

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleš Bokal

**Spletna platforma za analizo
hierarhičnih modelov pri odločitvenih
problemih**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aleksander Sadikov

Ljubljana, 2017

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Študent naj prouči procese analize odločitvenih problemov na podlagi metodologije MCDA. Izdela naj orodje za obravnavo odločitvenih problemov in njihovo analizo z uporabo procesov MCDA. Orodje naj vsebuje elemente zasnovane problema, vnos variant in naj grafično prikaže rezultate analiz. Orodje naj bo zasnovano za uporabo na čim večjem naboru naprav.

Zahvaljujem se mentorju viš. pred. dr. Aleksandru Sadikovu in as. dr. Martinu Možini za pomoč in nasvete pri izdelavi diplomskega dela. Prav tako se zahvaljujem Mateju Romihu za sodelovanje pri izdelavi projekta. Hvala tudi družini za podporo v času študija.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Tehnološko ozadje	5
2.1	Razlogi za uporabo tehnologij	5
2.1.1	Neodvisnost	6
2.1.2	Možnosti razširitev	7
2.2	Dostop do projekta	7
3	Uvod v MCDA	9
	Kriterij ali atribut?	10
3.1	MCDA	11
3.1.1	Postavi odločitveni problem	11
3.1.2	Izberi variante	14
3.1.3	Izberi kriterije	14
3.1.4	Oceni variante	16
	Manjvredne variante	17
3.1.5	Določi koristnostne funkcije	17
3.1.6	Uteževanje	17
3.1.7	Analiza rezultatov	20
3.1.8	Analiza občutljivosti	20

4	Analiza	21
4.1	Zvezdni diagram	21
4.2	Normalizirani podatki	22
4.2.1	Normalizacija	24
4.2.2	Leksikografska analiza	25
4.2.3	Maximin	25
4.2.4	Maximax	27
4.3	Manjvredne variante	27
4.4	Map analiza	27
4.5	Prispevek h koristnosti	29
4.6	Občutljivost navzgor	29
4.7	Občutljivost navzdol ali "Kaj če ..."	29
5	Sklepne ugotovitve	35

Povzetek

Naslov: Spletna platforma za analizo hierarhičnih modelov pri odločitvenih problemih.

Namen diplomskega dela je izdelati spletno platformo, ki bo omogočala izdelavo in analizo odločitvenih problemov. Cilj platforme je čim večja neodvisnost od operacijskih sistemov in tipov naprav kot tudi izvajanje vsebinskih nalog, ki jim je namenjena. Prav tako je namenjeno, da je platforma mobilna med osebnimi računalniki kot mobilnimi napravami, na primer telefoni in tablični računalniki. Spletna platforma bo sestavljena iz vnosa podatkov, gradnje hierarhičnega modela in analize modela. To diplomsko delo podrobneje opisuje razvoj in delovanje analize hierarhičnih modelov, metode in tehnologije, uporabljene pri razvoju. Diplomsko delo je hkrati tudi priročnik za uporabo spletne platforme.

Ključne besede: računalnik, splet, odločitveni sistemi, analiza, MCDA, utež, kriterij.

Abstract

Title: A web platform for the analysis of hierarchical models in decision-making problems.

The purpose of this thesis is to design and elaborate web application, which will provide tools for design and analysis of decision problems. The platform aims at maximizing independence from operating systems and different types of devices, as well as executing tasks in decision making process. Application is meant to be used on personal computers as much as mobile devices like, tablets and smart mobile phones. Web platform consists of input of data, designing hierarchical models and model analysis. This thesis describes in detail design and functions of hierarchical model analysis, methods and technologies used in the development. The thesis also represents user manual for the web platform.

Keywords: computer, web, decision systems, analysis, MCDA, weight, criteria.

Poglavje 1

Uvod

Namen diplomskega dela je izdelati aplikacijo, s katero bo mogoče analizirati in vizualizirati odločitvene probleme z uporabo MCDA. Tema odločitvenih sistemov je v svetu dobro poznana, vendar za učenje in prikaz rezultatov ne obstaja nobena platforma, ki bi bila temu namenjena. Tako je v sklopu tega diplomskega dela in diplomskega dela, ki ga je sestavil Matej Romih, nastal koncept za izdelavo platforme, ki bi omogočila analizo hierarhičnih odločitvenih modelov. Programska oprema namenjena tej problematiki sicer obstaja, vendar je namenjena ciljnim skupinam strokovnjakov, ki imajo na voljo sredstva. Tako na primer poskusna različica programske opreme HiView3 omogoča analize z le do petimi kriteriji. Programska oprema tega diplomskega dela torej omogoča analizo modelov brez omejitve števila kriterijev ali katerega drugega segmenta odločitvenega modela.

Projekt izgradnje platforme pod delovnim imenom Friview je razmejen na dve diplomski deli. Prvi del projekta, ki je delo Mateja Romiha, se ukvarja z zajemom podatkov in nastavitvami modela, torej izgradnjo modela za kasnejšo analizo. Vključuje vnos variant, nastavitve kriterijev variant in izgradnjodrevesnega modela za odločitveni problem. Izgradnja drevesnega modela obsega določitev atributov, po katerih bodo variante ocenjevane. To so neposredne lastnosti variant. Če so naše variante računalniki in izbiramo najboljšega za naše potrebe, bodo atributi variante na primer kapaciteta

trdega diska, arhitektura trdega diska, frekvenca procesorja ter kapaciteta RAM pomnilnika. Atribute lahko v drevesnem modelu tudi združujemo v skupna vozlišča. Tako lahko kapaciteto ter arhitekturo trdega diska združimo v skupno vozlišče, ki ga poimenujemo trdi disk. Atributi oziroma neposredne lastnosti v drevesu predstavljajo liste, združeni atributi pa vozlišča. Listom in vozliščem v modelu se nastavijo uteži, ki so osnova za izračun koristnosti variant. Na vozliščih se uteži nastavljajo neposredno s primerjavo njihovih otrok. Na primer v vozlišču trdi disk se nastavitva uteži atributoma kapaciteta trdega diska in arhitektura trdega diska, kjer nastavljammo razmerje, koliko je prvi bolj pomemben od drugega in obratno. Tu še ne upoštevamo, koliko je morebitni atribut cena pomemben proti kapaciteti in arhitekturi, zato se tudi seštevek vseh uteži v modelu ne sešteje v 1. Za to poskrbi normiranje uteži, ki izračuna dejanske uteži. Po končanem sestavljanju se vnesejo variante, med katerimi izbiramo. V tabelo vrednosti se vnese naziv variante in v vsak atribut vrednost variante pri tem atributu. Vnašajo se surove vrednosti, ki v tem koraku še nimajo skupnega razpona. Razponi numeričnih vrednosti se nastavljajo na listih v drevesnem modelu. V prejšnjem koraku pri gradnji drevesa se nastavijo koristnostne funkcije atributa. To so funkcije, ki določajo, kakšno vrednost naj ima varianta pri normalizaciji. Te funkcije vračajo vrednosti od 0 do 100 glede na nastavitve. Funkcije so lahko zvezne ali diskretne. Še posebej pomembne so pri ugotavljanju vrednosti atributov, kot so barve, kjer nastavimo, koliko nam pomeni katera barva.

Drugi del projekta analizira zajete podatke v kombinaciji z nastavitvami modela in predstavi ugotovitve v več različnih analizah, ki prikazujejo posamezne lastnosti modela. V tem delu se najprej izvede normalizacija podatkov na istem razponu vrednosti, ki je kasneje potreben za izračun koristnosti variant. Del analize torej predstavi normalizirane podatke, ki že vsebujejo rezultate leksikografske ureditve in najboljšo varianto izbrano po principih *minimax* in *maximax*. Pri vsakem atributu na pregledu normaliziranih podatkov je prikazana tudi že normirana utež atributa. Za normiranje uteži poskrbi model, medtem ko za računanje koristnosti variant za vsako vozlišče posebej

skrbi del v analizi. Normalizirani podatki so podlaga za vse ostale analize. V drugem delu se izvaja tudi analiza, ki prikazuje prispevek vsakega atributa h končni vrednosti variante. Ta prispevek je za vsak atribut sestavljen iz njegove normalizirane vrednosti in njemu določene uteži. Vsako varianto tako v grafu predstavlja stolpec, ki je razdeljen na deleže, ki predstavljajo posamezni atribut. Sklop analiz sestavljata tudi dve analizi občutljivosti. Prva je imenovana “Kaj če ...” analiza, znana tudi kot občutljivost navzdol, ki za vsak atribut pove, kako občutljiv je za spremembe uteži. Poleg občutljivosti ta analiza tudi prikaže, katera varianta bi po povečanju ali zmanjšanju uteži atributa zmagala namesto trenutne zmagovalne variante. Druga analiza občutljivosti imenovana Občutljivost navzgor je prikazana kot graf linearnih funkcij in prikazuje vsako varianto, trenutno utež in koliko se lahko utež poveča ali zmanjša, da se rezultat analize ne spremeni. Ta analiza zajema tako attribute kot vozlišča. Za vizualni prikaz variant in razlik med njihovimi normaliziranimi vrednostmi skrbi zvezdni diagram, kjer je vsak atribut variante predstavljen kot krak v zvezdi. Map analiza se ukvarja s primerjavami med vozlišči ali listi. Tako lahko ob primerno zasnovanem modelu primerjamo variante v smislu dobrih proti slabim lastnostim in na grafu vidimo, katere variante so boljše ali katere variante so slabše od vseh ostalih. Na zavihku z analizami so izpostavljene tudi manjvredne variante. To so variante, ki so po vseh atributih slabše ali enake neki drugi varianti in niso v nobenem primeru kandidatke za najboljšo varianto.

