosacījuma, ka dažādi projektu alternatīvas tiek raksturotas ar izplūdušām vērtībām, kuras nav zināmas precīzi. Proti, ja projektu parametri tiek aprakstīti kā izplūdušo vērtību varbūtību sadalījums.

Līdzīgi iepriekšējam lēmumu pieņemšanas metožu tipam, arī metodes, kurās ir apvienoti izplūdušās loģikas un varbūtību teorijas elementi, prasa daudz skaitļošanas resursu (izvedums tiek veikts saskaņā ar likumiem, tiek veikta pilna pārlase, līdz ar ko sarežģītība ir eksponenciāla, tādējādi, ja ir daudz parametru, tad metodes kļūst nelietojamas). Promocijas darbā tiek apskatīta pēdējā veida lēmumu pieņemšanas metode. Proti, tiek pētīta lēmumu pieņemšanas vide ar vairākiem nenoteiktības avotiem. Alternatīvu aprakstīšanai tiek izmantotas izplūdušās granulas, kuras ļauj vienlaicīgi izmantot gan izplūdušo, gan varbūtisko informāciju. Izmantojot izplūdušās granulas nav iespējams ievest aprakstā tādus līdzekļus kā izplūdušie kvantori [Zadeh, 1978], toties tas ļauj ierobežot skaitļošanas apjomu (jo netiek izmantoti izveduma likumi).

Darbā ir izstrādāta uz izplūdušām granulām balstīta metode, tai skaitā ir izstrādāta metode alternatīvu aprakstīšanai, vērtēšanai, alternatīvu ranžēšanai un jūtīguma analīzei. Jūtīguma analīzei var izmantot gan parastu procedūru [Clemen&Reilly, 2004], gan uz adaptīvā tīkla ANGIE (*Adaptive Network for Granular Information and Evidence Processing* – angļu val.) balstīto procedūru [Vališevskis, 2002]. Turklāt, ir izstrādāta uz entropijas balstīta metode alternatīvu informatīvuma novērtēšanai [Vališevskis&Borisov, 2004].

Pēc izstrādātas metodes apraksta ir apskatīti divi piemēri, lai parādītu kā izstrādāto metodi var izmantot praksē.

Problēmas aktualitāte

Tradicionāli lēmumu analīzes uzdevumos tiek apskatīts tikai viens nenoteiktības veids. Tomēr, reālajā pasaulē notikumi parasti ir gan izplūduši, gan nedetermināti. Proti, notikumi ir nedetermināti, jo tos var nodefinēt ar kādu subjektīvu varbūtību, un tie ir izplūduši, jo notikuma izpausme var būt tikai daļēja. Klasiskajā varbūtību teorijā tiek pieņemts, ka notikums vienmēr vai nu pavisam neizpildās, vai nu tas pilnīgi izpildās. Reālajā pasaulē notikums var izpildīties tikai daļēji vai pat pats notikums var būt izplūdis. Viens no jaunākajiem un straujāk attīstītākajiem virzieniem izplūdušās loģikas pētījumos ir uz percepcijas balstīta lēmumu pieņemšana, kurā šis uzdevums tiek apskatīts vairāku nenoteiktības avotu apstākļos. Izplūdušās granulas ir viena no uz percepcijas balstītas lēmumu pieņemšanas virziena jomām (sk. 1.3.3.1. paragrāfu). Promocijas darbā izstrādātā metode problēmas apgabala aprakstā ļauj izmantot gan izplūdušas, gan nedeterminētas vērtības.

Efektīva lēmumu pieņemšanas metode ir spējīga dot pozitīvu stimulu tautsaimniecībā iesaistītajiem uzņēmumiem, jo ar tās palīdzību ir iespējams pieņemt labāku un informatīvāku lēmumu, kas ņem vērā daudzveidīgu reālās pasaules nenoteiktību.

Pētījuma priekšmets

Pētījuma priekšmets ir lēmumu pieņemšanas metode, kura ir paredzēta izmantošanai vidē ar vairākiem nenoteiktības avotiem, proti, vidē, kura ir gan nedeterminēta, gan izplūdusi. Šo metodi ir paredzēts izmantot gadījumā, kad ir galīgs skaits kritēriju un alternatīvu.

Pētījuma hipotēzes

Pētījumu gaitā tiek izvirzītas šādas hipotēzes:

1. Apskatāmo lēmuma pieņemšanas vidi var pietiekoši precīzi aprakstīt ar izplūdušo granulu palīdzību, proti, izmantojot varbūtību teorijas un izplūdušo kopu teorijas līdzekļus.
2. Novērtējot alternatīvu informatīvumu tiek pieņemts, ka izveidotas sistēmas ir neatkarīgas.

Pētījuma metodes

Promocijas darbā tiek izmantota izplūdušo kopu teorija, iespējamību teorija, varbūtību teorija, matemātiskā analīze, informācijas teorija, optimizācijas teorija. Tai skaitā, tiek izmantotas izplūdušās granulas, Šenona entropija, Lagranža reizinātāju metode, gradienta lejupslīdes metode, izplūdušā izveduma metodes, lēmumu modeļa jūtīguma analīze.

Darba zinātniskais jaunieguvums

Darba gaitā tika veikti sekojošie pētījumi un izstrādāti šādi līdzekļi, kuri veido nenoteiktajā vidē izmantošanai paredzētu lēmumu pieņemšanas metodi:

1. Ir izstrādāts adaptīvais tīkls ANGIE (*Adaptive Network for Granular Information and Evidence Processing*) un algoritms tā apmācīšanai.
2. Ir izstrādāta metode, kura ļauj ar adaptīvā tīkla ANGIE palīdzību noteikt alternatīvu svarīgākos parametrus.
3. Šenona entropija tika vispārināta intervālu varbūtību gadījumam un tika parādīts, ka vispārinātā definīcija arī ir aditīva.
4. Ir izstrādāta alternatīvas informatīvuma novērtēšanas metode, kura izmanto vispārināto Šenona entropijas definīciju.
5. Ir izstrādāta alternatīvu sakārtošanas metode, ar kuras palīdzību ir iespējams iegūt pilno vai daļējo sakārtojumu.
6. Augstāk minētie līdzekļi tika apvienoti vidē ar vairākiem nenoteiktības avotiem izmantošanai paredzētajā lēmumu pieņemšanas metodē.

Praktiskā vērtība

Darbā ir izstrādāta nenoteiktajās vidēs ar diviem nenoteiktības avotiem izmantošanai paredzēta lēmumu pieņemšanas metode. Atšķirībā no citām metodēm, piedāvātājā metodē nav jākonstruē alternatīvu kritēriju vērtību matrica. Turklāt, izstrādātā metode tiek izmantota gadījumos, kad vide ir tik nenoteikta, ka šo matricu ar precīzām vērtībām nevar sastādīt. No otrās puses, izstrādātā metode prasa mazāk skaitļošanas resursu, nekā eksistējošās metodes, kuras ļauj vienlaicīgi izmantot dažādus nenoteiktības veidus. Darba gaitā ir izanalizēti daži praktiskie lēmumu pieņemšanas piemēri nenoteiktajā vidē, kuri parāda kā izstrādāto metodi var pielietot praksē.

Darba aprobācija

Par darba rezultātiem tika ziņots sekojošajās zinātniskajās konferencēs:

1. International Conference on Operational Research "Simulation and Optimisation in Business and Industry", SOBI 2006, Tallinn, Estonia, May 17-20, 2006.
2. The Tenth Estonian Winter School in Computer Science, Palmse, Estonia, February 27 – March 4, 2005.
3. International Conference "Managing Uncertainty in Decision Support Models" (MUDSM 2004), Coimbra, Portugal, September 22-24, 2004.
4. International Conference on Fuzzy Sets and Soft Computing in Economics and Finance, FSSCEF 2004, St. Petersburg, Russia, June 17-20, 2004.
5. The Ninth Estonian Winter School in Computer Science, Palmse, Estonia, February 29 – March 5, 2004.
6. International Conference on Soft Computing and Computing with Words in System Analysis, Decision and Control, ICSCCW'2003, Antalya, Turkey, September 9-11, 2003.
7. The Second Estonian Summer School on Computer and Systems Science, Taagepera, Estonia, August 10 – 14, 2003.
8. International conference "Environment. Technology. Resources.", Rezekne, Latvia, June 26-28, 2003.
9. Seventh International Conference on Cognitive and Neural Systems, ICCNS'2003, Boston, USA, May 28 – 31, 2003.
10. International Conference "Modelling and Simulation of Business Systems", MoSiBuS'2003, Vilnius, Lithuania, May 13-14, 2003.
11. The Eighth Estonian Winter School in Computer Science, Palmse, Estonia, March 2–7, 2003.
12. International Conference on Computational Intelligence for Modelling, Control and Automation CIMCA'2003, Vienna, Austria, February 12-14, 2003.
13. International conference "Information Society and Modern Business", Ventspils, Latvia, January 31 - February 1, 2003.

14. Fifth International Conference on Application of Fuzzy Systems and Soft Computing ICAFS-2002, Milan, Italy, September 17-19, 2003.
15. International scientific conference Traditions and Innovations in Sustainable Development of Society, Rezekne, Latvia, February 28-March 2, 2002.

Publikācijas

Pētījuma rezultāti ir publicēti 13 zinātniskajos rakstos. Piecas referātu tēzes ir iekļautas konferenču tēžu krājumos. Pilns publikāciju saraksts ir atrodams izmantotās literatūras sarakstā.

Zemāk ir saraksts ar referātu tēzēm:

1. Vališevskis A. (2005). Calculation of Shannon's Interval Entropy, The Tenth Estonian Winter School in Computer Science, Palmse, Estonia, February 27 – March 4.
2. Vališevskis A. (2004). Sets as Hypercube Points or Why do Midpoint Paradoxes Occur, The Ninth Estonian Winter School in Computer Science, Palmse, Estonia, February 29 – March 5.
3. Vališevskis A. (2003). Fuzzy-Information-Based Risk Assessment, The Second Estonian Summer School on Computer and Systems Science, Taagepera, Estonia, August 10 – 14.
4. Vališevskis A. (2003). Using Adaptive Networks to Process Fuzzy-Granular Information, Seventh International Conference on Cognitive and Neural Systems, ICCNS'2003, Boston, USA, May 28 – 31.
5. Vališevskis A. (2003). Interval-Valued Entropy, a Generalised Concept, The Eighth Estonian Winter School in Computer Science, Palmse, Estonia, March 2 – 7.

1. LĒMUMU PIEŅEMŠANA NEDETERMINĒTĪBAS UN IZPLŪDUMA APSTĀKĻOS

Kā ir norādīts iepriekšējā nodaļā, pēdējais lēmumu pieņemšanas metožu attīstības posms ir lēmumu analīze vidē ar vairākiem nenoteiktības avotiem. Proti, tiek analizētas alternatīvas, kuras tiek aprakstītas izmantojot gan izplūdušās vērtības, gan varbūtību vērtības. Tie ir divi dažādi nenoteiktības veidi [Kosko, 1992], jo izplūdums ir objekta raksturojuma neatņemama daļa, bet varbūtība ar laiku var izzust. Piemēram, ja garām braucošas mašīnas ātrums tika novērtēts kā "ļoti liels", tad šīs novērtējums būs spēkā arī ja tiks noskaidrots precīzs ātrums (kas principā nav iespējams, mērinstrumentu neprecizitātes dēļ). Savukārt, ja ātrums sākumā ir novērtēts kā varbūtību sadalījums, tad pēc precīza ātruma noteikšanas varbūtiskā nenoteiktība izzūd. Tādējādi, šie nenoteiktības veidi ir jāapskata atsevišķi.

Šajā nodaļā ir apskatīti ar izplūdušo kopu teoriju saistītie jautājumi, kā arī entropija informācijas teorijas kontekstā, citi entropijas jēdzieni un to saistība ar nejaušību.

1.1. Lēmumu pieņemšanas uzdevuma vispārīgā nostādne un lemšanas metožu klasifikācija

Lēmumu pieņemšanas uzdevums rodas, kad vēlamo rezultātu var sasniegt vairākos veidos. Vispārīgo lēmumu pieņemšanas uzdevuma nostādni var definēt sekojošā veidā. Pieņemsim, ka mums ir dota alternatīvu kopa *X* un iznākumu kopa *Y*. Pie kā, katra alternatīva $x_i \in X$ ir saistīta ar atbilstošu iznākumu $y_i \in Y$. Turklāt, ir alternatīvas izvēles kvalitātes novērtēšanas mehānisms. Parasti tiek vērtēta iznākuma kvalitāte. Dažos gadījumos var vērtēt alternatīvu kvalitāti un neanalizēt iznākumu kopu [Černoruckij, 2005]. Piemēram, ja iznākumu kopa ir bezgalīga vai visām alternatīvām iznākums atšķiras. Ir jāizvēlas alternatīva, kuras kvalitātes novērtējums ir vislabākais.

Ja saistība starp alternatīvām un iznākumiem ir viennozīmīga, tad saka, ka tiek risināts lēmumu pieņemšanas uzdevumus noteiktības apstākļos. Ja saistība nav deterministka, piemēram, ja ir zināms tikai iznākuma vērtību intervāls, tad saka, ka uzdevums tiek risināts nenoteiktības apstākļos. Ja saistība ir varbūtiska, tad saka, ka uzdevums tiek risināts riska apstākļos. Ja saistība ir izplūdusi, tad saka, ka uzdevums tiek risināts izplūduma apstākļos.

Ja katram iznākumam var piekārtot reālo skaitli, tad mērķa funkcija jeb kritērija funkcija f ir attēlojums $f : Y \rightarrow R$. Saka, ka iznākums y_i ir labāks par iznākumu y_j, ja $f(y_i) > f(y_j)$, iznākumi ir ekvivalenti, ja $f(y_i) = f(y_j)$. Šīs situācijas tiek apzīmētas, attiecīgi, šādi: $y_i \succ y_j$ un $y_i \sim y_j$.

Tomēr, parasti iznākums y tiek apzīmēts ar vairākām vērtībām, proti, ja ir vairāki kritēriji. Šajā gadījumā lēmuma pieņemšanas modeļi kļūst sarežģītāki, nekā viena kritērija gadījumā. Turklāt, parasti kritēriji ir pretrunīgi, proti, tie sasniedz maksimumu dažādos punktos. Tādējādi, risinot šādus uzdevumus rodas ne tikai algoritmiskas grūtības, bet arī konceptuālie jautājumi: kas šajā gadījumā tiek saprasts ar optimālo risinājumu? Tādēļ daudzkritēriju lēmumu pieņemšanas uzdevums ir atrast stabilu alternatīvu (ar jūtīguma analīzes palīdzību, sk. 2.1.7. paragrafu), kurai ir apmierinošs iznākums.

Lēmumu pieņemšanas metožu klasifikācijai ir izvirzītas šādas pazīmes (sk. 1.1. att.):

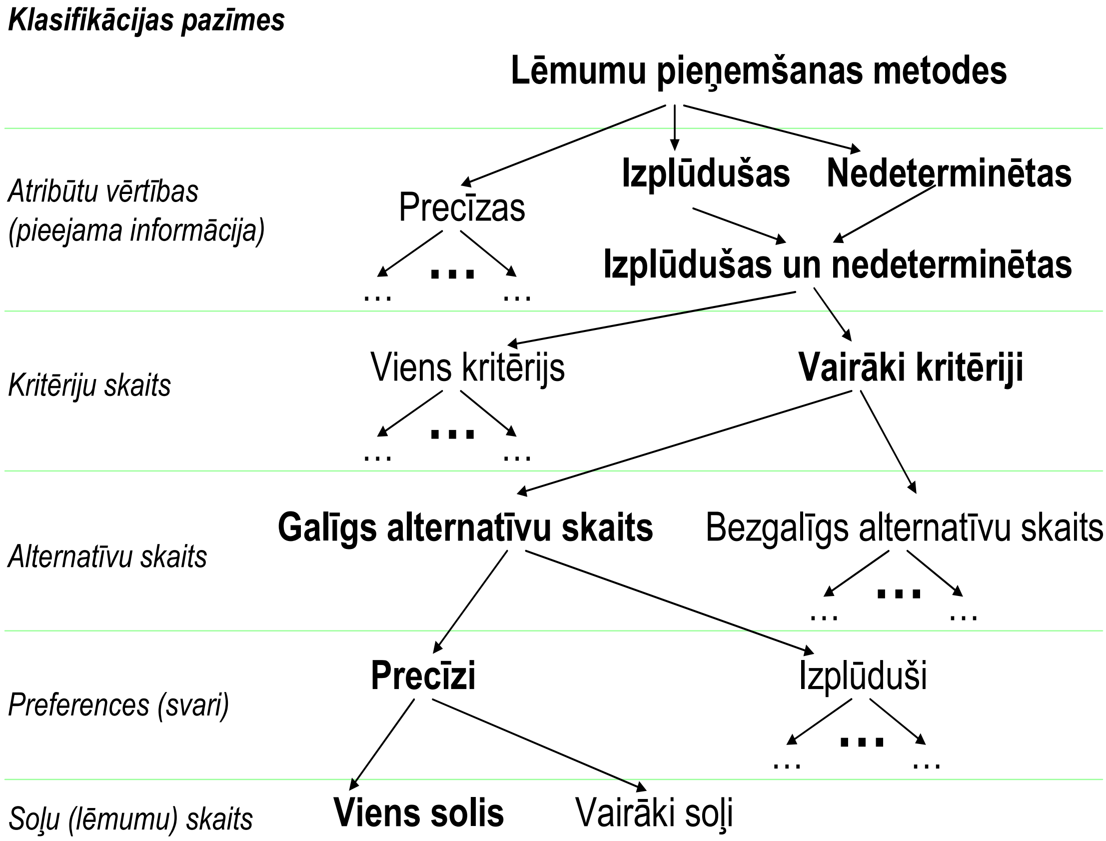
1. Atribūtu vērtību tips. Tās var būt precīzas, izplūdušās, nedeterminētas un gan izplūdušas, gan nedeterminētas.
2. Kritēriju skaits. Alternatīvas var vērtēt pēc viena vai vairākiem kritērijiem.
3. Alternatīvu skaits. Alternatīvu kopa var būt galīga vai bezgalīga.
4. Svaru vērtību tips. Tās var būt precīzas vai izplūdušās.
5. Soļu skaits. Proti, lēmums ir jāpieņem vienu reizi, vai vairākas reizes.

Promocijas darbā izstrādātā lēmumu pieņemšanas metode 1.1. attēlā it atzīmēta ar trekniem burtiem.

1.2. Pētījuma priekšmets un lēmumu pieņemšanas vides raksturojums

Pētījuma priekšmets ir lēmumu pieņemšanas metode, kura ir paredzēta izmantošanai vidē ar vairākiem nenoteiktības avotiem, proti, nedeterminētajās un izplūdušajās vidēs. Tiek izstrādāta lēmumu pieņemšanas metode ar vairākiem kritērijiem un alternatīvām, kuru skaits ir galīgs.

Lēmumu pieņemšanas vide ir raksturota ar to, ka tajā ir gan izplūdušās, gan varbūtiskās nenoteiktības avots. Proti, lai precīzi nomodelētu lēmuma problēmu, objektu aprakstā ir jāizmanto gan izplūdušās vērtības, gan varbūtības. Šo nenoteiktību apvienošana tiek panākta pateicoties tam, ka alternatīvu atribūtu aprakstā tiek izmantotas izplūdušas granulas (sk. 1.3.3.1. apakšparagrāfu), kuras ir izplūdušo vērtību varbūtību sadalījums.



1.1. att. Lēmumu pieņemšanas metožu klasifikācija

Šāda lēmumu pieņemšanas vide ir raksturīga uzdevumiem, kuros par analizējamām alternatīvām ir pieejams ārkārtīgi maz informācijas. Piemēram, ja aprakstā tiek izmantotas nākotnes vērtības, kuras šobrīd nav zināmas (piemēram, vēl neuzsākta projekta rādītāji tā pabeigšanas brīdī, ekspertu novērtējumi), vai ja aktuāla informācija nav precīzi zināma un ir pieejami aptuveni ekspertu novērtējumi vai līdzīgi dati. Piemēram, izstrādājot pārdošanas stratēģiju aktuāla informācija par pārdošanām nav pieejama, jo tā vispirms jāapkopo un tā tiek iegūta ar aizkavi.

1.3. Izplūdums un nenoteiktība lēmumu pieņemšanas uzdevumos

Izplūduma jēdziens, kā definēts rakstā [Zadeh, 1965], attiecas uz situācijām, kurās neprecizitātes avots ir nevis nejaušie mainīgie vai stohastiskie procesi, bet tādas jēdzienu klases, kurām nav uzlikti stingri definēti ierobežojumi, piemēram "jauno cilvēku klase", "klase, kas sastāv no skaitļiem daudz lielākiem par desmit" utt.

Kā iepriekš tika minēts, izplūdums ir mūsu pasaules neatņemama daļa un ļaujot modelēt to lēmumu pieņemšanas procesā, modeli ir iespējams padarīt precīzāku un tuvāku reālajai pasaulei. Līdz ar ko lēmējpersona varēs pieņemt informatīvāku lēmumu, jo izmantojot varbūtību teoriju vien, izplūdumu nevar pilnībā nomodelēt (sk. rakstu [Zadeh, 1986]).

1.3.1. Izplūdušas apakškopas un izplūdusi loģika

Izplūdusī apakškopa ir pamatjēdziens, ar kura palīdzību ir iespējams kvantitatīvi aprakstīt izplūdumu. Izplūdusi apakškopa ir klase, kurā katram elementam ir definēta piederības vērtība starp pilno piederību un nepiederību. Tādējādi izplūdusi apakškopa tiek raksturota ar piederības funkciju, kura katram elementam piekārto skaitli no intervāla [0, 1], kurš nosaka šī elementa piederības pakāpi izplūdušajai apakškopai.

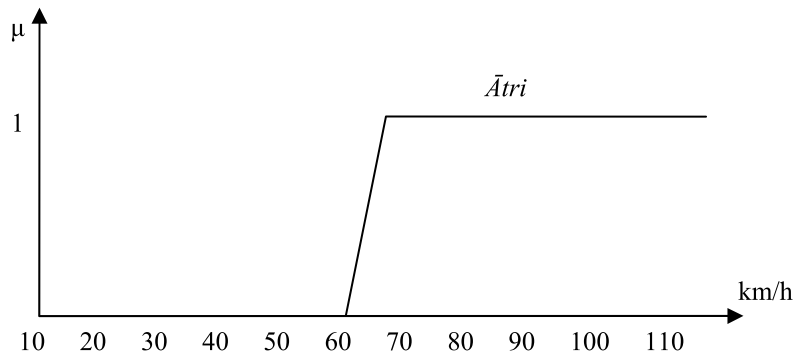
Parasti izplūdums rodas intervālu robežās. Apskatīsim vienu piemēru, kuru ir sniedzis prof. L. Zadē, aprakstot izplūduma jēdzienu. Ir zināms, ka daži cilvēki ir plikgalvji un daži nav. Ja mēs izveidojam rindu no visiem cilvēkiem un sakārtojam to pēc matu daudzuma cilvēku galvās, tad mēs nevarēsim atrast tādu vietu šajā rindā, kuru mēs varētu uzskatīt par *robežu* starp plikgalvjiem un cilvēkiem, kuri nav plikgalvji. Tādējādi, robeža starp klasēm "plikgalvis" un "neplikgalvis" ir izplūdusi, jo dažādi cilvēki ap šo robežpunktu ir noteiktā mērā gan plikgalvji, gan neplikgalvji (proti, cilvēki ar noteikto pakāpi pieder abām kopām).

Elementa *x* piederību kopai *A* raksturo piederības funkcija, kas tiek apzīmēta ar $\mu_A(x)$. Tādejādi izplūdusī kopa *A* telpā *X* tiek raksturota ar piederības funkciju μ_A, kura ir definēta apgabalā *X* un pieņem vērtības no intervāla *[0, 1]*, tādas, ka jo $\mu_A(x)$ vērtība ir tuvāk vieniniekam, jo augstāka ir *x* piederības pakāpe kopai *A* [Zadeh, 1965]. Piemēram, 1.2. attēlā ir parādīta izplūdusī kopa "Ātri".

Apskatīsim dažus izplūdušo kopu pamatjēdzienus un pamatoperācijas ar tām.

Vienlīdzība. Divas izplūdušās kopas *A* un *B* telpā *X* ir vienādas (*A*=*B*) tad un tikai tad, ja $\mu_A(x) = \mu_B(x),\ x \in X$.

Ietilpšana. Izplūdusī kopa *A* ietilpst izplūdušajā kopā B ($A \subset B$) tad un tikai tad, ja $\mu_A(x) \le \mu_B(x),\ x \in X$. Piemēram, izplūdusī kopa "skaitļi, daudz lielāki par desmit" ietilpst neizplūdušajā kopā "skaitļi, lielāki par desmit".



1.2. att. Izplūdusī kopa "Ātri"

Papildinājums. Izplūdusī kopa *A'* ir izplūdušās kopas *A* papildinājums tad un tikai tad, ja $\mu_{A'} = 1 - \mu_A$.

Apvienojums. Divu izplūdušo kopu *A* un *B* apvienojums ($A \cup B$) tiek definēts kā mazākā izplūdusī kopa, kurā ietilpst gan *A*, gan *B*. Tātad: $\mu_{A \cup B}(x) = \max[\mu_A(x),\ \mu_B(x)]$, $x \in X$.

Šķēlums. Divu izplūdušo kopu *A* un *B* šķēlums ($A \cap B$) tiek definēts kā lielākā izplūdusī kopa, kura ietilpst gan kopā *A*, gan kopā *B*. Tātad: $\mu_{A \cap B}(x) = \min[\mu_A(x),\ \mu_B(x)]$, $x \in X$.

Vispārīgā gadījumā apvienojuma un šķēluma operatorus apraksta ar tā saucamo *agregācijas operatoru* palīdzību. Agregācijas operatorus var sadalīt divās klasēs:

- *t-normas*, kuras atbilst šķēluma operatoriem, un
- *s-normas* (vai *t-konormas*), kuras atbilst apvienojuma operatoriem.

Definēsim šos agregācijas operatorus.

Funkcija T:[0,1]x[0,1] → [0,1] ir *t-norma* (trijstūra norma), ja tiek apmierināti šādi nosacījumi:

1) T(x,y)=T(y,x)
2) T(T(x,y),z)=T(x,T(y,z))
3) $x \le z \wedge y \le v \rightarrow T(x,y) \le T(z,v)$
4) T(x,1)=x

Funkcija S:[0,1]x[0,1] → [0,1] ir *t-konorma* (s-norma), ja tiek apmierināti šādi nosacījumi:

1) - 3) No t-normas definīcijas

4) S(x,0)=x

Augstāk aprakstītas šķēluma un apvienojuma operācijas ir nodefinētas gadījumam, kad izplūdušās kopas ir definētas uz vienas un tās pašas kopas. Ar *cilindriskā paplašinājuma* palīdzību var vispārināt šīs operācijas gadījumam, kad izplūdušās kopas ir definētas uz dažādām kopām.

Pieņemsim, ka *U* un *V* ir divas kopas, *A* ir kopā *U* definēta izplūdusī apakškopa, tad kopas *A* cilindriskais paplašinājums telpā $U \times V$ tiek apzīmēts ar $\overline{A}$ un ir definēts šādi:

$$\overline{A}(u,v) = A(u).$$

Apskatīsim vienu piemēru. Pieņemsim, ka *U* = {1,2,3} un *V* = {a,b} un *A* ir izplūdusī kopa, kas definēta kopā *U*, tāda, ka

$$A = \left\{ \frac{0.5}{1}, \frac{0.9}{2}, \frac{1}{3} \right\},$$

kur skaitītājā ir piederības funkcijas vērtība un saucējā ir kopas *U* elementi. Tad kopas *A* cilindriskais paplašinājums telpā $U \times V$ ir

$$\overline{A} = \left\{ \frac{0.5}{(1,a)}, \frac{0.5}{(1,b)}, \frac{0.9}{(2,a)}, \frac{0.9}{(2,b)}, \frac{1}{(3,a)}, \frac{1}{(3,b)} \right\}.$$

Tātad, mēs varām dot definīcijas dažādās kopās definētajām izplūdušo kopu apvienojuma un šķēluma operācijām. Pieņemsim, ka *U* un *V* ir divas kopas, *A* un *B* ir kopu *U* un *V* izplūdušās apakškopas. Kopu *A* un *B* šķēlumu un apvienojumu var nodefinēt šādi.

***Šķēlums*.** $C = A \cap B$, kur C ir $U \times V$ tāda izplūdušā apakškopa, ka $C(u,v) = A(u) \wedge B(v)$. Vai, izmantojot cilindriskā paplašinājuma definīciju, to var pārrakstīt sekojošajā veidā: $C = \overline{A} \cap \overline{B}$, kur $\overline{A}$ un $\overline{B}$ ir *A* un *B* cilindriskie paplašinājumi telpā $U \times V$.

***Apvienojums*.** $D = A \cup B$, kur D ir $U \times V$ tāda izplūdusī apakškopa, ka $D(u,v) = A(u) \vee B(v)$. Vai, izmantojot cilindriskā paplašinājuma definīciju, to var pārrakstīt sekojošajā veidā: $D = \overline{A} \cup \overline{B}$, kur $\overline{A}$ un $\overline{B}$ ir *A* un *B* cilindriskie paplašinājumi telpā $U \times V$.

Cilindriskajam paplašinājumam pretēja operācija ir izplūdušās apakškopas *projekcija*. Nodefinēsim arī to.

Pieņemsim, ka G ir $U \times V$ izplūdušā apakškopa, G projekcija uz U tiek apzīmēta kā $\underset{\text{on } U}{\text{Proj}}\ G$ un ir definēta šādi:

$$\underset{\text{on } U}{\text{Proj}}\ G(u) = \sup_{v \in V} G(u, v).$$

Apskatīsim vienu piemēru. Pieņemsim, ka U = {1,2,3}, V = {a,b} un

$$G = \left\{ \frac{0.7}{(1,a)}, \frac{0.5}{(1,b)}, \frac{0.6}{(2,a)}, \frac{0.8}{(2,b)}, \frac{0.9}{(3,a)}, \frac{0.1}{(3,b)} \right\}.$$

Šajā gadījumā:

$$F = \underset{\text{on } U}{\text{Proj}}\ G = \left\{ \frac{0.7}{1}, \frac{0.8}{2}, \frac{0.9}{3} \right\}.$$

Izplūdusī loģika

Tagad apskatīsim dažus ar izplūdušo loģiku saistītus jautājumus. Rakstā [Zadeh, 1988] ir dota sekojošā īsa un neformāla izplūdušās loģikas definīcija: "Izplūdusi loģika – tā ir loģika, kura balstās uz tuvinātām, nevis precīzām spriešanas metodēm."

Izplūdušās loģikas pamatideja ir tāda, ka dabiskās vai mākslīgās valodas izteiksmi p var aprakstīt kā kopu $C_1, \ldots, C_k$ elastīgu ierobežojumu, kas ierobežo mainīgo kopas $X = \{X_1, \ldots, X_n\}$ vērtības. Izplūdušajā loģikā izteiksmes p kanoniskā forma ir šāda:

$$p \rightarrow X \; is \; A, \qquad (1.1)$$

kur A ir izplūdušais predikāts jeb n-vietīga izplūdusī attieksme kopā U, kur $U = U_1 \times \ldots \times U_n$, bet U_i, $i=1, \ldots, n$ ir mainīgā X_i apgabals.

Izteiksme p formā (1.1) paredz, ka kopas X iespējamības sadalījums ir vienāds ar A:

$$\Pi_X = A. \qquad (1.2)$$

Izteiksmi (1.2) var pierakstīt arī šādi:

$$Poss\{X = u\} = \mu_A(u),\ u \in U,$$

kur μ_A ir kopas A piederības funkcija, $Poss\{X=u\}$ ir iespēja, ka X var pieņemt vērtību u. Tādējādi, ja p tiek izteikta formā (1.2), tas nozīmē, ka p inducē iespējamību sadalījumu Π_X vienādu ar A, kur A ir izteiksmes p mainīgā elastīgais ierobežojums. Proti, mainīgā X iespējamību sadalījums Π_X ir mainīgā X iespējamo vērtību kopa. Tādējādi, izteiksme p ierobežo vērtības, kuras var pieņemt X, un nosaka iespējamības sadalījumu.

Ievedīsim lingvistiskā mainīga jēdzienu. *Lingvistiskais mainīgais* ir izplūdušās loģikas pamatjēdziens. Lingvistiskais mainīgais ir mainīgais, kura vērtības ir ar izplūdušo apakškopu palīdzību definēti dabiskās vai mākslīgas valodas vārdi un teikumi. Piemēram, "Vecums" ir lingvistiskais mainīgais, ja tā iespējamas vērtības ir “jauns”, “pusaudzis”, “ļoti vecs” utt. Turklāt, lingvistiskā mainīgā vērtības var uzskatīt par iespējamības sadalījumu.

1.3.2. Uz adaptīvā tīkla balstītā izplūdušā izveduma sistēma

Par vienu no automātiskās izplūdušās lēmumu pieņemšanas sistēmu var uzskatīt uz adaptīvā tīklā balstīto izplūdušā izveduma sistēmu *ANFIS (Adaptive-Network-Based Fuzzy Inference System)*. Tā ir izplūdusi vadības sistēma, proti, tā uz izplūdušo likumu pamata pieņem lēmumus par kādas sistēmas vadību. Vadības signāla noteikšanai sistēma veic loģisko izvedumu. Vienā no šīs sistēmas svarīgākajām īpašībām ir tas, ka to var apmācīt, proti, ir iespējams veikt parametru automātisko regulēšanu, kā mākslīgajos neironu tīklos.

Uz adaptīvā tīkla balstītā izplūdusī izveduma sistēma *ANFIS (Adaptive-Network-Based Fuzzy Inference System)* ir piedāvāta rakstā [Jang, 1993]. Šajā tīklā ir realizēta izplūdusī izveduma procedūra. Konspektīvi izplūdušā izveduma problēmu var noformulēt šādi: Ir dots izplūdušo likumu kopums, kur likumu kreisajā pusē ir ieejas parametri un labajā pusē ir izejas parametri. Sistēmas ieejā tiek padoti dati un sistēmai, pamatojoties uz tajā esošajiem likumiem, ir jāaprēķina izejas parametru vērtības.

Pateicoties tam, ka tīkls ir adaptīvs, ir iespējams veikt tā parametru regulēšanu. Proti, var veikt piederības funkciju parametru regulēšanu. Tīkla apmācīšana notiek līdzīgi parasto neironu tīklu apmācīšanai: tiek izveidota apmācoša kopa, kur katram ieejas vektoram tiek piekārtota vēlama izejas vērtība. Pēc tam notiek apmācīšana, un saskaņā ar apmācīšanas algoritmu notiek sistēmas parametru regulēšana līdz tam brīdim, kamēr kļūda tīkla izejā nebūs mazākā par uzdoto vērtību (maksimālā pieļaujama kļūda).

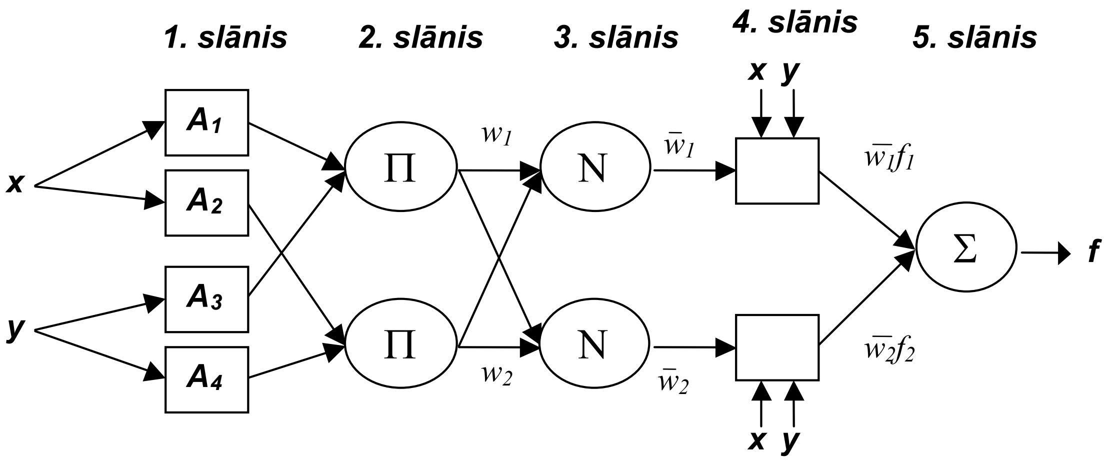
Vispirms apskatīsim sistēmas ANFIS arhitektūru. Apskatīsim izplūdušo sistēmu, kura sastāv no diviem Sudženo tipa likumiem:

$\Re_1$: if x is A_1 and y is A_3 then $z = p_1x + q_1y + r_1$

$\Re_2$: if x is A_2 and y is A_4 then $z = p_2x + q_2y + r_2$

Šai likumu bāzei atbilstoša sistēmas ANFIS arhitektūra ir parādīta 1.3. attēlā.

Taisnstūra elementu izeja ir atkarīga no parametriem, kurus var regulēt, bet ovāla elementu izeja nav atkarīga no parametriem. Savukārt, x un y ir ieejas vērtības.



1.3. att. Uz Sudženo tipa likumiem bāzēta sistēma ANFIS

ANFIS adaptīva tīkla elementu izeja katrā no slāņiem tiek definēta sekojošajā veidā.

Pirmais slānis. Šī slāņa i-tā elementa izejas vērtība ir

$$o_i^1 = \mu_{A_i}(x),$$

kur x ir ieejas vērtība i-tajā elementā un A_i ir ar šī elementa saistītā lingvistiskā vērtība.

Otrais slānis. Šī slāņa elementi sareizina ieejā saņemtas vērtības un sūta tālāk iegūto reizinājuma rezultātu. Katra elementa izeja raksturo atbilstošā likuma *aktivizēšanas spēku* w_i.

Trešais slānis. Šī slāņa izeja ir aktivizēšanas spēku normētās vērtības $\overline{w}_i$.

Ceturtais slānis. Šī slāņa i-tā elementa izejas vērtība ir aprēķināta pēc formulas (1.3):

$$o_i^4 = \overline{w}_i f_i = \overline{w}_i \left(p_i x + q_i y + r_i\right), \tag{1.3}$$

kur $\{p_i, q_i, r_i\}$ ir seku daļas parametru kopa.

Piektais slānis. Šajā slānī ir viens elements, kas sasummē visus ienākošos signālus, līdz ar ko tiek izrēķināta kopējā sistēmas izeja:

$$o^5 = \sum_i \overline{w}_i f_i \,.$$

Sistēmas ANFIS apmācīšanai paredzēts algoritms ir balstīts uz gradienta lejupslīdes metodes. Proti, katrā iterācijā svara izmaiņa tiek noteikta saskaņā ar kļūdas funkcijas parciālo atvasinājumu pēc sistēmas parametriem, kuri ir jāregulē.

Kļūdas funkcija ir definēta šādi:

$$E = \sum_{m=1}^{\#(L)} \left(T_m - O_m^L\right)^2,$$

kur L ir slāņu skaits, *#(L)* ir elementu skaits izejas slānī, T_m ir vēlamo vērtību vektora m-tais elements un O_m^L ir sistēmas izejas vektora m-tais elements.

Tagad apskatīsim kādā veidā mēs varam aprēķināt kļūdas funkcijas parciālo atvasinājumu pēc sistēmas parametriem.

Vispirms jāaprēķina kļūdas funkcijas atvasinājums pēc katra elementa izejas funkcijas $\partial E / \partial O$. Izejas slāņa elementiem tas izskatās šādi (1.4):

$$\frac{\partial E}{\partial O^L} = -2\left(T - O^L\right). \tag{1.4}$$

Iekšējā slāņa elementiem parciālo atvasinājumu var noteikt izmantojot *ķēdes likumu*:

$$\frac{\partial E}{\partial O_i^L} = \sum_{m=1}^{\#(k+1)} \frac{\partial E}{\partial O_m^{k+1}} \frac{\partial O_m^{k+1}}{\partial O_i^k}, \quad \text{kur } 1 \le k \le L-1.$$

Ja α ir dotās sistēmas parametrs, tad sistēmas kļūdas parciālo atvasinājumu pēc šī parametra var noteikt ar formulas (1.5) palīdzību:

$$\frac{\partial E}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E}{\partial O^*} \frac{\partial O^*}{\partial \alpha}, \tag{1.5}$$

kur kopa O^* satur tos elementus, kuru izeja ir atkarīga no parametra α.

Tādejādi, parametra α vērtības izmaiņu definē formula (1.6):

$$\Delta\alpha = -\eta \frac{\partial E}{\partial \alpha}, \tag{1.6}$$

kur η ir apmācīšanas ātrums.

1.3.3. Izplūdušo granulu izmantošana lēmumu pieņemšanā

Lēmumu pieņemšanas sistēmas, kuras apstrādā izplūdušās liecības parādījās septiņdesmito gadu beigās [Zadeh, 1979]. Šīs sistēmas nosaka, ar kādu varbūtību kritērija vērtība, kura tiek uzdota šajā alternatīvā, būs vienāda ar vēlamo vērtību (šai varbūtībai ir intervālu vērtība). Nākamajā paragrāfā tiek dziļāk apskatīts izplūdušo granulu jēdziens.

1.3.3.1. Izplūdušās informācijas granulas

Uz izplūdušām granulām bāzētas liecības tiek piedāvātas rakstā [Zadeh, 1979]. Izplūdušās liecības var uzskatīt par Dempstera-Šefera liecību teorijas vispārinājumu izplūdušajam gadījumam [Shafer, 1976; Yager, 1981]. Dempstera-Šefera teorijas ietvaros tiek apskatītas nejaušas kopas, turklāt, definējot zināšanas, var nomodelēt arī "nezināšanu". Ar to Dempstera-Šefera teorija atšķiras no varbūtību teorijas. Izplūdušās liecību teorijas galvenā īpašība ir tā, ka tajā tiek apvienota izplūdusī loģika un varbūtības teorija. Proti, šīs teorijas liecības var uzskatīt par izplūdušo apakškopu varbūtību sadalījumu.

Izplūdušās liecības sastāv no informācijas granulām, katra no kurām tiek raksturota ar nosacīta *JA-TAD* likuma palīdzību. Turklāt likumu kreisā daļa tiek uzdota ar *varbūtības sadalījuma* palīdzību, bet likumi tiek uzskatīti par *nosacītās iespējamības sadalījumu*. Līdz ar to izplūdušās liecības var uzskatīt par izplūdušo vērtību varbūtību sadalījumu.

Vispirms apskatīsim informācijas granularitātes jēdzienu. Reālajā pasaulē bieži gadās, ka mēraparatūras neprecizitātes un diskrētības dēļ nav iespējams saņemt absolūti precīzu informāciju. Šajā gadījumā var uzskatīt, ka šāda informācija ir *granulāra* tādā nozīmē, ka tā tiek reprezentēta ar nedalāmas granulas palīdzību, kura var aprakstīt vērtību apgabalu. Formāla granulas definīcija tiks dota vēlāk.

Rakstā [Zadeh, 1979] tiek apskatītas divu tipu granulas. Pirmā tipa granulas izskatās šādi:

$$\mathrm{g} = X \text{ is } G \text{ is } \lambda\,,$$

kur X ir mainīgais, kas pieņem vērtībās kopā U, G ir kopas U izplūdusī apakškopa, un λ ir izplūdusī varbūtība. Zemāk ir parādīti šādu granulu piemēri:

$$\mathrm{g} = X \text{ is } \mathit{SMALL} \text{ is } \mathit{LIKELY},$$

$$\mathrm{g} = X \text{ is } \mathit{MUCH\ LARGER\ THAN}\ \ Y \text{ is } \mathit{VERY\ UNLIKELY}.$$

Otrā tipa granulas ir pašā sākumā minētās nosacītās granulas, proti, tās ir granulas, kuras ir uzdotas ar *JA-TAD* likumu palīdzību. Šādas granulas izskatās šādi:

$$\mathrm{g} \;=\; \text{If } X = u \text{ then } Y \text{ is } G, \tag{1.7}$$

kur X pieņem vērtības saskaņā ar uzdotu varbūtības sadalījumu.

Granulārās informācijas jēdziens ir cieši saistīts ar *iespējamības sadalījuma* jēdzienu. Tātad, apskatīsim dažus jēdzienus no iespējamību teorijas, kurus mēs izmantosim, apskatot izplūdušās liecības.

Pieņemsim, ka X ir mainīgais, kas pieņem vērtības kopā U. Var dot sekojošo definīciju iespējamības sadalījumam: mainīgā X iespējamības sadalījums Π_X ir elastīgs ierobežojums vērtībām, kuras var pieņemt X. Proti, ja π_X ir Π_X piederības funkcija, tad:

$$\text{Poss}\{X{=}u\} = \pi_X(u),$$

kur kreisā vienādojuma daļa ir iespējamība, ar kuru mainīgais X var pieņemt vērtību u, un $\pi_X(u)$ ir Π_X piederības funkcijas vērtība punktā u. Tātad, iespējamības sadalījumu Π_X raksturo *iespējamības sadalījuma funkcija* $\pi_X : U \to [0,1]$.

Apskatīsim vienu piemēru. Iespējamības sadalījums var tikt uzdots saskaņā ar fizikālo ierobežojumu. Pieņemsim, ka mūsu gadījumā fizikālais ierobežojums ir tenisa bumbiņu skaits, kuru var ielikt metāla kastītē. Šajā gadījumā iespējamības sadalījums ir X, kas tiek uzdots ar piederības funkcijas palīdzību $\pi_X(u)$, kura nosaka, cik "viegli" (fizikālajā nozīmē) ir ielikt u bumbiņas metāla kastītē.

Vispārīgā gadījumā, ja mums ir spriedums formā (1.8):

$$\text{p} = N \text{ is } F, \tag{1.8}$$

kur F ir Dekarta reizinājumā $U = U_1 \times ... \times U_n$ definēta izplūdušā apakškopa un N ir mainīgais, tad ir spēkā sekojošais *iespējamības piešķiršanas vienādojums*:

$$N \textit{ is } F \to \Pi_{(X_1, \ldots, X_n)} = F,$$

kur simbols $\to$ apzīmē "pārveidojas par".

$$\text{X ir mazs} \to \Pi_X = \text{MAZS},$$

kur MAZS tiek uzdots ar kopas $\{x \mid x \in [0, \infty)\}$ izplūdušo apakškopu. Tātad, ja MAZS tiek aprakstīts ar piederības funkcijas μ_{MAZS} palīdzību, tad:

$$\text{Poss}\{X = u\} = \mu_{MAZS}(u), \qquad u \in [0, \infty)$$

Ja mums ir uzdots nosacītais spriedums formā p = If X is F then Y is G, tad tas inducē *nosacīto iespējamības sadalījumu* (1.9):

$$\text{If } X \text{ is } F \text{ then } Y \text{ is } G \to \Pi_{(Y|X)} = \overline{F'} \cup \overline{G}, \tag{1.9}$$

kur $\Pi_{(Y|X)}$ ir Y iespējamības sadalījums pie nosacījuma X, F un G ir attiecīgi U un V izplūdušās apakškopas, $\overline{F}$ un $\overline{G}$ ir F un G cilindriskie paplašinājumi telpā $U \times V$. *Nosacītās iespējamības sadalījuma funkcija* tiek uzdota sekojošajā veidā:

$$\pi_{(Y|X)}(v \mid u) = (1 - \mu_F(u)) \vee \mu_G(v), \qquad u \in U, v \in V,$$

kur μ_F un μ_G ir F un G piederības funkcijas un $\vee = \max$.

Pieņemsim, ka iespējamības sadalījums Π_X tika iegūts no sprieduma

$$p = X \text{ is } G,$$

un pieņemsim, ka *F* ir kopas *U* izplūdušā apakškopa. Tad *nosacītās iespējamības mēru* var nodefinēt sekojošajā veidā:

$$\text{Poss}\{X \text{ is } F \mid X \text{ is } G\} = \sup_u \left(\mu_F(u) \wedge \mu_G(u)\right)$$

Tagad apskatīsim, kādā veidā granulas tiek grupētas liecībās. Pieņemsim, ka *X* un *Y* ir mainīgie, kuri pieņem vērtības attiecīgi no kopas *U* un *V*. Tad *nosacītā* π*-granula* kopā *V* tiek uzdota saskaņā ar (1.7):

$$g = \text{If } X = u \text{ then } Y \text{ is } G,$$

kur *G* ir kopas *V* izplūdusī apakškopa. Granulas *g* iespējamības sadalījums tiek aprakstīts ar sekojošās iespējamības sadalījuma funkcijas palīdzību:

$$\pi_{(Y|X)}\left(v \mid u\right) = \text{Poss}\{Y = u \mid X = u\} = \mu_G(v)\ .$$

Par *liecības ķermeni* vai vienkārši *liecību* var uzskatīt struktūru, kura sastāv no spriedumiem. *Granulētā liecība* ir tāda liecība, kura sastāv no *granulām*:

$$E = \{g_1, ..., g_N\}$$

Otrā tipa liecības ir nosacīto π - granulu varbūtības sadalījums. Pieņemsim, ka mums ir zināms *(a)* varbūtības sadalījums $P_X = \{p_1, ..., p_N\}$, kur

$$p_i = \text{Prob}\{X = i\}, \qquad i = 1, ..., N$$

un *(b)* nosacītas iespējamības sadalījums $\Pi_{(Y|X)}$, kur

$$\Pi_{(Y|X=i)} = G_i\ , \qquad i = 1, ..., N$$

Tātad, liecību ir iespējams aprakstīt sekojošajā, īsākā veidā:

$$E = \{P_X, \Pi_{(Y|X)}\}\ . \tag{1.10}$$

Kā redzams, liecības tiek aprakstītas ar varbūtības sadalījuma un nosacītā iespējamības sadalījuma palīdzību.

Izveidojot sistēmu, kura sastāv no vairākām liecībām $E = \{E_1, ..., E_K\}$, mēs varam uzdot tai dažādus jautājumus par informāciju, kura tiek attēlota liecībās. Pamatproblēma, kura tiek apskatīta granulārās informācijas ietvaros ir saistīta ar notikuma (*Y* is *Q*) varbūtības noteikšanu, kur *Q* ir izplūdusī apakškopa. Turklāt šī varbūtība ir intervālu vērtība. Sagaidāmā iespējamība EΠ(*Q*) *(expected possibility)* ir varbūtības augšējā robeža un sagaidāmā noteiktība EC(*Q*) *(expected certainty)* ir varbūtības apakšējā robeža. Rakstā [Zadeh, 1979] ir piedāvātas sekojošās formulas sagaidāmas iespējamības un noteiktības aprēķināšanai, sk. (1.11) un (1.12).

$$\mathrm{E\Pi}(Q) = \sum_{i,j} p_{ij} \sup\left(Q \cap G_i \cap H_j\right), \tag{1.11}$$

$$\mathrm{EC}(Q) = 1 - \mathrm{E\Pi}(Q'), \tag{1.12}$$

kur Q' ir kopas Q papildinājums. Formula (1.11) ir paredzēta īpašam gadījumam, kad sistēma sastāv no divām liecībām. Vispārīgajā gadījumā šī formula izskatās šādi. Pieņemsim, ka ir n liecības un katrā no tām ir $m_1, m_2, ..., m_n$ granulas, tad sagaidāmo iespējamību var izskaitļot sekojošajā veidā:

$$\mathrm{E\Pi}(Q) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} ... \sum_{i_n=1}^{m_n} p_{i_1 i_2 ... i_n} \sup\left(Q \cap G_{i_1}^1 \cap G_{i_2}^2 \cap ... \cap G_{i_n}^n\right),$$

kur $p_{i_1 i_2 ... i_n} = p_{i_1} p_{i_2} ... p_{i_n}$ un G_i^j ir j-tās liecības i-tā granula.

1.3.4. Intervālu varbūtības lēmumu pieņemšanā

Iemesls, kāpēc lēmumu pieņemšanas uzdevumos rodas intervālu vērtības un, it īpaši, intervālu varbūtības, ir saistīts ar to, ka bieži pieejama informācija nav pietiekoši precīza, lai varētu novērtēt precīzas varbūtību vērtības [Kyburg, 1999].

