TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Oliver Oidekivi 163583IAPM

XMCDA 2.2.2 STANDARDI PÕHISE AHP RAKENDUSE LOOMINE JA TULEMUSTE TUNDLIKKUSE ANALÜÜS

magistritöö

Juhendaja: Tarmo Veskioja
Doktorikraad

Tallinn 2018

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Oliver Oidekivi

07.05.2018

Annotatsioon

Käesolev magistritöö käsitleb klassikalise Saaty AHP otsustusmudeli tundlikkuse analüüsi käigus lõpptulemuste kindlushinnangute arvutamist Monte Carlo simulatsioonimeetodi abil, kasutades selle jaoks loodud XMCDA veebiteenuste standardi põhist rakendust. Töö käigus uuritakse standardipõhise rakenduse loomise võimalikkust ning kuidas tundlikkuse analüüsi Monte Carlo tulemused erinevad klassikalise AHP mudeli lõppkaalude punkthinnangutest.

Töö esimeses pooles kirjeldatakse üldiseid AHP meetodi ning tundlikkuse analüüsi teostamise teooriaid. Lisaks, käsitletakse Decision Deck initsiatiivi ning nende poolt kirjeldatud XMCDA standardit, mis võimaldab luua AHP struktuuriga teenuseid. Töö teises pooles luuakse realisatsioon, mis teostab AHP mudeli arvutusi ning Monte Carlo meetodi põhist tundlikkuse analüüsi ning hinnatakse muudatuste mõju.

Töö tulemusena on loodud XMCDA standardit järgiv AHP näidismudel ning Java rakendus mis arvutab selle pealt lõpptulemused ning teostab nende peal automaatselt tundlikkuse analüüsi. On kirjeldatud Monte Carlo meetodi põhise tundlikkuse analüüsi tulemuste erinevused klassikalise AHP mudeli tulemustest ning uue lähenemise eelised ja puudused.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 61 leheküljel, 6 peatükki, 26 joonist, 13 tabelit.

Abstract

The Development of an AHP Sensitivity Analysis Application based on the XMCDA 2.2.2 Standard

The master's thesis is about improving the confidence level and interpretation of results obtained from classical Saaty's AHP decision model through Monte Carlo simulation based sensitivity analysis, using an XMCDA web-service standard based application for this. This work examines the possibilities of creating an application for AHP calculations and how automatic sensitivity analysis will affect the model's end results.

The first part of this work describes the theories behind Saaty's AHP model usage and sensitivity analysis reasons. Additionally, an overview of Decision Deck initiative and their XMCDA web-service standard, which allows to create AHP calculation applications, is given. In the second part of the work a software application which performs AHP calculations and automatic sensitivity analysis is created and it's effects are analysed.

As a result of this work an XMCDA standard based AHP model and a Java application which calculates classical AHP model results and performs automatic simulation based sensitivity analysis on those results is created. The differences between the classical AHP model results and the Monte Carlo sensitivity analysis results are analysed and an overview of advantages and disadvantages of the new solution are given.

The thesis is in Estonian and contains 61 pages of text, 6 chapters, 26 figures, 13 tables.

Lühendite ja mõistete sõnastik

MCDM	Multi-criteria decision-making, operatsioonianalüüsi haru, mis tegeleb kriteeriumite hulga hindamisega otsustustegevuses
MCDA	Multi-criteria decision analysis, vaata MCDM
AHP meetod	Analüütiliste hierarhiate meetod (Analytic Hierarchy Process)
AHP mudel	Mudel, mis koosneb algelemendist (Goal), kriteeriumite hierarhiast (Criteria) ja alternatiividest mille vahel otsust tehakse.
ANP	Analüütiline võrkmudeli protsess (Analytic Network Process)
Kriteerium	Tervikprobleemi osa, tegur AHP mudelis
Alternatiiv	Üks valikutest mille vahel otsust tehakse
Kooskõlaindeks	Näitab paaritiste võrdluste maatriksi kooskõla taset
Paaritised võrdlused	(ingl.k *pairwise comparison*), kahe elemendi omavaheline võrdlus mingi teise elemendi suhtes ning selle põhjal hinnangu andmine
Saaty skaala	AHP mudeli juures kasutatav skaala 1/9…1…9
JAMA	Java Matrix Package, lineaaralgebra pakett Java jaoks
Fuzzy AHP	Hägus AHP meetod
Monte Carlo meetod	Matemaatiline algoritm simulatsioonide läbi viimiseks
MC	Monte Carlo, vt. *Monte Carlo* meetod
XMCDA	Andmestandard
Decision Deck	Projekt vabavaraliste MCDA rakenduste ja teenuste arendamiseks
DD	Decision Deck

JAR	Java ARchive, faili formaat java projektide jaoks
Power Iteration	Algoritm maatriksi omaväärtuse ja vektori arvutamiseks
Shapiro-Wilk normaalsustest	Statistiline meetod kontrollimaks kas andmehulk on normaaljaotusega
Element	Mudeli osa, üldine nimetus alternatiivide ja kriteeriumite kohta
Tundlikkuse analüüs	(ingl.k *sensitivity analysis*), AHP meetodi puhul näitab kuidas ühe või mitme kriteeriumi osakaalu muutmine mõjutab meetodi lõpptulemust
Tundlikkuse ristanalüüs	(ingl. k *Tradeoff analysis*), üks tundlikkuse analüüsi võimalustest
Tundlikkusekordaja	Kasutatakse tundlikkuse analüüsis, näitab kuidas peab analüüsitavate elementide vahelisi kaale muutma, et lõpptulemus muutuks.
Paremuse kindlushinnang	Statistiline hinnang ühe alternatiivi paremuse kohta teise suhtes
Veakordaja	Kasutatakse paaritise maatriksi kõige ebakindlamate võrdluste leidmiseks.

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Magistritöö on kirjutatud teemal „XMCDA 2.2.2 standardi põhise AHP rakenduse loomine ja tulemuste tundlikkuse analüüs“. Teema valiti kuna autor soovib uurida kas tavapärase AHP mudeli poolt tagastatud tulemustele oleks võimalik teostada automaatset tulemuste tundlikkuse analüüsi ilma, et lõppkasutajal oleks vaja sellest detailseid teadmisi või spetsiaalset tarkvara ehk muuta AHP arvutuse käigus tagastatud tulemuste saamist lihtsamaks ja nende tõlgendamist kindlamaks.

Antud peatükis tutvustatakse magistritöö tausta ning ülesande püstitust. Selgitatakse miks antud teema on oluline ning kirjeldatakse millise metoodikaga jõutakse lõpptulemuseni. Lõpuks tehakse lühiülevaade töö dokumendi struktuurist

1.1 Taust ja probleem

Klassikalise AHP mudeli lõpptulemuste pealt on raske teha järeldusi kas üks alternatiiv on teisest alternatiivist selgelt parem. Klassikalise AHP mudeli lõpptulemuseks on alternatiivide kaalude vektor, kus igat alternatiivi iseloomustab üks lõppkaal. Analoogselt üksiku punkthinnangu varieeruvuse tõlgendamisraskustega ei näita ka klassikalise AHP mudeli lõppkaalud varieeruvust. Lõpptulemuste varieeruvus on esitatav hägusa AHP mudeli abil, aga hägusat AHP mudelit enamus otsustajatest (erinevatel põhjustel) tavaliselt ei kasuta. Osaliseks lahenduseks on klassikalise AHP mudeli tundlikkuse analüüs, aga põhjalikku tundlikkuse analüüsi on ilma tarkvaralise toeta tavalisel otsustajal raske läbi viia.

1.2 Ülesandepüstitus

Magistritöö peamiseks eesmärgiks on uurida AHP mudelite ning nende tulemustele teostatava tundlikkuse analüüsi üldiseid tööpõhimõtteid ning sellele tuginedes konkreetselt XMCDA standardi poolt pakutatavate andmestruktuuride kasutamise võimalust uue teenuse loomisel mis võimaldaks kasutajal automaatselt sellist analüüsi teostada. Selline rakendus vähendaks Monte Carlo analüüsi põhjal klassikalise AHP

meetodi tulemuste ebakindlust ning lihtsustaks tundlikkuse analüüsi tegemist. Eesmärgiks oleks klassikalise punkthinnangutega vektori asemel tagastada lõpptulemustena statistiline kindlushinnang iga alternatiivi kohta. Samuti oleks see kasutatav nende poolt, kellel ei ole teadmisi Monte Carlo meetodist või AHP tundlikkuse analüüsist. Töös antakse teoreetiline ülevaade kuidas teostada tundlikkuse analüüsi Monte Carlo meetodi abil. Lõpuks viiakse läbi näidisarvutus magistritöö jaoks loodud AHP mudeli ning loodud rakenduse peal ning võrreldakse tulemusi teiste meetodide abil saadud tulemustega.

Juhul kui see on võimalik ja toetatud, siis töö käigus üritatakse vähemalt alustada uurimist selle kohta kuidas oleks võimalik ka ANP puhul XMCDA standardit kasutada ja MC arvutusi teha.

Töö eesmärgid:

1. Uurida XMCDA standardi sobivust AHP mudeli arvutuste tegemise jaoks. Sealhulgas uurida milliseid AHP mudelites vajalikke andmestruktuure see toetab.

2. Defineerida vajadusel AHP jaoks vajalikud puuduvad andmestruktuuri tüübid. Lõputöös keskendutakse AHP mudelitele, samas võimalusel toetades ka ANP mudelite funktsionaalsust.

3. Genereerida etteantud Saaty AHP mudeli andmete põhjal selle mudeli kohta Monte Carlo tundlikkuse analüüsi statistikat.

4. Arvutada Monte Carlo statistika pealt tõenäosused, et üks alternatiiv on teisest alternatiivist parem.

1.3 Metoodika

Decision Deck initsiatiivi poolt loodud XMCDA standard pakub võimaluse luua MCDA-põhiseid arvutusi tegevaid veebiteenuseid, mis järgiks ühtset ülesehitust ja defineeritud sisendeid ja väljundeid. Antud töös keskendutakse Saaty AHP mudeli arvutustele. Et loodud sisendmudelist jõuda tagastatud lõpptulemusteni on vaja luua uus rakendus mis kasutaks XMCDA standardit läbi üldise skeemi järgimise sisendite ja väljundite defineerimisel kui ka olemasoleva viite projekti laiendamise magistritöö rakenduse loomisel.

Eesmärkide saavutamiseks luuakse standardit järgiv näidis AHP mudel, see antakse sisendiks töö käigus loodud Java-põhisele rakendusele ning tagastatakse lõpptulemus kas XML või JSON formaadis. Töö lõpuks hinnatakse ja analüüsitakse saadud tulemusi ning nende sobivust ülesandepüstituse lahenduseks.

1.4 Ülevaade tööst

Antud töö koosneb 7-st peatükist. Esimene peatükk on sissejuhatus, mis annab ülevaate magistritöö probleemist ja eesmärkidest ning kasutatud metoodikast.

Teises peatükis kirjeldatakse töö teoreetilisi aluseid ja hüpoteese. Tehakse ülevaade MCDA teadusharust, selle alla kuuluvast AHP mudelist, mis on antud töö keskseks osaks, ning tundlikkuse analüüsist mida antud mudeli lõpptulemuste peal sooritatakse. Lisaks kirjeldatakse Decision Deck initsiatiivi ning nende loodud XMCDA standardit. Viimaks, esitatakse ülevaade varasemalt samas valdkonnas tehtud töödest.

Kolmandas peatükis kirjeldatakse realisatsiooni jaoks kasutatud tehnoloogiaid. Kirjeldatakse põhjalikult XMCDA standardit ning olulisemaid andmestruktuure mida see toetab, samuti tehakse ülevaade teistest pakettidest mida kasutatakse arvutuste tegemiseks. Lisaks, tehakse ülevaade töö tulemuste valideerimiseks loodud näidismudeli loomisest.

Neljandas peatükis kirjeldatakse konkreetse Java põhise rakenduse loomist mis magistritöö eesmärke täidaks ning vaadeldakse selle toimimise loogikat. Samuti esitatakse ülevaade rakenduse poolt tagastatud tulemustest, nende võrdlusest ning tundlikkuse analüüsi sooritamisest. Peale selle uuritakse alternatiivseid lähenemisi sarnase probleemi lahendamiseks ja realisatsiooni teostamiseks.

Viiendas peatükis esitatakse järeldused realisatsiooni toimimise ja tulemuste põhjal.

Kuuendas peatükis esitatakse töö käigus selgunud edasiarendamise võimalused.

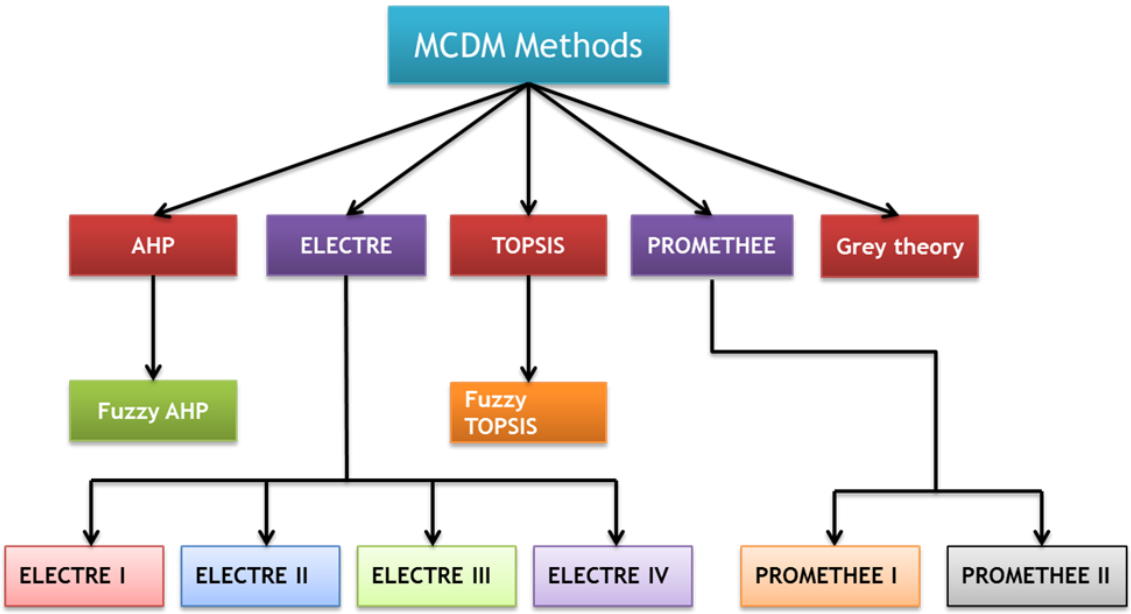
Seitsmendas peatükis tehakse kokkuvõte kogu tööst.

2 Töö teoreetilised alused ja hüpoteesid

Järgnevas peatükis antakse ülevaade peamistest teoreetilistest alustest mida antud magistritöös kasutatakse. Kirjeldatakse MCDA distsipliini ja selle alla kuuluvat AHP mudelit, selle positiivseid ja negatiivseid külgi ning alternatiivseid võimalusi. Antakse ülevaade tundlikkuse analüüsist ning Decision Deck initsiatiivist mis võimaldab MCDA põhiseid veebiteenuseid luua. Viimaks uuritakse varasemalt seotud teemadel kirjutatud töid.

2.1 MCDA

MCDA (Multi-criteria decision analysis) või MCDM (Multi-criteria decision making) on teadusharu, mis tegeleb otsustuste vastuvõtmise ja sellega seotud algoritmide uurimisega olukorras kus valiku tegemisel tuleb arvestada paljude erinevate kriteeriumitega. Töös kasutatav AHP mudel on üks näide selle distsipliini alla kuuluva meetodi kohta, kuid peale selle on veel väga palju teisi valikuid.



Joonis 1. Ülevaade enamlevinud MCDM algoritmidest

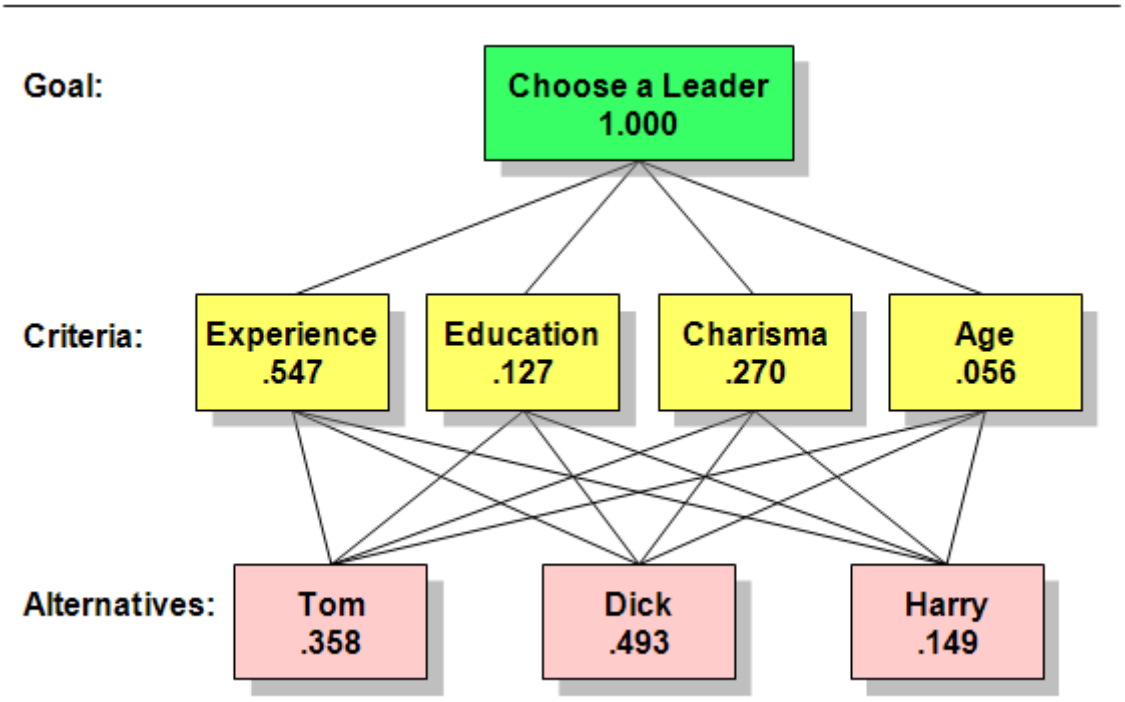
Allikas: Aruldoss, Martin, T. Miranda Lakshmi, and V. Prasanna Venkatesan. "A Survey on Multi Criteria Decision Making Methods and Its Applications."

