

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Metody vícekriteriálního rozhodování a jejich aplikace
Nikola Jačková

Bakalářská práce
2021

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Nikola Jačková**
Osobní číslo: **I18131**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Téma práce: **Metody vícekritériálního rozhodování a jejich aplikace**
Zadávací katedra: **Katedra informačních technologií**

Zásady pro vypracování

Práce by měla v teoretické části popsat jednotlivé metody vícekritériálního rozhodování a hodnocení variant za jistoty, jednotlivé metody by měly být ilustrovány praktickými příklady. Jednotlivé rozhodovací metody by měly být implementovány ve vhodném programovacím jazyce a ilustrovány výsledky pro konkrétní rozhodovací situace a grafy. Dále by vícekritériální rozhodování mělo být aplikováno do studentem vybrané praktické oblasti – rozhodování v ekonomii, manažerské rozhodování, logistika atd. Práce lze rozšířit o rozhodování na neurčitosti, kde lze využít aparát fuzzy množin.

Rozsah pracovní zprávy: **40**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

HEBÁK, Petr. Vícerozměrné statistické metody. 2., přeprac. vyd. Praha: Infomatorium, 2007-, 3 sv. ISBN 978-80-7333-056-9. TALAŠOVÁ, Jana. Fuzzy metody vícekritériálního hodnocení a rozhodování. 1. vyd. Olomouc: Univerzita Palackého, 2003, 179 s. ISBN 80-244-0614-4.

Vedoucí bakalářské práce: **Mgr. Alena Pozdílková, Ph.D.**
Katedra matematiky a fyziky

Datum zadání bakalářské práce: **31. října 2020**
Termín odevzdání bakalářské práce: **14. května 2021**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

LS.

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 26. února 2021

Prohlašuji:

Tuto práci jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 12. 5. 2021

Nikola Jačková v.r.

PODĚKOVÁNÍ

Ráda bych zde poděkovala své vedoucí bakalářské práce Mgr. Pozdílkové Aleně Ph.D. za její rady, ochotu a trpělivost při zhotovování této práce.

ANOTACE

Bakalářská práce se zabývá metodami vícekriteriálního rozhodování, přičemž se věnuje případům, kdy je množina variant konečná a kdy se jedná o vícekriteriální hodnocení variant za jistoty. Jednotlivé metody určení vah kritérií a následné určení nejlepší varianty popisuje v teoretické části. Vybrané metody jsou implementovány v aplikaci vytvořené ve vyšším programovacím jazyce C#.

KLÍČOVÁ SLOVA

Vícekriteriální rozhodování, metody vícekriteriálního rozhodování, kritéria, hodnocení variant za jistoty

TITLE

Methods of multicriterial decision-making and their application.

ANNOTATION

The bachelor's thesis deals with the methods of multicriteria decision-making, especially in cases where the set of variants is finite and when it is a multicriteria evaluation of variants with certainty. It describes and demonstrates the determination of the weights of the criteria and individual methods for choosing the best variant. Selected methods are implemented in an application created in the higher-level programming language C#.

KEYWORDS

Multiple-criteria decision analysis, MCDA, multiple-criteria decision-making, MCDM, criteria

OBSAH

Seznam obrázků	8
Seznam tabulek	9
Seznam zkratk	10
Úvod	11
1 Vícekriteriální rozhodování	12
1.1 Rozhodovací proces	13
1.2 Rozdělení úloh vícekriteriálního rozhodování.....	14
2 Vícekriteriální hodnocení variant za jistoty	16
2.1 Metody stanovení vah kritérií	16
2.1.1 Metody přímého stanovení vah kritérií.....	16
2.1.2 Metody založené na párovém porovnání	17
2.2 Metody výběru kompromisních variant.....	19
2.2.1 Metody nevyžadující informaci	19
2.2.2 Metody s nominální informací.....	19
2.2.3 Metody s ordinální informací	20
2.2.4 Metody s kardinální informací.....	23
3 Praktická část	26
3.1 Implementace	26
3.1.1 Struktura aplikace	26
3.1.2 UseCase diagram	27
3.1.3 Diagramy tříd.....	28
3.1.4 Uživatelské prostředí	35
3.2 Vlastní výsledky	37
3.2.1 Lexikografická metoda	39
3.2.2 Metoda ORESTE	39
3.2.3 Metoda TOPSIS	41
3.2.4 Metoda váženého součtu.....	42
3.2.5 Porovnání výsledků metod.....	43
3.3 Shrnutí.....	45
Závěr	46
Použitá literatura	47
Přílohy	48

SEZNAM OBRÁZKŮ

Obrázek 1: Schéma postupu preferenční analýzy [2]	22
Obrázek 2: Hierarchická struktura se třemi úrovněmi [2]	24
Obrázek 3: Struktura aplikace Decision Support System	27
Obrázek 4: UseCase diagram aplikace	28
Obrázek 5: Diagram tříd ve složce <i>DecisionProject</i>	29
Obrázek 6: Diagram třídy <i>WeightProcessor</i>	29
Obrázek 7: Diagram třídy <i>DecideMethodUtility</i>	30
Obrázek 8: Diagram třídy <i>LogWriter</i>	30
Obrázek 9: Diagram tříd potomků <i>Exception</i>	30
Obrázek 10: Diagram třídy <i>LexicographicMethod</i>	31
Obrázek 11: Diagram třídy <i>OresteMethod</i>	31
Obrázek 12: Algoritmus pro ohodnocení variant, metoda ORESTE [9]	32
Obrázek 13: Algoritmus pro výpočet matice <i>D</i> , metoda ORESTE	32
Obrázek 14: Diagram třídy <i>WeightedSumApproachMethod</i>	33
Obrázek 15: Algoritmus pro výpočet matice <i>R</i> , metoda váženého součtu	33
Obrázek 16: Algoritmus pro získání hodnot funkcí užítku, metoda váženého součtu	33
Obrázek 17: Diagram třídy <i>TopsisMethod</i>	34
Obrázek 18: Algoritmus pro získání matice <i>W</i> , metoda TOPSIS	34
Obrázek 19: Algoritmus pro získání vzdáleností, metoda TOPSIS	34
Obrázek 20: Uživatelské prostředí po spuštění aplikace	35
Obrázek 21: Uživatelské prostředí aplikace s daty	36
Obrázek 22: Relativní index vzdáleností u metody TOPSIS	44
Obrázek 23: Užitek variant při použití metody WSA	45

SEZNAM TABULEK

Tabulka 1: Schéma Fullerova trojúhelníku.....	17
Tabulka 2: Saatyho doporučená stupnice	18
Tabulka 3: Údaje o vybraných mobilních telefonech	37
Tabulka 4: Ohodnocení kritérií.....	38
Tabulka 5: Normované váhy kritérií.....	38
Tabulka 6: Matice s převedenými hodnotami v minimalizačních kritériích	39
Tabulka 7: Ohodnocení kritérií pořadím	39
Tabulka 8: Matice vzdáleností D	40
Tabulka 9: Ohodnocená matice R s řádkovými součty.....	40
Tabulka 10: Normalizovaná vážená matice W	41
Tabulka 11: Ohodnocení variant vzdálenostmi	41
Tabulka 12: Standardizovaná matice R	42
Tabulka 13: Užitek jednotlivých variant	42
Tabulka 14: Porovnání pořadí variant podle lexikografické metody a metody ORESTE.....	43
Tabulka 15: Porovnání pořadí variant podle metody TOPSIS a metody váženého součtu	44

SEZNAM ZKRATEK

AHP	Analytic Hierarchy Process
DLL	Dynamic Link Library
OS	Operační systém
RAM	Random Access Memory
TOPSIS Solution	Technique for Order of Preference by Similarity to Ideal
WPF	Windows Presentation Foundation
WSA	Weighted Sum Approach

ÚVOD

Proces rozhodování hraje v lidském životě velkou roli. Denně člověk udělá několik rozhodnutí. Velká část těchto rozhodnutí je krátkodobá a nenáročná a člověk se pro ně rozhoduje dle vlastní intuice. Některá složitější a dlouhodobá rozhodnutí však vyžadují patřičné zamyšlení se nad problémem a nad možnými důsledky. Člověk by se měl rozhodovat racionálně a s ohledem na to, aby mu výsledek přinesl co největší užitek. Příkladem může být rozhodování o výběru vysoké školy, nákupu rodinného auta, výběru zaměstnání nebo výběru bankovního produktu. Při velkém počtu variant je pak složitější vybrat tu správnou. V těchto situacích nám může pomoci aplikování metod vícekriteriálního rozhodování.

Rozhodovací problémy, u kterých se výsledné rozhodnutí posuzuje podle více kritérií, lze tedy zobrazit do modelů vícekriteriálního rozhodování. Účelem modelů je většinou určení nejvýhodnější varianty, uspořádání množiny variant dle příznivosti nebo vyloučení nevýhodných řešení. Tyto modely jsou také základem pro tvorbu počítačových systémů, které zpracovávají rozhodovací proces a nabízí uživateli vhodnou variantu, tzv. systémy pro podporu rozhodování.

Bakalářská práce je rozdělena na část teoretickou a praktickou. Cílem teoretické části je seznámení se základními pojmy vícekriteriálního rozhodování a představení různých metod vícekriteriálního hodnocení variant za jistoty rozdělených podle typů. Teoretická část se také věnuje metodám určení vah kritérií, což je proces, který předchází samotnému aplikování vícekriteriálních metod. V praktické části jsou vybrané metody implementované v aplikaci programované ve vyšším programovacím jazyce. Součástí praktické části je popsání uživatelského prostředí aplikace a použitých technologií a aplikace modelového příkladu a následné porovnání výsledků použitých metod.

1 VÍCEKRITERIÁLNÍ ROZHODOVÁNÍ

Vícekriteriální rozhodování se zabývá výběrem jedné varianty z množiny přípustných variant. Rozhodováním se rozumí proces, ve kterém jedinec nebo kolektiv vybírá jedno nebo více řešení z množiny variant, jehož cílem je výběr nejvhodnější varianty. Množina může být vyjádřena explicitně konkrétním výčtem prvků (diskrétní množina) a jedná se o vícekriteriální hodnocení variant, nebo je množina variant popsána implicitně soustavou omezujících podmínek (spojitá množina), které musí přípustná varianta splňovat, a jedná se o vícekriteriální programování. Tato práce se zaměřuje na vícekriteriální hodnocení variant. [1]

Rozhodovatel, což je jedinec nebo skupina jedinců, která činí samotné rozhodnutí, by měl k rozhodování přistupovat objektivně. V některých případech je rozhodovatelem (analytikem) jiný člověk než zadavatel úlohy. Výhoda tohoto přístupu spočívá v menší zaujatosti ze strany analytika, na druhou stranu však analytik nezná všechny podrobnosti rozhodovací úlohy. [2]

Konečná množina přípustných variant obsahuje m počet variant, což je konkrétní rozhodovací možnost. Přípustnou variantou se rozumí řešení, které je realizovatelné a je logicky smysluplné. [2]

Varianta, která dosahuje ve všech kritériích nejlepších výsledků, se označuje jako ideální. Obvykle tato varianta v úlohách vícekriteriálního rozhodování neexistuje. Opakem ideální varianty je bazální varianta, která má ve všech kritériích nejnižší hodnocení. [1]

Podle definice dominovaná varianta označuje variantu v_a , kterou dominuje varianta v_b , jestliže platí, že v_b je hodnocena ve všech kritériích lépe nebo stejně jako v_a a zároveň existuje alespoň jedno kritérium, které má v_b ohodnoceno lépe než v_a . [2]

Každá z variant má n kritérií. Kritérium je definováno jako měřítko hodnocení variant. Pokud platí, že $n=1$, nejedná se o vícekriteriální rozhodování, nýbrž o jednokriteriální [3], které se v praxi příliš nevyskytuje. Jednotlivá kritéria se mohou dělit dle jejich povahy na [2]:

- maximalizační – nejvyšší hodnoty kritéria znamenají nejlepší hodnoty, např. výnosy z prodeje produktu;
- minimalizační – nejnižší hodnoty kritéria znamenají nejlepší hodnoty, např. náklady na výrobu produktu.

Je žádoucí, aby všech n kritérií bylo stejného typu. Nejčastěji se pracuje s kritérii maximalizačními. Pro převádění kritérií se používají dva způsoby. Při prvním způsobu se vynásobí všechny hodnoty převáděného kritéria hodnotou -1 . Nevýhoda této transformace

spočívá v interpretaci nového kritéria. Například kritérium „záporné náklady na výrobu produktu“, které vzniklo při převodu kritéria „náklady na výrobu produktu“, nemusí na první pohled dávat smysl. Při využití druhého způsobu se vypočte hodnota, která udává zlepšení oproti nejhorší hodnotě, tzn. že kritérium „náklady na výrobu produktu“ se po převodu změní na „úspory nákladů oproti nejnákladnější výrobě produktu“, nejnákladnější varianta bude ohodnocena hodnotou nula a ostatní varianty budou mít kladnou hodnotu. [2]

Dále se kritéria dají dělit podle kvantifikovatelnosti [2]:

- kvantitativní – vyjádřené objektivně měřitelnou hodnotou, např. cena výrobku;
- kvalitativní – vyjádřené subjektivním odhadem uživatele, např. celkové zhodnocení výrobku.

Preference kritérií hraje při řešení úloh vícekritériálního rozhodování významnou roli. Určuje, v jaké míře je jedno kritérium významnější oproti ostatním. Pro vyjádření preferencí se může využít

- aspirační úroveň kritérií – udává hodnotu, které musí kritérium dosáhnout;
- pořadí kritérií – seřazení kritérií od nejvýznamnějšího po nejméně významné;
- váhy kritérií – míra důležitosti ve srovnání s ostatními kritérii;
- způsob kompenzace kritériálních hodnot – vyrovnání špatných hodnot lepšími hodnotami. [2]

Z hodnocení variant podle jednotlivých kritérií se vytváří kritériální matice $Y = (y_{ij})$. Prvek y_{ij} udává hodnocení i -té varianty podle j -tého kritéria. [2]

1.1 Rozhodovací proces

Pokud jsou známy výsledky variant a stav, který nastane, jedná se o rozhodování za jistoty. Mohou ale nastat situace, kdy důsledky rozhodnutí nejsou známy s jistotou a závisí na různých stavech systému. V případě, že není známa pravděpodobnost výskytů těchto stavů, jedná se o rozhodování při neurčitosti. Jestliže je pravděpodobnost výskytu stavů známa, jedná se o rozhodování při riziku. [1]

Podle Fotra v [4] lze celkový rozhodovací proces podrobně rozdělit do několika fází:

1. určení rozhodovacích problémů – zahrnuje shromažďování, analýzu a vyhodnocení různých dat o dané oblasti a identifikaci potencionálních situací vyžadujících řešení;

2. analýza a formulace rozhodovacích problémů – obsahuje hlubší analýzu problému, stanovení základních prvků a cílů řešení, kterých může být definováno více a mohou být vyjádřeny číselně (kvantitativní cíle) nebo slovně (kvalitativní cíle) [5];
3. stanovení kritérií hodnocení variant – definování kritérií, podle kterých se budou varianty hodnotit;
4. tvorba variant řešení rozhodovacího problému – určení množiny variant, které jsou pro daný problém přípustné;
5. stanovení důsledků variant rozhodování – vymezení potencionálních důsledků jednotlivých variant;
6. hodnocení důsledků a výběr varianty – vyčíslení jednotlivých variant a následné vybrání optimálního řešení;
7. realizace varianty – praktické implementování vybraného řešení;
8. kontrola výsledků realizované varianty – zahrnuje porovnání dosažených výsledků vzhledem k definovaným cílům, sledování dopadů zavedeného řešení.