Turklāt, intervāli var rasties, ja lēmuma pieņemšanas procesā piedalās vairākas personas. Viens no veidiem, kā var uzdot intervālu varbūtības, ir lineārās nevienādības [Whalen, 1994]. Tādējādi precīzs varbūtību sadalījums nav zināms un atsevišķas varbūtības ir uzdotas kā intervāli, kuru robežas nosaka lineārās nevienādības. Savukārt, ja lēmējpersona saņems pietiekoši daudz informācijas, tad šie intervāli kļūs par precīzām vērtībām.

Tagad apskatīsim kādēļ promocijas darbā parādās intervālu varbūtības un kāda tām ir nozīme. Šajā nolūkā apskatīsim Dempstera-Šefera liecību teoriju [Shafer, 1976; Zadeh, 1986a], uz kuras ir balstīta šajā darbā izmantotā izplūdušo granulu teorija [Zadeh, 1979]. Kā būs redzams, promocijas darbā intervālu novērtējumi rodas izplūdušo granulu apstrādes laikā, kad tiek noteiktas izplūdušo notikumu varbūtības.

Lai izskaidrotu Dempstera-Šefera liecību teorijas ideju apskatīsim relāciju datubāzi ar 2. kārtas vaicājumiem. Proti, šo vaicājumu parametri ir 1. kārtas vaicājumi. Paskaidrosim šo pieeju. Pieņemsim, ka mums ir tabula ar darbinieku vārdiem un vecumiem (sk. 1.1. tabulu).

1.1. tabula

Darbinieku vārdi un vecumi

Vārds	Vecums
Jānis	23
Ivo	28
Māris	21
Juris	27
Mārtiņš	30

Apskatīsim vienkārša vaicājuma piemēru: kādas darbinieku daļas vecums ir no 20 līdz 25 gadiem ieskaitot? Citiem vārdiem, kāda darbinieku daļa apmierina nosacījumu $Vecums(i) \in Q$, kur i ir darbinieks un Q ir vaicājuma kopa *Q=[20; 25]*. Nav grūti aprēķināt, ka atbilde ir *2/5*.

Tagad pieņemsim, ka darbinieku vecums nav precīzi zināms. Piemēram, ir zināms, ka darbinieka Jāņa vecums ir intervālā [22; 26]. Ja darbinieku vecums nav precīzi zināms, tad izteiksmei $Vecums(i) \in Q$ nav jēgas. Savukārt šajā gadījumā dabiski būtu ievest citus šīs izteiksmes novērtējumus, proti, iespējamību un noteiktību. Definēsim tās:

1. Izteiksme $Vecums(i) \in Q$ ir iespējama, ja vecuma intervāla *Vecums*(*i*) šķēlums ar intervālu *Q* nav tukša kopa. Ar D_i apzīmēsim intervālu *Vecums*(*i*), tad $D_i \cap Q \neq \Lambda$, kur Λ ir tukša kopa.
2. Izteiksme $Vecums(i) \in Q$ ir noteikta (vai nepieciešama), ja *Q* satur intervālu D_i, proti, ja $D_i \subset Q$.
3. $Vecums(i) \in Q$ nav iespējama, ja $D_i \cap Q = \Lambda$, kur Λ ir tukša kopa.

Pieņemsim, ka darbinieku vecumi ir definēti tā, kā tas ir parādīts 1.2. tabulā.

1.2. tabula

Darbinieku vārdi un vecumi

Vārds	Vecums
Jānis	[22; 26]
Ivo	[20; 22]
Māris	[30; 35]
Juris	[20; 22]
Mārtiņš	[28; 30]

Ja mēs tagad mēģināsim noskaidrot, kādas darbinieku daļas vecums ir no 20 līdz 25 gadiem ieskaitot, tad rezultātā iegūsim divus novērtējumus – *iespējamību* un *noteiktību*. Proti, tas būs intervālu novērtējums. Iemesls, kādēļ rodas intervāls, ir tāds, ka ieejas dati nav precīzi, proti, vecumi ir uzdoti kā intervāli. Savukārt, ja mēs norādīsim precīzas vecumu vērtības, tad vaicājuma intervāla novērtējuma vietā būs precīzs novērtējums. Atbildi uz šo vaicājumu var pierakstīt šādi:

$$Atbilde(Q) = \left[N(Q); \Pi(Q) \right],$$

kur *Atbilde*(*Q*) ir atbilde uz vaicājumu *Q*, *N*(*Q*) ir *Q* noteiktība un $\Pi(Q)$ ir *Q* iespējamība. Augstāk aprakstītajam piemēram šie novērtējumi ir sekojoši:

$$Atbilde([20;25]) = \left[N([20;25]) = 2/5; \Pi([20;25]) = 3/5 \right] = \left[\frac{2}{5}; \frac{3}{5} \right]. \quad (1.13)$$

Kā redzams, šajā gadījumā intervālu robežas norāda, kāds ir vismazākais un vislielākais iespējamais novērtējums, ja ieejas informācija nav precīza.

Apskatīsim šī uzdevuma vispārīgo gadījumu. Pieņemsim, ka 1.2. tabulas intervālu vērtību D_i konstruēšanai tiek izmantotas kopas $A_1, \ldots, A_k$, proti:

$$\begin{aligned} D_1 &= [22, 26] \\ D_2 &= [20, 22] \\ D_3 &= [30, 35] \\ D_4 &= [20, 22] \\ D_5 &= [28, 30] \end{aligned} \qquad \begin{aligned} A_1 &= [22, 26] \\ A_2 &= [20, 22] \\ A_3 &= [30, 35] \\ A_4 &= [28, 30] \end{aligned}$$

Tādējādi, uzdevumu var pārrakstīt granulu formā:

$$\Delta = \{(A_1, p_1), (A_2, p_2), \ldots, (A_k, p_k)\},$$

kur p_i apzīmē kopu D_j skaitu, kuras ir vienādas ar A_i.

Apskatītā piemēra granulārais pieraksts izskatās šādi:

$$\Delta = \{([22, 26], 1/5),\ ([20, 22], 2/5),\ ([30, 35], 1/5),\ ([28, 30], 1/5)\}.$$

Novērtējumu *N*(*Q*) un $\Pi(Q)$ definīciju var pārrakstīt šādā veidā:

$$N(Q) = \sum_s p_s, A_s \subset Q, s = 1, \ldots, k, \quad (1.14)$$

$$\Pi(Q) = \sum_s p_s, A_s \cap Q \neq \Lambda, s = 1, \ldots, k. \quad (1.15)$$

Izmantojot formulas (1.14) un (1.15), mēs iegūstam to pašu rezultātu, proti, (1.13).

1.4. Informācijas daudzums nenoteiktajos apstākļos un entropija

Termins entropija ir nācis no fizikas, kurā tas tiek definēts kā kārtas trūkuma vai nejaušības pakāpe [Pickett, 2000]. Savukārt, datorzinātnē ar entropiju parasti apzīmē sistēmas nenoteiktības pakāpi. Ir vairākas entropijas definīcijas, kuras tiek dotas dažādu teoriju ietvaros. Droši vien visizplātītākā ir Šenona definīcija, kuru viņš sniedz informācijas teorijas kontekstā [Shannon&Weaver, 1949]. Tomēr, ir arī citas entropijas definīcijas. Piemēram, A. Kolmogorova sarežģītība, kura tiek definēta ar algoritma garuma palīdzību [Kolmogorov, 1965], vai tai līdzīga Čeitina entropijas definīcija, kuru viņš sniedz algoritmiskās informācijas teorijas kontekstā [Chaitin, 2003]. Turklāt, Barts Kosko grāmatā [Kosko, 1992] definē izplūdušo kopu entropiju, kura parāda cik "izplūdusi" ir kopa. Savukārt, lēmumu pieņemšanas jomā entropija tiek lietota kritēriju svaru noteikšanai [Zavadskas u.c., 1994]. Saskaņā ar šo pieeju, jo lielāka ir starpība starp kritēriju vērtībām, jo lielāka uzmanība tiek pievērsta šādiem kritērijiem un to svars ir lielāks.

Termins entropija tiek izmantots arī filozofijā, lai aprakstītu evolūcijas procesu [Geraščenko, 2002]. Izstrādātajā evolūcijas modelī tiek apvienota entropija, informācija un enerģija.

Vairāki autori, piemēram, Čeitins un Kosko [Chaitin, 2003; Kosko, 1992], apspriežot entropiju apskata arī ar nejaušību un nepilnīgumu saistīti jautājumi. Arī Kolmogorovs piemin nejaušību [Kolmogorov, 1965]. Apspriežot nejaušību, jautājums bieži tiek reducēts uz "Vai Dievs spēlē kauliņus?" Acīmredzami, tādu "redukciju" ietekmēja Einšteina frāze "*I am convinced that He [God] does not play dice.*" ("Esmu pārliecināts, ka Dievs kauliņus nespēlē") [Einstein, 1926]. Dažādu rakstu autoriem ir dažādi uzskati par šo tematu. Sakarā ar to, var pieminēt rakstu [Hawking, 2000], kurā tiek apskatīti ar nejaušību saistīti jautājumi un kurā netieši tiek apskatīta entropija, lai gan šīs vārds rakstā netiek pieminēts. Šajā rakstā nejaušība tiek apskatīta no kvantu mehānikas un melno caurumu viedokļa.

1.4.1. Informācijas teorija un Šenona entropija

Vispirms apskatīsim entropijas definīciju, kura ir sniegta informācijas teorijas kontekstā [Shannon&Weaver, 1949]. Galvenie jautājumi, kuri tiek apskatīti šīs teorijas ietvaros, ir saistīti ar informācijas sūtīšanu. Piemēram, cik lielai jābūt datu kanāla caurlaidspējai, lai varētu veiksmīgi un bez aizkavēšanas aizsūtīt kādu ziņojumu.

Informācijas teorijas kontekstā sistēmas entropija tiek definēta šādā veidā. Pieņemsim, ka ir dota sistēma *S*, kurai jāizskaitļo entropija, un ir zināms, ka šī sistēma var atrasties *n* dažādos stāvokļos: $q_1, q_2, q_3, \ldots, q_n$.

Ar formulas (1.16) palīdzību tiek apzīmēts, ka sistēma *S* atrodas stāvoklī q_i ar varbūtību p_i:

$$P(S \sim q_i) = p_i, \quad i = 1, \ldots, n. \tag{1.16}$$

Ja ir zināmas varbūtības (1.16) visiem sistēmas stāvokļiem, tad šīs sistēmas entropija var tikt izskaitļota saskaņā ar formulu (1.17):

$$H = -\sum_{i=1}^{n} p_i \log p_i \ . \tag{1.17}$$

Turklāt, ir acīmredzams, ka varbūtību p_i summai jābūt vienādai ar vieninieku:

$$\sum_{i=1}^{n} p_i = 1 \, .$$

Ziņojuma *informatīvums* tiek definēts kā sistēmas entropijas izmaiņa pēc ziņojuma saņemšanas. Proti, informatīvums ir vienāds ar lielumu, par kuru ir samazinājusies entropija pēc ziņojuma saņemšanas. Ja ziņojums sniedz pilnīgu informāciju par sistēmu, tad tā informatīvums ir vienāds ar entropijas vērtību (jo sistēmas entropija kļūst vienāda ar nulli). Ja ziņojums nesniedz nekādu informāciju par sistēmu, tad tā informatīvums ir vienāds ar nulli, jo entropija nesamazinās.

Ar formulas (1.17) palīdzību uzdota entropija ir *aditīva*. Pierādījumu var atrast grāmatā [Ventcel, 2001].

Sīkāku informācijas teorijas aprakstu var atrast grāmatā [Ventcel, 2001] un rakstā [Shannon&Weaver, 1949].

1.4.2. Kolmogorova pieeja informācijas mērīšanai

Rakstā [Kolmogorov, 1965] tiek apskatas trīs pieejas informācijas mērīšanai. Pirmās divas pieejas, proti, kombinatoriska un varbūtiska pieeja, ir līdzīgas 1.4.1. paragrāfā aprakstītajai entropijai. Tomēr, trešā pieeja no tās atšķiras, jo A. Kolmogorova pieejā objekta sarežģītība tiek definēta ar programmas garuma palīdzību. A. Kolmogorovs to sauc par "algoritmisku pieeju informācijas mērīšanai".

Kolmogorovs to definē sekojošajā veidā. Ja ir zināms objekts *x*, tad par objekta *y* nosacītu sarežģītību tiek uzskatīts tādas visīsākās programmas *p* garums *l(p)*, ka ar programmas *p* palīdzību no *y* var iegūt *x*. Kā ir redzams, šāds novērtējums ir atkarīgs no programmēšanas valodas, kuru mēs izmantojam programmas *p* rakstīšanai. Programmēšanas valodu var uzskatīt par daļēji rekursīvu funkciju $\varphi(p,x)=y$, kura nosaka objektu *y*, izmantojot programmu *p* un objektu *x*. Tātad, iepriekš aprakstīto sarežģītības novērtējumu var uzdot ar formulas (1.18) palīdzību:

$$K_{\varphi}(y\mid x)=\begin{cases}\min\limits_{\varphi(p,x)=y} l(p),\\ \infty,\text{ ja nav tāda p, ka } \varphi(p,x)=y\end{cases}. \tag{1.18}$$

Rakstā [Kolmogorov, 1965] tiek parādīts, kā var atrast *asimptotiski optimālu* funkciju *A(p, x)*, kura apmierina nevienādību (1.19).

$$K_A(y\mid x)\le K_{\varphi}(y\mid x)+C_{\varphi}. \tag{1.19}$$

Objekta sarežģītība tiek definēta ar formulas (1.20) palīdzību. Citiem vārdiem, formula (1.20) parāda cik liela programma jāuzraksta, lai izrēķinātu objektu *y*.

$$K_A(y)=K_A(y\mid 1). \tag{1.20}$$

Informācijas apjoms par objektu *y*, kas atrodas objektā *x*, tiek definēts ar formulas (1.21) palīdzību:

$$I_A(x:y)=K_A(y)-K_A(y\mid x). \tag{1.21}$$

Arī formulai (1.21) var dot intuitīvu skaidrojumu. Tā parāda cik efektīvāk (proti, cik daudz var samazināt programmas garumu) var izskaitļot objektu *y*, ja ir dots objekts *x*. Jo vairāk kods tiek samazināts, jo vairāk objektā *x* ir informācijas par objektu *y*.

1.4.3. Sarežģītības jēdziens algoritmiskajā informācijas teorijā

Čeitins algoritmiskās informācijas teorijas ietvaros sniedz līdzīgu definīciju sarežģītības novērtējumam, kas arī ir balstīts uz minimālas programmas izmēra [Chaitin, 2003].

Vispirms apskatīsim kas ir pašnorobežojošs dators (*self-delimiting computer – angļu val.*). Par pašnorobežojošu datoru $C(p)\rightarrow output$ sauc tādu datoru, kuram ieejā padod programmu *p* un izejā saņem rezultātu *output*. Pie kā, nekādas pareizas programmas paplašinājums nav pareiza programma (ar "pareizu" tiek saprasta sintaktiski pareiza programma). Tātad, ja funkcija *C(p)* ir definēta, tad *C* nav definēta nevienai programmai, kurai *p* ir sākumdaļa. Citiem vārdiem, funkcijas *C* definēšanas apgabals ir bezprefiksa kopa (*prefix-free set – angļu val.*).

Ja šāds dators *C* ir dots, tad Čeitina sarežģītības novērtējums $H_C(x)$ ir tādas datora *C* vismazākās programmas izmērs bitos, kura izejā izdod *x*:

$$H_C(x) \equiv \min_{C(p)=x} |p| .$$

Čeitins šādas minimālas programmas sauc par *elegantām programmām*. Līdzīgi tam, kā iepriekšējā solī tika konstruēta asimptotiski optimāla funkcija *A*, šajā teorijā tiek konstruēts universāls dators *U*, kas apmierina sekojošu nevienādību:

$$H_U \le H_C + \text{constant} .$$

Objekta *x informatīvums* ir vienāds ar elegantas programmas garumu, kas izskaitļo objektu *x*. Tas tiek definēts ar formulas (1.22) palīdzību.

$$H(x) \equiv \min_{U(p)=x} |p| . \tag{1.22}$$

Divu objektu *x* un *y kopēja sarežģītība* ir vienāda ar elegantas programmas garumu, kura izskaitļo abus objektus:

$$H(x, y) \equiv \min_{U(p)=\langle x, y \rangle} |p| .$$

Algoritmiskās informācijas teorijas ietvaros *algoritmiskā neatkarība* tiek definēta šādā veidā:

$$H(x, y) \approx H(x) + H(y) . \tag{1.23}$$

Citiem vārdiem, divi objekti ir algoritmiski neatkarīgi, ja programmas garums, kura rēķina abus objektus ir apmēram vienāds ar individuālo programmu garumu summu. Tas nozīmē, ka šiem objektiem nav nekā kopīga.

Kopēja informācija diviem objektiem tiek definēta sekojošajā veidā:

$$H(x : y) \equiv H(x) + H(y) - H(x, y) .$$

Grāmatā [Chaitin, 2003] var atrast virkni citu definīciju un detalizētāku algoritmiskas informācijas teorijas aprakstu.

Algoritmiski nejauša virkne šajā teorijā tiek definēta ar formulas (1.24) palīdzību. Citiem vārdiem, mums ir jāzina virknes garums un pati virkne.

$$H(\text{ļoti sarežģīta N - bitu virkne}) \approx N + \log_2 N . \tag{1.24}$$

Ir vērts pieminēt arī *algoritmisku varbūtību P(x)*, kura tiek definēta ar formulas (1.25) palīdzību:

$$P(x) \equiv \sum_{U(p)=x} 2^{-|p|}. \tag{1.25}$$

Tā kā funkcijas U definīcijas apgabals ir bez-prefiksa kopa, izteiksmes (1.25) vērtība var būt no nulles līdz vieniniekam. Pie kā, izteiksmē (1.25) tiek ņemtas vērā visas programmas, ne tikai vismazākās, jeb elegantas programmas.

Var parādīt, ka sarežģītības novērtējumu var nodefinēt sekojošajā veidā:

$$H(x) = -\log_2 P(x) + O(1). \tag{1.26}$$

Formula (1.26) parāda, ka ir zināma līdzība starp algoritmisku informācijas teoriju un Šenona informācijas teoriju.

1.4.4. Izplūdušo kopu entropija

Grāmatā [Kosko, 1992] tiek piedāvāta izplūdušu kopu grafiskā interpretācija. Izplūdušo kopu entropija tiek definēta izmantojot šo grafisko interpretāciju, kas padara to intuitīvi saprotamu un pieņemamu.

Vispirms apskatīsim kādā veidā izplūdušas kopas var attēlot grafiski. Pie izplūdušo kopu grafiskas interpretācijas noved jautājums par to, kā izskatās visu kopas X izplūdušo apakškopu kopa $F(2^X)$? Šī kopa izskatās kā *kubs*. Un kas ir izplūdusī apakškopa? Tā ir šī kuba punkts. Visu izplūdušo apakškopu kopa ir vienības hiperkubs $I^n = [0,1]^n$, kur n ir elementu skaits kopā X. Izplūdusī apakškopa ir jebkurš punkts hiperkubā I^n. Tātad, ar (X, I^n) tiek definēta galīgas izplūdušas teorijas metriska telpa.

Ja ir dota kopa, kura sastāv no diviem elementiem $X=\{x_1, x_2\}$, tad šīs kopas visu apakškopu kopa izskatās sekojošajā veidā: $2^X = \{\Lambda, X, \{x_1\}, \{x_2\}\}$. Katra no apakškopām atbilst bitu vektoram (0 0), (1 1), (1 0) un (0 1), atkarībā no tā, kurš elements kādai apakškopai pieder.

Piemēram, apskatīsim kopas X izplūdušo apakškopu: $A = \left(\frac{1}{3}\ \frac{3}{4}\right)$. Šajā gadījumā, elementa x_1 piederības pakāpe kopai A ir $\frac{1}{3}$ un elementa x_2 piederības pakāpe kopai A ir $\frac{3}{4}$. Šajā gadījumā kopai 2^X atbilst hiperkubs I^2 un kopa A var tikt attēlota kā punkts šajā hiperkubā. Tas ir parādīts 1.4. attēlā.

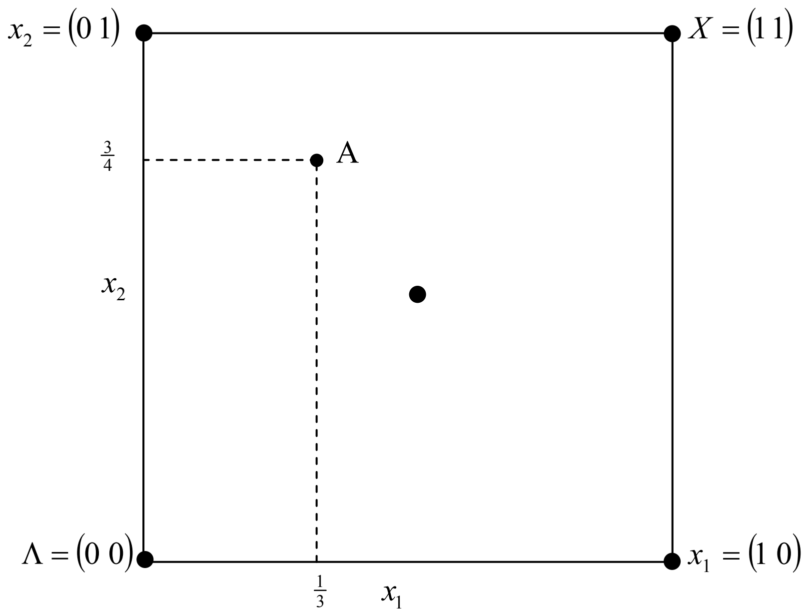
1.4. attēlā parādītas hiperkuba virsotnes ir precīzas, neizplūdušas kopas. Savukārt, visi pārējie punkti atbilst izplūdušām kopām. Hiperkuba centra punkts ir maksimāli izplūdis, šajā punktā visu piederības funkciju vērtības ir vienādas ar $\frac{1}{2}$. Ja vidējam punktam atbilstošo kopu apzīmē ar M un tās papildinājumu ar M^c, tad ir spēkā (1.27):

$$M = M \cap M^c = M \cup M^c = M^c . \tag{1.27}$$

Izplūdušo kopu apvienošana, šķelšana un papildināšana tiek definēta šādi:

$$\mu_{A\cap B} = \min(\mu_A, \mu_B), \quad \mu_{A\cup B} = \max(\mu_A, \mu_B), \quad \mu_{A^c} = 1 - \mu_A .$$

Papildināsim 1.4. attēlu ar kopām $A \cap A^c$, $A \cup A^c$ un A^c atbilstošiem punktiem. Rezultāts ir redzams 1.5. attēlā.



1.4. att. Izplūdušo kopu attēlošana ar punktu palīdzību

Cik liela ir izplūdusī kopa? Izplūdušo kopas kardinalitāti rēķina šādā veidā:

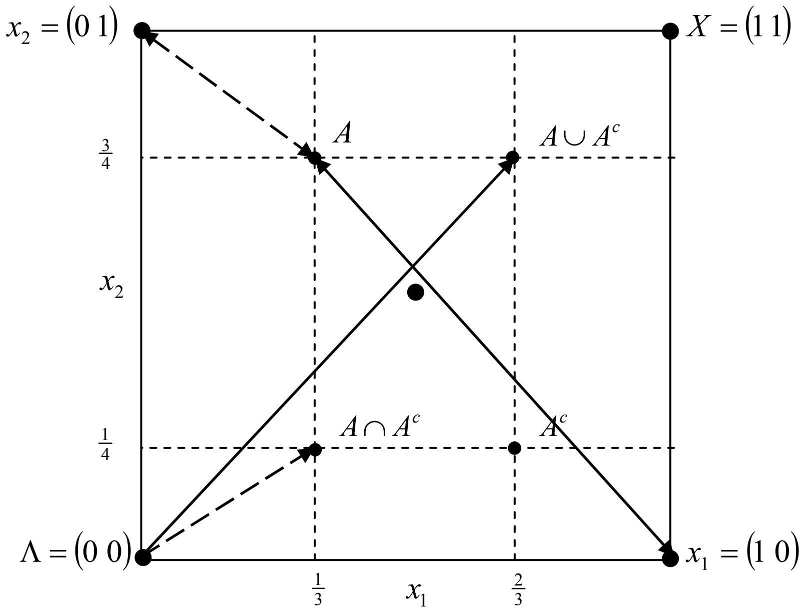
$$M(A) = \sum_{i=1}^{n} \mu_A(x_i) .$$

Ievedīsim šajā telpā Heminga metriku. Tā diviem punktiem tiek definēta šādā veidā:

$$l^1(A,B) = \sum_{i=1}^{n} |\mu_A(x_i) - \mu_B(x_i)| .$$

Ir viegli pamanīt saistību starp l^1 un $M(A)$:

$$M(A) = \sum_{i=1}^{n} \mu_A(x_i) = \sum_{i=1}^{n} |\mu_A(x_i) - 0| = \sum_{i=1}^{n} |\mu_A(x_i) - \mu_\Lambda(x_i)| = l^1(A, \Lambda).$$



1.5. att. Hiperkuba aizpildīšana un entropijas rēķināšana

Tagad apskatīsim kādā veidā var novērtēt entropiju izplūdušai kopai. Ar entropijas palīdzību var novērtēt sistēmas vai ziņojuma nenoteiktību.

Izplūdušas kopas *A* entropija *E(A)* var pieņemt vērtības no nulles līdz vieniniekam. Tā ir vienāda ar nulli, ja tā ir precīza, jeb neizplūdusi kopa, un tā ir vienāda ar vienāda ar vieninieku, ja tā ir "maksimāli izplūdusi", kas nozīmē, ka tā atrodas hiperkuba I^n centrā. Tādējādi, entropiju var nodefinēt kā attiecību (1.28), proti, entropija ir vienāda ar attālumu līdz tuvākai hiperkuba virsotnei dalītu ar attālumu līdz tālākai hiperkuba virsotnei.

$$E(A) = \frac{l^1(A, A_{near})}{l^1(A, A_{far})}. \tag{1.28}$$

1.5. attēlā nav grūti pamanīt, ka attālums līdz tuvākai virsotnei ir vienāds ar $M(A \cap A^c)$ un attālums līdz tālākai virsotnei ir vienāds ar $M(A \cup A^c)$. Tātad, attiecību (1.28) var pārrakstīt kā (1.29):

$$E(A) = \frac{M(A \cap A^c)}{M(A \cup A^c)}. \tag{1.29}$$

Kā ir redzams, ja kopa *A* nav izplūdusi, tad *E(A)=0*. Savukārt, izplūdušas kopas entropija ir vienāda ar nepretrunīguma likuma pārkāpšanu skaitu dalītu ar izslēgta trešā likuma pārkāpšanu skaitu.

1.4.5. Entropija un nejaušība

Dažos rakstos, apskatot informācijas teoriju un entropiju, tiek skarti ar nejaušību un nepilnību saistīti jautājumi.

Piemēram, rakstā [Chaitin, 2002] tiek apskatīti vairāki paradoksi un ir apskatīta algoritmiskās informācijas teorijas izmantošanas iespējas jaunu paradoksu aprakstīšanai. Šajā rakstā tiek apskatīts pazīstams meļa paradokss, Gēdela teorēma par nepilnīgumu un tiek piedāvātas dažas pieejas nepilnīguma pierādījumam, izmantojot algoritmisku informācijas teoriju.

Pirmā pieeja nav ļoti formāla un Čeitins norāda, ka tā ir nepilnības teorēmas vienkāršota versija. Visus naturālus skaitļus sadala divās kopās: "interesanti skaitļi" un "neinteresanti skaitļi". Pēc kaut kāda principa to var izdarīt. Tagad aplūkosim pirmo neinteresantu skaitli. To var atrast pēc ļoti vienkārša algoritma: mēs apskatām visus skaitļus pēc kārtas: 1, 2, 3... un kad sastopam pirmo skaitli, kas nav interesants, mēs apstājamies. Bet tas, ka tas ir tieši pirmais neinteresants skaitlis, padara to par interesantu skaitli. Kā ir redzams, te ir problēma.

Pieņemsim, ka mēs varām formāli pierādīt, ka kāds skaitlis ir vai nav interesants. Tagad apskatīsim pirmo skaitli, kuram mēs varām pierādīt, ka tas ir neinteresants skaitlis. Bet tas ir ļoti interesants fakts, ka tas ir *tieši pirmais* vesels skaitlis, kuram mēs varam pierādīt, ka tas ir neinteresants skaitlis. Tātad, ja nav pirmā skaitļa, kuram var pierādīt, ka tas ir neinteresants skaitlis, tad tas nozīmē, ka nekādam konkrētam veselam skaitlim nevar pierādīt, ka tas ir neinteresants skaitlis.

Turklāt, Čeitins parāda, ka nevar pierādīt vai programma ir *eleganta* (sk. 1.4.3. paragrāfu). Precīzāk sakot, to var izdarīt tikai tādām programmām, kuru garums ir mazāks par universāla datora programmas garumu. Apspriežot Einšteina jautājumu par to, vai "Dievs spēlē kauliņus", Čaitins raksta, ka viņš uzskata, ka "Dievs spēlē kauliņus" algoritmiskajā informācijas teorijā.

Ar nejaušību un nepilnību saistīti jautājumi tiek apskatīti arī B. Kosko grāmatā [Kosko, 1992]. Ar nejaušību šajā grāmatā tiek saprasta varbūtība un ir piedāvāti vairāki iemesli kāpēc ar varbūtisko parametru palīdzību nevar aprakstīt reālas pasaules nenoteiktību visās tās "izpausmēs". Viena no izplūdušas loģikas galvenajām iezīmēm ir tas, ka tā apskata nevis to, ar kādu varbūtību kaut kas notiks (precīzāk sakot, ne tikai to), bet *pakāpi*, kurā kaut kas notiks. Savukārt, saskaņā ar bivalentu loģiku, kaut kas var vai nu notikt, vai nenotikt, tātad viss ir vai nu *melns*, vai nu *balts*.

B. Kosko izplūdumu apskata kā *īpašību*, kas tam piemīt, un kas nav atkarīgs no eksperimentu skaita. Piemēram, pīļknābis tikai apmēram ir zīdītājs, bet liels pakalns tikai apmēram ir kalns, un šie novērtējumi nav atkarīgi no eksperimentu skaita. Tātad, B.Kosko secina, ka izplūdušajā loģikā Dievs kauliņus nespēlē.

Apskatīsim vienu piemēru, lai labāk izprastu atšķirību starp nejaušību un izplūdumu. Pieņemsim, ka mums ir zināms, ka ar varbūtību 50% ledusskapī atrodas ābols. Iespējams, ka šīs skaitlis tika iegūts veicot vairākus eksperimentus, bet tas nav būtiski. Tagad pieņemsim, ka mums ir zināms, ka ledusskapī ir ābola puse. Skaitliskā ziņā šīs situācijas ir ekvivalentas, tomēr pēc savas būtības tās atšķiras: pirmā ir *nejauša* un otrā ir *izplūdusi*. Turklāt, otrā situācija ir pilnīgi determinēta: mums ir zināmi visi fakti, tātad varbūtisko novērtējumu izmantošana šajā gadījumā būtu pret mūsu intuīciju.

Turklāt, grāmatā [Kosko, 1992] tiek apskatīti ar kvantu mehāniku saistīti jautājumi. Tiek apskatīts jautājums par to, vai elektrons atrodas kādā telpas daļā ar kādu varbūtību, vai tomēr ir jāapskata *pakāpe*, ar kuru elektrons aizņem kādu telpas daļu. Kā iepriekš, pirmā "interpretācija" ir varbūtiska, bet otrā – izplūdusi.

Grāmatā [Kosko, 1992] tiek apskatīti klasiskās loģikas un kopu teorijas paradoksi. Kā norāda autors, visi šie paradoksi parādās 1.4.4. paragrāfā aprakstītajā hiperkuba centra punktā (sk. 1.3. attēlu) un, ka šis punkts patiesībā ir klasiskās teorijas "melnais caurums": visiem pārējiem hiperkuba punktiem var atrast vistuvāko hiperkuba virstoni un "reducēt" punktu uz šo virsotni (citiem vārdiem, izplūdušai kopai tiek piemeklēta vistuvākā precīza kopa un izplūdusi kopa tiek reducēta uz šo precīzo kopu). Bet centra punktā tas nav iespējams, jo attālums līdz visām virsotnēm ir vienāds.

Apskatīsim Rassela paradoksu par bārddzini. Rassela bārddziņa frizētavā ir rakstīts, ka viņš skuj vīrieti tad un tikai tad, ja šīs vīrietis pats sevi neskuj. Tātad, kas skuj bārddzini? Ja bārddzinis pats sevi skuj, tad saskaņā ar uzrakstu frizētavā, viņš pats sevi neskuj, bet ja viņš pats sevi neskuj, tad, atkal, saskaņā ar uzrakstu, viņš pats skuj. Apzīmēsim spriedumu, ka bārddzinis skuj pats sevi ar *S* un spriedumu, ka viņš pats sevi neskuj ar *not-S*. Tā kā no *S* seko *not-S* un no *not-S* seko *S*, šie spriedumi ir ekvivalenti: *S* = *not-S*. Ekvivalentiem spriedumiem patiesuma vērtības ir vienādas, tātad, ja ar *t(S)* apzīmē sprieduma *S* patiesumu, tad:

$$t(S) = t(\text{not-}S) = 1 - t(S). \tag{1.30}$$

Atrisinot (1.30), dabūjam: $t(S) = 0.5$, kas ir hiperkuba centra punkts. Tātad, ja mēs atrodamies bivalentas loģikas pasaulē, mēs šo vērtību nevaram reducēt ne uz nulli, ne uz vieninieku, jo attālums līdz tiem ir vienāds un rodas paradokss. Izplūdušajā atrisinājumā paradokss neparādās un vērtība *0.5* tiek iegūta no uzdevuma nostādnes.

Kā atzīmē Kosko, cilvēks, kurš ar skepsi izturas pret izplūdušo loģiku un kurš ir nonācis pie hiperkuba centra punkta paradoksa, atgādina cilvēku, kurš ir pārliecināts, ka Zeme ir plakana un stāvot ziemeļpolā ar kompasu rokās mēģina doties uz dienvidiem [Kosko, 1992].

Rakstā [Hawking, 2000] var atrast diskusiju par to, cik daudz nejaušības ir mūsu pasaulē. Raksta autors ir pārliecināts, ka mūsu pasaule ir nejauša (izmantojot Einšteina terminoloģiju, tas nozīmē, ka "Dievs spēlē kauliņus"). Vispirms autors apskata kvantu mehāniku un Heicenberga nenoteiktības principu. Šī principa būtība ir tāda, ka nevar vienlaicīgi precīzi novērtēt daļiņas ātrumu un izvietojumu. Jo precīzāk mēra ātrumu, jo neprecīzāks ir izvietojuma novērtējums un otrādi. Tātad, no šī principa izriet tas, ka mūsu pasaulē ir kaut kas, ko nevar precīzi novērtēt. Savukārt, determinisma ideja ir tāda, ka ja mēs zinātu mūsu pasaules precīzu aprakstu, tad mēs varētu izrēķināt pasaules stāvokli jebkurā laika momentā nākotnē vai pagātnē. Heicenberga princips apgalvo, ka mēs nevaram precīzi novērtēt pasauli.

Tomēr, tas, ka mēs nevaram saņemt precīzu pasaules novērtējumu, nenozīmē, ka šāda novērtējuma vispār neeksistē. Tātad, iespējams, ka Dievam tāds novērtējums ir. Turklāt, apskatāmais jautājums taču ir, vai Dievs, nevis cilvēki, spēlē kauliņus. Šo iemeslu dēļ raksta [Hawking, 2000] autors turpina diskusiju un apskata melnos caurumus. Kā ir zināms, melno caurumu gravitācijas laukums ir tik spēcīgs, ka mēs tos neredzam, jo gaismas kvanti nevar pamest to gravitācijas laukumu. Tagad iedomāsimies, kas notiek ar objektiem, kuri nokļūst melnajos caurumos. Droši vien, ka tie tur arī paliks, bet kas notiek ar informāciju, kuru šie objekti satur? Ja melnie caurumi būtu mūžīgi, tad nekas slikts nenotiktu, jo informācija par šiem nelaimīgajiem objektiem paliktu melnajos caurumos, vienīgi mēs to nevarētu nolasīt vai ieraudzīt. Bet melnajiem caurumiem ir starojums, tātad ar laiku tie izzūd. Melno caurumu starojums ir nejaušs, proti, tas nekādā veidā nav saistīts ar to, kādi objekti ir nokļuvuši melnajā caurumā. Tādējādi, objektu informācija izzūd kopā ar melnajiem caurumiem. Raksta autors rezumē savu rakstu ar to, ka Einšteins ir divkārši kļūdījies, jo Dievs ne tikai met kauliņus, bet Viņš tos met tur, kur mēs tos nevaram ieraudzīt.

1.5. Pētījuma mērķis un uzdevumi

Promocijas darba mērķis ir izstrādāt vidē ar diviem nenoteiktības avotiem (nejaušība un izplūdums) izmantošanai paredzētu lēmumu pieņemšanas metodi. Tiek apskatīts gadījums, kad alternatīvu un kritēriju skaits ir galīgs. Metode sastāv no vairākiem posmiem. Proti, metodes izstrādei ir jāatrisina šādi apakšuzdevumi:

1. Jāizpēta eksistējošās nenoteiktajā vidē ar vienu vai vairākiem nenoteiktības avotiem izmantošanai paredzētas lēmumu pieņemšanas metodes.
2. Jāizstrādā adaptīvais tīkls izplūdušo granulu apstrādei. Jāizstrādā adaptīvā tīkla apmācīšanas algoritms, kuru var izmantot alternatīvu parametru svarīguma noteikšanai.
3. Alternatīvu informatīvuma noteikšanai izmanto Šenona entropiju. Tomēr, metodē tiek izmantotas intervālu varbūtības. Tādējādi, Šenona entropija ir jāvispārina intervālu gadījumam un jāpierāda, ka tā ir aditīva.
4. Jāizstrādā alternatīvu pilnās un daļējās sakārtošanas metode.
5. Atsevišķus izstrādātus līdzekļus ir jāapvieno lēmumu pieņemšanas metodē, kura ir paredzēta izmantošanas vidēs, kuras ir gan nedeterminētas, gan izplūdušas.
6. Jāizstrādā programmnodrošinājums eksperimentu veikšanai.
7. Izstrādātās metodes praktiskā lietojamība jāpierāda uz uzdevuma, kurš atbilst reālās pasaules apstākļiem.

1.6. Secinājumi par 1. nodaļu

Pirmajā nodaļā ir apskatīti vispārīgi ar lēmumu pieņemšanu saistīti jautājumi, kā arī izplūduma un entropijas jēdziens. Var izdarīt šādus secinājumus.

1. Lēmumu pieņemšana var norisināties dažādos apstākļos: noteiktības, nenoteiktības, riska un izplūduma apstākļos, kā arī, kad vide ir gan izplūdusī, gan nedeterminēta.
2. Izplūdušo loģiku var izmantot, lai aprakstītu objektus vai procesus, kuriem ir izplūdušas, jeb nenoteiktas īpašības. Līdz ar to objekta izplūdums kļūst par tā īpašību, nevis raksturojumu, kurš ar laiku var izmainīties vai izzust, piemēram, kā tas ir ar nedeterminētiem novērtējumiem.
3. Izplūdušās granulas var izmantot, lai modeļa aprakstā apvienotu izplūdumu un nenoteiktību. Proti, izplūdušas liecības var uzskatīt par izplūdušo vērtību varbūtību sadalījumu. Līdz ar to modelī var aprakstīt divus dažādus nenoteiktības avotus.
4. Nenoteiktības pakāpes novērtēšanai var izmantot entropiju. Informācijas teorijā tā tiek izmantota, lai noteiktu cik daudz informācijas par sistēmu nav zināms. Savukārt, fizikā entropija apzīmē kārtības trūkumu jeb nejaušību. Turklāt, entropija tiek izmantota algoritmiskajā informācijas teorijā un izplūdušo kopu teorijā nenoteiktības un informatīvuma novērtēšanai.

2. LĒMUMU PIEŅEMŠANAS METODES NENOTEIKTAJAI VIDEI

Zemāk ir aprakstītas lēmumu pieņemšanas metodes, kuras ir paredzētas izmantošanai nenoteiktajās vidēs ar vienu vai vairākiem nenoteiktības avotiem. Metodes tika izvēlētas tā, lai pēc iespējas pilnīgāk aptvertu dažāda veida pieejas lēmumu pieņemšanai. Sākumā tiek aprakstītas heiristiskas metodes PROMETHEE un ELECTRE, pēc tam tiek apskatītas AHP (Analytic Hierarchy Process) un Swing metodes, kuras var izmantot kritēriju svēršanai. Pēc svēršanas metodēm tiek apskatīta formāla lēmumu analīze, kurai seko lēmumu pieņemšanas metožu apraksts, kuras ir paredzētas videi ar vairākiem nenoteiktības avotiem. Paralēli lēmumu pieņemšanas metodēm tiek aprakstīti tajās izmantojamie palīglīdzekļi, proti, morfoloģiskā analīze un jūtīguma analīze.

2.1. Lēmumu pieņemšanas metodes un palīglīdzekļi

Šīs nodaļas sākumā tiks apskatīti daudzkritēriju lēmumu pieņemšanas metožu veidi un pēc tam tiks apskatītas daudzkritēriju lēmumu pieņemšanas metodes un uz kritēriju svēršanas balstītās preferenču modelēšanas metodes. Vispirms tiek apskatīta morfoloģiskā analīze, kura ir metode alternatīvu kopas veidošanai, pēc tam tiek apskatītas PROMETHEE un ELECTRE metodes, kuru pamatā ir tabula ar kritēriju vērtībām visām alternatīvām.

2.1.1. Daudzkritēriju lēmumu pieņemšanas metodes

Apskatīsim lēmumu pieņemšanas uzdevuma modeli, kurā ietilpst zemāk norādītie elemnti:

- X – alternatīvu kopa;
- Y – iznākumu kopa;
- f_i: $Y \rightarrow R, i = 1, \ldots, m$ – kritēriju (kvalitātes rādītāju) kopa;
- φ: $X \rightarrow Y$ – attiecība, kura katrai alternatīvai piekārto iznākumu.

Tomēr, kā tika norādīts iepriekš, iznākumu analīzi var izslēgt no lēmumu pieņemšanas uzdevuma analīzes. Ja lēmums tiek pieņemts noteiktības apstākļos, tad funkcija φ ir viennozīmīga.

Ērtības labad ievedīsim jaunu funkciju J, kura ir divu funkciju superpozīcija un tās vērtība ir alternatīvas kritērija novērtējums:

$$J_i(x) = f_i(\varphi(x)),\ i = 1,\ \dots,\ m$$

Tādējādi, alternatīvas novērtējums ir vektors:

$$J\colon x \rightarrow R^m,\ J=(J_1\ \dots\ J_m)$$

Ar F apzīmēsim novērtējumu kopu

$$J : X \to F \subset R^m .$$

Ir viegli ieraudzīt, ka ja kādā no kopām $X,\ Y,\ F$ ir uzdota preferenču attiecība, tad to pārnest arī uz citām kopām, ja uzdevums tiek risināts noteiktības apstākļos. Jo visas attiecības ir viennozīmīgas. Tādējādi, lēmumu pieņemšanas uzdevumu par pierakstīt šādi:

$$J_i(x) \to \max_{x \in X}, i = 1, \dots, m, X \subset R^n ,$$

kur n ir kritēriju skaits un m ir alternatīvu skaits. Kā redzams, katrai alternatīvai atbilst punkts $x \in R^n$.

2.1.1.1. Galvenā kritērija metode

Apskatīsim galvenā kritērija metodi [Černoruckij, 2005]. Šajā metodē par mērķa funkciju tiek izvēlēts kāds no funkcionāliem, piemēram, f_1, kurš vispilnīgāk (no lēmējpersonas viedokļa) apraksta lēmuma pieņemšanas uzdevuma mērķi. Pārējas prasības rezultātam, kuras apraksta pārējie funkcionāli, tiek uzdotas ar ierobežojumu palīdzību. Tādējādi, uzdevums tiek reducēts uz šādu vienkritērija uzdevumu:

$$f_1(x) \to \max_{x \in D'};\quad D' \subseteq D \subseteq R^n;$$
$$D' = \left\{ x \in D \mid f_i(x) \ge t_i, i = 2, \dots, m \right\}.$$

Šeit ar D tiek apzīmēta alternatīvu, jeb pieļaujamo risinājumu kopa. Kā redzams, tā ir ierobežota ar t_i parametru palīdzību. Rezultātā mums ir jāatrisina vienkāršotu maksimizēšanas uzdevumu. Ir acīmredzams, ka vienkāršots uzdevums un vispārējs uzdevums nav ekvivalenti.

Pielietojot galvenā kritērija metodi bieži izrādās, ka metode nav lietojama, jo ir vairāk par vienu "galveno" kritēriju. Turklāt, ne vienmēr ir skaidrs pēc kādiem principiem jāizvēlas t_i parametri. To nepareiza uzdošana var novest, piemēram, pie tukšās kopas D'.

2.1.1.2. Svērtas summas metode

Tā ir viena no visbiežāk lietotām "skalarizēšanas" (svērtas summas) metodēm, kas ļauj aizvietot vērtējuma vektoru $\boldsymbol{f} = (f_1, ...f_m)$ ar skaitli $J : D \rightarrow R$. Tā balstās uz visu mērķa funkcionālu lineāru apvienošanu [Power, 2002]:

$$J(x) = \sum_{i=1}^{m} \alpha_i f_i(x) \rightarrow \max_{x \in D'}; \ \alpha_i > 0, \sum_{i=1}^{m} \alpha_i = 1.$$

Svarus α_i var uzskatīt par atsevišķo kritēriju svarīguma rādītājiem. Jo lielāka ir svara vērtība, jo lielāka ir atbilstoša kritērija ieguldījums kopējā novērtējumā.

Šajā gadījumā grūtības var rasties, ja ir dažāda rakstura kritēriji, jo tad var būt grūti noteikt cik lielā mērā jāatšķiras kritēriju vērtībām.

2.1.1.3. Maxmin reducēšanas metode

Parasti tā tiek izmantota šādā formā [Černoruckij, 2005]:

$$J(x) = \min_i f_i(x) \rightarrow \max_{x \in D'}$$

Šajā gadījumā, atšķirībā no svērtas summas metodes, mērķa funkcionālu $J(x)$ ietekmē tikai tas kritērijs, kuram dotajā punktā x ir vismazākā funkcijas $f_i(x)$ vērtība. Tādējādi, pēc $J(x)$ vērtības var noteikt garantēto apakšējo novērtējumu visiem funkcionāliem $f_i(x)$.

Gadījumā, ja atsevišķu mērķa funkciju vērtības jānormē, lai izlīdzinātu to skalas, funkcionāli ir jāpareizina ar svariem, šajā gadījumā tiek izmantota svērta maxmin kritērija forma:

$$J(x) = \min_i \alpha_i f_i(x) \rightarrow \max_{x \in D'}; \ \alpha_i > 0, \sum_{i=1}^{m} \alpha_i = 1$$

2.1.1.4. Lēmumu pieņemšana nenoteiktības apstākļos

Viens no veidiem kā nenoteiktību var iekļaut lēmuma pieņemšanas modelī ir aprakstīts grāmatā [Černoruckij, 2005]. Nenoteiktība izpaužas ar to, ka mēs precīzi nezinām kāds iznākums būs alternatīvai. Tādējādi, varam izveidot šādu modeli: $X = \{x_1, ..., x_n\}$ ir alternatīvu kopa un $Z = \{z_1, ..., z_m\}$ ir stāvokļi, kurā var nokļūt daba (no tiem ir atkarīgi alternatīvu iznākumi). Tātad, iznākumu kopa ir $Y = \{y_{11}, \ldots, y_{nm}\}$ ir visi tie iznākumi, kuri var rasties visos vides stāvokļos z_i.

Dominēšanas un vienlīdzības attiecības tiek definētas šādi:

$$x_1 \succ x_2 \leftrightarrow \forall z \in Z : J(x_1, z) \geq J(x_2, z),$$

$$x_1 \sim x_2 \leftrightarrow \forall z \in Z : J(x_1, z) = J(x_2, z).$$

Ja lēmumu pieņemšana tiek veikta riska apstākļos, proti, kad ir pieejama informācija par vides stāvokļu iestāšanos varbūtībām, tad analīzē ir jāiekļauj stāvokļu z_i varbūtības.

Nākamajā nodaļā tiek apskatīta morfoloģiskā analīze, kuru var izmantot alternatīvu kopas veidošanai.

2.1.2. Morfoloģiskā analīze

Morfoloģiskā analīze ir domāta alternatīvu kopas izveidošanai. Tas ir līdzeklis, ar kura palīdzību ir iespējams ģenerēt jaunas alternatīvas [Clemen&Reilly, 2004; Polovinkin, 1988]. Morfoloģiskās analīzes ideja ir šāda: ir jāsastāda atribūtu saraksts un ir jānosaka kādas ir šo atribūtu iespējamas vērtības. Tad ir jāizveido tabula, kuras kolonas atbilst atribūtiem un tās šūnās ir jāieraksta atribūtu vērtības. Lai izveidotu jaunu alternatīvu katrā tabulas kolonā ir jāizvēlas viena vērtība. Izvēlēto vērtību kopums raksturos vienu alternatīvu. Lai šī ideja būtu skaidrāka, apskatīsim dažus piemērus.

2.1.2.1. Vienkāršas morfoloģiskās analīzes piemēri

1. piemērs. Pieņemsim, ka stingri konkurences apstākļi noveda pie tā, ka ātrās ēdināšanas restorānam *BurgerQueen* ir jāpiedāvā plašāks hamburgeru klāsts. Ar šo nolūku tika ievesta hamburgeru pasūtīšanas procedūra, kura paplašina hamburgeru klāstu līdz 64 vienībām. Papildus maizei un gaļai klients var izvēlēties vai pievienot hamburgeram šādus produktus:

1. Kečups: Jā / Nē
2. Majonēze: Jā / Nē
3. Sālīts gurķis: Jā / Nē
4. Tomāts: Jā / Nē
5. Siers: Jā / Nē
6. Papildus gaļas gabals: Jā / Nē

Šāda ēdienu pasūtīšanas procedūra dod klientiem vairāk brīvības. Savukārt, pārdevējs var ietaupīt nedrukājot garus sarakstus ar iespējamajiem hamburgeru veidiem.

Apskatīsim otro piemēru.

2. piemērs. Pieņemsim, ka ir jāizstrādā jauns lampas modelis. Lai to paveiktu ir jānosaka atribūti, jāsastāda tabula ar atribūtu vērtībām un jānoģenerē alternatīvas.

Pieņemsim, ka mūs interesē šādi atribūti:

1. Enerģijas avots
2. Spuldzes tips
3. Gaismas intensitāte
4. Izmērs
5. Stils
6. Apdare
7. Materiāls

Šo atribūtu iespējamas vērtības ir uzskaitītas 2.1. tabulā:

2.1. tabula

Lampas atribūtu iespējamas vērtības morfoloģiskajai analīzei

Enerģijas avots	**Spuldzes tips**	**Gaismas intensitāte**	**Izmērs**	**Stils**	**Apdare**	**Materiāls**
Baterija	Halogēna	Zema	Ļoti liels	Moderns	Melna	Metāls
Elektrības tīkls	Parasta spuldze	Vidēja	Liels	Antīks	Balta	Keramika
Saule	Dienas gaismas	Augsta	Vidējs	Romu	Metāliska	Betons
Ģenerators	Krāsaina spuldze	Maināma	Mazs	Jūgendstils	Audums	Kauls
Rokas piedziņa	Nav		Pārnesams	Industriāls	Emaljēts	Stikls
Gāze				Etniskais		Koks
Eļļa						Akmens
						Plastmasa

Tagad ar 2.1. tabulas palīdzību mēs varam nejaušā veidā veidot alternatīvus lampas modeļus. Kā arī mēs varam veidot tās mērķtiecīgi, ja mūs interesē kāds noteikts lampas veids. Piemēram, ar 2.1. tabulas palīdzību mēs varam izveidot šādu lampas modeli:

2.2. tabula

Lampas atribūtu vērtības vienam iespējamam modelim

Enerģijas avots	**Spuldzes tips**	**Gaismas intensitāte**	**Izmērs**	**Stils**	**Apdare**	**Materiāls**
Eļļa	Nav	Vidēja	Vidējs	Antīks	Balta	Keramika

Kā redzams 2.2. tabulā, viens no lampas modeļiem varētu būt antīkas lampas replika, kura ir paredzēta uzstādīšanai atbilstoša tipa restorānos vai interjeros.