Peamisteks huvigruppideks kellele MCDA-põhised meetodid olid algselt suunatud peetakse ettevõtteid, kellel on vaja otsustuste tegemisel abivahendeid. Nagu nimest välja

tuleb, siis MCDA algoritmid rõhuvad kriteeriumitele ja enamasti kogu otsustusprotsess toimub jagades lahendatava probleemi kriteeriumiteks ja alternatiivideks ning läbi kriteeriumitele olulisust määrava hinnangu (kaalu) andmise üritatakse jõuda sobivaima lahenduseni. Tihti lahendatavatel probleemidel ei ole ühte unikaalset parimat lahendust ning lõpplahendust milleni tahetakse jõuda võib defineerida erineval viisil. Levinumad eesmärgid on - parima valiku leidmine ette määratud hulgast, teatud hulga parimate valikute leidmine, alternatiivide grupeerimine kogumitesse jt. Joonisel 1. on toodud välja ülevaade enimlevinud MCDA meetoditest [1].

2.2 AHP meetod

AHP ehk hierarhilise analüüsi meetod (Analytic Hierarchy Process) on Thomas L. Saaty poolt 70.ndatel loodud meetod mille peamiseks eesmärgiks on leida subjektiivsete hinnangute alusel võimalikult objektiivne lahend mingile probleemile. AHP meetod kuulub MCDA-põhiste algoritmide gruppi.



Joonis 2. Ülevaade AHP struktuurist

Allikas: „Analytic hierarchy process – leader example“ [2]

AHP mudeli eesmärgiks on jagada üks suur probleem erinevateks kriteeriumiteks ehk alamosadeks, mille põhjal otsust teha, ja alternatiividekks, mille vahel lõppvalikut teha.

Põhimõtteks on see, et ühe suure ja keerulise probleemi korral on raske parimat lahendust leida või valida alternatiivide vahel, kuid kui probleem on jagatud osadeks, siis nende osade hindamine ja omavaheline võrdlus aitab kergemini lõpptulemuseni jõuda. Kogu protsess on küllaltki subjektiivne ja hinnangute andmine võib põhineda nii faktidel kui ka hindaja isiklikul arvamusel. Mudeli loomine toimub iteratiivselt ning uue informatsiooni avaldumisel võib mudelit muuta ja täiustada.

AHP meetodina on puustruktuuriga (vt. joonis 2), mis algab üksikust algelemendist mille jaoks otsust leitakse (näiteks: parima CMS tarkvara valik, parima liidri valik jms.) ning selle all on kõigepealt erinevad kriteeriumid (kihiti) ning pärast seda kõige viimasel kihil alternatiivid mille vahel valikut tehakse. Igal kriteeriumil võivad omakorda olla alamkriteeriumid ehk mudel võib olla rohkem kui kolme tasandiline. Igal erineval kihil või erineva elemendi all olevaid kriteeriumeid võrreldakse omavahel ja neile antakse tähtsuse hinnang üksteise suhtes. Sama tehakse ka alternatiividega neile eelnevate (ühenduses olevate) kriteeriumite all. Selline paarikaupa võrdlemine on subjektiivne ning võib tugineda nii konkreetsetele hinnangutele või parameetritele (arvutil 8GB vs 16GB mälu), kui ka autori isiklikule tunde järgi hinnangule (auto turvalisus on olulisem kui kiirus) [3].

Criteria	Experience	Education	Charisma	Age	Priority
Experience	1	4	3	7	0.547
Education	1/4	1	1/3	3	0.127
Charisma	1/3	3	1	5	0.270
Age	1/7	1/3	1/5	1	0.056
				Sum of Priorities	1.000
				Inconsistency	0.044

Joonis 3. AHP põhikriteeriumite võrdlusmaatriks koos tähtsushinnangute ja kooskõlaindeksiga

Allikas: „Analytic hierarchy process – leader example“ [2]

Joonisel 3 on näidatud kuidas Joonis 2 mudeli puhul näeb välja eesmärgi (ingl.k. *Goal*) elemendi all olevate kriteeriumite omavaheline võrdlemine. Mudeli definitsiooni poolt on ettekirjutatud see, et hinnanguid antakse skaalal 1-9 (kus 1 on võrdne ja 9 on märkimisväärselt parem teisest) ning kui ühele kriteeriumile antakse hinnang teise suhtes, siis see teine saab automaatselt hinnanguks arvu pöördväärtuse (joonis 3 põhjal, kui *Experience* hinnang *Age* suhtes on 7, siis *Age* hinnang *Experience* suhtes on 1/7). Seda skaalat kutsutakse Saaty skaalaks (vt. Lisa 2). Kui võrreldavaid kriteeriume või alternatiive on rohkem kui 2, siis arvutatakse välja kooskõlaindeks (mõnikord kutsutakse ebakooskõlaks), mis peab jääma teatud piiridesse, et tulemus oleks aktsepteeritav (näiteks kui A > B ja B > C, siis C > A puhul ei ole mudel kooskõlas, kooskõlaindeks ei ole aktsepteeritav ning mudel ei sobi). Aktsepteeritavaks piiriks peetakse tavaliselt vähem kui 0.1 väärtuseid. Peale kooskõlaindeksi arvutatakse iga maatriksi all välja võrreldavate elementide lokaalsed skoorid ning läbi nende hiljem ka globaalsed skoorid (lokaalne tähendab, et alternatiivi hinnatakse isolatsioonis kogu mudelist ja valikud peavad kokku andma 1 (näiteks LHV 0.35, Swedbank 0.33, SEB 0.32) ja globaalse skoori puhul korrutatakse need arvud läbi antud kriteeriumi osatähtsusega kogu mudelis (näiteks: kui investeerimistasude osatähtsus oleks 0.27, siis need samad pankade alternatiivid saavad hinnanguteks vastavalt 0,0945, 0,0891 ja 0,0864)) [3].

Education	Tom	Dick	Harry	Priority
Tom	1	3	1/5	0.188
Dick	1/3	1	1/7	0.081
Harry	5	7	1	0.731
			Sum of Priorities	1.000
			Inconsistency	0.062

Joonis 4. Alternatiivide võrdlusmaatriks koos tähtsushinnangute ja kooskõlaindeksiga

Allikas: „Analytic hierarchy process – leader example“ [2]

Kui iga kriteeriumi all on hinnangud olemas, siis nende tulemuste pealt arvutatakse punkthinnangutega vektor, kus igal alternatiivil on kindel lõpphinnang (vt. joonis 5, lõpphinnangute vektor on *Goal* tulbas)

	Priority with Respect to				
Candidate	Experience	Education	Charisma	Age	Goal
Tom	0.119	0.024	0.201	0.015	0.358
Dick	0.392	0.010	0.052	0.038	0.492
Harry	0.036	0.093	0.017	0.004	0.149
Totals:	0.547	0.127	0.270	0.056	1.000

Joonis 5. AHP mudeli lõpptulemused

Allikas: „Analytic hierarchy process – leader example“ [2]

AHP on mõeldud üldotstarbelise meetodina ja selle puhul ei ole otsest piirangut kindla ala suhtes ja see on väga laialdaselt kasutatav kuna probleeme on võimalik lihtsamateks alamprobleemideks ja alternatiivide valikuks muuta olenemata valdkonnast [3].

2.3 Tundlikkuse analüüs

Tundlikkuse analüüs uurib kuidas mudeli väljundi tulemused on mõjutatud selle sama mudeli sisendite ebakindluse poolt ehk tundlikkuse analüüsi eesmärgiks on ühe mudeli peal rakendada erinevaid eelduseid, muutes eelduste muudatuste põhjal mudeli parameetreid ning seeläbi vaadata kuidas tagastatud tulemused muutuvad ja uurida milline mõju on erinevatel muutujatel lõpptulemustele [4]. Teisisõnu uurib tundlikkuse analüüs kuidas mudelis või selle sisendites tehtavad muutused või selle vead mõjutavad mudeli lõpptulemusi ning nende pealt tehtavaid järeldusi. Tundlikkuse analüüsi kohta öeldakse ka, et see on ’mis-siis-kui’ (ingl.k *what-if*) tüüpi analüüs [5].

Tundlikkuse analüüsi teostamisel võib olla erinevaid eesmärke, mõned neist [5]:

- Testida optimaalse lõpptulemuse stabiilsust
- Uurida millistes tingimustes optimaalne lõpplahend muutuks (millised on kõige olulisemad muutujad, kriitilised väärtused, millised lahendid ei ole sobilikud jne.)
- Saada parem ülevaade kuidas mudeli väljundid on sisendite poolt mõjutatud
- Täiustada mudelit (täpsuse testimine, kalibreerimine, lihtsustamine jne.)

Eelneva eesmärgiks on tagada, et mudel oleks võimalikult stabiilne ja tulemused oleksid piisavalt kindlad, et selle pealt saaks anda soovitusi või võtta vastu parimaid otsuseid.

Kuna AHP puhul antakse tihti subjektiivseid hinnanguid paarikaupa võrdlustena, siis võib seda mudelit üleüldiselt pidada ligikaudseks ning seetõttu kasutatakse tulemuste kontrollimiseks ja täiustamiseks tundlikkuse analüüsi tuge. AHP mudeli puhul näitab tundlikkuse analüüs kas mudel antud kujul on sobilik või on vajalik teha uus iteratsioon. Otsustusmudeli koostamise eesmärgi korral on põhifookuseks mudeli testimine ja häälestamine. Olemasolevate objektide peal kasutamise korral on AHP tundlikkuse analüüsi juures eesmärgiks leida kõige tõenäolisem hinnangu muudatuste kombinatsioon, mis lõpptulemust muudab ja hinnata kui tõenäoline on selline muudatus [5] [6]. AHP puhul kasutatakse tihti tundlikkuse analüüsi sooritamiseks visuaalset abivahendit (mudeli üksikust elemendist sõltuvat tundlikkuse graafikut), mis näitab kui palju on mudeli ühe vaheelemendi (kriteeriumi) kaalu vaja muuta, et alternatiivide lõppjärjestus muutuks, juhul kui see on võimalik.

Tundlikkuse analüüsi saab teostada erinevatel viisidel. AHP meetodi puhul on olemas kolm peamist lähenemist mida kasutatakse – tehniliste vigade analüüs, hinnangute kooskõla põhine analüüs ning tundlikkuse ristanalüüs ehk sisuliste proportsionaalsuste analüüs. Esimesi kahte neist on võimalik automatiseerida, kuid viimast peab hindaja käsitsi läbi viima.

Tundlikkuse ristanalüüs (ingl.k *tradeoff analysis*) [7] keskendub kahele lõppelemendile ja kahele vaheelemendile ning üritab hinnata kas selliste 2x2 paaride proportsionaalsused on paigas ehk kas sisulised proportsionaalsused on kirjeldatud ligikaudselt õigesti. Lihtsamalt tähendab see, et võrreldakse näiteks kahe alternatiivi lõpptulemuste pealt kahte kriteeriumit ning hinnatakse kas nende suhe omavaheline suhe on paigas [8]. Antud magistritöös kirjeldatud meetod on erinev ning keskendub hinnangute kooskõlale.

Ühe tundlikkuse analüüsi meetodi läbi tegemine AHP meetodi ja selle tulemuste peal ei kaota ära vajadust teise kahe järgi ehk pärast antud töös kasutatavat Monte Carlo simulatsiooni põhist tundlikkuse analüüsi oleks lõplike tulemuste saamiseks vajalik läbi viia ka teised kaks meetodit. Ei ole kindlaid nõudeid millises järjestuses tundlikkuse analüüsi peaks sooritama, seega on võimalik automatiseerida esimesed kaks lähenemist ning viia kolmas pärast saadud tulemusi läbi, kuid on võimalik neid teha ka teises järjestuses.

2.3.1 Monte Carlo arvutus

Antud magistritöö eesmärgiks on tundlikkuse analüüsi teha automaatselt üle kogu mudeli (mudeli üksiku vaheelemendi visuaalse abivahendita), et kasutaja ei vajaks põhjalikke teadmisi tundlikkuse analüüsist ning, et see kajastuks kindlushinnangutena mudeli poolt tagastatud lõpptulemustes. Selle jaoks kasutatakse töös *Monte Carlo* simulatsiooni abi

Monte Carlo ehk juhusliku valimi mitmekordne simulatsioon on matemaatiline meetod, mille eesmärgiks on modelleerida erinevate tulemusteni jõudmise tõenäosusi olukorras kus sellist hinnangut on raske välja arvutada juhuslike muutujate olemasolu tõttu. *Monte Carlo* meetodi mudelid luuakse asendades simuleeritava meetodi ebakindlad väärtused tõenäosusjaotusest (nt. normaaljaotus) genereeritud väärtustega ning arvutades nende uute väärtuste põhjal iteratsiooni lõpptulemused. Selle protsessiga tehakse väga suur arv kordusi (nt. 1000-10000) ning igal korral salvestatakse simulatsiooni lõpptulemused ning selle põhjal tekib tõenäosusjaotus võimalikest tulemustest, mille pealt saab arvutada mingi kindla hinnanguni jõudmise tõenäosust. *Monte Carlo* simulatsioon ütleb nii seda mis võib juhtuda, kui ka seda kui tõenäoline sellise tulemuseni jõudmine on [9].

Monte Carlo simulatsiooni on töös kasutatud selle jaoks, et sooritada AHP meetodi tulemuste peal automaatset tundlikkuse analüüsi ilma graafilise liidese ning kasutajapoolsete arvutusteta, genereerides paarisvõrdluste maatriksite jaoks uued normaaljaotuse põhised juhuarvud ning arvutades nende abil uued iteratsiooni lõppkaalud.

Peatükis *2.5 Seotud tööd* on mainitud kuidas Monte Carlo simulatsiooni on varasemates töödes AHP mudeli peal rakendatud ning mis olid selle meetodi kasutamise põhjusteks ja milliste järeldusteni jõuti.

2.4 Decision Deck initsiatiiv

Decision Deck projekt on initsiatiiv, mille eesmärgiks on arendada vabavaralist tarkvara (veebiteenuseid), mis toetaksid MCDM algoritmide kasutamist tarkvaraliselt, et lihtsustada nende kasutatavust ning muuta sellised algoritmid laialdasemalt kättesaadavaks. Decision Deck eesmärgiks on muuta kõigi nende poolt loodud teenuste kasutamist võimalikult lihtsaks kuna nende kasutajad ei pruugi olla tarkvaraarendajad ega omada antud valdkonnast teadmisi (võivad olla teiste valdkondade teadlased, inimesed

aladelt kus neid meetodeid saaks kasutada, kuid mis ei nõua tarkvara arendust jne). Et soodustada veebiteenuste loomist on DD loonud XMCDA XML standardi, mida kõik teenused/meetodid peavad järgima. 2.0 versioon standardist lasti välja 2009 aastal, praegusel hetkel on see 3.0.2 juures mis avalikustati 2017 aastal ehk standardit uuendatakse siiamaani aktiivselt. Decision Deck toetab vabavaraliste teenuste loomist kõigi tarkvaraarendajate poolt juhul kui loodud programm täidab esitatud nõuded [10] ning läbib kontrolli nõuetele.

2.5 Seotud tööd

Antud magistritöö eesmärgiks on kindlat standardit jälgiva süsteemi loomine, mis võtaks sisendiks AHP mudeli, arvutaks selle lõpptulemused ning teostaks nende peal tundlikkuse analüüsi Monte Carlo arvutuse meetodil.

Nii klassikalise AHP mudeli kasutamise kohta üldiselt kui ka selle eri versioonide, tundlikkuse analüüsi teostamise jms. on varasemalt avalikustatud mitmeid erinevaid töid. See peatükk annab ülevaate eelnevatest töödest kus on kasutatud AHP mudelit koos Monte Carlo meetodil tundlikkuse analüüsi arvutustega või kus on võrreldud tavalist AHP kasutamist Monte Carlo arvutusega AHP (MCAHP) kasutamisega. Samuti uuritakse töid kus on kasutatud mõnda muud tundlikkuse analüüsi lähenemist koos AHP mudeliga. Monte Carlo meetod on üks võimalus kuidas teostada lihtsasti tundlikkuse analüüsi mudeli lõpptulemuste peal, mis teoreetiliselt annab suurema kindlusastmega tulemuse kui tavapärane lähenemine. Teiseks kuna töö fookus on luua süsteem, mis järgiks kindlat standardit, siis antakse ülevaade ka töö jaoks valitud XMCDA standardit käsitletavatest töödest. Viimaseks uuritakse töid kus on võrreldud tavalist AHP-d hägusa (fuzzy) AHP-ga kuna seda võib üheks alternatiivseks meetodiks Monte Carlo tundlikkuse analüüsi asemel pidada, kuid mis edaspidi ei ole selle magistritöö fookuses.

TTÜ-s on antud magistritöö kirjutamise hetkel raamatukogu digikogu otsingu järgi kirjutatud neli tööd, mis sisaldavad töö pealkirjas märksõna AHP, millest ükski ei uuri AHP tulemuste täpsust või sellele tehtavat tundlikkuse analüüsi. Siiski, antud teemaga seonduva õppeaine materjalides oli saadaval üks vanem magistritöö [11], mis uurib sarnast valdkonda ning millest tehakse järgnevas alapeatükis samuti ülevaade.

2.5.1 Ülevaade

[12] Artiklis oli autorite eesmärk võrrelda Monte Carlo põhist AHP meetodit (MCAHP) klassikalise AHP meetodi tulemustega ning selgitada kas ja millisel hetkel tekib nende vahel erinevus ja kas MCAHP meetodil on eeliseid tavapärase mudeli ees, et oleks põhjust seda kasutada. Töös loodi algoritm, mis simuleeriks hindamise olukordi nii, et oleks võimalik võrrelda neid kahte meetodit erineval tasemel ebakindluste puhul. Autorite hinnangul oli see kirjutamise hetkel esimene töö, mis võrdles nende kahe meetodi lõpptulemusi erineva astme ebakindluse all. Tulemusi valideeriti võrreldes mudelite väljundeid varasemalt olemasolevate kindlalt usaldusväärsete reitingutega (antud juhul kasutati interneti blogide kohta käivaid reitingud kuna nende kohta olid väidetavalt usaldusväärsed tulemused teistest allikatest varasemalt olemas). Tulemustest selgus, et väiksema ebakindluse astme korral ei ole MCAHP-l ja klassikalisel AHP-l statistiliselt olulist erinevust, kuid kui ebakindlus suureneb, siis MCAHP annab täpsema tulemuse. Lihtsamalt tähendab see seda, et kui hindaja(d) annavad täpseid ja järjepidevaid hinnanguid, siis ei ole suurt erinevust kumba meetodit kasutada, kuid kuna kogu AHP mudeli loomise protsess on väga subjektiivne, siis võib tihti tekkida olukord, kus hinnangud ei ole täpsed ega järjepidavad ning MCAHP meetodi kasutamisel on põhjus olemas.