Rozhodovacím procesem v širším slova smyslu se rozumí proces, který zahrnuje všechny výše uvedené etapy. Pokud se proces zaměřuje výhradně na zhodnocení důsledků a výběr varianty, jedná se o rozhodovací proces v užším slova smyslu. [6]

Všechny kroky kromě č. 6 silně závisí na konkrétním rozhodovacím problému, nelze určit přesný postup a návod k jejich realizaci. Ve fázi hodnocení důsledků a výběr varianty lze ale využít univerzální metody, které vedou k vyřešení rozhodovacího problému. Další část textu práce se zaměřuje právě na tuto fázi. Podrobným zpracováním všech etap rozhodovacího procesu se zabývá Fotr v [4]. [3]

1.2 Rozdělení úloh vícekritériálního rozhodování

Podle charakteru informace o preferencích mezi kritérii a variantami lze úlohy vícekritériálního hodnocení variant dělit na čtyři typy – úlohy s žádnou, nominální, ordinální nebo kardinální informací.

Prvním typem je úloha s žádnou informací o preferencích kritérií. Není ale přípustné neznat preference mezi variantami, jelikož by poté nebylo možné určit lepší nebo horší variantu a úloha by se stala nevyřešitelnou. Pro vyřešení úlohy lze považovat všechna kritéria za stejně důležitá, případně jim přidělit stejné váhy. [2], [6]

U úloh s nominální informací také platí, jako u přechozího typu, že tato informace je možná pouze u preferencí kritérií mezi sebou. Nominální informace je vyjádřena pomocí aspiračních úrovní, tzn. že jsou určeny nejhorší hodnoty, kterých může varianta dosáhnout. Množina variant je tedy rozdělena na vyhovující a nevyhovující varianty. Stejně jako u předchozího typu lze úlohu řešit způsobem ohodnocení kritérií stejnou důležitostí nebo váhami. [1], [2]

Ordinální informace stanovuje pořadí kritérií nebo variant podle jejich významnosti. Je tedy známo, které kritérium případně varianta je preferována před jinou, ale není určeno kolikrát je kritérium nebo varianta upřednostňovaná. [2]

Posledním typem je úloha s kardinální informací o preferencích kritérií. V této situaci je známo kolikrát nebo o kolik je dané kritérium či varianta preferována před jinými. Preference kritérií určují jejich váhy, v případě variant se jedná o jejich, většinou číselné, ohodnocení. Většina metod vícekritériálního hodnocení variant vyžaduje právě tuto informaci o kritériích, proto existují různé metody pro stanovení vah kritérií. [2]

Úlohy vícekritériálního rozhodování lze také rozdělit podle jejich cíle řešení na úlohy s cílem výběru jedné kompromisní varianty, úplného uspořádání nebo rozdělení variant na akceptovatelné a neakceptovatelné. [2]

Úlohy, které si dávají za cíl výběr jedné kompromisní varianty, vybírají z množiny variant variantu, která se jeví jako nejlepší dle zadaných kritérií. Výsledek výběru je ovlivněn zvolenou metodou, jelikož některé metody se mohou ve výsledcích výběru nejlepší varianty lišit. [2]

Postup v úlohách s cílem uspořádat množinu variant je obdobný jako u přechozího typu úloh. Postupuje se tak, že se z množiny variant vybere nejlepší, následně se tato varianta z množiny odebere a celý proces se opakuje, dokud není velikost množiny variant rovna jedné. Zpravidla se varianty řadí od nejlepší po nejhorší. [2]

Pokud je cílem úlohy rozdělení variant na dobré a špatné varianty, nabízí se dva postupy řešení. Jelikož pojmy „dobrý“ a „špatný“ jsou relativní, je potřeba zvolit hraniční hodnoty. Při prvním postupu řešení si rozhodovatel zvolí hodnoty jednotlivých kritérií, které musí mít varianta hodnocené stejně nebo lépe, případně pouze lépe, aby byla klasifikována jako dobrá. Je přípustné si určit, zda je možná potencionální kompenzace nízkých hodnot kritéria vysokými hodnotami jiného kritéria. V druhém postupu řešení si rozhodovatel vytvoří fiktivní variantu s hraničními hodnotami pro jednotlivá kritéria. Množinu variant následně uspořádá a všechny varianty umístěné nad fiktivní variantou mohou být považovány jako přijatelné. Příkladem může být rozdělení bakalářských prací na akceptovatelné a neakceptovatelné. [2]

2 VÍCEKRITERIÁLNÍ HODNOCENÍ VARIANT ZA JISTOTY

2.1 Metody stanovení vah kritérií

Výchozím krokem pro vícekritériální hodnocení variant za jistoty je většinou určení jednotlivých vah kritérií. Váhou kritéria se číselně vyjadřuje významnost tohoto kritéria, kde obecně platí, že čím vyšší ohodnocení, tím vyšší významnost. Pro stanovení vah kritérií existuje několik metod. Váhy kritérií se vyjadřují v intervalu $(0, 1)$ a jejich součet musí být roven jedné, proto je posledním krokem při jejich určování normalizace jednotlivých vah. Výsledkem metod stanovení vah je váhový vektor. [2], [3]

2.1.1 Metody přímého stanovení vah kritérií

Zástupci metod přímého stanovení vah kritérií jsou bodovací metoda a metoda pořadí. Při aplikování těchto metod se určují váhy kritérií přímo.

Při využití bodovací metody se kritériu přiřazuje určitý počet bodů v závislosti na jeho důležitosti v rámci určené bodovací stupnice. Platí, že čím je vyšší důležitost kritéria, tím je kritérium ohodnoceno více body. Je přípustné použít stejnou hodnotu ohodnocení pro více kritérií. Pokud je větší počet kritérií nebo úloha vyžaduje vyšší rozlišovací schopnost mezi důležitostmi kritérií, zvolí se stupnice s vyšším rozsahem [3]. Tato metoda se využívá zejména tehdy, kdy úlohu řeší více expertů. Výpočet samotných vah se provede dle vzorce

$$v_j = \frac{b_j}{\sum_{j=1}^n b_j}, j = 1, 2, \dots, n, \quad (2.1)$$

kde b_j vyjadřuje součet všech bodů, které byly j -tému kritériu přiděleny. Tento postup se nazývá normalizace vah kritérií. Speciálním typem této metody je metoda alokace 100 bodů, kdy se všechny body musí rozdělit mezi kritéria dle jejich významnosti. [2], [4]

Postup metody pořadí spočívá v seřazení kritérií od nejdůležitějšího po nejméně důležité. Nejdůležitějšímu kritériu je přiřazeno n bodů, kde n je počet kritérií, další kritérium bude ohodnoceno $n-1$ body atd. Váha kritérií se vyjádří, stejně jako u předchozí metody, podle postupu normalizace vah kritérií. [2]

Bažant v [3] uvádí, že pořadí kritérií lze určit etapovým uspořádáním. Tento typ metody pořadí značně sníží náročnost při stanovování pořadí v úlohách, kde je velké množství kritérií. V závislosti na počtu kritérií se proces rozdělí do několika etap. Seřazení kritérií probíhá tak, že se v každé etapě určí nejdůležitější a nejméně důležité kritérium, které se následně vyloučí

z množiny kritérií. Tento postup se znovu opakuje s množinou s redukováným počtem kritérií. Při určování vah se nejprve nejméně důležitému kritériu přiřadí hodnota jedna a následně se určí kolikrát je předchozí kritérium významnější, postup se znovu opakuje s třetím kritériem od konce atd. a až nakonec je nejvýznamnějšímu kritériu přiřazena hodnota, kolikrát je významnější než poslední kritérium. V posledním kroku se provede normalizace těchto hodnot.

2.1.2 Metody založené na párovém porovnání

Metody založené na párovém porovnání porovnávají každou dvojici kritérií a určuje se, které z kritérií je důležitější. Počet srovnání N se vypočítá dle vzorce

$$N = \binom{n}{2} = \frac{n(n-1)}{2}. \quad (2.2)$$

Nejpoužívanější metody využívající párové porovnání jsou metoda Fullerova trojúhelníku a Saatyho metoda. [1]

Při využití Fullerova trojúhelníku se v prvním kroku očíslojí kritéria čísly 1, 2, ..., n . Následně se u každé dvojice označí důležitější kritérium. Pokud jsou kritéria považována za stejně důležitá, označí se obě a při počítání preferencí se každému z nich přidá 0,5 bodu [3]. Počet označení se pro každé kritérium sečte a z těchto výsledků se vypočítají váhy dle

$$v_j = \frac{n_j}{N}, j = 1, 2, \dots, n, \quad (2.3)$$

kde n_j vyjadřuje počet označení j -tého prvku. [2]

Tabulka 1: Schéma Fullerova trojúhelníku

K ₁	K ₁	...	K ₁
K ₂	K ₃	...	K _n
	K ₂	...	K ₂
	K ₃	...	K _n
	
	
			K _{n-1}
			K _n

Zdroj: vlastní zpracování dle [2]

Pokud by počet preferencí u určitého kritéria byl roven nule, pak i jeho váha bude rovna nula. Po ukončení porovnávání lze této situaci předcházet zvýšením počtu preferencí u každého kritéria o jednu. Tímto krokem je upraven vztah (2.3) na vztah (2.4). [2]

$$v_j = \frac{n_j + 1}{N + n}, j = 1, 2, \dots, n. \quad (2.4)$$

Při využití Saatyho metody se pro ohodnocení párových kritérií používá 9 bodová stupnice, kde platí, že hodnota 1 znamená, že porovnávaná kritéria jsou rovnocenná a hodnota 9 pak vyjadřuje, že kritérium i je absolutně preferované před j . Podrobnější verbální vyjádření stupňů je uvedeno v tabulce 2. [2]

Tabulka 2: Saatyho doporučená stupnice

Počet bodů	Popis ohodnocení
1	rovnocenná kritéria i a j
3	slabě preferované kritérium i před j
5	silně preferované kritérium i před j
7	velmi silně preferované kritérium i před j
9	absolutně preferované kritérium i před j

Zdroj: vlastní zpracování dle [2]

Přípustné je využívat i mezistupně (2, 4, 6, 8). Pokud je kritérium méně důležité, ohodnotí se převrácenou hodnotou z dané stupnice. Po porovnání a ohodnocení všech dvojic vytvoří řešitel Saatyho matici $S = (s_{ij})$:

$$S = \begin{bmatrix} 1 & \dots & s_{1n} \\ \vdots & \ddots & \vdots \\ \frac{1}{s_{1n}} & \dots & 1 \end{bmatrix}. \quad (2.5)$$

Prvek s_{ij} pro všechna $i, j = 1, 2, \dots, n$ vyjadřuje odhad podílu vah mezi i -tým a j -tým kritériem. Prvky na diagonále mají vždy hodnotu jedna, protože pokud by teoreticky bylo kritérium porovnáváno vůči sobě, byla by důležitost kritéria stejně významná. [2]

Pro výpočet vah kritérií se využívá normalizovaný geometrický průměr řádků matice S . V prvním kroku se vypočítají hodnoty b_i jako geometrický průměr řádků

$$b_i = \sqrt[n]{\prod_{j=1}^n s_{ij}}. \quad (2.6)$$

Druhý krok spočívá v normalizaci hodnot b_i dle vzorce (2.1), jehož výsledkem je požadovaný vektor vah. [2]

2.2 Metody výběru kompromisních variant

2.2.1 Metody nevyžadující informaci

Při typu úloh, kde chybí informace o preferencích mezi jednotlivými kritérii a kde jsou známé pouze preference mezi variantami, se využívá bodovací metoda nebo metoda pořadí. [2]

Prvním krokem při uplatňování těchto metod je ohodnocení všech variant. Při využití bodovací metody se určí stupnice, dle které se následně určují hodnoty, obdobně jako u využití bodovací metody pro ohodnocení kritérií. V případě metody pořadí je variantám přiřazeno číslo mezi 1 a m , kde obecně platí, že nejlepší ohodnocení je rovno m . [2]

Následně se určí celkové ohodnocení b_i pro každou z variant, které se vypočítá jako součet dílčích hodnot. Nakonec se varianty seřadí dle vypočtené hodnoty b_i sestupně a z tohoto uspořádání se vybere první varianta, tedy nejlepší. Tyto metody nabízí rozšíření v podobě vah kritérií, kde platí, že hodnota b_i se vypočte jako vážený součet. [2]

2.2.2 Metody s nominální informací

Metody, které využívají nominální informaci o kritériích, jsou charakteristické tím, že nepoužívají váhový vektor, ale využívají aspirační úrovně kritérií a rozdělují podle nich varianty na akceptovatelné (jejich kritériální hodnoty jsou nad určenými mezemi) a neakceptovatelné (jejich kritériální hodnoty jsou pod určenými mezemi). Pokud v množině akceptovatelných variant je příliš mnoho členů, je potřeba zpřísnit aspirační úrovně. Pokud je množina akceptovatelných variant prázdná, je nutné aspirační úrovně uvolnit. Pro řešení tohoto typu úloh se využívá konjunktivní a disjunktivní metoda a metoda PRIAM. [2]

Při využití konjunktivní metody se připouští zahrnout do akceptovatelných variant pouze ty, u kterých platí $y_{ij} \geq z_j$ pro všechna $j = 1, 2, \dots, n$, kde z_j je aspirační úroveň kritéria j . V případě disjunktivní metody se do množiny akceptovatelných variant zahrnou varianty, u kterých platí $y_{ij} \geq z_j$ pro alespoň jedno $j = 1, 2, \dots, n$. Tyto metody lze použít pro zmenšení počtu variant a poté využít jinou metodu pro zpracování této množiny nebo jsou vhodné pro analyzování akceptovatelných variant při různých aspiračních úrovních kritérií. [2]

Při použití metody PRIAM se prohledává množina variant v několika krocích tak, aby bylo nalezeno právě jedno požadované řešení. Vektor kritériálních hodnot $\vec{y}_j \in Y$ je zobrazením varianty a_j . Aspirační úrovně kritérií v s -tém kroku vyjadřuje $z^{(s)} = (z_1^{(s)}, z_2^{(s)}, \dots, z_n^{(s)})$, změny aspiračních úrovní jsou vyjádřené $\delta z^{(s)}$, jestli je změna kladná nebo záporná závisí na tom, jestli jsou kritéria minimalizační nebo maximalizační. Většinou jsou aspirační úrovně v prvním

kroku, tj. $z^{(1)}$, rovny bazální variantě úlohy. Cílem je najít varianty, pro jejichž kritériální hodnoty platí

$$\vec{y}_j \geq z^{(s)} \quad (2.7)$$

a jejichž počet vyjadřuje číslo d , podle kterého rozhodovatel upravuje aspirační úroveň pro krok $s+1$ [2]:

$$z^{(s+1)} = z^{(s)} + \delta z^{(s)}. \quad (2.8)$$

Hodnota d může nabývat tří hodnot:

- $d > 1$, rozhodovatel změní aspirační úroveň tak, aby docílil snížení počtu akceptovatelných variant;
- $d = 1$, rozhodovatel našel nedominovanou variantu;
- $d = 0$, nebyla nalezena vyhovující varianta a hledá se ta nejbližší k zadaným aspiračním úrovním. [2]

Pokud nastane třetí možnost, je potřeba vypočítat vzdálenosti jednotlivých variant od zadaných aspiračních úrovní kritérií:

$$\sum_{i=1}^n \frac{1}{y_i^*} |z_i^{(s)} - y_{ij}|, \quad (2.9)$$

kde y_i^* , $i=1, 2, \dots, n$ představuje ideální kritériální hodnoty. Výslednou variantou je ta, která má odchylku od ideálních hodnot nejnižší. [2]

2.2.3 Metody s ordinální informací

Pro využití metod, které pracují s ordinální informací o kritériích a/nebo variantách, je vyžadováno zadání pořadí důležitosti kritérií a pořadí variant. Zástupcem tohoto typu metod je lexikografická metoda a metoda ORESTE.