2.1.2.2. Funkcionālā pieeja morfoloģiskajai analīzei

Šajā apakšnodaļā tiek apskatīta funkcionālā pieeja morfoloģiskajai analīzei [Polovinkin, 1988; Andrejčikov&Andrejčikova, 2004].

Morfoloģiskā analīze ir kombinatoriska metode, proti, vispirms mēs nosakām alternatīvu vai izstrādājumu atribūtus un katra atribūta iespējamas vērtības. Kombinējot savā starpā šīs vērtības var iegūt dažāda veida izstrādājumus vai risinājumus, arī tādus, kuriem ir praktiskā vērtība. Kā ir redzams, šī metode ir balstīta uz morfoloģiskās tabulas izveidošanas, to šūnu aizpildīšanas ar atribūtu iespējamām vērtībām un risinājumu ģenerēšanas.

Morfoloģiskā analīze sākas ar funkcionālas struktūras izveidošanu. Apskatīsim vienkāršoto piemēru no grāmatas [Polovinkin, 1988]. Pieņemsim, ka mums jāapraksta mājas sastāvdaļu funkcijas. Ar šo nolūku izveidojam tabulu 2.3., kurā šīs funkcijas ir aprakstītas.

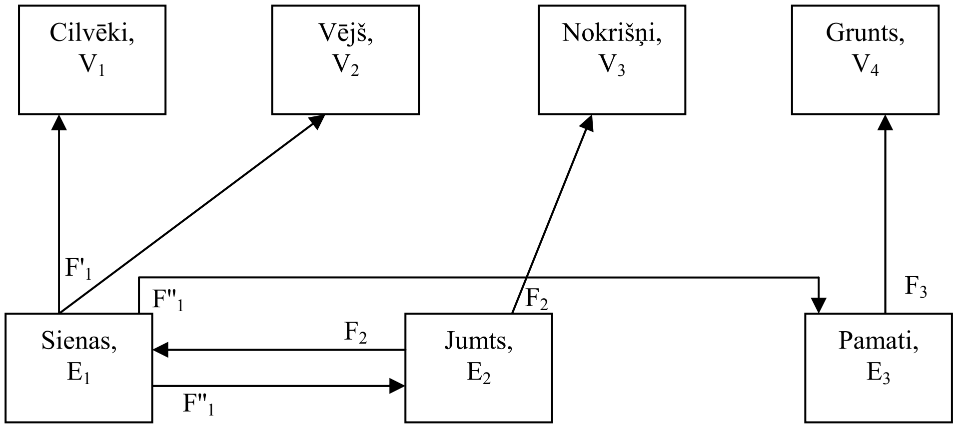
2.3. tabula

Mājas funkcijas

Elementi		Funkcijas	
Apzīmējums	Nosaukums	Apzīmējums	Apraksts
E_1	Sienas	F'_1	Cilvēku (V_1) aizsardzība no vēja (V_2)
		F''_1	Jumta (E_2) slodzes nodošana uz pamatiem (E_3)
E_2	Jumts	F_2	Sienu (E_1) aizsardzība no nokrišņiem (V_3)
E_3	Pamati	F_3	Masas iedarbības nodošana uz grunts (V_4)

2.3. tabulā ar E_i ir apzīmēti mājas elementi, ar F_i ir apzīmēti to funkcijas un ar V_i ir apzīmēti apkārtējās vides objekti.

Izmantojot 2.3. tabulu mēs varam izveidot mājas funkcionālo struktūru, kura ir parādīta 2.1. attēlā.



2.1. att. Mājas konstruktīvā funkcionālā struktūra

Funkcionālo struktūru izmanto morfoloģiskās tabulas veidošanai. Tabula sastāv no vairākām kolonām, kuru skaits parasti sakrīt ar funkcionālo elementu skaitu. 2.1. attēlā paradītajai struktūrai atbilst 2.4. tabula.

2.4. tabula

Māju morfoloģiskā tabula

F'_1, F''_1, F'''_1	F_2	F_3
Koka sienas	Spāres ar skārda pārklājumu	Betona pamati
Fibo bloku sienas	Spāres ar dakstiņiem	Lentes pamati
Gāzbetona bloku sienas		Pāļu pamati
Dzelzsbetona sienas		

Aizpildot morfoloģisko tabulu jāseko tam, lai atribūtu iespējamas vērtības atbilstu efektīviem un reāliem risinājumiem. Var izmantot:

- savas zināšanas un no ekspertiem iegūto informāciju;
- enciklopēdijas un rokasgrāmatas;
- tehnisko funkciju vārdnīcas;
- patentu katalogu;
- izstāžu katalogus.

Turklāt, aizpildot tabulu var izmantot tādas alternatīvas metodes kā prāta vētras un citas heiristiskās metodes.

Tagad apskatīsim kādā veidā izmantojot morfoloģisko tabulu var izvēlēties vispiemērotākos vai visefektīvākos risinājumus, jo visas kombinācijas parasti nav iespējams apskatīt to liela skaita dēļ. Nav grūti izrēķināt visu iespējamu kombināciju skaitu. Tas ir vienāds ar atribūtu iespējamo vērtību skaita reizinājumu. Piemēram, izmantojot 2.4. tabulu var izveidot *4 * 2 * 3 = 24* kombinācijas.

Iespējamu risinājumu skaitu var samazināt samazinot atribūtu vērtību skaitu, jeb atribūtu skaitu. Atribūtu vērtību skaitu var samazināt, pakāpeniski atmetot neefektīvus risinājumus. Var rīkoties šādi. Apskatam divus atribūtus ar vismazāko iespējamo vērtību skaitu. 2.4. tabulā tā ir 2. un 3. kolonna. Analizējam visas iespējamas kombinācijas, kuras var iegūt no šiem diviem atribūtiem. Ja mēs apskatam 2.4. tabulu, kopējais kombināciju skaits ir 6. Pieņemsim, ka pēc analīzes mēs ieguvām 2.5. tabulā parādītus rezultāts. Atribūtu iespējamas vērtības ir apzīmētas ar $A_{i,j}$. 2.5. tabulā neefektīvie vai nederīgie varianti ir aizsvītroti un turpmāk netiek analizēti. Kā ir redzams, 2.5. tabulā palika tikai 3 kombinācijas.

2.5. tabula

Māju 2. un 3. atribūta kombinācijas

F_3 \ F_2	$A_{2,1}$	$A_{2,2}$
$A_{3,1}$	~~$A_{2,1}$ $A_{3,1}$~~	$A_{2,2}$ $A_{3,1}$
$A_{3,2}$	$A_{2,1}$ $A_{3,2}$	$A_{2,2}$ $A_{3,2}$
$A_{3,3}$	~~$A_{2,1}$ $A_{3,3}$~~	~~$A_{2,2}$ $A_{3,3}$~~

Nākamajā solī pievienojam vēl neapskatītu atribūtu, kuram ir vismazākais iespējamo vērtību skaits, un apskatam šī atribūta kombinācijas ar risinājumiem, kuri palika pēc pirmo divu kolonnu analīzes. 2.4. tabulā palika tikai viena neapskatīta kolonna, proti, 1. kolonna. 2.6. tabulā ir attēloti visas iespējamas šīs kolonnas un 2.5. tabulā neaizsvītroto elementu kombinācijas.

2.6. tabula

Māju 1. atribūta un 3.4. tabulas elementu kombinācijas

F_1 \ F_2 F_3	$A_{2,2}$ $A_{3,1}$	$A_{2,1}$ $A_{3,2}$	$A_{2,2}$ $A_{3,2}$
$A_{1,1}$	~~$A_{1,1}$ $A_{2,2}$ $A_{3,1}$~~	$A_{1,1}$ $A_{2,1}$ $A_{3,2}$	$A_{1,1}$ $A_{2,2}$ $A_{3,2}$
$A_{1,2}$	$A_{1,2}$ $A_{2,2}$ $A_{3,1}$	~~$A_{1,2}$ $A_{2,1}$ $A_{3,2}$~~	$A_{1,2}$ $A_{2,2}$ $A_{3,2}$
$A_{1,3}$	~~$A_{1,3}$ $A_{2,2}$ $A_{3,1}$~~	$A_{1,3}$ $A_{2,1}$ $A_{3,2}$	~~$A_{1,3}$ $A_{2,2}$ $A_{3,2}$~~
$A_{1,4}$	~~$A_{1,4}$ $A_{2,2}$ $A_{3,1}$~~	~~$A_{1,4}$ $A_{2,1}$ $A_{3,2}$~~	~~$A_{1,4}$ $A_{2,2}$ $A_{3,2}$~~

Kā ir redzams 2.6. tabulā, mums palika tikai 5 alternatīvas kombinācijas, kuras mums turpmāk jāizanalizē. Šīs kombinācijas ir attēlotas zemāk. $A_{i,j}$ ir atbilstoša atribūta vērtība.

$A_{1,1}$ $A_{2,1}$ $A_{3,2}$

$A_{1,1}$ $A_{2,2}$ $A_{3,2}$

$A_{1,2}$ $A_{2,2}$ $A_{3,1}$

$A_{1,2}$ $A_{2,2}$ $A_{3,2}$

$A_{1,3}$ $A_{2,1}$ $A_{3,2}$

Tādējādi, mums nav jāanalizē visas 24 iespējamas kombinācijas.

Vispārīgā gadījumā, ja paliek vēl neapskatīti atribūti, jārīkojas analoģiskā veidā, kamēr tiks apskatīti visi atribūti. Pēc visu atribūtu apskatīšanas ir jāizanalizē atlikušas kombinācijas.

2.1.3. PROMETHEE lēmumu pieņemšanas metode

Pirms PROMETHEE metodes pielietošanas ir jāsastāda tabula ar kritēriju vērtībām katrai alternatīvai. Tas nozīmē, ka šī metode nepieļauj neprecizitāti ieejas datos, proti, visām vērtībām ir jābūt determinētām un precīzām.

PROMETHEE metode sastāv no diviem posmiem. Pirmajā posmā izmantojot vispārinātus kritērijus (*generalized criteria*) ir jāuzbūvē tā sauktā pārspēšanas attiecība (*outranking relation*), kura ir definēta alternatīvu apgabalā, un otrajā posmā izmantojot šo attiecību ir jāuzbūvē daļējs vai pilnais sakārtojums, kas tiek iegūts rēķinot ienākošās un izejošās plūsmas.

No citām metodēm PROMETHEE atšķiras ar to, ka tajā tiek izmantoti vispārinātie kritēriji. Šo kritēriju pamatideja ir tāda, ka dažādu kritēriju vērtības tiek salīdzinātas dažādos veidos, kas nozīmē, ka atšķiras arī preferences definēšana. Rakstā [Brans u.c., 1986] ir definēti seši dažādi kritēriju tipi.

Pieņemsim, ka K ir galīga alternatīvu kopa. Šīs kopas elementiem ir definēti reālie kritēriji:

$$f : K \rightarrow R$$

Preferences funkcija tiek definēta sekojošajā veidā:

$$P : K \times K \rightarrow [0; 1]$$

Ja P ir tuvu vieniniekam, tas nozīmē, ka pirmā alternatīva pārspēj otro. Ja P vērtība ir tuvu nullei, tas nozīmē, ka starp abām alternatīvām ir vienaldzības relācija, vai ka pirmā alternatīva nepārspēj otro. Preferences funkcija ir definēta kritēriju vērtību starpību apgabalā:

$$d = f(a) - f(b)$$

$$P(a, b) = \pi (d)$$

Lai labāk attēlotu vienaldzības intervālu vispārināto kritēriju definēšanai tiek izmantota funkcija $H(d)$:

$$H(d) = \begin{cases} P(a,b), & d \geq 0 \\ P(b,a), & d < 0 \end{cases}$$

2.7. tabulā ir parādīti seši PROMETHEE metodē izmantojamie kritēriju veidi.

Pieņemsim, ka lēmumu pieņemoša persona ir nodefinējusi savas preferences ar svaru palīdzību, katram kritērijam f_i atbilst svars w_i. Tad daudzkritēriju preferences attiecība Π tiek definēta šādā veidā:

$$\Pi(a,b) = \frac{\sum_{i=1}^{k} \pi_i P_i(a,b)}{\sum_{i=1}^{k} \pi_i}.$$

Attiecība $\Pi(a, b)$ parāda cik augstāk lēmumu pieņemoša persona vērtē alternatīvu a par alternatīvu b.

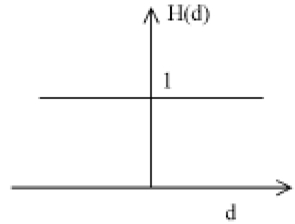
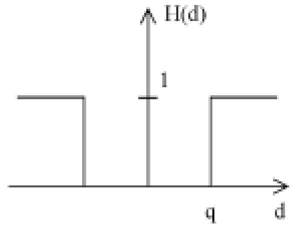
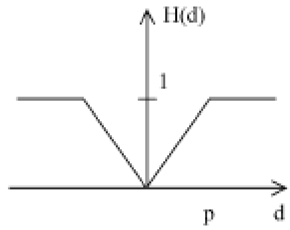
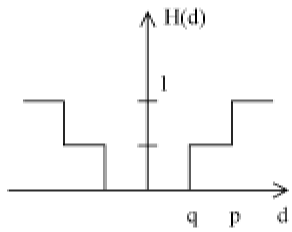
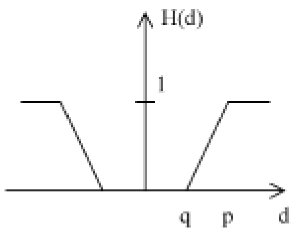
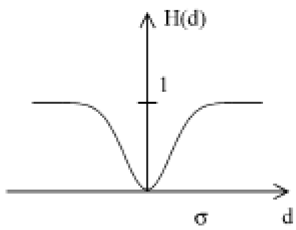
Alternatīvu sakārtojuma iegūšanai ir jāizrēķina katras alternatīvas izejošā un ienākošā plūsma. Tās ir definētas šādā veidā:

Izejošā plūsma: $\phi^{+}(a) = \sum_{b \in K} \Pi(a,b)$

Ienākoša plūsma: $\phi^{-}(a) = \sum_{b \in K} \Pi(b,a)$

2.7. tabula

PROMETHEE metodes kritēriju tipi

N.p.k.	**Definīcija**	**Grafiks**
1.	*Parastais kritērijs*: $H(d)=\begin{cases}0, & d=0\\ 1, & d\neq 0\end{cases}.$	
2.	*Kvazi-kritērijs*: $$H(d)=\begin{cases}0, & \lvert d\rvert\le q\\ 1, & \lvert d\rvert>q\end{cases},$$ kur q ir sliekšņa parametrs.	
3.	*Preferences kritērijs*: $$H(d)=\begin{cases}\frac{\lvert d\rvert}{p}, & \lvert d\rvert\le p\\ 1, & \lvert d\rvert>p\end{cases},$$ kur p ir sliekšņa parametrs.	
4.	*Līmeņa kritērijs*: $$H(d)=\begin{cases}0, & \lvert d\rvert\le q\\ \frac{1}{2}, & q<\lvert d\rvert\le p\\ 1, & p<\lvert d\rvert\end{cases},$$ kur q ir vienaldzības slieksnis un p ir preferences slieksnis.	
5.	*Kritērijs ar lineāru preferenci un vienaldzības zonu*: $$H(d)=\begin{cases}0, & \lvert d\rvert\le q\\ (\lvert d\rvert-q)/(p-q), & q<\lvert d\rvert\le p\\ 1, & p<\lvert d\rvert\end{cases},$$ kur q ir vienaldzības slieksnis un p ir preferences slieksnis.	
6.	*Gausa kritērijs*: $$H(d)=1-\exp(-d^2/2\sigma^2),$$ šī kritērija izmantošanai ir jānosaka parametra σ vērtība.	

Nodefinēsim preferences un vienaldzības attiecības:

$$\begin{array}{ll} aP^{+}b & \text{tad un tikai tad, ja } \phi^{+}(a) > \phi^{+}(b) \\ aP^{-}b & \text{tad un tikai tad, ja } \phi^{-}(a) < \phi^{-}(b) \\ aI^{+}b & \text{tad un tikai tad, ja } \phi^{+}(a) = \phi^{+}(b) \\ aI^{-}b & \text{tad un tikai tad, ja } \phi^{-}(a) = \phi^{-}(b) \end{array}$$

Daļējs sakārtojums tiek definēts ar *(P_I, I_I, R)* attiecību palīdzību, kuras ir nodefinētas šādā veidā:

$$\begin{cases} aP_I b, & \text{ja } (aP^{+}b \wedge aP^{-}b) \vee (aP^{+}b \wedge aI^{-}b) \vee (aI^{+}b \wedge aP^{-}b), \\ aI_I b, & \text{ja } (aI^{+}b \wedge aI^{-}b), \\ aRb, & \text{pretējā gadījumā.} \end{cases}$$

P ir preferences attiecība, *I* ir vienaldzības attiecība un *R* ir nesalīdzinabības attiecība.

Pilns sakārtojums tiek definēts sekojošajā veidā:

$$\phi(a) = \phi^{+}(a) - \phi^{-}(a),$$

$$\begin{cases} aP_{II}b, & \text{ja } \phi(a) > \phi(b), \\ aI_{II}b, & \text{ja } \phi(a) = \phi(b). \end{cases}$$

Kā ir redzams, pilnajā sakārtojumā tiek pazusta informācija par nesalīdzināmām alternatīvām. Tātad, ar daļējā sakārtojumu palīdzību var iegūt vairāk informācijas, jo tas ļauj uzzināt kuras no alternatīvām ir nesalīdzināmas. Pilnais sakārtojums var būt maldinošs, jo tajā tiek izmantota neto plūsma. Tomēr, pieņemot lēmumu ir jāizmanto visa pieejamā informācija.

Viens no PROMETHEE metodes trūkumiem ir tas, ka apriori ir jābūt zināmai precīzai informācijai par visu kritēriju vērtībām visām alternatīvām. Tomēr, ja lēmums tiek pieņemts nenoteiktības apstākļos, tad daļa no šīs informācijas var būt nepieejama. Daļēji šo problēmu var atrisināt ar jūtīguma analīzes palīdzību, apskatot modeļa uzvedību ar dažādiem parametriem. Tomēr, jūtīguma analīzes laikā tiek mainīts pats modelis, bet iekļaut nenoteiktību modelī nav iespējams.

2.1.4. ELECTRE lēmumu pieņemšanas metode

Līdzīgi PROMETHEE metodei, ELECTRE metodes pamatā ir tabula ar kritēriju vērtībām un beigās tiek iegūts alternatīvu sakārtojums [Roy, 1991]. ELECTRE metodē ir nodefinēts tā sauktais vienaldzības slieksnis *q*, kuram ir šāda nozīme:

$$aPb \Leftrightarrow f(a) > f(b) + q$$
$$aIb \Leftrightarrow |f(a) - f(b)| \leq q$$

Sakārtojums tiek iegūts ar *S* pārspēšanas (*outranking* – angļu val.) attiecības palīdzību, tā tiek rēķināta izmantojot konkordances un diskordances indeksus. Kritēriju *aIb* kopu sadalīsim šādā veidā:

I^+ - alternatīvas A novērtējums ir labāks nekā alternatīvas B novērtējums,

$I^=$ - alternatīvas A un B novērtējums ir vienāds,

I^- - alternatīvas B novērtējums ir labāks nekā alternatīvas A novērtējums.

Turklāt, pieņemsim, ka katram kritērijam ir noteikts svars w_i. Tad konkordances indekss tiek rēķināts šādā veidā:

$$C(A,B) = \frac{\sum_{i \in I^+, I^=} w_i}{\sum_{i=1}^{N} w_i},$$

kur w_i ir *i*-tā kritērija svars.

Diskordances indekss tiek rēķināts šādā veidā:

$$d(A,B) = \max_{i \in I^-} (x_i^B - x_i^A) / m_i,$$

kur x_i^A un x_i^B ir alternatīvas A vai B *i*-tā kritērija novērtējums, m_i ir kritērija definēšanas apgabala diapazons.

Ievedīsim divus sliekšņa parametrus, viens no tiem ir *p* un tā vērtība ir tuvu vieniniekam un otrs ir *q*, tā vērtība ir tuvu nullei. Pārspēšanas attiecība tiek definēta šādā veidā:

$$S(A,B) \Leftrightarrow \left(C(A,B) \geq p\right) \wedge \left(d(A,B) \leq q\right)$$

Ja kādām alternatīvām šī attiecība nav spēkā, tad uzskatīsim, ka šīs alternatīvas ir nesalīdzināmas.

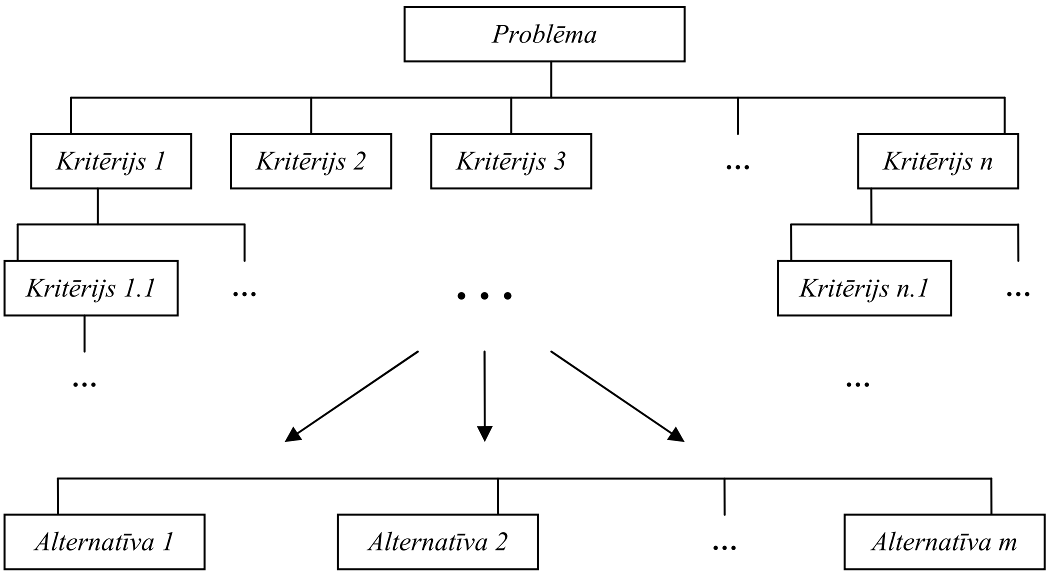
Līdzīgi PROMETHEE metodei, lai izmantotu ELECTRE metodi apriori ir jāzina kritēriju precīzās vērtības.

2.1.5. Analītiskais hierarhijas process un Swing metode

Abu metožu pamatā ir kritēriju svēršana, tomēr tās var izmantot arī lēmumu pieņemšanai.

AHP, jeb Analītiskā hierarhijas procesa pamatā ir kritēriju hierarhijas izveidošana un viena līmeņa kritēriju bināra salīdzināšana [Saaty, 1980]. Hierarhija ir koks, tā pirmajā līmeni ir mērķis, kurš ir jāsasniedz. Proti, pirmajā līmeni ir viens mezgls ar risināmās problēmas aprakstu. Otrajā līmenī ir kritēriji, pēc kuriem tiek vērtētas alternatīvas. Trešajā līmeni ir šo kritēriju apakškritēriji, ceturtajā līmenī ir apakškritēriju apakšapakškritēriji un tā tālāk līdz tiks sasniegta nepieciešama detalizācija. It svarīgi, lai katrā līmenī būtu kritēriji ar vienādu granulāciju, un lai tos varētu savā starpā salīdzināt. Katram apakšējā līmeņa mezglam tiek pievienoti mezgli ar pieejamām alternatīvam. AHP struktūra ir parādīta 2.2. attēlā.

Pēc hierarhijas izveidošanas ir jāsalīdzina katra līmeņa elementi. Salīdzinot alternatīvas un nosakot cik lielā mērā viena alternatīva ir labāka vai sliktāka par kādu citu alternatīvu var izmantot pieejamus datus, piemēram, alternatīvas cenu vai izmaksas. Salīdzinot alternatīvas un kritērijus vispirms ir jānosaka kāda alternatīva vai kritērijs ir svarīgāks un pēc tam – cik lielā mērā tas ir svarīgāks. Lai noteiktu priekšrokas *intensitāti* tiek izmantota skala no 1 līdz 9, kur 1 nozīmē vienādu priekšroku un 9 nozīmē ļoti stingru priekšroku (skala tiek izmantota gadījumā, ja nav pieejami skaitliskie dati, no kuriem būtu iespējams inducēt intensitāti). Pēc preferenču un to intensitātes noteikšanas tiek pārbaudīts vai tās nav pretrunīgas.



2.2. att. AHP procesa struktūra

Beigās rekursīvi, ņemot vērā noteiktas preferences un sākot no apakšējā līmeņa, katrai alternatīvai tiek izrēķināts novērtējums.

Viens no AHP galvenajiem trūkumiem ir tas, ka savstarpējo salīdzinājumu skaits ir kombinatorisks, jo katrā līmenī katrā mezglā savā starpā ir jāsalīdzina visi tā bērni. Tā kā salīdzinājumu skaits ir liels, palielinās varbūtība, ka salīdzinājumi būs pretrunīgi (piemēram, tranzitivitātes likums var nebūt spēkā). Proti, ja risināmai problēmai ir liels kritēriju un alternatīvu skaits, metode var kļūt nepielietojama praksē.

Tagad apskatīsim Swing metodi, jeb precīzāk – tās intervālu versiju, kura ļauj modelī iekļaut neprecizitāti [Mustajoki u.c., 2001]. Swing metode ir kritēriju svēršanas metode, kura var tikt izmantota arī lēmumu pieņemšanai. Rakstā [Mustajoki u.c., 2001] ir šīs metodes vispārinājums intervālu gadījumam. Ar šīs metodes palīdzību ir iespējams uzdot gan intervālu svarus, gan intervālu atribūtus. Tomēr, šī metode neļauj iekļaut modelī nedeterminētas vai izplūdušas vērtības.

Swing procedūras sākuma ir jāizvēlas viens balsta kritērijs, uz kura pamata tiks noteiktas citu kritēriju relatīvas svarīguma vērtības. Balsta kritērijam tiek dots noteikts punktu skaits, apzīmēsim to ar *ref*. Katram citam kritērijam tiek noteikts relatīvais svarīguma koeficients, kas var būt intervālu. Piemēram, ja kritērija *A* relatīvais svarīgums ir $[min_A; max_A]$, tad ir spēkā šāds svaru attiecību ierobežojums:

$$\frac{ref}{\max_A} \le \frac{w_{ref}}{w_A} \le \frac{ref}{\min_A},$$

kur w_{ref} un w_A ir balsta kritērija un kritērija *A* svari.

Pēc šādu ierobežojumu noteikšanas ar lineāras programmēšanas palīdzību ir jānosaka katra svara apakšējā un augšējā robeža:

$$\underline{v}(x) = \min \sum_{i=1}^{n} w_i \underline{v}_i(x), \tag{2.1}$$

$$\overline{v}(x) = \max \sum_{i=1}^{n} w_i \overline{v}_i(x), \tag{2.2}$$

kur $\underline{v}_i(x)$ un $\overline{v}_i(x)$ ir alternatīvas atribūta (kritērija novērtējuma) $v_i(x)$ mazākā un lielākā vērtība.

Uzdevumus (2.1) un (2.2) var atrisināt ar lineārās programmēšanas palīdzību [Salo&Hamalainen, 1995]. Pēc alternatīvu intervālu novērtējumu iegūšanas var noteikt kuras alternatīvas ir dominējošas un kuras ir nesalīdzināmas.

Swing ir laba metode kritēriju svaru noteikšanai. Proti, ar tās palīdzību var intuitīvi un ātri noteikt kritēriju vērtības, to ir ērti pielietot praksē. Pat ja ir ekspertu vai lēmumu pieņemošo personu grupa, un viņi nevar nonākt pie konsensa, var norādīt intervālu vērtības, kuras apmierinātu visas iesaistītās personas. Tomēr, šajā metodē maz uzmanības ir pievērsts kritēriju vērtību analīzei. Tiek atbalstītas kritēriju intervālu vērtības, bet nav iespējams izmantot nedeterminētas vai izplūdušas vērtības.

2.1.6. Uz lietderīguma teorijas balstīta daudzkritēriju lēmumu analīze

Lēmumu analīze ir uz lēmumu kokiem un lietderīguma teorijas balstītā formālā lēmumu pieņemšanas metode. Šī metode sastāv no dažiem soļiem [Clemen&Reilly, 2004; Turban u.c., 2004]:

- Pirmais solis ir lēmumu pieņemšanas situācija apzināšanās un mērķu noteikšana.
- Otrais solis ir alternatīvu noteikšana.
- Trešais solis ir problēmas modelēšana. Šajā solī ir jānomodelē problēmas struktūra, nenoteiktība un preferences.
- Ceturtais solis ir optimālas alternatīvas izvēle, saskaņā ar izveidoto lēmuma pieņemšanas modeli.
- Piektajā solī ir jāveic uzbūvētā modeļa jūtīguma analīze.
- Pēdējais solis ir rezultātu analīze un lemšana par to, vai ir jāveic papildus analīze.

Apskatīsim trešo soli, jo pārējie ir līdzīgi citu metožu soļiem. Ar lēmumu strukturēšanu tiek saprasta mērķu hierarhijas konstruēšana, nenoteiktības avotu noteikšana, alternatīvu noteikšana, dažādu entītiju savstarpējas sakarības noteikšana. Struktūras modelēšanai tiek izmantotas influenču diagrammas un lēmumu koki.

Nenoteiktība lēmumu analīzē tiek modelēta ar objektīvo vai subjektīvo varbūtību palīdzību. Turklāt, varbūtību sadalījuma noteikšanai var izmantot arī Montekarlo simulāciju.

Preferences lēmumu analīzē tiek modelētas ne tikai ar svaru palīdzību, bet arī modelējot attieksmi pret risku. Cilvēka attieksme pret risku var būt neitrāla, un tam var patikt vai nepatikt riskēt. To ir iespējams nomodelēt ar lietderīguma funkciju palīdzību. Ja cilvēkam nepatīk riskēt, tad paaugstinoties ieguldījumam, lietderīguma vērtība palielinās ne tik strauji. Proti, lietderīguma funkcija ir ieliekta. Ja cilvēkam patīk riskēt, tad viņa lietderīguma funkcija ir izliekta.

Lēmumu analīze ir formāli pamatota metode, tā nav tik heiristiskā kā 2.1.3. un 2.1.4. nodaļās apskatītās metodes. Tomēr, ja ir daudz kritēriju, tad lietderīguma funkciju noteikšana var būt praktiski nepielietojama. Jo katram kritērijam ar vairāku jautājumu palīdzību būs jānosaka lēmumu pieņemošas personas attieksme pret risku un dažādu kritēriju vērtību lietderīgumu. Tas pats attiecās uz svariem, kuri attēlo kompromisa koeficientus un vairāku kritēriju gadījumā to noteikšana var būt sarežģīta un darbietilpīga.

2.1.7. Lēmumu pieņemšanas modeļu jūtīguma analīze

Lēmumu modeļu jūtīguma analīze tiek saukta arī par robustuma analīzi, jo ar tās palīdzību var noteikt cik stabils ir lēmumu modelis un cik liela ietekme var būt modeļa atribūtu vērtību izmaiņai. Jūtīguma analīzes laikā tiek mainītas lēmumu modeļa parametri, piemēram, varbūtību vērtības, kritēriju vērtības, kritēriju svaru vērtības un tiek analizēts kā šīs izmaiņas ietekmē lēmumu vai alternatīvu sakārtojumu. Ar šīs analīzes palīdzību var noteikt vai pieņemamais lēmums ir stabils, proti, ir jābūt ievērojamām parametru izmaiņām, lai cita alternatīva kļūtu labāka, vai tas nav stabils un nelielas izmaiņas modeļa parametros noved pie cita risinājuma. It īpaši tas ir svarīgi pieņemot lēmumus nenoteiktības apstākļos.

Praksē tiek izmantoti divi jūtīguma analīzes veidi: vienvirziena un divvirzienu. Vienvirziena jūtīguma analīzē vienlaicīgi tiek mainīta tikai viena parametra vērtība, bet divvirzienu jūtīguma analīzē vienlaicīgi tiek mainītas divas vērtības. Šo analīžu rezultātus var attēlot ar tā sauktajām *tornado* diagrammām vai divdimensiju zīmējumiem [Clemen&Reilly, 2004]. Ja analizē vienlaicīgi tiek apskatīti vairāk nekā divi parametri, piemēram, izmantojot rakstā [Vališevskis, 2002] vai 3.1.1. paragrāfā aprakstīto modeli, tad grafiski attēlot rezultātus ir daudz sarežģītāk. Nākamajās apakšnodaļās sīkāk apskatīsim vienvirziena un divvirzienu jūtīguma analīzi.

2.1.7.1. Vienvirziena jūtīguma analīze un "tornado" diagrammas

Vienvirziena jūtīguma analīzes laikā vienlaicīgi tiek mainīta tikai viena parametra vērtība, pārējo parametru vērtības paliek fiksētas. Vienvirziena parametrus var ērti attēlot grafiskajā formā, izmantojot tā saukto "tornado" diagrammu. Diagrammas nosaukums radies no tās izskata.

Apskatīsim vienu piemēru. Pieņemsim, ka banka veido jaunu vērtspapīru fondu un cenšas noskaidrot vai fonds nesīs peļņu. Ar šo nolūku bankas eksperti veic vienvirziena jūtīguma analīzi un noskaidro cik liela ir katra parametra ietekme uz kopējo fonda ienesīgumu. Fonda parametri ir šādi:

1) Kontu skaits.
2) Peļņa par katru kontu.
3) Izdevumi par katru kontu.
4) Fiksētās izmaksas.

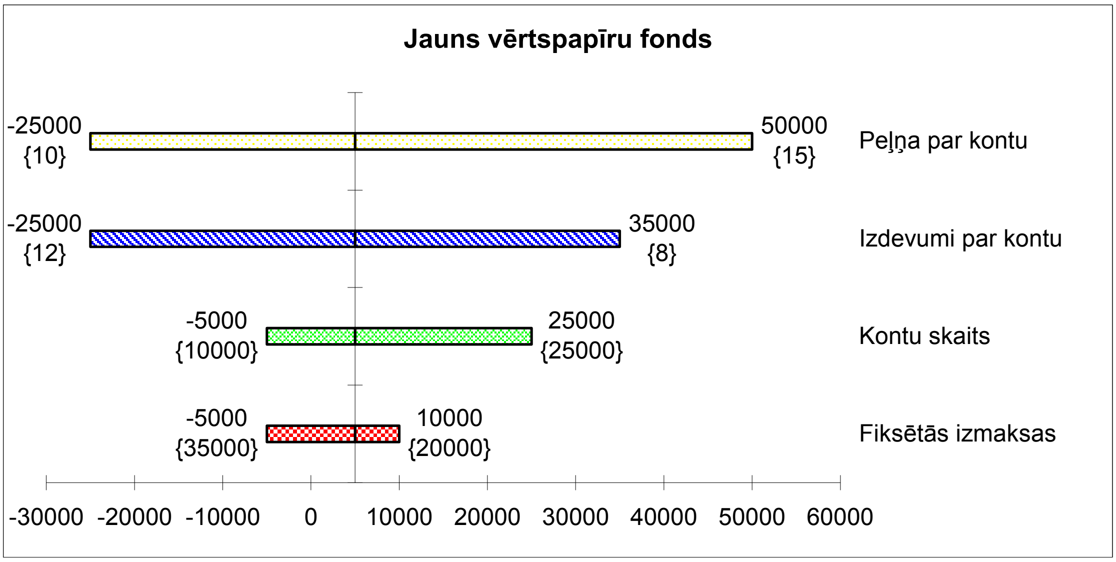
2.8. tabulā ir norādītas šo parametru minimālas, maksimālas un pamata vērtības. Pamata vērtība atbilst tornado diagrammas asij.

2.8. tabula

Vērtspapīru fonda parametru vērtības

Jauns vērtspapīru fonds	Pamata	Min	Max
Kontu skaits	15000	10000	25000
Peļņa par kontu	12	10	15
Izdevumi par kontu	10	8	12
Fiksētās izmaksas	25000	20000	35000

Izmantojot 2.8. tabulas datus var izveidot 2.3. attēlā parādīto tornado diagrammu, kas ir vienvirziena jūtīguma analīzes rezultāts. Katra diagrammas horizontālā josla atbilst vienam parametram un tā norāda cik lielā mērā šis parametrs ietekmē kopējo fonda peļņu. Joslas galos ir parādīta fonda peļņa, kad parametram ir figūriekavās esošā robežvērtība. Parametra vērtības tiek mainītas no minimālās līdz maksimālajai (minimālās un maksimālās vērtības ir norādītas figūriekavās). Kā tika minēts iepriekš, vertikālā ass atbilst parametru pamata vērtībām.

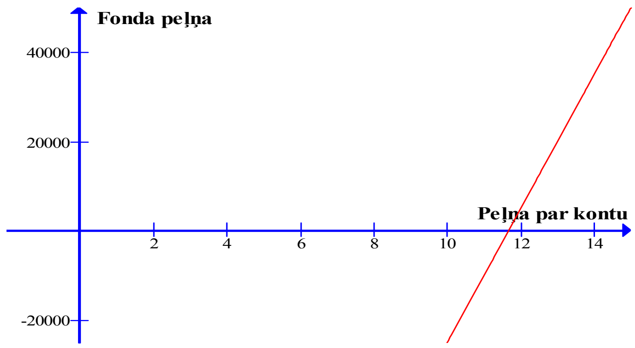


2.3. att. Tornado diagramma

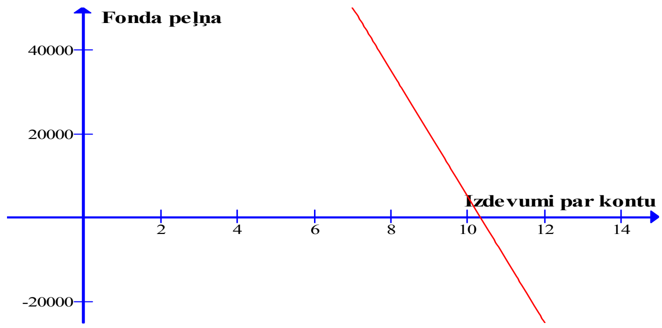
Katram parametram var uzzīmēt atsevišķu grafiku, kas parāda peļņas atkarību no atbilstošā parametra. Šie grafiki ir paradīti 2.4., 2.5., 2.6. un 2.7. attēlā.

Grafiku sastādīšanai tiek izmantota šāda funkcija:

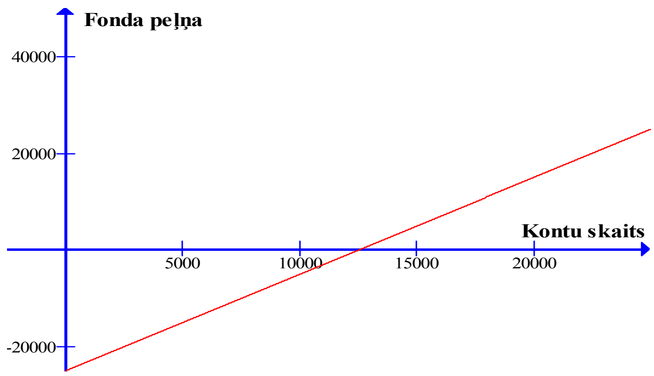
*Fonda peļņa = Kontu skaits * (Peļņa par kontu – Izdevumi par kontu) – Fiksētās izmaksas*



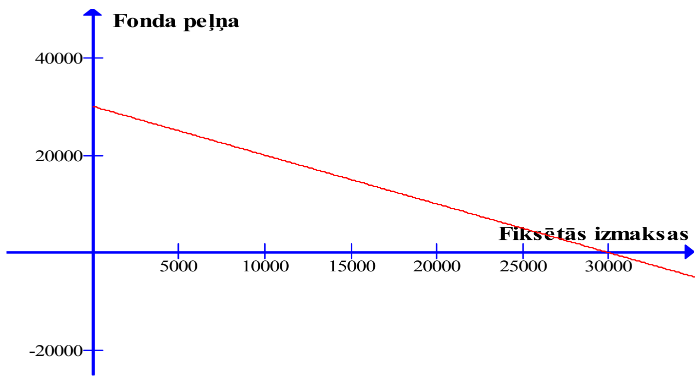
2.4. att. Fonda peļņas atkarība no parametra "Peļņa par kontu"



2.5. att. Fonda peļņas atkarība no parametra "Izdevumi par kontu"



2.6. att. Fonda peļņas atkarība no parametra "Kontu skaits"



2.7. att. Fonda peļņas atkarība no parametra "Fiksētās izmaksas"

Iegūtā diagrammu, kas ir parādīta 2.3. attēlā, tiek izmantota modeļa jūtīguma analīzei. Kā ir redzams, modelis ir visjutīgākais pret parametriem "*Peļņa par kontu*" un "*Izdevumi par kontu*". Tādējādi, pieņemot lēmumu šo parametru novērtēšanai ir jāpievērš vislielākā uzmanība, jo to labvēlīgo vērtību rezultātā peļņa var būtiski palielināties, bet nelabvēlīgo vērtību rezultāta zaudējumi var būt tik pat ievērojami. Dažos gadījumos, ja risināmā problēma to pieļauj, var samazināt zaudējumu risku, izmantojot atbilstošo risku pārvaldes metodi, kas ļautu samazināt riskus, apdrošināties pret nelabvēlīgām situācijām. Šajā nodaļā apskatīta uzdevuma gadījumā bankas klientiem var piedāvāt dinamisku konta ienesīgumu, kas ļaus samazināt izdevumus par kontu, ja situāciju vērtspapīru tirgū būs nelabvēlīga. Modeļa jūtīgums pret parametriem "*Kontu skaits*" un "*Fiksētās izmaksas*" nav tik liels, bet ir jāatceras, ka arī šo parametru nelabvēlīgo vērtību dēļ var rasties zaudējumi.

Nākamajā apakšnodaļā apskatīsim gadījumu, kad vienlaicīgi tiek mainītas divu parametru vērtības. Grafiski ir sarežģīti attēlot gadījumu, kad vienlaicīgi tiek mainīti vairāk par diviem parametriem, tātad tradicionāli neapskata jūtīguma analīzi, kad tiek mainīts vairāk par diviem parametriem. Savukārt, izmantojot jūtīguma analīzei disertācijā izstrādāto adaptīvo tīklu ANGIE (sk. 3.1.1. paragrāfu) analīzes laikā tiek apskatīti visi parametri.

2.1.7.2. Divvirzienu jūtīguma analīze

Divvirzienu jūtīguma analīzes laikā vienlaicīgi tiek mainītas divu parametru vērtības. Rezultāts tiek attēlots grafiski kā plakne ar vienu vai vairākām līnijām, kas to sadala vairākos apgabalos, atkarībā no kritērija vērtības.

Pieņemsim, aviokompānija vēlas nopirkt jaunu nelielu 10-vietīgu lidmašīnu un vēlas novērtēt kā lidmašīnas piepildījums un kārtējās izmaksas ietekmē no šīs lidmašīnas izmantošanas iegūtu peļņu. Kompānija vēlas noskaidrot kādām jābūt šo parametru vērtībām, lai peļņa būtu vienāda vismaz ar $4200.

Lai atrisinātu šo uzdevumu ir jāatrisina sekojoša nevienādība:

$$\begin{gathered}\textit{Čartera lidojumu proporcija * Lidojumu stundu skaits * Čartera lidojumu cena +}\\ \textit{+ (1 - Čartera lidojumu proporcija) * Lidojumu stundu skaits * Biļetes cena * Vietu skaits *}\\ \textit{* Lidmašīnas piepildījums < (Lidojumu stundu skaits * Kārtējas izmaksas) +}\\ \textit{+ Apdrošināšana + (Lidmašīnas cena * Bankas nofinansēta daļa *}\\ \textit{* Kredīta procentu likme) + 4200}\end{gathered} \tag{2.3}$$

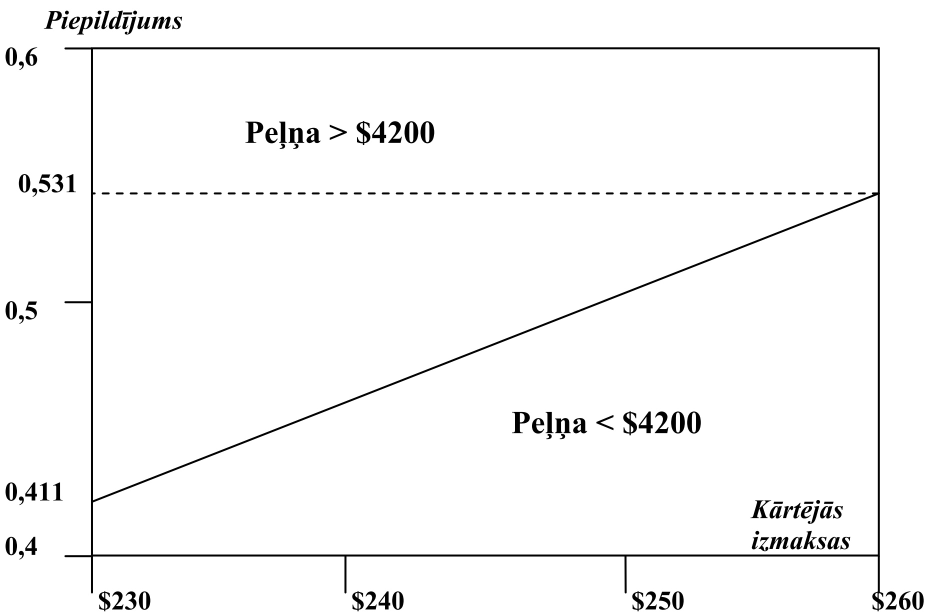
Uzskatīsim, ka visu parametru vērtības ir zināmas, izņemot lidmašīnas piepildījumu un kārtējās izmaksas. Ievietojot izteiksmē (2.3) zināmās vērtības mēs iegūstam:

$$0{,}5 * 800 * 325 + 0{,}5 * 800 * 100 * 5 * \mathit{Piepildījums} <$$
$$< 800 * \mathit{Kārtējās\ izmaksas} + 20000 + 87500 * 0{,}4 * 0{,}115 + 4200$$

To nav grūti atrisināt, rezultātā mēs iegūstam šādu nevienādību:

$$\mathit{Piepildījums} < 0{,}004 * \mathit{Kārtējās\ izmaksas} – 0{,}509$$

Apskatot gadījumu, kad piepildījums ir no 0,4 līdz 0,6 un kārtējās izmaksas ir no $230 līdz $260 var izveidot 2.8. attēlā parādīto divvirzienu jūtīguma analīzes grafiku.



2.8. att. Divvirzienu jūtīguma analīze

Izanalizēsim 2.8. attēlā parādīto grafu. Tajā ir redzama peļņas atkarība uzreiz no diviem parametriem: no "*Lidmašīna piepildījuma*" un "*Kārtējām izmaksām*". No grafa mēs redzam kādam jābūt lidmašīnas piepildījumam katrai kārtējo izmaksu vērtībai, lai peļņa būtu vēlamajā līmenī. Un otrādi, tajā ir redzams kādai jābūt kārtējo izmaksu vērtībai katram konkrētam lidmašīnas piepildījumam, lai peļņa būtu vienāda vismaz ar $4200.

3.1.1. paragrāfā tiek apskatīta metode, kura ļauj vienlaicīgi analizēt vairāk par diviem parametriem, turklāt, izmantojot šo metodi var noteikt parametrus, kurus var turpmāk analizēt izmantojot vienvirziena un divvirzienu jūtīguma analīzes metodes.

2.2. Lēmumu pieņemšanas metodes vidēm ar vairākiem nenoteiktības avotiem

Šajā apakšnodaļā ir aprakstītas lēmumu pieņemšanas metodes, kuras ir paredzētas vidēm ar vairākiem nenoteiktības avotiem. Vispirms tiek apskatīta izplūdušo kopu un varbūtību blīvuma funkcijas agregācija, kas balstīta uz stohastiskās dominances un pēc tam tiek apskatīta PRUF metode dabiskās valodas izteicienu pārveidošanai mainīgo izplūdušajos ierobežojumus, kas ļauj tos analizēt.

2.2.1. Uz stohastiskās dominēšanas balstīta sajaukto novērtējumu agregācija

Rakstā [Ben Amor u.c., 2004] ir aprakstīta lēmumu pieņemšanas metode, kura ļauj problēmas aprakstā vienlaicīgi izmantot gan precīzo, gan nedeterminēto, gan izplūdušo informāciju. Metode ir balstīta uz stohastiskās dominēšanas. Vispirms problēma tiek aprakstīta ar tabulas palīdzību, līdzīgi PROMETHEE un ELECTRE metodēm. Tomēr, šajā metodē tabulā var būt ne tikai precīzas vērtības, bet arī varbūtību sadalījumi un izplūdušas vērtības.

Pirmajā solī alternatīvas tiek salīdzinātas savā starpā ar stohastiskās dominances palīdzību. Otrajā solī šī lokālā salīdzinājuma rezultāti tiek izmantoti, lai uzkonstruētu globālu preferences attiecību. Tā kā alternatīvu salīdzināšanai tiek izmantota stohastiskā dominance, visas vērtības tiek pārveidotas varbūtību blīvuma funkcijās. Tas attiecās arī uz izplūdušajām vērtībām. Tomēr, nedeterminētām vērtībām un izplūdušām vērtībām ir dažāda daba [Kosko, 1992] un pārveidojot vienu vērtību citā formā var pazaudēt informāciju. Tādējādi, šīs metodes agregācija balstās uz dažāda tipa vērtību pārveidošanas varbūtību blīvuma funkcijās. Labāks un dabiskāks risinājums būtu tāda informācijas apraksta līdzekļa izstrāde, kurš ļautu izmantot dažāda tipa datus (nedeterminētus, izplūdušus) un ļautu šos datus salīdzināt savā starpā bez papildus pārveidošanas, jeb izveidojot universālo formu, kurā būtu iespējams pārveidot visa veida nenoteiktības, kā ir izdarīts PRUF [Zadeh, 1978] un CW [Zadeh, 1996] metodēs.

Ar šīs metodes palīdzību dažādām alternatīvām vienu un to pašu kritēriju var aprakstīt ar nedeterminētu, izplūdušu vai precīzu vērtību palīdzību, kas neskatoties uz metodes trūkumiem ir būtisks solis uz priekšu, salīdzinājumā ar tādām metodēm kā ELECTRE vai PROMETHEE (aprakstīšanas spējas ziņā).

Šīs metodes shēma ir tāda pati kā ELECTRE metodei izņemot to, ka alternatīvu sakārtojums tiek iegūts izmantojot stohastiskās dominances metodi, kura balstās uz kritēriju tabulas vērtībām.

2.2.2. Uz iespējamības teoriju balstīta metode dabiskās valodas aprakstīšanai

PRUF ir metode dabiskās valodas izteiksmju pārveidošanai un aprakstīšanai ar izplūdušās loģikas palīdzību [Zadeh, 1978].

PRUF metodē dabiskās valodas izteiksmes tiek aprakstītas kā elastīgie ierobežojumi uz izplūdušo mainīgo vērtībām jeb kā iespējamības sadalījumi. Turklāt, kā jau tika minēts, šīs metodes pamatā ir izplūdusī loģika, kas ļauj izmantot izplūdušas patiesuma vērtības, piemēram, patiess, nepatiess, ļoti patiess, ne īpaši patiess utt. Arī kvantori var būt izplūduši, piemēram, var izmantot tādus kvantorus kā lielākā daļa, daudz, daži, ne īpaši daudz, gandrīz visi utt.