Teise punktina mainiti samas töös [12] ka AHP mudeli statistilise tõlgendamise puudumist ehk kui kaks lõpptulemust on väga lähedal üksteisele, siis see ei väljendu lõpptulemustes mida on võimalik MCAHP kasutamisega lahendada ja mis on ka antud magistritöö üks eesmärkidest, lisada tulemustele juurde statistiline hinnang kui palju üks alternatiiv on teisest parem.

Töös [12] on viidatud vähemalt 5-le teisele artiklile, kus kasutati MCAHP meetodit, mis ei ole vabalt kättesaadavad, kuid teadusartiklis [12] on välja toodud lõppjäreldused nende kohta, milles korratakse väidet, et juhul kui ebakindlus eksisteerib, siis on soovitatav kasutada MCAHP meetodit klassikalise mudeli ees.

[13] Artiklis uuriti MCAHP meetodi rakendamist reaalselt eksisteerivas olukorras, et parandada ettevõtte külastajate rahulolu. Antud artikkel ei võrrelnud AHP tulemusi MCAHP meetodi tulemustega, kuid selles võrreldi tulemusi varasemalt tehtud uuringutega, mis olid kasutanud lineaarregressiooni ja statistilist analüüsi, et lõpptulemusteni ja järeldusteni jõuda. On mainutud, et MCAHP meetodi kasuks otsustati

kuna erinevate artiklite põhjal see vähendab ebatäpsuste riski võrreldes tavapärase AHP meetodiga. Sarnaselt eelmisele [12] artiklile mainitakse klassikalise AHP mudeli tõenäosuste puudumist, viidates ühele varasemale 2009 aastal kirjutatud artiklile.

Artiklis [13] koostati uuring hindajate seas, kelleks olid ettevõtte külastajad, ning nende vastuste pealt arvutati välja tulemused. Artikli [13] tulemuste järeldustest selgus, et MCAHP kasutamine võrreldes varasemate uuringutega andis täielikult erinevad lõpphinnangud (ükski hinnang varasematega ei kattunud), mis tõi nende jaoks välja uue perspektiivi ja võimaluse selgitada välja uusi alternatiive otsuste tegemisel ning hinnata ümber seniseid järeldusi ja eelduseid klientide rahulolu kohta.

[14] artikkel uuris MCAHP meetodi kasutamist ühe konkreetse Hiina riiki puudutava juhtumiuurimise peal. Antud magistritöö kontekstis ei oma see mille kohta [13] või [14] artiklitest juhtumiuurimiste puhul võrdlusi tehti otsest tähtsust kuna oluline on ainult AHP kasutamine ning tulemuste täpsus ja nende erinevus MCAHP tulemustest, mistõttu oli selle töö kõige olulisem osa see, et oli võrreldud Monte Carlo lähenemist ka klassikalise AHP-ga ning leiti, et esimene meetod vähendas ebakindlust võrreldes viimasega ning seeläbi parandas otsustajate üksmeelt lõpptulemustes. On oluline välja tuua, et [14] artiklis küll kasutati veel lisaks muule algoritmi, mida nad kutsusid PERT-ks, mis võis mõjutada MCAHP lõpptulemusi ja anda sellele eelise, kuid mida antud lõputöö kontekstis kasutama ei hakata.

Nii [12], [13] kui ka [14] on välja toodud teineteisest erinevad moodused kuidas Monte Carlo tundlikkuse analüüsi arvutusi AHP mudeli peal teostati. [14] töö kasutas lisaks veel arvutust, mida nad nimetasid PERT algoritmiks, kuid mida antud magistritöös ei kasutata. [13] on Monte Carlo arvutused tehtud üle erinevate otsustajate, aga antud töös on ette nähtud, et seda tehakse esialgu üksiku otsustaja tasemel. Hilisemalt ei ole välistatud grupiotsustuse lisamine, kuid Monte Carlo tundlikkuse analüüs jääb alati üksiku otsustaja tasemele.

Punktis 2.5 on mainitud, et ka TTÜ-s on kirjutatud üks antud tööga seotud teemat käsitlev magistritöö [11]. Mainitud töö uuris kuidas parandada või lihtsustada AHP meetodile ja selle mudelitele tehtavat tundlikkuse analüüsi. Töös on välja toodud kaks peamist probleemi mis tundlikkuse analüüsi korral esile võivad kerkida – alternatiivide eemaldamisel mudelist võisid allesjäänud alternatiivid kohad vahetada ilma mudelis

muudatusi tegemata ning ülevaate puudumine kriteeriumite tasemel kooskõladest. Töös keskenduti peamiselt kohavahetuse probleemi lahendamisele täiustades elementide arvutuskäiku ning luues vaated, mis otsustajad abistaksid hindamisel. Võrreldes selle tööga ei sisaldanud mainitud töö Monte Carlo simulatsiooni ning keskendus pigem tehniliste vigade analüüsile samas kui see töö uurib pigem hinnangute kooskõlapõhiseid vigasid ning sellega seotud tundlikkuse analüüsi (vt. peatükk 2.3). Arvestades eelnevalt mainitud kolme erineva lähenemise vajalikkust lõpliku tundlikkuse analüüsi jaoks (peatükk 2.3) üritatakse nende kahe tööga ära lahendada kaks tundlikkuse analüüsi variantidest.

[15] artikkel on seotud konkreetse MCDM standardiga mida antud töös kasutatakse – XMCDA standard - ning see on kirjutatud kahe autori poolt, kes ka lõid selle standardi, ning töös nad toovad välja motivatsiooni selle eksisteerimiseks ja kasutamiseks. Suur osa tööst kirjeldab baastüüpe mida XMCDA kasutab/võimaldab kasutada. Viidatud töö ei ole seotud AHP ega MCAHP meetoditega vaid annab ainult tehnilise ülevaate skeemist mille abil neid kirjeldada on võimalik, mis on antud magistritöö üks eesmärkidest. Töös on öeldud, et standard on aktiivses arendamisjärgus ning eesmärgiks on võimaldada enamikku MCDM põhiseid algoritme kasutada läbi ühtse allika veebiteenustena, ilma, et kasutaja oleks sunnitud erinevaid tarkvaralahendusi enda arvutisse laadima.

[16] on Inglismaa ülikoolis kirjutatud magistritöö milles kirjeldatakse ühe olemasoleva tähtsuse järjekorra hindamise tööriista (PriEsT) täiendamist grupiotsustuse võimaluse lisamisega. Käesoleva magistritööga on see seotud nii sellepoolest, et PriEsT otsustuste aluseks on AHP meetod, kui ka sellepoolest, et töös üritatakse samuti rakendada XMCDA standardit täienduste tegemisel, et pakkuda edasiarendatud lahendust veebiteenustena. Peale selle väidetakse [16] töös, et AHP mudeli puhul on oluline sensitiivsusanalüüsi teostamine, kuid töö alapeatükis milles kirjeldatakse loodud lahenduse piiranguid mainitakse, et autori poolt kirjutatud töö raames tundlikkuse analüüsi teostamist ei hakata toetama kuna olemasolev sensitiivsusanalüüsi meetod ei olnud sobilik XMCDA formaati salvestamiseks. Samuti väidetakse, et sensitiivsusanalüüsi lisamine AHP-le võib olla oluline uurimisobjekt tulevikuks. Praegune (kirjutatav) töö küll ei kasuta ega toetu PriEsT lahendusele ühelgi moel nagu see töö millele on viidatud, kuid ka see töö kirjeldab sarnaseid uurimissuundi ning teoreetiliselt võib keskenduda ühele eelneva töö puudusele/ edasiarenduse ettepanekule.

Alternatiivsete lähenemistena Monte Carlo meetodi kasutamisele on pakutud ka ühe võimalusena hägusa (fuzzy) AHP meetodi kasutamist [17] [18]. Ühe erinevusena klassikalise AHP-ga võrreldes eemaldab hägus AHP meetod lõpptulemustest kõik alternatiivid mida peetakse ebaoluliseks kõigi hindajate poolt [17]. [17] on väidetud, et eelnev on üks põhjustest miks teatud osa eksperte ei aktsepteeri selle meetodi tulemusi ning on öeldud, et just mõnede Euroopa teadlaste jaoks on hägusate kogumike teooria kultuurilistel põhjustel raskemini aktsepteeritav/kasutatav, kuid näiteks Jaapanis küllaltki laialdaselt kasutatav. Teiseks, hägusa AHP peal ei ole klassikalise AHP kindlusindeksi kasutamine võimalik, seega klassikaline AHP ja hägus AHP ei ole otseselt samades tingimustes võrreldavad/kasutatavad [17].

Praegusel hetkel on raske leida töid, mis tegeleks otseselt sellist tüüpi Monte Carlo sensitiivsusanalüüsi mida praeguses töös kasutada tahetakse kõrvutamisega hägusa AHP meetodi kasutamisega ning nende omavaheliste tulemuste võrdlemistega, mistõttu on keeruline välja tuua ühe või teise meetodi selgeid eeliseid lõpptulemustes teineteise ees. Siiski eelmises paragrahvis välja toodud erinevused ja hägusa meetodi iseärasused ning üleüldine klassikalise mudeli suurem kasutatavus on põhjused miks antud magistritöös keskendutakse klassikalise mudeli poolt tagastatavate tulemsute tundlikkuse analüüsi uurimisele.

Negatiivsest küljest on [12] artiklis kirjeldatud erinevaid AHP mudeli puudujääke, nii selliseid mis on ümberlükatud või millele on lahendus loodud, kui ka selliseid, mis on aktuaalsed. Vaatamata sellele on nii selles samas töös kui ka näiteks [13], [14], [18] jt. töödes näidatud, et mudel suudab leida erinevatel aladel rakendust.

2.5.2 Kokkuvõte

Antud peatükis anti ülevaade eelnevalt valdkonnas tehtud töödest. Esimesest artiklist selguse, et juhul kui hinnangud on küllaltki kindlad ja järjepidavad, siis MCAHP ei anna klassikalise AHP ees eelist, kuid ebakindluse suurenemisel nende erinevused suurenevad ning võrreldes usaldusväärsete tulemustega teistest allikatest tagastab MCAHP meetod täpsemaid tulemusi.

Tuli välja klassikalise AHP mudeli tõenäosushinnangute puudumine lõpptulemustest, mis ei anna kasutajale selget ülevaadet alternatiivide paremusest üksteise suhtes.

Erinevates teadustöödes on kasutatud mitmeid erinevaid lahendusi kuidas Monte Carlo arvutust kasutati tundlikkuse analüüsi tulemusteni jõudmiseni, mis võib hilisemalt anda ka antud magistritöö jaoks näiteid kuidas seda oleks kõige optimaalseim teha.

Antakse ülevaade tööst milles kirjeldatakse standardit mida antud töö kasutab ning ühest magistritööst mis on seda varasemalt kasutanud ning kus on mainitud tulevikusuunana sarnase eesmärgiga lahenduse loomist.

Uuriti ka peamist meetodit mida võib alternatiivseks lähenemiseks pidada ning üldiseid praeguses töös kasutatava mudeli puuduseid, kuid jõuti järelduseni, et praegusel hetkel on mudeli kasutamine põhjendatud ja aktuaalne.

3 Töös kasutatavad tehnoloogiad ning analüüs

Järgnevas peatükis antakse ülevaade töö käigus kasutatavatest tehnoloogiatest mis võimaldavad luua rakenduse mille abil saab AHP mudelile tundlikkuse analüüsi teostada. Peamiseks fookuseks on XMCDA standard ja peamised andmestruktuurid mis on vajalikud et magistritöö rakendust oleks võimalik luua. Lisaks kirjeldatakse näidismudelit mille peal hiljem arvutusi hakatakse teostama.

3.1 Töös kasutatavad tehnoloogiad

Alapeatükis kirjeldatakse rakenduse jaoks kasutatavaid tehnoloogiaid.

3.1.1 XMCDA standard

3.1.1.1 Ülevaade

XMCDA on andmestandard mille eesmärgiks on võimaldada kirjeldada MCDA andmeelemente XML-formaadis järgides kindlat struktuuri. XMCDA on UMCDA-ML instants, mis on Decision Deck-i poolt loodud universaalne modelleerimise keel mille eesmärgiks on aidata kaasa MCDA ja otsustustegevuste protsessidele [19].

XMCDA eesmärgiks on lihtsustada erinevate MCDA algoritmide omavahelist suhtlust, lihtsustada algoritmide kasutamist ning visuaalselt väljendada MCDA algoritmide andmestruktuure läbi tavapäraste vahendite nagu veebilehitsejad [15].

XMCDA standardi loomise aluseks oli 4 peamist põhjust, mis avaldusid MCDA algoritmide laialdasemal kasutusele võtmisel. Esimene põhjus oli erinevate algoritmide loomine eraldiseisvates tarkvaratoodetes ehk ei olnud ühte keskset tarkvara mis kõike võimaldaks. Teiseks põhjuseks olid raskused erinevate toodete kasutamisel kuna andmete sisendstruktuurid olid erinevad. Kolmandaks, mitmed teadusartiklites avaldatud MCDA algoritmide ei olnud hiljem kättesaadavad ning viimaseks mitmed MCDA tööriistad ei olnud vabavaraliselt kättesaadavad. Lahenduseks alustati 2009. aastal XMCDA standardi loomist

XMCDA on XML-põhine skeem ning Decision Deck-i eesmärgiks on MCDA algoritmid luua selle põhjal ning need hiljem veebiteenustena vabavaraliselt kättesaadavaks teha. Praegusel hetkel on XMCDA peamiseks sihiks sooritada vajalikud arvutussammud ning

sisenditest jõuda väljunditeni, see ei keskendu näiteks rollidele kes otsusega seotud on [15].

Lisaks veebiteenustele on Decision Deck loonud ka tarkvara mis võimaldab selliseid teenuseid visualiseerida – Diviz [19].

Töö kirjutamise hetkel oli ametlikuks XMCDA standardi versiooniks 2.2.2. Kuigi versioon 3.0.1 oli juba välja lastud, siis seda ei peetud veel töökorras olevaks versiooniks ning seetõttu kasutati kogu töö käigus esimesena mainitud standardit [19].

3.1.1.2 Peamised vajalikud andmestruktuurid rakenduse jaoks

Magistritöö eesmärgiks on luua AHP arvutusi ja selle tagastatud tulemustele tehtavat tundlikkuse analüüsi sooritavat teenust. Kuna töö kirjutamise hetkel ei olnud Decision Deck XMCDA veebiteenuste listis [20] ühtegi teenust mis kasutaks sisemiselt AHP mudeli struktuuri, siis oli töö üheks nõudeks uurida kas XMCDA võimaldab sellist teenust luua ning millised andmestruktuurid selle jaoks kasutatavad on.

Uurimise käigus selgus, et töö tegemisel ametlikult kasutusel oleva versiooni, 2.2.2, puhul on võimalik kirjeldada XMCDA põhiselt teenuseid mis sooritavad AHP arvutusi. Selle jaoks on vaja kasutada 8-t peamist XML tüüpi (lisaks nende alamelemendid), millest tehakse järgnevalt ülevaade. Visuaalne ülevaade kuidas need andmetüübid on sisemiselt omavahel seotud on nähtav Lisa 1-s.

Alternatives (alternatiivid) -

```
<alternatives>
  <description>
    <title>List of alternatives </title>
  </description>
  <alternative id="a01" name="Alternative1 name">
    <description>
      <comment>Description about alternative1 (e.g. name)</comment>
    </description>
  </alternative>
  <alternative id="a02" name="Alternative2 name">
    <description>
      <comment>Description about alternative2 (e.g. name)</comment>
    </description>
  </alternative>
</alternatives>
```

Alternatiivide tüüp koosneb ühest või mitmest üksik alternatiivist (Alternative) ja selle all kirjeldatakse ära kõik alternatiivid mille vahel otsust tehakse, samamoodi nagu seda on eeldatud ka tavapärase AHP mudeli puhul. Antud skeemi puhul on oluline, kirjeldada nii ID, kuna seda kasutatakse sisemiselt maatriksite loomisel ja arvutuste tegemisel, kui ka nimi, et kasutaja saaks lõpptulemustest aru milline variant lõpptulemustest on kõige parem.

Alternatiivi tüübile viidatakse alternatiivide võrdluste (alternativesComparisons) tüübist.

AlternativesComparisons (alternatiivide võrdlused)–

```
<alternativesComparisons>
      <description>
         <comment>Alternatives comparisons are only done under criteria
      elements connected with alternatives (last level before
      alternatives)</comment>
      </description>
      <criterionID>c4</criterionID>
      <pairs>
         <description><comment>Can be considered as alternatives matrix
      pairwise comparisons under c4 criterion</comment></description>
         <pair>
            <description><comment>Comment regarding
      pair</comment></description>
            <initial>
               <alternativeID>a01</alternativeID>
            </initial>
            <terminal>
               <alternativeID>a02</alternativeID>
            </terminal>
            <value>
               <real>3.0</real>
            </value>
         </pair>
         <pair>
            <initial><alternativeID>a03</alternativeID></initial>
            <terminal><alternativeID>a04</alternativeID></terminal>
            <value>
               <real>3.0</real>
            </value>
         </pair>
      </pairs>
   </alternativesComparisons>
```

Iga üksiku alternatiivide võrdluse tüübi all kirjeldatakse kriteeriumi või alamkriteeriumi ID ja kõigi alternatiivide paaritised võrdlused selle kriteeriumi suhtes. Kriteerium millele viidatakse peab mudelis olema viimasel tasandil enne alternatiividele viitamist (otsene ühendus kriteeriumi ja alternatiivide vahel). Iga alternatiivide võrdluse tüübi all peab kirjeldama kõigi võimalike paari võrdlused ning selliseid tüüpe peab olema sama palju kui on otseses alternatiividega ühenduses olevaid kriteeriumeid.

Alternatiivide ID viitab eelnevalt kirjeldatud alternatiivide tüübi all kirjeldatud alternatiivi ID-le ja peab olema täpne vaste sellega.

Loodava rakenduse puhul ei ole nõutud, et selle tüübi all peaks lisama võrdlused iseendaga või pöördelised võrdlused (ehk kui A1 vs A2 on olemas, siis A2 vs A1 ei ole vajalik) kuigi see on oluline AHP mudeli MxM maatriksite jaoks, sest neid arve on võimalik automaatselt genereerida.