Lexikografická metoda se řadí mezi jednoduché postupy při řešení vícekritériálního rozhodování. Její princip spočívá v tom, že největší vliv na výběr varianty má nejdůležitější kritérium. Pokud existuje více variant, které jsou ohodnoceny stejně v nejdůležitějším kritériu, porovnávají se podle druhého nejdůležitějšího kritéria atd. Nevýhodou tohoto přístupu je úplné zanedbání hodnot nejméně důležitých kritérií. [1], [2]

Při použití metody ORESTE je od rozhodovatele požadováno úplné uspořádání kritérií a variant podle jednotlivých kritérií. Metoda ORESTE se rozděluje na dvě části. [2]

V první části se určí vzdálenost každé varianty podle jednotlivých kritérií od fiktivního počátku, kde fiktivní varianta a fiktivní kritérium je ohodnoceno hodnotou 0. Následujícím krokem je uspořádání variant dle určitých pravidel. [2]

Vektor, který vyjadřuje pořadová čísla kritérií, kde indiferentní kritéria jsou ohodnocena průměrnými pořadovými čísly, je označen jako $q = (q_1, q_2, \dots, q_n)$. Uspořádání variant podle jednotlivých kritérií je vyjádřeno pomocí matice $P = (p_{ij})$, $i = 1, 2, \dots, m, j = 1, 2, \dots, n$. Pomocí Dujmičovy metriky se vypočte matice D , jejíž prvky d_{ij} vyjadřují vzdálenost od fiktivního počátku $p_{0j} = (0, 0, \dots, 0)$:

$$d_{ij} = (0,5(p_{ij})^r + 0,5(q_j)^r)^{\frac{1}{r}} \quad i = 1, 2, \dots, m, j = 1, 2, \dots, n, \quad (2.10)$$

kde číslo r je reálné číslo. Zpravidla se hodnota tohoto čísla pohybuje kolem hodnoty 3. Vypočítané vzdálenosti d_{ij} se následně vzestupně uspořádají a ohodnotí pořadovými čísly r_{ij} nebo, v případě stejných hodnot, se ohodnotí průměrnými pořadovými čísly. Ze vzniklé matice $R = (r_{ij})$ určíme pro každý řádek jeho součet r_i . Uspořádání variant získáme vzestupným uspořádáním hodnot r_i . [2]

Druhá část metody obsahuje preferenční analýzu, díky které lze pro každou dvojici variant zjistit jejich preference P , indiference I a nesrovnatelnosti N . Nejprve se na základě získaných hodnot r_{ij} vypočítají tzv. hodnoty preferenčních intenzit:

$$c_{ij} = \sum_{h \in K} (r_{jh} - r_{ih}), \quad i, j = 1, 2, \dots, n, \quad (2.11)$$

kde K označuje množinu indexů kritérií, kdy je varianta a_i lepší než varianta a_j . Relativní preferenční intenzity vzhledem k maximální možné hodnotě jsou

$$c_{ij}^N = \frac{c_{ij}}{c_{max}}, \quad i, j = 1, 2, \dots, n, \quad (2.12)$$

kde c^{max} označuje maximální hodnotu, kterou může dosáhnout preferenční intenzita a která se vypočítá $c^{max} = m^2 (n-1)$. Předpokládá se, že platí $c_{ij}^N \geq c_{ji}^N$. Pro další pokračování musí rozhodovatel určit prahové hodnoty α , β a γ . V následném kroku se provede test indiference. Jestliže platí

$$c_{ij}^N \leq \alpha \text{ a } c_{ij}^N - c_{ji}^N \leq \beta, \quad (2.13)$$

pak jsou varianty a_i a a_j indiferentní. V opačném případě nastane vztah preference nebo nesrovnalosti. Jestliže platí

$$\frac{c_{ij}^N}{c_{ij}^N - c_{ji}^N} \geq \gamma, \quad (2.14)$$

pak jsou varianty a_i a a_j vzájemně nesrovnatelné, tzn. že se nelze rozhodnout o preferenci nebo o indiferenci variant na základě zadaných informací. V opačném případě varianta a_i preferuje variantu a_j . [2]

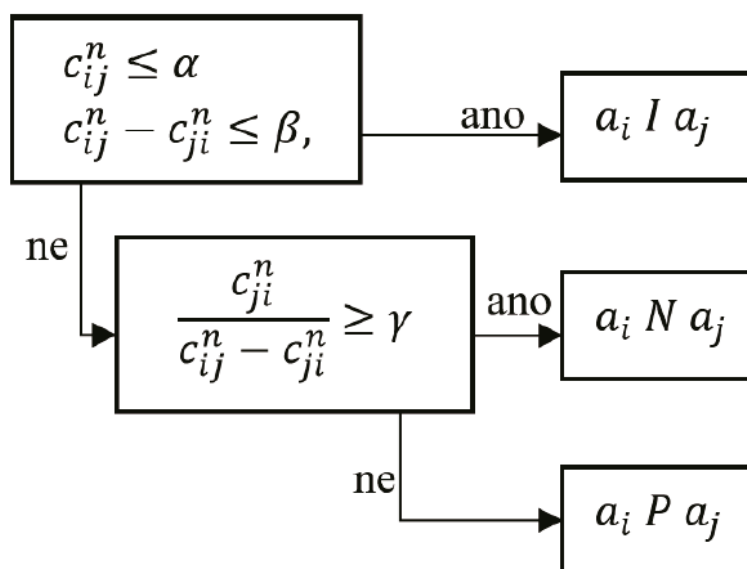
Výsledky preferenční analýzy jsou značně ovlivněny volbou prahů α , β a γ . Pro horní meze prahů α a β se dají odvodit následující hodnoty:

$$\alpha \leq \frac{1}{2(m-1)}, \beta \leq \frac{1}{n(m-1)}. \quad (2.15)$$

Pro dolní mez prahu γ hodnoty:

$$\gamma \geq \frac{n-2}{4}. \quad (2.16)$$

Postup preferenční analýzy lze schematicky vyjádřit, jak je to na obrázku 1.



Obrázek 1: Schéma postupu preferenční analýzy [2]

Výsledky preferenční analýzy se zachycují v matici, kde její řádky i sloupce odpovídají variantám. Prvky matice pak vyjadřují informaci o vzájemném vztahu mezi jednotlivými variantami. Symbol I se používá pro indiferenci variant, symbol N pro nesrovnalost, symbol „+“ znamená, že a_i je preferovaná před a_j a symbol „-“ znamená, že a_j je preferovaná před a_i . [2]

2.2.4 Metody s kardinální informací

Pro metody vyžadující kardinální informaci platí, že k jejich vyřešení musí řešitel mít k dispozici vektor vah kritérií a kritériální matici. Tato kapitola představí metodu váženého součtu, metodu AHP a metodu TOPSIS. Mezi další metody patří např. PROMETHEE, metody třídy ELECTRE nebo MAPPAC. [1]

Metodu váženého součtu (WSA) lze použít jak pro vyhledání nejlepší možné varianty, tak pro uspořádání množiny variant dle důležitosti, protože vytváří celkové hodnocení pro každou z variant [2]. Metoda je založena na principu maximalizace užitku, tzn. že je určena hodnota užitku, kterou výběr varianty přinese. Užitek je vyjádřen hodnotou ze stupnice 0 až 1, kde platí, že čím vyšší hodnota, tím lepší varianta. [1]

Prvním krokem je vytvoření kritériální matice $R = (r_{ij})$, jejíž prvky získáme pomocí vzorce

$$r_{ij} = \frac{y_{ij} - d_j}{h_j - d_j}, \quad (2.17)$$

kde h_j představuje ohodnocení j -tého prvku v určené ideální variantě H a kde d_j představuje ohodnocení j -tého prvku v určené bazální variantě D . Prvky matice R pak představují hodnoty funkce užitku jednotlivých variant podle jednotlivých kritérií. Celkový užitek varianty a_i se vypočítá dle

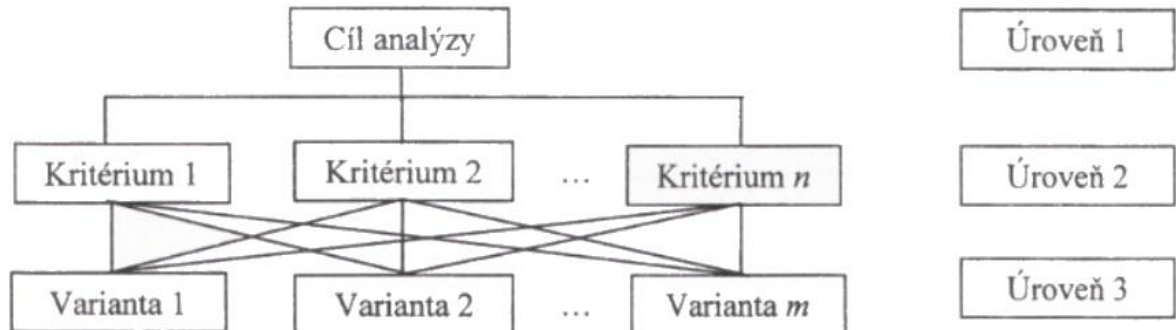
$$u(a_i) = \sum_{j=1}^n v_j r_{ij}. \quad (2.18)$$

Varianta s nejvyšší hodnotou $u(a_i)$ je hledaná nejlepší varianta, případně sestupným seřazením variant podle hodnot $u(a_i)$ získáme uspořádanou množinu variant. [2]

Metoda AHP vytváří hierarchický systém problému, tzn. rozklad struktury složitých situací na jednodušší části. Hodí se tedy pro řešení rozsáhlých rozhodovacích problémů. Hierarchickou strukturu tvoří několik úrovní, kde každá z nich má několik prvků. Ve struktuře platí, že čím obecnější jsou prvky ve vztahu k danému problému, tím jsou v hierarchii na vyšší úrovni.

Typicky rozhodovací úloha obsahuje tři úrovně [2]:

- úroveň 1 – cíl vyhodnocování,
- úroveň 2 – kritéria vyhodnocování,
- úroveň 3 – posuzované varianty.



Obrázek 2: Hierarchická struktura se třemi úrovněmi [2]

Po vytvoření hierarchické struktury se na každé úrovni provádí párové srovnání prvků pomocí Saatyho metody. Prvku nejvyšší úrovně se určí hodnota, typicky je to hodnota 1, a ta se rozdělí mezi prvky druhé úrovně. Ohodnocení prvků druhé úrovně tedy odpovídá váhám kritérií v_j pro $j = 1, 2, \dots, n$ a výsledkem porovnávání je matice $n \times n$. Ohodnocení prvků druhé úrovně se následně rozdělí mezi prvky třetí úrovně w_{ij} pro $i = 1, 2, \dots, n$ a $j = 1, 2, \dots, m$. Výsledkem porovnávání prvků třetí úrovně je n matic s rozměrem $m \times m$. Hodnocení varianty z hlediska všech kritérií získáme součtem jejich hodnot pro všechna kritéria. Podle těchto vypočtených hodnot lze následně uspořádat množinu variant a vybrat nejlepší variantu. [2]

Metoda TOPSIS využívá pro řešení rozhodovacích úloh vzdálenost jednotlivých variant od ideální a bazální varianty. Nejlepší varianta má nejmenší vzdálenost od ideální varianty a největší vzdálenost od bazální varianty. Metoda také nabízí úplné uspořádání množiny variant, hodí se tedy pro řešení úloh, které vyžadují uvedení pořadí variant. Pro použití metody je třeba převést všechna minimalizační kritéria na maximalizační jejich vynásobením konstantou -1 . [2]

Podle Brožové v [2] lze postup rozdělit do následujících kroků:

1. Dle následujícího vzorce (2.19) se vytvoří matice $R = (r_{ij})$.

$$r_{ij} = \frac{y_{ij}}{\sqrt{\sum_{i=1}^p y_{ij}^2}} \quad (2.19)$$

2. Pomocí $w_{ij} = v_j r_{ij}$ se vypočítá normalizovaná vážená kritériální matice $W = (w_{ij})$.

3. Vzhledem k vypočítaným hodnotám v matici W řešitel určí ideální variantu h s ohodnocením h_1, h_2, \dots, h_n a bazální variantu d s ohodnocením d_1, d_2, \dots, d_n . Při určování ideální, resp. bazální varianty platí, že hodnota h_j , resp. d_j , vyjadřuje maximální, resp. minimální hodnotu prvku w_{ij} pro $j = 1, 2, \dots, n$ [1].
4. Dle následujícího vzorce (2.20), resp. (2.21), se vypočítá vzdálenost od ideální varianty, resp. bazální varianty.

$$d_i^+ = \sqrt{\sum_{j=1}^k (w_{ij} - h_j)^2} \quad (2.20)$$

$$d_i^- = \sqrt{\sum_{j=1}^k (w_{ij} - d_j)^2} \quad (2.21)$$

5. Relativní index vzdáleností jednotlivých variant od bazální varianty vypočítáme dle vzorce (2.22).

$$c_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad (2.22)$$

Hodnoty c_i jsou v intervalu $\langle 0, 1 \rangle$, kde bazální varianta má hodnotu 0 a ideální varianta má hodnotu 1.