Poglavje 2

Tehnološko ozadje

Spletna platforma je sestavljena pretežno z uporabo HTML strani in je programirana v jeziku Javascript. Tako je uporaba kar se da enostavna, saj se direktorij z aplikacijo naloži na pomnilnik v napravi in zažene prvo HTML stran aplikacije. S tem se izognemo uporabi aplikacijskega strežnika in dosežemo, da je naša aplikacija neodvisna in prenosna. Izdelana aplikacija deluje v spletnem brskalniku, kar pomeni, da lahko aplikacijo uporabljamo na namiznih računalnikih ali mobilnih napravah.

Uporabili smo:

- HTML,
- Javascript,
- jQuery,
- JSON,
- Javascript knjižnice (JQWidgets, D3)

2.1 Razlogi za uporabo tehnologij

Glavni vodili pri razvoju naše aplikacije za pomoč pri odločanju sta bili prenosnost in odprtost, kar je privedlo do odločitve, da bo naša aplikacija

spletna platforma. Sprva je nabor mogočih tehnologij obsegal samostoječe aplikacije na določenem operacijskem sistemu, kot so .exe aplikacije v operacijskem sistemu Windows. Takšen pristop smo opustili, saj želimo doseči, da bi kar se da veliko uporabnikov lahko uporabljalo aplikacijo. Tak pristop bi izločil uporabnike Mac OS X operacijskega sistema in uporabnike Linux distribucij. Spletno platformo smo si sprva zamislili kot MVC spletno aplikacijo, saj obstaja veliko okolij, ki so dostopna in nudijo širok nabor funkcionalnosti. Ta vrsta tehnologije se je izkazala za nustrezno, saj potrebujejo spletne aplikacije tega tipa aplikacijski strežnik, ki ni bil na voljo. Drugi razlog je bil, da so MVC aplikacije namenjene delu s podatki iz podatkovnih baz, v naši aplikaciji pa količina podatkov ni v takšnem obsegu, da bi za to potrebovali podatkovno bazo.

2.1.1 Neodvisnost

Cilj aplikacije je bil, da bi bila kar se da neodvisna od operacijskih sistemov, brskalnikov in specifičnih tehnologij, ki jih brskalniki podpirajo ali ne podpirajo. Skratka uporabnika nismo želeli omejevati s specifikacijami naprave, na kateri aplikacijo uporablja. Ugotovili smo, da so uporabljene tehnologije najbolj razširjene na vseh sodobnih napravah, kjer bi bila aplikacija uporabljena, kot so osebni računalniki, tablični računalniki in pametni telefoni. Neodvisnost pomeni tudi neodvisnost aplikacije same, namreč aplikacija za delovanje potrebuje osnovno strojno opremo naprave in ne potrebuje aplikacijskega strežnika. Tako izključimo nedostopnost aplikacije v primeru težav z aplikacijskim strežnikom. Aplikacijo je seveda mogoče naložiti na spletni strežnik in zagotoviti dostop do nje v primeru, če želimo nuditi storitev, kjer uporabnik ne potrebuje aplikacije na lokalnem datotečnem sistemu in do nje samo dostopa, vendar pa to ni pogoj za uporabo te aplikacije.

2.1.2 Možnosti razširitev

S tem pojmom smo želeli zajeti širok nabor možnosti nadgradenj z razmeroma enostavno implementacijo. Pri analizah modela se lahko enostavno in brez poseganja v že izdelane dele aplikacije dodajajo novi moduli z analizami, ki uporabljajo zbrane podatke. Uporabljene tehnologije, še posebej uporaba Javascripta omogočajo enostavno integracijo z drugimi knjižnicami tega programskega jezika. Za namen grafičnega prikaza rezultatov analiz oziroma metod odločanja je bila uporabljena Javascript knjižnica JQWidgets, ki nudi širok nabor prikaza v obliki grafov, diagramov in tabel. Vključeni so tudi gradniki uporabniškega vmesnika, kot so orodne vrstice, meniji, gumbi in različni okvirji. Uporaba teh knjižnic omogoča bolj sodobno programsko opremo z možnostjo prilagoditev kot uporaba bolj zaprtih tipov gradnje uporabniških vmesnikov, na primer javanska knjižnica Swing.

2.2 Dostop do projekta

Glede na naravo in namen projekta je izvorna koda tudi javno objavljena na spletnem mediju GitHub, ki je namenjen shranjevanju odprtokodnih projektov in nazdoru različic.

Naslov za dostop do izvorne kode projekta je:

- <https://github.com/martinmozina/friview>

Projekt je odložen na repozitoriju as. dr. Martina Možine z razlogom ohranjanja konsistentnosti in administracije repozitorija, če bi v prihodnosti še kdo nadgrajeval ta projekt.

Poglavje 3

Uvod v MCDA

MCDA (Multi-criteria decision analysis) ali slovensko večkriterijska odločitvena analiza predstavlja pristop in tudi nabor metod, s katerimi ocenimo variante pri nekem odločitvenem problemu in natančno ovrednotimo nasprotujoče si kriterije[1], s katerimi variante ocenjujemo. Nasprotujoče si kriterije ponavadi predstavljajo stroški, ki jih primerjamo s koristmi ali učinkovitostjo na drugi strani. V vsakdanjem življenju se srečujemo z odločitvami, katerih podlaga je največkrat intuicija. Kadar pa gre za pomembne odločitve z veliko težo, oziroma so posledice teh odločitev velike in imajo velik vpliv, pa uporabimo metodične pristope, kot je na primer odločitvena analiza.

Analiza odločanja se uporablja za reševanje odločitvenih problemov na različnih področjih, tako v gospodarskih organizacijah kot v javnih ustanovah, kjer je potrebno predlagane rešitve utemeljiti[2]. Prav tako je uporabna tudi za odkrivanje kompromisov pri podobnih možnostih[3]. Večkriterijska analiza je način pogleda na probleme v smislu organizacije podatkov in deljenja kompleksnega problema na manjše, bolj obvladljive enote. Namen MCDA ni odločanje kot tako, temveč je njena vloga pomoč pri odločanju. Glavne naloge te analize so, da predstavi podatke, oceni variante in s tem pomaga pri reševanju problema.

Princip večkriterijske odločitvene analize deluje po naslednjih korakih:

1. postavi odločitveni problem,

2. izberi variante,
3. izberi kriterije,
4. oceni variante (surove vrednosti),
5. določi koristnostne funkcije (preslikave iz surovih vrednosti),
6. uteževanje,
7. analiza rezultatov,
8. analiza občutljivosti.

To diplomsko delo se ukvarja z analitičnim pogledom na odločitveni problem in je del skupnega projekta v sodelovanju z Matejem Romihom, čigar delo temelji na izgradnji odločitvenega modela. Če pogledamo korake večkriterijske analize odločanja, jih ne moremo razdeliti na polovico in obravnavati samo prve ali druge polovice, saj so nekateri koraki pri analizi pomembni od začetka do konca analize. Koraka analiza rezultatov in analiza občutljivosti sta v celoti stvar tega diplomskega dela, s korakoma, ki zajemata določevanje koristnostnih funkcij in ocenjevanje variant, pa se v večji meri ukvarja delo Mateja Romiha. Za ostale korake velja, da so pomembni pri obeh delih projekta, saj je na primer ključni korak za analizo, torej postavitev odločitvenega problema, pomemben pri zasnovi modela, kot tudi za analizo problema. Za potrebe utemeljevanja razlag metod in njihovih rezultatov to diplomsko delo opiše vse korake pri odločitveni analizi MCDA, deli, ki so v večji meri usmerjeni k analizi odločitvenega problema, pa so podrobneje opisani.

Kriterij ali atribut?

Pri obravnavi odločitvenih modelov se bomo srečevali z dvema izrazoma, to sta *kriterij* in *atribut*. Kriteriji so vsi listi in vozliča v drevesu odločitvenega modela, atributi pa so samo listi. Ko govorimo o atributih, so to neodvisne

Vozlišče/list	kriterij	atribut
lastnosti	✓	
senzor	✓	
cena	✓	✓
baterija	✓	✓
teža	✓	✓
resolucija	✓	✓
kvaliteta_fotografij	✓	✓

Tabela 3.1: Kriteriji in atributi.

lastnosti variante od drugih lastnosti. Kriteriji pa lahko zajemajo tudi vozlišča, v katera so združeni atributi. Tabela 3.1 prikazuje attribute in kriterije iz drevesa na sliki 3.2.