Ideaalselt peaks võrdluste hinnangud järgima Saaty skaala väärtuseid ehk hinnangud 1-9 ja pöördelised hinnangud 1/9-1/1. Saaty skaala peamised defineeritud hinnangud ja arvude vastavus sõnadele on näidatud Lisa 2-s.

Alternatiivide võrdluste tüüp ei vaja ID-d kuna sellele mujalt ei viidata.

AlternativesSets (alternatiivide kogumid)–

```
<alternativesSet id="AS">
     <element>
        <alternativeID>a01</alternativeID>
     </element>
     <element>
        <alternativeID>a02</alternativeID>
     </element>
     <element>
        <alternativeID>a03</alternativeID>
     </element>
     <element>
        <alternativeID>a04</alternativeID>
     </element>
  </alternativesSet>
```

Alternatiivide kogumi tüüp sisaldab endas kõigi kirjeldatud alternatiivide ID-sid ja iseenda ID-d, viidates alternatiividele läbi üldise elemendi tüübi, mis on eraldi loodud iga alternatiivi jaoks.

Alternatiivide kogumi tüüpi kasutatakse hierarhia loomisel, et saaks kirjeldada millised kriteeriumid on otseses seoses alternatiividega.

Criteria (kriteeriumid)–

```
<criteria>
   <criterion id="c0" name="criterion name1"></criterion>
   <criterion id="c1" name="criterion name2"></criterion>
   <criterion id="c2" name="criterion name3"></criterion>
</criteria>
```

Kriteeriumite element koosneb mitmest ükskikust kriteeriumist. Igal kriteeriumil peab olema ID ja nimi, et neile oleks võimalik viidata. Kriteeriumite idee on sama mis AHP mudeli puhul ja siia alla kuuluvad nii kõik kriteeriumid kui ka kõik nende alamkriteeriumid. Esimene element peaks olema eesmärgi kriteerium mida kasutatakse hiljem mudeli üles ehitamise jaoks. Samuti peaks siin järgima Saaty AHP mudeli nõuet, et oleks olemas vähemalt 2 kriteeriumit mis tulemust mõjutavad kuna vähemal juhul ei ole alternatiive millegi all omavahel võrrelda.

Kriteeriumite elemendile viidatakse nii kriteeriumi kogumi kui kriteeriumi võrdluste elementidest.

CriteriaComparisons (kriteeriumite võrdlused)–

```
<criteriaComparisons>
   <description>
      <comment>Criteria comparisons are done for child criteria elements
      under the parent</comment>
   </description>
   <pairs>
      <pair>
         <initial>
            <criterionID>c1</criterionID>
         </initial>
         <terminal>
            <criterionID>c2</criterionID>
         </terminal>
         <value>
            <real>3.0</real>
         </value>
      </pair>
      <pair>
         <initial>
            <criterionID>c1</criterionID>
         </initial>
         <terminal>
            <criterionID>c3</criterionID>
         </terminal>
         <value>
            <real>1.0</real>
         </value>
      </pair>
   </pairs>
</criteriaComparisons>
```

Kriteeriumite võrdluse element on sarnane alternatiivide võrdluse elemendile ning see sisaldab endas kõiki ühte kriteeriumite kogumisse kuuluvate kriteeriumite paaritisi võrdlusi ehk mudeli kirjeldamisel ei tohiks võrrelda kriteeriumeid mis ei kuulu samasse kogumisse, mis tähendab, et nad on erinevatel tasanditel ning pole võrreldavad.

Peamine erinevus alternatiivide võrdluse elemendiga on see, et kriteeriumite võrdluse all saab kõik hinnangud ühe komplekstüübi all ära kirjeldada (kõik kriteeriumite paarid ühe CriteriaComparisons tüübi all, alternatiivide puhul on vaja luua mitu AlternativesComparisons komplekstüüpi selle jaoks). Kriteeriumitele viidatakse läbi nende ID.

Samamoodi nagu alternatiivide puhul ei ole siin vaja lisada iseendaga ega pöördelisi võrdluseid ning hinnangud peaksid järgima Saaty arvskaalat (Lisa 2).

Kriteeriumite võrdluse elemendile ei viidata mujalt.

CriteriaSets (kriteeriumite kogumid)-

```
<criteriaSets>
   <criteriaSet id="CS1">
      <element>
         <description>
            <comment>Criteria under Goal element</comment>
         </description>
         <criterionID>c1</criterionID>
      </element>
      <element>
         <criterionID>c2</criterionID>
      </element>
      <element>
         <criterionID>c3</criterionID>
      </element>
   </criteriaSet>
</criteriaSets>
```

Kriteeriumite kogumite all kirjeldatakse kõik samal tasandil olevad või ühe kriteeriumi alamkriteeriumiteks olevad kriteeriumid ühtse kogumina. Kriteeriumitele viidatakse läbi nende ID, mis peab olema täpne vaste. Kriteeriumite kogumeid kasutatakse selle jaoks, et oleks võimalik defineerida mudeli tasandite järjestus arvutuste jaoks. Üks alternatiiv ei tohiks kuuluda rohkem kui ühte kogumisse.

Kriteeriumite kogumitele viidatakse hierarhia tüübist.

Hierarchy (hierarhia)-

```
<hierarchy>
  <node>
     <description><comment>References to goal element. If goal element is
     not explictly created then it can be left
     empty</comment></description>
     <criterionID>c0</criterionID>
     <node>
        <criteriaSetID>CS1</criteriaSetID>
     </node>
  </node>
  <node>
     <criterionID>c1</criterionID>
     <node>
        <criteriaSetID>CS2</criteriaSetID>
     </node>
  </node>

  <node>
     <criterionID>c4</criterionID>
     <node>
        <alternativesSetID>AS</alternativesSetID>
     </node>
  </node>
</hierarchy>
```

Hierarhia element on väga oluline kuna see võimaldab kirjeldada AHP mudeli jaoks vajalikku puustruktuuri ning peaks teoreetiliselt võimaldama ka võrkstruktuuri kirjeldamist. Element koosneb sõlmpunktide (Node) elementidest, mis sisaldavad endas kriteeriumi viidet ja järgmist sõlmpunkti elementi mille sees on kas alternatiivide kogumi või kriteeriumite kogumi viide. Sellisel viisil on võimalik kogu mudeli järjestus defineerida.

Hierarhia elemendile endale ei viidata mujalt, kuid seda kasutatakse rakenduse siseselt viidete loomisel.

AlternativesMatrix (alternatiivide väärtused) –

Seda tüüpi kasutatakse väljundite salvestamiseks ning see koosneb algsest alternatiivist ja selle algse alternatiiviga seotud alternatiividest ning nende omavahelistest kindlushinnangutest.

3.1.2 XMCDA Java viite projekt

Selleks, et lihtsustada XMCDA standardi kasutamist on Decision Deck poolt loodud viite projektid erinevatele programmeerimiskeeltele (Java, Python, R). Selles viiteprojektis on objektidena/tüüpidena defineeritud kõik peamised andmestruktuurid mis XML skeemiski ja samuti on olemas vajalikud seosed. Peale andmestruktuuride defineerimise on olemas erinevad tööriistad mis lihtsustavad uue projekti loomist, näiteks sisendite parsimine (ingl.k *parse*) ja andmestruktuuridesse salvestamine ning väljundite automaatne kirjutamine failidesse.

Töös laiendatakse Java-põhist projekti, et kasutada võimalikult palju eelnevalt defineeritud objekte ning andmestruktuure ja selleks, et kasutada standardset andmete sisse lugemist ja kirjutamist.

3.1.3 Muud rakenduses kasutatud tehnoloogiad

Lisaks AHP mudelit kirjeldada võimaldavale XMCDA standardile ja Java viite projektile on rakenduses kasutatud paari vabavaralist matemaatilisi arvutusi lihtsustavat teeki

3.1.3.1 JAMA

JAMA on lihtne lineaaralgebra pakett Java programmeerimiskeele jaoks ning selle peamine kasutusala on maatrikstehete sooritamise jaoks võimalikult lihtsalt [21].

Antud magistritöö jaoks kasutatakse seda, et sooritada mõningaid maatrikarvutusi, näiteks maatriksi omaväärtuse leidmiseks.

3.1.3.2 Apache Commons Math

Apache poolt pakutav kergekaaluline teek mille eesmärk on pakkuda funktsioone enimlevinud statistika või matemaatika probleemide jaoks mis standardses Java programmeerimiskeeles või Commons Lang paketis ei ole saadaval [22].

Töös kasutatakse konkreetselt *Apache Commons-math3* versiooni ning selle JAR fail on lisatud projekti ressursside kausta. Teeki kasutatakse eesmärgil genereerida juhuslikke pöördvõrdelisi jaotuseid Monte Carlo simulatsiooni arvutuste jaoks. Samuti kasutatakse teeki lõpptulemuste pealt järelduste tegemise jaoks vajalike normaaljaotuse pealt tõenäosuste arvutamise jaoks vajalike tehete jaoks.

3.1.3.3 JDistlib

JDistlib on Java programmeerimise keele jaoks loodud vabavaraline matemaatiline pakett, mille eesmärgiks on võimaldada sooritada arvutusi ja teste statistikaliste jaotuste peal (normaaljaotus, binoomjaotus jne). Pakett on ümbertõlge R programmeerimiskeelest ning üldisel juhul pakub arvutustes samasugust täpsust [23].

Antud magistritöös on see vajalik, et lõpptulemuste tõlgendamisel testida kas tundlikkuse kordajate pealt arvutatud standardskoori väärtused on normaaljaotusega või mitte. Selle jaoks tehakse kogu kogumi peal normaalsustest. Normaalsusteste on erinevaid ning ka see pakett pakub suuremat osa enimlevinuid võimalusi, kuid artiklite [24] ja [25] tulemustele tuginedes osutus valikuks Shapiro-Wilk normaalsustest kuna leitakse, et sellel on parimad tulemused hinnangu andmisel.

3.2 Rakenduse jaoks loodud AHP mudel

Alapeatükis kirjeldatakse mudelit mis rakenduse sisendiks loodi ja mille peal tulemused arvutatakse.

3.2.1 Ülevaade

Antud töös loodava rakenduse ja selle tulemuste hindamise jaoks loodi näidis AHP mudel, mis hinnanguliselt oleks piisava keerukusega, et katta ära enam levinud stsenaariumid mudeli ja arvutuste juures. Mudeli keerukuse hindamisel lähtuti nii internetis enam levinud AHP mudelite näidetest, kui ka TTÜ-s sama teemaga seotud ainest ning seal loodavate aineprojektide keskmisest keerukusest ja aineprojekti nõuetest. Näide töös kasutatavast mudelist on Lisa-s 3.

Peamine nõue mudeli juures oli, et mudel koosneks rohkem kui kolmest tasandist ehk alamkriteeriumite lisamine oleks toetatud kuna see on tavapärane nõue AHP mudelite juures. Alternatiivide arvul ei olnud kindlat nõuet peale AHP eelduse, et alternatiive on 2 või enam. Paarikaupa hindamisel mõeldi piltlik olukord kus igal alternatiivil on omad eelised ja puudused ning, et hinnangute lõpptulemused oleksid lähedal üksteisele.

Näidismudel ei näita maksimaalse keerukusega võimalikku mudelit. Kuna hierarhia kirjeldamine on võimalik lõpmatult ja skooride arvutamisel ei ole piirangut, siis peaks tehniliselt olema toetatud ka palju sügavamate mudelite koostamine. Peamine mida peaks

arvestama mudeli kasvu juures on see, et tundlikkuse analüüsi teostamine on eksponentsiaalse raskusastmega ning mudeli kasvades tuleb arvestada ka ajakasvu suurenemisega.

3.2.2 Mudeli koostamine

Et rakendust kasutada on vaja kogu mudel kirjeldada XML-skeemina vastavalt XMCDA skeemi nõuetele. (Selle protsessi lihtsustamiseks lõppkasutaja jaoks ja kasutatavuse suurendamiseks esitatakse töö lõpus soovitusi visuaalse vahendi loomiseks, mis selle töö automaatselt ära teeks läbi graafilise kasutajaliidese).

Peatükis *3.1.1.2 Peamised vajalikud andmestruktuurid rakenduse jaoks* toodi välja kõik peamised XML tüübid mida on AHP mudeli jaoks vaja kasutada. Iga kirjeldatud tüübi kohta on vajalik teha eraldiseisev fail. Kõik failid peavad eksisteerima, et tulemusi oleks võimalik arvutada. Samuti peavad kõik failid järgima XML skeemi nõudeid ning sisaldama päises vajalikke viiteid ning faili sisu peab olema defineeritud XMCDA tüübi sees. Järgnevalt on näitena esitatud täidetud CriteriaSets fail koos kõigi vajalike XML kirjeldustega

```
<?xml version="1.0" ?>
      <xmc:XMCDA xmlns:xmc="http://www.decision-deck.org/2017/XMCDA-
      2.2.2"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.decision-deck.org/2017/XMCDA-2.2.2
       http://www.decision-deck.org/xmcda/_downloads/XMCDA-2.2.2.xsd">

         <criteriaSets>
            <criteriaSet id="CS1">
               <element>
                  <description>
                     <comment>Criteria under element1</comment>
                  </description>
                  <criterionID>c1</criterionID>
               </element>
               <element>
                  <criterionID>c2</criterionID>
               </element>
               <element>
                  <criterionID>c3</criterionID>
               </element>
            </criteriaSet>
            <criteriaSet id="CS2">
               <element>
```

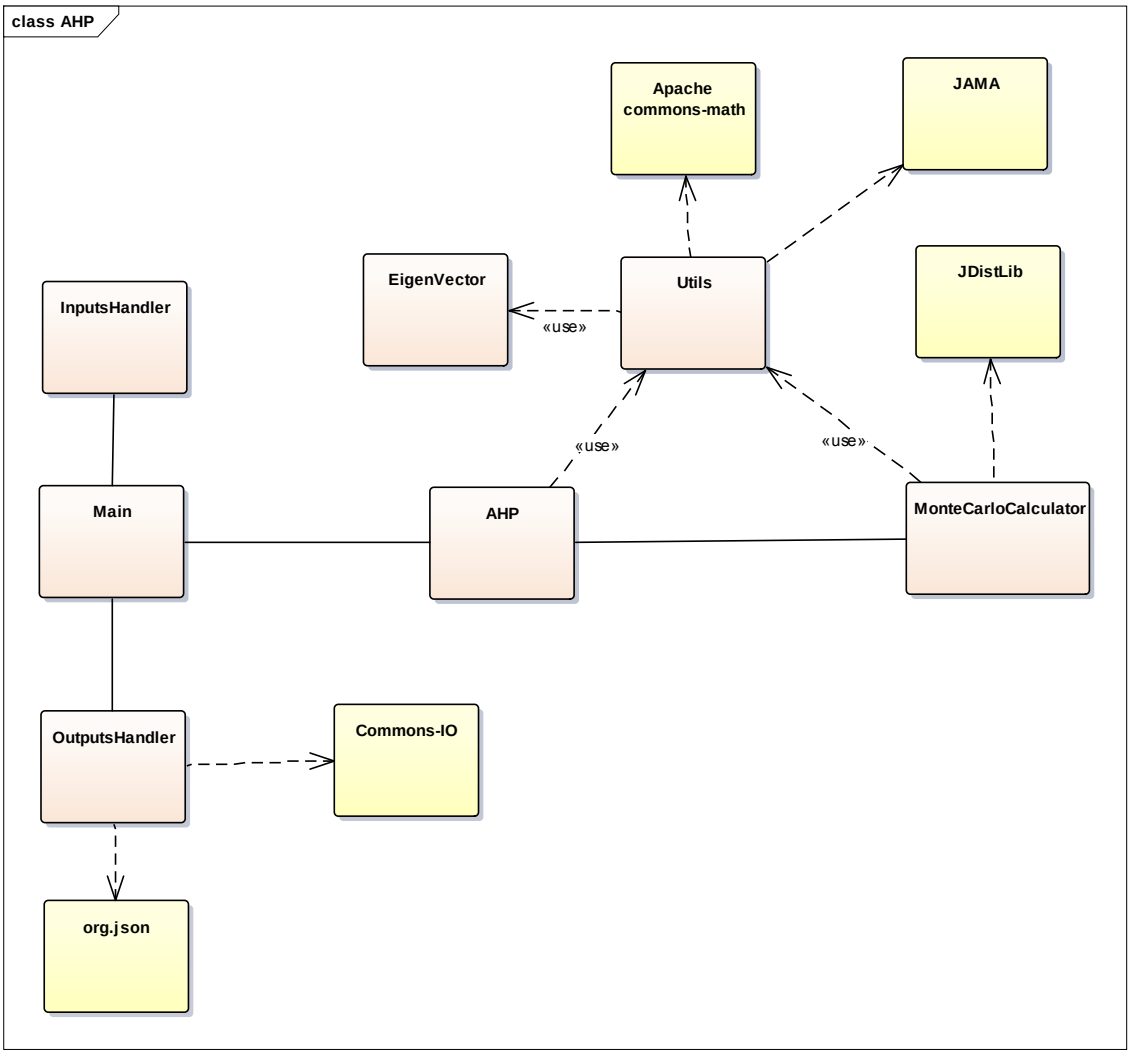
```
            <description>
               <comment>Criteria under element2 </comment>
            </description>
            <criterionID>c4</criterionID>
         </element>
         <element>
            <criterionID>c5</criterionID>
         </element>
      </criteriaSet>
      <criteriaSet id="CS3">
         <element>
            <description>
             <comment> Criteria under element3</comment>
            </description>
            <criterionID>c6</criterionID>
         </element>
         <element>
            <criterionID>c7</criterionID>
         </element>
      </criteriaSet>
      <criteriaSet id="CS4">
         <element>
            <description>
             <comment>Criteria under element4</comment>
            </description>
            <criterionID>c8</criterionID>
         </element>
         <element>
            <criterionID>c9</criterionID>
         </element>
      </criteriaSet>
   </criteriaSets>

</xmc:XMCDA>
```

4 Realisatsiooni teostamine

Järgnevas peatükis antakse ülevaade magistritöö käigus realiseeritud rakendusest. Kirjeldatakse rakenduse toimimist, andmete sisendit ja väljundit, üksiku AHP mudeli tulemuste arvutamist ning Monte Carlo simulatsiooni abil tundlikkuse analüüsi mudeli lõpptulemuste peal. Lisaks võrreldakse algseid ja tundlikkuse analüüsi põhiseid tulemusi ning neid erinevusi. Peatüki lõpus uuritakse võimalikke teistsuguseid lähenemisi rakenduse loomiseks ja kas on teisi meetodeid mis võimaldaks AHP tundlikkuse analüüsi asemel tulemusi arvutada.