6. Podle hodnot c_i se varianty sestupně uspořádají. Prvních x variant tohoto uspořádání představuje x požadovaných variant pro vyřešení rozhodovacího problému.

3 PRAKTICKÁ ČÁST

Cílem praktické části bylo vytvořit jednoduchou aplikaci pro vícekriteriální rozhodování a demonstrovat v ní konkrétní rozhodovací situaci s využitím několika metod vícekriteriálního rozhodování. Výstupem praktické části, kromě samotné aplikace, je porovnání výsledků jednotlivých rozhodovacích metod použitých v aplikaci. Při vytváření diagramu užití a třídnicích diagramů byla využita kniha [7]. Při vývoji aplikace byla využívána dokumentace [8].

3.1 Implementace

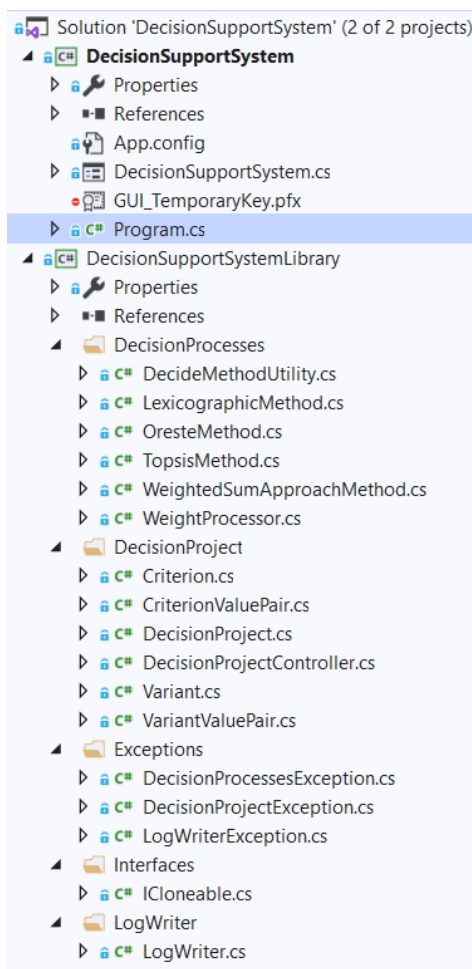
Pro vytvoření aplikace Decision Support System byl zvolen objektově orientovaný programovací jazyk C# s využitím frameworku .NET ve verzi 4.8 z důvodu jeho relativní jednoduchosti. Rozhodujícím faktorem byl také fakt, že tento jazyk je vyučován v rámci studia na Fakultě elektrotechniky a informatiky Univerzity Pardubice. Grafické uživatelské prostředí bylo vytvořeno pomocí knihovny tříd Windows Forms. Aplikace byla vytvářena ve vývojovém prostředí Microsoft Visual Studio 2019 v bezplatné verzi Visual Studio Community na přenosném počítači s operačním systémem Windows 10.

3.1.1 Struktura aplikace

Aplikace Decision Support System je rozdělena na dva projekty. První projekt je DLL knihovna *DecisionSupportSystemLibrary*, která zajišťuje operace potřebné pro učinění vícekriteriálního rozhodnutí. Druhým projektem je projekt *DecisionSupportSystem*, který obsahuje formuláře a třídy zajišťující grafické uživatelské prostředí.

Knihovna *DecisionSupportSystemLibrary* obsahuje pět složek se zdrojovými soubory. První složka *DecisionProcesses* obsahuje statické třídy určené pro zpracování rozhodovacího problému a určení pořadí variant. Každá metoda má svoji třídu. Složka *DecisionProject* obsahuje zdrojové soubory tříd pro ukládání struktury rozhodovacího problému. Další složka *Exceptions* obsahuje zdrojové soubory tří tříd, které slouží pro vyvolávání vlastních výjimek. Čtvrtá složka *Interfaces* obsahuje zdrojový soubor jednoho rozhraní. Pátá složka *LogWriter* obsahuje jednu třídu *LogWriter*, která zajišťuje zapisování výsledků průběžných výpočtů při rozhodování.

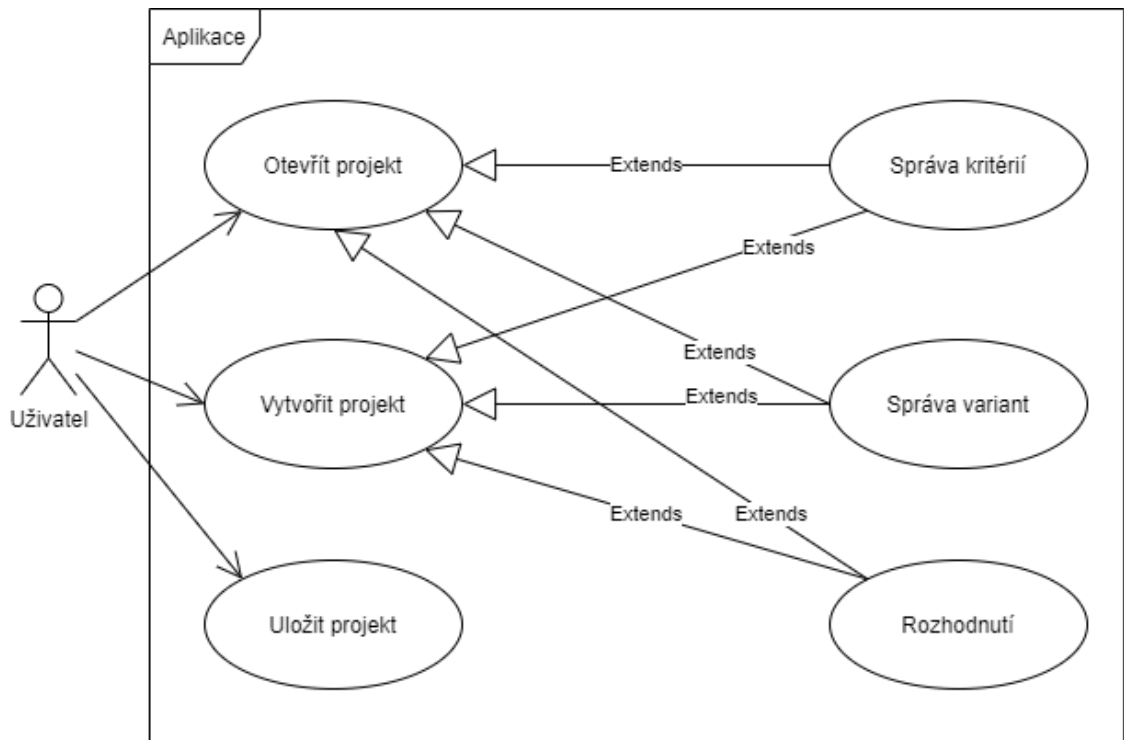
Projekt *DecisionSupportSystem* obsahuje jedno formulářové okno *DecisionSupportSystem* a třídu *Program*, která obsahuje jednu metodu *Main()*. Tato metoda je hlavní vstupní bod aplikace a je volána při jejím spuštění.



Obrázek 3: Struktura aplikace Decision Support System

3.1.2 UseCase diagram

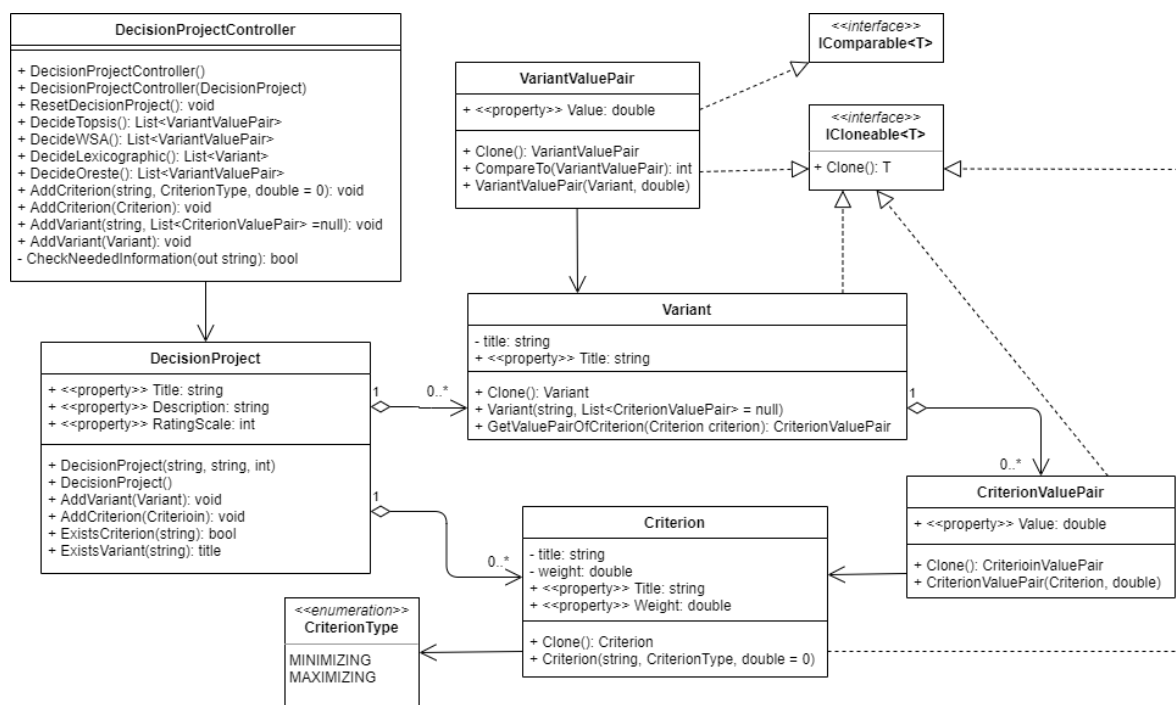
Aplikace má jednoho aktéra, a tím je uživatel, který aplikaci spustí. Uživatel má možnost otevřít si již vytvořený projekt, vytvořit nový projekt nebo projekt uložit. V rámci otevřeného nebo vytvořeného projektu může editovat jednotlivá kritéria a váhově je ohodnotit, editovat varianty a ohodnotit je v jednotlivých kritériích a provést rozhodnutí.



Obrázek 4: UseCase diagram aplikace

3.1.3 Diagramy tříd

Na obrázku 5 je zobrazen diagram tříd ze složky *DecisionProject* a rozhraní ze složky *Interfaces*. Třída *DecisionProjectController* poskytuje metody pro práci s instancí třídy *DecisionProject*. Třída *DecisionProject* představuje rozhodovací projekt, který obsahuje mj. seznam variant uložený ve struktuře *List<Variant>* a seznam kritérií uložený ve struktuře *List<Criterion>*. Třída *Variant* představuje jednotlivé varianty, která má mj. atribut typu *List<CriterionValueType>* reprezentující seznam kritérií a k nim přiřazených hodnot pro danou variantu. Třída *Criterion* reprezentuje jednotlivá kritéria. Obsahuje atribut typu *CriterionType*, který udává, jestli je kritérium minimalizační nebo maximalizační. Třída *VariantValuePair* reprezentuje variantu a k ní přiřazenou hodnotu. Tato třída se používá např. při využití rozhodovací metody TOPSIS. Třídy *VariantValuePair*, *Variant*, *Criterion* a *CriterionValuePair* implementují rozhraní *ICloneable<T>*, které slouží pro vytváření hluboké kopie objektů.



Obrázek 5: Diagram tříd ve složce *DecisionProject*

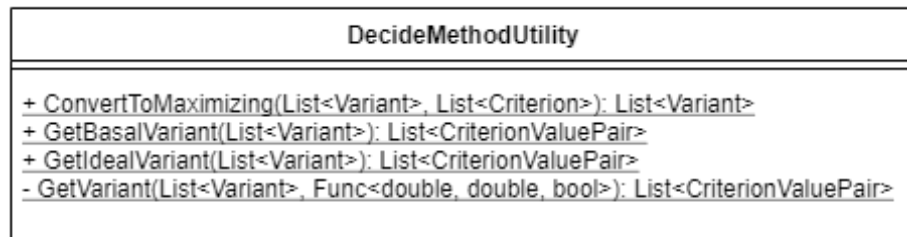
Třída *WeightProcessor*, jejíž diagram je zobrazen na obrázku 6, slouží pro zpracování vah kritérií pomocí bodové metody. Obsahuje veřejnou metodu *StandardizeWeightsOfCriteria*(*List<Criterion>*), jejíž návratová hodnota představuje jednotlivá ohodnocená kritéria. Tato metoda využívá privátní metodu *CheckWeights()*, která ověřuje, že jsou všechna kritéria ohodnocena body a žádné není tedy ohodnoceno hodnotou 0. Metoda *StandardizeWeights()* pak již provádí samotné ohodnocení kritérií.



Obrázek 6: Diagram třídy *WeightProcessor*

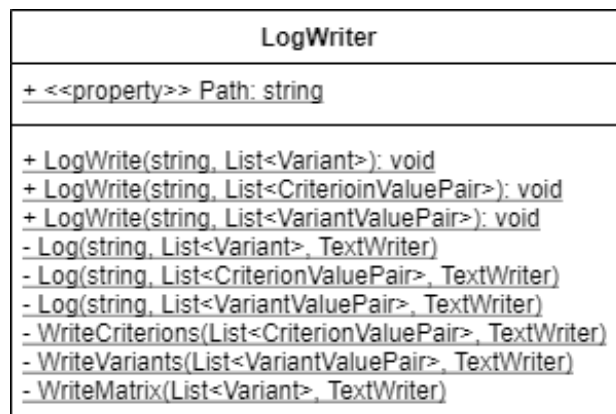
Třída *DecideMethodUtility* zobrazená na obrázku 7 obsahuje pomocné metody, které jsou využívány ve více než jedné rozhodovací metodě. Metoda *ConvertToMaximizing*(*List<Variant>*, *List<Criterion>*) převádí hodnoty v minimalizačních kritériích na maximalizační vynásobením původní hodnoty hodnotou -1 . Metody *GetIdealVariant*(*List<Variant>*) a *GetBasalVariant*(*List<Variant>*) vrací ideální, resp.

bazální variantu předané matice v parametru. Tyto metody k výpočtu využívají metodu *GetVariant(List<Variant>, Func<double, double, bool>)*.