3.1 MCDA

To poglavje govori o procesu analiziranja odločitvenega problema od začetka po korakih kako zastaviti vprašanje, oziroma postaviti odločitveni problem, ki je temelj naše analize, kako lahko nepremišljeno zastavljen odločitveni problem pripelje do napačne analize, kako pomemben je pravilen pogled na variante in kako dobro sestavljen nabor kriterijev pripelje do pravilne analize. V nadaljevanju bodo poglavja podrobneje opisala: postavitve odločitvenega problema, izbor variant, izbor kriterijev, ocenjevanje variant, določanje koristnostnih funkcij, uteževanje kriterijev ter analizo rezultatov in občutljivosti.

3.1.1 Postavi odločitveni problem

To je prvi korak pri našem problemu, kjer je pomembno, da čimbolj natančno opredelimo problem. Postavitve problema je videti trivialna, vendar je ta korak zelo pomemben in predstavlja temelj naše analize. Slabo zastavljen odločitveni problem pomeni manj natančen ali celo napačen rezultat,

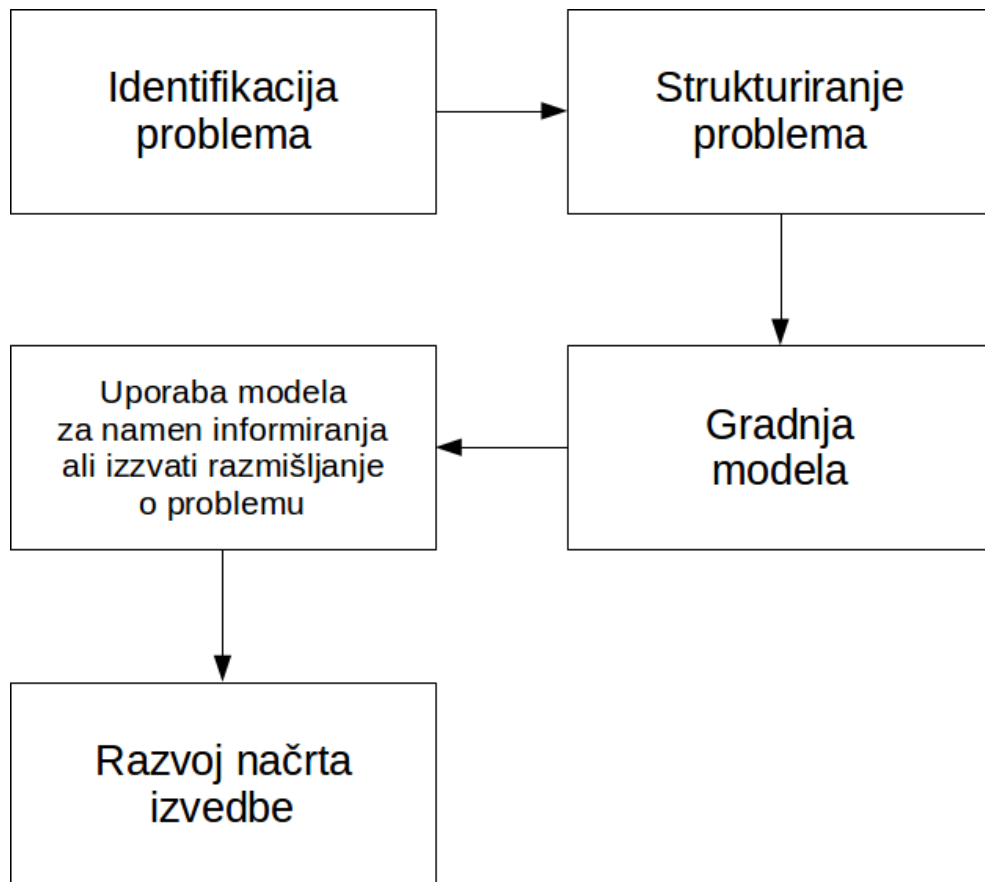
namreč z enakim naborom kriterijev lahko rešujemo več različno zastavljenih problemov in nočemo rešiti "napačnega" vprašanja. Torej, dobro strukturiran problem sestoji iz[4]:

- nabora variant ali odločitvenega prostora, med katerimi se odločamo,
- nabora kriterijev, s katerimi ocenimo variante,
- modela ali metode, ki ju bomo uporabili za odločanje.

Pri strukturiranju problema oziroma postavljanju vprašanja ponavadi nismo edini, ki odločamo, zato je potrebno upoštevati vse interesne skupine, ki pričakujejo tak ali drugačen rezultat analize. Pri vsaki odločitvi, osebni ali znotraj organizacije, je verjetno, da se bodo pojavile različne interesne skupine. Naj bodo to tisti, ki odločitev sprejemajo, ali tisti, na katere odločitev vpliva. Koliko se te skupine upoštevajo, je stvar tistega, ki bo problem zasnoval.

Razlogi, kot so različni pogledi interesnih skupin, rezultat kreativnega razmišljanja o problemu ali posledica kritičnega pogleda na vsebino problema, lahko privedejo do različnih oblik problema, s katerim se nameravamo ukvarjati. Na primer:

- **Problem 1: Kakšen računalnik potrebuje petčlanska družina, da z najmanjšim stroškom zamenja starega z novim?** Kdo bo računalnik uporabljal? Kakšen je namen uporabe, urejanje besedil ali zahtevni grafični procesi?
- **Problem 2: Koliko računalnikov potrebuje petčlanska družina, da nihče od članov ne bo prikrajšan za uporabo?** Kolikšen delež potrebe po uporabi računalnika zahteva vsak družinski član? S čim se družinski člani ukvarjajo? Tu se pojavijo različna nova vprašanja, na primer ali potrebujemo zelo zmogljiv računalnik, da zadovoljimo potrebe, ali več računalnikov z zmogljivostmi, usmerjenimi v določena področja za različne skupine uporabnikov?



Slika 3.1: Proces analiziranja v MCDA.

Pri sestavljanju problemskega vprašanja je potrebno upoštevati tudi psihološki vidik. Če gledamo na problem kot na vprašanje, je zelo pomembno, kako sestavimo besedilo vprašanja. Že postavitev besed v vprašanju namreč vpliva na različne rezultate. Če na primer v vprašanje postavimo besedi “segrevanje ozračja”, lahko dobimo drugačne podatke kot v primeru, če vprašanje vsebuje besedi “podnebne spremembe” [5]. Drugi primer vpliva vprašanja na odgovor je lahko točna referenca na objekt. Če vprašanje sprašuje po točno določeni zadevi, je večja verjetnost, da nam bo tisti, ki odgovarja, dal manj točen oziroma manj odločen odgovor, kot če vprašanje sprašuje po zadevi na splošno [6].

3.1.2 Izberi variante

Varianta je posamezni objekt v množici, ki ga opazujemo in primerjamo z ostalimi, in hkrati predstavlja enega od možnih rezultatov[7], med katerimi mora tisti, ki odloča, izbirati. V tem delu procesa sestavimo nabor variant, med katerimi izbiramo. Primer nabora variant so lahko različni kandidati za delovno mesto.

Pri izbiranju variant ločimo izvedljive (ustrezne) in neizvedljive oziroma neustrezne. Ustrezna varianta zadovolji vse omejitve, ki jih je odločevalec določil. Te omejitve lahko imenujemo predkriteriji, saj niso dejanski kriteriji, po katerih bomo variante ocenjevali, ampak zgolj okvirji, v katerih naj bodo variante, med katerimi se odločamo. Primer je odločitveni model za nakup šopka rož. Odločimo se, da bomo naš šopek ali rože ocenjevali glede na barvo, in sicer so rože lahko rdeče, rumene, oranžne ali zelene, nikakor pa ne smejo biti roza barve. Torej so variante šopkov, ki vsebujejo roza barvo, neustrezne.

3.1.3 Izberi kriterije

Kriterij je lastnost variante, s katerim primerjamo variante med seboj. Lahko ga opišemo kot funkcijo na množici variant A , s katero si pomagamo primerjati dve varianti[8], glede na zorni kot, s katerega opazujemo varianto, in nam vrne empirični podatek za varianti x in y , torej $g(x)$ in $g(y)$.