4.1 Rakendus

Antud magistritöös on loodud Java-põhine konsoolirakendus, mis järgib eelnevalt kirjeldatud XMCDA standardit (vt. peatükk 3.1.1.) ning selle jaoks loodud Java viite projekti (vt. peatükk 3.1.2.). Joonisel 9 on esitatud loodud rakenduse klasside vahelised seosed. Helekollase värviga (katkendlik joon, ilma märketa) on esitatud sõltuvused välistest pakettidest (vt. peatükk 3.1.3). Rakendus võtab sisendiks standardi põhjal kirjeldatud XML failid, mille põhjal luuakse rakenduse siseselt vajalik AHP struktuur ja arvutatakse AHP mudeli lõpptulemused. Lisaks klassikalise AHP mudeli tulemuste arvutamise võimaldab rakendus teostada Monte Carlo simulatsiooni põhjal tundlikkuse analüüsi ning anda lisaks alternatiivide lõppkaalude hinnangutele ka alternatiivide järjestuse kindlushinnangud. Lõpuks tagastatakse kõik tulemused XML ja ka JSON failina. Kogu arvutuskäik on kirjeldatud järgmistes alapeatükkides.



Joonis 6. Rakenduse klasside vahelised seosed ja sõltuvused

4.1.1 Sisend ja andmete töötlus

Lühiülevaade kuidas töö jaoks loodud sisendmudel koostati on esitatud alapeatükis (3.2)

Sisendite (ja väljundite) välja võtmisel (ja salvestamisel) on järgitud Decision Decki poolt loodud näiteprojekti [26] loogikat, et järgida ühtset lähenemist erinevate projektide vahel. XMCDA Java viiteprojektis on olemas parser, mis eraldab automaatselt XML failidest andmed ja jagab need õigetesse andmestruktuuridesse (objektidesse). Kõik andmed on algselt salvestatud Java veebiteenuste objektidena (JAXB) üksikus XMCDA Java objektis, mida töödeldakse rakenduse poolt, et saada ainult arvutuste jaoks vajalikud andmed ja seosed programmi jaoks. Samuti teostatakse kontrollid, et kasutaja on sisestanud kõik vajalikud andmed AHP mudeli jaoks. Rakenduse arvutuste tegemisel

eelistatakse liste ja andmemassiive andmete töötlemisel objektide asemel seega pärast sisendite töötlemist kasutakse ka neid andmestruktuure.

Kui kõik andmed on kasutajapoolsest sisendist välja võetud ja läbi töödeldud, siis need salvestatakse ühisesse sisend (Input) klassi (vt. Lisa 5), mis antakse hiljem arvutusfunktsioone teostavale klassile sisendiks.

4.1.2 Üksiku mudeli tulemuste arvutamine

Eelnevas alapeatükis mainitud sisendi (Input) klass edastatakse AHP arvutusi tegevale klassile ja sealsetele funktsioonidele. Kogu AHP arvutuste osa saab jagada neljaks peamiseks sammuks: alternatiivide ja kriteeriumite lokaalsete skooride leidmine ja nende tulemuste põhjal mõlemale globaalsete skooride leidmine.

Kõik lokaalsed skoorid leitakse üksiku maatriksi pealt ja n.ö. isolatsioonis ülejäänud mudelist (lõpptulemused maatriksi all annavad kokku väärtuseks 1) ehk need lõpptulemused näitavad elementide suhet üksteisesse selles võrdluses, kuid ei näita veel kogu tähtsust mudelis ega alternatiivide võrdluse puhul nende lõppjärjestust.

Tabel 1. Elementide lokaalsete skooride tabel

	Element1	Element2	Element3	Lõpptulemused
Element1	1	3	7	**0,669**
Element2	1/3	1	3	**0,243**
Element3	1/5	1/3	1	**0,088**

Maatriksite kaalud on arvutatud kasutades astme iteratsiooni meetodit. Algoritmi jaoks kasutatakse varasemalt teiste autorite poolt tehtud vabavaralisi Java näiteid.

Kuna kriteeriumitel võib ka alamkriteeriumeid olla, kuid alternatiividel ei saa alamalternatiive olla, siis nende kahe globaalsete tulemuste tehniline pool erineb, kuid idee on sama.

Alamkriteeriumite tulemusi arvutatakse rekursiivselt kuna ei ole teada kui mitu kihti alamkriteeriumeid on algmudelis ning, et oleks võimalik ka keerukamaid mudeleid rakenduses kasutada. Et saada teada globaalsed skoorid tuleb kõik alamkriteeriumid korrutada läbi nende ülema elemendi skooriga, alustades mudeli kõige ülemisest ehk

eesmärgi punktist ja liikudes järjest allapoole mudelis kuni ei ole enam ühtegi alamkriteeriumit (näitlik struktuur on Lisa 1-s). Kõik alternatiividega ühenduses olevad kriteeriumid peavad lõpuks andma kokku väärtuseks 1. Järgneval skeemil on piltlik näide kuidas Tabel 1 lokaalsetest skooridest arvutatakse globaalsed skoorid juhul kui tegemist oleks alamkriteeriumite hinnangutega mille ülemkriteeriumi kaaluks kogu mudelis on 0,25.

$$\begin{matrix} 0{,}669 \\ 0{,}243 \\ 0{,}088 \end{matrix} * 0{,}25 = \begin{matrix} 0{,}167 \\ 0{,}061 \\ 0{,}022 \end{matrix}$$

Alternatiivide puhul ei pea arvestama mitme tasemega ning arvutuskäik on lihtsam, kuid idee poolest tuleb samuti iga alternatiivi tulemused korrutada läbi iga nendega vahetuses ühenduses oleva kriteeriumi kaaluga. Liites iga alternatiivi kaalud eraldi kokku saadakse vektor, kus on toodud välja alternatiivide lõplikud kaalud mudelis. Kaalud on esitatud punkthinnangutena.

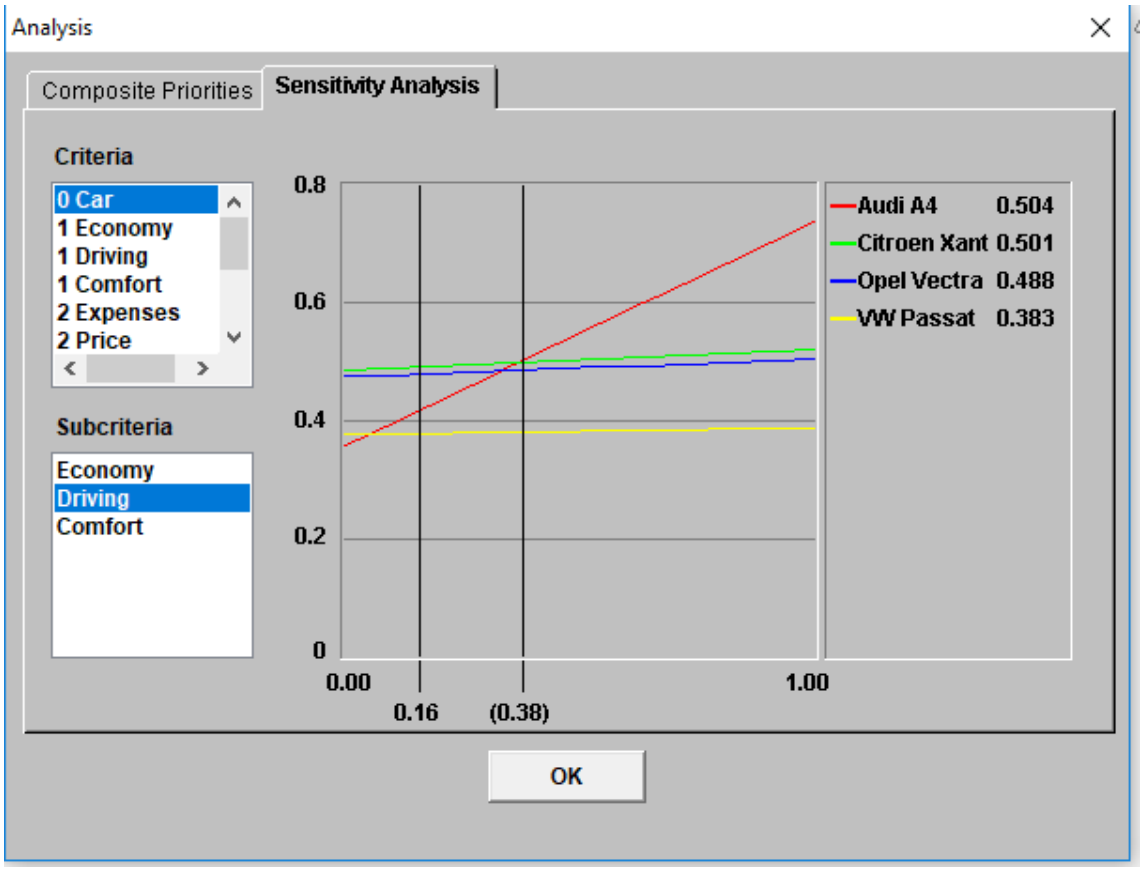
Tabel 2. Üksiku AHP arvutuse lõpphinnangud

Alternatiiv	Lõpphinnang
Alternatiiv1	0,333
Alternatiiv2	0,167
Alternatiiv3	0,500

4.1.3 Tundlikkuse analüüsi rakendamine Monte Carlo meetodi abil

Eelmises alapeatükis kirjeldati kuidas jõutakse kogu mudeli lõppkaaludeni. Antud juhul oli tegemist ühe iteratsiooni lõppkaaludega, mis on leitud samasel viisil nagu enamlevinud AHP arvutusi pakkuvad rakendused või lihtsad tabelarvutusprogrammid seda võimaldavad. Kuna AHP on oma olemuselt ebakindlate sisenditega (hinnangutega), siis ei saa neid tulemusi täielikult usaldusväärseks pidada ilma mingit tüüpi usaldusväärsus testi läbi viimata. Enamasti on selle jaoks AHP puhul kasutusel tundlikkuse analüüs, mis üritab välja selgitada kas mudel praegusel kujul on sobilik või on vajalik uus iteratsioon teha ehk hinnangud ümber muuta.

Üheks klassikaliseks viisiks kuidas mudelile tundlikkuse analüüsi saab teha on kasutades graafilist liidest, mis näitab millisel hetkel osakaalude muutmisel muutub ka alternatiivide järjestus.



Joonis 7. Käsitsi tundlikkuse analüüs Web-Hipre põhjal.

Allikas: Web-Hipre Java applet, http://hipre.aalto.fi/

Eelneval pildil (joonis 10) on näha käsitsi tundlikkuse analüüsi jaoks mõeldud tööriista, kuid see nõuab pärast hinnangute muudatuste välja selgitamist seda, et arvutatakse tundlikkuse kordajad [27], mille põhjal analüüsitakse kas muudatus on tõenäoline või mitte. Juhul kui on kahtlus, et muudatus on tõenäoline, siis katsetatakse muudatus algsete paaritiste võrdlushinnangute juures läbi ning pärast seda antakse uuesti nende pealt subjektiivsed hinnangud kas mudeli muudatus on oodatav või mitte. Samuti on sellist analüüsi vaja teha iga kriteeriumi all. Lõppkasutaja jaoks tähendab see, et tal on vaja teadmisi kuidas antud liidest kasutada ning kuidas selle pealt tundlikkuse kordajaid arvutada ja hinnanguid anda ning uusi kaale paarisvõrdlustele leida.

$$TundlikkuseKordaja = \frac{uusOsakaal}{1 - uusOsakaal} * \frac{1 - algneOsakaal}{algneOsakaal}$$

Et muuta kogu protsess lihtsamaks tavakasutaja jaoks ning automatiseerida tundlikkuse analüüsi protsess sooritatakse antud töös analüüs kasutades Monte Carlo simulatsiooni ehk Monte Carlo iteratsiooni lõpptulemused (alternatiivide järjestused ja kaalud) arvutatakse välja suurel hulgal kordi ning lõpuks leitakse selle pealt optimaalse lahendi kindlusaste kasutades binoom- või normaaljaotus arvutusi.

Nagu peatükis *2.3.1 Monte Carlo arvutus* on kirjeldatud siis Monte Carlo simulatsioon töötab asendades igal iteratsioonil ebakindlust sisaldavad võrdlushinnangud tõenäosusjaotusest (enamasti normaaljaotus) valitud arvudega ning see läbi arvutades AHP mudeli uue tulemuse ning lõpuks iga üksiku MC iteratsiooni lõpptulemused näitavad ühte võimalikku lahendust. MC tulemuste pealt saab välja selgitada kui tõenäoline on mingi tulemuseni jõudmine või antud juhul kui kindel on, et alternatiiv on mingil üheselt määratud positsioonil paremusjärjestuses.

AHP mudeli puhul eksisteerib ebakindlus kõigis paarikaupa võrdlustes kuna enamasti need hinnangud on antud vähemalt osaliselt subjektiivselt. Seega sooritatakse töös MC simulatsioon asendades igal iteratsioonil kõigi paarikaupa võrdluste väärtused uute normaaljaotuse põhjal genereeritud väärtustega, kus normaaljaotuse keskväärtuseks on algne võrdlushinnang ja normaaljaotuse standardhälbeks on antud võrdlusmaatriksi veakordajate hajuvuse pealt arvutatud standardhälve.

Uute paarikaupa võrdluste hinnangute leidmine ühe kriteeriumi all:

1. Algses mudelis antud paarikaupa hinnangud (hinnangud on näidiseks)

Tabel 3. Võrdlusmaatriksi algsed hinnangud

	Element 1	Element 2	Element 3
Element 1	1.0	3.0	2.0
Element 2	1/3	1.0	1/2
Element 3	1/2	2.0	1.0

2. Iga maatriksi puhul leitakse hinnangute põhjal veakordajad (paremal pool peadiagonaali olevate väärtuste põhjal).

   Veakordajate leidmise valem üksiku võrdlushinnangu kooskõlastamiseks [11]:

$$\frac{1}{Prod\big(col(n-1)\big) * Prod(row(n) * nVal^{veerge})^{\frac{1}{veerge-2}}},$$

$$n = alternatiivi\ indeks$$

Tabel 4. Algsete hinnangute pealt leitud veakordajad

	Element 1	Element 2	Element 3
Element 1	-	$1\frac{1}{3}$	0.75
Element 2	-	-	$1\frac{1}{3}$
Element 3	-	-	-

3. Veakordajate peal kasutatakse logaritmi (antud näites naturaallogaritmi), sest algse veakordajate suhteskaalal veakordajad ei järgi normaaljaotust.

Tabel 5. Veakordajate pealt kasutatakse logaritmi

	Element 1	Element 2	Element 3
Element 1	-	$\ln 1\frac{1}{3}$	ln 0.75
Element 2	-	-	$\ln 1\frac{1}{3}$
Element 3	-	-	-

Eelnenud tabeli (Tabel. 3) kõigi logaritmitud väärtuste pealt leitakse ka üksikväärtusena standardhälve, mida kasutatakse järgnevates arvutustes.

4. Genereeritakse juhuarvud logaritmitud veakordajate normaaljaotusest. Rakenduses kasutatakse selle jaoks Apache Commons-math funktsiooni kumulatiivsest tõenäosuse leidmiseks normaaljaotusest.

Tabel 6. Juhuarvude genereerimine Apache Commons-math paketi abil

	Element 1	Element 2	Element 3
Element 1	-	**normalDistribution .inverseCumulativeProbability(Math.*random*())**	**normalDistribution .inverseCumulativeProbability(Math.*random*())**
Element 2	-	-	**normalDistribution .inverseCumulativeProbability(Math.*random*())**
Element 3	-	-	-

5. Eelnevalt genereeritud juhuarvud korrutatakse läbi standardhälbega mis leiti 3-nda punkti tulemuste pealt.

Tabel 7. Juhuarvude ja standardhälbe abil uute MC pooljuhuslike logaritmitud veakordajate leidmine

	Element 1	Element 2	Element 3
Element 1	-	Rand * stDev	Rand * stDev
Element 2	-	-	Rand * stDev
Element 3	-	-	-

6. Korrutatud arvudele rakendatakse logaritmimise pöördtehet (antud näites viiakse e-astmeks), tulemuseks on uued MC pooljuhuslikud veakordajad.

Tabel 8. Pooljuhuslike MC logaritmitud veakordajate e-astmele viimine

	Element 1	Element 2	Element 3
Element 1	-	$e^{rand*stDev}$	$e^{rand*stDev}$
Element 2	-	-	$e^{rand*stDev}$
Element 3	-	-	-

7. Uued paarikaupa hinnangud

Tabel 9. Uued paarikaupa hinnangud

	Element 1	Element 2	Element 3
Element 1	1.0	AlgneHinnang * $e^{rand*stDev}$	AlgneHinnang * $e^{rand*stDev}$
Element 2	Pöördväärtus	1.0	AlgneHinnang * $e^{rand*stDev}$
Element 3	Pöördväärtus	Pöördväärtus	1.0

Lõpptulemuste pealt arvutatakse uued lokaalsed kaalud maatriksi tulemustele mida kasutatakse globaalsete tulemuste leidmiseks. Protsessi läbi viimine ja juhuarvude genereerimine viib selleni, et iga Monte Carlo iteratsiooni lõpptulemusena saab teoorias iga alternatiiv uue lõppkaalu. Näide kuidas lõpptulemuste kaalud erinevad lõpuks üksteisest on esitatud Lisa-s 4.

Kõigi iteratsiooni tulemuste pealt arvutatakse hääletustabel. Hääletustabel on MxM maatriks kus on kirjas kõigi alternatiivide omavahelised tulemused ehk kui mitu korda alternatiivid võitsid või kaotasid üksteisele. Mida suurem on ühe alternatiivi võitude arv teise suhtes seda kindlamalt saab väita, et see alternatiiv on ka parem valik.

Tabel 10. Monte Carlo tulemuste pealt loodud hääletustabel

	Alternatiiv 1	Alternatiiv 2	Alternatiiv 3
Alternatiiv 1	0	555	675
Alternatiiv 2	445	0	808
Alternatiiv 3	325	192	0

Parima pingerea leidmiseks on võimalik arvutada milline alternatiiv oli teise suhtes rohkem võitnud ehk tuleb leida vähima riketega pingerida. Sellisel viisil, aga ei saa hinnata alternatiivi paremust teise suhtes ehk ei oleks võimalik anda kindlushinnangut.

Kindlushinnangu leidmiseks kahe alternatiivi vahel kasutatakse antud magistritöös kahte lahendust: standardskoori ja binoomjaotuse abil hindamist. Lõpptulemustes kajastatakse ainult optimaalset lahendit.