PRUF metodē ir četras izteiksmju pārveidošanas likumu grupas. Pirmajā grupā ir modifikācijas likumi. Ar to palīdzību ir iespējams uzlikt jaunus ierobežojumus uz izplūdušā mainīga. Piemēram, tie der tādu izteiksmju kā "X vērtība ir ļoti maza" pārveidošanai. Otrajā grupā ir kompozīcijas likumi. Ar to palīdzību var apvienot dažādu izplūdušo mainīgo aprakstus. Piemēram "X vērtība nav īpaši liela un Y vērtība ir ļoti liela". Trešajā grupā ir likumi, kuri tiek izmantoti kvantoru ieviešanai. Piemēram, "Lielākai daļai Latvijas iedzīvotāju ir tumši mati". Ceturtajā grupā ir kvalifikācijas likumi. Tie tiek izmantoti patiesuma, varbūtības vai iespējamības kvalifikācijai. Piemēram, "ir ļoti iespējams, ka Māra ir jauna".

Pēc dabiskās valodas izteiksmju aprakstīšanas un lingvistisko mainīgo ierobežojumu uzdošanas ir iespējams noteikt kāda ir šī mainīgā vērtība, vai kāda ir iespējamība, ka šim mainīgajam ir kāda vērtība. Lai atbildētu uz šo jautājumu tiek izmantots izplūdušais izvedums.

Ar PRUF metodes palīdzību var aprakstīt dažāda veida nenoteiktību, kura tiek aprakstīta kā elastīgie ierobežojumi uz izplūdušā mainīgā vērtībām. Rēķināšanas laikā netiek izmantotas nekādas konvertācijas, tādēļ, analīzes laikā informācija netiek zusta. Tomēr, lai iegūtu atbildi uz uzstādāmiem jautājumiem par mainīgā vērtībām, ir jāizmanto metodes, kuras prasa daudz skaitļošanas resursu. Tādēļ šo metodi nav iespējams izmantot praksē, it īpaši tādos uzdevumos, kuros ir jāapstrādā daudz nenoteiktas informācijas.

PRUF metodi var izmantot, lai pārveidotu un analizētu izteiksmes, kuru piemēri ir attēloti zemāk:

Ronalds ir daudz maz jauns
Miriama bija ļoti bagāta
Lielākai daļai zviedru ir blondi mati
Nav ļoti patiesi, ka lielāka daļa zviedru ir gari
Suzanna uzdāvināja dažas dārgas dāvanas katram no saviem tuvajiem draugiem

Kā redzams, PRUF analīzē var iekļaut dažāda veida nenoteiktības – izplūdušās vērtības, izplūdušos kvantorus, relācijas, varbūtības, iespējamību.

Par izplūdušo kopu deskriptoriem tiek saukti izplūdušo kopu nosaukumi, kuri atšķir doto izplūdušo kopu no citām izplūdušām kopām. Daži no piemēriem ir šādi:

Ļoti liela auguma cilvēks
Ļoti liela auguma cilvēks ar brūnu cepuri
Neliels skaitlis
Visi
Lielāka daļa
Daudz augstāks par

Atšķirībā no izteiksmes, deskriptors tiek pārveidots relācijā nevis iespējamības sadalījumā.

Viens no svarīgākajiem jēdzieniem PRUF metodē ir jautājumi, jo tie tiek izmantoti, lai noteiktu dabiskās valodas izteiksmju pārveidojumu virzienu un līdz ar to arī visas analīzes gaitu. No izteiksmēm jautājumi atšķiras ar to, ka tajos ir nezināmā vērtība. Tādējādi, jautājumi tiek pārveidoti izteiksmēs ar nezināmām vērtībām, kuras tiek noteiktas analīzes gaitā. Daži no jautājumu pārveidojumu piemēriem ir šādi (nezināmo vērtību priekšā ir jautājuma zīme):

Kāds ir Toma augums ?T → Toma augums ir ?T

Kur dzīvo Toms → Toms dzīvo ?a

Vai ir patiess, ka Frankam ir blondi mati → Frankam ir blondi mati ir ?t

Pēc zināšanu aprakstīšanas izmantojot izteiksmes dabiskā valodā, tās ir jāpārveido izmantojot rakstā [Zadeh, 1978] aprakstītos likumus. Pēc izteiksmju pārveidošanas ir jānosaka jautājuma nezināmā vērtība. Lai to izdarītu tiek izmantotas izplūdušā izveduma procedūras un likumi, daži no kuriem ir aprakstīti tajā pašā rakstā.

Tomēr, šīs procedūras ir sarežģītās tajā ziņā, ka tos ir grūti efektīvi realizēt datora programmas veidā. Kā vairākkārt norādīja prof. L. Zadē, šo metožu efektīva izmantošana būs iespējama tikai līdz ar izplūdušās skaitļošanas aparatūras parādīšanos.

Par rakstā [Zadeh, 1978] uzsāktā pētījuma turpinājumu var uzskatīt skaitļošanu ar vārdiem, kura ir piedāvāta rakstā [Zadeh, 1996]. Tā ir jauna skaitļošanas paradigma, kura balstās uz izplūdušo loģiku. Kā norāda pats autors, šī metode ir jāizmanto, kad informācijas nenoteiktība neļauj izmantot precīzas vērtības, kad precīzs modelis prasa pārāk lielus ieguldījumus, vai kad izplūdusi skaitļošana ir nepieciešama, lai nodrošinātu rezultāta stabilitāti un drošumu.

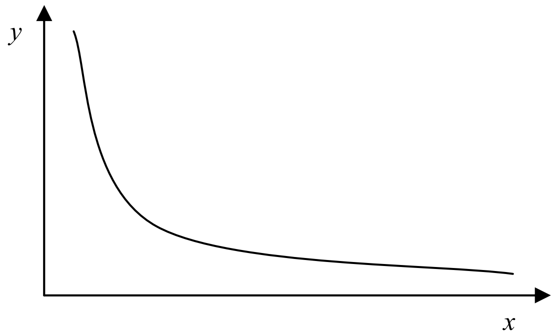
Viens no skaitļošanas ar vārdiem galvenajiem jēdzieniem ir *granula*, kura tiek definēta kā līdzīgo izplūdušo punktu kopa. Turklāt, tā definē ierobežojumu uz izplūdušā mainīgā vērtībām un visa metode balstās uz šo ierobežojumu pārveidošanu ar likumu palīdzību.

Problēma tiek aprakstīta dabiskā valodā ar tā sauktās sākuma datu kopas (SDK) palīdzību. Izmantojot izplūdušā izveduma metodes no SDK ir jāsaņem atbilde uz mūs interesējošo jautājumu.

Šajā metodē tiek izmantots arī grafika izplūdušā apraksta jēdziens [Ishibuchi u.c., 2005], kas ir izplūdis funkcijas apraksts, proti, tā ir izplūdusī relācija, kura ir uzdota šādā veidā:

Ja y ir mazs, tad x ir ļoti liels
Ja y ir vidējs, tad x ir liels
Ja y ir liels, tad x ir mazs

Šīs izplūdušais apraksts atbilst 2.9. attēlā parādītajai funkcijai [Zadeh, 1996].



2.9. att. Funkcijas grafiks, kurai ir dots izplūdušais apraksts

Lai paplašinātu dabiskās valodas izteiksmju tipus, kurus ir iespējams izmantot šajā metodē, tika ievesta tā sauktā "isr" vispārinātā relācija, kura vispārīgā gadījumā izskatās šādi:

X isr R

Relācija "isr" nosaka kādā veidā R ierobežo X vērtības, kas ir atkarīgs no burta "r" vērtības (sk. 2.9. tabulu).

2.9. tabula

Vispārinātās relācijas burta "r" vērtības

Burts "r"	**Nozīme**
e:	vienāds
d:	disjunkcija
c:	konjunkcija
p:	varbūtības sadalījums
l:	varbūtība
u:	parastums (*usuality*)
rs:	nejauša kopa
rsf:	nejauša izplūdusī kopa
fg:	izplūdušais grafs
ps:	rupja kopa (Pavlaka kopa)

Piemēram, ja mums ir izteiksme šādā formā:

p = Džons pārvalda angļu, franču un vācu valodu,

tad to var pārveidot izmantojot konjunktīvo attiecību, piemēram, šādi:

ValoduZināšana(Džons) isc (*1,0*/Angļu + *0,7*/Franču + *0,6*/Vācu),

kur *1,0* , *0,7* un *0,6* vērtības raksturo cik labi Džons prot minētās valodas.

Zemāk ir parādīts kā var uzdot dažāda veida nenoteiktībai atbilstošus ierobežojumus.

Varbūtisko ierobežojumu var uzdot šādi:

X isp N(m, σ^2)

Y isp (*0,2*\a + *0,5*\b + *0,3*\c)

Ierobežojums "X isu R" atbilst šādai izteiksmei:

usually(X is R),

savukārt, tas nozīmē:

Prob{X is R} is usually

Pēc šo pārveidojumu veikšanas izplūdušie ierobežojumi ir jāpārnes uz nezināmā mainīgā vērtību, lai noteiktu tā vērtību. To var paveikt ar likumu palīdzību, kuri ir aprakstīti rakstā [Zadeh, 1996].

Profesors Zadē norāda, ka šīs metodes efektīvai izmantošanai ir jāizmanto izplūdušā skaitļošanas aparatūra vai kā vienu no iespējamajiem risinājumiem viņš piedāvā izmantot ģenētiskos algoritmus, jo tie ir īpaši piemēroti šādu optimizācijas uzdevumu risināšanai.

2.3. Secinājumi par 2. nodaļu

Otrajā nodaļā ir apskatītas vairākas lēmumu pieņemšanas metodes. Ir parādīti galvenie metožu trūkumi, kuri zemāk ir aprakstītas sīkāk. Analizējamo metožu trūkumi pārsvarā ir saistīti ar to, ka metodi nav iespējams izmantot nenoteiktajā vidē, vai ar to, ka tā prasa daudz skaitļošanas resursu.

1. Lēmumu pieņemšanas metodes PROMETHEE un ELECTRE ir heiristiskas metodes, kuras ir pietuvinātas cilvēku intuitīvajam domāšanas veidam pieņemot lēmumus. Tās var izmantot uzdevumos ar grūti formalizējamām attiecībām starp kritērijiem.
2. AHP un Swing metodes var izmantot kritēriju svēršanai, tomēr AHP metode ir ietilpīga skaitļošanas ziņā.
3. Lēmumu analīze ir formālā metode, kas balstīta uz lietderīguma teorijas un varbūtību teorijas. To var izmantot gadījumos, kad ir jāizveido formālais lēmuma modelis un ja ir tikai varbūtiska nenoteiktība.
4. Uz stohastiskās dominances balstītai metodei izplūdušo un varbūtisko novērtējumu agregācijai ir viens trūkums – tajā visi novērtējumi tiek pārveidoti varbūtību blīvuma funkcijās, kas nesaskan ar intuīciju, jo izplūdums un nejaušība ir dažādi nenoteiktības veidi.
5. Uz iespējamības teorijas balstīta metode un skaitļošana ar vārdiem alternatīvu aprakstā ļauj vienlaicīgi izmantot dažāda veida nenoteiktības, tomēr to autors norāda uz to, ka skaitļošanas ziņā tie ir pārāk ietilpīgie.
6. Vidēs ar vairākiem nenoteiktības avotiem izmantošanai paredzētas lēmumu pieņemšanas metodes ir viens no modernākajiem lēmumu zinātnes attīstības virzieniem. Lai precīzi aprakstītu vairākas problēmas ir jāizmanto gan izplūdusī, gan nedeterminēta informācija.

3. UZ GRANULĀRAS INFORMĀCIJAS BALSTĪTĀS LĒMUMU PIEŅEMŠANAS METODES IZSTRĀDE

Šajā nodaļā ir aprakstīta izstrādātā lēmumu pieņemšanas metode, kas ir balstīta uz izplūdušajām granulām. Vispirms tiek aprakstīti izstrādātie palīglīdzekļi, kuri tiek izmantoti metodē. Pirmais līdzeklis ir adaptīvais tīkls ANGIE (*Adaptive Network for Granular Information and Evidence Processing*) ar kura palīdzību var analizēt izplūdušās granulas ar nolūku gūt informāciju par tajās esošo informāciju. Turklāt, adaptīvā tīkla ANGIE apmācīšanas algoritmu var izmantot lēmumu modeļa jūtīguma analīzei. Otrais palīglīdzeklis ir intervālu entropija, kas ir Šenona entropijas vispārinājums gadījumam, kad varbūtības ir intervālu. To izmanto katras alternatīvas informatīvuma novērtēšanai [Vališevskis, 2003].

Palīglīdzekļu aprakstam seko izstrādātās lēmumu pieņemšanas metodes apraksts.

3.1. Piedāvātajā lēmumu pieņemšanas metodē izmantojamo līdzekļu izstrāde

Pirms lēmumu pieņemšanas metodes aprakstīšanas, nodefinēsim palīglīdzekļus, kuri ir izstrādāti pētījumu gaitā un kuri tiek izmantoti izstrādātajā metodē. Proti, šajā apakšnodaļā ir aprakstīts adaptīvais tīkls ANGIE, tā apmācīšanas algoritms un intervālu gadījumam vispārināta Šenona entropija.

Tā kā izplūdušu granulu rēķini prasa daudz laika un skaitļošanas resursu, ir izstrādāts adaptīvais tīkls ANGIE (*Adaptive Network for Granular Information and Evidence Processing* – angļu val.), ar kura palīdzību ir iespējams noteikt tikko pieminētas varbūtības apakšējo un augšējo robežu [Vališevskis&Borisov, 2002]. Šīs pieejas priekšrocība ir tāda, ka izmantojot rēķiniem tīklu, kurā rēķini notiek paralēli, ir iespējams paātrināt skaitļošanas ātrumu. Protams, ātrums palielināsies tikai tajā gadījumā, ja šis tīkls tiks realizēts ar aparatūras nodrošinājuma palīdzību, kas atbalsta paralēlus rēķinus.

Tā kā izstrādātais tīkls ir adaptīvs, ir iespējams veikt tā apmācīšanu, jeb, citiem vārdiem, tā parametru automatizēto regulēšanu. Tātad, ir jāizstrādā sistēmas *ANGIE* apmācīšanai paredzēts algoritms. Tāpat kā neironu tīklu apmācīšanas algoritmi, sistēmas *ANGIE* apmācīšanai paredzēts algoritms ir balstīts uz gradienta lejupslīdes metodes.

Tīklā veikto rēķinu specifikas dēļ ir iespējams veikt tikai varbūtību vērtību regulēšanu. Tā kā algoritms veic tikai varbūtību vērtību regulēšanu un neņem vērā to konsistenci, lai to summa būtu vienāda ar vieninieku, ir jāveic varbūtību vērtību normēšana. Ir iespējami trīs varianti: 1) veikt normēšanu pēc katras iterācijas; 2) veikt normēšanu pēc katras *n*-tās iterācijas un 3) veikt normēšanu pēc sistēmas apmācīšanas. Analīzes gaitā ir parādīts, ka šajā gadījumā vispiemērotākais ir 3. variants [Vališevskis&Borisov, 2003].

Sistēmas *ANGIE* apmācīšanas rezultātā var noskaidrot kurš no parametriem un cik lielā mērā ietekmē to, ka kritērijs pieņem vēlamo vērtību. Kā ir parādīts tālāk, šīs uzdevums ir saistīts ar lēmumu pieņemšanas modeļa jūtīguma analīzes uzdevumu. Adaptīvā tīkla ANGIE arhitektūra un apmācības algoritms ir apskatīti 3.1.1. paragrafā.

3.1.1. Adaptīvais tīkls ANGIE un algoritms tā apmācībai

Izplūdušo granulu apstrādei ir izstrādāts adaptīvais tīkls ANGIE, kas ir līdzīgs 1.3.2. paragrāfā apskatītajai sistēmai ANFIS, kura ir balstīta uz adaptīvā tīkla, kas veic izplūdušo loģisko izvedumu. Sistēmas ANFIS galvenā priekšrocība ir tas, ka tā ir *adaptīva*. Proti, ir iespējams veikt sistēmas parametru automatizēto regulēšanu (citiem vārdiem, sistēmu ir iespējams *apmācīt*). Sistēmas ANFIS apmācīšanas algoritms ir līdzīgs neironu tīklu apmācīšanas algoritmam [Rumelhart u.c., 1986] un ir balstīts uz gradienta lejupslīdes metodes.

Šajā nodaļa tiek piedāvāta uz adaptīva tīkla balstīta lēmumu pieņemšanas sistēma *ANGIE* [Vališevskis&Borisov, 2002]. Sistēma ANGIE ir paredzēta izplūdušo liecību apstrādei. Izplūdušo liecību galvenie aspekti tika apskatīti 1.3.3.1. paragrāfā. Turklāt, tur tiek aprakstīts sistēmas ANGIE apmācīšanai paredzēts algoritms. Līdzīgi sistēmas ANFIS apmācīšanas algoritmam, šis algoritms ir balstīts uz gradienta lejupslīdes metodes.

Piedāvāto apmācīšanas algoritmu var izmantot izplūdušo parametru *ieguldījuma* noteikšanai kritērija vēlamas vērtības sasniegšanai. Kā būs redzams tālāk tekstā, to var izmantot lēmumu pieņemšanas modeļa jūtīguma analīzē [Clemen&Reilly, 2004]. Ja mums ir uzdota kritērija *vēlama vērtība*, tad ar šīs sistēmas palīdzību ir iespējams noteikt katra izplūdušā parametra *svarīgumu*, kas nosaka cik svarīgs ir dotais parametrs, lai kritērijs pieņemtu uzdoto vēlamo vērtību. Citiem vārdiem, lielāka *svarīguma* vērtība liecina par to, ka dotā parametra vērtība lielākā mērā veicina to, ka kritērijs pieņem vēlamo vērtību.

3.1.1.1. Adaptīvā tīkla ANGIE topoloģija

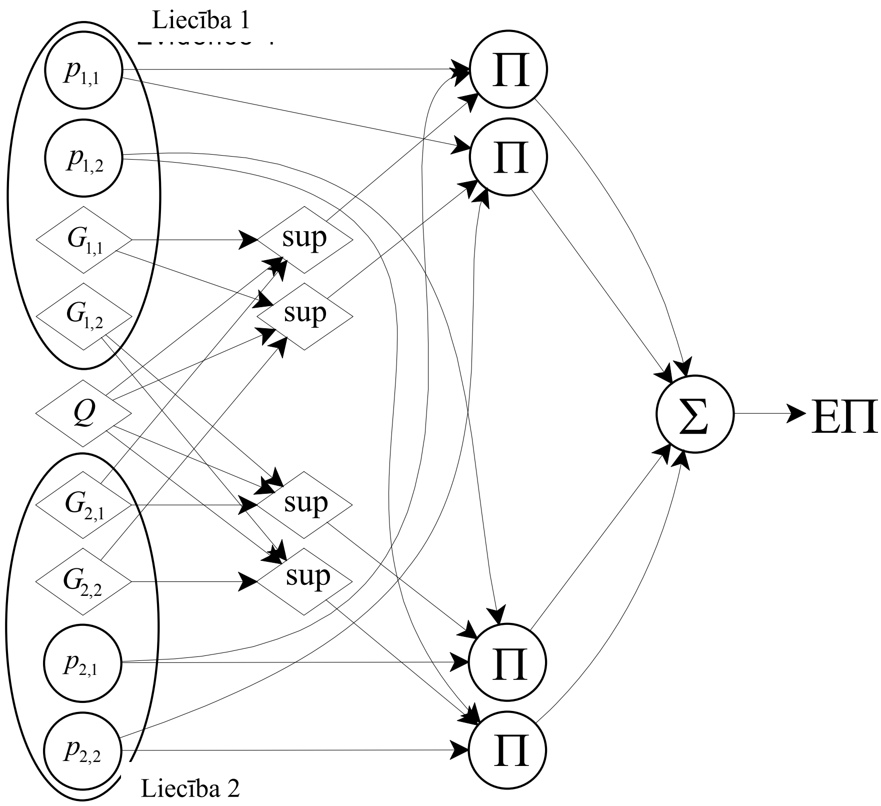
Apskatīsim adaptīvā tīkla ANGIE topoloģiju. Apskatīsim sistēmu, kura sastāv no divām liecībām:

I. IF (X_1 is B_{11}) with p_{11} THEN (Y is G_{11})
   IF (X_1 is B_{12}) with p_{12} THEN (Y is G_{12})

II. IF (X_2 is B_{21}) with p_{21} THEN (Y is G_{21})
   IF (X_2 is B_{22}) with p_{22} THEN (Y is G_{22})

Adaptīvais tīkls, kas ir paredzēts augšējas varbūtības robežas (1.11) noteikšanai ir parādīts 3.1. attēlā [Vališevskis&Borisov, 2003c].

Apskatīsim kādas darbības veic un par ko ir atbildīgs katrs no tīkla elementiem. Ovālie elementi ir paredzēti neizplūdušās informācijas apstrādei, bet romba elementi ir paredzēti izplūdušas informācijas apstrādei.



3.1. att. Augšējas varbūtības robežas noteikšanai paredzēts adaptīvais tīkls

Pirmais slānis.

Elementa $\boldsymbol{G_{ij}}$ izeja ir izplūdusī vērtība, kura atbilst *i*-tās liecības *j*-tās granulas labajā pusē esošai vērtībai.

Elementa $\boldsymbol{p_{ij}}$ izeja ir varbūtības vērtība, kas ir saistīta ar *i*-tās liecības *j*-tās granulas kreisajā pusē esošo vērtību.

Elementa ***Q*** izeja ir kritērija vēlamā vērtība.

Otrais slānis.

Elements, kas tiek apzīmēts ar ***sup*** ir paredzēts tajā ienākošo izplūdušo vērtību šķēluma suprēmuma noteikšanai. Izplūdušās vērtības, kuras elements saņem ieejā, tiek sakopotas ar šķēluma funkcijas palīdzību. Šī elementa aktivēšanas funkcija ir suprēmuma funkcija. Tātad, elementa ***sup*** izeja ir ieejā saņemto izplūdušo vērtību šķēluma suprēmums.

Trešais slānis.

Elementa Π izeja ir tajā ieejošo vērtību reizinājums.

Ceturtais slānis.

Elementa Σ izeja ir tajā ieejošo vērtību summa.

Sagaidāmas noteiktības *EC* vērtību var noteikt līdzīgā veidā. Vienīga atšķirība ir tāda, ka elementa ***Q*** vietā būs elements ***Q'***, kas apzīmē izplūdušās kopas ***Q*** papildinājumu un tīkla izeja būs $(1-\Sigma)$, nevis Σ.

3.1.1.2. Adaptīvā tīkla ANGIE apmācības algoritms

Tagad aprakstīsim adaptīvā tīkla ANGIE apmācības algoritmu. Sistēmas ANGIE apmācīšanai tiek izmantota gradienta lejupslīdes metode. Tas nozīmē, ka parametru regulēšanai mums ir jānosaka kļūdas funkcijas parciālie atvasinājumi pēc parametriem, kuru regulēšanu mēs gribam veikt. Sīkāk uz gradienta lejupslīdes balstītie apmācības algoritmi ir aprakstīti rakstā [Vališevskis, 2001]. Izstrādātais algoritms ir līdzīgs ANFIS adaptīvajam tīklam izstrādātajam algoritmam [Vališevskis, 2002a].

Izmantosim tādu pati kļūdas funkciju, kura tiek izmantota neironu tīklos [Rumelhart u.c., 1986]:

$$\mathrm{E} = \tfrac{1}{2}(d-o)^2,$$

kur *d* ir vēlama izejas vērtība un *o* ir iegūta izejas vērtība.

Sistēmas izeja ir definēta ar funkcijas (1.11) palīdzību. Ir acīmredzams, ka nav iespējams veikt piederības funkciju parametru regulēšanu, jo signāls, kas iet no elementiem, kuri sūta izplūdušās vērtības, tiek apstrādāts elementā ***sup*** un, tādējādi, nav iespējams noteikt kļūdas funkcijas parciālos atvasinājumus pēc piederības funkciju parametriem.

Tagad noteiksim parciālos atvasinājumus pēc sistēmas parametriem. Saskaņā ar ķēdes likumu:

$$\frac{dE}{dp_i} = \frac{dE}{do}\frac{do}{dp_i},$$

$$\frac{dE}{do} = (o - d), \quad \text{un} \quad \frac{do}{dp_i} = \sum_j p_{ij} \sup\left(Q \cap G_i \cap H_j\right),$$

tātad, $$\frac{dE}{dp_i} = (o - d)\sum_j p_{ij} \sup\left(Q \cap G_i \cap H_j\right). \tag{3.1}$$

Formula (3.1) ir speciālais gadījums, kad sistēmā ir divas liecības. Parciālais atvasinājumus pēc dotā varbūtiska parametra ir tādi $\mathrm{E}\Pi(Q)$ elementi, kuros ir šis varbūtiskais parametrs. Vispārīgā gadījumā parciālais atvasinājums izskatās šādi:

$$\frac{dE}{dp_{i_k}} = (o - d)\sum_{i_1=1}^{m_1}\sum_{i_2=1}^{m_2}\ldots\sum_{i_{k-1}=1}^{m_n}\sum_{i_{k+1}=1}^{m_n}\ldots\sum_{i_n=1}^{m_n} p_{i_1 i_2 \ldots i_n} \sup\left(Q \cap G_{i_1}^1 \cap G_{i_2}^2 \cap \ldots \cap G_{i_n}^n\right),$$

kur $p_{i_1 i_2 \ldots i_n} = p_{i_1} p_{i_2} \ldots p_{i_n}$ un G_i^j ir *j*-tās liecības *i*-tā granula.

Izmantojot šo formulu var apmācīt adaptīvo tīklu ANGIE.

Bet ko nozīmē “veikt lēmumu pieņemšanas sistēmas parametru regulēšanu” vai “veikt varbūtību vērtību regulēšanu”? Kādam nolūkam būtu jāveic varbūtību vērtību regulēšana ar adaptīvā tīkla palīdzību? Apskatīsim šos jautājumus sīkāk. Kā ir norādīts rakstā [Vaļiševskis, 2002], varbūtību vērtību regulēšanu ir iespējams veikt ja mums ir uzdots liecību kopums, kas apraksta kādu kritēriju un kad ir iespējams noteikt šī kritērija vēlamo vērtību. Šajā gadījumā varbūtību parametru regulēšana var tikt interpretēta kā varbūtību vērtību meklēšana, ar kurām uzdotās liecības vislielākā mērā apmierina kritērija vēlamo vērtību. Turklāt, iegūtās varbūtību vērtības var uzskatīt par dotas granulas *svaru* vai *svarīgumu* kritērija vēlamas vērtības sasniegšanai. Citiem vārdiem, jo lielāka varbūtības vērtība, jo lielāka attiecīgas granulas kreisajā pusē uzdotās parametra vērtības *pieļaujamība* vai *preference*.

3.1.1.3. Adaptīvā tīkla ANGIE apmācības piemērs

Kā piemēru apskatīsim lēmumu pieņemšanas sistēmu, kura nosaka labāko (ņemot vērā dotos datus) alternatīvu starp dažiem iespējamiem augstsprieguma līnijas būvēšanas projektiem. Apskatīsim vienkāršoto piemēru, jo tagad mūsu mērķis ir apskatīt varbūtību vērtību regulēšanas nozīmi lēmumu pieņemšanas sistēmās.

Pieņemsim, ka mums ir šādi parametri:

- X_1 = iedzīvotāju blīvums;
- X_2 = augstsprieguma līnijas garums.

Turklāt, mums ir uzdots liecību kopums, kas apraksta kritēriju “Y = vizuālais efekts”:

1. Liecība:

JA X_1 = LIELS ar varbūtību p_{11} TAD Y = LIELS,

JA X_1 = VIDĒJAIS ar varbūtību p_{12} TAD Y = VIDĒJAIS,

2. Liecība:

JA $X_2 \approx 30$ km ar varbūtību p_{21} TAD Y = LIELS,

JA $X_2 \approx 50$ km ar varbūtību p_{22} TAD Y = ĻOTI LIELS,

kur $p_{11} + p_{12} = p_{21} + p_{22} = 1.$

Pēc liecību kopuma definēšanas un sistēmas ANGIE uzbūvēšanas, ir jāveic adaptīvā tīkla apmācīšana. Pieņemsim, ka par kritērija *Y* vēlamo vērtību mēs izvēlējamies izplūdušo vērtību *Q*. Pēc sistēmas apmācīšanas mēs iegūstam varbūtību vērtības.

Pieņemsim, ka mēs ieguvām sekojošās vērtības:

$$p_{11} = 0,79,\ p_{12} = 0,21$$

$$p_{21} = 0,47,\ p_{22} = 0,53$$

Šīs vērtības var interpretēt sekojošajā veidā. Lai kritērijs *Y* pieņemtu vērtību *Q* ir pietiekams, lai notikuma $(X_1 = LIELS)$ varbūtība būtu vienāda ar 0,79, notikuma $(X_1 = VIDĒJAIS)$ varbūtību būtu vienāda ar 0,21, notikuma $(X_2 \approx 30\ km)$ varbūtību būtu vienāda ar 0,47 un notikuma $(X_2 \approx 50\ km)$ varbūtību būtu vienāda ar 0,53.

Citiem vārdiem, notikuma $(X_1 = LIELS)$ *svarīgums* ir 0,79 un notikuma $(X_1 = VIDĒJAIS)$ svarīgums ir 0,21. Savukārt, notikumu $(X_2 \approx 30\ km)$ un $(X_2 \approx 50\ km)$ svarīgums ir gandrīz vienāds, kas varētu liecināt par to, ka parametrs X_2 neietekmē notikuma *(Y ir Q)* varbūtību.

Ir acīmredzams, ka varbūtību noteikšanai ir jāizmanto tīkls ANGIE, kas ir paredzēts noteikuma *(Y ir Q)* sagaidāmas noteiktības aprēķināšanai, jo tā ir varbūtības apakšējā robeža.

Pirms tīkla apmācīšanas varbūtības ir jānoinicializē ar vērtību $\frac{1}{N}$, kur N ir granulu skaits liecībā, līdz ar to sākumā visas varbūtības būs līdzvērtīgas.

Iegūto informāciju var izmantot veicot lēmumu pieņemšanas modeļa jūtīguma analīzi. Viens no uzdevumiem, kurš tiek apskatīts jūtīguma analīzes kontekstā ir priekšrokas atkarības no dažādu notikumu varbūtības pētīšana [Clemen&Reilly, 2004]. Citiem vārdiem, tiek pētīta priekšrokas izmaiņa mainoties vienam vai vairākiem varbūtiskiem parametriem. Piemēram, vienvirzienā jūtīguma analīzē visi sistēmas parametri, izņemot vienu, tiek fiksēti un tiek pētīta priekšrokas atkarība no viena mainīgā parametra. *Tornado* diagrammas ir viens no vienvirziena jūtīguma analīzes piemēriem. Divvirzienu jūtīguma analīzē ir divi mainīgie parametri.

Veicot vienvirziena vai divvirzienu jūtīguma analīzi, iegūtos rezultātus ir iespējams attēlot grafiski. Bet kas ir jādara, ja mums ir jānoskaidro triju vai vairāku varbūtisko parametru mijiedarbība? Attēlot šāda uzdevuma risinājumu grafiski ir sarežģīti. Šī uzdevuma apakšuzdevumu var noformulēt šādi: jānoskaidro apgabals varbūtisko parametru telpā, kas apmierinātu kritērija vēlamas vērtības sasniegšanas noteikumu. Turklāt, ir iespējams izpildīt vairākus eksperimentus, kuru rezultātā var iegūt dažus punktus parametru telpā, katrs no kuriem reprezentētu dažādas alternatīvas.

Šo uzdevumu ir iespējams risināt ar sistēmas ANGIE palīdzību. Pēc sistēmas apmācīšanas iegūtās varbūtību vērtības var uzskatīt par tuvinājumu kritērija vēlamas vērtības sasniegšanas nosacījuma apmierināšanai.

Turpmākajā analīzē var noteikt attālumu starp iegūtajiem punktiem varbūtisko parametru telpā, kuri reprezentē dažādas alternatīvas.

Šajā nodaļā tika aprakstīts adaptīvais tīkls, kas ir paredzēts izmantošanai uz izplūdušajām liecībām balstītās lēmumu pieņemšanas sistēmās. Tīklu var izmantot kā skaitļošanas tīklu varbūtības apakšējas un augšējas robežas noteikšanai. Turklāt, piedāvāto tīkla apmācīšanas algoritmu var izmantot, lai noskaidrotu izplūdušo granulu “ieguldījumu” kritērija vēlamās vērtības sasniegšanā.

3.1.1.4. Adaptīvā tīkla ANGIE izmantošana svarīguma noteikšanai

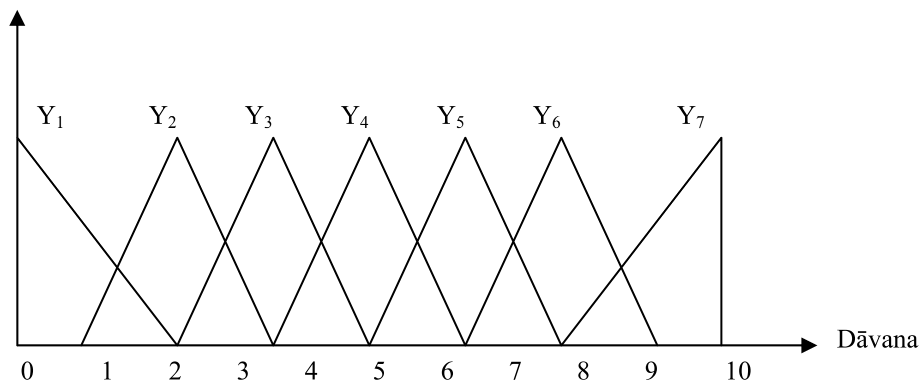
Šajā nodaļā tiek risināts uzdevums, kurā tiek noteikta dažādu dāvanu raksturojošo parametru ietekme uz dāvanas novērtējumu.

Dotais uzdevums ir līdzīgs jūtīguma analīzes uzdevumam. Tomēr, tradicionālajā jūtīguma analīzē tiek aplūkota viena parametra ietekme uz priekšrokas izmaiņu (vienvirziena jūtīguma analīze, piemēram, *Tornado diagrammas*), vai divu varbūtisko parametru mijiedarbība (divvirzienu jūtīguma analīze). Savukārt, šajā eksperimentā tiek analizēta 20 varbūtisko parametru mijiedarbība. Veicot vienvirziena vai divvirzienu jūtīguma analīzi, risinājumu ir iespējams attēlot grafiski, savukārt, ir acīmredzams, ka šajā uzdevuma tas nav iespējams, jo uzdevuma risinājums ir punkti daudzdimensionālā telpā.

Novērtējumu veiksim pēc viena kritērija, proti, pēc *"Y = Dāvanas novērtējums"*. Dažas šī kritērija vērtības ir norādītas zemāk.

1. Y_1 = Labāk to nedāvināt
2. Y_2 = Nevajadzīgs sīkums
3. Y_3 = Sīkums, kuram var atrast pielietojumu
4. Y_4 = Saimniecībā vajadzīga lieta
5. Y_5 = Saņēmējs būs priecīgs
6. Y_6 = Es pats gribētu šādu dāvanu
7. Y_7 = Labāk nesamulsināt saņēmēju

Atribūti un to vērtības, kurus var izmantot, lai aprakstītu dāvanu, ir norādītas 3.1. tabulā. Savukārt, kritērija "dāvanas novērtējums" lingvistiskās vērtības, kuras ir definētas uz nosacītas skalas no 0 līdz 10, ir parādīti 3.2. attēlā.



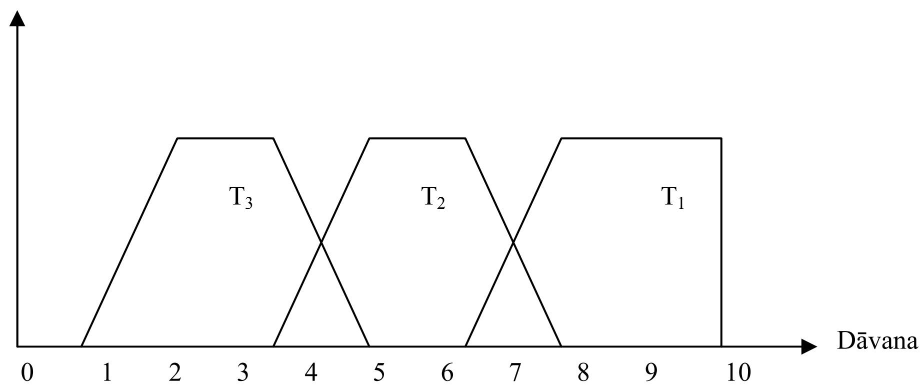
3.2. att. Kritērija izplūdušo vērtību piederības funkcijas

3.1. tabula

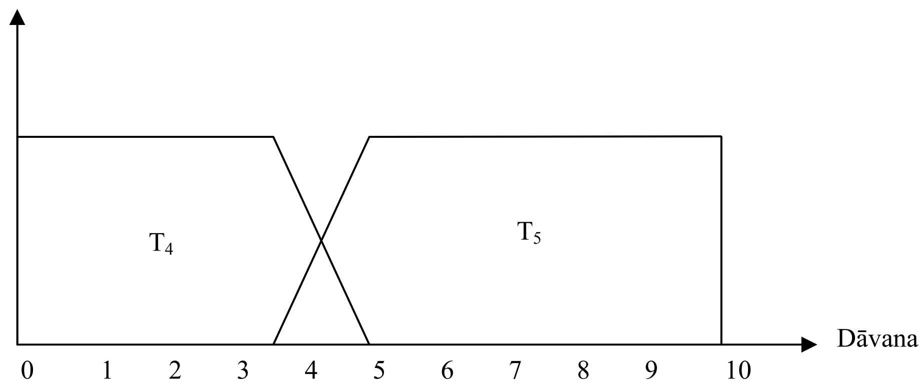
Atribūti un to vērtības

Atribūts	Vērtības
Cena	Ļoti liela Liela Pieņemama Zema Aizdomīgi zema
Krāsa	Raiba Spilgta Mērena Blāva
Izdevumi, kuri rodas dāvanas ekspluatācijā	Atbilst dāvanas cenai Nav lieli Nav
Cik ilgi cilvēkam dāvana būs interesanta	Ilgums nav ierobežots Līdz nākamai dzimšanas dienai Mēness Nedēļa Pusstunda
Svars	Ļoti smags Smags Vidēji smags Viegls Nesverams
Cik ilgi var izmantot bez papildus izmaksām	Ir nepieciešamas uzreiz Nepietiekami ilgi Pietiekami ilgi Papildus izmaksu nav

Nodefinēsim papildus piederības funkcijas, kuras tiek izmantotas liecībās, tās ir parādītas 3.3. un 3.4. attēlā.



3.3. att. Papildus piederības funkcijas



3.4. att. Papildus piederības funkcijas

Zemāk ir aprakstītas liecības, uz kuru pamata tiek uzdota sistēma ANGIE. Apskatītajā uzdevumā ir jānoskaidro varbūtisko parametru vērtības, kuras nodrošinātu to, ka notikuma *(Y ir Q)* varbūtības apakšējā robeža ir vienāda ar vieninieku, kur *Q* ir vēlamā kritērija vērtība.

1. liecība.
Parametram X_1 = Cena.
Ja X_1 = *ĻOTI LIELA* ar varbūtību p_{11}, tad Y = Y_7
Ja X_1 = *LIELA* ar varbūtību p_{12}, tad Y = T_1
Ja X_1 = *PIEŅEMAMA* ar varbūtību p_{13}, tad Y = T_2
Ja X_1 = *ZEMA* ar varbūtību p_{14}, tad Y = T_3
Ja X_1 = *AIZDOMĪGI ZEMA* ar varbūtību p_{15}, tad Y = Y_1

2. liecība.
Parametram X_2 = Krāsa.
Ja X_2 = *SPILGTA* ar varbūtību p_{21}, tad Y = T_4
Ja X_2 = *MĒRENA* ar varbūtību p_{22}, tad Y = T_5
Ja X_2 = *BLĀVA* ar varbūtību p_{23}, tad Y = T_2

3. liecība.
Parametram X_3 = Izdevumi, kuri rodas dāvanas ekspluatācijā.
Ja X_3 = *ATBILST DĀVANAS CENAI* ar varbūtību p_{31}, tad Y = Y_1
Ja X_3 = *NAV LIELI* ar varbūtību p_{32}, tad Y = T_2
Ja X_3 = *NAV* ar varbūtību p_{33}, tad Y = T_1

4. liecība.
Parametram X_4 = Cik ilgi cilvēkam dāvana būs interesanta.
Ja X_4 = *ILGUMS NAV IEROBEŽOTS* ar varbūtību p_{41}, tad Y = T_1
Ja X_4 = *LĪDZ NĀKAMAI DZIMŠANAS DIENAI* ar varbūtību p_{42}, tad Y = T_2
Ja X_4 = *MĒNESS* ar varbūtību p_{43}, tad Y = T_3
Ja X_4 = *NEDĒĻA* ar varbūtību p_{44}, tad Y = Y_2
Ja X_4 = *PUSSTUNDA* ar varbūtību p_{45}, tad Y = Y_1

5. liecība.
Parametram X_5 = Svars.
Ja X_5 = *LIELS* ar varbūtību p_{51}, tad Y = Y_1
Ja X_5 = *VIDĒJS* ar varbūtību p_{52}, tad Y = T_2
Ja X_5 = *VIEGLS* ar varbūtību p_{53}, tad Y = Y_6
Ja X_5 = *NESVERAMS* ar varbūtību p_{54}, tad Y = T_1

6. liecība.
Parametram X_6 = Cik ilgi var izmantot bez papildus izmaksām.
Ja X_6 = *IR NEPIECIEŠAMAS UZREIZ* ar varbūtību p_{61}, tad Y = Y_1
Ja X_6 = *NEPIETIEKAMI ILGI* ar varbūtību p_{62}, tad Y = Y_3
Ja X_6 = *PIETIEKAMI ILGI* ar varbūtību p_{63}, tad Y = T_2
Ja X_6 = *PAPILDUS IZMAKSU NAV* ar varbūtību p_{64}, tad Y = T_1

Vēlama kritērija vērtība ir vienāda ar *"Y_6 = Es pats gribētu šādu dāvanu".*

Apmācīšanas ātrums ir vienāds ar 0,3.

Tika veikti divi eksperimenti: pirmajā eksperimentā varbūtību vērtību normēšana tika veikta pēc katras 5. iterācijas; otrajā eksperimentā normēšana tika veikta pēc sistēmas apmācīšanas pabeigšanas.

Sistēma, kurā normēšana tika veikta pēc katras 5. iterācijas.

Pirmā sistēma konverģēja 140. iterācijā. Konverģences līnija (resnāka līnija) un varbūtību vērtību dinamika ir parādīta 3.5. attēlā.

Pēc sistēmas apmācīšanas tika iegūtas šādas varbūtību vērtības:

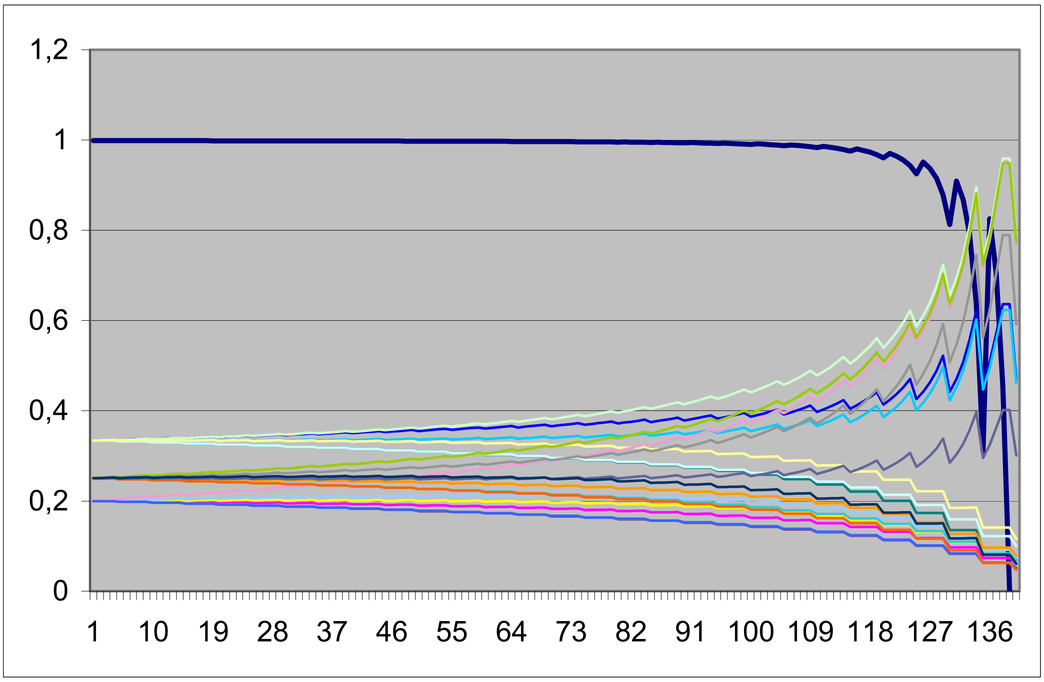
p_{11} =0.0599144 p_{12} =0.0672988 p_{13} =0.770188 p_{14} =0.0512992 p_{15} =0.0512992	p_{21} =0.0683335 p_{22} =0.470676 p_{23} =0.46099	p_{31} =0.0995307 p_{32} =0.78486 p_{33} =0.11561
p_{41} =0.0745073 p_{42} =0.770528 p_{43} =0.051655 p_{44} =0.051655 p_{45} =0.051655	p_{51} =0.0681319 p_{52} =0.774718 p_{53} =0.0785752 p_{54} =0.0785752	p_{61} =0.0467137 p_{62} =0.301006 p_{63} =0.591625 p_{64} =0.0606553

Sistēma, kurā normēšana tika veikta pēc apmācīšanas procesa pabeigšanas.

Otrā sistēma konverģēja 78. iterācijā. Konverģences līnija (resnāka līnija) un varbūtību vērtības ir parādītas 3.6. attēlā.

Pēc sistēmas apmācīšanas tika iegūtas sekojošas varbūtību vērtības:

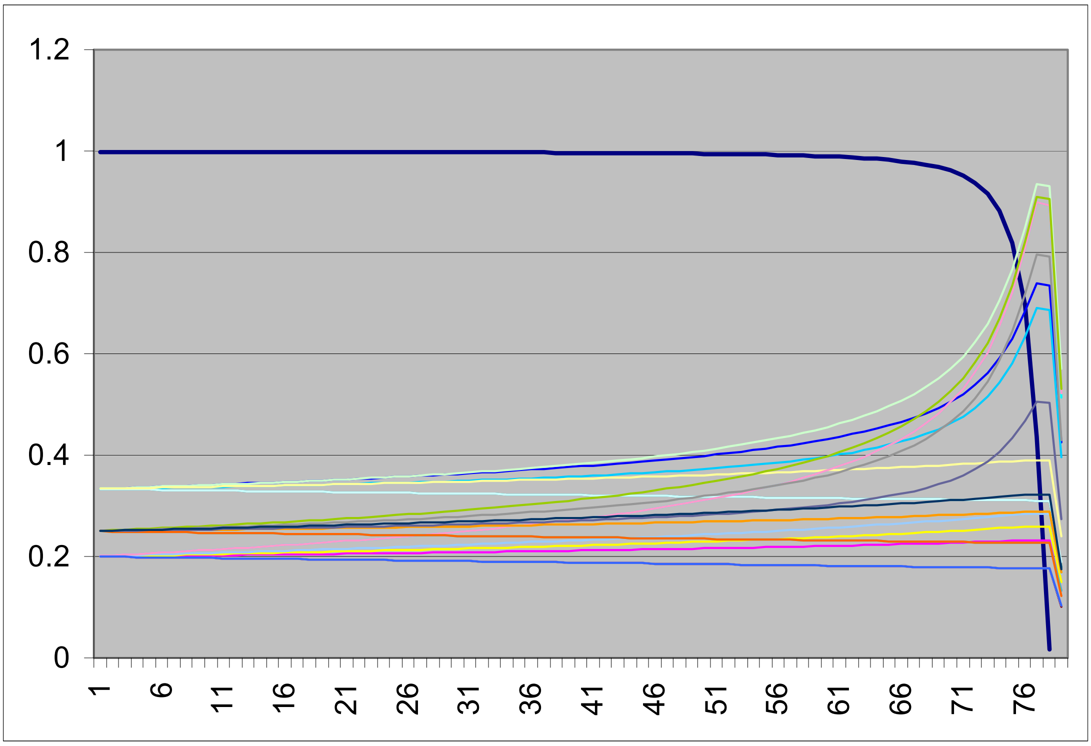
p_{11} =0.133561 p_{12} =0.149396 p_{13} =0.513799 p_{14} =0.101622 p_{15} =0.101622	p_{21} =0.178997 p_{22} =0.424877 p_{23} =0.396126	p_{31} =0.190155 p_{32} =0.570355 p_{33} =0.23949
p_{41} =0.166968 p_{42} =0.522816 p_{43} =0.103405 p_{44} =0.103405 p_{45} =0.103405	p_{51} =0.132685 p_{52} =0.529669 p_{53} =0.168823 p_{54} =0.168823	p_{61} =0.123005 p_{62} =0.272749 p_{63} =0.429368 p_{64} =0.174877



3.5. att. Konverģences līnija un varbūtības (normēšana pēc katras 5. iterācijas)

Ir jāatzīmē tas, ka otrā sistēma atšķiras no pirmās sistēmas ne tikai ar to, ka varbūtību normēšana notiek tikai pēc apmācīšanas procesa pabeigšanas. Atšķirībā no pirmās sistēmas, otrajā sistēmā tiek pielietots elementu sodīšanas princips. Ja dotā elementa apmācošs signāls ir vienāds ar nulli, tad ar tā saistīta varbūtības vērtība tiek samazināta (pirmajā sistēmā tā nemainās). To var pamatot ar to, ka ja elements neietekmē notikuma *(Y ir Q)* varbūtību, tad tas ir jāsoda un jāsamazina ar to saistīta varbūtības vērtība.

Salīdzinot 3.5. un 3.6. attēlus, ir redzams, ka otrā sistēma konverģē ātrāk nekā pirmā sistēma. To var izskaidrot ar to, ka pirmajā sistēmā regulāri notiek varbūtību vērtību normēšana. Savukārt, otrajā sistēmā apmācīšanas laikā iespējamas varbūtību vērtības nekādi netiek ierobežotas (piemēram, nav nosacījuma, ka to summai jābūt vienādai ar vieninieku), tās tiek normētas pēc apmācīšanas pabeigšanas.



3.6. att. Konverģences līnija un varbūtības (normēšana pēc apmācīšanas pabeigšanas)

Var izdarīt sekojošu secinājumu: veicot varbūtību vērtību normēšanu apmācīšanas laikā, rezultātā mēs iegūstam skaitliski precīzāku atrisinājumu, bet veikto iterāciju skaits apmācīšanas laikā ir lielāks. Veicot varbūtību vērtību normēšanu pēc sistēmas apmācīšanas mēs iegūstam intuitīvi pareizo atrisinājumu, kas skaitliski varētu būt ne tik precīzs kā pirmās sistēmas atrisinājums. No praktiskā viedokļa, sistēmas, kurās normēšana tiek veikta pēc apmācīšanas pabeigšanas, ir pievilcīgākas, jo tās prasa mazāk skaitļošanas resursus, bet lēmumu pieņemšanas uzdevumi parasti ir diezgan ietilpīgi no skaitļošanas viedokļa.

Bet cik svarīgs ir skaitliski precīzāks atrisinājums? Ņemot vērā to, ka šajā sistēmā varbūtību vērtības zaudē savu sākotnējo nozīmi un tiek apskatītas kā attiecīgo parametru *svarīguma mērs*, var uzskatīt, ka varbūtību vērtībām nav noteicošā loma. Svarīgāka ir varbūtību vērtību savstarpēja attiecība, kas ļauj noteikt kurš no parametriem ir vairāk vai mazāk *svarīgs*.