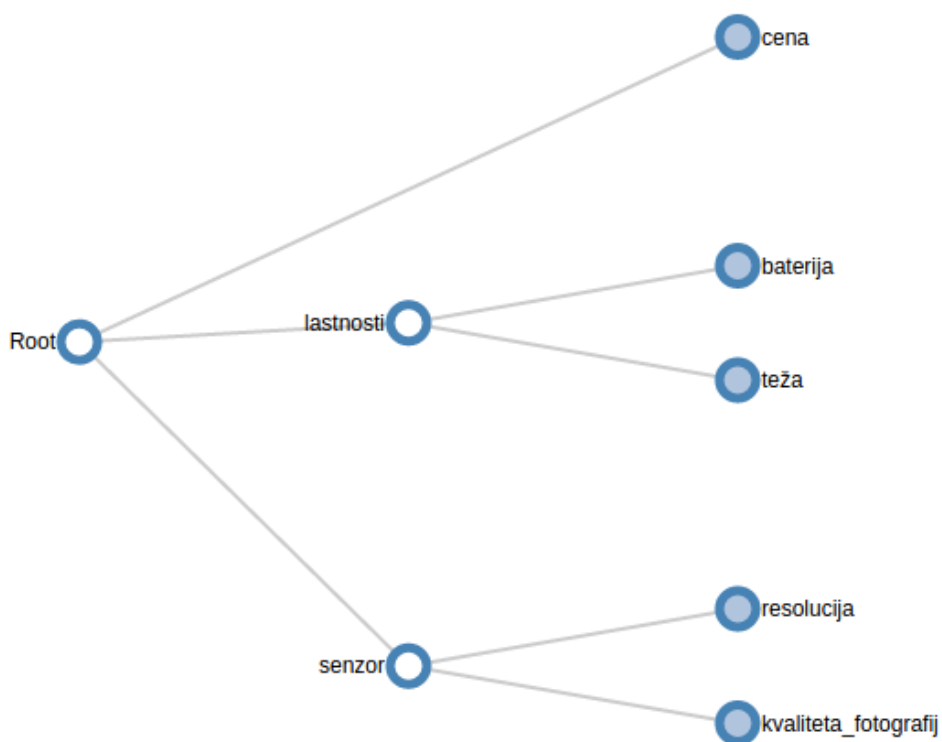
Če smo v prejšnjem koraku iskali primernega kandidata za delovno mesto med prijavljenimi, sedaj določimo, po katerih lastnostih jih bomo ocenjevali.

Pri izbiri oziroma sestavljanju kriterijev je dobro, da vsi, ki so udeleženi v postopku odločanja, razumejo te kriterije in jih sprejemajo. To pomeni, da se vse interesne skupine strinjajo z njimi, tudi če se ne strinjajo, koliko so ti kriteriji pomembni[9]. Ko je kriterij sestavljen, se morajo vse interesne skupine strinjati tudi z metodo, po kateri kriterij vrača vrednosti variant za določeno lastnost.

Kriterije vizualiziramo v obliki drevesa (slika 3.2), kjer listi drevesa predstavljajo kriterije, po katerih ocenjujemo varianto. Z vozlišči si lahko poma-

gamo tako, da združujemo kriterije v listih v smiselne skupine. Prav tako v vozlišča lahko združujemo druga vozlišča [3], odvisno od kompleksnosti odločitvenega problema. Smiselne skupine so množice kriterijev ali množice podmnožic kriterijev, za katere se odločimo, da imajo skupne lastnosti. Lastnosti, kot sta *barva* in *oblika*, lahko združimo v vozlišče *izgled*. Glavni razlogi za združevanje kriterijev so: procesu omogoča lažji nadzor glede primernosti kriterija za analizo našega problema ter olajšanje procesa analize, saj lahko primerjamo naenkrat celo skupino kriterijev proti nekemu drugemu kriteriju, ki je sicer neodvisen od skupine, ali pa primerjamo skupino kriterijev proti drugi skupini kriterijev. Za sestavo dobrega nabora kriterijev je potrebno biti pozoren na nekaj lastnosti dobre prakse:

- **Popolnost:** Smo s svojim naborom zajeli vse lastnosti variant, ki jih potrebujemo za našo analizo?
- **Odvečnost:** Smo določili kriterije, ki jih pravzaprav ne potrebujemo? Obstajajo kriteriji, katerih podatki se prekrivajo z drugimi podatki? Odvečen kriterij je tudi tisti, ki bo na rezultat analize vplival zanemarljivo.
- **Izvedljivost:** Pomeni, da je vsako varianto možno opisati s tem kriterijem. Za doseg dobre izvedljivosti kriterija je ponavadi priporočljivo razdeliti kriterij na podkriterije.
- **Medsebojna neodvisnost vrednosti:** Če lahko priredimo vrednost varianti pri nekem kriteriju, ne da bi poznali vrednost te variante pri nekem drugem kriteriju, to pomeni, da je kriterij neodvisen.
- **Dvojno štetje:** Se pojavi v primeru, če podatke zbiramo v obliki vprašanj in se vsebina teh vprašanj prekriva. Vprašanje lahko sprašuje po informacijah, za katere že imamo odgovor iz drugega vprašanja.
- **Število kriterijev:** Preveliko število kriterijev vodi k večjemu naporu pri zbiranju in organiziranju podatkov, pri tem pa tako (preveliko) število kriterijev zanemarljivo vpliva na točnost rezultata.



Slika 3.2: Drevo kriterijev.

- **Vplivi, ki se pokažejo čez čas:** Če izvajamo analizo za ukrep, ki bo trajal dalj časa, je potrebno upoštevati tudi ta zorni kot. Na tej točki je pomembno oceniti tudi to, ali bo kriterij še pomemben ob koncu analize.

3.1.4 Oцени variante

Ko so variante izbrane, jih ocenimo, da izločimo izstopajoče variante. Eden od načinov je, da ugotovimo, katere variante so manjvredne od ostalih. Va-

riante lahko izstopajo tudi po vrednostih atributov. Če je eden od atributov cena ali strošek variante, so nekatere lahko občutno predrage ali preveč cene, da bi jih sploh upoštevali pri našem problemu.

Manjvredne variante

Varianta A je manjvredna, če obstaja varianta B, ki se od A razlikuje v vrednosti vsaj enega atributa in za vsak atribut velja, da je njegova koristnost pri varianti B večja ali vsaj enaka koristnosti istega kriterija v varianti A. Če je A v več kriterijih slabša od B, je A manjvredna. Za nemanjvredno varianto lahko trdimo, da vedno obstaja nek model, kjer bi ta varianta lahko zmagala.

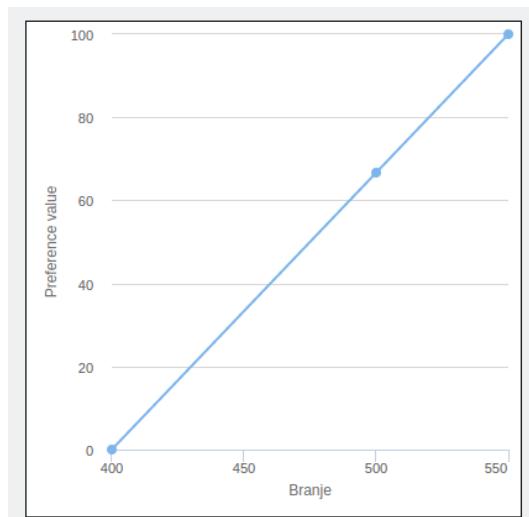
3.1.5 Določi koristnostne funkcije

Pri tem koraku vsakemu atributu variante, torej lastnosti, po kateri jo ocenjujemo, določimo koristnostno funkcijo. Ta funkcija prevede surovo vrednost variante v normalizirano vrednost, in sicer vrne vrednosti od 0 do 100[10].

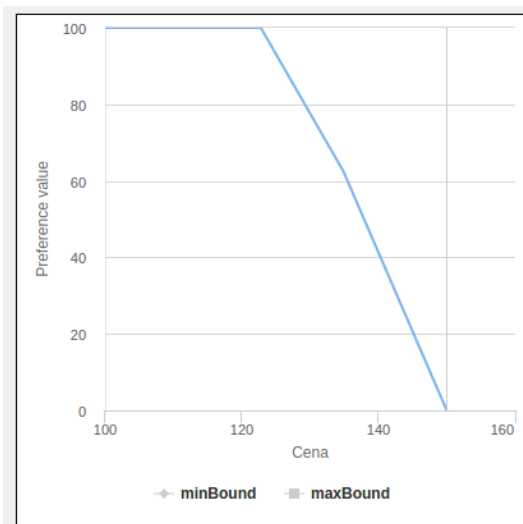
Koristnostne funkcije so različne za različne vrste podatkov. Poznamo zvezne in diskretne koristnostne funkcije, kjer se slednje uporabljajo pri diskretnih vrednostih, na primer barvah, če jih predstavimo z imeni in ne frekvencami ali šolskimi ocenami, kjer ni vmesnih vrednosti. Zvezne koristnostne funkcije pa ločimo na linearne in nelinearne. Linearne (slika 3.3) funkcije poznamo tudi v obliki odsekovnih funkcij (slika 3.4), ki jim lahko določimo padanje koristnosti do neke surove vrednosti in od te vrednosti naprej naraščanje ali stagniranje.

3.1.6 Uteževanje

Vsak list in vozlišče utežimo glede na njegovo pomembnost tako, da na vsakem vozlišču določimo utež njegovih otrok. Uteži nastavljamo v razponu od 0 do 100. Najbolj znan način uteževanja, ki je uporabljen tudi v aplikaciji, je model utežene vsote (ang: weighted-sum model) [11, 12]. Če predpostavimo,



Slika 3.3: Primer linearne koristnostne funkcije.