Selleks, et teha kindlaks kumba viisi kasutada arvutatakse kahe võrreldava alternatiivi kõigi MC iteratsioonide lõpptulemuste pealt tundlikkuse kordajad ning viiakse need logaritmskaalale. Logaritmitud väärtuste peal tehakse normaalsustest ehk kontrollitakse kas tegemist on normaaljaotusega. Alapeatükis 3.1.3.4 on mainitud, et antud töös kasutatakse *Shapiro-Wilk* normaalsustesti läbi JDistlib paketi selle jaoks. Normaalsuse hindamise määraks on valitud testi statistiline väärtus (w) > 0.99 ja p-value > 0.05 vastavalt enamlevinud soovitustele [28] ehk kui need tingimused on täidetud, siis ei saa ümber lükata väidet, et väärtused on normaaljaotusega. Juhul kui *Shapiro-Wilk* normaalsustest näitab, et logaritmitud tundlikkuse kordaja väärtused on normaaljaotusega, siis leitakse lõpptulemuste kindlushinnangud läbi logaritmitud tundlikkusekordajate standardskoorile (z-skoor) teisendamise ja sellelt järelduste tegemise meetodile. Logaritmitud tundlikkusekordajate standardskooridest moodustub standard skaala (joonis 11).

Standardskoor näitab kui mitu standardhälvet on vaadeldav arv kaugemal vaadeldavate arvude keskväärtustest mille suhtes hinnangut antakse. See on kasulik kuna annab võimaluse arvutada välja punkti esinemise tõenäosuse normaaljaotuses. Teiseks laseb see meetod võrrelda skoore mis pärinevad erinevatest normaaljaotustest. Meetodi kasutamiseks on vajalik teisendada kõik vaadeldavad arvud standard arvudeks, et luua standard normaaljaotus [29].

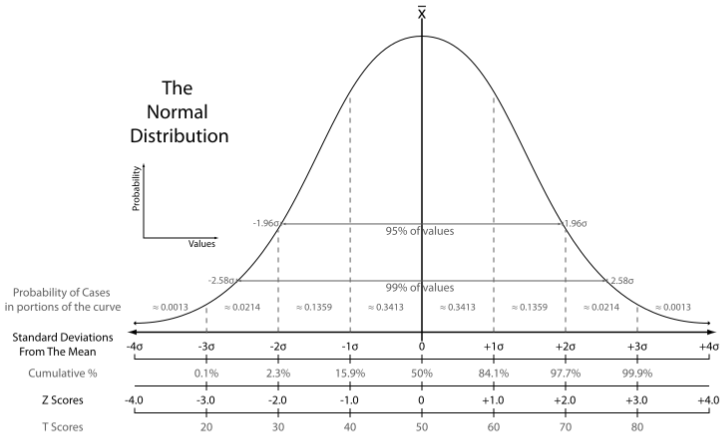
Töös viiakse logaritmitud tundlikkusekordajad standardskoorile (z-skoorile) kasutades tavapärast valemit

$$Z = \frac{hinnang - keskmine}{standardHälve}$$

, kus keskmise väärtuseks on logaritmitud tundlikkusekordajate keskmine ja hinnanguks on üksik logaritmitud tundlikkusekordaja.

Normaaljaotuses punkti esinemise tõenäosuse arvutamiseks on kasutatud töös Apache Commons-math3 NormalDistribution klassi cumulativeProbability(double meanValue) funktsiooni. Initsialiseerides normaaljaotuse klassi kasutades selleks standardskooride pealt arvutatud standardhälvet ja keskmist väärtust ning kasutades nimetatud funktsiooni sisendina algsete logaritmitud tundlikkusekordajate keskmist väärtust, saab tagastatud tulemuseks standardhälvete põhjal hinnangu mida kasutatakse selleks, et öelda kui suure tõenäosusega on üks alternatiiv teisest parem.

Kumulatiivse tõenäosuse (cumulativeProbability) funktsioon toimib statistilise ’68 95 99.7’ reegli [30] põhimõttel, kus ühe standardhälbe kaugusele keskmisest jääb 68% väärtustest, kahe standardhälbe kaugusele 95% väärtustest jne. Funktsioon võtab sisendiks standardhälbe väärtuse ja arvutab selle pealt protsentuaalse väärtuse (paremuse kindlushinnangu).



Joonis 8. Standard skaala (z-skaala)

Allikas: https://en.wikipedia.org/wiki/Standard_score#/media/File:The_Normal_Distribution.svg

Juhul kui *Shapiro-Wilk* normaalsustest näitab, et logaritmitud tundlikkuse kordaja väärtused ei ole normaaljaotusega leitakse lõpptulemuste kindlushinnangud binoomjaotuse arvutuste abil.

Binoomjaotust kasutatakse ainult siis, kui nullhüpotees, et tegemist on normaaljaotusega on ümber lükatud ja ei saa kasutada normaaljaotust eeldavaid arvutusi. Binoomjaotus arvutuste jaoks on töös kasutatud Apache Commons-math paketi BinomialTest() funktsiooni, milles edukateks tulemusteks on valitud selle alternatiivi tulemuste arv millel oli neid teisest rohkem. Binoomkatsete sündmuse tõenäosus on määratud tasemel 0.5 ning kasutatakse ühepoolset (one-tail probability) usalduspiiri.

Tabel 11. Binoomkalkulaatori sisenditeks valitud parameetrid

$BinomialTest(x, y, z, n)$
X = Monte Carlo iteratsioonide arv
Y = Paremuste arv
Z = 0.5 (tõenäosus)
N = ühepoolne (one-tail)

Tulemusena tagastab selline binoomjaotuse põhine arvutus samamoodi protsentuaalse väärtusena alternatiivide vahelise paremuse kindlushinnangu nagu seda tegi kirjeldatud normaaljaotuse põhine arvutus.

Kuna juhuslikud arvud on genereeritud normaaljaotusest ning kõik lõpptulemused on seoses algsete paarikaupa võrdlustega, siis selgus suurema osa arvutuste korral, et logaritmitud tundlikkuse kordajad on normaaljaotusega ning binoomjaotuse põhjal on tulemusi vaja harva leida.

4.1.4 Väljund ja tulemuste tõlgendamine

Eelmises peatükis kirjeldatud tundlikkuse analüüsi järel tagastatakse kasutajale tulemid, milles esitatakse alternatiivide vahelised paremused ning selle kindlushinnangud vastavalt peatüki 4.1.3 statistilisele analüüsile. Tulemused esitatakse nii standardipõhise XML failina , kui ka JSON failina, et suurendada universaalsust arvestades mõlema formaadi levikut ja üldist laia kasutatavust. Tagastatud tulemusi on võimalik rakendada lihtsalt kasutajapoolseks edasiseks analüüsimiseks ja järelduste tegemiseks, aga on

võimalik soovi korral edastada ka sisendiks mõnele teisele arvutusi tegevale rakendusele, mis on üheks Decision Deck-i eesmärgiks vastavalt eelmistele peatükkide kirjeldustele (vt. peatükk 3.1.1.1).

```
{
    "a02":[
        {"a04":0.91857},
        {"a03":0.65321}
    ],
    "a01":[
        {"a02":0.88249},
        {"a04":0.9951},
        {"a03":0.94312}
    ],
    "a03":[
        {"a04":0.84171}
    ]
}
```

Joonis 9. JSON formaadis lõpptulemused

Joonis 12. peal on näidatud JSON formaadis salvestatud lõpptulemused. Tulemused on tagastatud sellise mõttega, et näidata alternatiivi ID-d ning teiste alternatiivide ID-sid millest see parem on ning rakenduse poolt arvutatud paremuste kindlushinnanguid. Antud magistritöö käigus loodud rakendus väljastab statistilised kindlushinnangud, mille sisu kirjeldus on järgnev: juhul kui alternatiiv 1 on parem kui alternatiiv 2 ning nendevaheline kindlushinnang on 0.800, siis seda saab tõlgendada kui alternatiiv 1-e paremust neljal juhul viiest, samas kui viiendal juhul võib osutada alternatiiv 2 paremaks valikuks. Juhul kui rakendust kasutada klassikalisel moel ning ühekordseks tulemuste arvutamiseks ilma tundlikkuse analüüsi sooritamata, on võimalik esitada ka klassikalise AHP mudeli lõppkaalude punkthinnangud. Tulemuste pealt on ka näha, et lõpuks tagastatakse kõik alternatiivid mis on vähemalt ühest teisest alternatiivist kindlushinnangu põhjal üle (Joonis 12, a04 jääb kõigile teistele alla ja seda ülemelemendina ei esitata). Lõplikult hinnanguliselt parimaks alternatiiviks on objekt, millel on kõige rohkem alamelemente.

4.2 Üksiku iteratsiooni ja tundlikkuse analüüsi kaudu saadud tulemuste võrdlus ja hinnang

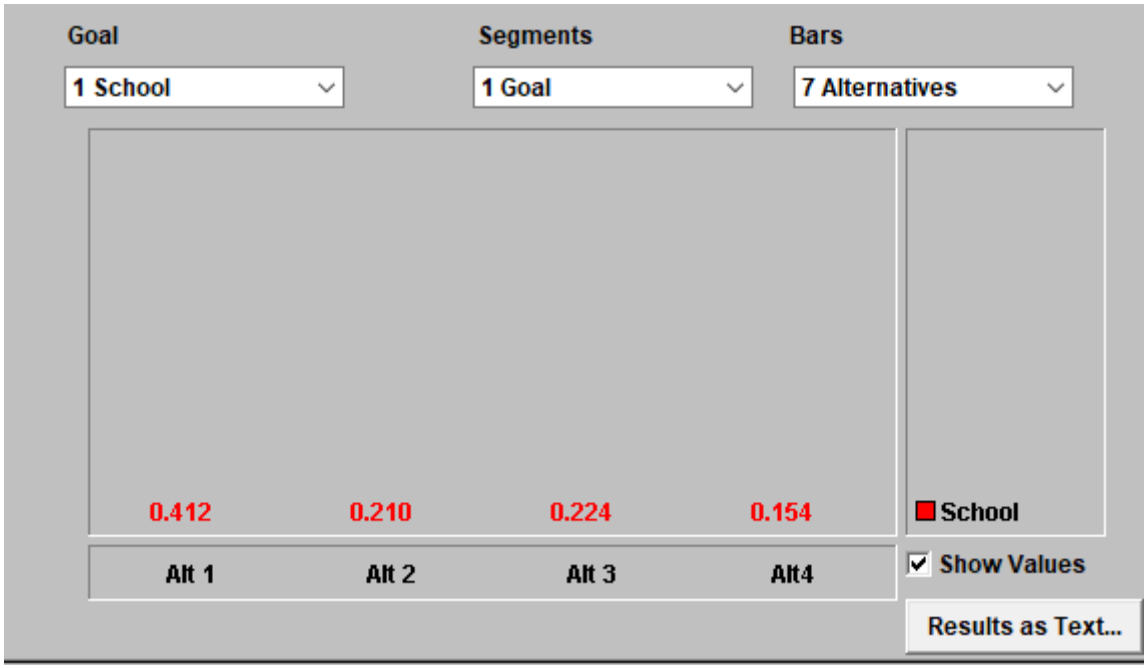
Selles peatükis esitatakse ülevaade esmalt AHP mudeli lõppkaaludest punkthinnangutena rakenduses ning võrdlus olemasoleva Web-Hipre rakenduse ja sealsete lõppkaaludega. Pärast võrdlust vaadatakse kuidas tundlikkuse analüüs mõjutab lõpptulemusi ning selle pealt tehtavaid järeldusi. Arvutuste tegemise jaoks loodi Lisa 3-s esitatud AHP mudel nii XMCDA standardi põhise skeemina uue rakenduse jaoks kui ka Web-Hipre rakenduses, et kontrollida tulemeid (AHP mudeli lõppkaalusid punkthinnangutena).

Ühekordse arvutamise lõpptulemusena saadi rakenduse väljastatud tulemiks AHP klassikalise meetodi laadselt alternatiivide lõppkaalude vektor (vt. joonis 13), milles on esitatud alternatiivi ID ning sellele vastav punkthinnang. Tabelis on esitatud ümardamata kogu lõpphinnang.

Alternatiiv	**Punkthinnang**
Alternatiiv1	0.4118324127502424
Alternatiiv2	0.20959308943459218
Alternatiiv3	0.2244778052968966
Alternatiiv4	0.15409669251826874

Joonis 10. Rakenduse ühekordse läbiarvutamise lõpptulemused näite põhjal

Järgnevalt esitatakse Web-Hipre rakenduse poolt tagastatud tulemused (joonis 14).



Joonis 11. Web-Hipres arvutatud tulemused

Võrreldes esimesi tulemusi Web-Hipre rakenduse poolt tagastatavate tulemitega sama mudeli ja samade hinnangutega (vt. Lisa 3 ja Lisa 6) on näha, et mõlema lõpptulemused on peaaegu kattuvad. Suurimaks erinevuseks on arvude ümardusaste, Web-Hipre esitab kuni 3 kohta pärast koma, kuid magistritöö rakenduse puhul on näidatud kogu arvu täpsus.

Järgnevalt esitatakse tundlikkuse analüüsi tulemused sama meetodi ja kaalude põhjal (joonis 15). Esimeste tulemuste puhul tehti simulatsioon läbi 10000 korda.

```
{
"a02":
       [{"a04":0.64499}],
"a01":
       [{"a02":0.82319},{"a04":0.9031},{"a03":0.80494}],
"a03":
       [{"a02":0.52718},{"a04":0.67003}]
}
```

Joonis 12. Simulatsiooni poolt tagastatud tulemused 10000 iteratsiooni korral

Tulemustest järeldub, et alternatiiv 4, mis sai kõige halvema tulemuse ei ole kunagi suurema kindlushinnanguga kui 50% teistest parem ja seetõttu see lõpptulemustes

põhielemendina ei kajastu. Samas, alternatiiv 2, mis oli teisena kõige nõrgema tulemusega, on parem kui alternatiiv 4 65% juhtudest, mis näitab, et 35% juhtudest võib võrdluses alternatiiv 2-ga alternatiiv 4 siiski olla parem valik Sama põhimõtet järgivad rakenduse väljundis ka kõik teised võrdlused ning alternatiiv 1 mis oli selgelt kõige tugevam valik ühekordsetes hinnangutes on üle 90% kindlusega parem valik kui alternatiiv 4.

Peale eelneva on näha ka, et alternatiiv 3-e hinnang alternatiiv 2 suhtes on 0,52718, mis tähendab, et umbes 52,7% juhtudest on esimene parem valik teise suhtes ehk, et need on praktiliselt võrdsed. See kajastub ka nende sarnases hinnangus alternatiiv 4-ja ning nende lähedases punkthinnangus (vastavalt 0,209 ja 0,225).

Antud juhul ei vahetunud alternatiivide järjestus võrreldes ühekordsete punkthinnangutega, kuid väga lähedaste hinnangute korral on see võimalik.

Et vaadelda kas ja kuidas iteratsioonide arv mõjutab lõpptulemusi tehti sama protsess läbi ka 100 ja 1000 iteratsioonina. Tulemused on esitatud järgnevates tabelites.

Saja iteratsiooni puhul tagastatud tulemused on esitatud Joonisel 16.


```
{
"a02":
       [{"a04":0.62251}],
"a01":
       [{"a02":0.84413},{"a04":0.90719},{"a03":0.80693}],
"a03":
       [{"a02":0.55762},{"a04":0.67617}]
}
```


Joonis 13. Simulatsiooni poolt tagastatud tulemused 100 iteratsiooni korral

Tuhande iteratsiooni puhul tagastatud tulemused on esitatud Joonisel 17.


```
{
"a02":
       [{"a04":0.64312}],
"a01":
       [{"a02":0.8281},{"a04":0.9055},{"a03":0.80543}],
"a03":
       [{"a02":0.53408},{"a04":0.67449}]
}
```


Joonis 14. Simulatsiooni poolt tagastatud tulemused 1000 iteratsiooni korral

Tulemuste võrdluses selgub, et kuigi antud mudeli puhul näitab juba 100 iteratsiooniga simulatsioon küllaltki ligilähedast tulemust võrreldes algselt esitatud 10k iteratsiooni läbi teinud lõpptulemusega, siis iteratsioonide suurendamisel hakkas lõpptulemus koonduma ühte punkti ning annab täpsema tulemuse. Võrdlustest saab järeldada, et lihtsamate mudelite puhul mille alternatiivide järjestused on kindlamad piisaks madalamast simulatsiooni iteratsioonide arvust, et küllaltki täpsed tulemused saada, kuid keerukamate mudelite puhul, milles alternatiivide järjestus on ebakindlam võib olla vajalik tõsta simulatsiooni iteratsioonide arvu. Monte Carlo esialgsete tulemuste alamhulkade pealt saaks automaatselt hinnata kui palju täiendavaid MC iteratsioone on vaja teha. Seda antud lõputöös ei ole tehtud ja see jääb võimalike edasiste arenduste valdkonda.

4.3 Alternatiivsed meetodid ja tehnoloogiad realisatsiooni koostamiseks

Töö üheks algseks eesmärgiks oli välja selgitada millised veebiteenuste standardid võimaldaksid kirjeldada AHP struktuuriga teenust. Algseteks uurimisobjektideks olid peale XMCDA standardi veel DMN [31] ning PMML [32] standardid. Siiski nende standardite hindamisel selgus, et kumbki neist ei ole senistes versioonides mõeldud MCDA probleemide lahendamiseks ning suure tõenäosusega ei ole seda ka võimalik nende abil toetada. DMN standardi lubavaks küljeks oli üldise objektide vahelise struktuuri loomise võimaldamine, kuid kogu protsess ise põhines otsustustabelitel ning puudusid võrdlusmaatriksid. Teiseks peamiseks uurimisaluseks oli PMML standard, mida kaaluti kuna see toetab palju erinevaid otsustusmudeleid, s.h. ka puustruktuuriga mudeleid ja närvivõrke, kuid uurimise käigus oli kirjeldatud, et standard on mõeldud andmekaeve probleemide lahendamiseks ning seda ei oleks saanud kerge vaevaga AHP mudelite jaoks kohaldada. Seega XMCDA standard, mis on konkreetselt suunatud MCDA probleemide lahendamiseks, osutus valituks, et lahendada töös seatud eesmärgid.