Mēģināsim ar vārdiem interpretēt iegūto rezultātu. Kā jau iepriekš tika atzīmēts, kritērija vēlama vērtība ir *“Y_6 = Es pats gribētu šādu dāvanu”*. Zemāk ir piedāvāta intuitīva iegūto svarīguma vērtību interpretācija.

Tādejādi, lai kritērija vērtība *“Es pats gribētu šādu dāvanu”* būtu maksimāli apmierināta, ir spēkā sekojoši apgalvojumi:

- Vispiemērotākā dāvanas cena ir “pieņemama cena”, otrā pēc svarīguma vērtība ir “liela cena”, kurai seko “ļoti liela cena”. Viszemāk piemērotas vērtības ir “zema cena” un “aizdomīgi zema cena”.
- Dāvanas krāsa var būt kā “mērena”, tā arī “blāva” (tomēr piemērotāka ir “mērena krāsa”), jo šo parametru svarīguma vērtības ir gandrīz vienādas. Viszemāk piemērotāka ir “spilgta krāsa”.
- Vispiemērotākais ar dāvanas ekspluatāciju saistīto izdevumu līmenis ir “neliels”, kuram seko “izdevumu nav”. Vismazāk piemēroti ir “izdevumi, kuri atbilst dāvanas cenai”.
- Parametra “cik ilgi cilvēkam dāvana būs interesanta” vislielākais svarīguma koeficients ir vērtībai “līdz nākamai dzimšanas dienai”, tam seko vērtība “ilgums nav ierobežots”. Viszemākās svarīguma koeficienta vērtības ir parametriem “mēness”, “nedēļa” un “pusstunda”.
- Vispiemērotākais dāvanas svars ir “vidējs”, tam seko vērtības “viegls” un “nesverams”, kurām ir vienādi svarīguma koeficienti. Viszemāk piemērots ir “liels svars”.
- Parametra “cik ilgi var izmantot bez papildus izmaksām” vispiemērotākā vērtība ir “pietiekami ilgi”. Nākoša pēc svarīguma vērtība ir “nepietiekami ilgi”, kurai seko “papildus izmaksu nav”. Sarakstu noslēdz vērtība “papildus izmaksas ir nepieciešamas uzreiz”, kurai ir viszemākais svarīguma koeficients.

3.1.1.5. Adaptīvā tīkla ANGIE izmantošana svarīguma noteikšanai. Secinājumi

Tā kā sistēma *ANGIE* ir balstīta uz adaptīvo tīklu, ir iespējams veikt šīs sistēmas apmācīšanu. Sistēmas apmācīšanas rezultātā mēs iegūstam izplūdušo parametru *svarīguma* koeficientus, kuri nosaka cik *svarīgs* ir dotais parametrs, lai kritērija vērtība būtu vienāda ar vēlamo vērtību. Tātad, varbūtības nozīme šajā uzdevuma mainās un tās tiek uzskatītas par *svarīguma koeficientiem*. Tomēr, paliek nosacījums, ka varbūtību vērtību summai jābūt vienādai ar vieninieku. Tas nozīmē, ka sistēmas apmācīšanas laikā ir jāveic varbūtību vērtību normēšana. Ir trīs iespējas: normēšanu var veikt vai nu pēc katras iterācijas, vai pēc katras *n*-tās iterācijas, vai tikai apmācīšanas beigās. Eksperimentu rezultāti liecina par to, ka veicot normēšanu pēc katras iterācijas, sistēmas apmācīšanas konverģences ātrums samazinās un apmācīšanas process vispār var diverģēt. Veicot normēšanu pēc katras *n*-tās iterācijas, sistēma konverģē ātrāk nekā iepriekšēja sistēma un iegūtie rezultāti ir precīzākie, nekā sistēmai, kurā normēšana notiek tikai pēc apmācīšanas beigām. Tomēr, tika secināts, ka varbūtību vērtību skaitliskai precizitātei nav būtiskas lomas, jo, kā jau tika atzīmēts, varbūtības vērtības pārvēršanas svarīguma koeficientos. Tādejādi, būtiska ir šo koeficientu savstarpēja attiecība, neviss skaitliskā vērtība, lai varētu noteikt kurš no parametriem ir *svarīgāks*.

3.1.2. Entropijas vispārinājums intervālu varbūtību gadījumam

Piedāvātajā metodē tiek izmantota intervālu entropija, proti, tā ir jārēķina balstoties uz intervālu varbūtībām. Līdz ar to Šenona entropija [Shannon&Weaver, 1949] ir jāvispārina intervālu gadījumam. Rakstā [Viliševskis&Borisov, 2003a] šis uzdevums ir nodefinēts un ir aprakstītas dažas heiristiskas pieejas entropijas vispārināšanai. Tomēr, šīs metodes nav precīzas un tām nav iespējams novērtēt kļūdu. Analītiskais risinājums ir izklāstīts rakstā [Viliševskis&Borisov, 2006].

Šajā nodaļā vispārinājuma uzdevums ir aprakstīts kā optimizācijas uzdevums ar ierobežojumiem nevienādojumu veidā. Vispirms tiks dota uzdevuma formālā nostādne (sk. 3.1.2.1. apakšparagrāfu), pēc tam tas tiks atrisināts analītiski ar Lagranža reizinātāju metodes palīdzību [Cornuejols u.c., 1998]. Nodaļas beigās ir daži piemēri, kuri parāda kādā veidā aprakstīto risinājumu var izmantot praksē.

3.1.2.1. Vispārīga uzdevuma nostādne

Pieņemsim, ka sistēma var atrasties *n* stāvokļos un varbūtība, ka sistēma atrodas *i*-tajā stāvoklī ir intervālu un ir vienāda ar $[p_i^{min}, p_i^{max}]$. Gadījumā, ja varbūtības ir precīzas, nevis intervālu, to summai jābūt vienādai ar vieninieku, proti:

$$\sum_i p_i = 1. \tag{3.2}$$

Ja varbūtības ir intervālu, tad (3.2) var pārrakstīt kā (3.3).

$$\sum_i p_i^{\min} \le 1 \le \sum_i p_i^{\max} . \tag{3.3}$$

Ir viegli parādīt, ka (3.2) ir nevienādības (3.3) speciālgadījums kad

$$\forall i : p_i^{\min} = p_i^{\max} .$$

Entropija arī ir intervālu: $H = [H^{\min}, H^{\max}]$. Intervālu entropijas rēķināšanas vispārīgu uzdevumu var nodefinēt šādi:

Pieņemsim, ka $[p_1^{min}, p_1^{max}]$, $[p_2^{min}, p_2^{max}]$, ... , $[p_n^{min}, p_n^{max}]$ ir intervālu varbūtības, tad intervālu entropijas apakšējo un augšējo robežu var noteikt sekojošajā veidā:

Apakšējā robeža: $H^{\min} = \min\left(-\sum_{i=1}^{n} p_i \ln p_i \right)$

un augšējā robeža: $H^{\max} = \max\left(-\sum_{i=1}^{n} p_i \ln p_i \right)$, (3.4)

ar sekojošajiem nosacījumiem:

$p_i^{\min} \le p_i \le p_i^{\max}, \qquad i = \overline{1,n}$,

un $\sum_{i=1}^{n} p_i = 1$

Ērtības labad tiek izmantoti naturālie logaritmi, bet tas nemaina problēmas būtību.

3.1.2.2. Uzdevuma analītiskais atrisinājums.

Uzdevuma (3.4) atrisināšanai tiek izmantota Lagranža reizinātāju metode. Vispirms visi ierobežojumi ar $\ge$ zīmi jāpārveido nevienādībās ar $\le$ zīmi. Turklāt, minimizēšanas uzdevums ir jāpārveido maksimizēšanas uzdevumā. Vispirms apskatīsim maksimizēšanas uzdevuma (3.5) risinājumu.

$H^{\max} : -\sum_{i=1}^{n} p_i \ln p_i \to \max$

ar sekojošajiem ierobežojumiem: (3.5)

$-p_i \le -p_i^{\min}, \quad p_i \le p_i^{\max}, \qquad i = \overline{1,n}.$

$\sum_{i=1}^{n} p_i = 1$

Lagranža vienādojums uzdevumam (3.5) izskatās šādi:

$$
\begin{aligned}
&L\left(p_1,...,p_n,\lambda,\mu_{11},\mu_{12},\mu_{21},\mu_{22},...,\mu_{n1},\mu_{n2},\right)= \\
&=-\sum_{i=1}^{n} p_i \ln p_i + \lambda\left(\sum_{i=1}^{n} p_i - 1\right) + \sum_{i=1}^{n}\left(\mu_{i1}\left(p_i - p_i^{\min}\right) + \mu_{i2}\left(p_i^{\max} - p_i\right)\right)
\end{aligned}
\tag{3.6}
$$

Lagranža vienādojuma (3.6) parciālie atvasinājumi izskatās sekojošajā veidā:

$$\frac{\partial L}{\partial p_i} = -\ln p_i - 1 + \lambda + \mu_{i1} - \mu_{i2}, \qquad i = \overline{1,n}.$$

Tādējādi, ir jāatrisina šāds uzdevums:

$$
\left|
\begin{array}{lr}
-\ln p_i - 1 + \lambda + \mu_{i1} - \mu_{i2} = 0, \qquad i = \overline{1,n}, & (3.7) \\
\sum_{i=1}^{n} p_i = 1, & (3.8) \\
\mu_{i1}\left(p_i - p_i^{\min}\right) = 0, \quad \mu_{i2}\left(p_i^{\max} - p_i\right) = 0, \quad i = \overline{1,n}, & (3.9) \\
p_i^{\min} \le p_i \le p_i^{\max}, \ \mu_{ij} \ge 0, \qquad i = \overline{1,n}, \ j = 1,2. & (3.10)
\end{array}
\right.
$$

Tagad ir jāapskata 2^{2n} gadījumi, kuri atšķiras ar μ_{ij} vērtībām nosacījumā (3.9). Proti, atkarībā no tā vai μ_{ij} vērtība ir vai nav vienāda ar nulli.

1. Ja $\mu_{ij} = 0, i = \overline{1,n}, j = 1,2$ un visas intervālu varbūtības satur vērtību $\frac{1}{n}$, tad tas ir maksimizēšanas uzdevuma atrisinājums. Pretējā gadījumā jāpāriet pie nākošā soļa.
2. Jāpārbauda visas konfigurācijas, kuras atšķiras ar to, vai attiecīgā μ_{ij} vērtība ir vienāda ar nulli. Var pamanīt, ka ja μ_{i1} vai μ_{i2} vērtība nav vienāda ar nulli, tad saskaņā ar (3.9) izteiksmju $\left(p_i - p_i^{\min}\right)$ vai $\left(p_i^{\max} - p_i\right)$ vērtībai jābūt vienādai ar nulli. Tādējādi, ja μ_{i1} vai μ_{i2} vērtība nav vienāda ar nulli, tad, attiecīgi, $p_i = p_i^{\min}$ vai $p_i = p_i^{\max}$.

   Tātad, vispirms visiem pozitīviem μ_{i1} vai μ_{i2} ir jānosaka varbūtību vērtības. Visu pārējo varbūtību vērtībām jābūt vienādām, kas seko no (3.7) un no tā, ka $\mu_{j1} = 0$ un $\mu_{j2} = 0$ visām p_j, kuras nav iespējams noteikt. Varbūtību p_j vērtības var noteikt ar (3.8) palīdzību, izmantojot faktu, ka visas nezināmās varbūtības ir vienādas (kā minēts augstāk, tas izriet no (3.7)). Tagad visas apskatāmas konfigurācijas varbūtību vērtības ir zināmas.

Ir jāpārbauda, vai varbūtību summa ir vienāda ar vieninieku un vai, saskaņā ar (3.10), visas μ_{ij} vērtības ir nenegatīvas. Lai pārbaudītu pēdējo faktu, ir jāatrisina no (3.7) iegūtā lineāru vienādojumu sistēma.

Ja visi nosacījumi ir apmierināti, tad atbilstošajā punktā ir jāizrēķina entropijas vērtība. Pēc visu konfigurāciju apskatīšanas jāizvēlas vislielākā vērtībā, kura ir meklējamā entropijas augšējā robeža $H^{\max}$.

Minimizēšanas uzdevuma (3.11) risinājums ir analoģisks uzdevuma (3.5) risinājumam.

$$H^{\min} : -\sum_{i=1}^{n} p_i \ln p_i \rightarrow \min$$

ar sekojošajiem ierobežojumiem:

$$-p_i \le -p_i^{\min}, \quad p_i \le p_i^{\max}, \qquad i = \overline{1,n}. \tag{3.11}$$

$$\sum_{i=1}^{n} p_i = 1$$

Lai atrisinātu uzdevumu (3.11), tas ir jāpārveido maksimizēšanas problēmā, pareizinot mērķa funkciju ar -1. Pēc šīs operācijas uzdevums (3.11) izskatās šādi:

$$H^{\min} : \sum_{i=1}^{n} p_i \ln p_i \rightarrow \max$$

ar sekojošajiem ierobežojumiem:

$$-p_i \le -p_i^{\min}, \quad p_i \le p_i^{\max}, \qquad i = \overline{1,n}. \tag{3.12}$$

$$\sum_{i=1}^{n} p_i = 1$$

Uzdevuma (3.12) risinājums ir līdzīgs uzdevuma (3.5) risinājumam, izņemot pēdējo soli, kurā ir jāizvēlas vismazākā vērtība, kura atbilst intervālu entropijas apakšējai robežai $H^{\min}$. Turklāt, Lagranža vienādojums un parciālie atvasinājumi arī atšķiras:

Uzdevuma (3.12) Lagranža vienādojums ir šāds:

$$L(p_1,\ldots,p_n,\lambda,\mu_{11},\mu_{12},\mu_{21},\mu_{22},\ldots,\mu_{n1},\mu_{n2},) = \sum_{i=1}^{n} p_i \ln p_i + \lambda\left(\sum_{i=1}^{n} p_i - 1\right) + \sum_{i=1}^{n}\left(\mu_{i1}\left(p_i - p_i^{\min}\right) + \mu_{i2}\left(p_i^{\max} - p_i\right)\right) \tag{3.13}$$

Vienādojuma (3.13) parciālie atvasinājumi ir sekojošie:

$$\frac{\partial L}{\partial p_i} = \ln p_i + 1 + \lambda + \mu_{i1} - \mu_{i2}, \; i = \overline{1,n}.$$

Tagad pierādīsim, ka iegūtā entropija ir aditīva.

3.1.2.3. Intervālu entropijas aditivitātes pierādījums

Lai pierādītu, ka pēc augstāk aprakstītas metodes izrēķinātā entropija ir aditīva, apskatīsim sekojošus gadījumus. Saskaņā ar [Shannon&Weaver, 1949], ja divas sistēmas *x* un *y* ir neatkarīgas, tad $H(x,y) = H(x) + H(y)$.

Ja entropijas vērtības ir intervālu: $H(x) = [H_x^{\min}, H_x^{\max}]$ un $H(y) = [H_y^{\min}, H_y^{\max}]$, tad lai noskaidrotu apvienotas sistēmas entropiju ar minimālo iespējamu entropijas vērtību, ir jāsasummē atbilstošu intervālu apakšējas robežas. Ja eksistē sistēma ar mazāku entropijas vērtību, tas nozīmē, ka vai nu sistēmas *x* un *y* nav neatkarīgas, vai nu vērtības $H_x^{\min}$ vai $H_y^{\min}$ ir izvelētas nepareizi. Pieņemsim, ka ir tādas vērtības, ka $`H_x^{\min} + `H_y^{\min} \le H_x^{\min} + H_y^{\min}$, tad $`H_x^{\min} \le H_x^{\min}$ vai $`H_y^{\min} \le H_y^{\min}$, bet tā kā $H_x^{\min}$ un $H_y^{\min}$ ir individuālo sistēmu minimālās entropijas vērtības, tad: $`H_x^{\min} = H_x^{\min}$ un $`H_y^{\min} = H_y^{\min}$.

Pierādījums augšējai intervāla robežai ir analoģisks.

Tādējādi, mēs pierādījām, ka ja $H(x) = [H_x^{\min}, H_x^{\max}]$, $H(y) = [H_y^{\min}, H_y^{\max}]$ un sistēmas *x* un *y* ir neatkarīgas, tad:

$$H(x,y) = [H_x^{\min} + H_y^{\min}, H_x^{\max} + H_y^{\max}]. \tag{3.14}$$

Tagad apskatīsim dažus aprakstītā risinājuma izmantošanas piemērus, kuros tiek rēķināta entropija sistēmām ar intervālu varbūtībām.

3.1.2.4. Pirmais intervālu entropijas pielietošanas piemērs

Šajā piemērā tiks noteikta intervālu entropijas vērtība $H = [H^{\min}, H^{\max}]$ sistēmai ar diviem stāvokļiem, kuru varbūtības ir $p_1 = [0.2; 0.4]$ un $p_2 = [0.6; 0.8]$.

Ierobežojumi ir noteikti sekojošajā veidā:

$$\begin{array}{ll} -p_1 \le -0.2; & -p_2 \le -0.6 \\ p_1 \le 0.4; & p_2 \le 0.8 \\ p_1 + p_2 = 1 & \end{array} \quad . \tag{3.15}$$

Vispirms izrēķināsim augšējās robežas vērtību $H^{\max} : -p_1 \ln p_1 - p_2 \ln p_2 \to \max$. Šajā gadījumā Lagranža vienādojums izskatās sekojošajā veidā:

$$\begin{aligned} & L(p_1, p_2, \lambda, \mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}) = \\ & \qquad -p_1 \ln p_1 - p_2 \ln p_2 + \lambda(p_1 + p_2 - 1) + \\ & \qquad +\mu_{11}(p_1 - 0.2) + \mu_{12}(0.4 - p_1) + \mu_{21}(p_2 - 0.6) + \mu_{22}(0.8 - p_2) \end{aligned}$$

Lagranža vienādojuma parciālie atvasinājumi ir sekojošie:

$$\frac{\partial L}{\partial p_1} = -\ln p_1 - 1 + \lambda + \mu_{11} - \mu_{12}$$
$$\frac{\partial L}{\partial p_2} = -\ln p_2 - 1 + \lambda + \mu_{21} - \mu_{22}$$

Tātad, ir jāatrisina sekojošais uzdevums:

$$\left| \begin{array}{l} -\ln p_1 - 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ -\ln p_2 - 1 + \lambda + \mu_{21} - \mu_{22} = 0 \\ \mu_{11}(p_1 - 0.2) = 0 \\ \mu_{12}(0.4 - p_1) = 0 \\ \mu_{21}(p_2 - 0.6) = 0 \\ \mu_{22}(0.8 - p_2) = 0 \\ -p_1 \le -0.2; \quad -p_2 \le -0.6 \\ p_1 \le 0.4; \qquad p_2 \le 0.8 \\ p_1 + p_2 = 1 \\ \mu_{11}, \mu_{12}, \mu_{21}, \mu_{22} \ge 0 . \end{array} \right. \tag{3.16}$$

Tagad ir jāapskata 2^4=16 gadījumi, kuri atšķiras ar μ_{ij} vērtībām:

1) $\mu_{11} = \mu_{12} = \mu_{21} = \mu_{22} = 0$. Tātad, $p_1 = p_2 = \frac{1}{2}$, kas ir pretrunā ar (3.15)
2) $\mu_{11} = \mu_{12} = \mu_{21} = 0$. Tātad, $p_1 = 0.2; p_2 = 0.8$, kas ir potenciāls risinājums
3) $\mu_{11} = \mu_{12} = \mu_{22} = 0$. Tātad, $p_1 = 0.4; p_2 = 0.6$, kas ir potenciāls risinājums
4) $\mu_{11} = \mu_{12} = 0$. Tātad, $p_2 = 0.8; p_2 = 0.6$, pretruna
5) $\mu_{11} = \mu_{21} = \mu_{22} = 0$. Tātad, $p_1 = 0.4; p_2 = 0.6$, kas ir potenciāls risinājums
6) $\mu_{11} = \mu_{21} = 0$. Tātad, $p_1 = 0.4; p_2 = 0.8$, kas ir pretrunā ar (3.15)
7) $\mu_{11} = \mu_{22} = 0$. Tātad, $p_1 = 0.4; p_2 = 0.6$, kas ir potenciāls risinājums
8) $\mu_{11} = 0$. Tātad, $p_1 = 0.4; p_2 = 0.8; p_2 = 0.6$, pretruna
9) $\mu_{12} = \mu_{21} = \mu_{22} = 0$. Tātad, $p_1 = 0.2; p_2 = 0.8$, kas ir potenciāls risinājums
10) $\mu_{12} = \mu_{21} = 0$. Tātad, $p_1 = 0.2; p_2 = 0.8$, kas ir potenciāls risinājums

11) $\mu_{12} = \mu_{22} = 0$. Tātad, $p_1 = 0.2; p_2 = 0.6$, kas ir pretrunā ar (3.15)
12) $\mu_{12} = 0$. Tātad, $p_2 = 0.6; p_2 = 0.8$, pretruna
13) $\mu_{21} = \mu_{22} = 0$. Tātad, $p_1 = 0.2; p_1 = 0.4$, pretruna
14) $\mu_{21} = 0$. Tātad, $p_1 = 0.2; p_1 = 0.4$, pretruna
15) $\mu_{22} = 0$. Tātad, $p_1 = 0.2; p_1 = 0.4$, pretruna
16) $\mu_{11}\mu_{12}\mu_{21}\mu_{22} \neq 0$. Tātad, $p_1 = 0.2; p_1 = 0.4$, pretruna

Konfigurācijas <2>, <3>, <5>, <7>, <9>, <10> ir potenciālie risinājumi. Lai pārbaudītu, vai nosacījums (3.16) ir apmierināts, ir jāatrisina sekojošā sistēma attiecībā uz μ_{ij}:

$$\begin{cases} -\ln p_1 - 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ -\ln p_2 - 1 + \lambda + \mu_{21} - \mu_{22} = 0 \end{cases} \tag{3.17}$$

Vispirms pārbaudīsim konfigurācijas <2>, <9> un <10>, kuras atbilst kortežam (0.2; 0.8). Šajā gadījumā sistēma (3.17) izskatās sekojošajā veidā:

$$\begin{cases} 1.609 - 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ 0.22 - 1 + \lambda + \mu_{21} - \mu_{22} = 0 \end{cases}$$

Ir viegli parādīt, ka konfigurācija <2> neapmierina nosacījumu (3.16), jo $\mu_{22} < 0$; konfigurācija <9> neapmierina nosacījumu (3.16), jo $\mu_{11} < 0$; neapmierina nosacījumu (3.16), jo $\mu_{11} < 0$ vai $\mu_{22} < 0$.

Tagad pārbaudīsim konfigurācijas <3>, <5>, <7>, kuras atbilst kortežam (0.4;0.6). Šajā gadījumā sistēma (3.17) izskatās sekojošajā veidā:

$$\begin{cases} 0.9 - 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ 0.51 - 1 + \lambda + \mu_{21} - \mu_{22} = 0 \end{cases}$$

Ir viegli parādīt, ka konfigurācijas <3>, <5>, <7> apmierina visus nosacījumus, jo var atrast attiecīgās nenegatīvas μ_{ij} vērtības, lai kompensētu starpību divās vienādībās katrai λ konstantei.

Tādejādi, meklējamā entropijas augšējā robeža ir vienāda ar

$$H^{\max} = H(0.4, 0.6) = -0.4 \ln 0.4 - 0.6 \ln 0.6 \approx 0.673.$$

Tagad noteiksim intervālu entropijas apakšējās robežas vērtību $H^{\min}: -p_1 \ln p_1 - p_2 \ln p_2 \to \min$. Lai to izdarītu, minimizēšanas problēma ir jāpārveido par maksimizēšanas problēmu. Rezultātā mēs iegūstam $H^{\min}: p_1 \ln p_1 + p_2 \ln p_2 \to \max$.

Šīs problēmas risinājums ir analoģisks maksimizēšanas uzdevumam. Risināmā problēma tiek definēta šādi:

$$\left| \begin{array}{l} \ln p_1 + 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ \ln p_2 + 1 + \lambda + \mu_{21} - \mu_{22} = 0 \\ \mu_{11}(p_1 - 0.2) = 0 \\ \mu_{12}(0.4 - p_1) = 0 \\ \mu_{21}(p_2 - 0.6) = 0 \\ \mu_{22}(0.8 - p_1) = 0 \\ -p_1 \le -0.2; \quad -p_2 \le -0.6 \\ p_1 \le 0.4; \qquad p_2 \le 0.8 \\ p_1 + p_2 = 1 \\ \mu_{11}, \mu_{12}, \mu_{21}, \mu_{22} \ge 0 . \end{array} \right.$$

Potenciālie risinājumi ir tādi paši un tie ir <2>, <3>, <5>, <7>, <9>, <10>.

Vispirms pārbaudīsim konfigurācijas <2>, <9> un <10>, kuras atbilst kortežam (0.2; 0.8). Šajā gadījumā vienādojumu sistēma izskatās šādi:

$$\begin{cases} -1.609 + 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ -0.22 + 1 + \lambda + \mu_{21} - \mu_{22} = 0 \end{cases}$$

Ir viegli parādīt, ka konfigurācijas <2>, <9>, <10> apmierina visus nosacījumus, jo var atrast attiecīgās nenegatīvas μ_{ij} vērtības, lai kompensētu starpību divās vienādībās katrai konstantei λ.

Tagad pārbaudīsim konfigurācijas <3>, <5>, <7>, kuras atbilst kortežam (0.4;0.6). Šajā gadījumā sistēma (3.17) izskatās sekojošajā veidā:

$$\begin{cases} -0.9 + 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ -0.51 + 1 + \lambda + \mu_{21} - \mu_{22} = 0 \end{cases}$$

Ir viegli parādīt, ka konfigurācija <3> neapmierina nosacījumu (3.16), jo $\mu_{21} < 0$; konfigurācija <5> neapmierina nosacījumu (3.16), jo $\mu_{12} < 0$; konfigurācija <7> neapmierina nosacījumu (3.16), jo $\mu_{12} < 0$ vai $\mu_{21} < 0$.

Tādejādi, meklējamā entropijas apakšējā robeža ir vienāda ar

$$H^{\min} = H(0.2, 0.8) = -0.2 \ln 0.2 - 0.8 \ln 0.8 \approx 0.5 .$$

Līdz ar to ir iegūts sekojošais problēmas risinājums: $H = [H^{\min}, H^{\max}] = [0.5, 0.673]$

3.1.2.5. Otrais intervālu entropijas pielietošanas piemērs

Šajā piemērā tiks noteikta intervālu entropijas vērtība $H = [H^{\min}, H^{\max}]$ sistēmai ar trim stāvokļiem, kuru varbūtības ir $p_1 = [0.4; 0.6]$, $p_2 = [0.1; 0.3]$ un $p_3 = [0.5; 0.7]$.

Ierobežojumi ir noteikti sekojošajā veidā:

$$\begin{aligned}
&-p_1 \le -0.4; \quad -p_2 \le -0.1; \quad -p_3 \le -0.5 \\
&p_1 \le 0.6; \quad\quad p_2 \le 0.3; \quad\quad p_3 \le 0.7 \\
&p_1 + p_2 + p_3 = 1
\end{aligned} \tag{3.18}$$

Vispirms izrēķināsim augšējās robežas vērtību $H^{\max} : -p_1 \ln p_1 - p_2 \ln p_2 \to \max$. Šajā gadījumā Lagranža vienādojums izskatās sekojošajā veidā:

$$\begin{aligned}
&L(p_1, p_2, p_3, \lambda, \mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}, \mu_{31}, \mu_{32}) = \\
&\quad -p_1 \ln p_1 - p_2 \ln p_2 - p_3 \ln p_3 + \lambda(p_1 + p_2 + p_3 - 1) + \\
&\quad +\mu_{11}(p_1 - 0.4) + \mu_{12}(0.6 - p_1) + \mu_{21}(p_2 - 0.1) + \mu_{22}(0.3 - p_2) + \\
&\quad +\mu_{31}(p_3 - 0.5) + \mu_{32}(0.7 - p_3)
\end{aligned}$$

Lagranža vienādojuma parciālie atvasinājumi ir šādi:

$$\begin{aligned}
\frac{\partial L}{\partial p_1} &= -\ln p_1 - 1 + \lambda + \mu_{11} - \mu_{12} \\
\frac{\partial L}{\partial p_2} &= -\ln p_2 - 1 + \lambda + \mu_{21} - \mu_{22} \\
\frac{\partial L}{\partial p_3} &= -\ln p_3 - 1 + \lambda + \mu_{31} - \mu_{32}
\end{aligned}$$

Tātad, ir jāatrisina sekojošais uzdevums:

$$\left|\begin{aligned}
&-\ln p_1 - 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\
&-\ln p_2 - 1 + \lambda + \mu_{21} - \mu_{22} = 0 \\
&-\ln p_3 - 1 + \lambda + \mu_{31} - \mu_{32} = 0 \\
&\mu_{11}(p_1 - 0.4) = 0 \\
&\mu_{12}(0.6 - p_1) = 0 \\
&\mu_{21}(p_2 - 0.1) = 0 \\
&\mu_{22}(0.3 - p_2) = 0 \\
&\mu_{31}(p_3 - 0.5) = 0 \\
&\mu_{32}(0.7 - p_3) = 0 \\
&-p_1 \le -0.4; \quad -p_2 \le -0.1; \quad -p_3 \le -0.5 \\
&p_1 \le 0.6; \quad\quad p_2 \le 0.3; \quad\quad p_3 \le 0.7 \\
&p_1 + p_2 + p_3 = 1 \\
&\mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}, \mu_{31}, \mu_{32} \ge 0 .
\end{aligned}\right. \tag{3.19}$$

Tagad ir jāapskata 2^6=64 gadījumi, bet kā var pamanīt, nav jēgas apskatīt gadījumus, kad $\mu_{i1} = 0$ un $\mu_{i2} = 0$ jebkuram $i = \overline{1,n}$. Tādējādi, mēs varam ierobežoties ar 3^3=27 gadījumu apskatīšanu, kuri atšķiras ar μ_{ij} vērtībām:

1) $\mu_{11} = \mu_{12} = \mu_{21} = \mu_{22} = \mu_{31} = \mu_{32} = 0$. Tātad, $p_1 = p_2 = p_3 = \frac{1}{3}$, kas ir pretrunā ar (3.18)
2) $\mu_{32} \neq 0$. Tātad, $p_1 = p_2 = 0.15; p_3 = 0.7$, kas ir pretrunā ar (3.18)
3) $\mu_{31} \neq 0$. Tātad, $p_1 = p_2 = 0.25; p_3 = 0.5$, kas ir pretrunā ar (3.18)
4) $\mu_{22} \neq 0$. Tātad, $p_1 = p_3 = 0.35; p_2 = 0.3$, kas ir pretrunā ar (3.18)
5) $\mu_{22} \neq 0, \mu_{32} \neq 0$. Tātad, $p_1 = 0; p_2 = 0.3; p_3 = 0.7$, kas ir pretrunā ar (3.18)
6) $\mu_{22} \neq 0, \mu_{31} \neq 0$. Tātad, $p_1 = 0.2; p_2 = 0.3; p_3 = 0.5$, kas ir pretrunā ar (3.18)
7) $\mu_{21} \neq 0$. Tātad, $p_1 = p_3 = 0.45; p_2 = 0.1$, kas ir pretrunā ar (3.18)
8) $\mu_{21} \neq 0, \mu_{32} \neq 0$. Tātad, $p_1 = 0.2; p_2 = 0.1; p_3 = 0.7$, kas ir pretrunā ar (3.18)
9) $\mu_{21} \neq 0, \mu_{31} \neq 0$. Tātad, $p_1 = 0.4; p_2 = 0.1; p_3 = 0.5$, kas ir potenciāls risinājums
10) $\mu_{12} \neq 0$. Tātad, $p_1 = 0.6; p_2 = p_3 = 0.2$, kas ir pretrunā ar (3.18)
11) $\mu_{12} \neq 0, \mu_{32} \neq 0$. Tātad, $p_1 = 0.6; p_3 = 0.7$, kas ir pretrunā ar (3.18)
12) $\mu_{12} \neq 0, \mu_{31} \neq 0$. Tātad, $p_1 = 0.6; p_3 = 0.5$, kas ir pretrunā ar (3.18)
13) $\mu_{12} \neq 0, \mu_{22} \neq 0$. Tātad, $p_1 = 0.6; p_2 = 0.3; p_3 = 0.1$, kas ir pretrunā ar (3.18)
14) $\mu_{12} \neq 0, \mu_{22} \neq 0, \mu_{32} \neq 0$. Tātad, $p_1 = 0.6; p_2 = 0.3; p_3 = 0.7$, kas ir pretrunā ar (3.18)
15) $\mu_{12} \neq 0, \mu_{22} \neq 0, \mu_{31} \neq 0$. Tātad, $p_1 = 0.6; p_2 = 0.3; p_3 = 0.5$, kas ir pretrunā ar (3.18)
16) $\mu_{12} \neq 0, \mu_{21} \neq 0$. Tātad, $p_1 = 0.6; p_2 = 0.1; p_3 = 0.3$, kas ir pretrunā ar (3.18)
17) $\mu_{12} \neq 0, \mu_{21} \neq 0, \mu_{32} \neq 0$. Tātad, $p_1 = 0.6; p_2 = 0.1; p_3 = 0.7$, kas ir pretrunā ar (3.18)
18) $\mu_{12} \neq 0, \mu_{21} \neq 0, \mu_{31} \neq 0$. Tātad, $p_1 = 0.6; p_2 = 0.1; p_3 = 0.5$, kas ir pretrunā ar (3.18)
19) $\mu_{11} \neq 0$. Tātad, $p_1 = 0.4; p_2 = p_3 = 0.3$, kas ir pretrunā ar (3.18)
20) $\mu_{11} \neq 0, \mu_{32} \neq 0$. Tātad, $p_1 = 0.4; p_3 = 0.7$, kas ir pretrunā ar (3.18)
21) $\mu_{11} \neq 0, \mu_{31} \neq 0$. Tātad, $p_1 = 0.4; p_2 = 0.1; p_3 = 0.5$, kas ir potenciāls risinājums
22) $\mu_{11} \neq 0, \mu_{22} \neq 0$. Tātad, $p_1 = 0.4; p_2 = 0.3; p_3 = 0.3$, kas ir pretrunā ar (3.18)
23) $\mu_{11} \neq 0, \mu_{22} \neq 0, \mu_{32} \neq 0$. Tātad, $p_1 = 0.4; p_2 = 0.3; p_3 = 0.7$, kas ir pretrunā ar (3.18)
24) $\mu_{11} \neq 0, \mu_{22} \neq 0, \mu_{31} \neq 0$. Tātad, $p_1 = 0.4; p_2 = 0.3; p_3 = 0.5$, kas ir pretrunā ar (3.18)
25) $\mu_{11} \neq 0, \mu_{21} \neq 0$. Tātad, $p_1 = 0.4; p_2 = 0.1; p_3 = 0.5$, kas ir potenciāls risinājums
26) $\mu_{11} \neq 0, \mu_{21} \neq 0, \mu_{32} \neq 0$. Tātad, $p_1 = 0.4; p_2 = 0.1; p_3 = 0.7$, kas ir pretrunā ar (3.18)
27) $\mu_{11} \neq 0, \mu_{21} \neq 0, \mu_{31} \neq 0$. Tātad, $p_1 = 0.4; p_2 = 0.1; p_3 = 0.5$, kas ir potenciāls risinājums

Konfigurācijas <9>, <21>, <25>, <27> ir potenciālie risinājumi. Lai pārbaudītu, vai nosacījums (3.19) ir apmierināts, ir jāatrisina sekojošā sistēma attiecībā uz μ_{ij}:

$$\begin{cases} -\ln p_1 - 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ -\ln p_2 - 1 + \lambda + \mu_{21} - \mu_{22} = 0 \\ -\ln p_3 - 1 + \lambda + \mu_{31} - \mu_{32} = 0 \end{cases}$$

Visi potenciālie risinājumi atbilst kortežam (0.4; 0.1; 0.5). Līdz ar to, var izmantot sekojošo sistēmu, lai pārbaudītu vai nosacījums (3.19) ir apmierināts.

$$\begin{cases} 0.916 - 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ 2.303 - 1 + \lambda + \mu_{21} - \mu_{22} = 0 \\ 0.693 - 1 + \lambda + \mu_{31} - \mu_{32} = 0 \end{cases}$$

Ir viegli parādīt, ka konfigurācija <9> neapmierina nosacījumu (3.19), jo $\mu_{21} < 0$; konfigurācija <25> neapmierina nosacījumu (3.19), jo $\mu_{21} < 0$ un $\mu_{11} < 0$.

Ir viegli parādīt, ka konfigurācijas <21> un <27> apmierina (3.19), jo var atrast attiecīgās nenegatīvas μ_{ij} vērtības, lai kompensētu starpību divās vienādībās katrai konstantei λ.

Tādejādi, meklējamā entropijas augšējā robeža ir vienāda ar

$$H^{\max} = H(0.4, 0.6) = -0.4 \ln 0.4 - 0.6 \ln 0.6 \approx 0.673 .$$

Tagad noteiksim intervālu entropijas apakšējās robežas vērtību $H^{\min} : -p_1 \ln p_1 - p_2 \ln p_2 - p_3 \ln p_3 \to \min$. Lai to izdarītu, minimizēšanas problēma ir jāpārveido par maksimizēšanas problēmu. Rezultātā mēs iegūstam $H^{\min} : p_1 \ln p_1 + p_2 \ln p_2 + p_3 \ln p_3 \to \max$.

Šīs problēmas risinājums ir gandrīz tāds pats kā $H^{\max}$ vērtībai. Risināmā problēma tiek definēta sekojošajā veidā:

$$\left| \begin{array}{l} \ln p_1 + 1 + \lambda + \mu_{11} - \mu_{12} = 0 \\ \ln p_2 + 1 + \lambda + \mu_{21} - \mu_{22} = 0 \\ \ln p_3 + 1 + \lambda + \mu_{31} - \mu_{32} = 0 \\ \mu_{11}(p_1 - 0.4) = 0 \\ \mu_{12}(0.6 - p_1) = 0 \\ \mu_{21}(p_2 - 0.1) = 0 \\ \mu_{22}(0.3 - p_2) = 0 \\ \mu_{31}(p_3 - 0.5) = 0 \\ \mu_{32}(0.7 - p_3) = 0 \\ -p_1 \le -0.4; \quad -p_2 \le -0.1; \quad -p_3 \le -0.5 \\ p_1 \le 0.6; \quad p_2 \le 0.3; \quad p_3 \le 0.7 \\ p_1 + p_2 + p_3 = 1 \\ \mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}, \mu_{31}, \mu_{32} \ge 0 . \end{array} \right.$$

Potenciālie risinājumi ir tādi paši un tie ir <9>, <21>, <25> un <27>. Pārbaudīsim, vai tie apmierina visus nosacījumus. Izmantosim sekojošo sistēmu:

$$\begin{cases} -0.916+1+\lambda+\mu_{11}-\mu_{12}=0 \\ -2.303+1+\lambda+\mu_{21}-\mu_{22}=0 \\ -0.693+1+\lambda+\mu_{31}-\mu_{32}=0 \end{cases}$$

Ir viegli parādīt, ka konfigurācija <9> neapmierina nosacījumu (3.19), jo $\mu_{31}<0$; konfigurācija <21> neapmierina nosacījumu (3.19), jo $\mu_{11}<0$ un $\mu_{31}<0$;

Ir viegli parādīt, ka konfigurācijas <25> un <27> apmierina (3.19), jo var atrast attiecīgās nenegatīvas μ_{ij} vērtības, lai kompensētu starpību divās vienādībās katrai konstantei λ.

Tādejādi, meklējamā entropijas apakšējā robeža ir vienāda ar

$$H^{\min}=H(0.4,0.1,0.5)=-0.4\ln 0.4-0.1\ln 0.1-0.5\ln 0.5\approx 0.943$$

Līdz ar to ir iegūts sekojošais problēmas risinājums:

$$H=[H^{\min},H^{\max}]=[0.943,0.943]=0.943$$

3.1. apakšnodaļā ir aprakstīts adaptīvais tīkls ANGIE izplūdušo granulu apstrādei un intervālu entropija, šie līdzekļi ir izstrādāti strādājot pie turpmākajās apakšnodaļās aprakstītās lēmumu pieņemšanas metodes.

3.2. Vispārējs izstrādātās metodes apraksts

Sekojošajās nodaļās tiks aprakstīta uz f-granulām balstīta lēmumu pieņemšanas metode [Vališevskis, 2004]. F-granulas veido šīs metodes kodolu, jo ar to palīdzību tiek aprakstītas alternatīvas. Kā jau tika minēts, viena no f-granulu svarīgākajām īpašībām ir tas, ka tās var izmantojot gan izplūdušās, gan nedeterminētas informācijas aprakstīšanai. F-granulas var uzskatīt par izplūdušo vērtību varbūtību sadalījumu.

Viena no aprakstāmās metodes īpašībām ir tas, ka tajā alternatīva tiek vērtēta nevis pamatojoties uz tās kritēriju vērtībām, bet uz varbūtības, ka alternatīvas kritēriju vērtības būs tuvu vēlamajām vērtībām. Proti, tā ir varbūtība, ka alternatīva būs līdzīga vēlamajai alternatīvai, kura tika nodefinēta pirms alternatīvu salīdzināšanas. Šīs varbūtības ir intervālu, kas padara alternatīvu salīdzināšanas uzdevumu par netriviālu problēmu, jo ir iespējams gadījums kad alternatīvas ir *nesalīdzināmas*. Tādējādi, vispārīgā gadījumā ir iespējama tikai daļēja alternatīvu sakārtošana. Tas ir saskaņā ar mūsu intuīciju, jo no sistēmas nevar gaidīt precīzu atbildi, ja tai tika uzdots neprecīzs jautājums, kurā ir divu tipu nenoteiktības. Tas atgādina situāciju, kura ir tādās lēmumu pieņemšanas metodēs kā PROMETHEE [Brans u.c., 1986] vai ELECTRE [Roy, 1991]. Parasti nav iespējams iegūt pilnu alternatīvu sakārtojumu bez informācijas zušanas, jo risināmais lēmuma pieņemšanas uzdevums var būt pārāk sarežģīts, lai izmantojamā metode varētu atrast kompromisu starp konfliktējošām alternatīvām. Tādējādi, parasti tiek veikts daļējs alternatīvu sakārtojums un kompromisa atrašanas uzdevums tiek deleģēts lēmumu pieņemošai personai.

Kā jebkurā citā lēmumu pieņemšanas metodē, kura ir paredzēta izmantošanai vidē ar nenoteiktību, optimalitātes jēdzienu nevar stingri nodefinēt. Tādējādi, galvenais uzdevums ir nevis optimālas alternatīvas noteikšana, bet tādu līdzekļu kā alternatīvu sakārtošana un jūtīguma analīze izmantošana, lai palīdzētu lēmumu pieņemošajai personai pieņemt labāku un informatīvāku lēmumu.

Šī metode atšķiras no līdzīgām lēmumu pieņemšanas metodēm vairākos aspektos. Galvenā atšķirība ir tas, ka piedāvāto lēmumu pieņemšanas metodi var izmantot kad pieejamā informācija ir ļoti nenoteikta. Tā var būt tik nenoteikta, ka nav iespējams izveidot kritēriju vērtību tabulu, vai aprakstīt problēmas apgabalu ar neizplūdušas metodes palīdzību.

Kā jau tika minēts, piedāvātajā metodē tiek izmantotas tā sauktas izplūdušās f-granulas. Tās ļauj alternatīvu aprakstā izmantot gan izplūdušo, gan nedeterminētu informāciju. To var uzskatīt par kompromisu starp neizplūdušām lēmumu pieņemšanas metodēm, piemēram, rakstos [Brans u.c., 1986; Roy, 1991; Clemen&Reilly, 2004 un Saaty, 1980] aprakstītajām, kuras neļauj izmantot izplūdušo informāciju alternatīvu aprakstā un tādām metodēm kā PRUF [Zadeh, 1978], kurās izplūdusī informācija tiek izmantota tik intensīvi, ka skaitliski šis uzdevums kļūst pārāk sarežģīts.

Kā redzams 3.7. attēlā, piedāvāto metodi var sadalīt vairākos soļos:

1. Noteikt alternatīvu un kritēriju kopu.
2. Aprakstīt alternatīvas ar izplūdušo granulu palīdzību.
3. Katram kritērijam izrēķināt varbūtību, ka dotajai alternatīvai šī kritērija vērtība būs vēlama.
4. Noteikt alternatīvu informatīvumu.
5. Nomodelēt preferences.
6. Izveidot alternatīvu daļējo vai pilno sakārtojumu.
7. Veikt jutīguma analīzi.

8. Izanalizēt iegūtos rezultātus.
9. Parādīt rezultātus.

Turpmākās apakšnodaļās šie soļi tiek apskatīti sīkāk.

3.3. Alternatīvu un kritēriju kopas noteikšana

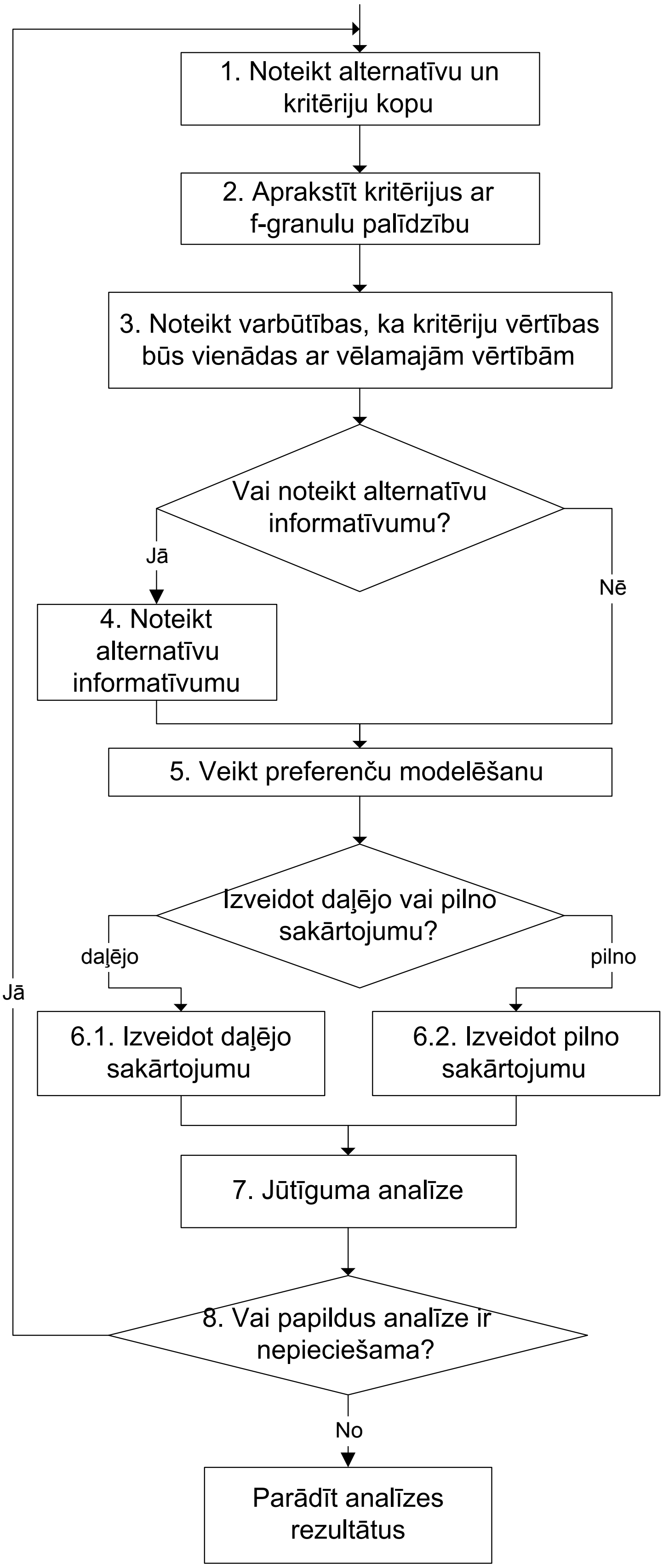
Šajā solī tiek izmantotas citu autoru darbos aprakstītas metodes [Clemen&Reilly, 2004; Mustajoki u.c., 2001; Saaty, 1980; Salo&Hamalainen, 1995;]. Vispirms ir jānosaka kādas ir alternatīvas (tiek apskatīts gadījums, kad to skaits ir galīgs) un kādi kritēriji tiek izmantoti alternatīvu vērtēšanai. Kā citās metodēs, alternatīvu kopu var noteikt rīkojot prātavētras (*brainstorming* – angļu val.) sesijas vai rūpīgi analizējot lēmumu pieņemšanas uzdevumu. Turklāt, var izmantot morfoloģisko analīzi (sk. 2.1.2. paragrāfu).

Kritēriji ir tās alternatīvu īpašības, kuras mūs interesē visvairāk. Ar dažādu kritēriju palīdzību mēs varam atšķirt labāku risinājumu, jeb alternatīvu, no sliktāka. Un otrādi, risinot lēmuma pieņemšanas uzdevumu, mums jāatceras, ka analīzē tiek apskatīti tikai izvēlētie kritēriji. Piemēram, apskatot elektrostacijas būvēšanas projektu, daži no kritērijiem varētu būt šīs elektrostacijas prognozējama ietekme uz vidi, elektrostacijas būvēšanas sarežģītība, ienesīgums un līdzīgi kritēriji.

3.4. Problēmas apgabala aprakstīšana ar izplūdušo granulu palīdzību

Pēc alternatīvu un kritēriju kopas noteikšanas ir jāizveido liecības, kuras sastāv no f-granulām, kā aprakstīts 1.3.3.1. un 3.1.1. paragrāfos. Katra liecība satur granulas, kuras apraksta kāda nedeterminēta notikuma iespējamus iznākumus un šiem iznākumiem atbilstošas kritērija vērtības. Alternatīvu apraksts ir liecību kopa, kura nosaka iespējamas alternatīvas kritēriju vērtības. Kā būs redzams zemāk, alternatīvas tiek salīdzinātas balstoties uz varbūtībām, ka to kritēriju vērtības būs tuvu vēlamajām vērtībām.

Informācija par vienu alternatīvu var būt sadalīta vairākās liecībās. Tomēr, katra liecība apraksta tikai vienu kritēriju. Pēc liecību kopas noteikšanas un apstiprināšanas (ja ir vairākas par lēmumu atbildīgas personas) var iegūt informāciju un šīm liecībām, nosakot varbūtību, ka liecībā aprakstītā kritērija vērtība būs vienāda ar kādu vērtību.



3.7. att. Izstrādātās lēmumu pieņemšanas metodes etapi

Kā jau tika minēts, nosacītas f-granulas var uzskatīt par izplūdušo vērtību varbūtību sadalījumu. Citiem vārdiem, tās uzliek varbūtiskus ierobežojumus uz kritēriju iespējamības sadalījumu. Izplūdušās f-granulas var nodefinēt ar JA-TAD likumu palīdzību. Kreisajā likumu pusē tiek aprakstīti iespējami nedeterminēti iznākumi, jeb iespējamas situācijas, kuras ir nedeterminētas un no kurām ir atkarīgas kritēriju vērtības. Iznākumi, kuri attiecās uz vienu un to pašu parādību tiek grupēti vienā liecībā. Likumu labajā pusē ir izplūdušās kritērija vērtības, kuras uzstāda iespējamības ierobežojumu uz tā vērtībām.