Slika 3.4: Primer linearne odsekovne koristnostne funkcije.

	C1	C2	C3	C4
Uteži	0.35	0.25	0.25	0.15
V1	0	20	100	50
V2	55	100	0	100
V3	100	0	75	25

Tabela 3.2: Model utežene vsote.

da imamo M variant in N kriterijev, lahko izračunamo najboljšo varianto A^* z naslednjim izrazom:

$$K^* = \max_{M \geq i \geq 1} \sum_{i=1}^N a_{ij} w_j \quad (3.1)$$

kjer je K^* koristnost najboljše variante, a_{ij} predstavlja vrednost variante i pri atributu j in normirani uteži w_j tega atributa. Enačba 3.2 predstavlja izraz za izračun koristnosti posamezne variante i , kjer je w_j utež trenutnega kriterija j in a_{ij} vrednost variante pri trenutnem kriteriju v tabeli 3.2.

$$\text{koristnost}_i = \sum_{j=1}^n w_j a_{ij} \quad (3.2)$$

Torej se koristnost za varianto V1, kjer bo $i = 1$ in $N \geq j \geq 1$ izračuna:

$$\text{koristnost}_1 = 0.35 * 0 + 0.25 * 20 + 0.25 * 100 + 0.15 * 50 = 37.5 \quad (3.3)$$

Pri računanju koristnosti variant v odločitvenih modelih upoštevamo vrednosti samo neodvisnih kriterijev, torej atributov. Prav tako morajo biti uteži normirane, saj jih v celotnem drevesu ne delimo po deležih, ki bi se sešteli v 1, ampak z vrednostjo na intervalu $[0,100]$ povemo, koliko so otroci vozlišča pomembni med seboj. Pri tem ne upoštevamo staršev vozlišča ali otrok otrok vozlišča. Za nas uteži normira aplikacija[13].

Model utežene vsote je pri odločanju potrebno uporabiti na normaliziranih in ne surovih podatkih, kjer so vse vrednosti variant pri atributih izražene v istem razponu, ponavadi od 0 do 100. Če uporabimo surove podatke, lahko

pride do velikih razlik v vrednostih in postanejo kriteriji zanemarljivi zaradi majhnih vrednosti, kljub temu da jim mogoče določimo (naj)večjo težo [14].

3.1.7 Analiza rezultatov

V ta sklop spadajo metode, s katerimi ocenimo ali opazujemo variante [15]. To so prikaz normaliziranih podatkov z utežmi, analize variant z minimax, maximax in leksikografsko ureditvijo. Prav tako rezultate analiziramo s pomočjo razčlenitve koristnosti variant na posamezne deleže, ki jih prispevajo atributi k skupni koristnosti. Pri analiziranju rezultatov ugotavljamo, zakaj je neka varianta zmagala in kako blizu ji druge variante sledijo, lahko pa naredimo tudi primerjavo variant po izbranih kriterijih. Map analiza nam omogoča primerjavo vseh variant po dveh kriterijih naenkrat, njena najbolj razširjena uporaba je opazovanje hkrati stroškov in koristi variant, torej za kakšen strošek dobimo največ koristi[16].

3.1.8 Analiza občutljivosti

Pri analizi občutljivosti se sprašujemo, kako občutljivi so kriteriji na spremembe uteži. Predvsem se večkrat vprašamo, “kaj če ...” se zgodi to ali ono. Kako bo vplivalo na rezultat analize, če nekemu kriteriju povečamo utež? Izračun občutljivosti nam tudi pove, za koliko se lahko utež kriterija spremeni, pa to ne bo vplivalo na trenutno najboljšo varianto. Analiza občutljivosti nam prikaže območja v modelu, ki lahko vplivajo na rezultat analize modela, zato je pogosto uporabljena za ugotavljanje do katere meje je smiselno upoštevati razlike v mnenjih interesnih skupin[3, 17]. Tako poznamo analizi **občutljivost navzdol** in **občutljivost navzgor**, ki ocenjujeta občutljivost glede na pogled z vrha ali dna drevesnega modela.

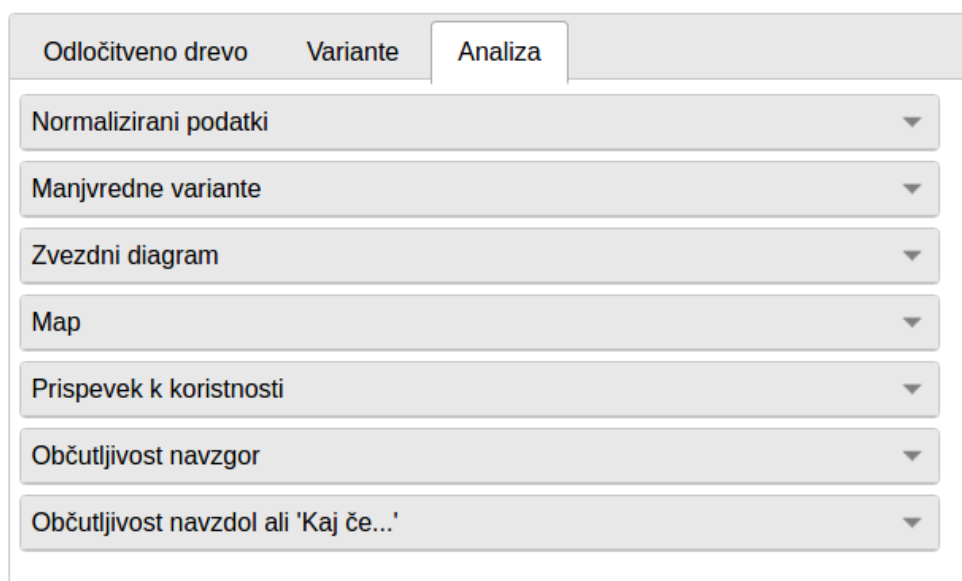
Poglavje 4

Analiza

Predel analize v aplikaciji predstavlja zavihek *Analiza* na sliki 4.1 in združuje posamezne okvirje, ki vsebujejo prikaze rezultatov posameznih metod analiziranja. Nekateri okvirji, na primer Normalizirani podatki, vsebujejo več analiz skupaj. To poglavje vsebuje analizo rezultatov in občutljivosti ter vsako metodo ovrednotenja variant podrobneje opiše.

4.1 Zvezdni diagram

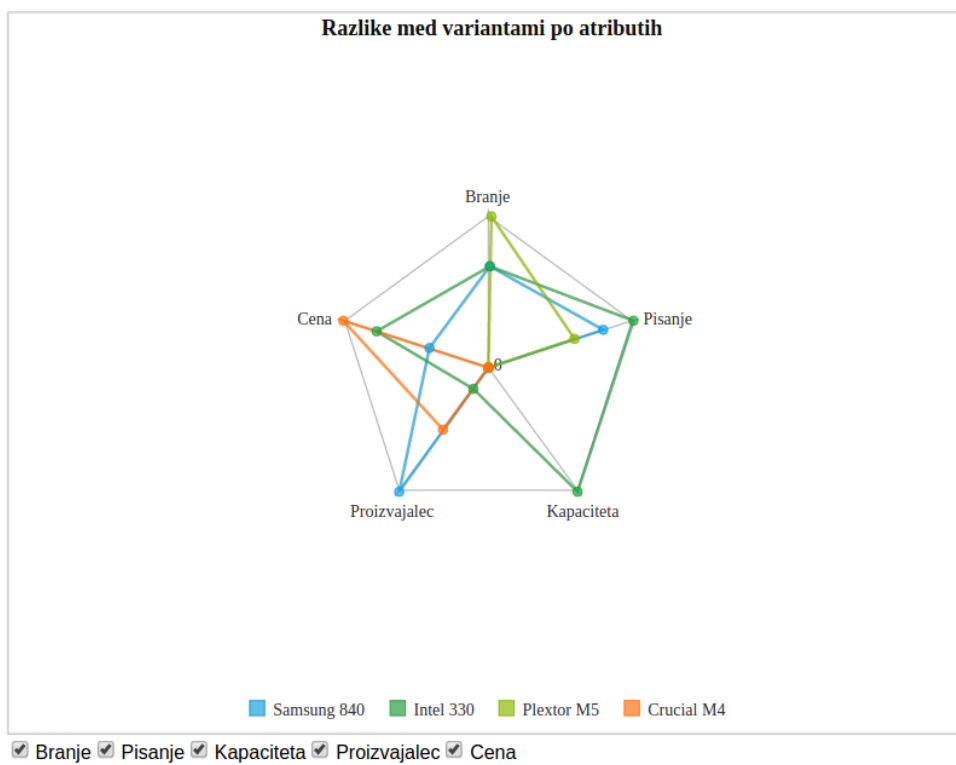
Graf (slika 4.2) je namenjen hitremu vizualnemu pregledu našega modela po variantah in kriterijih. Vsaka smer ali krak grafa pomeni atribut variante, barve linij pa ločijo variante med seboj. Center grafa ima vrednost 0, medtem ko najbolj zunanje meje grafa pomenijo največje vrednosti. Ko so točke na črtah, ki predstavljajo kriterije, povezane, tvorijo lik, katerega ploščino opazujemo. Lik (varianta) z največjo ploščino predstavlja tudi najbolj verjetno varianto. Iz diagrama tudi razberemo, v katero smer se nagiba varianta. Pri tem diagramu lahko primerjamo vse kombinacije atributov in variant. Lahko izberemo, katere attribute želimo opazovati in jih označimo, aplikacija pa nam prikaže samo zelene attribute. Prav tako lahko opazujemo le določene variante, tako da jih označimo v legendi diagrama.