Võrreldes masinõppe meetoditega erineb AHP ja üleüldiselt MCDA algoritmid peamiselt sellepoolest, et AHP puhul ei ole olemas suurt andmehulka varasemalt lahendatud mudelitest ja probleemidest, mida saaks analüüsida masinõppe algoritmide poolt, et siis selle pealt parimat otsust vastu võtta. Ehk AHP meetodi puhul on tegemist ühe otsustajaga või otsustajate grupiga, kelle subjektiivsete hinnangute pealt on vajalik arvutada võimalikult objektiivne lõpptulem, mille jaoks luuakse uus AHP mudel ja sisestatakse

uued hinnangud. Samas masinõppe meetodid vajavad suurel hulgal varasemaid andmeid, et treenida mudelit ja siis hiljem seda kasutada.

Üheks alternatiiviks AHP mudelile ja selle tundlikkuse analüüsile on pakutud hägusa (ingl.k *fuzzy*) AHP mudelit, mis siiski hinnanguliselt ei oma sarnast kasutatavust klassikalisele AHP-le. Lühiülevaade hägusa AHP meetodi võrdlusest klassikalise AHP meetodiga on esitatud peatükis 2.5.1. Artikli [33] tehtud uurimuses on mainitud, et kuigi hägusat AHP-d on soovitatud varasemalt mitmetes teadusartiklites, siis ei ole esitatud empiirilist tõendust, et sellel meetodil on eeliseid klassikalise AHP ees. Veelgi enam, samas artiklis [33] esitatakse omapoolne töö nende kahe mudeli võrdluse kohta ning antakse hinnang, et hägusal AHP-l ei olnud olulisi eeliseid klassikalise mudeli ees juhul kui paarikaupa võrdlused olid antud piisavalt täpselt. Arvestades hägusate mudelite keerukust kasutamisel võib see olla ka edaspidi põhjuseks miks klassikaline AHP jääb paremaks valikuks nende vahel.

5 Realisatsiooni järeldused

Peatükis 4.2. *Üksiku iteratsiooni ja tundlikkuse analüüsi kaudu saadud tulemuste võrdlus ja hinnang* on esitatud ülevaade töö käigus loodud XMCDA standardi põhise rakenduse poolt tagastatud tulemustest, nende tulemuste võrdlusest ühe teise rakenduse tulemustega ning tehtud tundlikkuse analüüsi mõjudest lõpptulemustele. Kõik tulemused arvutati ühe ja sama AHP mudeli pealt, mis loodi näidiseks magistritöö jaoks.

Mõlema rakenduse poolt tagastatud tulemuste võrdluses selgub, et uue rakenduse tulemused on väga ligilähedased nendele hinnangutele mida kasutati valideerimiseks ehk üldine arvutuskäik on mõlemal sarnane ning uue rakenduse ühekordse läbiarvutamise lõpptulemused on sobilikud ning edaspidi kasutatavad, et sooritada nende peal tundlikkuse analüüsi.

Tundlikkuse analüüs teostati Monte Carlo simulatsiooni abil genereerides iga iteratsiooni korral normaaljaotusest juhuarvud, mille põhjal uued lõpphinnangud arvutada. Tundlikkuse analüüsi lõpptulemused näitasid, et näitemudeli ning selle hinnangute korral ei muutunud alternatiivide lõppjärjestus pärast tundlikkuse analüüsi sooritamist, kuid paranes lõpptulemuste loetavus, s.t. punkthinnangute asemel esitati kasutajale normaaljaotuse põhised statistilised tulemused, mille pealt on lihtsam järeldusi teha. Tagastades lõpptulemustena kõigi alternatiividena omavahelise paremuse ning kindlushinnangud oli võimalik hinnata kõigi alternatiivide järjestuse kindlusastet.

Lisaks, uuriti kuidas Monte Carlo simulatsiooni korral kasutaja poolt valitud iteratsioonide arv mõjutab lõpptulemusi. Selgus, et kuigi juba väiksema arvu iteratsioonide korral (näiteks: 100) võivad tulemused olla sarnased suurema iteratsiooni korral saadud tulemustele, siis iteratsiooni arvu suurendades (näiteks: 1000-10000) toimub koondumine ja lõpphinnangud paranevad. See tähendab, et väiksemad iteratsiooni arvud on teatud määral kasutatavad ja paraneb ka rakenduse jõudlus, kuid keerukamate mudelite ja selliste alternatiivide korral mis on väga ligilähedased üksteisele ning üleüldiselt täpsete tulemuste saamiseks on iteratsioonide arvu suurendamine eelistatud.

Lõppkasutaja seisukohalt kadus uue rakenduse puhul vajadus selle järgi, et kasutaja peaks manuaalselt hakkama tundlikkuse analüüsi tegema, mis tähendab, et kasutajal ei ole vajalik omada teadmisi sellest valdkonnast, et siiski saada parandatu tulemused rakendust

kasutades. Hinnanguliselt paraneb lõppkasutaja jaoks märgatavalt ka aeg, kui ta oleks käsitsi tundlikkuse analüüsi sooritama hakanud kuna eemaldatakse vajadus arvutuste läbiviimiseks ja muudatuste analüüsimiseks.

Probleemseks kohaks kerkis vajadus käsitsi luua AHP mudel vastavalt XML standardile kuna lõppkasutaja ei pruugi teada standardi piiranguid ning üleüldiselt on mitmete erinevate standardi failide loomine aeganõudev ja mitte kasutajasõbralik.

Rakendus AHP mudeli arvutuste jaoks loodi XMCDA veebiteenuste standardi põhjal, kuid peale selle uuriti algselt ka teisi võimalusi ja standardeid. Peatükis 4.3. tehti alternatiivsetest võimalustest ülevaade, milles selgitati, et valitud standard oli optimaalseim lahendus magistritöös esitatud probleemi lahendamiseks.

6 Edasiarendamise võimalused

Selles peatükis antakse ülevaade kuidas loodud rakendust saaks täiustada ning millised probleemid võivad praegu veel esineda.

Üheks suuremaks kitsaskohaks mida on mainitud ka varasemas peatükis osutus XMCDA standardi põhise AHP mudeli loomise keerukus lõppkasutaja jaoks. Kuna kogu rakenduse poolt kasutatav AHP mudel tuleb kirjeldada XML failidena, siis see tähendab, et kuigi see on lihtsasti kasutatav veebiteenusena ning juhul kui on juba loodud failid olemas, siis kasutaja peab väga detailselt teadma XMCDA standardi nõudeid, et ise uut AHP mudelit luua. Kuigi on olemas abistavad skeemi valideerijad, siis et see oleks lihtsasti kasutatav lõppkasutaja jaoks oleks vaja luua graafiline kasutajaliides, mille põhjal saaks kõik mudeli seosed ja paarikaupa võrdlused ära kirjeldada ning mille põhjal luuakse ka vajalikud failid mudeli sisendiks. Selline lahendus võiks olla ühenduses XMCDA veebiteenusega.

Teiseks edasiarenduseks võiks olla ANP mudelite toetamine. Kuna XMCDA standard toetab AHP-d ning võimaldab luua hierarhilisi seoseid elementide vahel, siis peaks olema toetatud ka veidi keerulisem võrgustruktuuriga ANP meetod, mis on samuti MCDA meetodite alla kuuluv alternatiivide valiku meetod.

Lisaks, täiendavaks edasiarenduseks võiks olla grupiotsustuste võimaluse lisamise rakendusele, mis vajaks olemasoleva lahenduse täiendamist sellisel viisil, et tuvastada erinevate otsustajate poolt antud hinnangud elementidele.

Kokkuvõte

Magistritöö „XMCDA 2.2.2 standardi põhise AHP rakenduse loomine ja tulemuste tundlikkuse analüüs“ eesmärgiks oli klassikalise AHP mudeli tulemusi tagastava rakenduse loomine ning sooritada selle poolt tagastatavate tulemuste peal automaatselt tundlikkuse analüüsi, et parandada klassikalise mudeli tulemuste kindlushinnanguid ja vähendada üleüldist ebakindlust mudelis. Automaatselt tehtava tundlikkuse analüüsi eesmärgiks oli eemaldada vajadus selle järgi, et kasutajal oleks põhjalikud teadmised tundlikkuse analüüsi sooritamisest. Uuriti nii MCDA meetodite, AHP mudeli kui ka tundlikkuse analüüsi teooriaid. Et luua rakendust oli üheks eesmärgiks uurida erinevaid veebiteenuste standardeid, mis seda võimaldaksid, et hiljem oleks võimalik rakendus teha kättesaadavaks veebiteenusena. Töö tulemuste hindamiseks loodi uus Java-põhine rakendus ja viidi arvutused läbi loodud näidismudeli peal. Algseid tulemusi võrreldi valideerimiseks teise AHP rakenduse poolt tagastatud tulemustega. Tundlikkuse analüüsi tulemusi võrreldi algsete tulemustega ning võrreldi nende erinevusi ning kuidas see lõppkasutajat mõjutab.

Magistritöö käigus selgus, et XMCDA andmestandard võimaldab luua kõiki vajalikke seoseid, et teha veebiteenuseid mis pakuvad AHP mudelite arvutuste tegemist. Leiti, et tundlikkuse analüüsi tulemused parandasid üldist hinnangute tõlgendamise lihtsust ja samas andsid selgema ülevaate lõpptulemuste kindlusastmest üksteise suhtes.

Tööd oleks vajalik edasi arendada kasutajamugavuse parandamise seisukohalt, luues graafilise kasutajaliidese mis oleks ühenduses veebiteenusega ja võimaldaks läbi selle kõiki standardi põhiseid vajalikke faile luua. Samuti saaks töös edasi uurida ANP võrkmudeli rakendamist ja arvutus meetodite lisamist.

Töö üldine eesmärk saavutati, loodi standardipõhine rakendus, arvutuskäik oli võimaldatud ning tulemused valideeriti varasemate rakenduste tulemuste põhjal. Tundlikkuse analüüsi sooritamine oli võimaldatud ning see ei vajanud põhjalikke teadmisi lõppkasutajalt. Samuti selgus kui palju ja kuidas Monte Carlo simulatsiooni põhine tundlikkuse analüüs mõjutab AHP mudeli poolt tagastatud tulemusi.

Summary

The purpose of this Document „ The Development of an AHP Sensitivity Analysis Application based on the XMCDA 2.2.2 Standard“, was to create a XMCDA web-service standard based application that calculates the results of an AHP model and then automatically perform sensitivity analysis on those results in order to estimate uncertainty of the model and improve confidence rating of returned values by returning normal distribution based confidence ratings instead of classical results. At the same time one of the goals was to simplify the process of sensitivity analysis on AHP models by doing it automatically through Monte Carlo simulation and removing the need from end-user to have knowledge of that field, yet still get improved end results. In order to solve the goals set for this thesis theories of MCDA methods in general, AHP method and AHP sensitivity analysis methodology had to be investigated. For the created application different web-service standards were examined and compared in order to find one that could support necessary functions. In the end a Java based application was created that performs the necessary Monte Carlo calculations and an example AHP model was created in order to validate the results. An overview of differences between classical AHP results and new Monte Carlo results that went through sensitivity analysis was done.

During the thesis work it was discovered that XMCDA standard is the most optimal one for solving MCDA algorithm related problems that includes the AHP model. It was concluded that the new returned results made it easier to interpret the final values and gave a better overview of the confidence levels of different alternatives positions relative to each other. For further improvements it is recommended to concentrate on the usability of application regarding end-user side as a graphical user interface could improve the process of creating a new AHP model which could be connected to the web-service side. Additionally, it was noted that ANP model could also be supported by the standard, but it would require creating an example model and adding calculation functions.

Overall, the objectives of the work were accomplished, a standard based application was created, which was able to calculate both the classical AHP model results and also perform sensitivity analysis on it, which didn’t require knowledge of the field from end-user. Additionally, it was identified how much such kind of simulation based sensitivity analysis affects the end results of an AHP model.

Kasutatud kirjandus

[1] „A Survey on Multi Criteria Decision Making Methods and Its Applications,“ Science and Education Publishing, 2013. [Võrgumaterjal]. Available: http://pubs.sciepub.com/ajis/1/1/5/. [Kasutatud 21 April 2018].

[2] Wikipedia, „Analytic hierarchy process – leader example,“ [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Analytic_hierarchy_process_%E2%80%93_leader_example. [Kasutatud 23 April 2018].

[3] „Analytic Hierarchy Process (What is AHP?),“ [Võrgumaterjal]. Available: web.cjcu.edu.tw/~lcc/Courses/TUTORIAL/AHP%20Tutorial.doc. [Kasutatud 21 April 2018].

[4] A. Saltelli, „Sensitivity Analysis for Importance Assessment,“ *Risk Analysis*, kd. 22, nr 3, 2002.

[5] D. J. Pannell, „Sensitivity Analysis of Normative Economic Models: Theoretical Framework and Practical Strategies,“ %1 *Australian Agricultural and Resource Economics Society*, 1996.

[6] T. Veskioja, „Subjektiivsetest hinnangutest objektiivsete tulemusteni - Näidisprojekt,“ 2006.

[7] L. Zhu, A. Aurum, I. Gorton ja R. Jeffery, „Tradeoff and Sensitivity Analysis in Software Architecture Evaluation Using Analytic Hierarchy Process,“ [Võrgumaterjal]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.7019&rep=rep1&type=pdf. [Kasutatud 05 May 2018].

[8] V. Sedašev, „GPS seadmete võrdlemise teenus - Bakalaureusetöö,“ 2009. [Võrgumaterjal]. Available: http://maurus.ttu.ee/ained/IDN5120/doc/23/Loputoo_Vladimir_Sedashev.pdf. [Kasutatud 06 May 2018].

[9] „Monte Carlo Simulation,“ Palisade, [Võrgumaterjal]. Available: http://www.palisade.com/risk/monte_carlo_simulation.asp. [Kasutatud 23 April 2018].

[10] Decision Deck, „How to create an XMCDA web service-able program?,“ Decision Deck, [Võrgumaterjal]. Available: https://www.decision-deck.org/ws/howto.createAWebService.html. [Kasutatud 05 May 2018].

[11] A. Kitsik, „Improving the sensitivity analysis of the Analytic Hierarchy Process,“ 2007. [Võrgumaterjal]. Available: http://maurus.ttu.ee/ained/IDN5120/doc/16/Ahti_Kitsik_magt66_2007.pdf. [Kasutatud 05 May 2018].

[12] N. Yaraghi, P. Tabesh, P. Guan ja J. Zhuang, „Comparison of AHP and Monte Carlo AHP Under Different Levels of Uncertainty,“ *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT*, kd. 62, nr 1, 2015.

[13] J. R. Banzon, L. R. Bacudio ja M. A. B. Promentilla, „Application of Monte Carlo Analytic Hierarchical Process (MCAHP) in the Prioritization of Theme Park Service Quality Elements,“ *International Conference on Industrial Engineering and Operations Management*, 2016.

[14] L. Jing, B. Chen, B. Zhang, P. Li ja J. Zheng, „Monte Carlo Simulation–Aided Analytic Hierarchy Process Approach: Case Study of Assessing Preferred Non-Point-Source Pollution Control Best Management Practices,“ 2013.

[15] P. Meyer ja S. Bigaret, „XMCDA: An XML encoding of MCDA data,“ 2009.

[16] W. Wei, *Enhancing PriEsT with Group Decision Making,* 2013.

[17] A. ÖZDAĞOĞLU ja G. ÖZDAĞOĞLU, *Comparison of AHP and fuzzy AHP for the multicriteria decision making process with linguistic evaluations,* 2007.

[18] S. AYDIN ja C. KAHRAMAN, *A Modified Fuzzy Analytic Hierarchy Process Based Multicriteria Decision making Methodology for Assessing E-commerce Website Quality: A Case Study in Turkey,* 2011.

[19] „XMCDA,“ Decision Deck, [Võrgumaterjal]. Available: https://www.decision-deck.org/xmcda/. [Kasutatud 21 April 2018].

[20] Decision Deck, „List of available XMCDA web services,“ [Võrgumaterjal]. Available: https://www.decision-deck.org/ws/webServices.html. [Kasutatud 05 May 2018].

[21] „JAMA : A Java Matrix Package,“ [Võrgumaterjal]. Available: https://math.nist.gov/javanumerics/jama/. [Kasutatud 21 April 2018].

[22] „Apache Commons Math,“ Apache, [Võrgumaterjal]. Available: http://commons.apache.org/proper/commons-math/. [Kasutatud 21 April 2018].

[23] „JDistlib—Java Statistical Distribution Library,“ [Võrgumaterjal]. Available: http://jdistlib.sourceforge.net/. [Kasutatud 21 April 2018].

[24] „Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling Tests,“ [Võrgumaterjal]. Available: https://www.researchgate.net/publication/267205556_Power_Comparisons_of_Shapiro-Wilk_Kolmogorov-Smirnov_Lilliefors_and_Anderson-Darling_Tests. [Kasutatud 21 April 2018].

[25] „Normality Tests for Statistical Analysis: A Guide for Non-Statisticians,“ 20 April 2012. [Võrgumaterjal]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3693611/. [Kasutatud 21 April 2018].

[26] „WeightedSum,“ [Võrgumaterjal]. Available: https://www.diviz.org/pd-weightedSum-PyXMCDA.html. [Kasutatud 25 April 2018].

[27] T. Veskioja, „Subjektiivsetest hinnangutest objektiivsete tulemusteni - Näidisprojekt,“ Tallinn, 2012.

[28] „W values from the Shapiro-Wilk test visualized with different datasets,“ [Võrgumaterjal]. Available: http://emilkirkegaard.dk/en/?p=4452. [Kasutatud 21 April 2018].

[29] „Standard Score,“ Laerd Statistics, [Võrgumaterjal]. Available: https://statistics.laerd.com/statistical-guides/standard-score.php. [Kasutatud 21 April 2018].

[30] „68 95 99.7 Rule in Statistics,“ [Võrgumaterjal]. Available: http://www.statisticshowto.com/68-95-99-7-rule/. [Kasutatud 25 April 2018].