Apskatīsim nelielu piemēru. Pieņemsim, ka ir jānovērtē būvniecības projekta risks, kurš ir atkarīgs no augsnes tipa, kurš nav precīzi zināms. Šajā gadījumā risks ir kritērijs un augsnes tips ir nedeterminēta parādība. Pēc konsultēšanās ar ekspertiem un pēc augsnes paraugu pārbaudes tika secināts, ka šo problēmu var aprakstīt sekojošajā veidā:

JA (*augsne* ir akmeņaina) ar varbūtību *0,8* TAD *risks* ir ĻOTI ZEMS
JA (*augsne* ir smilšaina) ar varbūtību *0,15* TAD *risks* ir VIRS VĪDĒJA
JA (*augsne* ir purvaina) ar varbūtību *0,05* TAD *risks* ir ĻOTI AUGSTS

Katra liecība ir izplūdušo vērtību varbūtību sadalījums. Piemēram, augstāk minēto liecību var interpretēt kā sekojošo varbūtību sadalījumu:

P(*risks* ir ĻOTI ZEMS) = *0,8*
P(*risks* ir VIRS VĪDĒJA) = *0,15*
P(*risks* ir ĻOTI AUGSTS) = *0,05*

Ja ir nodefinēta šādu liecību kopa, kura apraksta kritēriju "risks" un, iespējams, arī citus kritērijus, var noteikt kāda ir varbūtība, ka dotajam būvniecības projektam riska vērtība būs ZEMA. Varbūtības *P(risks ir ZEMS)* vērtība ir intervālu. Norādījumus šīs varbūtības rēķināšanai var atrast 1.3.3.1. apakšparagrāfā.

Piedāvātajā metodē tas ir pamata rīks alternatīvu aprakstīšanai un vērtēšanai.

3.5. Kritēriju vērtēšana

Ja ir dota liecību kopa, kura apraksta kādu kritēriju, tad ir iespējams noteikt varbūtību, ka kritērijs būs vienāds ar kādu izplūdušo vērtību. Piemēram, ja liecībās ir aprakstīts kritērijs "risks", tad ir iespējams noteikt varbūtību, ka tā vērtība būs "ZEMA", kur "ZEMS" ir izplūdusī vērtība. Iegūta varbūtība ir intervālu, tas ir saskaņā ar intuīciju, jo alternatīvu apraksts ir nenoteikts, tajā ir gan izplūdusī, gan nedeterminēta informācija, līdz ar ko sistēma nevar dot precīzu atbildi.

Izplūdušo granulu paradigmas apraksts un paņēmieni varbūtības augšējas un apakšējas robežas skaitļošanas ir aprakstīti *1.3.3.1.* paragrāfā. Izmantojot *1.3.3.1.* paragrāfā aprakstītās formulas var noteikt varbūtību, ka kritērija vērtība būs vienāda ar kādu izplūdušo vērtību. To var izmantot sekojošajā veidā: vispirms ir jānosaka kādas kritēriju vērtības ir vēlamas un pēc tam katrai alternatīvai ir jāizskaitļo varbūtība, ka tās kritēriju vērtības būs tuvu vēlamām vērtībām. Tādējādi, iegūtās intervālu varbūtības var uzskatīt par atribūtiem, kuri ir jāmaksimizē. Proti, jo lielāka ir varbūtības vērtība, jo labāka ir alternatīva. Turklāt, var noteikt īpaši nevēlamas kritēriju vērtības un izskaitļot varbūtību, ka alternatīvu kritēriju vērtības būs ar to vienādas. Šīs varbūtības var uzskatīt par minimizēšanas atribūtiem.

3.6. Uz entropijas balstīta alternatīvu informatīvuma novērtēšana

Kā darbā iepriekš norādīts (sk. 1.2. apakšnodaļu), nosacītas f-granulas ļauj veikt lēmumu analīzi nenoteiktajās vidē, kad pieejama informācija ir ierobežota. Šajā nodaļā ir aprakstīts kā var noteikt cik informācijas ir pieejams par katru no alternatīvām. Šīs metodes pamatā ir entropijas jēdziens no informācijas teorijas, kurš tiek izmantots sistēmas nenoteiktības apzīmēšanai. Entropijas vērtības parāda cik daudz informācijas trūkst. Tātad, to var izmantot kā alternatīvu nenoteiktības mēru [Vaļiševskis, 2003a].

Lai izrēķinātu entropiju šis uzdevums vispirms ir jāapraksta informācijas teorētiskajā formā – tiek uzbūvētas vairākas sistēmas, kuras var atrasties dažādos stāvokļos. Katra sistēma atbilst kādam kritērijam un tās stāvokļi ir vērtības, kuras attiecīgais kritērijs var pieņemt [Vališevskis&Borisov, 2003b]. Sistēmas stāvokļu varbūtības ir intervālu. Tātad, ir jāizmanto 3.1.2. paragrāfā aprakstītā intervālu entropijas skaitļošanas metode.

Rezultātā tiks iegūts papildus kritērijs, kurš raksturos *zināšanu trūkumu* par alternatīvām. Gribētos uzsvērt, ka tas ir tikai viens no kritērijiem, pēc kuriem tiek vērtētas alternatīvas.

Piedāvāto entropijas skaitļošanas metodi var sadalīt četros posmos, pie kā pirmie divi soļi sakrīt ar piedāvātās lēmumu pieņemšanas metodes soļiem. Tādējādi, ja informatīvums netiek noteikts lēmumu pieņemšanas metodes ietvaros, tad sākumā ir jāizpilda tās pirmie soļi.

1) Kritēriju kopas noteikšana. Šajā posmā ir jānoskaidro kādi kritēriji tiks izmantoti alternatīvu vērtēšanai (sk. 3.3. apakšnodaļu).

2) Alternatīvu aprakstīšana ar izplūdušo granulu palīdzību. Šajā posmā visas alternatīvas tiek aprakstītas ar izplūdušo granulu palīdzību (sk. 3.4. apakšnodaļu).
3) Sistēmu konstruēšana. Šajā posmā informatīvuma analīzes uzdevums tiek pārveidots informācijas teorētiskajā formā. Sīkāk šis posms tiek apskatīts 3.6.1. paragrāfā.
4) Entropijas vērtības noteikšana. Tā kā izstrādātajā metodē entropija tiek izmantota kā informatīvuma novērtējums, līdz ar entropijas izskaitļošanu mēs iegūstam alternatīvas informatīvuma novērtējumu. Sīkāk šis posms tiek apskatīts 3.6.2. paragrāfā.

3.6.1. Informatīvuma novērtēšanas uzdevuma uzstādīšana informācijas teorētiskajā formā

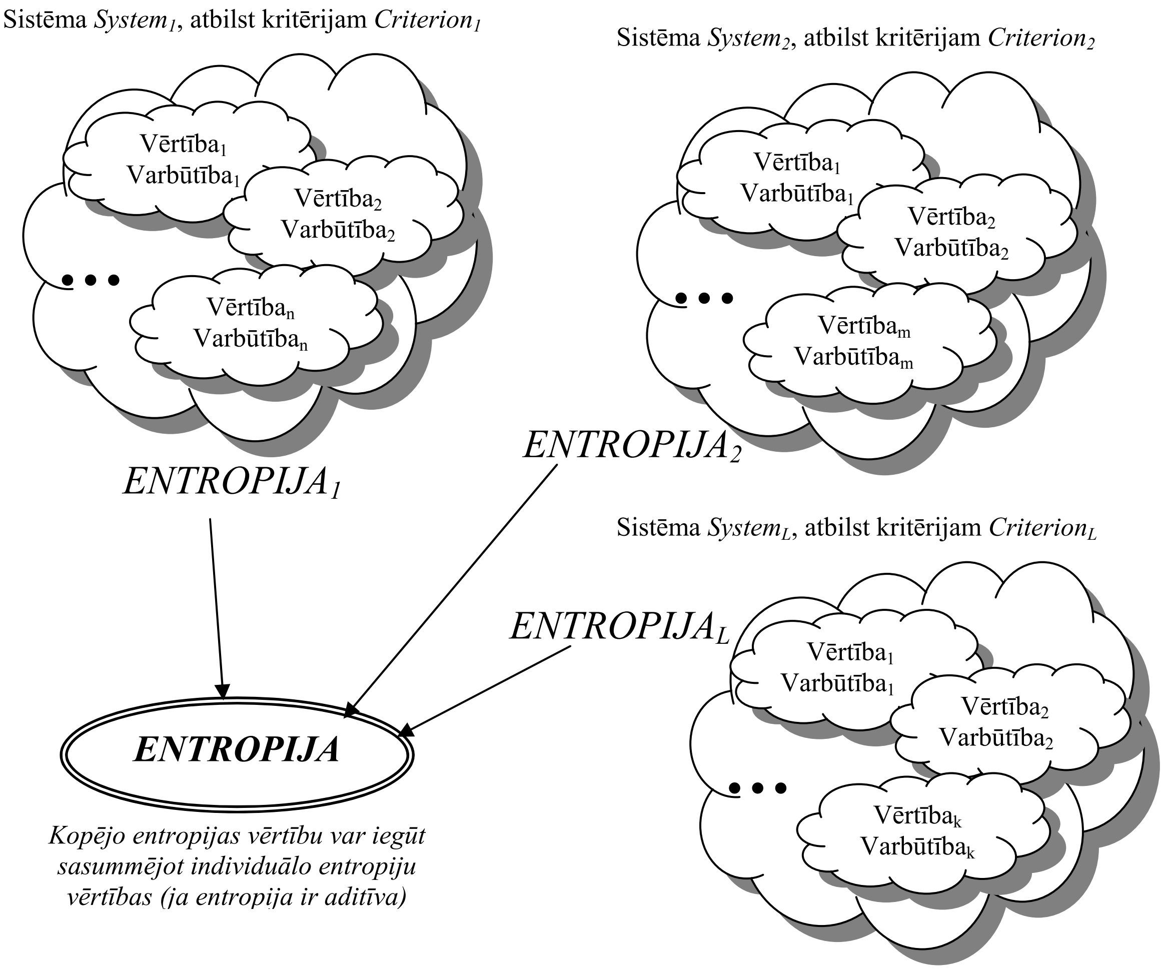
Otrajā solī aprakstītas liecības var izmantot, lai noteiktu varbūtību, ka kādam kritērijam būs kāda vērtība. Šī varbūtība ir intervālu vērtība $[p_{min}, p_{max}]$, kura nosaka varbūtības augšējo un apakšējo robežu. Šo robežu noteikšanai var izmantot adaptīvo tīklu ANGIE, kurš ir aprakstīts 3.1.1. paragrāfā.

Tādā veidā var izskaitļot varbūtības, ka kritērijam būs dažādas vērtības. Piemēram, var izskaitļot varbūtību tam, ka kritērija "risks" vērtība būs vienāda ar "ZEMS", "NE ĪPAŠI AUGSTS", "AUGSTS". Turklāt, šo operāciju var atkārtot pārējiem kritērijiem, nosakot šo kritēriju vērtību varbūtības.

Tagad apskatīsim kā tas viss attiecās uz informatīvuma novērtēšanas metodi, kuras būtība ir parādīta 3.8. attēlā, un kādā veidā uzdevums tiek pārveidots informācijas teorētiskajā formā.

Dažādi kritēriji atbilst dažādām **sistēmām**. Savukārt, par šo sistēmu **stāvokļiem** tiek uzskatītas šo parametru iespējamas vērtības. Tādējādi, varbūtība, ka sistēma S_i atrodas stāvoklī q_k ir varbūtība, ka sistēmai S_i atbilstošajam kritērijam ir stāvoklim q_k atbilstošā vērtība. Citiem vārdiem, teiciens "sistēma atrodas stāvoklī" ir ekvivalents teicienam "kritērijs ir vienāds ar vērtību".

Piemēram, ja mēs apskatam elektrostacijas būvēšanas projektu, tad mums ir jāizveido vismaz viena sistēma, kura atbilst kritērijam "ietekme uz apkārtējo vidi" un kurai būtu tik daudz stāvokļu, cik, pēc mūsu domām, šim kritērijam var būt dažādu vērtību. Aprakstot kritēriju ar izplūdušu granulu palīdzību, mēs katram sistēmas stāvoklim varam noteikt varbūtību, ka sistēma atrodas šajā stāvoklī.



3.8. att. Informatīvuma novērtēšanas metode

Šādā veidā ir jāapraksta visi kritēriji. Rezultātā mēs iegūsim sistēmu kopumu, katra no kurām atbilst kādam kritērijam. To var uzskatīt par *aprakstīšanas* posma beigām un par *skaitļošanas* posma sākumu. Proti, pēc sistēmu aprakstīšanas ir jāizrēķina sistēmu entropija.

3.6.2. Entropijas rēķināšana

Sākotnēja informatīvuma novērtēšanas problēma ir reducēta uz informācijas teorijas problēmu. Vienīgā atšķirība ir tas, ka sistēmu varbūtības ir intervālu, nevis precīzas vērtības. Tas ir arī šķērslis tam, ka entropijas vērtību nevar izskaitļot, izmantojot, piemēram, Šenona sniegtu formulu entropijas skaitļošanai [Ventcel, 2001]. 3.1.2. paragrāfā entropija ir vispārināta gadījumam, kad varbūtības ir intervālu vērtības.

Turklāt, ar sistēmu individuālu entropiju izskaitļošanu informatīvuma novērtēšanas process nebeidzās. Proti, ir jāiegūst kopējais entropijas novērtējums alternatīvai. Šajā gadījumā var izmantot to, ka iegūtā vispārināta entropija ir aditīva. Entropijas aditivitāte nozīmē, ka ja mēs apvienojam vairākas neatkarīgas sistēmas, tad apvienotas sistēmas entropija ir vienāda ar apvienojamo sistēmu individuālo entropiju summu. Proti, ja ir dotas divas sistēmas X, Y, un ar $H(X)$, $H(Y)$ apzīmē šo sistēmu entropiju, tad aditivitāte nozīmē, ka apvienotās sistēmas entropija $H(X, Y)$ ir vienāda ar: $H(X, Y) = H(X) + H(Y)$. 3.1.2. paragrāfā ir parādīts, ka vispārināta entropija ir aditīva (sk. izteiksmi (3.14)). Tātad, kopējo alternatīvas entropijas novērtējumu var iegūt sasummējot sistēmu individuālas entropijas.

Iegūtais kopējais entropijas novērtējums ir informatīvuma novērtējums dotajai alternatīvai. Lai iegūtu novērtējumu citām alternatīvām, šī procedūra ir jāatkārto.

3.7. Kritēriju svēršana

Šajā solī tiek izmantotas citu autoru darbos aprakstītas metodes. Preferenču modelēšanai tiek piedāvāts izmantot kādu no svēršanas shēmām. Proti, var izmantot uz vērtību kokiem bāzēto metodi AHP [Saaty, 1980]. Turklāt, var izmantot tādas svaru novērtēšanas metodes kā SMART [Edwards, 1977], Swing [Von Winterfeldt&Edwards, 1986] vai to vispārinājumu intervālu gadījumam [Mustajoki u.c., 2001]. Pēdējās pieejas pamatideja ir nofiksēt *atbalsta atribūtu*, piešķirt tam kādu vērtību un noteikt citu atribūtu *relatīvu svarīgumu*, piešķirot tiem vērtības par pamatu ņemot atbalsta atribūta vērtību. Pēc šīs procedūras pabeigšanas svarus var noteikt normējot iegūtās atribūtu vērtības tā, lai to summa būtu vienāda ar vieninieku.

Savukārt, AHP procedūras pamatā ir koka konstruēšana, kurš attēlo kritēriju un apakškritēriju hierarhiju. Svaru piešķiršana balstās uz pāru salīdzinājumiem starp kritērijiem, kuriem ir viens un tas pats vecāks.

3.8. Alternatīvu sakārtojuma iegūšana

Piedāvātā lēmumu pieņemšanas metode ir paredzēta izmantošanai nenoteiktības apstākļos. Tādējādi, ar tādu uzdevuma nostādni ir ļoti sarežģīti, vai pat neiespējams nodefinēt optimalitātes kritēriju. Pamatrīki, kuri palīdz lēmumu pieņemošai personai izdarīt labāku un informatīvāku lēmumu ir alternatīvu sakārtojums un izstrādāta lēmumu modeļa jūtīguma analīze. Alternatīvas tiek salīdzinātas uz intervālu varbūtību pamata un vispārējā gadījumā ir iespējams uzkonstruēt tikai daļēju sakārtojumu.

Zemāk ir aprakstīta daļējā alternatīvu sakārtojuma iegūšanas procedūra, kura ir līdzīga PROMETHEE metodei [Brans u.c., 1986], jo tā balstās uz izejošo un ieejošo plūsmu novērtēšanas. Vispirms jāievada dažas definīcijas.

Pieņemsim, ka ir dota alternatīvu kopa $A = \{a_1, a_2, \dots, a_n\}$, kur n ir alternatīvu skaits. Kritēriju kopa, kura tiek izmantota alternatīvu vērtēšanai ir $C = \{c_1, c_2, \dots, c_m\}$, kur m ir kritēriju skaits. Turklāt, pieņemsim, ka izmantojot kādu no svēršanas metodēm ir noteikti sekojošie kritēriju svari: $w_1, w_2, \dots, w_m$. Katras alternatīvas novērtēšanai tiek izmantoti m kritēriji un to novērtējumi ir intervālu varbūtības. Alternatīvas A_i kritēriju novērtējumi ir E_1^i, $E_2^i, \dots, E_m^i$. Šie novērtējumi ir intervālu, tātad: $E_j^i = \left({}^{\min}e_j^i,\ {}^{\max}e_j^i\right)$.

Izejošā un ieejošā plūsma tiek definēta sekojošajā veidā:

$$\phi^-\left(A_i\right) = \sum_{k=1}^{m} w_k \cdot {}^{\min}e_k^i$$

$$\phi^+\left(A_i\right) = \sum_{k=1}^{m} w_k \cdot {}^{\max}e_k^i$$

Daļējais sakārtojums ir definēts sekojošajā veidā:

$$\begin{cases} aPb, & \text{ja}\ \phi^-\left(a\right) > \phi^+\left(b\right), \\ aIb, & \text{ja}\ \phi^-\left(a\right) = \phi^-\left(b\right)\ \text{un}\ \phi^+\left(a\right) = \phi^+\left(b\right), \\ aRb, & \text{pretējā gadījumā}, \end{cases} \tag{3.20}$$

kur a un b ir divas alternatīvas no kopas A, P ir pārspēšanas (*outranking*) relācija, I ir vienlīdzības (*indifference*) relācija un R ir nesalīdzināmības relācija (3.20).

Lai iegūtu pilnu sakārtojumu intervālu varbūtības ir jāreducē uz precīzu vērtību. Tomēr, kā ir norādīts literatūrā [Brans u.c., 1986], pilnais sakārtojums ir iespējams tikai tajā gadījumā, ja informācijas daļa tiek paslēpta, vai tā netiek ņemta vērā. Intervālu varbūtības var reducēt sekojošajā veidā: $e_i^j = \dfrac{{}^{\min}e_i^j + {}^{\max}e_i^j}{2}$

Alternatīvas efektivitāte tiek definēta sekojošajā veidā:

$$p_i = \sum_{k=1}^{m} w_k e_k^i$$

Pilno sakārtojumu var nodefinēt sekojošajā veidā, kur a_i un a_j ir divas alternatīvas no kopas A:

$$\begin{cases} a_i P a_j, & \text{ja } p_i > p_j \\ a_i I a_j, & \text{ja } p_i = p_j \end{cases}$$

Tomēr, ka jau tika minēts iepriekš, pilnais sakārtojums var būt maldinošs, jo tiek slēptas tās alternatīvu īpašības, kuras padara tās par nesalīdzināmām.

Līdz ar to ir pabeigta daļējā un pilnā sakārtojuma iegūšanas procedūras apraksts, kas ir viens no svarīgākajiem līdzekļiem lēmumu analīzei nenoteiktības apstākļos. Nākamajā nodaļā tiek apskatīta uzbūvētā lēmumu modeļa jūtīguma analīze.

3.9. Uzbūvētā modeļa jūtīguma analīze un adaptīva tīkla ANGIE izmantošana kritēriju svarīguma vērtēšanai

Lēmumu modeļa jūtīguma analīze ir svarīgs un jaudīgs lēmumu pieņemošas personas rīks, jo ar tās palīdzību ir iespējams noteikt cik jūtīgs ir lēmumu modelis pret tā parametru mainīšanu.

Līdzīgi kā tas notiek citās metodēs, izstrādātajā lēmumu pieņemšanas metodē ir iespējams mainīt svaru vērtības un sekot izmaiņām alternatīvu sakārtojumā (sk. 2.1.7. paragrāfu).

Ja nepieciešams, var mainīt arī alternatīvu definīcijas vai vēlamo kritērija vērtību. Pēc šīm izmaiņām būs jāpārrēķina varbūtību vērtības (sk. 3.5. apakšnodaļu).

Piedāvātās metodes ietvaros tika izstrādāta arī cita pieeja, kura papildina jūtīguma analīzi. Tā balstās uz alternatīvas vēlamo īpašību noteikšanas (sk. 3.1.1. paragrāfu). To var uzskatīt par atgriezeniskās inženierijas paveidu: vispirms ir jānosaka kādas kritēriju vērtības ir vēlamas un pēc tam ir iespējams noteikt vispievilcīgāko alternatīvu, kura vislielākā mērā atbilst šīm vērtībām. Šīs analīzes rezultātus var izmantot alternatīvu kritisko jeb svarīgāko īpašību noteikšanai. Šī analīze balstās uz 3.1.1. paragrāfā aprakstīto adaptīvo tīklu ANGIE.

3.10. Rezultātu analīze

Piedāvātās metodes pēdējā solī ir jānosaka vai ir jāveic papildus analīze. Lēmumu analīze ir risinājuma iteratīva meklēšana. Tādējādi, pēc vienas iterācijas pabeigšanas ir rūpīgi jāizanalizē vai ir jāpapildina ieejas dati. Daži no iemesliem varētu būt alternatīvu informatīvuma analīzes rezultāti. Proti, ja izrādās, ka par kādu no alternatīvām ir pieejams daudz mazāk informācijas, nekā par citām alternatīvam, tad ir jāapsver papildus informācijas iegūšanas iespēja. Ja tas ir racionāli (izmaksu un iegūtā lietderīguma ziņā), tad modelis ir jāpapildina ar šo informāciju. Citi iemesli ir kritisko alternatīvu īpašību noteikšana, kuras netika pienācīgi izpētītas, papildus jūtīguma analīzes nepieciešamība, jauno datu iegūšana utt. Ja pieejamas informācijas pietiek, lai lēmējpersona varētu izdarīt pietiekoši informatīvu lēmumu, tad tālākā analīze nav nepieciešama.

3.11. Secinājumi par 3. nodaļu

1. Ir izstrādāts adaptīvais tīkls ANGIE. Ar tā palīdzību var apstrādāt izplūdušās liecības. Skaitļošana šajā tīklā notiek paralēli, kas ļauj paaugstināt skaitļošanas ātrumu izpildot programmu sistēmā, kas atbalsta paralēlo skaitļošanu. Adaptīvā tīkla ANGIE apmācīšanas algoritmu var izmantot lēmumu pieņemšanas modeļa jūtīguma analīzē.
2. Piedāvātajā metodē tiek izmantota intervālu entropija, proti, varbūtību vērtības ir intervālu. Līdz ar to Šenona entropija ir vispārināta gadījumam, kad varbūtību vērtības ir intervālu. Turklāt, ir pierādīts, ka iegūtā intervālu entropija ir aditīva. Tas ļauj rēķināt sistēmas entropiju, kuras stāvokļu varbūtības ir intervālu.
3. Ir izstrādāta lēmumu pieņemšanas metode. Piedāvātā metode balstās uz izplūdušajām f-granulām, kas alternatīvu aprakstā ļauj izmantot gan nedeterminētu, gan izplūdušu informāciju. Piedāvātajā metodē alternatīvas tiek salīdzinātas balstoties uz varbūtībām, ka to kritēriju vērtības ir tuvu vēlamajām kritēriju vērtībām.
4. Ir izstrādāta pilnā un daļējā sakārtojuma konstruēšanas metode. Vispārējā gadījumā ir iespējams tikai daļējais alternatīvu sakārtojums, jo lēmumu uzdevuma nostādne ir ļoti nenoteikta. Tomēr, ir aprakstīta procedūra arī pilnā sakārtojuma iegūšanai, kas balstās uz intervālu varbūtības reducēšanas uz precīzo reālo vērtību.
5. Ir iespējams veikt divu veidu jūtīguma analīzi. Pirmā pieeja ir tradicionālā, tajā tiek mainīti lēmumu modeļa parametri un tiek apskatīta alternatīvu sakārtojuma izmaiņa. Otrā pieeja balstās uz adaptīvā tīkla ANGIE un tās rezultātā var noteikt alternatīvu kritiskākos parametrus. Šos rezultātus var izmantot tradicionālajā jūtīguma analīzē. Izmantojot tīklu ANGIE tiek apstrādāti visi parametri vienlaicīgi.

4. PRAKTISKO UZDEVUMU RISINĀŠANA

Lai parādītu kā izstrādāto metodi var izmantot praksē, šajā nodaļā ir izanalizēti divi praktiskie uzdevumi, kuru ir piedāvāts risināt ar izstrādāto metodi.

Kā darbā vairākkārt tika minēts, metode ir paredzēta izmantošanai gadījumos, kad lēmums jāpieņem ļoti nenoteiktajā vide, kura tiek raksturota gan ar izplūdumu, gan ar nejaušību.

Pirmais piemērs ir saistīts ar projektu izvēli. Proti, ar celulozes rūpnīcas būvēšanas projekta izvēli. Vide ir ļoti nenoteikta, jo projekta atribūti, kuri ir saistīti ar nākotni (piemēram, devums tautsaimniecības attīstībai pēc projekta pabeigšanas) ir ne tikai izplūduši, bet arī nedetermināti.

Otrais piemērs ir produkta dzīves cikla posma noteikšana. Šajā gadījumā problēmas sarežģītība ir saistīta ar to, ka informācija par pārdošanas apjomiem tiek iegūta ar aizkavi, jo dati vispirms ir jāapkopo. Bet mainoties produkta dzīves ciklam ir jārīkojas uzreiz, tātad lēmums ir jāpieņem uz ekspertu aptuvenas informācijas pamata, kura vēl nav pieejama. Tiek piedāvāts ekspertu informāciju aprakstīt ar izplūdušo granulu palīdzību, jo tajos var izmantot gan nedeterminētas, gan izplūdušās vērtības, kas atbilst uzdevuma nosacījumiem.

4.1. Programmatūra eksperimentu veikšanai

Eksperimentu veikšanai *Microsoft Visual C++ 6.0* vidē tika izstrādāta klašu virkne, kas ļauj veikt šādas operācijas:

- veikt rēķinus ar dažāda veida izplūdušajiem skaitļiem: trijstūra, trapecveida un zvanveida;
- veikt operācijas ar izplūdušajām liecībām (pievienot granulas, normēt varbūtības, iekļaut liecības adaptīvajā tīklā);
- veidot un veikt operācijas ar adaptīvo tīklu *ANGIE* (pievienot liecības, rēķināt intervālu varbūtības, apmācīt tīklu), sk. 3.1.1. paragrāfu;
- rēķināt intervālu entropiju (sk. 3.1.2. paragrāfu).

Izstrādātās klases, kā arī norādījumi to izmantošanai ir sniegti *Pielikumā*.

4.2. Būvniecības projekta izvēle

Ir jāizvēlas celulozes rūpnīcas būvēšanas projekts. Ir pieejamas vairākas būvēšanas alternatīvas, kas atšķiras ar būvēšanas vietu. Rūpnīcu var uzbūvēt (1) Liepusalā, (2) pie Daugavbūdas vai (3) Mazupē. Problēmas vide ir gan nedeterminēta, gan izplūdusī, jo tiek izmantota ekspertu prognozes. Tādējādi, šī uzdevuma risināšanai jāizmanto uz granulārām liecībām balstītā lēmumu pieņemšanas metode (sk. 3. nodaļu).

Lai atrisinātu šo uzdevumu ir jāizpilda šādi soļi:

- ir jānosaka kritēriji, pēc kuriem tiks vērtētas alternatīvas, un kuri ir svarīgi lēmumu pieņemošai personai;
- katra alternatīva ir jāapraksta ar izplūdušo liecību palīdzību, kurās būtu aprakstīti kritēriju vērtības un notikumi, no kuriem tās ir atkarīgas;
- ir jānosaka vēlamas kritēriju vērtības un jānovērtē varbūtība, ka dotajai alternatīvai dotā kritērija vērtība būs tuvu vēlamai vērtībai;
- ir jānovērtē katras alternatīvas informatīvums, proti, cik daudz informācijas ir pieejams par katru no alternatīvām;
- ir jānomodelē preferences starp kritērijiem;
- ir jāizveido daļējs un pilnais alternatīvu sakārtojums;
- ir jāizanalizē iegūtie rezultāti.

4.2.1. Vides analīze

Kā jau tika minēts, vide ir ļoti nenoteikta, jo projekta atribūti, kuri ir saistīti ar nākotni (piemēram, devums tautsaimniecības attīstībai pēc projekta pabeigšanas) ir ne tikai izplūduši, bet arī nedeterminēti. Izplūdušo granulu pielietošana šajā gadījumā ir pamatota, jo pretējā gadījumā modelī tiktu ievesta maldinoša precizitāte (proti, ignorējot izplūdumu vai nejaušību). Ja atribūts nav saistīts ar kādu no nenoteiktības veidiem, tad aprakstā tiek izmantotas precīzas vai determinētas vērtības.

Tādējādi, izmantojot izstrādāto lēmumu pieņemšanas metodi (sk. 3. nodaļu) ir sniegta nepieciešama pielāgojamība uzdevuma risināšanai, jo tā ļauj izmantot kā izplūdušo un nedeterminēto informāciju, tā arī informāciju, kura ir gan nedeterminētā, gan izplūdusī.

4.2.2. Kritēriju analīze

Pēc konsultācijām ar ekspertiem un ieinteresēto pušu pārstāvjiem tika izvēlēti šādi kritēriji, kuri atspoguļo dažādas projektu puses:

- Loģistika
    - Izejvielu piegādes ērtība
    - Produkcijas realizācijas ērtība
    - Infrastruktūras un sakaru attīstības līmenis
- Rūpnīcas būvēšanas izmaksas
- Ekonomiskā dzīvotspēja
    - Prognozējama investīciju atmaksa
    - Darba vietu radīšanas potenciāls
    - Fiksētās izmaksas
    - Mainīgās izmaksas
- Ietekme uz vidi
    - Ietekme uz mežiem
    - Ietekme uz ūdenstilpēm
    - Ietekme uz augsni
    - Ietekme uz cilvēkiem
    - Ietekme uz lauksaimniecību
    - Ietekme uz zvejniecību
    - Ietekme uz dzīvniekiem
- Personāls
    - Kvalificēta personāla pieejamība būvēšanas vietā
    - Apgabala kopējais ekonomiskais stāvoklis
    - Ar darba personāla pārvietošanu saistītas izmaksas
    - Ar darba personāla apmācību saistītas izmaksas

Tagad ir jānosaka no kā šo kritēriju vērtības ir atkarīgas un vai tās var precīzi noteikt. Ja nevar noteikt precīzi, tad ir jānosaka kādi iznākumi ir nedeterminētam notikumam, no kura ir atkarīgas kritērija vērtības un kāds ir kritērija novērtējums katram iznākumam. To var izdarīt ar izplūdušo granulu palīdzību (sk. 4.2.3. paragrāfu).

Šim uzdevumam nevar uzbūvēt kritēriju vērtību tabulu, jo dažas vērtības ir nedeterminētas, līdz ar to šī uzdevuma risināšanai nevar izmantot tādas metodes kā ELECTRE vai PROMETHEE. Turklāt, uzdevuma aprakstā tiek izmantotas izplūdušas vērtības, tātad nevar izmantot arī lēmumu analīzes metodi.

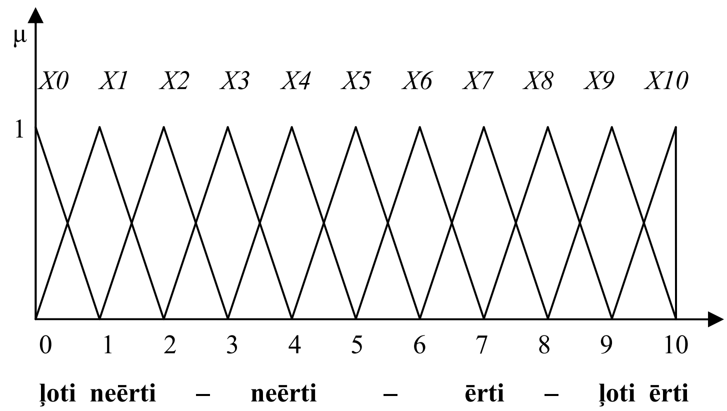
4.2.3. Problēmas apgabala aprakstīšana ar izplūdušo granulu palīdzību

Katra alternatīva ir jāapraksta ar izplūdušo granulu palīdzību. Granulās tiek grupētas liecībās un katra liecība attēlo kādas nedeterminētas parādības iespējamus iznākumus, no kuriem ir atkarīga kritērija vērtība.

Kritēriju izplūdušas vērtības

Vispirms apskatīsim kādas var būt kritēriju izplūdušas vērtības. Tās tiks izmantotas alternatīvu aprakstā, tātad to granularitātei ir jābūt pietiekoši sīkai, lai ar to palīdzību varētu pietiekoši precīzi aprakstīt ekspertu un lēmumu pieņemošas personas zināšanas un viedokli par kritēriju vērtībām. Tomēr, tai nevajadzētu būt pārāk sīkai, lai saglabātu atbilstošu nenoteiktības pakāpi. Proti, lai problēmas aprakstu nepadarītu par maldinoši precīzu. Zemāk ir norādīti kritēriju apzīmējumi un šiem kritērijiem atbilstošās izplūdušās apakškopas (sk. 4.1. - 4.10. att.).

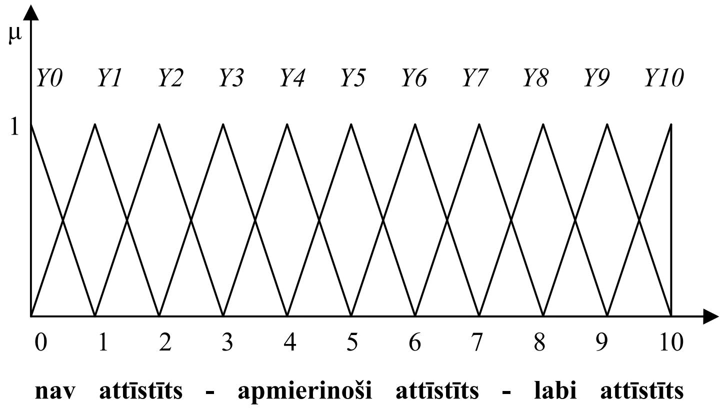
Izejvielu piegādes ērtība – C1
Produkcijas realizācijas ērtība – C2



4.1. att. Kritēriju C1 un C2 granulācija

4.1. attēlā "ērtība" ir raksturota ar nosacītās skalas 0-10 palīdzību. Uz šīs skalas ir nodefinētas vairākas izplūdušās vērtības, kuras raksturo dažādas ērtības pakāpes.

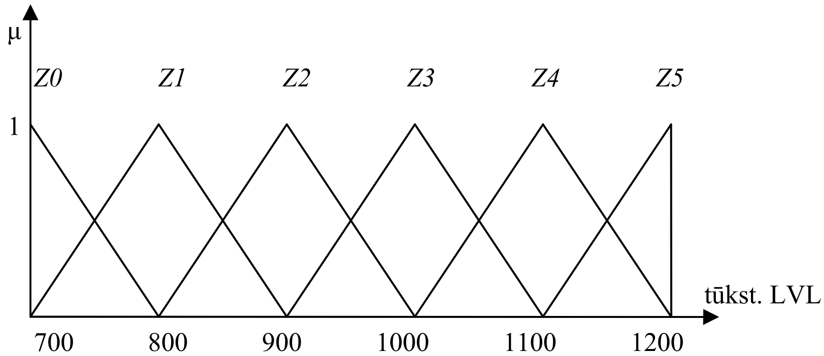
Infrastruktūras un sakaru attīstības līmenis – C3



4.2. att. Kritērija C3 granulācija

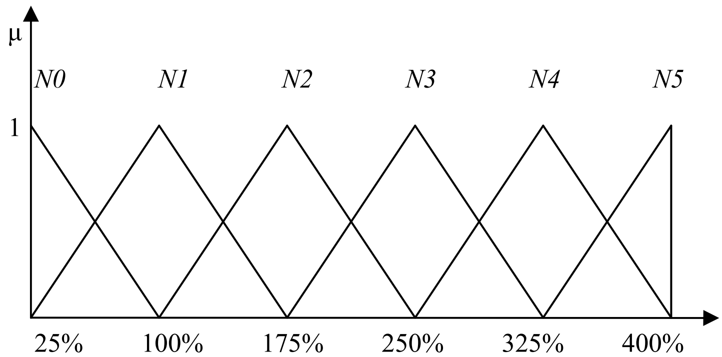
4.2. attēlā "attīstības līmenis" ir raksturots ar nosacītās skalas 0-10 palīdzību. Uz šīs skalas ir nodefinētas vairākas izplūdušās vērtības, kuras raksturo dažādus attīstības līmeņus.

Rūpnīcas būvēšanas izmaksas – C4



4.3. att. Kritērija C4 granulācija

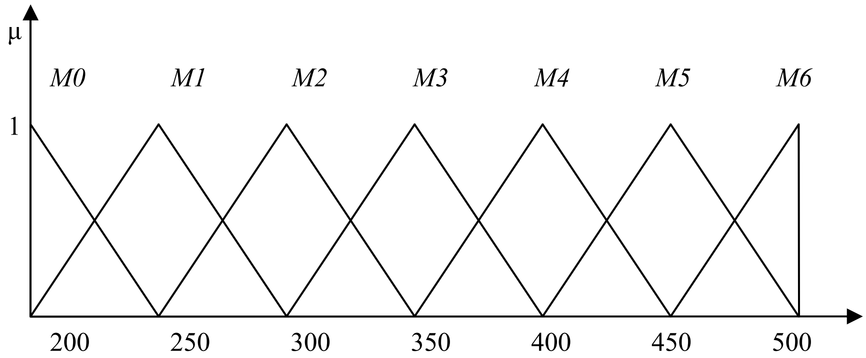
Prognozējama investīcijas atmaksa – C5



4.4. att. Kritērija C5 granulācija

4.4. attēlā ir parādīta prognozējamā ROI (investīciju atmaksas) vērtība.

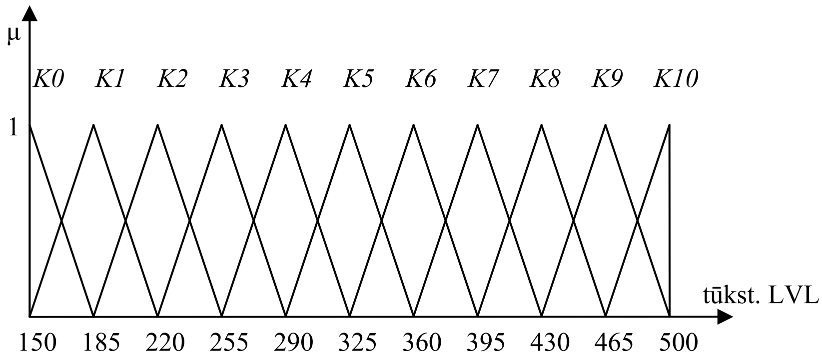
Darba vietu radīšanas potenciāls – C6



4.5. att. Kritērija C6 granulācija

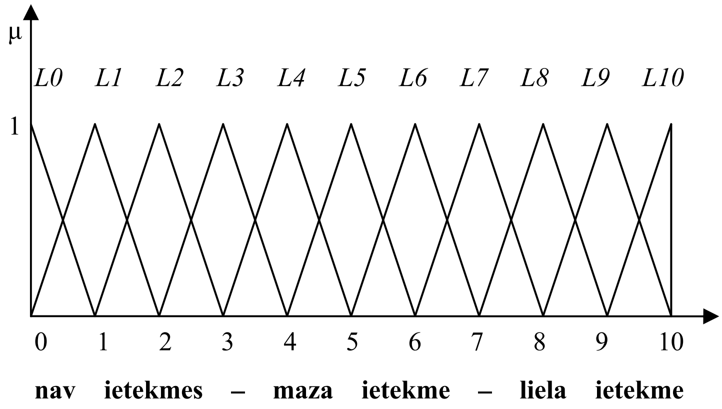
4.5. attēlā ir redzamas izplūdušās vērtībās, kuras raksturo potenciālo jauno darbavietu skaitu, kuras parādīsies rūpnīcas būvēšanas rezultātā.

Fiksētās izmaksas – C7
Mainīgās izmaksas – C8



4.6. att. Kritēriju C7 un C8 granulācija

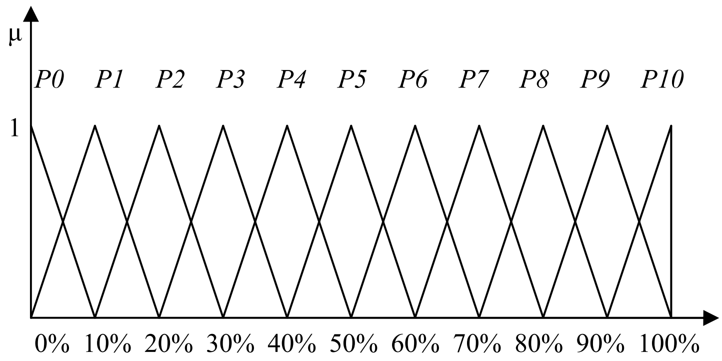
Ietekme uz mežiem – C9
Ietekme uz augsn – C11
Ietekme uz lauksaimniecību – C13
Ietekme uz dzīvniekiem – C15
Ietekme uz ūdenstilpēm – C10
Ietekme uz cilvēkiem – C12
Ietekme uz zvejniecību – C14



4.7. att. Kritēriju C9, C10, C11, C12, C13, C14 un C15 granulācija

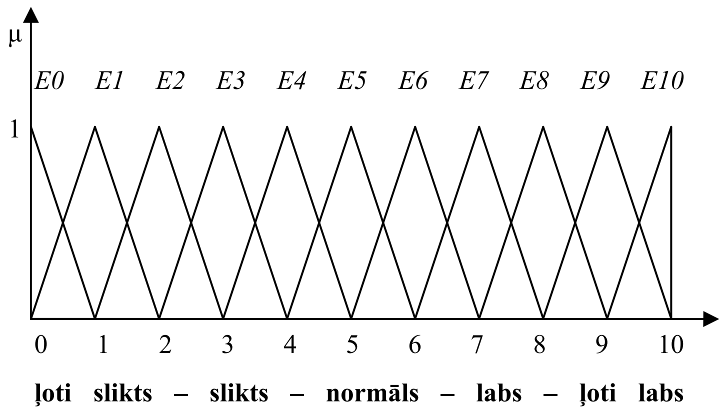
4.7. attēlā "ietekme" ir raksturota ar nosacītās skalas 0-10 palīdzību. Uz šīs skalas ir nodefinētas vairākas izplūdušās vērtības, kuras raksturo dažādas ietekmes pakāpes.

Kvalificēta personāla pieejamība būvēšanas vietā – C16



4.8. att. Kritērija C16 granulācija

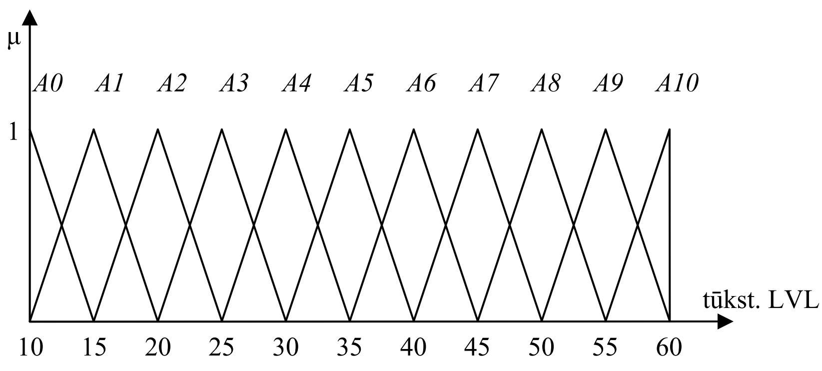
Apgabala kopējais ekonomiskais stāvoklis – C17



4.9. att. Kritērija C17 granulācija

4.9. attēlā "ekonomiskais stāvoklis" ir raksturots ar nosacītās skalas 0-10 palīdzību. Uz šīs skalas ir nodefinētas vairākas izplūdušās vērtības, kuras raksturo dažādus ekonomiskos stāvokļus.

Ar darba personāla pārvietošanu saistītas izmaksas – C18
Ar darba personāla apmācību saistītas izmaksas – C19



4.10. att. Kritēriju C18 un C19 granulācija

Nākamajās nodaļās izmantojot nodefinētās izplūdušās vērtības tiek aprakstīta par alternatīvām pieejamā informācija.

4.2.4. Alternatīvu aprakstīšana ar izplūdušo granulu palīdzību

4.2.4.1. Alternatīvas "Liepusala" aprakstīšana

Pēc kārtas apskatīsim visus kritērijus, notikumus, no kuriem tie ir atkarīgi un nodefinēsim attiecīgas izplūdušas granulas.

Kritērijs C1 = "Izejvielu piegādes ērtība"

Šis kritērijs ir atkarīgs no tādiem determinētiem notikumiem kā rūpnīcas atrašanas vieta un pieejama ceļu infrastruktūra. Turklāt, tas ir atkarīgs arī no tā, cik lielā mērā tiks attīstīta apkārtēja infrastruktūra, kas ir nedeterminēts notikums, jo šī attīstība ir atkarīga no vietējām pašpārvaldēm.

Notikumu "atrašanas vieta" un "pieejama ceļu infrastruktūra" vērtības ir zināmas. Ņemot tos vērā, noteiksim kritērija vērtības, atkarībā no pašpārvalžu sadarbības līmeņa, kurš var būt zems, vidējs vai augsts ar sekojošo varbūtību sadalījumu:

JA (sadarbības līmenis = zems) ar varbūtību 0,1 TAD C1 = X4
JA (sadarbības līmenis = vidējs) ar varbūtību 0,3 TAD C1 = X7
JA (sadarbības līmenis = augsts) ar varbūtību 0,6 TAD C1 = X9

Pirmo granulu var interpretēt kā šādu ekspertu sniegtu informāciju:

Ja pašpārvalžu sadarbības līmenis ir zems, tad kritērija "Izejvielu piegādes ērtība" vērtība ir vienāda ar izplūdušo vērtību X4 (sk. 4.1. att.). Turklāt, ir zināms, ka sadarbības līmenis ar pašpārvaldēm būs zems ar varbūtību 0,1.

Piezīme:

Augstāk aprakstītas granulas veido liecību, kura atbilst sekojošajām izplūdušo vērtību varbūtības sadalījumam:

P(C1 = X4) = 0,1
P(C1 = X7) = 0,3
P(C1 = X9) = 0,6

Kritērijs C2 = "Produkcijas realizācijas ērtība"

Šo kritēriju ietekmē tie paši notikumi, kuri ietekmē kritērija C1 vērtības, līdz ar to šo kritēriju var aprakstīt šādi:

JA (sadarbības līmenis = zems) ar varbūtību 0,1 TAD C2 = X5
JA (sadarbības līmenis = vidējs) ar varbūtību 0,3 TAD C2 = X8
JA (sadarbības līmenis = augsts) ar varbūtību 0,6 TAD C2 = X9

Kritērijs C3 = "Infrastruktūras un sakaru attīstības līmenis"

Šis kritērijs reprezentē ne tikai ceļu infrastruktūru, bet arī citu komunikāciju un sakaru līniju attīstības līmeni. To ietekmē tie paši notikumi, kuri ietekmē kritērijus C1 un C2, kā arī to ietekmē sakaru kompāniju attīstības plāni dotajā apgabalā. Plānu var vispār nebūt, tie var būt nelieli uzturēšanas plāni, vidēji attīstības plāni un nozīmīgi plāni.

Pieejamās zināšanas var aprakstīt šādi:

JA (sadarbības līmenis = zems) ar varbūtību 0,1 TAD C3 = Y3
JA (sadarbības līmenis = vidējs) ar varbūtību 0,3 TAD C3 = Y7
JA (sadarbības līmenis = augsts) ar varbūtību 0,6 TAD C3 = Y10

JA (attīstības plāni = nav) ar varbūtību 0,3 TAD C3 = Y3
JA (attīstības plāni = nelieli) ar varbūtību 0,5 TAD C3 = Y4
JA (attīstības plāni = vidēji) ar varbūtību 0,15 TAD C3 = Y8
JA (attīstības plāni = grandiozi) ar varbūtību 0,05 TAD C3 = Y10

Kritērijs C4 = "Rūpnīcas būvēšanas izmaksas"

Būvēšanas izmaksas ir atkarīgas no ekonomiskas situācijas, kura ir raksturota ar valūtu kursu svārstībām, bankas kredītu procentu likmēm, un kopēja ekonomikas stāvokļa labvēlīguma. Ņemot vēra ekspertu novērtējumus tika nodefinētas šādas granulas:

JA (ekonomiska situācija = slikta) ar varbūtību 0,2 TAD C4 = Z5
JA (ekonomiska situācija = vidēja) ar varbūtību 0,5 TAD C4 = Z3
JA (ekonomiska situācija = laba) ar varbūtību 0,3 TAD C4 = Z1

Kritērijs C5 = "Prognozējama investīciju atmaksa"

Investīciju atmaksa ir atkarīga no ekonomiskas situācijas pēc rūpnīcas uzbūvēšanas, no pieprasījuma un no iekarotas tirgus daļas. Proti, tiek apskatīta prognozējama ilgtermiņa atmaksa piecu gadu laikā

Ņemot vērā ekspertu novērtējumus, tika secināts, ka investīciju atmaksa būs sekojošā:

P(C5 = N2) = 0,3
P(C5 = N4) = 0,6
P(C5 = N5) = 0,1

Kritērijs C6 = "Darba vietu radīšanas potenciāls"

Šis kritērijs ir determinēts, un tā vērtība ir nodefinēta šādā veidā:

P(C6 = M3) = 1,0

Kritērijs C7 = "Fiksētās izmaksas"

Fiksētajās izmaksās ietilpst administratīvas izmaksas, ar plānošanu un marketingu saistītas izmaksas. Tos var pietiekoši precīzi novērtēt, jo tie nav atkarīgi no ražošanas apjoma:

P(C7 = K2) = 1,0

Kritērijs C8 = "Mainīgās izmaksas"

Mainīgās izmaksas ir atkarīgas no ražošanas apjoma, no pieejama darbaspēka, no degvielas, izejvielu un elektroenerģijas cenas.

Ņemot vērā ekspertu novērtējumus, var nodefinēt sekojošās liecības:

JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C8 = K3
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C8 = K5
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C8 = K7
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C8 = K8

JA (darbaspēka kvalifikācija = zema) ar varbūtību 0,1 TAD C8 = K4
JA (darbaspēka kvalifikācija = vidēja) ar varbūtību 0,3 TAD C8 = K2
JA (darbaspēka kvalifikācija = augsta) ar varbūtību 0,6 TAD C8 = K0

Pēdējas liecības ideja ir tāda, ka jo augstāka ir darba kvalifikācija, jo optimālākā veidā var izmantot resursus.