Slika 4.1: Zavihek z analizami.

4.2 Normalizirani podatki

Normalizirane podatke predstavlja tabela, ki jo uporabljamo za vse ostale analize. Podatki so pridobljeni iz koristnostnih funkcij, ki smo jih nastavili v modelu. Ti podatki še ne vsebujejo uteženih vrednosti. Prikazujejo samo vrednosti posameznih variant v istem podatkovnem okvirju od 1 do 100. Slika 4.3 predstavlja izgled normaliziranih podatkov v aplikaciji. Poleg podatkov tabela vsebuje še tri dodatne analize, in sicer leksikografsko, maximin in maximax. Na začetku izbrane variante po kriterijih razporedimo v tabelo, ki vsebuje surove vrednosti takšne, kot so. Ta tabela se nahaja na zavihku Variante na sliki 4.1. V kakšni domeni zapisujemo vrednosti, ni pomembno, saj se kasneje podatki za analizo normalizirajo. Primer surovih podatkov je v tabeli 4.1.



Slika 4.2: Zezdni diagram.

Varianta	cena[0.35]	kvaliteta_fotografij[0.35]	resolucija[0.15]	baterija[0.15]
[1] A [MAXIMIN] [MAXIMAX]	33.33	100.00	100.00	25.00
[2] B	100.00	100.00	0.00	0.00
[3] C	0.00	0.00	100.00	100.00

Go to page: Show rows: 1-3 of 3

Slika 4.3: Prikaz normaliziranih podatkov.

Varianta	Teža	Baterija	Cena	Resolucija	Kvaliteta fotografij
A	450	600	800	25	80
B	550	500	600	20	80
C	750	900	900	25	70
D	850	800	1000	20	70

Tabela 4.1: Surovi podatki.

4.2.1 Normalizacija

Normalizacija pomeni prevedbo podatkov, zajetih z različnimi metodami, merskimi enotami ali v različnih razponih vrednosti, na skupno skalo [18, 12], ki jo potrebujemo, če želimo variante po kriterijih primerjati med seboj. Pri normalizaciji podatkov igrajo vlogo koristnostne funkcije [13], ki smo jih določili pri gradnji modela. Te nam vrnejo normalizirane vrednosti variante glede na določeno funkcijo. Vrnjene vrednosti so na intervalu $[0, 100]$, kjer je 0 najslabša in 100 vedno najboljša vrednost variante po določenem kriteriju. Najboljši in najslabši vrednosti se privzeto priredita 0 in 100, ne glede na vhodne vrednosti variant. Na primer, če imamo variante A, B in C s surovimi vrednostmi kriterija C1: 500, 750, 1000 in želimo z linearno koristnostno funkcijo normalizirati podatke, bo pri tem kriteriju minimalna vrednost 500 in maksimalna vrednost 1000, torej bodo normalizirani podatki za kriterij C1: 0, 50, 100. Lahko se zgodi, da kriterij opazujemo na drugačnem razponu vrednosti. Če kriterij C1 predstavlja ceno računalnika, mi pa želimo podatke opazovati z razponom vrednosti v resničnem življenju, lahko kriteriju nastavimo minimalno in maksimalno opazovano vrednost, na primer od 0 € do 1000 €, linearna koristnostna funkcija bo vrnila podatke za C1: 50, 75, 100.

Varianta	Teža	Baterija	Cena	Resolucija	Kvaliteta fotografij
A	100	25	33	100	100
B	67	0	100	0	100
C	0	100	0	100	0

Tabela 4.2: Normalizirani podatki brez manjvrednih variant.

4.2.2 Leksikografska analiza

V tabeli normaliziranih podatkov so variante razporejene od najboljše označene z 1 do najslabše n , kjer je n število variant. Variante so označene na sliki 4.3 z “[1]”, “[2]” in “[3]” pred imenom variante. V tem primeru je po Leksikografski analizi A najboljša in C najslabša varianta.

Leksikografska analiza sloni na principu urejanja besed v slovarju. Torej tako, da najprej uredimo variante po najpomembnejšem kriteriju, potem po drugem in tako naprej po vseh kriterijih, dokler ne pridemo do edinstvenega zaključka oziroma zmagovalne variante[19].

4.2.3 Maximin

Rezultat te analize je najboljša varianta, zato pri maximin najboljšo varianto označimo z $[MAXIMIN]$ v tabeli na sliki 4.3. Pri maximin analizi iščemo varianto z največjim minimumom. Za primer si vzemimo vrednosti iz tabele 4.2. Vrednosti za varianto A so 100, 25, 33, 100 in 100. Za vsako varianto poiščemo minimum med njenimi vrednostmi. V tabeli minimumov 4.3 za A je $\min(A) = 25$, sledi $\min(b) = 0$ in $\min(c) = 0$. Najboljša varianta po maximin analizi je torej A. Če se pojavita dva kandidata z enakim največjim minimumom, pozabimo vse trenutne minimalne vrednosti in začnemo novo iteracijo ugotavljanja največje najmanjše vrednosti, kot je prikazano v tabeli 4.4.

Varianta	Minimum variante
A	25
B	0
C	0

Tabela 4.3: Minimum variant za maximin analizo.

1. iteracija			
Varianta	Vrednost 1	Vrednost 2	Vrednost 3
A	50	75	80
B	20	60	100
C	70	50	85
2. iteracija			
Varianta	Vrednost 1	Vrednost 2	Vrednost 3
A	X	75	80
B	X	60	100
C	70	X	85

Tabela 4.4: Tabela vrednosti za maximin z istim miniumom.

4.2.4 Maximax

Najboljšo varianto po analizi maximax predstavimo tako, da poleg naziva variante napišemo oznako [*MAXIMAX*]. Ta analiza deluje po istem principu kot maximin, le da tukaj iščemo varianto z največjim maksimumom.

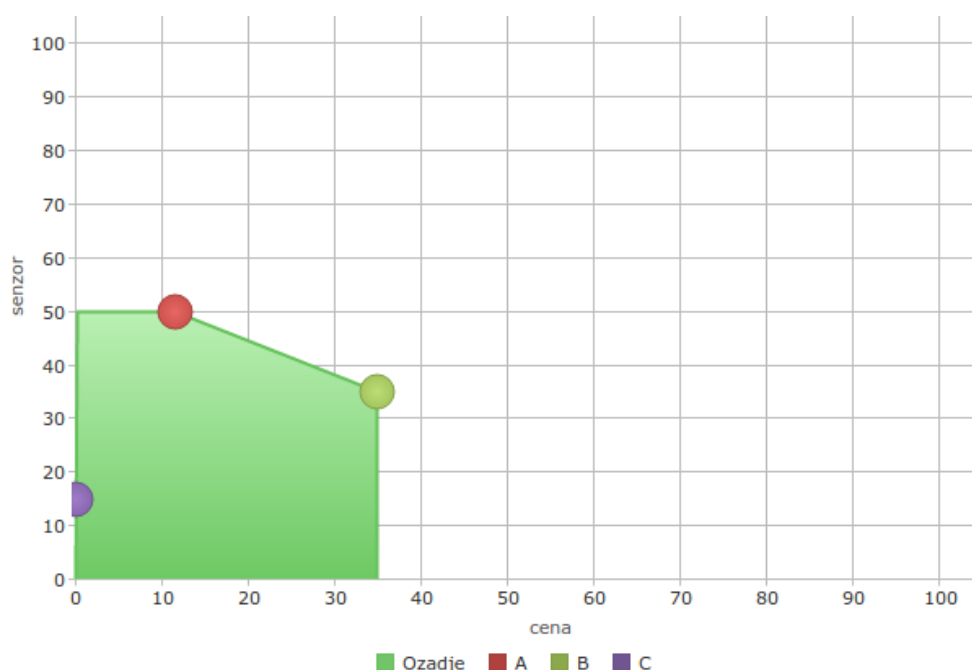
4.3 Manjvredne variante

Pri tej analizi gre le za prikaz manjvrednih variant v modelu, zato je izgled tega dela analize precej trivialen. Manjvredne variante računamo tako, da vsako varianto primerjamo z vsako drugo varianto. Za primer lahko vzamemo podatke iz tabele 4.1. Vsaka varianta je na začetku kandidat za manjvredno varianto. Ali je *A* manjvredna, ugotovimo po kriterijih. Teža variante *A* je 450 (pri tem kriteriju velja, da je manj boljše), $B = 550$, $C = 750$ in $D = 850$. Nobena vrednost ostalih variant ni boljša od *A*, zato že po teži *A* ne more biti manjvredna.