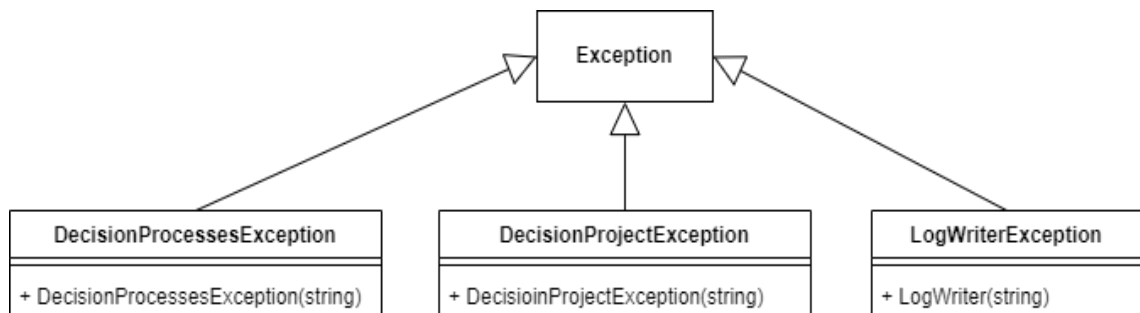
Obrázek 7: Diagram třídy *DecideMethodUtility*

Třída *LogWriter*, jejíž diagram je zobrazen na obrázku 8, slouží pro ukládání průběžných výsledků výpočtů při aplikování různých rozhodovacích metod. Průběžné výsledky se ukládají do textového souboru, který se nachází v adresáři, ve kterém je spouštěcí soubor (*DecisionSupportSystem.exe*).



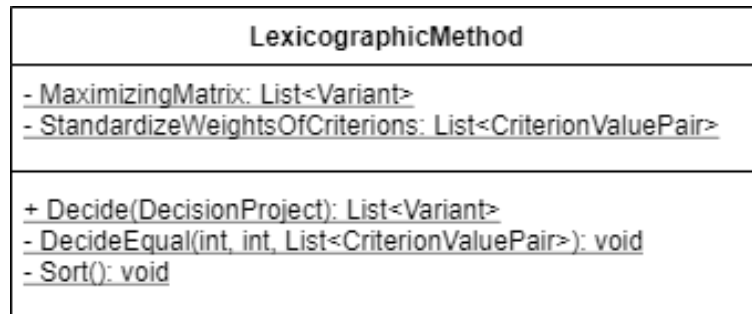
Obrázek 8: Diagram třídy *LogWriter*

Obrázek 9 zobrazuje diagram tří tříd, které slouží pro vyvolávání vlastně definovaných výjimek. Všechny třídy dědí ze systémové třídy *Exception*.



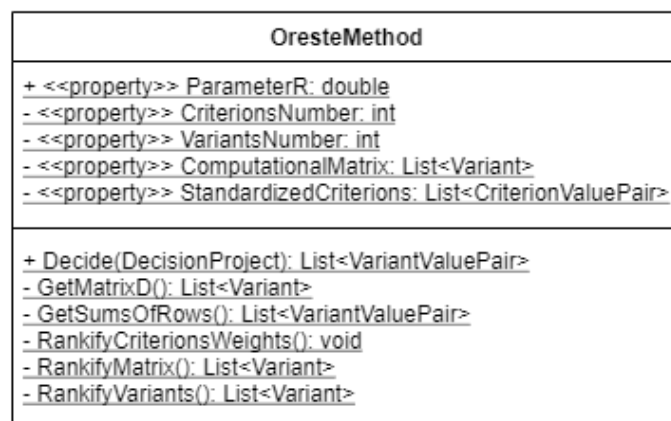
Obrázek 9: Diagram tříd potomků *Exception*

Následující obrázky 10-19 zobrazují diagramy statických tříd jednotlivých metod rozhodování, které byly implementovány, a jejich stěžejní algoritmy. Každá z těchto tříd obsahuje veřejnou metodu *Decide(DecisionProject)*, která přijímá jeden parametr typu *DecisionProject* a má návratový typ *List<VariantValuePair>*, nebo *List<Variant>* v případě lexikografické metody. Samotnému rozhodování pomocí dané metody předchází převedení hodnot v minimalizačních kritériích na maximalizační a normalizace vah kritérií. Tyto operace jsou provedeny s využitím již představených metod ve třídách *WeightProcessor* a *DecideMethodUtility*.



Obrázek 10: Diagram třídy *LexicographicMethod*

Pro rozhodování s využitím třídy *LexicographicMethod*, tedy rozhodování pomocí lexikografické metody, je stěžejní metoda *Sort()*, která řadí seznam variant dle nejdůležitějšího kritéria. Pokud se hodnota variant v nejdůležitějším kritériu rovná, využívá se rekurzivní metoda *DecideEqual(int, int, List<CriterionValuePair>)*, která rozhoduje v druhém nejdůležitějším kritériu atd.



Obrázek 11: Diagram třídy *OresteMethod*

Pro rozhodování s využitím třídy *OresteMethod*, tedy rozhodování pomocí metody ORESTE, se využívá několik metod. Nejprve je potřeba pro každé kritérium stanovit pořadí variant podle

daného kritéria. Tuto operaci zajišťuje metoda *RankifyVariants()* s využitím algoritmu zobrazeného na obrázku 12.

```

for (int i = 0; i < CriterionsNumber; i++)
{
    for (int j = 0; j < VariantsNumber; j++)
    {
        int r = 1, s = 1;
        for (int k = 0; k < VariantsNumber; k++)
        {
            if (j != k && ComputationalMatrix[k].Criterions[i].Value > ComputationalMatrix[j].Criterions[i].Value)
            {
                r += 1;
            }
            if (j != k && ComputationalMatrix[k].Criterions[i].Value == ComputationalMatrix[j].Criterions[i].Value)
            {
                s += 1;
            }
        }
        variants[j].Criterions[i].Value = r + (s - 1) / (double)2;
    }
}

```

Obrázek 12: Algoritmus pro ohodnocení variant, metoda ORESTE [9]

Dalším krokem je získání matice vzdáleností $D = (d_{ij})$ pomocí vzorce (2.10). Parametr r je implicitně nastaven na hodnotu 3, je jej ale možnost explicitně definovat předáním volitelného parametru při volání metody *Decide(DecisionProject, int)*.

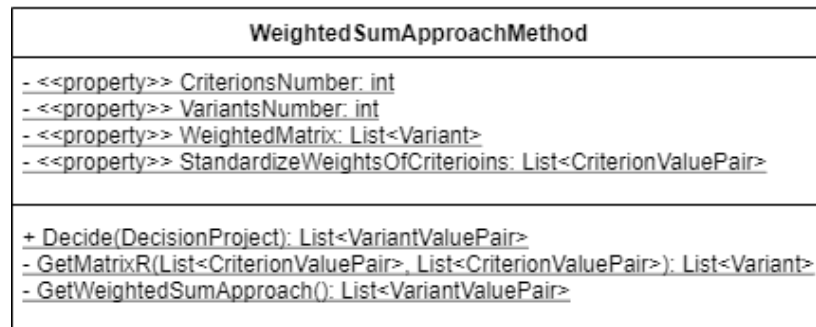
```

for (int i = 0; i < VariantsNumber; i++)
{
    for (int j = 0; j < CriterionsNumber; j++)
    {
        double value = Math.Pow(0.5 * Math.Pow(matrixD[i].Criterions[j].Value, ParameterR) +
            0.5 * Math.Pow(StandardizedCriterions[j].Value, ParameterR), 1 / ParameterR);
        matrixD[i].Criterions[j].Value = Math.Round(value,8);
    }
}

```

Obrázek 13: Algoritmus pro výpočet matice D , metoda ORESTE

Prvky matice D je následně nutno ohodnotit pořadovými čísly s využitím metody *RankifyMatrix()*. Posledním krokem výpočtu je vypočítání součtů jednotlivých řádků ohodnocené matice. Tento krok se provede pomocí metody *GetSumsOfRows()*, jejíž návratová hodnota představuje varianty a k nim přiřazené hodnoty součtu jejich řádků. Pro získání uspořádání variant je potřeba tyto varianty seřadit podle součtu řádků.



Obrázek 14: Diagram třídy *WeightedSumApproachMethod*

Při rozhodování s využitím třídy *WeightedSumApproachMethod*, tedy rozhodování pomocí metody váženého součtu, se při rozhodování používají v zásadě dvě metody. Metoda *GetMatrixR(List<CriterionValuePair>, List<CriterionValuePair>)*, kde parametry představují ideální, resp. bazální variantu, vypočítává standardizovanou kritériální matici $R = (r_{ij})$.

```

for (int i = 0; i < VariantsNumber; i++)
{
    for (int j = 0; j < CriteriaNumber; j++)
    {
        double denominator = idealVariant[j].Value - basalVariant[j].Value;
        matrixR[i].Criteria[j].Value = Math.Round(
            (WeightedMatrix[i].Criteria[j].Value - basalVariant[j].Value) / denominator, 8);
    }
}

```

Obrázek 15: Algoritmus pro výpočet matice R , metoda váženého součtu

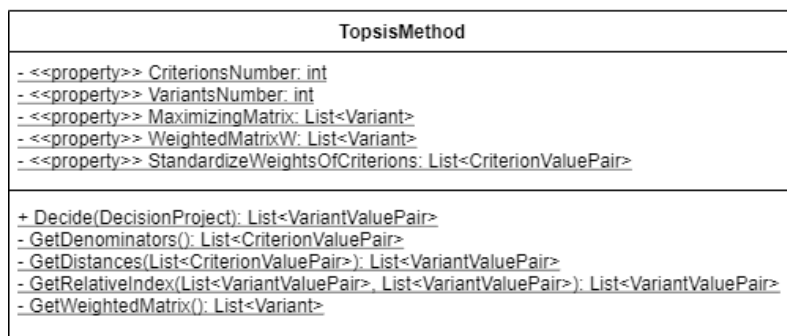
Následně se ze získané matice R vypočítá hodnota funkce užitku pomocí metody *GetWeightedSumApproach()*, jejíž návratová hodnota představuje varianty a k nim přiřazené hodnoty funkce užitku. Posledním krokem pro učinění rozhodnutí je seřazení seznamu variant dle jejich hodnoty.

```

for (int i = 0; i < VariantsNumber; i++)
{
    double sum = 0;
    for (int j = 0; j < CriteriaNumber; j++)
    {
        sum += StandardizeWeightsOfCriteriaoins[j].Value * WeightedMatrix[i].Criteria[j].Value;
    }
    variants.Add(new VariantValuePair(WeightedMatrix[i], sum));
}

```

Obrázek 16: Algoritmus pro získání hodnot funkcí užitku, metoda váženého součtu



Obrázek 17: Diagram třídy *TopsisMethod*

Pro rozhodování s využitím třídy *TopsisMethod*, tedy rozhodování pomocí metody TOPSIS, je k dosažení výsledků z důvodů vyšší početní složitosti využíváno více metod. Prvním krokem je získání vážené kritériální matice $W = (w_{ij})$, která se získá využitím metody *GetWeightedMatrix()*. V této metodě se také využívá metoda *GetDenominators()*, která vypočítává jmenovatel ve vzorci (2.19).

```

List<CriterionValuePair> criterionsDenominators = GetDenominators();
for (int i = 0; i < CriteriaNumber; i++)
{
    for (int j = 0; j < VariantsNumber; j++)
    {
        double oldValue = weightedMatrix[j].Criteria[i].Value;
        weightedMatrix[j].Criteria[i].Value = Math.Round(oldValue / criterionsDenominators[i].Value
            * StandardizeWeightsOfCriteria[i].Value, 8);
    }
}

```

Obrázek 18: Algoritmus pro získání matice W , metoda TOPSIS

Dalším krokem je využití metod *GetIdealVariant(List<Variant>)* a *GetBasalVariant(List<Variant>)* ze třídy *DecideMethodUtility*. Následujícím krokem je výpočet vzdáleností variant od ideální a bazální varianty. Pro vykonání tohoto kroku je využita metoda *GetDistances(List<CriterionValuePair>)*, která přijímá parametr představující ideální nebo bazální variantu.

```

for (int i = 0; i < VariantsNumber; i++)
{
    double sum = 0;
    for (int j = 0; j < CriteriaNumber; j++)
    {
        sum += Math.Pow(WeightedMatrixW[i].Criteria[j].Value - variant[j].Value, 2);
    }
    sum = Math.Sqrt(sum);
    distance.Add(new VariantValuePair(WeightedMatrixW[i].Clone(), Math.Round(sum, 8)));
}

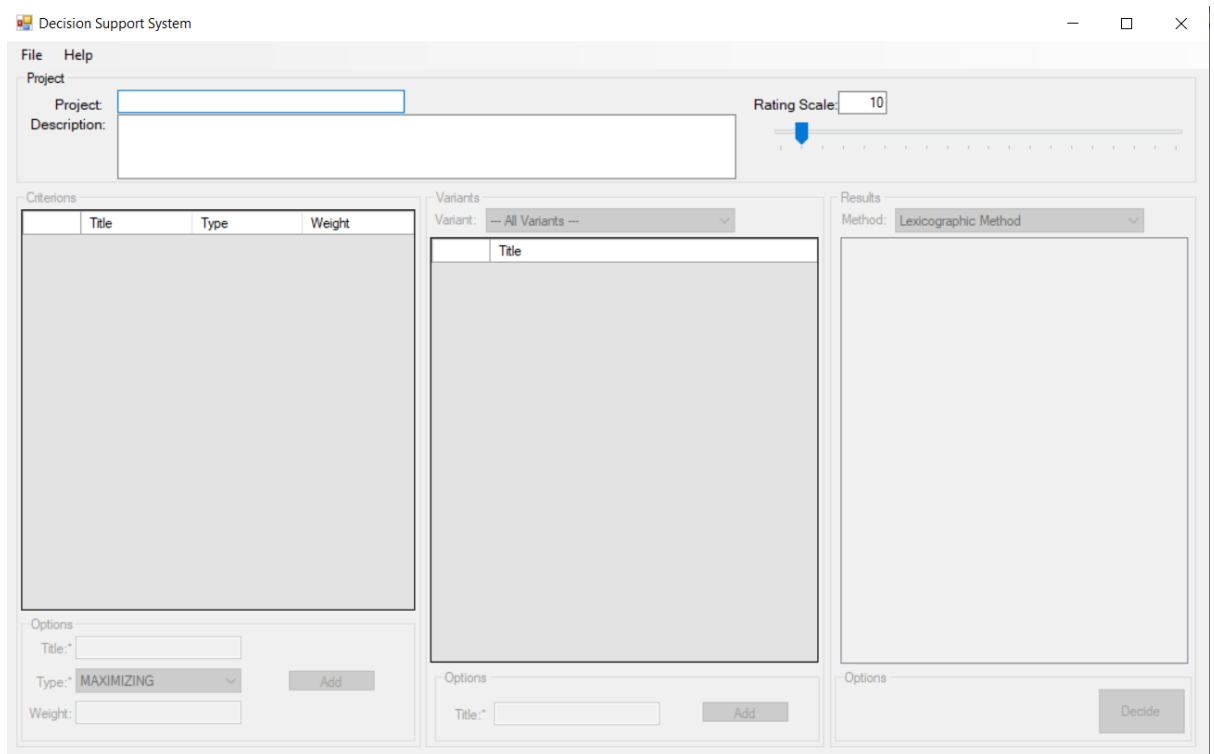
```

Obrázek 19: Algoritmus pro získání vzdáleností, metoda TOPSIS

Dalším krokem je získání relativních indexů vzdáleností od bazální varianty pro jednotlivé varianty pomocí metody *GetRelativeIndex(List<VariantValuePair>, List<VariantValuePair>)*, kde její parametry představují vzdálenosti pro jednotlivé varianty získané v předchozím kroku. Návrátová hodnota této metody představuje varianty a k nim přiřazené hodnoty relativního indexu vzdáleností. Posledním krokem je seřazení variant podle indexu vzdáleností.