JA (resursu cena = zema) ar varbūtību 0,05 TAD C8 = K2
JA (resursu cena = vidēja) ar varbūtību 0,15 TAD C8 = K4
JA (resursu cena = augsta) ar varbūtību 0,8 TAD C8 = K6

Kritērijs C9 = "Ietekme uz mežiem"

Tā kā rūpnīcas būvēšanas projekts ir pietiekoši labi izstrādāts, šo kritēriju var uzskatīt par determinētu:

P(C9 = L3) = 1,0

Kritērijs C10 = "Ietekme uz ūdenstilpēm"

Ir saņemts Vides Ministrijas novērtējums par projekta ietekmi uz dabas resursiem, tātad šo kritēriju arī var precīzi nodefinēt:

P(C10 = L7) = 1,0

Kritērijs C11 = "Ietekme uz augsni"

Izmantojot Vides Ministrijas novērtējumu var izdarīt sekojošo secinājumu:

P(C11 = L5) = 1,0

Kritērijs C12 = "Ietekme uz cilvēkiem"

Šis kritērijs raksturo rūpnīcas ietekmi uz apkārtnē dzīvojošajiem cilvēkiem, tas ir atkarīgs no rūpnīcas ražošanas apjoma, kas ir nedeterminēts notikums un to tādiem determinētiem faktoriem kā rūpnīcas tuvums apdzīvotājai vietai, iedzīvotāju blīvums rūpnīcas tuvumā esošajā teritorijā. Tātad, pieejamā informācija tiek aprakstīta sekojošajā veidā:

JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C12 = L2
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C12 = L3
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C12 = L6
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C12 = L7

Kritērijs C13 = "Ietekme uz lauksaimniecību"

Līdzīgi kritērijam C12, šis kritērijs ir atkarīgs no ražošanas apjoma un no rūpnīcas tuvumā esošajiem laukiem. Tikai pirmais faktors ir nedeterminēts. Šo kritēriju var aprakstīt sekojošajā veidā:

JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C13 = L4
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C13 = L6
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C13 = L7
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C13 = L9

Kritērijs C14 = "Ietekme uz zvejniecību"

Līdzīgi iepriekšējiem kritērijiem, šis kritērijs ir atkarīgs no ražošanas apjoma. Šo kritēriju var aprakstīt sekojošajā veidā:

JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C14 = L7
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C14 = L8
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C14 = L9
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C14 = L10

Kritērijs C15 = "Ietekme uz dzīvniekiem"

Līdzīgi iepriekšējiem kritērijiem, šis kritērijs ir atkarīgs no ražošanas apjoma. Šis kritērijs raksturo rūpnīcas ietekmi uz apkārtējo faunu. Šo kritēriju var aprakstīt sekojošajā veidā:

JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C15 = L1
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C15 = L2
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C15 = L3
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C15 = L4

Kritērijs C16 = "Kvalificēta personāla pieejamība būvēšanas vietā"

Šis kritērijs parāda cik cilvēku ar atbilstošu kvalifikāciju ir pieejams būvēšanas rajonā no kopēja rūpnīcā nepieciešamo darbinieku skaita.

Pēc konsultēšanās ar ekspertiem tika iegūti sekojošie rezultāti:

P(C16 = P5) = 0,3
P(C16 = P6) = 0,4
P(C16 = P7) = 0,3

Kritērijs C17 = "Apgabala kopējais ekonomiskais stāvoklis"

Šis kritērijs ir determinēts un to var nodefinēt sekojošajā veidā:

P(C17 = E7) = 1,0

Kritērijs C18 = "Ar darba personāla pārvietošanu saistītas izmaksas"

Šī kritērija vērtība ir atkarīga no būvēšanas rajonā pieejama kvalificēta personāla. Pēc demogrāfiskas informācijas pētīšanas un konsultēšanas ar ekspertiem tika iegūta sekojošā informācija:

P(C18 = A2) = 0,3
P(C18 = A3) = 0,4
P(C18 = A5) = 0,3

Kritērijs C19 = "Ar darba personāla apmācību saistītas izmaksas"

Šī kritērija vērtība ir atkarīga no būvēšanas rajonā pieejama kvalificēta personāla. Pēc demogrāfiskas informācijas pētīšanas un konsultēšanas ar ekspertiem tika iegūta sekojošā informācija:

P(C19 = A1) = 0,3
P(C19 = A4) = 0,4
P(C19 = A7) = 0,3

4.2.4.2. Alternatīvas "Daugavbūda" aprakstīšana

Pēc kārtas apskatīsim visus kritērijus, parādības, no kurām tie ir atkarīgi un nodefinēsim attiecīgas izplūdušas granulas.

Kritērijs C1 = "Izejvielu piegādes ērtība"
JA (sadarbības līmenis = zems) ar varbūtību 0,2 TAD C1 = X3
JA (sadarbības līmenis = vidējs) ar varbūtību 0,4 TAD C1 = X6
JA (sadarbības līmenis = augsts) ar varbūtību 0,4 TAD C1 = X8

Kritērijs C2 = "Produkcijas realizācijas ērtība"
JA (sadarbības līmenis = zems) ar varbūtību 0,2 TAD C2 = X5
JA (sadarbības līmenis = vidējs) ar varbūtību 0,4 TAD C2 = X7
JA (sadarbības līmenis = augsts) ar varbūtību 0,4 TAD C2 = X9

Kritērijs C3 = "Infrastruktūras un sakaru attīstības līmenis"
JA (sadarbības līmenis = zems) ar varbūtību 0,2 TAD C3 = Y4
JA (sadarbības līmenis = vidējs) ar varbūtību 0,4 TAD C3 = Y6
JA (sadarbības līmenis = augsts) ar varbūtību 0,4 TAD C3 = Y9

JA (attīstības plāni = nav) ar varbūtību 0,6 TAD C3 = Y4
JA (attīstības plāni = nelieli) ar varbūtību 0,3 TAD C3 = Y5
JA (attīstības plāni = vidēji) ar varbūtību 0,1 TAD C3 = Y6

Kritērijs C4 = "Rūpnīcas būvēšanas izmaksas"
JA (ekonomiska situācija = slikta) ar varbūtību 0,2 TAD C4 = Z4
JA (ekonomiska situācija = vidēja) ar varbūtību 0,5 TAD C4 = Z3
JA (ekonomiska situācija = laba) ar varbūtību 0,3 TAD C4 = Z0

Kritērijs C5 = "Prognozējama investīciju atmaksa"
P(C5 = N2) = 0,3
P(C5 = N4) = 0,6
P(C5 = N5) = 0,1

Kritērijs C6 = "Darba vietu radīšanas potenciāls"
P(C6 = M3) = 1,0

Kritērijs C7 = "Fiksētās izmaksas"
P(C7 = K3) = 1,0

Kritērijs C8 = "Mainīgās izmaksas"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C8 = K3
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C8 = K5
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C8 = K7
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C8 = K8

JA (darbaspēka kvalifikācija = zema) ar varbūtību 0,4 TAD C8 = K4
JA (darbaspēka kvalifikācija = vidēja) ar varbūtību 0,5 TAD C8 = K2
JA (darbaspēka kvalifikācija = augsta) ar varbūtību 0,1 TAD C8 = K0

JA (resursu cena = zema) ar varbūtību 0,05 TAD C8 = K2
JA (resursu cena = vidēja) ar varbūtību 0,15 TAD C8 = K4
JA (resursu cena = augsta) ar varbūtību 0,8 TAD C8 = K6

Kritērijs C9 = "Ietekme uz mežiem"
P(C9 = L3) = 1,0

Kritērijs C10 = "Ietekme uz ūdenstilpēm"
P(C10 = L8) = 1,0

Kritērijs C11 = "Ietekme uz augsni"
P(C11 = L6) = 1,0

Kritērijs C12 = "Ietekme uz cilvēkiem"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C12 = L4
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C12 = L5
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C12 = L7
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C12 = L9

Kritērijs C13 = "Ietekme uz lauksaimniecību"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C13 = L4
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C13 = L6
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C13 = L7
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C13 = L10

Kritērijs C14 = "Ietekme uz zvejniecību"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C14 = L7
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C14 = L8
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C14 = L9
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C14 = L10

Kritērijs C15 = "Ietekme uz dzīvniekiem"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C15 = L1
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C15 = L2
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C15 = L3
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C15 = L4

Kritērijs C16 = "Kvalificēta personāla pieejamība būvēšanas vietā"
P(C16 = P3) = 0,6
P(C16 = P4) = 0,3
P(C16 = P5) = 0,1

Kritērijs C17 = "Apgabala kopējais ekonomiskais stāvoklis"
P(C17 = E4) = 1,0

Kritērijs C18 = "Ar darba personāla pārvietošanu saistītas izmaksas"
P(C18 = A2) = 0,2
P(C18 = A4) = 0,3
P(C18 = A6) = 0,5

Kritērijs C19 = "Ar darba personāla apmācību saistītas izmaksas"
P(C19 = A4) = 0,2
P(C19 = A6) = 0,35
P(C19 = A9) = 0,45

4.2.4.3. Alternatīvas "Mazupe" aprakstīšana

Pēc kārtas apskatīsim visus kritērijus, parādības, no kurām tie ir atkarīgi un nodefinēsim attiecīgas izplūdušas granulas.

Kritērijs C1 = "Izejvielu piegādes ērtība"
JA (sadarbības līmenis = zems) ar varbūtību 0,1 TAD C1 = X5
JA (sadarbības līmenis = vidējs) ar varbūtību 0,5 TAD C1 = X8
JA (sadarbības līmenis = augsts) ar varbūtību 0,4 TAD C1 = X10

Kritērijs C2 = "Produkcijas realizācijas ērtība"
JA (sadarbības līmenis = zems) ar varbūtību 0,1 TAD C2 = X6
JA (sadarbības līmenis = vidējs) ar varbūtību 0,5 TAD C2 = X8
JA (sadarbības līmenis = augsts) ar varbūtību 0,4 TAD C2 = X10

Kritērijs C3 = "Infrastruktūras un sakaru attīstības līmenis"
JA (sadarbības līmenis = zems) ar varbūtību 0,1 TAD C3 = Y5
JA (sadarbības līmenis = vidējs) ar varbūtību 0,5 TAD C3 = Y7
JA (sadarbības līmenis = augsts) ar varbūtību 0,4 TAD C3 = Y10

JA (attīstības plāni = nav) ar varbūtību 0,1 TAD C3 = Y6
JA (attīstības plāni = nelieli) ar varbūtību 0,3 TAD C3 = Y7
JA (attīstības plāni = lieli) ar varbūtību 0,6 TAD C3 = Y9

Kritērijs C4 = "Rūpnīcas būvēšanas izmaksas"
JA (ekonomiska situācija = slikta) ar varbūtību 0,2 TAD C4 = Z5
JA (ekonomiska situācija = vidēja) ar varbūtību 0,5 TAD C4 = Z4
JA (ekonomiska situācija = laba) ar varbūtību 0,3 TAD C4 = Z2

Kritērijs C5 = "Prognozējama investīciju atmaksa"
P(C5 = N2) = 0,3
P(C5 = N4) = 0,6
P(C5 = N5) = 0,1

Kritērijs C6 = "Darba vietu radīšanas potenciāls"
P(C6 = M3) = 1,0

Kritērijs C7 = "Fiksētās izmaksas"
P(C7 = K4) = 1,0

Kritērijs C8 = "Mainīgās izmaksas"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C8 = K4
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C8 = K6
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C8 = K8
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C8 = K10

JA (darbaspēka kvalifikācija = zema) ar varbūtību 0,4 TAD C8 = K4
JA (darbaspēka kvalifikācija = vidēja) ar varbūtību 0,5 TAD C8 = K2
JA (darbaspēka kvalifikācija = augsta) ar varbūtību 0,1 TAD C8 = K0

JA (resursu cena = zema) ar varbūtību 0,05 TAD C8 = K2
JA (resursu cena = vidēja) ar varbūtību 0,15 TAD C8 = K4
JA (resursu cena = augsta) ar varbūtību 0,8 TAD C8 = K6

Kritērijs C9 = "Ietekme uz mežiem"
P(C9 = L1) = 1,0

Kritērijs C10 = "Ietekme uz ūdenstilpēm"
P(C10 = L6) = 1,0

Kritērijs C11 = "Ietekme uz augsni"
P(C11 = L3) = 1,0

Kritērijs C12 = "Ietekme uz cilvēkiem"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C12 = L5
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C12 = L7
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C12 = L9
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C12 = L10

Kritērijs C13 = "Ietekme uz lauksaimniecību"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C13 = L3
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C13 = L5
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C13 = L6
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C13 = L8

Kritērijs C14 = "Ietekme uz zvejniecību"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C14 = L5
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C14 = L6
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C14 = L7
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C14 = L8

Kritērijs C15 = "Ietekme uz dzīvniekiem"
JA (ražošanas apjoms = mazs) ar varbūtību 0,1 TAD C15 = L1
JA (ražošanas apjoms = vidējs) ar varbūtību 0,2 TAD C15 = L2
JA (ražošanas apjoms = liels) ar varbūtību 0,4 TAD C15 = L3
JA (ražošanas apjoms = ļoti liels) ar varbūtību 0,3 TAD C15 = L4

Kritērijs C16 = "Kvalificēta personāla pieejamība būvēšanas vietā"
P(C16 = P5) = 0,2
P(C16 = P7) = 0,3
P(C16 = P9) = 0,5

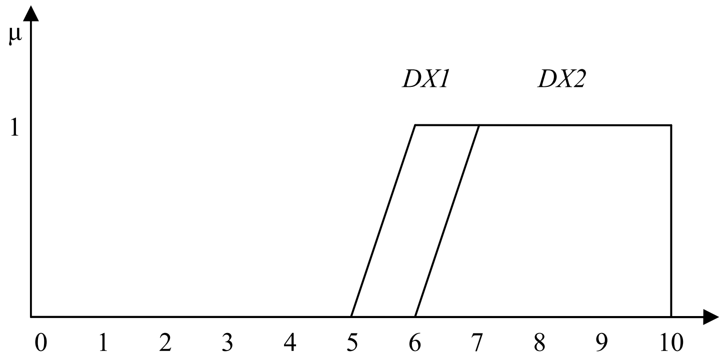
Kritērijs C17 = "Apgabala kopējais ekonomiskais stāvoklis"
P(C17 = E8) = 1,0

Kritērijs C18 = "Ar darba personāla pārvietošanu saistītas izmaksas"
P(C18 = A1) = 0,5
P(C18 = A2) = 0,3
P(C18 = A4) = 0,2

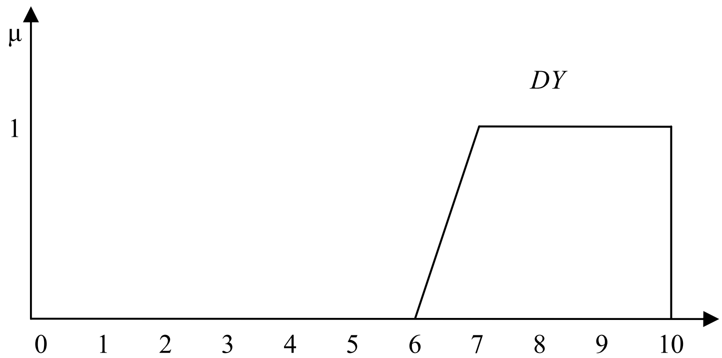
Kritērijs C19 = "Ar darba personāla apmācību saistītas izmaksas"
P(C19 = A1) = 0,5
P(C19 = A3) = 0,3
P(C19 = A4) = 0,2

4.2.5. Kritēriju vēlamo vērtību noteikšana

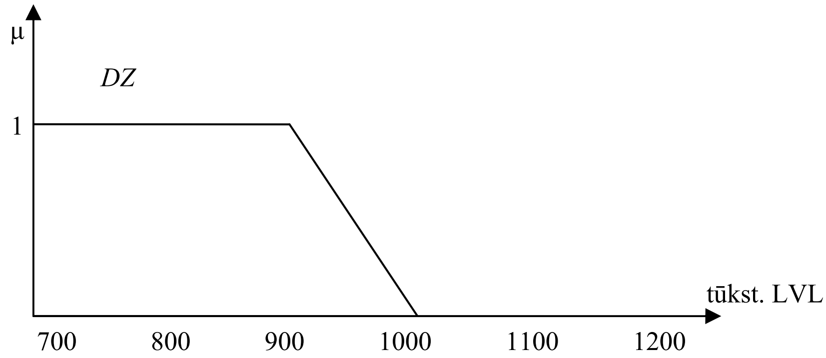
Pēc alternatīvu aprakstīšanas ir jānodefinē kritēriju vēlamās vērtības. Vispirms aprakstīsim izplūdušās vērtības, kuras atbilst vēlamajām vērtībām un pēc tam norādīsim katra kritērija vēlamo vērtību. Tās ir parādītas 4.11. – 4.20. attēlos. Skalu aprakstu ir atrodams 4.2.3. paragrāfā, jo šie grafiki ir uzdoti uz tām pašām skalām kā, attiecīgi, 4.1. - 4.10. attēli.



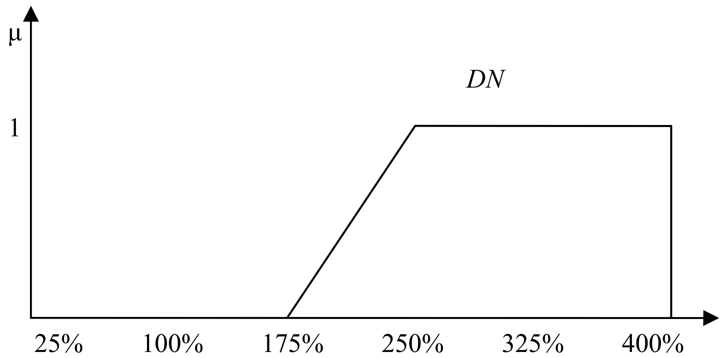
4.11. att. Kritēriju C1 un C2 vēlamās vērtības



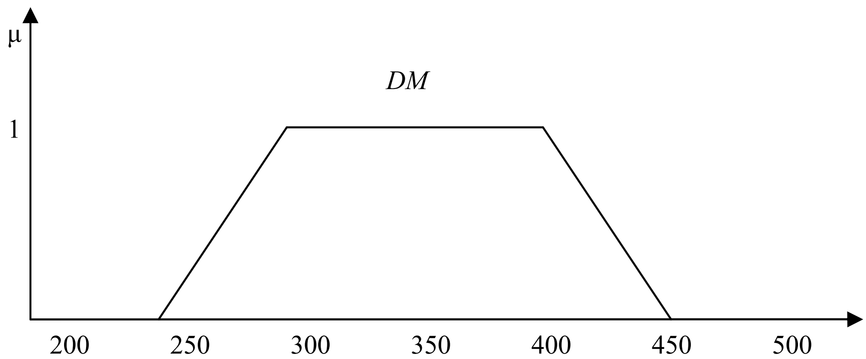
4.12. att. Kritērija C3 vēlamā vērtība



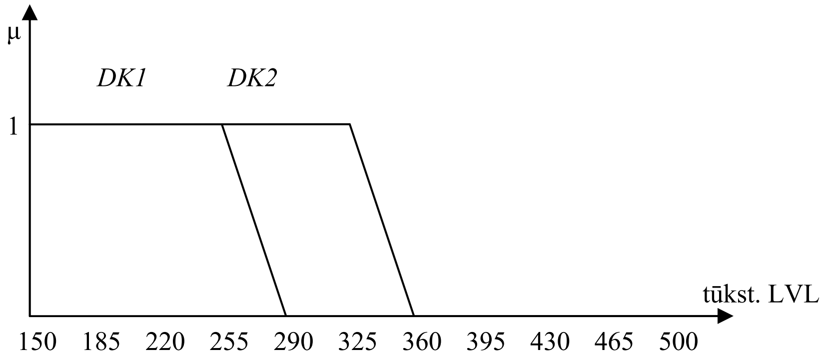
4.13. att. Kritērija C4 vēlamā vērtība



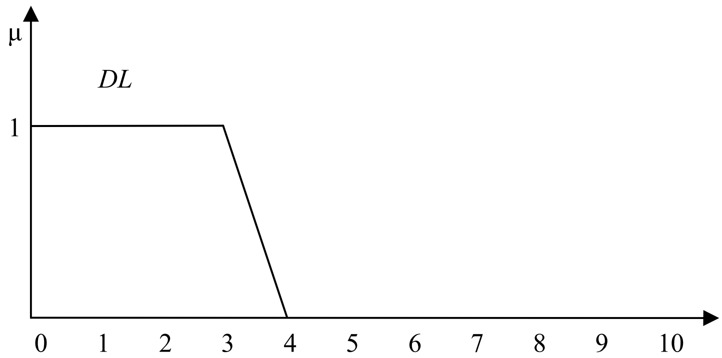
4.14. att. Kritērija C5 vēlamā vērtība



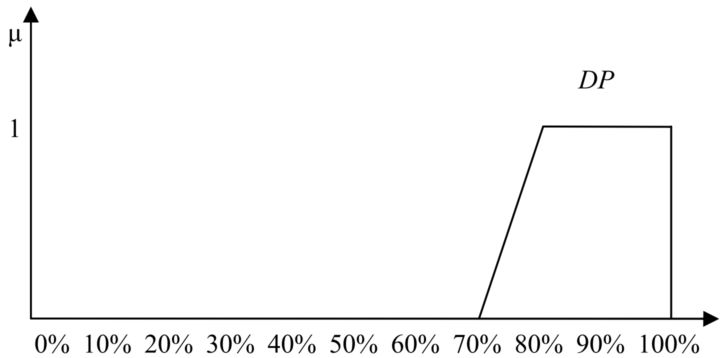
4.15. att. Kritērija C6 vēlamā vērtība



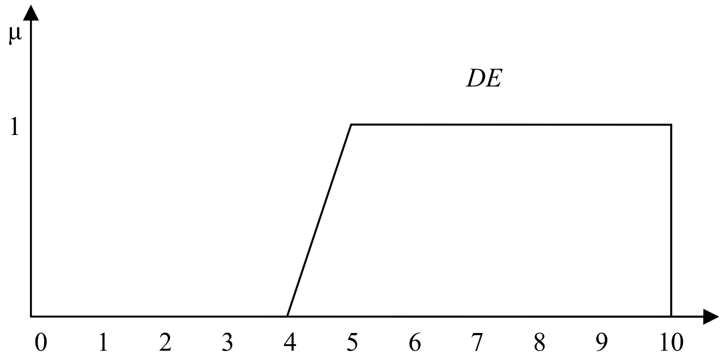
4.16. att. Kritēriju C7 un C8 vēlamās vērtības



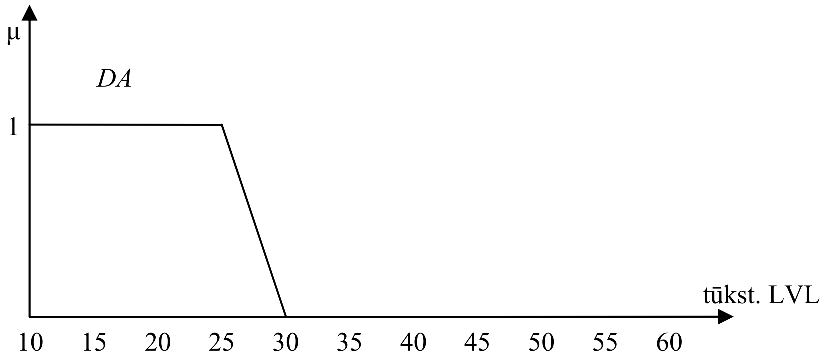
4.17. att. Kritēriju C9, C10, C11, C12, C13, C14 un C15 vēlamā vērtība



4.18. att. Kritērija C16 vēlamā vērtība



4.19. att. Kritērija C17 vēlamā vērtība



4.20. att. Kritēriju C18 un C19 vēlamā vērtība

4.1. tabulā ir norādītas kritēriju vēlamas vērtības. Tās tiks izmantotas alternatīvu salīdzināšanai, jo alternatīvu novērtējums ir atkarīgs no varbūtības, ka tās kritēriju vērtības būs vienādas ar vēlamajām vērtībām.

4.1. tabula

Kritēriju vēlamās vērtības

Kritērijs	**Vēlamā vērtība**
Izejvielu piegādes ērtība – C1	DX1
Produkcijas realizācijas ērtība – C2	DX2
Infrastruktūras un sakaru attīstības līmenis – C3	DY
Rūpnīcas būvēšanas izmaksas – C4	DZ
Prognozējama investīcijas atmaksa – C5	DN
Darba vietu radīšanas potenciāls – C6	DM
Fiksētās izmaksas – C7	DK1
Mainīgās izmaksas – C8	DK2
Ietekme uz mežiem – C9	DL
Ietekme uz ūdenstilpēm – C10	DL
Ietekme uz augsni – C11	DL
Ietekme uz cilvēkiem – C12	DL
Ietekme uz lauksaimniecību – C13	DL
Ietekme uz zvejniecību – C14	DL
Ietekme uz dzīvniekiem – C15	DL
Kvalificēta personāla pieejamība būvēšanas vietā – C16	DP
Apgabala kopējais ekonomiskais stāvoklis – C17	DE
Ar darba personāla pārvietošanu saistītas izmaksas – C18	DA
Ar darba personāla apmācību saistītas izmaksas – C19	DA

4.2.6. Varbūtību vērtību noteikšana

Šajā nodaļā katrai alternatīvai tiks noteiktas varbūtību vērtības tam, ka kritēriju vērtības būs vienādas ar vēlamajām vērtībām. Varbūtību vērtību noteikšanai tiek izmantots tīkls ANGIE [Vališevskis&Borisov, 2002]. Tīkla pamatā ir profesora L. Zadē izstrādāta metode izplūdušo granulu apkopošanai [Zadeh, 1979].

Vispirms izanalizēsim izplūdušas granulas, kuras apraksta katru no alternatīvām. Proti, mēģināsim optimizēt apskatāmus kritērijus un izslēgt tos kritērijus, kuri katrai alternatīvai ir aprakstīti ar vienādo liecību palīdzību. Tādi kritēriji neietekmē alternatīvu sakārtojumu. Kā ir redzams, kritēriji *C5, C6* un *C15* ir aprakstīti ar vienādām liecībām, līdz ar to tos var izslēgt no analīzes.

Ar ANGIE adaptīvā tīkla palīdzību tika noteiktas 4.2. tabulā esošās varbūtību vērtības.

4.2. tabula

Ar ANGIE tīklu izrēķinātas varbūtību vērtības

Alternatīva "Liepusala"	*Alternatīva "Daugavbūda"*	*Alternatīva "Mazupe"*
P(C1 ir DX1) = [0,3 ; 0,48]	P(C1 ir DX1) = [0,4 ; 0,64]	P(C1 ir DX1) = [0,2 ; 0,42]
P(C2 ir DX2) = [0,32 ; 0,5]	P(C2 ir DX2) = [0,35 ; 0,59]	P(C2 ir DX2) = [0,27 ; 0,49]
P(C3 ir DY) = [0,02 ; 0,09]	P(C3 ir DY) = [0,02 ; 0,232]	P(C3 ir DY) = [0,18 ; 0,22]
P(C4 ir DZ) = [0,18 ; 0,32]	P(C4 ir DZ) = [0,3 ; 0,41]	P(C4 ir DZ) = [0,09 ; 0,3]
P(C7 ir DK1) = [0,33 ; 0,43]	P(C7 ir DK1) = [0,14 ; 0,14]	P(C7 ir DK1) = [0,1 ; 0,11]
P(C8 ir DK2) = [0,003 ; 0,01]	P(C8 ir DK2) = [0,01 ; 0,02]	P(C8 ir DK2) = [0,002 ; 0,005]
P(C9 ir DL) = [0 ; 0,25]	P(C9 ir DL) = [0 ; 0,25]	P(C9 ir DL) = [0,5 ; 0,75]
P(C10 ir DL) = [0 ; 0]	P(C10 ir DL) = [0 ; 0]	P(C10 ir DL) = [0 ; 0]
P(C11 ir DL) = [0 ; 0]	P(C11 ir DL) = [0 ; 0]	P(C11 ir DL) = [0 ; 0,25]
P(C12 ir DL) = [0,03 ; 0,1]	P(C12 ir DL) = [0 ; 0]	P(C12 ir DL) = [0 ; 0]
P(C13 ir DL) = [0 ; 0]	P(C13 ir DL) = [0 ; 0]	P(C13 ir DL) = [0 ; 0,025]
P(C14 ir DL) = [0 ; 0]	P(C14 ir DL) = [0 ; 0]	P(C14 ir DL) = [0 ; 0]
P(C16 ir DP) = [0 ; 0,15]	P(C16 ir DP) = [0 ; 0]	P(C16 ir DP) = [0,25 ; 0,65]
P(C17 ir DE) = [0,5 ; 0,66]	P(C17 ir DE) = [0 ; 0,5]	P(C17 ir DE) = [0,34 ; 0,5]
P(C18 ir DA) = [0,07 ; 0,25]	P(C18 ir DA) = [0,05 ; 0,1]	P(C18 ir DA) = [0,32 ; 0,53]
P(C19 ir DA) = [0,15 ; 0,23]	P(C19 ir DA) = [0 ; 0]	P(C19 ir DA) = [0,25 ; 0,45]

Kā ir redzams, visām alternatīvām varbūtība, ka kritēriji *C10, C14* sasniegs vēlamo vērtību ir vienāda ar nulli. Tādējādi, šie kritēriji neietekmē alternatīvu sakārtojumu un tos var izslēgs no analīzes. Tomēr, ir jāņem vērā, ka neviena alternatīva neapmierina šī kritērija vēlamo vērtību.

4.2.7. Informatīvuma noteikšana

Tagad novērtēsim cik informācijas ir pieejams par katru no alternatīvām. Ar šo nolūku pārveidosim lēmumu pieņemšanas modeli informācijas teorētiskajā formā un izrēķināsim entropiju. Pārveidošana tiek veikta sekojošajā veidā: katram alternatīvas kritērijam tiek uzkonstruēta sistēma, kuras stāvokļu skaits ir vienāds ar attiecīgā kritērija iespējamo vērtību skaitu (proti, tas ir vienāds ar izplūdušajās liecībās definēto kritērija vērtību skaitu). Tādējādi, varbūtība, ka sistēma atrodas kādā no stāvokļiem, ir vienāda ar varbūtību, ka atbilstošajam kritērijam būs attiecīgā vērtība. Ja vienam kritērijam atbilst vairākas liecības, tad vispirms ar ANGIE adaptīvā tīkla palīdzību ir jāizrēķina varbūtības, ka kritērijs ir vienāds ar kādām no vērtībām. Vērtību kopumam jābūt tādam, lai attiecīgu varbūtību intervālu robežu summa apmierinātu sekojošo nosacījumu: $p_{min} \leq 1 \leq p_{max}$.

Šajā gadījumā varbūtības ir intervālu, tātad ir jāizmanto promocijas darbā piedāvātā metode intervālu entropijas rēķināšanai (sk. 3.1.2. paragrāfu).

Pēc entropijas vērtību noteikšanas visiem kritērijiem, tās ir jāsasummē, lai iegūtu alternatīvas informatīvuma novērtējumu.

4.3. tabula

Alternatīvas "Liepusala" informatīvums

Sistēma $S_{1,1}$ Trīs stāvokļi ar varbūtībām 0,1 ; 0,3 ; 0,6 Entropija H_1 = 0,9	Sistēma $S_{1,2}$ Entropija H_2 = 0,9	Sistēma $S_{1,3}$ Četri stāvokļi: $P(S_{1,3} \sim q_1)$ = [0,15 ; 0,2] $P(S_{1,3} \sim q_2)$ = [0,22 ; 0,45] $P(S_{1,3} \sim q_3)$ = [0,2 ; 0,32] $P(S_{1,3} \sim q_4)$ = [0,1 ; 0,25] Entropija H_3 =[1,17 ; 1,38]
Sistēma $S_{1,4}$ Entropija H_4 = 1,03	Sistēma $S_{1,7}$ Entropija H_7 = 0	Sistēma $S_{1,8}$ Četri stāvokļi: $P(S_{1,8} \sim q_1)$ = [0,1 ; 0,15] $P(S_{1,8} \sim q_2)$ = [0,07 ; 0,1] $P(S_{1,8} \sim q_3)$ = [0,2 ; 0,4] $P(S_{1,8} \sim q_4)$ = [0,3 ; 0,5] Entropija H_8 = [1,09 ; 1,25]
Sistēma $S_{1,9}$ Entropija H_9 = 0	Sistēma $S_{1,11}$ Entropija H_{11} = 0	Sistēma $S_{1,12}$ Entropija H_{12} = 1,28
Sistēma $S_{1,13}$ Entropija H_{13} = 1,28	Sistēma $S_{1,14}$ Entropija H_{14} = 1,28	Sistēma $S_{1,16}$ Entropija H_{16} = 1,09
Sistēma $S_{1,17}$ Entropija H_{17} = 0	Sistēma $S_{1,18}$ Entropija H_{18} = 1,09	Sistēma $S_{1,19}$ Entropija H_{19} = 1,09
Kopējais entropijas novērtējums alternatīvai $\boldsymbol{H_L}$ ***= [12,2 ; 12,57]***		

4.4. tabula

Alternatīvas "Daugavbūda" informatīvums

Sistēma $S_{2,1}$ Entropija $H_1 = 1,06$	Sistēma $S_{2,2}$ Entropija $H_2 = 1,06$	Sistēma $S_{2,3}$ Četri stāvokļi: $P(S_{1,3} \sim q_1) = [0,2 ; 0,3]$ $P(S_{1,3} \sim q_2) = [0,05 ; 0,15]$ $P(S_{1,3} \sim q_3) = [0,25 ; 0,37]$ $P(S_{1,3} \sim q_4) = [0,1 ; 0,3]$ Entropija $H_3 = [1,05 ; 1,36]$
Sistēma $S_{2,4}$ Entropija $H_4 = 1,03$	Sistēma $S_{2,7}$ Entropija $H_7 = 0$	Sistēma $S_{2,8}$ Četri stāvokļi: $P(S_{1,8} \sim q_1) = [0,09 ; 0,16]$ $P(S_{1,8} \sim q_2) = [0,05 ; 0,12]$ $P(S_{1,8} \sim q_3) = [0,23 ; 0,41]$ $P(S_{1,8} \sim q_4) = [0,35 ; 0,5]$ Entropija $H_8 = [1,05 ; 1,28]$
Sistēma $S_{2,9}$ Entropija $H_9 = 0$	Sistēma $S_{2,11}$ Entropija $H_{11} = 0$	Sistēma $S_{2,12}$ Entropija $H_{12} = 1,28$
Sistēma $S_{2,13}$ Entropija $H_{13} = 1,28$	Sistēma $S_{2,14}$ Entropija $H_{14} = 1,28$	Sistēma $S_{2,16}$ Entropija $H_{16} = 0,9$
Sistēma $S_{2,17}$ Entropija $H_{17} = 0$	Sistēma $S_{2,18}$ Entropija $H_{18} = 1,03$	Sistēma $S_{2,19}$ Entropija $H_{19} = 1,05$
Kopējais entropijas novērtējums alternatīvai ***H_D* = [12,07 ; 12,61]**		

4.5. tabula

Alternatīvas "Mazupe" informatīvums

Sistēma $S_{3,1}$ Entropija $H_1 = 0,94$	Sistēma $S_{3,2}$ Entropija $H_2 = 0,94$	Sistēma $S_{3,3}$ Četri stāvokļi: $P(S_{1,3} \sim q_1) = [0,14 ; 0,23]$ $P(S_{1,3} \sim q_2) = [0,25 ; 0,3]$ $P(S_{1,3} \sim q_3) = [0,15 ; 0,2]$ $P(S_{1,3} \sim q_4) = [0,3 ; 0,45]$ Entropija $H_3 = [1,27 ; 1,38]$
Sistēma $S_{3,4}$ Entropija $H_4 = 1,03$	Sistēma $S_{3,7}$ Entropija $H_7 = 0$	Sistēma $S_{3,8}$ Četri stāvokļi: $P(S_{1,8} \sim q_1) = [0,08 ; 0,14]$ $P(S_{1,8} \sim q_2) = [0,17 ; 0,24]$ $P(S_{1,8} \sim q_3) = [0,2 ; 0,45]$ $P(S_{1,8} \sim q_4) = [0,23 ; 0,27]$ Entropija $H_8 = [1,16 ; 1,34]$
Sistēma $S_{3,9}$ Entropija $H_9 = 0$	Sistēma $S_{3,11}$ Entropija $H_{11} = 0$	Sistēma $S_{3,12}$ Entropija $H_{12} = 1,28$
Sistēma $S_{3,13}$ Entropija $H_{13} = 1,28$	Sistēma $S_{3,14}$ Entropija $H_{14} = 1,28$	Sistēma $S_{3,16}$ Entropija $H_{16} = 1,03$
Sistēma $S_{3,17}$ Entropija $H_{17} = 0$	Sistēma $S_{3,18}$ Entropija $H_{18} = 1,03$	Sistēma $S_{3,19}$ Entropija $H_{19} = 1,03$
Kopējais entropijas novērtējums alternatīvai ***H_M* = [12,27 ; 12,56]**		

Informatīvuma novērtējumu var izmantot kā papildus kritēriju, kurš attēlo cik informācijas ir pieejams par katru no alternatīvām. Turklāt, ja starpība starp novērtējumiem ir liela, tad tas nozīmē, ka par kādu no alternatīvām ir pieejams mazāk informācijas nekā par citām. Šajā gadījumā ir jāapsver iespēja iegūt papildus datus.

4.2.8. Preferenču modelēšana

Preferences tiek raksturotas ar kritēriju svariem. Lai noteiktu svaru vērtības piešķirsim kādam no kritērijiem 100 punktus un baltoties uz tā noteiksim pārējo kritēriju punktu skaitu. Pēc punktu sadalīšanas svarīguma vērtības ir jānormē, lai to summa būtu vienāda ar 1. Šīs analīzes rezultāti ir attēloti 4.6. tabulā:

4.6. tabula

Preferences starp kritērijiem

Kritērijs	**Punkti**	**Svars**
Izejvielu piegādes ērtība – C1	80	0.052288
Produkcijas realizācijas ērtība – C2	100	0.065359
Infrastruktūras un sakaru attīstības līmenis – C3	70	0.045752
Rūpnīcas būvēšanas izmaksas – C4	110	0.071895
Prognozējama investīcijas atmaksa – C5	130	0.084967
Darba vietu radīšanas potenciāls – C6	80	0.052288
Fiksētās izmaksas – C7	90	0.058824
Mainīgās izmaksas – C8	90	0.058824
Ietekme uz mežiem – C9	60	0.039216
Ietekme uz ūdenstilpēm – C10	60	0.039216
Ietekme uz augsni – C11	60	0.039216
Ietekme uz cilvēkiem – C12	70	0.045752
Ietekme uz lauksaimniecību – C13	60	0.039216
Ietekme uz zvejniecību – C14	60	0.039216
Ietekme uz dzīvniekiem – C15	60	0.039216
Kvalificēta personāla pieejamība būvēšanas vietā – C16	110	0.071895
Apgabala kopējais ekonomiskais stāvoklis – C17	60	0.039216
Ar darba personāla pārvietošanu saistītas izmaksas – C18	90	0.058824
Ar darba personāla apmācību saistītas izmaksas – C19	90	0.058824

4.2.9. Daļējs alternatīvu sakārtojums

Lai uzkonstruētu daļēju alternatīvu sakārtojumu katrai alternatīvai ir jāizskaitļo ieejošā un izejošā plūsma, saskaņā ar šādām formulām (sk. 3.8. apakšnodaļu):

$$\phi^{-}\left(A_i\right)=\sum_{k=1}^{m} w_k \cdot {}^{\min}e_k^i \qquad \phi^{+}\left(A_i\right)=\sum_{k=1}^{m} w_k \cdot {}^{\max}e_k^i$$

Daļējs alternatīvu sakārtojums ir definēts ar sekojošo relāciju palīdzību:

$$\begin{cases} aPb, & \text{ja}\ \phi^{-}\left(a\right)>\phi^{+}\left(b\right), \\ aIb, & \text{ja}\ \phi^{-}\left(a\right)=\phi^{-}\left(b\right) \text{un}\ \phi^{+}\left(a\right)=\phi^{+}\left(b\right), \\ aRb, & \text{pretējā gadījumā,} \end{cases}$$

kur P ir pārspēšanas relācija, I ir vienlīdzības relācija un R ir nesalīdzināmības relācija.

Vispirms izrēķināsim plūsmas neņemot vērā entropijas novērtējumu. Alternatīvām ir sekojošās ienākošas un izejošas plūsmas:

$$\begin{aligned} &\phi^{-}\left(A_{Liepusala}\right)=0.104, && \phi^{+}\left(A_{Liepusala}\right)=0.19, \\ &\phi^{-}\left(A_{Daugavbūda}\right)=0.078, && \phi^{+}\left(A_{Daugavbūda}\right)=0.157, \\ &\phi^{-}\left(A_{Mazupe}\right)=0.133, && \phi^{+}\left(A_{Mazupe}\right)=0.257. \end{aligned} \tag{4.1}$$

Iekļaujot analīzē entropijas vērtības, kuras attēlo pieejamas informācijas daudzumu par katru no alternatīvām, tiek iegūti sekojošie rezultāti. Entropijas svars rēķinot plūsmas ir vienāds ar 0,006, kas atbilst 4.6. tabulas 10 punktiem:

$$\begin{aligned} &\phi_{E}^{-}\left(A_{Liepusala}\right)=0.177, && \phi_{E}^{+}\left(A_{Liepusala}\right)=0.266, \\ &\phi_{E}^{-}\left(A_{Daugavbūda}\right)=0.151, && \phi_{E}^{+}\left(A_{Daugavbūda}\right)=0.233, \\ &\phi_{E}^{-}\left(A_{Mazupe}\right)=0.207, && \phi_{E}^{+}\left(A_{Mazupe}\right)=0.332. \end{aligned} \tag{4.2}$$

Kā ir redzams, gan pirmajā (4.1), gan otrajā (4.2) gadījumā visas alternatīvas ir nesalīdzināmas, jo nav dominējušo alternatīvu. Nākošajā nodaļā tiek noteikts pilns alternatīvu sakārtojums.

4.2.10. Pilns alternatīvu sakārtojums

Lai iegūtu pilnu sakārtojumu varbūtības tiek pārveidotas šādā veidā:

$$e_i^j = \frac{{}^{\min}e_i^j + {}^{\max}e_i^j}{2}.$$

Alternatīvas efektivitāte tiek noteikta sekojošajā veidā:

$$p_i = \sum_{k=1}^{m} w_k e_k^i$$

Pilns sakārtojums ir definēts ar sekojošo relāciju palīdzību (sk. 3.8. apakšnodaļu):

$$\begin{cases} a_i P a_j, & \text{if } p_i > p_j \\ a_i I a_j, & \text{if } p_i = p_j \end{cases}$$

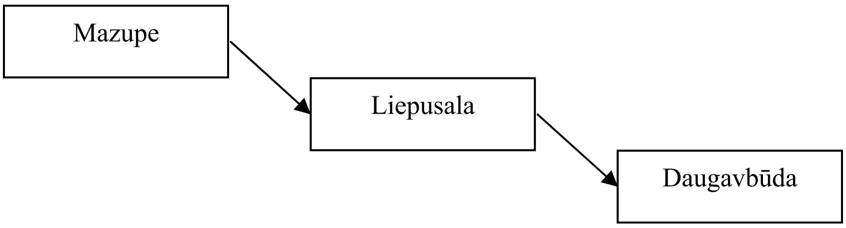
Vispirms izrēķināsim alternatīvu efektivitāti neņemot vērā entropijas vērtību:

$$p_{Liepusala}=0.147, \qquad p_{Daugavbūda}=0.117, \qquad p_{Mazupe}=0.195. \tag{4.3}$$

Ņemot vērā entropija tiek iegūtas sekojošās efektivitātes vērtības:

$$p_{Liepusala}^{E}=0.221, \qquad p_{Daugavbūda}^{E}=0.192, \qquad p_{Mazupe}^{E}=0.269. \tag{4.4}$$

Kā redzams, gan pirmajā (4.3), gan otrajā (4.4) gadījumā alternatīvu sakārtojums ir vienāds un tas ir parādīts 4.21. attēlā.



4.21. att. Alternatīvu sakārtojums

Kā ir redzams, saskaņā ar granulās esošo informāciju, nodefinētajiem kritērijiem un preferencēm vislabākā alternatīva ir rūpnīcas būvēšana Mazupē.

4.2.11. Risinājuma analīze un secinājumi

Uzdevuma risināšanas gaitā tika izanalizētas un novērtētas visas trīs alternatīvas, balstoties uz to aprakstu, izvēlētajiem kritērijiem un preferencēm. Beigās tika iegūti alternatīvu sakārtojumi, kuri raksturo alternatīvu labumu, atkarībā no sniegtās informācijas. Problēma ir tik nenoteikta, ka ar daļējo alternatīvu sakārtojumu nav iespējams noteikt dominējošas alternatīvas. Tādējādi, tiek uzbūvēts pilnais sakārtojums. Tomēr, kā zināms [Brans u.c., 1986] pilnajā sakārtojuma tiek pazusta informācija par analīzes datu neprecizitāti (intervālu novērtējumi tiek pārveidoti precīzajos novērtējumos), tātad tas ir jāuzskata par indikatīvu līdzekli. Lēmums ir jāpieņem lēmumu pieņemošai personai, bet lēmumu pieņemšanas metodes mērķis ir palīdzēt šai personai padarīt problēmu par pārskatāmāku un sniegt papildus informāciju par alternatīvām.

4.3. Izstrādājuma pieprasījuma prognozēšana

Ir jāizstrādā produkta dzīves cikla posmu izmaiņas prognozēšanas metode. Vidi var raksturot kā ļoti nenoteiktu, jo lēmums tiek pieņemts pamatojoties uz datiem, kuri tiek saņemti ar noteiktu aizkāvi. Proti, posma izmaiņas konstatēšana tiek veikta pamatojoties uz tirgus analīzes datiem (pārdošanas apjomi, dati par konkurentiem utt.). Tomēr, šie dati tiek iegūti ar aizkavi, bet dzīves cikla posma izmaiņa ir jānosaka kad tā notiek, jo zaudējumu minimizēšanai nekavējoties ir jāmaina produkta tirgus stratēģija [Comninos, 2002]. Metodei jābūt elastīgai, lai to varētu ātri papildināt ar jauniem ekspertu datiem. Šo uzdevumu var sadalīt šādos apakšuzdevumos:

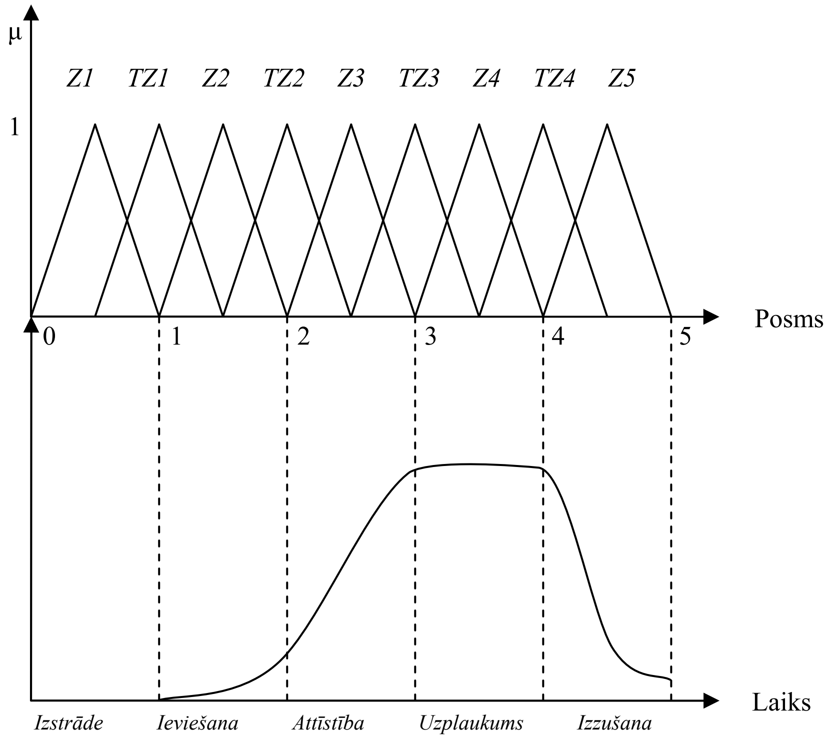
- Ir jānosaka pēc kādiem atribūtiem var atklāt produkta dzīves cikla posma izmaiņu.
- Jānodrošina iespēja produkta modeļa aprakstā izmantot vēsturiskus datus.
- Katrai situācijai ir jāizveido darbību kopums, kuras būtu jāveic pēc produkta stāvokļa noteikšanas.
- Jānodrošina iespēja modeli papildināt un mainīt saskaņā ar no ekspertiem saņemtās informācijas.

4.3.1. Vides analīze

Kā minēts iepriekš, vidi, kuras ietvaros tiek risināts šis uzdevums, var raksturot kā nenoteiktu. Daudziem atribūtiem nav iespējams noteikt precīzu vērtību. Uzdevuma risināšanai tiek izmantotas izplūdušas granulas, kuras ļauj izmantot gan izplūdušo, gan nedeterminēto informāciju. Proti, izplūdušas granulas veido izplūdušas liecības, kuras var uzskatīt par *izplūdušo vērtību varbūtību sadalījumu*.

4.3.2. Kritēriju analīze

Tā kā galvenais uzdevums ir noteikt varbūtību, ka produkts atrodas kādā no tā dzīves cikla posmiem, vispirms definēsim izplūdušās vērtības, kuras atbilst dažādiem produkta dzīves cikla posmiem:



4.22. att. Produkta dzīves cikla posmi un tiem atbilstošas izplūdušās apakškopas

4.22. attēlā izplūdušās apakškopas Z1, ..., Z5 atbilst produkta dzīves cikla posmiem, bet izplūdušās apakškopas TZ1, ..., TZ4 atbilst pārejām no viena dzīves cikla posma uz citu.

Lai gan 4.22. attēlā produkta dzīves cikla līnijai ir noteikta forma, analizētājiem produktiem tai var būt jebkāda forma, jo tiek noteikts izplūdušai vērtībai atbilstošais posms, nevis līkne (precīzāk sākot, tiek noteikta varbūtība, ka produkts *pašreiz* atrodas kādā no posmiem, bet netiek izvirzītas hipotēzes par vispārējo produkta dzīves cikla līkni).

Lai būtu iespējams izveidot modeli, ar kura palīdzību varētu prognozēt kad notiek pāreja no viena posma uz citu, ir jānosaka pēc kādiem atribūtiem var konstatēt pāreju. Literatūrā ir norādīti vairāki šādi parametri [Clifford, 1965], [Comninos, 2002], tie ir pārskaitīti zemāk:

- Informācija par produkta uzvedību pēdējo gadu laikā (ieskaitot cenu, pārdotu vienību skaitu, ieņēmumi, investīciju atmaksa ROI, iekarota tirgus daļa)

- Konkurentu īslaicīgas stratēģijas analīze (jauni produkti, konkurentu izsludinātas ziņas par ražošanas apjomu palielināšanu, ražošanas iekārtu modernizēšana un produktu virzīšana tirgū.
- Konkurentu tirgus daļu analīze.
- Informācija par līdzīgo produktu dzīves ciklu.
- Prognoze par pārdoto vienību skaitu tuvākajā nākotnē.
- Tendences tirgū.

Var pieminēt arī citus faktorus, kuriem ir heiristiska nevis analītiska būtība:

- Uzņēmuma stratēģiskie lēmumu un vadības lēmumi.
- Vēsturisko heiristisko zināšanu izmantošana.

Ir jāņem vērā tas fakts, ka pieejamā informācija nav precīza, turklāt nelielas cenu vai pieprasījuma fluktuācijas var būt par iemeslu neprecīziem secinājumiem par produkta dzīves cikla posmu izmaiņu.