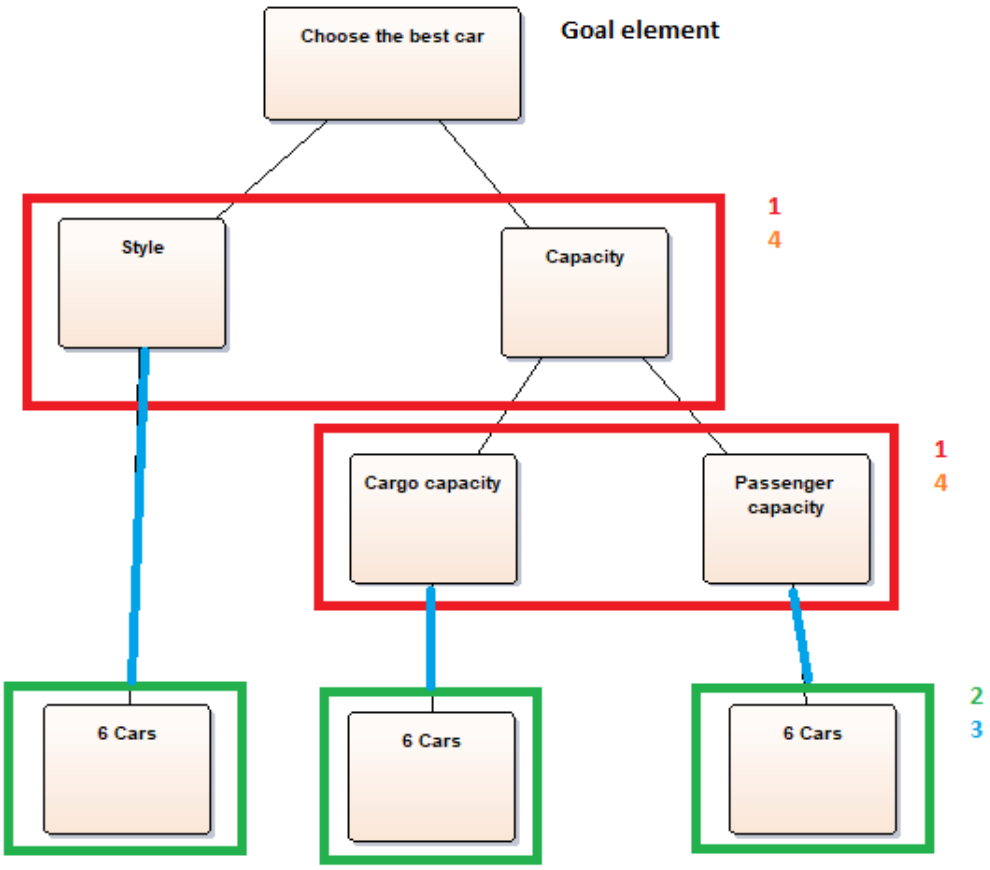
[31] Object Management Group, „About-DMN,“ [Võrgumaterjal]. Available: https://www.omg.org/spec/DMN/About-DMN/. [Kasutatud 04 May 2018].

[32] Data Mining Group, „PMML Version 4.3,“ [Võrgumaterjal]. Available: http://dmg.org/pmml/pmml-v4-3.html. [Kasutatud 04 May 2018].

[33] D. CHATTERJEE ja B. MUKHERJEE, „A Study on The Comparison of AHP And Fuzzy AHP Evaluations of Private Technical Institutions in India,“ *(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH,* kd. 1, nr 4, 2013.

Lisa 1 – XMCDA XML tüüpide omavaheline sõltuvus rakenduses

Joonisel on esitatud lihtsustatud ülevaade sellest kuidas standardipõhised XML tüübid on omavahel seotud, et luua programmisiseselt AHP struktuur.



Joonis 15. XML tüüpide omavaheliste seoste ülevaade

1. CriteriaSets: Sisaldab 1-mitmele seoseid ülema alternatiivi ja selle alla kuuluvate alternatiivide grupi vahel

2. AlternativesSets: Sisaldab ühte kogumit kõikide alternatiivide ID-dega

3. AlternativesComparisons: Viide alternatiivide kogumi (AlternativesSets) ja sellega seoses oleva kriteeriumi ID vahel. Sisaldab kõki konkreetse kriteeriumi põhiseid paarikaupa võrdlusi alternatiivide vahel.

4. CriteriaComparisons: Sisaldab kriteeriumite vahelisi paarikaupa võrdluseid. Ainult ühte kogumisse kuuluvaid kriteeriumeid saab omavahel võrrelda.

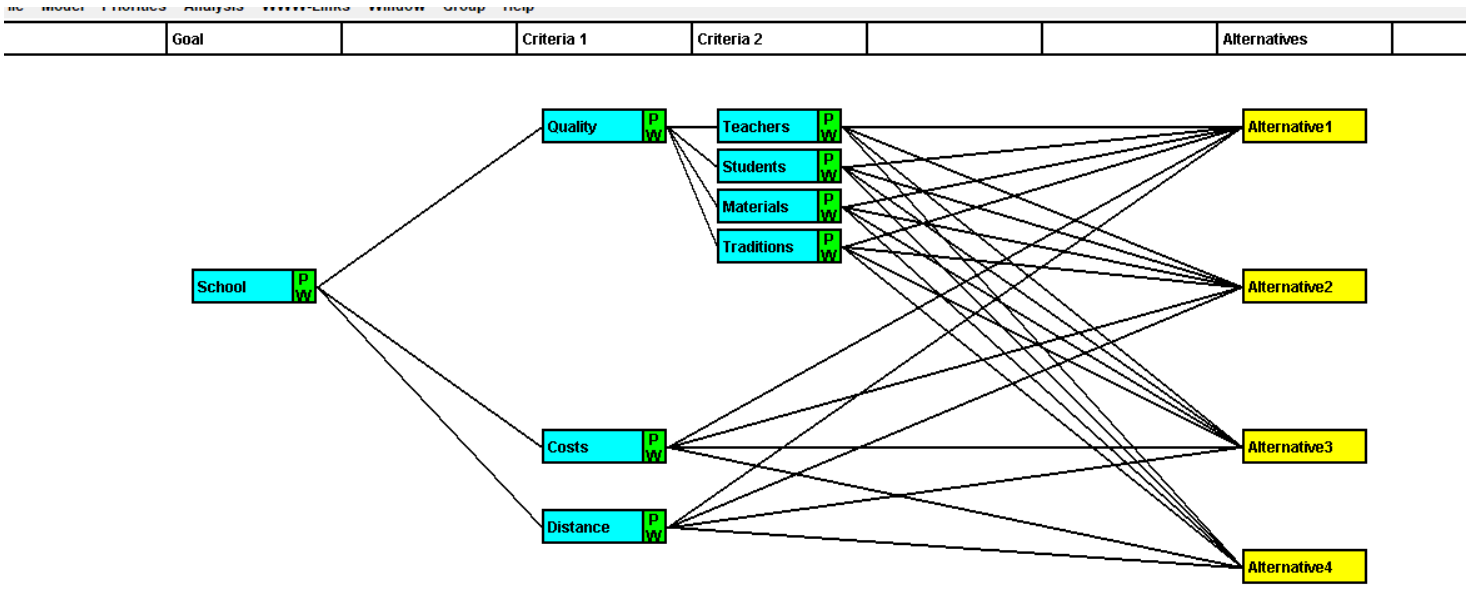
5. Hierarchy: Loob elementide vahelised viited, ühendades kogumid nende ülemate elementidega – kriteeriumite kogum -> kriteeriumi ID, alternatiivide kogum -> kriteeriumi ID

Lisa 2 – Saaty skaala väärtused

Tabel 12. Saaty skaala hinnangud

Saaty arvskaala	Saaty sõnaskaala
9	Ekstreemselt parem
7	Väga tugevalt parem
5	Oluliselt parem
3	Mõõdukalt parem
1	Võrdne
1/3	Mõõdukalt halvem
1/5	Oluliselt halvem
1/7	Väga tugevalt halvem
1/9	Ekstreemselt halvem

Lisa 3 – Töös kasutatav näidismudel tulemuste hindamiseks



Joonis 16. Töös kasutatud AHP näidismudel Web-Hipres

Iga mudeli elemendi sisse kuulub võrdlusmaatriks, milles on paarikaupa võrdlused alamelementide kohta.

	A	B	C	D
A Teachers	1.0	5.0	3.0	7.0
B Students	0.2	1.0	0.33	1.0
C Materials	0.33	3.0	1.0	3.0
D Traditions	0.14	1.0	0.33	1.0

Joonis 17. Võrdlusmaatriksi näide Web-Hipres

Lisa 4 – Monte Carlo simulatsiooni tulemusel saadud lõppkaalude kogum

Alternatiivi ID -> lõppkaal (punkthinnang, milline alternatiiv on parim)

- 0 = {HashMap@2254} size = 4
- 1 = {HashMap@2255} size = 4
- 2 = {HashMap@2256} size = 4
- 3 = {HashMap@2257} size = 4
- 4 = {HashMap@2258} size = 4
- 5 = {HashMap@2259} size = 4
  - 0 = {HashMap$Node@2440} "a02" -> "0.20160388228507775"
  - 1 = {HashMap$Node@2441} "a01" -> "0.4139718757477276"
  - 2 = {HashMap$Node@2442} "a04" -> "0.12182808609057036"
  - 3 = {HashMap$Node@2443} "a03" -> "0.2625961558766243"
- 6 = {HashMap@2260} size = 4
  - 0 = {HashMap$Node@2430} "a02" -> "0.2158469821199467"
  - 1 = {HashMap$Node@2431} "a01" -> "0.37474714748543836"
  - 2 = {HashMap$Node@2432} "a04" -> "0.15523947268803406"
  - 3 = {HashMap$Node@2433} "a03" -> "0.254166397706581"
- 7 = {HashMap@2261} size = 4
  - 0 = {HashMap$Node@2420} "a02" -> "0.21243985446154187"
  - 1 = {HashMap$Node@2421} "a01" -> "0.41485939988378334"
  - 2 = {HashMap$Node@2422} "a04" -> "0.1344997586898356"
  - 3 = {HashMap$Node@2423} "a03" -> "0.23820098696483907"
- 8 = {HashMap@2262} size = 4
  - 0 = {HashMap$Node@2410} "a02" -> "0.2373049820042431"
  - 1 = {HashMap$Node@2411} "a01" -> "0.38347258586292937"
  - 2 = {HashMap$Node@2412} "a04" -> "0.10441275826615468"
  - 3 = {HashMap$Node@2413} "a03" -> "0.2748096738666728"
- 9 = {HashMap@2263} size = 4
  - 0 = {HashMap$Node@2400} "a02" -> "0.21417801278474963"
  - 1 = {HashMap$Node@2401} "a01" -> "0.42305204013288145"
  - 2 = {HashMap$Node@2402} "a04" -> "0.17319399608059846"
  - 3 = {HashMap$Node@2403} "a03" -> "0.18957595100177044"
- 10 = {HashMap@2264} size = 4
  - 0 = {HashMap$Node@2390} "a02" -> "0.20157873138842847"
  - 1 = {HashMap$Node@2391} "a01" -> "0.4178872164482846"
  - 2 = {HashMap$Node@2392} "a04" -> "0.15085651169752143"
  - 3 = {HashMap$Node@2393} "a03" -> "0.22967754046576566"
- 11 = {HashMap@2265} size = 4
  - 0 = {HashMap$Node@2380} "a02" -> "0.21797005361227348"
  - 1 = {HashMap$Node@2381} "a01" -> "0.3406130935783715"
  - 2 = {HashMap$Node@2382} "a04" -> "0.23369170400067163"
  - 3 = {HashMap$Node@2383} "a03" -> "0.20772514880868348"
- 12 = {HashMap@2266} size = 4
  - 0 = {HashMap$Node@2370} "a02" -> "0.19132485850434816"
  - 1 = {HashMap$Node@2371} "a01" -> "0.4611306684583506"
  - 2 = {HashMap$Node@2372} "a04" -> "0.17830666829804914"
  - 3 = {HashMap$Node@2373} "a03" -> "0.16923780473925215"

Lisa 5 – Dokumendis mainitud peamised koodiosad

Sisend objekti (Input) salvestatavad andmed, mida kasutatakse AHP mudeli arvutuste tegemisel:

```
public class Inputs {

  /**
   * The list of alternatives ids
   */
  Map<String, List<AlternativesComparisons.Pairs.Pair>>
alternativesPairwiseComparisons;

  /**
   * The list of criterion ids
   */
  Map<String, List<CriteriaComparisons.Pairs.Pair>>
criteriaPairwiseComparisons;

  /**
   * Criterion and it's subcriteria. First element is parent, second element
shows what is under it.
   */
  Map<String, String> criterionSubCriteriaSetReference;

  /**
   * Which criterions belong to the same set. They are either subcriteria
under the same criterion or the first level after goal element.
   */
  Map<String, List<String>> criteriaSetCriterionElements;
```

Joonis 18. Andmestruktuuride klass eraldatud sisend andmete jaoks

Monte Carlo meetodi korral kutsutav peamine klass:

```
private List<Map<String, Double>> doCalculationsUsingMonteCarloMethod(int
count) {
  List<Map<String, Double>> finalWeights = new ArrayList<>();

  for(int i = 0; i < count; i++) {
    // Get new weights by applying Monte Carlo calculations through error
multiplier
    Map<String, Map<String, Double>> criteriaLocalWeights =
getCriteriaLocalWeightsUsingMC();
    Map<String, Map<String, Double>> alternativesLocalWeights =
getAlternativesLocalWeightsUsingMC();
    Map<String, Double> criteriaSetCriterionGlobalScores =
getCriterionGlobalScores(criteriaLocalWeights);
    Map<String, Double> criterionAlternativesGlobalScores =
getAlternativesGlobalScores(criteriaSetCriterionGlobalScores,
alternativesLocalWeights );

    finalWeights.add(criterionAlternativesGlobalScores);
  }

  return finalWeights;
}
```

Joonis 19. Monte Carlo arvutuste ülemklass

Maatriksi veakordajate leidmise funktsioon mida kasutatakse iga paarikaupa võrdluseid sisaldava maatriksi korral (kriteeriumid ja alternatiivid):

```
private double[][] getMatrixErrorRatios(double[][] pairwiseMatrix) {
  int elementCount = pairwiseMatrix.length;
  double[][] errorRatios = new double[elementCount][elementCount];

  for(int row = 0; row < elementCount; row++) {
    for(int column = row + 1; column < elementCount; column++) {
      double value = pairwiseMatrix[row][column];
      double rowProduct = getRowProduct(pairwiseMatrix, column);
      double columnProduct = getColumnProduct(pairwiseMatrix, row);
      double number = rowProduct * columnProduct * Math.pow(value,
elementCount);
      double powerValue = elementCount > 2 ? 1.0 / (elementCount - 2.0) : 1;
      double tripletConsistency = Math.pow(number, powerValue);
      double finalValue = 1 / tripletConsistency;

      errorRatios[row][column] = finalValue;
    }
  }

  return errorRatios;
}
```

Joonis 20. Maatriksi veakordajate leidmine Javas

Tundlikkusekordajate leidmise ja logaritmimise funktsioon:

```
private double[] getLogSensitivityMultipliers(int row, int column,
    List<Map<String, Double>> alternativeResults,
    Map<Integer, String> sequence) {

  double[] sensitivityMultipliers = new double[alternativeResults.size()];
  String element1 = sequence.get(row);
  String element2 = sequence.get(column);

  int count = 0;
  for(Map<String, Double> iteration: alternativeResults) {
    double div1 = iteration.get(element1) / (1 - iteration.get(element1));
    double div2 = iteration.get(element2) / (1 - iteration.get(element2));
    double multiplier = div1/div2;
    double logSensitivityMultiplier = Math.log(multiplier);
    sensitivityMultipliers[count] = logSensitivityMultiplier;
    count++;
  }

  return sensitivityMultipliers;
}
```

Joonis 21. Logaritmitud tundlikkusekordajate leidmise funktsioon Javas

Kumulatiivse tõenäosuse leidmine kasutades Apache Commons-math3 paketti:

```
private double calculateCumulativeProbabilty(double[] zScores, double point)
{
  StandardDeviation sd = new StandardDeviation();
  Mean mean = new Mean();
  double stDeviation = sd.evaluate(zScores);
  double meanValue = mean.evaluate(zScores);
  NormalDistribution normal = new NormalDistribution(meanValue, stDeviation);

  return Utils.round(normal.cumulativeProbability(point), 5);
}
```

Joonis 22. Kumulatiivse tõenäosuse leidmise funktsioon Javas

Lisa 6 – Näidismudeli paarikaupa võrdlused.

Lisa 3-s esitatud mudeli kõik paarikaupa hinnangud

Eesmärgi elemendi all olevad hinnangud

	A	B	C
A Quality	1.0	3.0	5.0
B Costs	0.33	1.0	2.0
C Distance	0.2	0.5	1.0

Kvaliteedi kriteeriumi hinnangud

	A	B	C	D
A Teachers	1.0	5.0	3.0	7.0
B Students	0.2	1.0	0.33	1.0
C Materials	0.33	3.0	1.0	3.0
D Traditions	0.14	1.0	0.33	1.0

Hinna kriteeriumi hinnangud

	A alternatiiv1	B alternatiiv2	C alternatiiv3	D alternatiiv4
A alternatiiv1	1.0	0.33	0.11	0.14
B alternatiiv2	3.0	1.0	0.2	0.33
C alternatiiv3	9.0	5.0	1.0	3.0
D alternatiiv4	7.0	3.0	0.33	1.0

Kauguse kriteeriumi hinnangud

	A alternatiiv1	B alternatiiv2	C alternatiiv3	D alternatiiv4
A alternatiiv1	1.0	0.33	0.2	1.0
B alternatiiv2	3.0	1.0	1.0	3.0
C alternatiiv3	5.0	1.0	1.0	5.0
D alternatiiv4	1.0	0.33	0.2	1.0

Õppejõudude kriteeriumi hinnangud

	A alternatiiv1	B alternatiiv2	C alternatiiv3	D alternatiiv4
A alternatiiv1	1.0	3.0	7.0	5.0
B alternatiiv2	0.33	1.0	5.0	3.0
C alternatiiv3	0.14	0.2	1.0	0.33
D alternatiiv4	0.2	0.33	3.0	1.0

Õpilaste kriteeriumi hinnangud

	A alternatiiv1	B alternatiiv2	C alternatiiv3	D alternatiiv4
A alternatiiv1	1.0	1.0	5.0	3.0
B alternatiiv2	1.0	1.0	3.0	1.0
C alternatiiv3	0.2	0.33	1.0	0.33
D alternatiiv4	0.33	1.0	3.0	1.0

Materjalide kriteeriumi hinnangud

	A alternatiiv1	B alternatiiv2	C alternatiiv3	D alternatiiv4
A alternatiiv1	1.0	3.0	9.0	5.0
B alternatiiv2	0.33	1.0	7.0	3.0
C alternatiiv3	0.11	0.14	1.0	0.33
D alternatiiv4	0.2	0.33	3.0	1.0

Traditsioonide kriteeriumi hinnangud

	A alternatiiv1	B alternatiiv2	C alternatiiv3	D alternatiiv4
A alternatiiv1	1.0	3.0	5.0	5.0
B alternatiiv2	0.33	1.0	3.0	3.0
C alternatiiv3	0.2	0.33	1.0	1.0
D alternatiiv4	0.2	0.33	1.0	1.0