3.1.4 Uživatelské prostředí

Uživatelské prostředí bylo tvořeno s důrazem na jednoduchost a funkčnost. Pro jeho tvorbu byly využity základní komponenty nabízené frameworkem s využitím grafického designeru. Důvodem volby využití Windows Forms oproti frameworku WPF byla jednoduchost a využívání tohoto frameworku při výuce.



Obrázek 20: Uživatelské prostředí po spuštění aplikace

Uživatelské prostředí se skládá z jednoho formulářového okna, které je rozděleno na čtyři základní části a obsahuje také horní menu se dvěma položkami. Horní menu je vytvořeno s využitím komponenty *MenuStrip*, která obsahuje komponenty *ToolStripMenuItem*. Jednotlivé položky mají definované události reagující na kliknutí na položku.

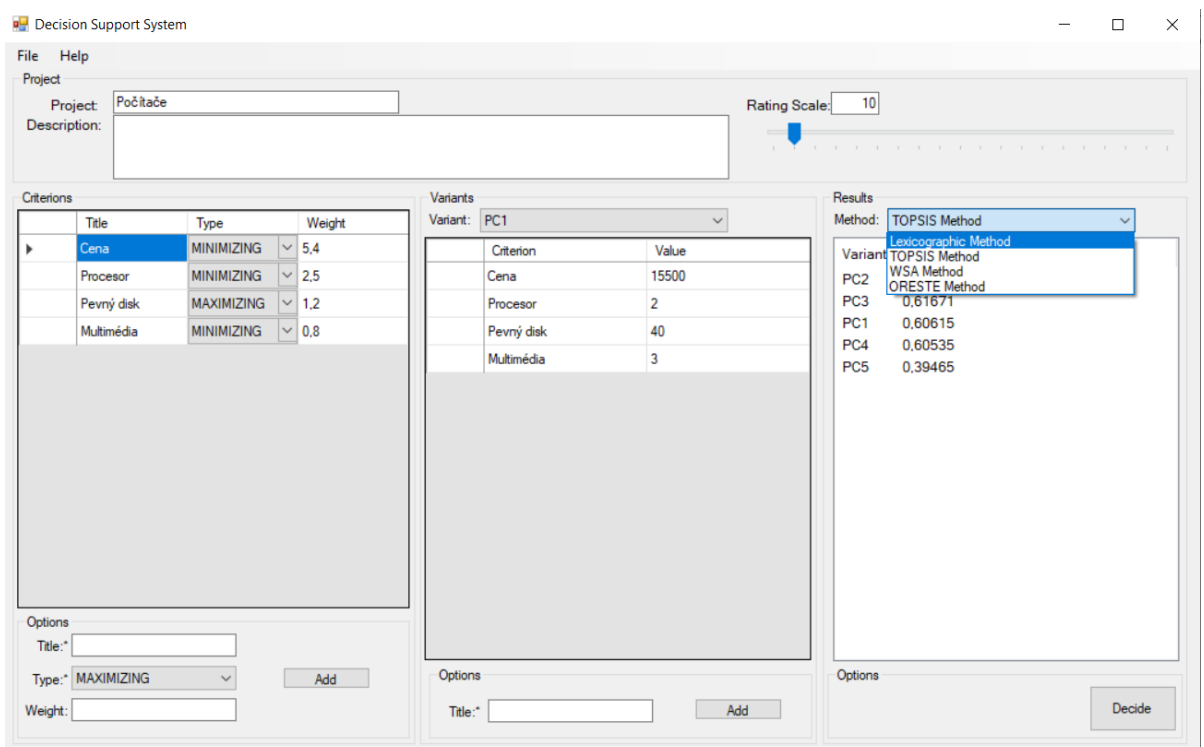
V horní části formuláře se nachází komponenta *GroupBox* s názvem Project, která obsahuje položky pro nastavení rozhodovacího problému. Pro vyplnění textových údajů byla využita

komponenta *TextBox*. Pro určení bodové stupnice byl využit prvek *TrackBar*.

Levá část formuláře obsahuje prvek *GroupBox* s názvem *Criteria*s. Tento prvek obsahuje další komponenty pro zobrazování a editaci kritérií. V dolní části se nachází další *GroupBox*, který obsahuje textová pole a tlačítko pro přidání kritéria. Nachází se zde také prvek *ComboBox*, který slouží pro výběr typu kritéria. Kritéria jsou zobrazována pomocí komponenty *DataGridView*, kde je lze také editovat a mazat.

Prostřední část formuláře obsahuje prvek *GroupBox* s názvem *Variants*. Tento prvek obsahuje další komponenty pro zobrazování a editaci variant. Obsahuje také prvek *ComboBox*, který slouží pro výběr varianty a její následné ohodnocení. Varianty jsou zobrazovány v prvku *DataGridView*.

Pravá část formuláře obsahuje komponentu *GroupBox* s názvem *Results*. Tento prvek obsahuje další prvky pro zobrazení výsledků rozhodování. Po výběru konkrétní metody rozhodování v komponentě *ComboBox*, která se nachází v horní části, a stisku tlačítka *Decide* se výsledky zobrazují do dvou sloupců, resp. jednoho v případě lexikografické metody, pomocí komponenty *ListView*.



Obrázek 21: Uživatelské prostředí aplikace s daty

Podrobnější popis uživatelského prostředí a návod k použití aplikace je součástí přílohy B.

3.2 Vlastní výsledky

Pro demonstrování výsledků jednotlivých metod byl zvolen modelový příklad výběru mobilního telefonu. Rozhodovatel, který je zároveň i zadavatelem úlohy, identifikoval celkem sedm kritérií, podle kterých se bude rozhodovat. Byly určeny také maximální, resp. minimální hodnoty, které varianty musí nabývat. Seznam kritérií a jejich maximálních, resp. minimálních hodnot je následující:

- cena maximálně 10 000 Kč,
- rozlišení fotoaparátu minimálně 48 Mpix,
- uživatelská paměť minimálně 64 GB,
- paměť RAM minimálně 4096 MB,
- minimální verze operačního systému Android 8,
- váha maximálně 210 g,
- kapacita baterie minimálně 4500 mAh.

Rozhodovatel bude mobilní telefon využívat k fotografování, proto jsou pro něj rozhodující rozlišení fotoaparátu a paměť telefonu. Nezanedbatelným kritériem je také cena mobilního telefonu, verze operačního systému, jeho váha a kapacita baterie. Frekvence procesoru nebo rozlišení displeje nejsou pro rozhodovatele rozhodujícími kritérii vzhledem k plánovanému využití telefonu.

Na základě výše uvedených parametrů bylo vybráno celkem pět mobilních telefonů. Název mobilního telefonu a sledované parametry jsou uvedené v tabulce 3.

Tabulka 3: Údaje o vybraných mobilních telefonech

Název mobilního telefonu	Cena	Rozlišení fotoaparátu	Uživatelská paměť	Paměť RAM	Verze operačního systému	Váha	Kapacita baterie
Samsung Galaxy A42	8990 Kč	48 Mpix	128 GB	6144 MB	10	193 g	5000 mAh
Xiaomi Mi Note 10 Lite	7390 Kč	64 Mpix	64 GB	6144 MB	10	204 g	5260 mAh
Xiaomi Redmi Note 10	5490 Kč	48 Mpix	128 GB	4096 MB	11	179 g	5000 mAh
Realme 7i	3299 Kč	48 Mpix	64 GB	4096 MB	10	208 g	6000 mAh
Samsung Galaxy A52	8990 Kč	64 Mpix	128 GB	6144 MB	11	189 g	4500 mAh

Všechny údaje v tabulce 3 byly získány s využitím informací v [10] až [14]. Ceny uvedené v tabulce 3 jsou ceny uvedené za daný mobilní telefon v internetovém obchodě maloobchodního prodejce elektronického zboží DATART [15] ke dni 18. 4. 2021.

Následujícím krokem je ohodnocení jednotlivých kritérií. Tento krok je proveden pomocí bodovací metody se zvolenou bodovou stupnicí 1-10. Kritéria byla ohodnocena následovně:

Tabulka 4: Ohodnocení kritérií

Kritérium	Počet bodů
Cena	10
Rozlišení fotoaparátu	9
Uživatelská paměť	7
Paměť RAM	8
Verze operačního systému	3
Váha	5
Kapacita baterie	4

Všechny výše uvedené informace jsou již dostačující pro vyhodnocení rozhodovacího problému v aplikaci pomocí jedné ze čtyř metod. Prvními kroky, které jsou pro všechny metody stejné, jsou určení normovaných vah kritérií a převedení hodnot v minimalizačních kritériích. Číselné údaje v tabulkách 5-13 jsou zaokrouhleny na pět desetinných míst.

Tabulka 5: Normované váhy kritérií

Kritérium	Váha
Cena	0,21739
Rozlišení fotoaparátu	0,19565
Uživatelská paměť	0,15217
Paměť RAM	0,17391
Verze operačního systému	0,06522
Váha	0,1087
Kapacita baterie	0,08696

Tabulka 6 zobrazuje hodnoty v jednotlivých variantách po převedení hodnot v minimalizačních kritériích. Změna proběhla u dvou kritérií, a to u kritéria cena a váha.

Tabulka 6: Matice s převedenými hodnotami v minimalizačních kritériích

Název	Cena	Rozlišení fotoaparátu	Uživatelská paměť	Paměť RAM	Verze operačního systému	Váha	Kapacita baterie
Samsung Galaxy A42	-8990	48	128	6144	10	-193	5000
Xiaomi Mi Note 10 Lite	-7390	64	64	6144	10	-204	5260
Xiaomi Redmi Note 10	-5490	48	128	4096	11	-179	5000
Realme 7i	-3299	48	64	4096	10	-208	6000
Samsung Galaxy A52	-8990	64	128	6144	11	-189	4500

3.2.1 Lexikografická metoda

Při využití lexikografické metody byly varianty seřazeny:

1. Realme 7i,
2. Xiaomi Redmi Note 10,
3. Xiaomi Mi Note 10 Lite,
4. Samsung Galaxy A52,
5. Samsung Galaxy A42.

Rozhodujícím kritériem bylo kritérium cena, proto aplikace doporučila ke koupi nejlevnější mobilní telefon Realme 7i. V případě mobilních telefonů značky Samsung, které mají stejnou cenu, se pořadí určovalo podle druhého nejvýznamnějšího kritéria – rozlišení fotoaparátu.

3.2.2 Metoda ORESTE

Při použití metody ORESTE se nejprve ohodnotí jednotlivá kritéria jejich pořadovými čísly dle jejich váhy. Při stejné hodnotě váhy se kritérium ohodnotí průměrným pořadovým číslem.

Tabulka 7: Ohodnocení kritérií pořadím

Kritérium	Pořadí
Cena	1
Rozlišení fotoaparátu	2
Uživatelská paměť	4
Paměť RAM	3
Verze operačního systému	7
Váha	5
Kapacita baterie	6

Následujícími kroky jsou určení pořadí variant dle jednotlivých kritérií a vypočítání matice D . Matice byla počítána s parametrem $r = 3$.

Tabulka 8: Matice vzdáleností D

Varianta	Cena	Rozlišení fotoaparátu	Uživatelská paměť	Paměť RAM	Verze OS	Váha	Kapacita baterie
Samsung Galaxy A42	3,58467	3,30193	3,30193	2,59625	5,88195	4,23582	5,05848
Xiaomi Mi Note 10 Lite	2,41014	1,78501	4,26466	2,59625	5,88195	4,55488	4,82028
Xiaomi Redmi Note 10	1,65096	3,30193	3,30193	3,89437	5,57407	3,97906	5,05848
Realme 7i	1	3,30193	4,26466	3,89437	5,88195	5	4,76954
Samsung Galaxy A52	3,58467	1,78501	3,30193	2,59625	5,57407	4,05142	5,54508

Posledním krokem je ohodnocení matice D pořadovými čísly a poté vypočítat pro jednotlivé varianty součet řádku. Podle tohoto součtu se pak seřadí varianty.

Tabulka 9: Ohodnocená matice R s řádkovými součty

Varianta	Cena	Rozlišení fotoaparátu	Uživatelská paměť	Paměť RAM	Verze OS	Váha	Kapacita baterie	Součet
Samsung Galaxy A42	15,5	11,5	11,5	7	34	21	28,5	129
Xiaomi Mi Note 10 Lite	5	3,5	22,5	7	34	24	26	122
Xiaomi Redmi Note 10	2	11,5	11,5	17,5	31,5	19	28,5	121,5
Realme 7i	1	11,5	22,5	17,5	34	27	25	138,5
Samsung Galaxy A52	15,5	3,5	11,5	7	31,5	20	30	119

Pořadí variant při použití metody ORESTE je:

1. Samsung Galaxy A52,
2. Xiaomi Redmi Note 10,
3. Xiaomi Mi Note 10 Lite,
4. Samsung Galaxy A42,
5. Realme 7i.

3.2.3 Metoda TOPSIS

Při využití metody TOPSIS se vypočítává normalizovaná vážená kritériální matice W . Její hodnoty u modelového příkladu jsou uvedené v tabulce 10.

Tabulka 10: Normalizovaná vážená matice W

Varianta	Cena	Rozlišení fotoaparátu	Uživatelská paměť	Paměť RAM	Verze OS	Váha	Kapacita baterie
Samsung Galaxy A42	-0,12184	0,07641	0,08134	0,08819	0,02801	-0,04814	0,03757
Xiaomi Mi Note 10 Lite	-0,10016	0,10189	0,04067	0,08819	0,02801	-0,05089	0,03953
Xiaomi Redmi Note 10	-0,07441	0,07641	0,08134	0,05879	0,03082	-0,04465	0,03757
Realme 7i	-0,04471	0,07641	0,04067	0,05879	0,02801	-0,05188	0,04509
Samsung Galaxy A52	-0,12184	0,10189	0,08134	0,08819	0,03082	-0,04715	0,03382

Po určení ideální varianty $h = (-0,04471, 0,10189, 0,08819, 0,03082, -0,04465, 0,04509)$ a bazální varianty $d = (-0,12184, 0,07641, 0,04067, 0,05879, 0,02801, -0,05188, 0,03382)$ se vypočítají vzdálenosti od těchto variant a následně relativní index, podle kterého se varianty seřadí.