4.3.3. Problēmas apgabala aprakstīšana ar izplūdušo granulu palīdzību

Nākošajā solī pieejamā ekspertu informācija ir jāapraksta ar izplūdušo granulu palīdzību. Zemāk seko produkta aprakstīšanas piemērs:

JA (*prognozējama pārdošanas apjomu samazināšanās* = NAV) ar varbūtību 0.2
TAD posms = Z4
JA (*prognozējama pārdošanas apjomu samazināšanās* = MAZA) ar varbūtību 0.7
TAD posms = TZ4
JA (*prognozējama pārdošanas apjomu samazināšanās* = LIELA) ar varbūtību 0.1
TAD posms = Z5

JA (*konkurentu produktu skaits* = NENOZĪMĪGS) ar varbūtību 0.1 TAD *posms* = Z3
JA (*konkurentu produktu skaits* = LĪDZSVAROTS) ar varbūtību 0.7 TAD *posms* = Z4
JA (*konkurentu produktu skaits* = LIELS) ar varbūtību 0.2 TAD *posms* = TZ4

JA (*iekarota tirgus daļa* = LIELA) ar varbūtību 0.5 TAD *posms* = TZ3
JA (*iekarota tirgus daļa* = VIDĒJA) ar varbūtību 0.4 TAD *posms* = Z4
JA (*iekarota tirgus daļa* = NELIELA) ar varbūtību 0.1 TAD *posms* = TZ4

Šīs granulas veido trīs liecības, kuras definē elastīgus ierobežojumus uz mainīgā "posms" vērtībām. Šo liecību var papildināt ar citām granulām, saskaņā ar pieejamo informāciju. Turklāt, iegūstot precīzākus datus vai papildus ekspertu datus esošas granulas var mainīt. Granulas var papildināt un mainīt saskaņā ar tādiem primārajiem parametriem kā uzņēmuma vadības vai stratēģiskie lēmumi, iepriekšējā pieredze utt. Piemēram, papildus var nodefinēt sekojošās granulas:

JA (*konkurējušo produktu cena* = SAMAZINĀS **un** *tirgus tendences izmaiņa = NEGATĪVA*) ar varbūtību 0.12 TAD *posms* = TZ3
JA (*konkurējušo produktu cena* = SAMAZINĀS **un** *tirgus tendences izmaiņa = NEMAINĀS*) ar varbūtību 0.28 TAD *posms* = Z4
JA (*konkurējušo produktu cena* = NEMAINĀS **un** *tirgus tendences izmaiņa = NEGATĪVA*) ar varbūtību 0.18 TAD *posms* = Z3
JA (*konkurējušo produktu cena* = NEMAINĀS **un** *tirgus tendences izmaiņa = NEMAINĀS*) ar varbūtību 0.42 TAD *posms* = TZ3

Šo granulu var pārrakstīt sekojošajā veidā:

P(posms = Z3) = 0.18
P(posms = TZ3) = 0.12 + 0.42 = 0.54
P(posms = Z4) = 0.28

Nākošajā solī, izmantojot iegūtas liecības ir jānosaka varbūtība, ka produkts pašreiz atrodas kādā no posmiem, šim nolūkam var izmantot adaptīvo tīklu ANGIE, kurš ir paredzēts izplūdušo granulu apstrādei [Vališevskis&Borisov, 2002]. Varbūtību vērtības ir intervālu.

Izmantojot augstāk aprakstītas liecības tika noteiktas sekojošās varbūtību vērtības:

P(posms = Z3) = [0 ; 0.108]
P(posms = TZ3) = [0.052 ; 0.289]
P(posms = Z4) = [0.108 ; 0.393]
P(posms = TZ4) = [0.057 ; 0.181]
P(posms = Z5) = [0 ; 0.09]

4.3.4. Risinājuma analīze un secinājumi

Pēc iegūtajiem rezultātiem var secināt, ka ar vislielāko varbūtību produkta attīstības posms atbilst stāvokļiem TZ3 vai Z4. Proti, ar vislielāko varbūtību notiek pāreja no posma "attīstība" uz posmu "uzplaukums". Gadījumā, ja tiek saņemta papildus informācija no ekspertiem, vai tiek iegūti jauni dati, produkta aprakstu, atribūtus, to vērtības var mainīt un papildināt.

Tika izmantoti tikai daži izstrādātās metodes elementi, jo nostādītājā uzdevumā ir jārēķina varbūtības, tajā nav definētas vairākas alternatīvas, tādējādi visu izstrādāto metožu izmantošana nav nepieciešama.

4.4. Secinājumi par 4. nodaļu

1. Izstrādāto lēmumu pieņemšanas metodi var izmantot dažāda tipa uzdevumu risināšanai.
2. Metožu un līdzekļu izmantošana ir atkarīga no uzdevuma un risināmās problēmas.
3. Ja vide ir ļoti nenoteikta, tad var būt neiespējams uzbūvēt daļēju sakārtojumu, kura būtu dominējošas alternatīvas. Tas nozīmē, ka lēmumu pieņemšanas metodes funkcija ir palīdzēt lēmumu pieņemošai personai labāk izprast problēmu un vidi, kurā tiek pieņemts lēmums.
4. Izplūdušas granulas ir elastīgas izmantošanas ziņā. Tajos var iekļaut izplūdumu un nejaušību. Bet gadījumā, ja kāda no šīm nenoteiktībām neattiecās uz doto parametru, tad var uzdot precīzu un/vai determinētu vērtību.
5. Uz entropijas balstīto alternatīvu informatīvuma novērtēšanas metodi var izmantot, lai noskaidrotu cik informācijas ir pieejams par katru no alternatīvām.

DARBA REZULTĀTI

Promocijas darbā ir izstrādāta lēmumu pieņemšanas metode, kura ir paredzēta izmantošanai vidē ar diviem nenoteiktības avotiem, proti, vidē, kura ir gan izplūdusī, gan nedeterminēta. Metode ir balstīta uz izplūdušajām granulām, kuras aprakstot alternatīvas ļauj izmantot gan izplūdušas vērtības, gan to varbūtības. Darba gaitā radās papildus problēmas, kuras promocijas darbā ir atrisinātas. Piemēram, izstrādājot metodi alternatīvu informatīvuma novērtēšanai, Šenona entropija tika vispārināta intervālu varbūtību gadījumam un tika pierādīts, ka tā ir aditīva, jo izstrādātajā metodē tiek izmantotas intervālu varbūtības. Turklāt, izplūdušo granulu apstrādei ir izstrādāts adaptīvais tīkls ANGIE. Ar tīkla ANGIE apmācības algoritma palīdzību tiek risināts vairākparametru jūtīguma analīzes uzdevumus. Darbs noslēdzas ar praktisko piemēru apskatīšanu, kuri parāda kā izstrādāto metodi var izmantot praksē. Zemāk darbā paveiktais ir aprakstīts sīkāk.

1. Darbā ir izpētītas nenoteiktajai videi paredzētas lēmumu pieņemšanas metodes un tika secināts, ka eksistējošas metodes vai nu neapskata vairāk par vienu nenoteiktības veidu, vai nu tās ir pārāk sarežģītas skaitļošanas ziņā, vai nu agregācijas operatori tajās nepamatoti reducē pieejamo informāciju.
2. Ir parādīts kā izplūdušās granulas var izmantot, lai modeļa aprakstā apvienotu izplūdumu un nenoteiktību. Proti, izplūdušas liecības var uzskatīt par izplūdušo vērtību varbūtību sadalījumu. Līdz ar to modelī var aprakstīt divus dažādus nenoteiktības avotus. Gadījumā, ja kāda no šīm nenoteiktībām neattiecās uz doto parametru, var uzdot precīzu un/vai determinētu vērtību.
3. Ir izstrādāts adaptīvais tīkls ANGIE, ar kura palīdzību var apstrādāt izplūdušās liecības. Skaitļošana šajā tīklā notiek paralēli, kas ļauj paaugstināt skaitļošanas ātrumu izpildot programmu sistēmā, kas atbalsta paralēlo skaitļošanu. Adaptīvā tīkla ANGIE apmācīšanas algoritmu var izmantot lēmumu pieņemšanas modeļa jūtīguma analīzei.
4. Šenona entropija ir vispārināta gadījumam, kad varbūtības ir intervālu. Turklāt, ir pierādīts, ka vispārinātā entropija ir aditīva.
5. Ir izstrādāta metode alternatīvu pilnai un daļējai sakārtošanai. Līdzīgi PROMETHEE metodei, tā balstās uz negatīvajām un pozitīvajām plūsmām.
6. Ir izstrādāta lēmumu pieņemšanas metode, kura ietver sevī augstāk minētos līdzekļus, kuri ir nepieciešami alternatīvu analīzei un lēmuma pieņemšanai. Izstrādātā metode ir paredzēta izmantošanai vidē, kura ir gan nedeterminēta, gan izplūdusī.
7. Izstrādātā metode ir pārbaudīta uz diviem praktiskajiem piemēriem, kas demonstrē kā izstrādāto metodi var pielietot praksē.

DARBA NOVITĀTE

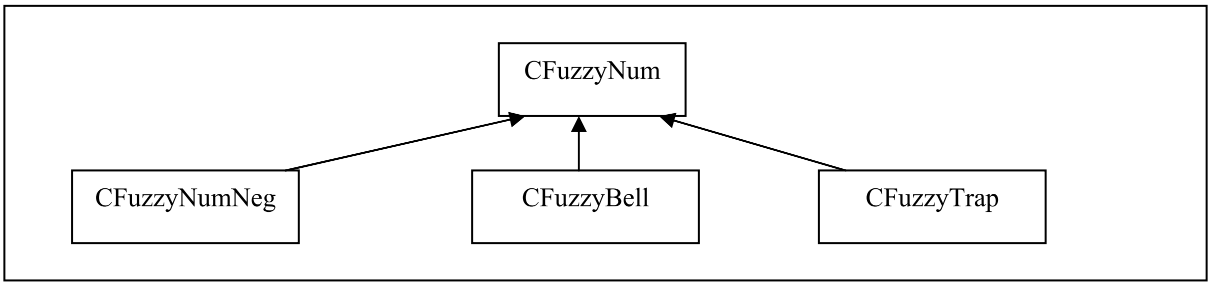
1. Darba galvenais jaunieguvums ir izstrādātā lēmumu pieņemšanas metode, kura ir paredzēta izmantošanai vidē ar diviem nenoteiktības avotiem, proti, vidē, kura ir gan izplūdusī, gan nedeterminēta. No eksistējošiem metodēm tā atšķiras vai nu ar to, ka tā ļauj analīzē iekļaut gan izplūdušo, gan nedeterminētu informāciju, vai nu ar to, ka tā prasa relatīvi maz skaitļošanas resursu un ir praktiski lietojama liela parametru skaita gadījumā.
2. Pārējie jaunieguvumi ir saistīti ar darba gaitā sastaptajām un atrisinātajām problēmām. Proti, izstrādājot metodi alternatīvu informatīvuma novērtēšanai, Šenona entropija tika vispārināta intervālu varbūtību gadījumam, jo izstrādātajā metodē tiek izmantotas intervālu varbūtības. Ir pierādīts, ka vispārinātā entropija ir aditīva.
3. Turklāt, izplūdušo granulu apstrādei ir izstrādāts adaptīvais tīkls ANGIE un algoritms tā apmācībai, kurš balstās uz gradienta lejupslīdes metodi. Tīkls ANGIE tiek izmantots izplūdušu liecību analīzei. Ar tīkla ANGIE apmācības algoritma palīdzību tiek risināts vairākparametru jūtīguma analīzes uzdevums.

PIELIKUMS

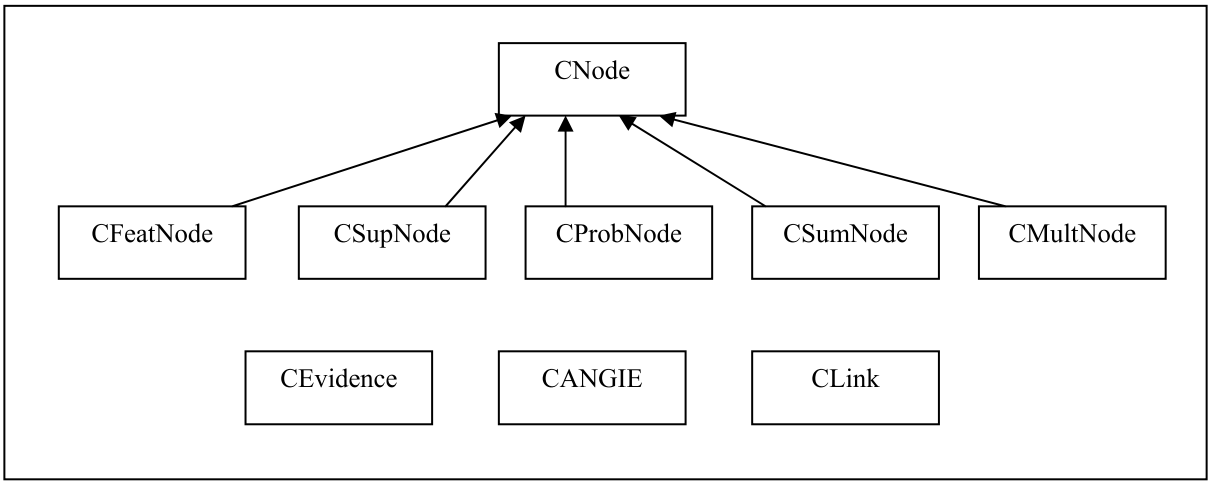
Programmas teksts

Šajā pielikumā ir aprakstītas eksperimentu veikšanai *MS Visual C++ 6.0* vidē izstrādātās klases. Klases ir paredzētas operāciju veikšanai ar izplūdušajām vērtībām, kā arī adaptīvā tīkla ANGIE izveidošanai un izmantošanai.

Klašu hierarhija ir parādīta P.1. un P.2. attēlā.



P.1. att. Klašu hierarhija klasēm, kas reprezentē izplūdušās vērtības



P.2. att. Klašu hierarhija klasēm, kas reprezentē adaptīvo tīklu ANGIE un tā elementus

Vispirms tiek dots klašu kods un pēc tam ir apskatīts kā šīs klases var izmantot izplūdušo granulu apstrādei.

```
///////////////////////////////////////
// System ANGIE
///////////////////////////////////////

class CANGIE
{
	struct cntEl
```

```
	{
		int position;
		int total;
	};

	enum
	{
		EP,
		EC,
	};


public:
	CANGIE();
	double GetOutput();
	void Build();  // Constructs the adaptive network
	void Ask(CFeatNode * quest);  // Set the question
	void SetDesired(double des);  // Set the desired value of the system output
	void SetType(int mode);		// Indicates whether the network calculates EC or
EP
	int GetType();
	void Run();
	void Learn();
	void Load();
	void Save();
	void AddEvidence(CEvidence * evid);
	int GetEvidCount();	// Returns the number of evidences in the system
	void Normalize();	// Apply normalization procedure to the probabilities
	void PrintProbs();
	void WriteProbs(ofstream &outfile);

protected:
	CPtrList Evidences;
	CFeatNode * question;
	double output;
	double desired;		// Desired value of the system output
	int type;			// Indicates whether the network calculates EC (1) or EP
(0)
	int evid_count;		// Number of evidences in the system
	int term_count;		// Number of terms, equal to element count in the 2nd
layer

	CPtrList SupNodes, MultNodes; //, Links;	// Represent nodes and links as linked
lists
	CSumNode SumNode;	// Summation node

private:
	CNode * GetIthNode(CPtrList * list, int i);  // Return the i-th element from the list
	bool IncreaseCounter(CPtrList * cntr);		// Increases the counter, or
returns false
							// if it can't be increased
};

CANGIE::CANGIE()
{
	output = 0;
	desired = 0;
	evid_count = 0;
	term_count = 0;
	type = EP;	// By default, calculate EP
}

void CANGIE::SetType(int mode)
{
	if(mode == EP) type = EP;
	else if(mode == EC) type = EC;
	else printf("Invalid mode\n");
}

int CANGIE::GetType() { return type;}

double CANGIE::GetOutput()
{
	if(type==EP)
		return output;
	else
		return 1 - output;
```

```
}

bool CANGIE::IncreaseCounter(CPtrList * counter)
{
	POSITION posCounter = counter->GetTailPosition();
	cntEl * curCounter;

	do
	{
		curCounter = (cntEl * ) counter->GetPrev(posCounter);
		if(curCounter->position < (curCounter->total - 1))
		{
			curCounter->position += 1;

			POSITION posTmp = posCounter;
			if(posTmp == NULL) posTmp = counter->GetHeadPosition();
			else curCounter = (cntEl * ) counter->GetNext(posTmp);

			curCounter = (cntEl * ) counter->GetNext(posTmp);	//this element is
the head

			if(posTmp != NULL)
				do
				{
					curCounter = (cntEl * ) counter->GetNext(posTmp);
					curCounter->position = 0;
				} while(posTmp != NULL);

			return true;
		}

	} while(posCounter != NULL);

	return false;
}



CNode * CANGIE::GetIthNode(CPtrList * list, int i)
{
	POSITION pos = list->GetHeadPosition();
	CNode * node;
	for(int j=0; j<i; j++)
	{
		if(pos == NULL) {printf("Error in GetIthNode function!\n"); exit(0);}

		node = (CNode *) list->GetNext(pos);
	}

	return node;

}

int CANGIE::GetEvidCount() {return Evidences.GetCount();};

void CANGIE::Ask(CFeatNode * quest)
{
	question = quest;
}


void CANGIE::SetDesired(double des) {desired = des;};

void CANGIE::Load() {};
void CANGIE::Save() {};

void CANGIE::AddEvidence(CEvidence *evid)
{
	Evidences.AddTail(evid);
}

void CANGIE::Build()
{
	// Determine the number of elements in each layer
	evid_count = Evidences.GetCount();

	int i;
```

```
int cnt = 1;

POSITION pos;
CEvidence * evid;

CPtrList counter;
cntEl * element;

for(pos = Evidences.GetHeadPosition(); pos != NULL;)
{
        evid = (CEvidence *) Evidences.GetNext(pos);
        cnt *= evid->GetCount();

        element = new cntEl;
        element->position = 0;
        element->total = evid->GetCount();
        counter.AddTail(element);
}

term_count = cnt;

// Create the necessary objects

CSupNode * sn;
CMultNode * mn;
CLink * link;

for(i=0; i<cnt; i++)
{
        sn = new CSupNode;
        mn = new CMultNode;
        SupNodes.AddTail(sn);
        MultNodes.AddTail(mn);
}

int link_count = 7 * term_count;

// Connect the created nodes with the created links

POSITION pos_evid, pos_sup, pos_mult;
pos_sup = SupNodes.GetHeadPosition();
pos_mult = MultNodes.GetHeadPosition();

CSupNode * curSup;
CMultNode * curMult;

CNode * node, * node2;

do
{
        curSup = (CSupNode *) SupNodes.GetNext(pos_sup);
        curMult = (CMultNode *) MultNodes.GetNext(pos_mult);

        pos_evid = Evidences.GetHeadPosition();
        for(pos = counter.GetHeadPosition();pos != NULL;)
        {
                link = new CLink;
                evid = (CEvidence *) Evidences.GetNext(pos_evid);
                element = (cntEl *) counter.GetNext(pos);

                // Connect elements from the 1st layer to SUP and MULT nodes

                // Connect Feature element to SUP element
                node = GetIthNode(&evid->Features, (element->position) + 1);
                node->Create_Link_To(*curSup, link);

                link = new CLink;

                //Connect Probability element to MULT element
                node = GetIthNode(&evid->Probs, (element->position) + 1);
                node->Create_Link_To(*curMult, link);

        }
} while (IncreaseCounter(&counter));

// Create other links
```

```
        pos_mult = MultNodes.GetHeadPosition();
        for(pos = SupNodes.GetHeadPosition(); pos != NULL;)
        {
                node = (CSupNode *) SupNodes.GetNext(pos);
                node2 = (CMultNode *) MultNodes.GetNext(pos_mult);
                link = new CLink;

                node->Create_Link_To(*node2, link);
                link = new CLink;

                node2->Create_Link_To(SumNode, link);
        }

        // Create link from question to SUP

        for(pos_sup = SupNodes.GetHeadPosition(); pos_sup != NULL;)
        {
                node = (CSupNode *) SupNodes.GetNext(pos_sup);
                link = new CLink;

                question->Create_Link_To(*node, link);
        }
}



void CANGIE::Run()
{
/*      if(type==EP)
                question->fuzzy_value.neg = false;
        else
                question->fuzzy_value.neg = true;
*/

        POSITION pos;
        CEvidence * evid;
        CNode * node;

        // Run the evidences
        for(pos = Evidences.GetHeadPosition(); pos != NULL;)
        {
                evid = (CEvidence *) Evidences.GetNext(pos);
                evid->Run();
        }

        // Run the Question element
        question->Run();

        // Run SUP elements
        for(pos = SupNodes.GetHeadPosition(); pos != NULL;)
        {
                node = (CSupNode *) SupNodes.GetNext(pos);
                node->Run();
        }

        // Run MULT elements
        for(pos = MultNodes.GetHeadPosition(); pos != NULL;)
        {
                node = (CMultNode *) MultNodes.GetNext(pos);
                node->Run();
        }

        // Run the summation node
        SumNode.Run();

        if(type==EP)
                output = SumNode.output;
        else
                output = 1 - SumNode.output;
}



void CANGIE::Learn()
{

        SumNode.desired = desired;
```

```
        // Run the Training procedure at the summation node
        SumNode.Learn();

        POSITION pos;
        CNode * node;
        CEvidence * evid;

        // Run the Training procedure at the multiplication nodes
        for(pos = MultNodes.GetHeadPosition(); pos != NULL;)
        {
                node = (CMultNode *) MultNodes.GetNext(pos);
                node->Learn();
        }

        // Run the Training procedure at the probability nodes in evidences
        for(pos = Evidences.GetHeadPosition(); pos != NULL;)
        {
                evid = (CEvidence *) Evidences.GetNext(pos);
                evid->Learn();
        }
}


void CANGIE::Normalize()
{
        POSITION pos;
        CEvidence * evid;
        for(pos = Evidences.GetHeadPosition(); pos != NULL;)
        {
                evid = (CEvidence *) Evidences.GetNext(pos);
                evid->Normalize();
        }
}


void CANGIE::PrintProbs()
{
        POSITION pos;
        CEvidence * evid;
        for(pos = Evidences.GetHeadPosition(); pos != NULL;)
        {
                evid = (CEvidence *) Evidences.GetNext(pos);
                printf("Evidence %d with name %s:\n",evid->GetID(), evid->GetName());
                evid->PrintProbs();
        }
        printf("\n");

}

void CANGIE::WriteProbs(ofstream &outfile)
{
        POSITION pos;
        CEvidence * evid;
        for(pos = Evidences.GetHeadPosition(); pos != NULL;)
        {
                evid = (CEvidence *) Evidences.GetNext(pos);
                evid->WriteProbs(outfile);
                outfile << "\t";
        }
        outfile << endl;
}



class CFuzzyNum         //Trijstūra izplūdusī vērtība
{
protected:
        double a,b,c;  // Izplūdušās vērtības parametri, a <= b <= c
public:
        CFuzzyNum(double _a=0,double _b=1,double _c=2);
        bool neg;       // Negācija
        virtual int setParams(double aa, double bb, double cc);
// Parametru uzstādīšana
        CString name;  // Izplūdušās vērtības nosaukums
        virtual double getValue(double x);
                        // Piederības funkcijas vērtība punktā X
```

```
};

CFuzzyNum::CFuzzyNum(double _a, double _b, double _c)
{ a = _a; b = _b; c = _c; neg = false; }

int CFuzzyNum::setParams(double aa, double bb, double cc)
{
	if((aa<=bb) && (bb<=cc))
	{
		a=aa; b=bb; c=cc;
		return 0;
	}
	else
	{
		printf("Invalid parameters for a triangular fuzzy number!");
		return -1;
	}
}

double CFuzzyNum::getValue(double x)
{
	double res = 0;
	if(!neg)
	{
		if(x<a) res = 0;
		else if(x<b) res = (x-a)/(b-a);
		else if(x==b) res = 1;
		else if(x<c) res = (x-c)/(b-c);
		else res = 0;
		//printf("Ordinary...\n");
	}
	else
	{
		if(x<a) res = 1;
		else if(x<b) res = (1-(x-a)/(b-a));
		else if(x==b) res = 0;
		else if(x<c) res = (1-(x-c)/(b-c));
		else res = 1;
		//printf("Neg. node: %f\n",res);
	}

	return res;
}

class CFuzzyNumNeg : public CFuzzyNum // Negation of triangular fuzzy number
{
public:
	double getValue(double x);    //Get membership function value at X
};


double CFuzzyNumNeg::getValue(double x)
{
	double res = 0;

	if(x<a) res = 1;
	else if(x<b) res = (1-(x-a)/(b-a));
	else if(x==b) res = 0;
	else if(x<c) res = (1-(x-c)/(b-c));
	else res = 1;

	//printf("Neg. node: %f\n",res);
	return res;
}


class CFuzzyBell : public CFuzzyNum  // Zvanveida izplūdusī vērtība
{
public:
	int setParams(double aa, double bb, double cc);
	double getValue(double x);
};

int CFuzzyBellNeg::setParams(double aa, double bb, double cc)
{
	a=aa; b=bb; c=cc;
	return 0;
```

```
}

double CFuzzyBellNeg::getValue(double x)
{
	return (1 - exp(-pow(((x-c)/a), 2*b)));
}


class CFuzzyTrap : public CFuzzyNum  // Trapecveida izplūdusī vērtība
{
protected:
	double d;  // Papildus parametrs

public:
	CFuzzyTrap(double _a, double _b, double _c, double _d);
	void setParams(double _a, double _b, double _c, double _d);
	double getValue(double x);  // Vērtība punktā X
};

CFuzzyTrap::CFuzzyTrap(double _a, double _b, double _c, double _d)
{
	a = _a; b = _b; c = _c; d = _d;
}

void CFuzzyTrap::setParams(double _a, double _b, double _c, double _d)
{
	a = _a; b = _b; c = _c; d = _d;
}

double CFuzzyTrap::getValue(double x)
{
	double res;
	if(x<a) res = 0;
	else if(x<b) res = (x-a)/(b-a);
	else if(x<c) res = 1;
	else if(x==c) res = 1;
	else if(x<d) res = (d-x)/(c-b);
	else res = 0;
	if(neg) return (1 - res);
	else return res;
}


class CEvidence  // izplūdusī liecība
{
friend class CANGIE;
public:
	CEvidence(CString _name);
	void AddGranule(CProbNode * pn, CFeatNode * fn);
// Pievienot izplūdušo granulu
	void Run(void);       // Nosūtīt granulu vērtības tālāk tīklā ANGIE
	void Learn(void);     // Apmācīt liecību saskaņā ar apm. algoritmu
	int GetCount();       // Granulu skaits liecībā
	void Normalize();     // Normēt varbūtību vērtības
	CString GetName();
	int GetID();
	void PrintProbs();    // Izdrukāt varbūtību vērtības
	void WriteProbs(ofstream &outfile);

protected:
	CPtrList Features;    // Tīkla ANGIE izplūdušo vērtību mezgli
	CPtrList Probs;       // Tīkla ANGIE varbūtības mezgli
	CString name;
	int id;

private:
	static int ticket;
};

CEvidence::CEvidence(CString _name)
{
	name = _name;
	id = ticket++;
}

CString CEvidence::GetName() {return name;}
int CEvidence::GetID() {return id;}
```

```
int CEvidence::ticket = 0;

void CEvidence::Load() {};
void CEvidence::Save() {};

void CEvidence::AddGranule(CProbNode * pn, CFeatNode * fn)
{
	Probs.AddTail(pn);
	Features.AddTail(fn);
}

void CEvidence::Run(void)
{
	POSITION pos;
	CNode * node;
	for(pos = Probs.GetHeadPosition(); pos != NULL;)
	{
		node = (CProbNode *) Probs.GetNext(pos);
		node->Run();
	}
	for(pos = Features.GetHeadPosition(); pos != NULL;)
	{
		node = (CProbNode *) Features.GetNext(pos);

		node->Run();
	}
}

void CEvidence::Learn()
{
	POSITION pos;
	CNode * node;

	for(pos = Probs.GetHeadPosition(); pos != NULL;)
	{
		node = (CProbNode *) Probs.GetNext(pos);

		node->Learn();
	}
}


//double CEvidence::Output() {return out_val;}

int CEvidence::GetCount()
{
	return Probs.GetCount();
}

void CEvidence::Normalize()
{
	POSITION pos;
	CNode * node;
	double neg = 0;		// If there're negative probabilities, shift everything to the
right

	// Checks whether there're negative probabilities
	for(pos = Probs.GetHeadPosition(); pos != NULL;)
	{
		node = (CProbNode *) Probs.GetNext(pos);
		//node->value = 15 * pow(node->value, 3);
		if(node->value < 0) neg = node->value;
	}

	// Determine sum of the probabilities
	double sum=0;
	for(pos = Probs.GetHeadPosition(); pos != NULL;)
	{
		node = (CProbNode *) Probs.GetNext(pos);
		sum += node->value - neg;
	}

	// Normalize the probabilities
	double new_val;
	for(pos = Probs.GetHeadPosition(); pos != NULL;)
	{
```

```
                node = (CProbNode *) Probs.GetNext(pos);
                new_val = (node->value - neg) / sum;
                node->value = new_val;
        }
}


void CEvidence::PrintProbs()
{
        POSITION pos, pos_feat;
        CNode * node, * node_feat;

        pos_feat = Features.GetHeadPosition();
        for(pos = Probs.GetHeadPosition(); pos != NULL;)
        {
                node = (CProbNode *) Probs.GetNext(pos);
                node_feat = (CFeatNode *) Features.GetNext(pos_feat);
                printf("Probability of parameter %s is %f.\n", node_feat->fuzzy_value.name,
node->value);
        }
}


void CEvidence::WriteProbs(ofstream &outfile)
{
        POSITION pos;
        CNode * node;

        for(pos = Probs.GetHeadPosition(); pos != NULL;)
        {
                node  = (CProbNode *) Probs.GetNext(pos);
                outfile << node->value << "\t";
        }
}


class CNode    // Tīkla ANGIE mezgls
{
friend class CEvidence;
friend class CANGIE;
public:
        CNode();
        virtual void SetValue(double val);
        virtual void SetValue(CFuzzyNum fuz_val);
        virtual void GetValue(double & val);
        virtual void GetValue(CFuzzyNum * fuz_val);
        virtual void Run(void) = 0;
        virtual void Learn(void);  // Izpildīt apmācības algoritma iterāciju
        virtual void Load(void);
        virtual void Save(void);
        void SetError(double err) {error = err;};
        void Create_Link_To(CNode &to_node, CLink *link);
// Izveidot saiti starp mezgliem
        int GetID();
        CString GetType();


protected:
        CPtrList *In_Links(void);     // Ieejošo saišu saraksts
        CPtrList *Out_Links(void);    // Izejošo saišu saraksts

        CFuzzyNum fuzzy_value;
        double value;
        double error;
        CString type;
        int id;

private:
        static int ticket;

};

CNode::CNode() {error=0; id=ticket++;}
int CNode::ticket = 0;
int CNode::GetID() {return id;}
CString CNode::GetType() {return type;}
void CNode::SetValue(double val) {value = val;}
```

```
void CNode::SetValue(CFuzzyNum fuz_val) {fuzzy_value = fuz_val;}
void CNode::Learn() {};
//double CNode::TransferFunction(double x) {return x;};
//void CNode::TransferFunction(void) {};

void CNode::Load() {};
void CNode::Save() {};
void CNode::GetValue(double &val) { val = value; }
void CNode::GetValue(CFuzzyNum *fuz_val) { fuz_val = &fuzzy_value; }

CPtrList *CNode::In_Links( void ) { return &in_links; }
CPtrList *CNode::Out_Links( void ) { return &out_links; }

void CNode::Create_Link_To( CNode &to_node, CLink *link )
{
        static int cnt = 1;
        out_links.AddTail(link);
        to_node.In_Links()->AddTail(link);
        link->Set_In_Node(this);
        link->Set_Out_Node(&to_node);
}


class CFeatNode : public CNode  // Tīkla ANGIE izplūdušo vērtību mezgls
{
public:
        CFeatNode();
        void Run(void);        // Veikt rēķinus un nosūtīt vērtības tālāk
};

CFeatNode::CFeatNode() : CNode()
{
        type = "Feature";
}

void CFeatNode::Run()
{
        POSITION pos;
        CLink * link;

        //Send the fuzzy values to all the outgoing links
        for(pos = out_links.GetHeadPosition(); pos != NULL;)
        {
                link = (CLink *) out_links.GetNext(pos);
                link->In_Value(fuzzy_value);
        }
}


class CProbNode : public CNode  // Tīkla ANGIE varbūtības mezgls
{
public:
        CProbNode();
        void Run(void);        // Veikt rēķinus un nosūtīt vērtības tālāk
        void Learn(void);      // Izpildīt apmācības algoritma iterāciju
};

CProbNode::CProbNode() : CNode()
{
        type = "Probability";
}

void CProbNode::Run()
{
        POSITION pos;
        CLink * link;

        //Send the probabilities to all the outgoing links
        for(pos = out_links.GetHeadPosition(); pos != NULL;)
        {
                link = (CLink *) out_links.GetNext(pos);
                link->In_Value(value);
        }
}

void CProbNode::Learn()
{
```

```
        //Sum all the error rates from the links and adjust the probability value
        double delt = 0;
        POSITION pos;
        CLink * link;

        //If the probability is zero, don't adjust the value
        if(value == 0) return;

        for(pos = out_links.GetHeadPosition(); pos != NULL;)
        {
                link = (CLink *) out_links.GetNext(pos);
                delt += (link->error) / value;
        }

        //error = delt * value;
        //printf("Adjustment: %f\n",error);

        if(delt == 0) delt = 0.001;   // Punishment

        //Adjust the probability value
        value -= LEARN_SPEED * delt;
}


class CSumNode : public CNode  // Tīkla ANGIE summēšanas mezgls
{
public:
        CSumNode();
        void Run();
        void Learn();
        double desired; // Vēlamā izejas vērtība
        double output; // Faktiskā izejas vērtība
};

CSumNode::CSumNode() : CNode()
{
        type = "Sum";
}

void CSumNode::Run()
{
        double net = 0;
        double nxt;

        POSITION pos;
        CLink * link;

        // Find the sum of incoming signals
        for(pos = in_links.GetHeadPosition(); pos != NULL;)
        {
                link = (CLink *) in_links.GetNext(pos);
                link->Out_Value(nxt);
                net += nxt;
        }

        output = net;
}

void CSumNode::Learn()        // Execute after Run() is executed
{
        POSITION pos;
        CLink * link;

        // Calculate the error
        error = output - desired;

        // Propagate it to the previous layer
        for(pos = in_links.GetHeadPosition(); pos != NULL;)
        {
                link = (CLink *) in_links.GetNext(pos);
                link->error = error;
        }
}


class CMultNode : public CNode  // Tīkla ANGIE reizināšanas mezgls
{
```

```
public:
	CMultNode();
	void Run();
	void Learn();
};

CMultNode::CMultNode() : CNode()
{
	type = "Multiplication";
}

void CMultNode::Run()
{
	double net = 1;
	double nxt;

	POSITION pos;
	CLink * link;

	// Find the multiplication of incoming signals
	for(pos = in_links.GetHeadPosition(); pos != NULL;)
	{
		link = (CLink *) in_links.GetNext(pos);
		link->Out_Value(nxt);
		net *= nxt;
	}

	// Send the result to all outgoing links
	for(pos = out_links.GetHeadPosition(); pos != NULL;)
	{
		link = (CLink *) out_links.GetNext(pos);
		link->In_Value(net);
	}

	value = net;   // Save this value for learning


}

void CMultNode::Learn()
{
	// Sum all the error rates from the previous layer

	double delt = 0;
	POSITION pos;
	CLink * link;

	for(pos = out_links.GetHeadPosition(); pos != NULL;)
	{
		link = (CLink *) out_links.GetNext(pos);
		delt += (link->error);
	}

	error = delt * value;


	// Sum

	// Propagate the error rate to previous layers
	for(pos = in_links.GetHeadPosition(); pos != NULL;)
	{
		link = (CLink *) in_links.GetNext(pos);
		link->error = error;
	}
}


class CSupNode : public CNode  // Tīkla ANGIE suprēmuma rēķināšanas mezgls
{
public:
	CSupNode(double _from = 0, double _to = 10, int _discr = 50);
	void Run();

private:
	int DISCR; // Diskretizācija
	double from, to; // Intervāla sākums un beigas
};
```

```
CSupNode::CSupNode(double _from, double _to, int _discr) : CNode()
{
	from = _from; to = _to; DISCR = _discr;
	type = "Sup";
}


void CSupNode::Run()
{
	//Find the supremum...
	double supr = 0, temp, cur;
	double step = (to - from) / DISCR;
	POSITION pos;
	CLink * link;
	for(double x = from; x <= to; x += step)
	{
		temp = 1;
		for(pos = in_links.GetHeadPosition(); pos != NULL;)
		{
			link = (CLink *) in_links.GetNext(pos);
			cur = link->fuzzy_value.getValue(x);
			if(cur < temp) {temp = cur;}
		}
		if(temp > supr) {supr = temp;}
	}
	// Now the supremum is represented by variable SUPR, propagate it further
	for(pos = out_links.GetHeadPosition(); pos != NULL;)
	{
		link = (CLink *) out_links.GetNext(pos);
		link->In_Value(supr);
		link->sup_val = supr;
	}
	value = supr;
}

class CLink  // Tīkla ANGIE saite
{
friend class CSupNode;
private:
	static int ticket;
public:
	CLink();
	double error;
	double weight;
	virtual void Set_In_Node(CNode * node);  // Definēt ieejošo mezglu
	virtual void Set_Out_Node(CNode * node);  // Definēt izejošo mezglu
	CNode * In_Node(void);
	CNode * Out_Node(void);
	virtual void In_Value(double val);
	virtual void In_Value(CFuzzyNum fuz_val);
	virtual void Out_Value(double & outval);
	virtual void Out_Value(CFuzzyNum * outval);
	int GetID();
	double sup_val;  // Suprēmuma vērtība, SUP elementu izejošām saitēm

protected:
	int id;
	double value;  // Value for link
	CFuzzyNum fuzzy_value;        // Fuzzy value for a link
	//int value_size;
	CNode * in_node;
	CNode * out_node;
};

CLink::CLink() {error=0; weight=1; sup_val=0; id=ticket++;}
int CLink::ticket = 0;
int CLink::GetID() { return id; }
void CLink::Set_In_Node(CNode *node) {in_node = node;}
void CLink::Set_Out_Node(CNode *node) {out_node = node;}
CNode * CLink::In_Node(void) {return in_node;}
CNode * CLink::Out_Node(void) {return out_node;}
void CLink::In_Value(double val) { value = val; }
void CLink::In_Value(CFuzzyNum fuz_val) { fuzzy_value = fuz_val; }
void CLink::Out_Value(double & val) { val = value; }
void CLink::Out_Value(CFuzzyNum * fuz_val) { fuz_val = &fuzzy_value; }
```

Klašu izmantošana izplūdušo granulu apstrādei

1. Vispirms ir jādefinē *CEvidence* tipa mainīgos, kuri atbilst liecībām.
2. Pēc tam ir jādefinē *CANGIE* tipa mainīgais, kas atbilst ANGIE tīklam.
3. Tad ir jāizveido divdimensiju masīvs, kura elementi ir *CProbNode* un *CFeatNode* tipa. Turklāt, ir jādeklarē viens papildus *CFeatNode* tipa mainīgais, kas atbilst jautājuma mezglam. Neskaidrību gadījumā var apskatīt 3.1.1. paragrāfu, kurā ir aprakstīts tīkls ANGIE.
4. Tad ir jādefinē pietiekošs *CFuzzyNum* tipa vai tā apakštipa (sk. P.1. att.) mainīgo skaits, kas atbilst visām sistēmā izmantojamajiem izplūdušajām vērtībām.
5. Tad ir jādefinē granulu varbūtību vērtības, izmantojot iepriekš nodefinētu masīvu ar *CProbNode* tipa elementiem.
6. Tad *CFeatNode* tipa masīva elementiem ir jāpiekārto atbilstošas izplūdušās vērtības.
7. Ja vēl netika definēta jautājumā esošajai izplūdušajai vērtībai atbilstošais mainīgais, tad tas ir jādefinē un šī vērtība ir jāpiekārto 3. solī speciāli izveidotajam *CFeatNode* tipa mainīgajam. Ja tiek rēķināts apakšējais varbūtības novērtējums, tad atribūtam *CFuzzyNum.neg* jābūt vienādam ar *true*. Pretējā gadījumā tam ir jābūt vienādam ar *false*.
8. Ir jāizveido liecības, ievietojot tajās nodefinētās granulas.
9. Tad liecības ir jāievieto sistēmā ANGIE.
10. Ar *CANGIE::Ask(CFeatNode * quest)* palīdzību var uzstādīt jautājumam atbilstošo izplūdušo vērtību.
11. Procedūra *CANGIE::Build()* konstruē sistēmu ANGIE.
12. Ar *CANGIE::SetType(int mode)* var pārslēgt sistēmas ANGIE režīmu. Proti, ja mainīgais *mode* ir vienāds ar 0, tad tiek rēķināta augšējā robeža, bet ja tas ir vienāds ar 1, tad tiek rēķināta augšējā robeža.
13. Procedūra *CANGIE::Run()* izrēķina sistēmas izejas vērtību, kuru var uzzināt ar funkcijas *CANGIE::GetOutput()* palīdzību.

Zemāk ir kods, kas izrēķina augšējo robežu varbūtībai *P(Y is DX1)*, ja ir dota viena liecība, kura atbilst šādam izplūdušo vērtību sadalījumam:

P(Y is X4) = 0,1

P(Y is X7) = 0,3

P(Y is X9) = 0,6

```
int main(int argc, char* argv[])
{
        CEvidence evid1("Evidence 1"),evid2("Evidence 2"),evid3("Evidence 3");
        CANGIE angsys;

        CProbNode pn[3][4];
        CFeatNode fn[3][4], qn;

        CFuzzyNum X3(2,3,4);
        CFuzzyNum X4(3,4,5);
        CFuzzyNum X5(4,5,6);
        CFuzzyNum X6(5,6,7);
        CFuzzyNum X7(6,7,8);
        CFuzzyNum X8(7,8,9);
        CFuzzyNum X9(8,9,10);
        CFuzzyNum X10(9,10,10);

        pn[0][0].SetValue(0.1);
        pn[0][1].SetValue(0.3);
        pn[0][2].SetValue(0.6);

        fn[0][0].SetValue(X4);
        fn[0][1].SetValue(X7);
        fn[0][2].SetValue(X9);

        CFuzzyTrap DX1(5,6,10,10);
        DX1.neg = false;
        qn.SetValue(DX1);

        evid1.AddGranule(&pn[0][0], &fn[0][0]);
        evid1.AddGranule(&pn[0][1], &fn[0][1]);
        evid1.AddGranule(&pn[0][2], &fn[0][2]);

        angsys.AddEvidence(&evid1);
        angsys.Ask(&qn);
        angsys.Build();
        angsys.SetType(0);
        angsys.Run();
        cout << "EP: " << angsys.GetOutput() << endl;

        return 0;
}
```

IZMANTOTĀS LITERATŪRAS SARAKSTS

[Ben Amor u.c., 2004] Ben Amor S., Jabeur K., Martel J.-M. Multiple criteria aggregation procedure for mixed evaluations// Proceedings of MUDSM-2004, Portugal, Coimbra, September 22-25, 2004.

[Brans u.c., 1986] Brans J.P., Vincke Ph., Mareschal B. How to select and how to rank projects: The PROMETHEE method// European Journal of Operations Research 24, 1986, 228-238 p.

[Chaitin, 2003] Chaitin G. From Philosophy to Program Size. Key Ideas and Methods// Lecture Notes on Algorithmic Information Theory. Estonian Winter School in Computer Science, Palmse, Estonia, March 2-7, 2003, available for download at http://www.umcs.maine.edu/~chaitin/eesti.html

[Clemen&Reilly, 2004] Clemen R.T., Reilly T. Making Hard Decisions. Duxbury Press., 2004, 752 p.

[Clifford, 1965] Clifford D. Managing the Product Life Cycle// European Business Journal, July 1969.

[Comninos, 2002] Comninos I. Product Life Cycle Management, Aristotle University of Thessaloniki, 2002.

[Cornuejols u.c., 1998] Cornuejols G., Trick M. Course Notes on Quantitative Methods for the Management Sciences. Carnegie Mellon University, 1998, 43-53 p.

[Edwards, 1977] Edwards W. How to use multiattribute utility measurement for social decisionmaking// IEEE Transactions on Systems, Man and Cybernetics, 7, 1977, 326-340 p.

[Einstein, 1926] Einstein A. Letter to Max Born (1926)// In: The Little Oxford Dictionary of Quotations. Ratcliffe S. (Ed.), Oxford University Press, 1994.

[Hawking, 2000] Hawking S. Does God Play Dice? Public lectures, 2000, http://www.hawking.org.uk/

[Ishibuchi u.c., 2005] Ishibuchi H., Nakashima T., Nii M. Classification and Modeling with Linguistic Information Granules, Springer-Verlag Berlin Heidelberg, 2005.

[Keeney&Raiffa, 1976] Keeney R.L., Raiffa H. Decisions with Multiple Objectives. John Wiley and Sons, New York, 1976.

[Kosko, 1992] Kosko B. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, 1992.

[Kyburg, 1999] Kyburg H. Interval-Valued Probabilities. SIPTA, http://www.sipta.org/

[Mustajoki u.c., 2001] Mustajoki J., Hamalainen R.P., Salo A. Decision support by interval SMART/SWING – methods to incorporate uncertainty into multiattribute analysis, Manuscript. December 17, 2001.

[Pickett, 2000] Pickett J.P. The American Heritage® Dictionary of the English Language. Fourth edition. Boston, Houghton Mifflin Company, 2000.

[Power, 2002] Power D.J. Decision Support Systems: Concepts and Resources for Managers. Greenwood Publishing, 2002.

[Roy, 1991] Roy B. The Outranking Approach and the Foundations of ELECTRE Methods. Theory and Decision 31, 1991, 49-73 p.

[Rumelhart u.c., 1986] Rumelhart D.E, Hinton G.E. et al. Learning Internal Representation by Error Propagation. Parallel Distributed Systems, MIT Press, Massachusetts, 1986, 318.-362. p.

[Saaty, 1980] Saaty T.L. The Analytic Hierarchy Process. McGraw-Hill, New York, 1980.

[Salo&Hamalainen, 1995] Salo A., Hamalainen R.P. Preference programming through approximate ratio comparisons, European Journal of Operational Research 82, 458-475 p.

[Shafer, 1976] Shafer G. Mathematical Theory of Evidence. Princeton Univ Press, 1976.

[Shannon&Weaver, 1949] Shannon C., Weaver W. The mathematical theory of communication, Urbana: University of Illinois Press, 1949.

[Turban u.c., 2004] Turban E., Aronson J.E., Liang T.-P. Decision Support Systems and Intelligent Systems. Prentice Hall, 2004.

[Vališevskis, 2004] Vališevskis A. Granular-Information-Based Decision Aid Methodology// Proceedings of the International Conference "Managing Uncertainty in Decision Support Models" (MUDSM 2004), Coimbra, Portugal, September 22-24, 2004.

[Vališevskis, 2003] Vališevskis A. Granular Information-Based Decisions// Proceedings of the international conference Information Society and Modern Business, Ventspils, Latvia, January 31 - February 1, 2003, 327-333 p.

[Vališevskis, 2003a] Vališevskis A. Granular Information-Based Risk Analysis in Uncertain Situations// Proceedings of the international conference "Environment. Technology. Resources", Rezekne, Latvia, June 26-28, 2003, 385-391 p.

[Vališevskis, 2002] Vališevskis A. Using Granular-Evidence-based Adaptive Networks for Sensitivity Analysis// In: Merkuryev J, Dzemyda G et al. (eds), Scientific Proceedings of Riga Technical University, vol. 10, 2002, 150-155 p.

[Vališevskis, 2002a] Vališevskis A. Adaptive Learning Algorithm for Hybrid Fuzzy System// Proceedings of the International scientific conference Traditions and Innovations in Sustainable Development of Society, Rezekne, Latvia, February 28-March 2, 281-288 p.

[Vališevskis, 2001] Vališevskis A. Comparative Analysis of Different Approaches towards Multilayer Perceptron Training// Scientific Proceedings of Riga Technical University, vol. 5, 157-167 p.

[Vališevskis&Borisov, 2006] Vališevskis A., Borisov A. F-Granule-Based Decision-Making// Proceedings of the International Conference on Operational Research "Simulation and Optimisation in Business and Industry", SOBI 2006, Tallinn, Estonia, May 17-20, 2006, 164-169 p.

[Vališevskis&Borisov, 2004] Vališevskis A., Borisov A. Risk Analysis in Granular-Information-Based Decision Aid Models// Proceedings of the International Conference on Fuzzy Sets and Soft Computing in Economics and Finance, FSSCEF 2004, St. Petersburg, Russia, June 17-20, 2004, 466-473 p.

[Vališevskis&Borisov, 2003] Vališevskis A., Borisov A. Normalization Issues in Granular Evidence-Based Adaptive Network ANGIE// Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation CIMCA'2003, Vienna, Austria, February 12-14, 2003, 117-124 p.

[Vališevskis&Borisov, 2003a] Vališevskis A., Borisov A. Using Interval-Valued Entropy for Risk Assessment// Proceedings of “Modelling and Simulation of Business Systems”, Kaunas University of Technology Press – Technologija, Lithuania, 2003, 74-78 p.

[Vališevskis&Borisov, 2003b] Vališevskis A., Borisov A. Fuzzy-Granule-Based Information Theoretic Approach towards Risk Analysis// Proceedings of the Second International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control ICSCCW-2003, Antalya, Turkey, September 9-11, 2003, 275-281 p.

[Vališevskis&Borisov, 2003c] Vališevskis A., Borisov A. Design Considerations of the Granular-Evidence-Based Adaptive Network ANGIE// Scientific Proceedings of Riga Technical University, vol. 14., 28-35 p.

[Vališevskis&Borisov, 2002] Vališevskis A., Borisov A. ANGIE: Adaptive Network for Granular Information and Evidence Processing// Proceedings of the Fifth International Conference on Application of Fuzzy Systems and Soft Computing ICAFS-2002, Milan, Italy, September 17-19, 2002, 166-173 p.

[Von Winterfeldt&Edwards, 1986] Von Winterfeldt D., Edwards W. Decision analysis and behavioral research. Cambridge University Press, 1986.

[Whalen, 1994] Whalen T. Interval Probabilities Induced by Decision Problems// Advances in the Dempster-Shafer Theory of Evidence. Editors: Fedrizzi, Kacprzyk, Yager. Wiley, N.Y., 1994, pp. 353-374.

[Yager, 1981] Yager R.R. Approximate Reasoning and Possibilistic Models in Classification// International Journal of Computer and Information Sciences, Vol. 10, No. 2, (1981), 141-175 p.

[Zadeh, 1996] Zadeh L.A. Fuzzy Logic = Computing with Words, IEEE transactions on fuzzy systems, vol. 4, No. 2, May 1996.

[Zadeh, 1988] Zadeh L.A. Fuzzy Logic// Computer Magazine, No.4, 1988, 83-93 p.

[Zadeh, 1986] Zadeh L.A. Is probability theory sufficient for dealing with uncertainty in AI: A negative view// Uncertainty in AI, L.N. Kanal and J.F. Lemmer, eds., Elsevier Science Publishers, New York, N.Y., 1986, 103-116 p.

[Zadeh, 1986a] Zadeh, L.A., A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. AI Magazine, 7(2), pp. 85-90, 1986.

[Zadeh, 1979] Zadeh L.A. Fuzzy Sets and Information Granularity// Advances in Fuzzy Set Theory and Applications, Editors: Gupta M.M., Ragade R.K et al., North-Holland Publishing Company, 1979, 3-18 p.

[Zadeh, 1978] Zadeh L.A. PRUF-a meaning representation language for natural languages// Int. J. Man-Machine Studies 10, 1978, 395-460 p.

[Zadeh, 1965] Zadeh L.A. Fuzzy Sets and Systems// Proceedings of Symposium on System Theory, Polytechnic Institute of Brooklyn, 1965, 29-37 p.

[Zavadskas u.c., 1994] Zavadskas E., Peldschus F., Kaklauskas A. Multiple Criteria Evaluation of Projects in Construction. Vilnius, "Technika", 1994.

[Andrejčikov&Andrejčikova, 2004] Андрейчиков А. В., Андрейчикова О. Н. Анализ, синтез, планирование решений в экономике. «Финансы и статистика», 2004, 464 с.

[Ventcel, 2001] Вентцель Е.С. Теория вероятностей. Седьмое издание, Москва, «Высшая школа», 2001, 575 с.

[Geraščenko, 2002] Геращенко А.Л. Принципиальная модель эволюции природы, 2002, http://www.principics.narod.ru/

[Kolmogorov, 1965] Колмогоров А.Н. Три подхода к определению понятия "Количество информации"// Проблемы передачи информации, т. I , вып. 1, 1965, 3-11 с.

[Polovinkin, 1988] Половинкин А.И. Основы инженерного творчества , Москва, «Машиностроение», 1988, 368 с.

[Černoruckij, 2005] Черноруцкий И.Г. Методы принятия решений, Санкт-Петербург, «БХВ-Петербург», 2005, 416 с.