Varianta *D* je primer manjvredne variante, saj je v vsakem kriteriju neka druga varianta enako dobra ali boljša. Pri teži je boljša že varianta *C*, pri trajanju baterije *D* ni najslabša, ampak je *C* vseeno boljša. Pri kriterijih resolucija in kvaliteta fotografij pa ima ta varianta enakovredni tekmi. Iz tega sledi, da varianta *D* ne bo v nobenem kriteriju zmagala, zato jo označimo kot manjvredno in izločimo iz modela.

4.4 Map analiza

Map analiza predstavlja podatke po principu Paretove učinkovitosti. S to analizo primerjamo vrednosti variant po dveh kriterijih ali skupinah kriterijev med seboj. Podatki so prikazani na grafu, ki je v osnovi razpršeni diagram, s tem da razpon vrednosti sega od 0 do 100. Vsaka točka na diagramu predstavlja varianto, katere koordinati sta vrednosti pri vsakem diagramu. Slika 4.4 prikazuje map analizo med sensorjem in ceno variante za izbor najboljšega fotoaparata. Najboljše variante prikazane na grafu so najdlje



Slika 4.4: Map analiza kriterija senzor proti kriteriju cena.

oddaljene od koordinatnega izhodišča in najbližje točki (100,100). Zato je dodana funkcionalnost ozadja obarvanega zeleno. Variante, ki ležijo na robu zelenega ozadja, sodijo med najboljše variante po izbranih kriterijih, medtem ko za variante, ki so v celoti znotraj meja ozadja, vemo, da so zagotovo slabše od variant na robu[20].

Iz analize na sliki 4.4 torej lahko razberemo naslednje. Varianta C nikakor ne pride v poštev, saj je po obeh kriterijih slabša od A in B. Za A velja, da ima od vseh najboljši senzor, vendar je v ceni slabša od variante B, ki ima najboljšo ceno. Pri teh dveh variantah se sprašujemo, kaj nam pomeni več: dober senzor, ki je tudi dražji, ali cenejši in slabši senzor. Opazimo, da ima varianta B manj slab senzor, kot pa je cenejša od A. In obratno, razlika v senzorju med A in B je premajhna, da bi A upravičil razliko cene B.

4.5 Prispevek h koristnosti

Pri tej analizi gre za vizualizacijo deleža koristnosti, ki jo vsak atribut prispeva h končni koristnosti variante. Slika 4.5 prikazuje deleže koristnosti za tri variante A, B in C. Deleži stolpca vsake variante predstavljajo koristnosti kriterijev. Pri varianti C kvaliteta fotografij ne predstavlja deleža, saj je na tem področju ocenjevanja varianta C najslabša in ima vrednost 0. Delež vsakega atributa se izračuna z množenjem normirane uteži atributa in normalizirane vrednosti variante pri tem atributu.

$$D_{ij} = a_{ij}w_j \quad (4.1)$$

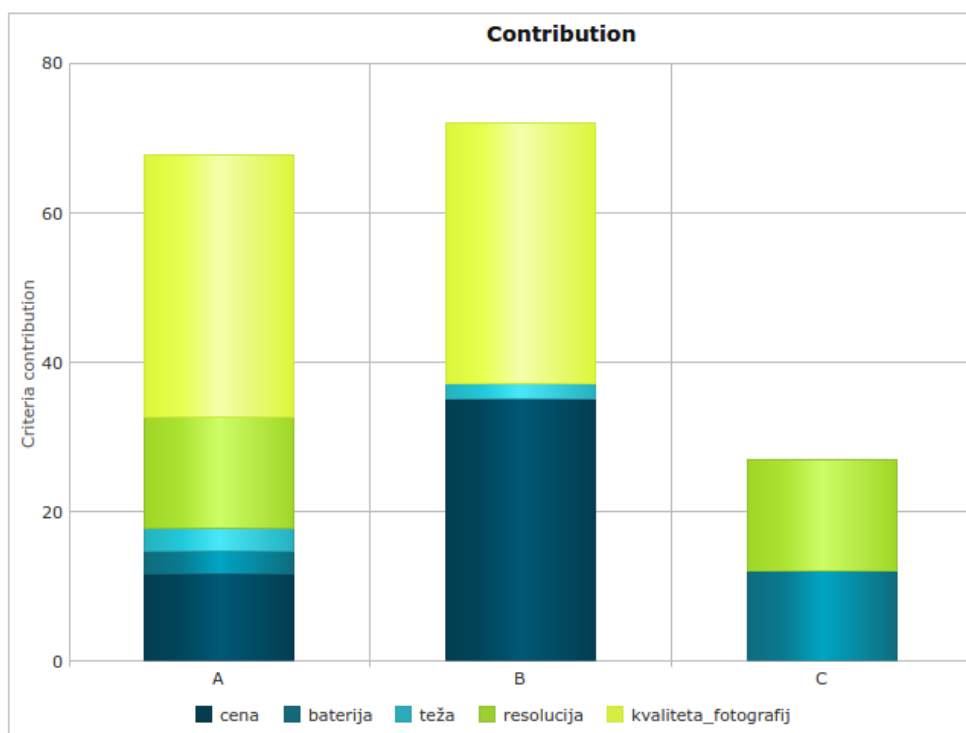
Enačba 4.1 prikazuje izračun deleža D_{ij} za varianto i pri atributu j , kjer je a_{ij} vrednost variante pri atributu in w_j predstavlja normirano utež za ta atribut.

4.6 Občutljivost navzgor

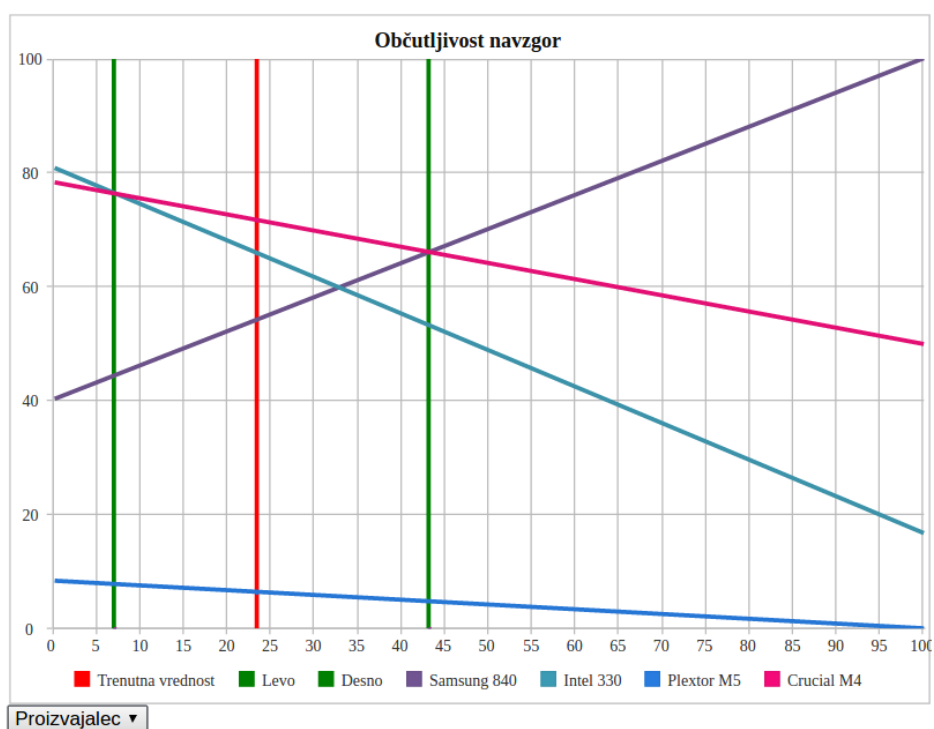
Ta analiza prikazuje obnašanje variant za določen kriterij. Graf za vsako od variant predstavlja koristnost pri določeni uteži. Na sliki 4.6 je rdeča črta postavljena na trenutno vrednost uteži za kriterij *Proizvajalec*. Na točkah presečišča variant in trenutne vrednosti so trenutne koristnosti za vsako varianto. Zeleni črti, poimenovani *Levo* in *Desno*, predstavljata meji, ki sta najbližji koristnosti variant pri trenutni uteži, kjer bi nad trenutno najboljšo varianto, ki je v tem primeru Crucial M4, prevladala druga varianta. Levo in Desno sta poimenovani zato, ker leva meja predstavlja vedno najbližje presečišče levo od trenutne uteži, desna pa desno od le te.

4.7 Občutljivost navzdol ali “Kaj če ...”

Analiza “Kaj če ...” prikazuje samo attribute, torej tiste kriterije, ki se nahajajo v listih odločitvenega drevesa. Pri tej analizi ugotavljamo, kaj se bo zgodilo, če se spremeni katera od normiranih uteži atributa. Poudariti je



Slika 4.5: Analiza prispevka h koristnosti.