Tabulka 11: Ohodnocení variant vzdálenostmi

Varianta	Vzdálenost d^+	Vzdálenost d^-	Relativní index
Samsung Galaxy A42	0,0817	0,05046	0,38182
Xiaomi Mi Note 10 Lite	0,06932	0,04491	0,39313
Xiaomi Redmi Note 10	0,04951	0,06307	0,56025
Realme 7i	0,05681	0,07795	0,57845
Samsung Galaxy A52	0,07799	0,05654	0,42029

Pořadí variant při využití metody TOPSIS je:

1. Realme 7i,
2. Xiaomi Redmi Note 10,
3. Samsung Galaxy A52,
4. Xiaomi Mi Note 10 Lite,
5. Samsung Galaxy A42.

3.2.4 Metoda váženého součtu

Po určení ideální varianty $h = (-3299, 64, 128, 6144, 11, -179, 6000)$ a bazální varianty $d = (-8990, 48, 64, 4096, 10, -208, 4500)$ se vypočítá standardizovaná kritériální matice R .

Tabulka 12: Standardizovaná matice R

Varianta	Cena	Rozlišení fotoaparátu	Uživatelská paměť	Paměť RAM	Verze OS	Váha	Kapacita baterie
Samsung Galaxy A42	0	0	1	1	0	0,51724	0,33333
Xiaomi Mi Note 10 Lite	0,28115	1	0	1	0	0,13793	0,50667
Xiaomi Redmi Note 10	0,61501	0	1	0	1	1	0,33333
Realme 7i	1	0	0	0	0	0	1
Samsung Galaxy A52	0	1	1	1	1	0,65517	0

Nakonec se vypočítá hodnota funkce užitku pro každou variantu, podle které se varianty následně seřadí.

Tabulka 13: Užitek jednotlivých variant

Varianta	Užitek
Samsung Galaxy A42	0,41129
Xiaomi Mi Note 10 Lite	0,48973
Xiaomi Redmi Note 10	0,48877
Realme 7i	0,30435
Samsung Galaxy A52	0,65817

Pořadí variant při použití metody váženého součtu je:

1. Samsung Galaxy A52,
2. Xiaomi Mi Note 10 Lite,
3. Xiaomi Redmi Note 10,
4. Samsung Galaxy A42,
5. Realme 7i.

3.2.5 Porovnání výsledků metod

Z metod vyžadujících ordinální informaci byla aplikována lexikografická metoda a metoda ORESTE. Tyto metody seřadí varianty od nejlepší po nejhorší. Pořadí variant uspořádané podle těchto dvou metod v modelovém příkladě se v některých variantách výrazně lišilo.

Tabulka 14: Porovnání pořadí variant podle lexikografické metody a metody ORESTE

Lexikografická metoda	Metoda ORESTE
Realme 7i	Samsung Galaxy A52
Xiaomi Redmi Note 10	Xiaomi Redmi Note 10
Xiaomi Mi Note 10 Lite	Xiaomi Mi Note 10 Lite
Samsung Galaxy A52	Samsung Galaxy A42
Samsung Galaxy A42	Realme 7i

Lexikografická metoda doporučila ke koupi nejlevnější variantu. Naopak nejdražším variantám byla udělena poslední místa v pořadí. Výhoda této metody spočívá v její jednoduchosti použití. Naopak nevýhodou je úplné zanedbání méně významných kritérií. Se zvyšujícím se počtem kritérií v úloze se zanedbává více kritérií, je tedy vhodnější pro použití u méně rozsáhlých rozhodovacích problémů.

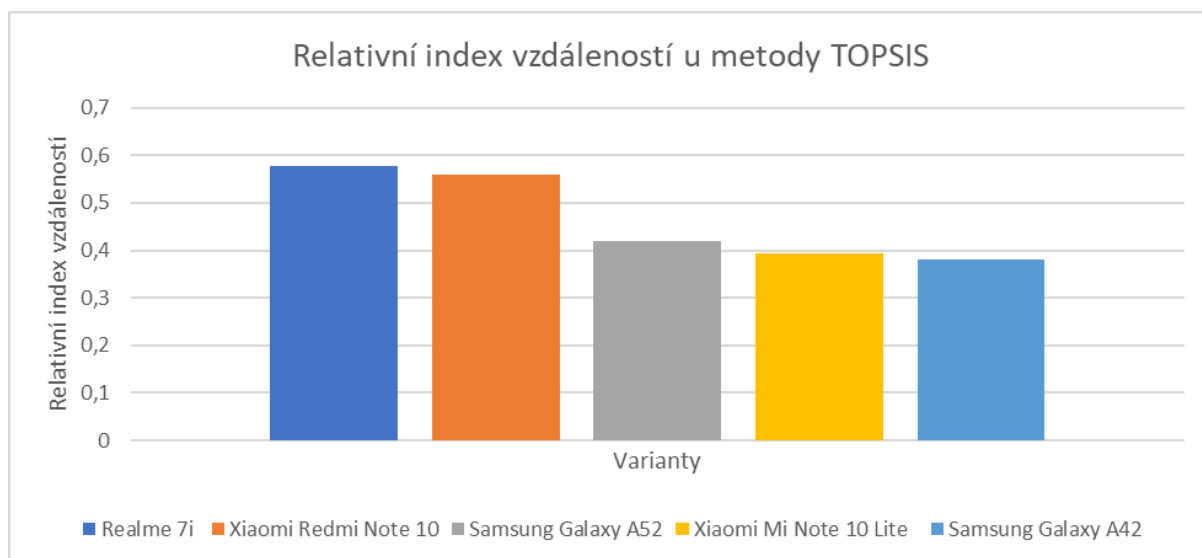
Metoda ORESTE doporučila ke koupi jednu z nejdražších variant. Naopak nejlevnější variantu umístila na poslední místo. Ostatní varianty seřadily obě metody víceméně stejně. Důvodem umístění mobilního telefonu Samsung Galaxy A52 na první místo je fakt, že tento telefon je u ostatních kritérií hodnocen lépe než zbývající varianty. Výhodou metody ORESTE oproti přechodí metodě je zahrnutí všech hodnot variant v kritériích do výpočtu. Nevýhodou metody je zastírání rozdílů mezi hodnotami variant v kritériích, protože jednotlivé hodnoty jsou ohodnoceny pořadovými čísly.

Z metod vyžadujících kardinální informaci byla aplikována metoda TOPSIS a metoda váženého součtu. Obě z těchto metod seřadily varianty výrazně odlišně.

Tabulka 15: Porovnání pořadí variant podle metody TOPSIS a metody váženého součtu

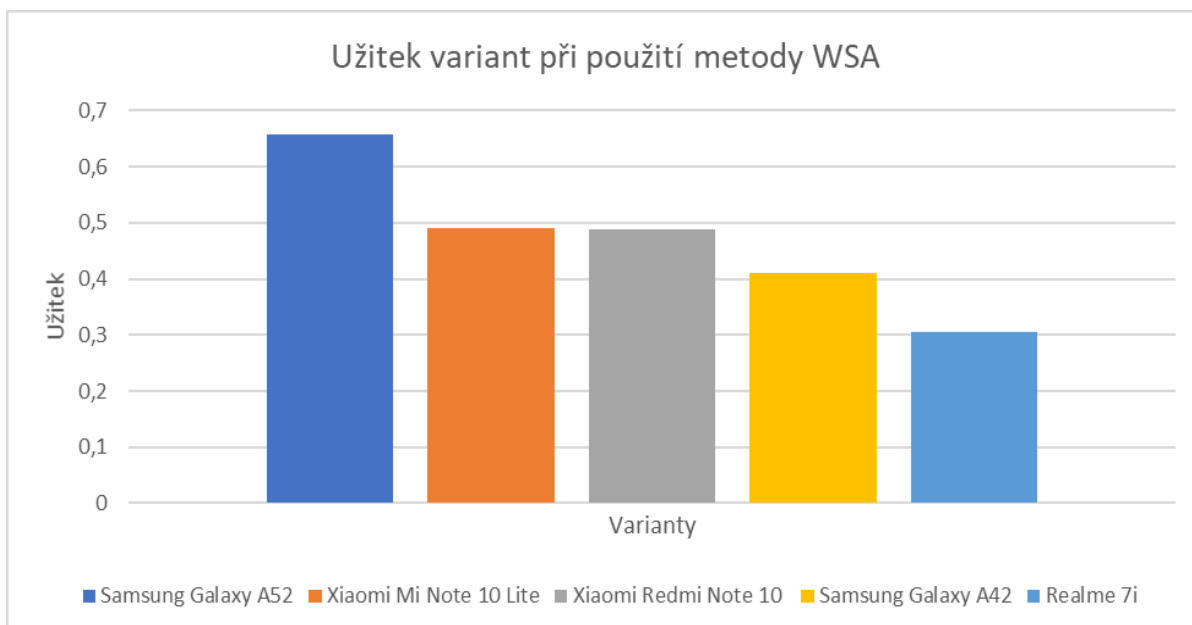
Metoda TOPSIS	Metoda váženého součtu
Realme 7i	Samsung Galaxy A52
Xiaomi Redmi Note 10	Xiaomi Mi Note 10 Lite
Samsung Galaxy A52	Xiaomi Redmi Note 10
Xiaomi Mi Note 10 Lite	Samsung Galaxy A42
Samsung Galaxy A42	Realme 7i

Metoda TOPSIS v modelovém příkladu doporučila ke koupi variantu Realme 7i, avšak rozdíl mezi prvním a druhým místem není nijak výrazný. Obě varianty měly podobnou vzdálenost od ideální varianty, ale první varianta měla větší vzdálenost od bazální varianty. Nejvýraznější rozdíl v relativním indexu vzdáleností je mezi druhým a třetím místem.



Obrázek 22: Relativní index vzdáleností u metody TOPSIS

Metoda váženého součtu doporučila ke koupi variantu Samsung Galaxy A52 navzdory její vysoké ceně. Je to způsobené tím, že většina jeho parametrů patří mezi ty nejlepší a byly ohodnoceny hodnotou 1 v standardizované matici R . Naopak tato metoda umístila nejlevnější variantu Realme 7i na poslední místo z důvodu jejich horších hodnot v ostatních kritériích, které byly ohodnoceny hodnotou 0 ve standardizované matici R .



Obrázek 23: Užitek variant při použití metody WSA

3.3 Shrnutí

Všechny výsledky, které byly uvedeny výše, byly vypočítány s využitím aplikace Decision Support System. Aplikace obsahuje knihovnu DLL, která zajišťuje proces rozhodování a strukturu rozhodovacího problému, a uživatelské prostředí. Knihovna DLL může být využita nejen aplikací Decision Support System, ale může být základem i jiného nového systému pro podporu rozhodování. Jako případné rozšíření této knihovny se nabízí přidání dalších rozhodovacích metod nebo přidání stanovení vah kritérií pomocí metod založených na párovém porovnávání.

Aplikace je použitelná na jakýkoli rozhodovací problém. Uživatel může do aplikace přidat neomezené množství kritérií a neomezené množství variant. Omezením může být možnost výběru ze čtyř metod, tento počet je ale pro běžné rozhodovací situace dostačující. V případě, že by uživatel chtěl využít jinou metodu rozhodování, která není implementována, musí být doplněna do knihovny.

Výsledky jednotlivých metod ukazují, že každá metoda určí pořadí variant odlišně. Je to způsobené jejich odlišným přístupem k řešení rozhodovacího problému. Při řešení rozhodovacího problému je tedy žádoucí využít více než jednu metodu, aby si rozhodovatel mohl vytvořit komplexnější pohled na rozhodovací problém.

ZÁVĚR

Cílem této bakalářské práce bylo popsání základních pojmů a metod z vícekritériálního rozhodování, na tomto základu implementovat vybrané metody do aplikace vytvořené ve vyšším programovacím jazyce a následně vytvořit v aplikaci modelový příklad rozhodování a porovnat výsledky použitých metod.

Teoretická část popisuje nejprve základní pojmy vícekritériálního rozhodování, které jsou stěžejní pro pochopení dalších kapitol. Dále jsou zde popsány metody stanovení vah kritérií, jejichž aplikace ve většině případů předchází samotnému procesu rozhodování. Metody stanovení vah kritérií jsou rozděleny na metody přímého stanovení vah a metody založené na párovém porovnání. Metody rozhodnutí jsou pak rozděleny do čtyř skupin podle charakteru informace o preferencích mezi kritérii a variantami. K vybraným metodám jsou popsány jejich postupy a matematické vzorce.

Součástí praktické části je aplikace vytvořená v programovacím jazyce C#, která slouží jako systém pro podporu rozhodování. Aplikace implementuje čtyři rozhodovací metody. Uživateli nabízí definici kritérií a variant a následné uspořádání variant s využitím jedné z metod. První část praktické části popisuje případy užití aplikace, její třídící diagramy a její strukturu.

Další kapitoly praktické části popisují výsledky modelového příkladu. Jako modelový příklad byl zvolen výběr mobilního telefonu. Modelový příklad obsahoval sedm kritérií a pět variant, což je dostatečné na demonstraci výsledků jednotlivých metod. Z výsledků je patrné, že každá metoda uvádí jiné pořadí variant. Důvodem jsou rozdílné přístupy každé z metod k rozhodovacímu problému. Je tedy žádoucí, aby rozhodovatel využil pro získání řešení rozhodovacího problému více metod než jednu a vytvořil si tak souhrnný pohled na problém.

Ačkoli systémy pro podporu rozhodování jsou užitečnými nástroji, do svých výpočtů nezapočítávají subjektivní vnímání problému. Poskytují rozhodovateli pouze nabídku řešení a urychlení výpočtů, ale samotné rozhodnutí nakonec vždy záleží na rozhodovateli.