Slika 4.6: Analiza občutljivosti navzgor.

potrebno, da pri tej analizi opazujemo le spremembe enega atributa naenkrat in ne kombinacij sprememb uteži atributov. Stolpci v tabeli na sliki 4.7 prikazujejo naslednje:

1. **Kriterij:** Naziv atributa.
2. **Kandidat –:** Varianta, ki bo zmagala, če temu atributu zmanjšamo utež.
3. **Kandidat +:** Varianta, ki bo zmagala, če temu atributu povečamo utež.
4. **Zmanjšanje uteži:** Vrednost, za koliko moramo zmanjšati utež atributu.
5. **Povečanje uteži:** Vrednost, za koliko moramo povečati utež atributu.

Polja v stolpcih “Zmanjšanje uteži” in “Povečanje uteži” so obarvana z rdečo, zeleno ali rumeno. Te barve predstavljajo intervale, na katerih je atribut občutljiv, in sicer, če je vrednost na intervalu $[0,5)$, je polje obarvano rdeče. To pomeni, da je kriterij zelo občutljiv in lahko druga varianta zmagaja že z majhno spremembo uteži. Pri vrednostih na intervalu $[5,15)$ se polje obarva rumeno, kar pomeni, da je atribut zmerno občutljiv na spremembo uteži. Če pa je razlika uteži, pri kateri bi zmagala druga varianta, 15 ali več, se polje obarva zeleno. Pri takšnih razlikah lahko rečemo, da je atribut dokaj neobčutljiv na spremembe uteži, saj bi potrebovali veliko spremembo uteži, kar pomeni tudi nezanemarljiv poseg v odločitveno drevo.

Kriterij	Kandidat -	Zmanjšanje uteži	Povečanje uteži	Kandidat +
Branje			8	Intel 330
Pisanje			6	Intel 330
Kapaciteta			1	Intel 330
Proizvajalec	Intel 330	17	20	Samsung 840
Cena	Intel 330	15		

Go to page: 1 Show rows: 10 1-5 of 5

Slika 4.7: “Kaj če ...” analiza, kjer je v poljih kandidat + ali - predstavljena varianta, ki bi prevladala ob povečanju ali zmanjšanju uteži. Barve polj ponazarjajo občutljivost kriterija na spremembo uteži.

Poglavje 5

Sklepne ugotovitve

Namen projekta, katerega del je to diplomsko delo, je bil izdelati odprtokodno aplikacijo za pomoč pri analizi hierarhičnih odločitvenih modelov. Z izdelavo platforme je bil ta cilj dosežen, saj se le ta lahko uporablja na resničnih problemih in ne samo za izobraževalne namene. To diplomsko delo se osredotoča na analitični del platforme in je hkrati tudi priročnik za razumevanje procesov pri večkriterijskem odločanju in uporabniški priročnik za uporabo aplikacije.

Zastavljeni cilji diplomskega dela, ki so bili doseženi, so:

1. **Zasnova odprtokodne aplikacije:** Uporabljene tehnologije, večinoma programski jezik Javascript, so omogočile kar se da pregledno in neodvisno zasnovo aplikacije. Nodvisnost aplikacije se kaže tudi pri njeni nezahtevnosti glede strojne opreme in aplikacijskih strežnikov. Hkrati je aplikacija zasnovana tako, da jo je mogoče uporabiti na več različnih tipih naprav, kot so mobilni telefoni in tablični računalniki.
2. **Predstavitev procesov pri večkriterijskem odločanju:** Diplomsko delo razčleni proces pomoči pri odločitvah MCDA na več manjših procesov in jih opiše. Predstavi tudi praktične primere uporabe in kaj so njihove ključne lastnosti. Predstavljen je tudi namen uporabe metodologije MCDA in kako poteka pomoč pri odločanju od zasnove do analize.

3. **Uporaba metodologije MCDA:** Uporaba spletne platforme je zasnovana na podlagi smernic večkriterijskega odločanja in vsebuje tri glavne sklope: gradnjo odločitvenega drevesa, vnos surovih vrednosti variant in na koncu predstavitev metod analiziranja našega modela. Vsaka od metod nam predstavi različen zorni kot in nam poda specifično informacijo o variantah in modelu.

4. **Analiza odločitvenega modela:** Eden od namenov gradnje aplikacije in aplikacije kot take je bil izdelati platformo, ki jo lahko uporabimo na dejanskih problemih. Aplikacija, ki nam jo je uspelo izdelati, omogoča nabor metod, ki jih lahko uporabimo na problemu in s tem pomagamo pri odločitvi. Zavihek z analizami nam predstavlja 10 metod, s katerimi si pomagamo pri odločanju: prikaz normaliziranih podatkov z normiranimi utežmi, minimax ureditev, maximax ureditev, leksikografska ureditev, prikaz manjvrednih variant, zvezdni diagram variant, map analiza, pregled prispevka koristnosti atributov k skupni koristnosti variante, analiza občutljivosti navzgor in analiza občutljivosti navzdol.

Ker je izdelana platforma odprtokodna, predstavlja osnovo za možne nadaljnje nadgradnje in modifikacije glede na specifične zahteve. Zasnova analize je modularna, saj je vsaka metoda, ki se na platformi uporablja, zajeta v ločenem okvirju in se vsako možno novo metodo ali vrsto analize razmeroma enostavno vključi v aplikacijo.

Literatura

- [1] Bernard Roy and Daniel Vanderpooten. The european school of mcda: Emergence, basic features and current works. *Journal of Multi-Criteria Decision Analysis*, 5(1):22–38, 1996.
- [2] Jose Figueira. *Multiple criteria decision analysis : state of the art surveys*. Springer, New York, 2005.
- [3] *Multi-criteria analysis : a manual*. Communities and Local Government, Wetherby, 2009.
- [4] Matthias Ehrgott. *Trends in multiple criteria decision analysis*. Springer, New York London, 2010.
- [5] Jonathon P. Schuldt, Sara H. Konrath, and Norbert Schwarz. “global warming” or “climate change”? whether the planet is warming depends on question wording. *Public Opinion Quarterly*, 2011.
- [6] Elizabeth F. Loftus and Guido Zanni. Eyewitness testimony: The influence of the wording of a question. *Bulletin of the Psychonomic Society*, 5(1):86–88, 1975.
- [7] Jacek Malczewski and Claus Rinner. *Introduction to GIS-MCDA*, strani 23–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
- [8] Dimos Loukas. A descriptive-analytic approach to mcda: the case of correspondence analysis. *Operational Research*, 8(2):141–152, 2008.

-
- [9] D. Bouyssou. Building criteria: A prerequisite for MCDA. In C.A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, strani 58–80. Springer-Verlag, Berlin, 1990.
- [10] Thokala P., Devlin N., and Marsh K. et al. multiple criteria decision analysis for health care decision making—an introduction: Report 1 of the ispor mcda emerging good practices task force. strani 1–13, 2016.
- [11] Praveen Thokala and Alejandra Duenas. Multiple criteria decision analysis for health technology assessment. *Value in Health*, 15(8):1172 – 1181, 2012.
- [12] Florian Helff, Le Gruenwald, and Laurent d’Orazio. Weighted sum model for multi-objective query optimization for mobile-cloud database environments. In Themis Palpanas and Kostas Stefanidis, editors, *Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference, EDBT/ICDT Workshops 2016, Bordeaux, France, March 15, 2016.*, volume 1558 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.
- [13] Matej Romih. *Spletna platforma za gradnjo hierarhičnih modelov v problemih odločanja - diplomsko delo*. Univerza v Ljubljani Fakulteta za računalništvo in informatiko, 2017, v pripravi.
- [14] Evangelos Triantaphyllou and Chi-Tun Lin. Development and evaluation of five fuzzy multiattribute decision-making methods. *International Journal of Approximate Reasoning*, 14(4):281 – 310, 1996.
- [15] Catalyze Ltd. Hiview3 - starter guide, 2011.
- [16] Mehdi Ahmadi, Mazdak Arabi, Darrell G. Fontane, and Bernard A. Engel. Application of multicriteria decision analysis with a priori knowledge to identify optimal nonpoint source pollution control plans. *Journal of Water Resources Planning and Management*, 141(2):04014054, 2015.

-
- [17] Gilberto Montibeller and Alberto Franco. *Multi-Criteria Decision Analysis for Strategic Decision Making*, strani 25–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [18] Wikipedia. Normalization (statistics) — wikipedia, the free encyclopedia, 2016. [Online; dostopano 30. junij 2016].
- [19] I. Linkov, A. Varghese, S. Jamil, T. P. Seager, G. Kiker, and T. Bridges. *Multi-Criteria Decision Analysis: A Framework for Structuring Remedial Decisions at Contaminated Sites*, strani 15–54. Springer Netherlands, Dordrecht, 2005.
- [20] Wikipedia. Pareto efficiency — wikipedia, the free encyclopedia, 2016. [Online; dostopano 9. december 2016].