POUŽITÁ LITERATURA

- [1] FIALA, P. *Modely a metody rozhodování*. 2. přeprac. vyd. Praha: Oeconomica, 2008. ISBN: 9788024513454.
- [2] BROŽOVÁ, H., HOUŠKA, M., ŠUBRT, T. *Modely pro vícekriteriální rozhodování*. Praha: Credit, 2003. ISBN 80-213-1019-7.
- [3] BAŽANT, M. *Řešení vybraných provozních problémů osobní železniční stanice v rámci simulačního modelu*. Pardubice, 2009. 95 s. Univerzita Pardubice
- [4] FOTR, J. *Manažerské rozhodování: postupy, metody a nástroje*. Praha: Ekopress, s. r. o., 2006. 409 str. ISBN 80-86929-15-9.
- [5] GRASSEOVÁ, M., MAŠLEJ, M., BRECHTA, B. *Manažerské rozhodování: teoretická východiska a praktické příklady*. Část 1. Brno: Univerzita obrany, 2010. ISBN 978-80-7231-730-1.
- [6] RAMÍK J. *Vícekriteriální rozhodování – analytický hierarchický proces (AHP)*. Karviná: Slezská univerzita v Opavě. 211 str. ISBN: 80-7248-047-2.
- [7] KANISOVÁ, H., MÜLLER, M. *UML srozumitelně*. Brno: Computer Press, 2004. ISBN 80-251-0231-9.
- [8] *Microsoft Docs: Dokumentace k jazyku C#* [online]. [cit. 2021-04-20]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/>
- [9] MAHAPATRA, Sayan. *GeeksForGeeks: Rank of all elements in an array* [online]. 7. 1. 2018 [cit. 2021-04-19]. Dostupné z: <https://www.geeksforgeeks.org/rank-elements-array/>
- [10] *Heureka: Samsung Galaxy A42 A426B 5G 6GB/128GB Dual SIM* [online]. [cit. 2021-04-20]. Dostupné z: <https://mobilni-telefony.heureka.cz/samsung-galaxy-a42-a426b-5g-6gb-128gb-dual-sim/>
- [11] *Heureka: Xiaomi Mi Note 10 Lite 6GB/64GB* [online]. [cit. 2021-04-20]. Dostupné z: <https://mobilni-telefony.heureka.cz/xiaomi-mi-note-10-lite-6gb-64gb/>
- [12] *Heureka: Xiaomi Redmi Note 10 4GB/128GB* [online]. [cit. 2021-04-20]. Dostupné z: <https://mobilni-telefony.heureka.cz/xiaomi-redmi-note-10-4gb-128gb/>
- [13] *Heureka: Realme 7i 4GB/64GB* [online]. [cit. 2021-04-20]. Dostupné z: <https://mobilni-telefony.heureka.cz/realme-7i-4gb-64gb/>
- [14] *Heureka: Samsung Galaxy A52 A525F 6GB/128GB* [online]. [cit. 2021-04-20]. Dostupné z: <https://mobilni-telefony.heureka.cz/samsung-galaxy-a52-a525f-6gb-128gb/>
- [15] *Datart* [online]. [cit. 2021-04-18]. Dostupné z: <https://www.datart.cz/>

PŘÍLOHY

Příloha A – CD	49
Příloha B – Uživatelský manuál aplikace Decision Support System	50

PŘÍLOHA A – CD

Přiložené CD obsahuje následující složky:

- Aplikace – obsahuje spustitelný soubor,
- Bakalářská práce – obsahuje soubor s textem této práce,
- Příklady – obsahuje soubory s modelovými příklady rozhodování,
- Uživatelský manuál – obsahuje uživatelský manuál k aplikaci,
- Zdrojové kódy – obsahuje zdrojové kódy aplikace.

PŘÍLOHA B – UŽIVATELSKÝ MANUÁL APLIKACE DECISION SUPPORT SYSTEM

Příloha B obsahuje uživatelský manuál aplikace Decision Support System.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Uživatelský manuál aplikace Decision Support System
Nikola Jačková

2021

OBSAH

Seznam obrázků	3
Úvod	4
1 Aplikace	5
1.1 Horní menu	5
1.2 Část Project	7
1.3 Část Criteria	7
1.4 Část Variants.....	9
1.5 Část Results.....	11
2 Logovací soubor	12
Závěr	13

SEZNAM OBRÁZKŮ

Obrázek 1: Hlavní okno aplikace.....	5
Obrázek 2: Horní menu, položka File.....	6
Obrázek 3: Horní menu, položka Help	6
Obrázek 4: Část Project	7
Obrázek 5: Criterions, přidání kritéria	8
Obrázek 6: Criterions, úprava názvu kritéria.....	8
Obrázek 7: Criterions, označený řádek pro smazání	9
Obrázek 8: Variants, přidání varianty	9
Obrázek 9: Variants, ohodnocení varianty	10
Obrázek 10: Results, zobrazení výsledků	11
Obrázek 11: Obsah logovacího souboru	12

ÚVOD

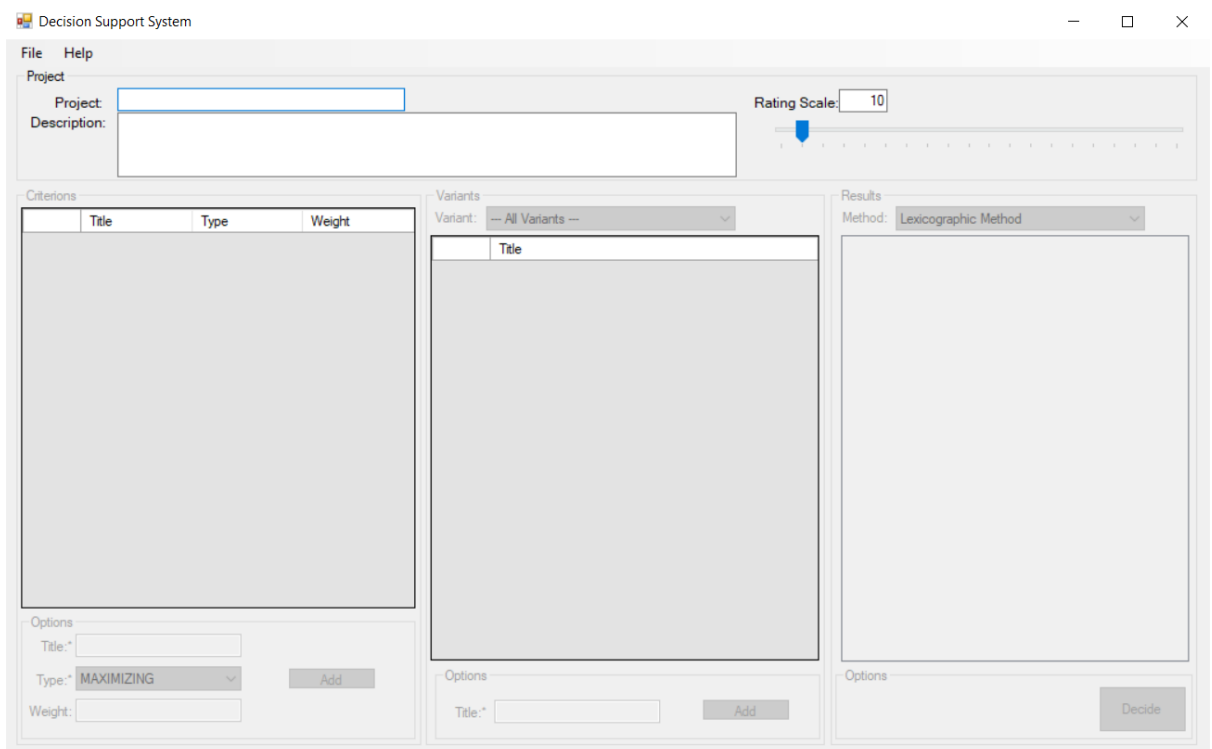
Uživatelský manuál popisuje spuštění aplikace Decision Support System a poskytuje návod k jejímu použití. Aplikace Decision Support System je součástí bakalářské práce na téma metody vícekriteriálního rozhodování a jejich aplikace. Umožňuje uživatelům definovat si kritéria a varianty a následně učinit rozhodnutí dle vybrané metody. Aplikace implementuje čtyři metody vícekriteriálního rozhodování.

1 APLIKACE

Aplikaci lze spustit otevřením souboru *DecisionSupportSystem.exe*, který je součástí bakalářské práce a lze jej nalézt na přiloženém CD.

Po spuštění aplikace se zobrazí hlavní formulářové okno aplikace. Formulář je rozdělen na čtyři hlavní části:

- Project,
- Criteria,
- Variants,
- Results.



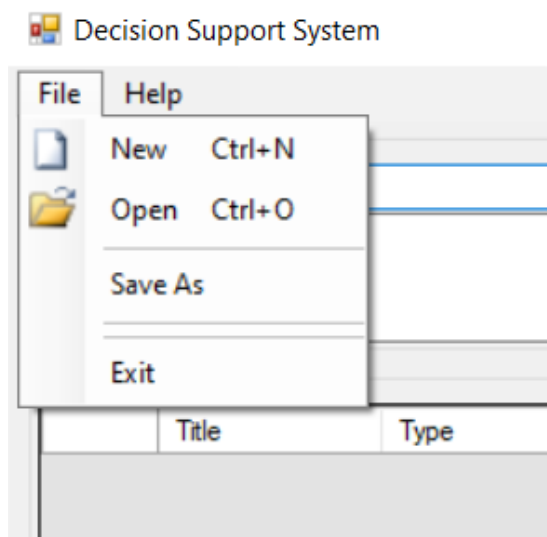
Obrázek 1: Hlavní okno aplikace

Aplikace po spuštění povoluje přístup pouze do části Project a do položek horního menu. Pro zpřístupnění dalších částí je potřeba vyplnit název projektu.

1.1 Horní menu

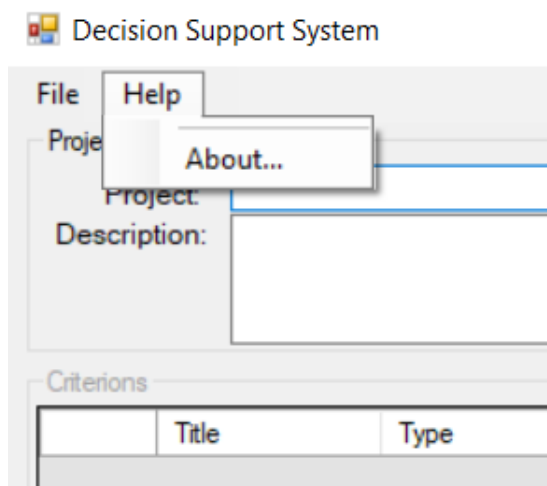
Aplikace obsahuje dvě položky v horním menu:

- File,
- Help.



Obrázek 2: Horní menu, položka File

První položka, položka New, vytváří nový, prázdný projekt. Další položka Open zobrazí dialog pro zvolení souboru, který obsahuje uložený projekt. Tyto soubory pro správnou funkčnost musí být vytvořeny samotnou aplikací a mají koncovku *.dat*. Po potvrzení dialogu se do aplikace načte daný projekt. Položka Save As zobrazí dialog pro výběr místa uložení souboru s daty k aktuálnímu projektu. Po potvrzení dialogu uloží všechny informace o projektu do souboru s koncovkou *.dat*. Poslední položka Exit ukončí aplikaci.

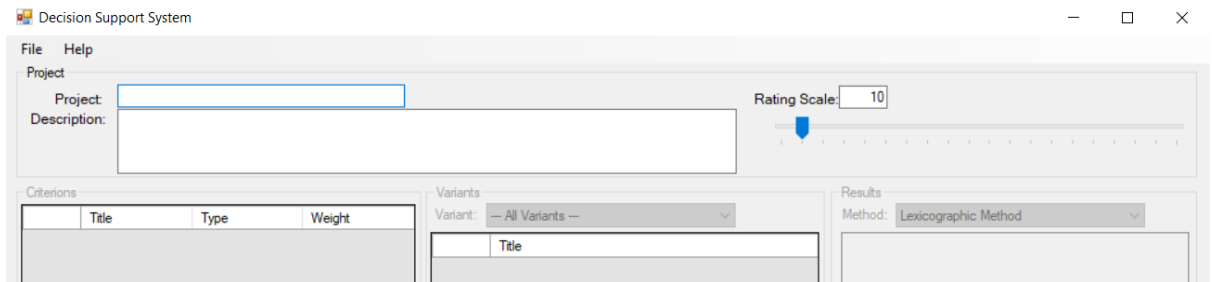


Obrázek 3: Horní menu, položka Help

Druhá položka Help obsahuje jednu položku About..., která po jejím zvolení zobrazí dialogové okno s informací o aplikaci.

1.2 Část Project

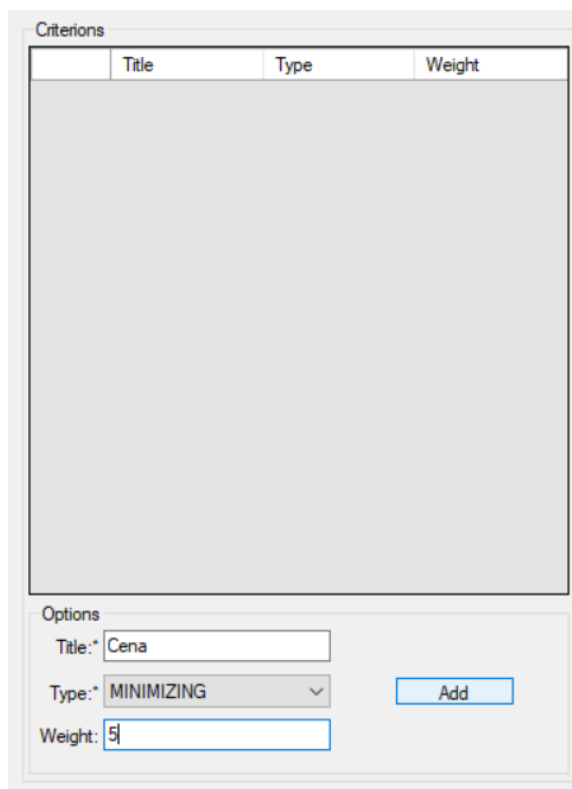
Část Project, která se nachází v horní části formulářového okna, slouží pro nastavování atributů samotného rozhodovacího projektu. První textové pole slouží pro vyplnění názvu projektu. Toto pole musí být vyplněné, pokud chce uživatel pokračovat do dalších částí aplikace. Druhé textové pole slouží pro vyplnění popisu projektu. V pravé části se nachází posuvník, který slouží pro určení bodové stupnice pro určování vah kritérií. Tuto hodnotu lze nastavit buď posuvníkem, nebo vepsáním hodnoty do textového pole nad posuvníkem.



Obrázek 4: Část Project

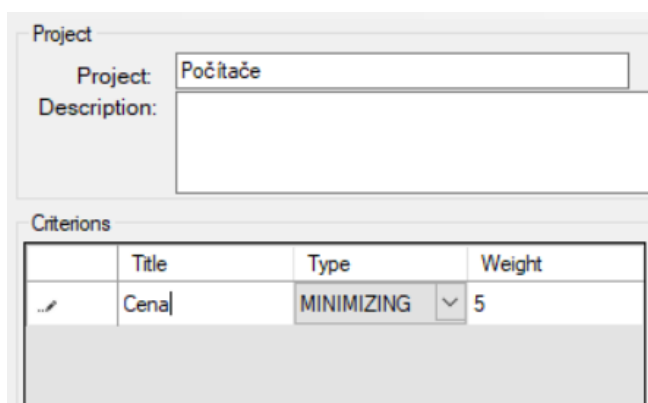
1.3 Část Criteria

Část Criteria slouží pro vytváření jednotlivých kritérií a ohodnocení kritérií váhou v rozmezí určené bodové stupnice. Kritérium se přidá po stisknutí tlačítka Add, které se nachází v dolní části Options. Pro přidání kritéria je povinné vyplnění pole Title, které slouží pro vyplnění názvu kritéria. Pole pod ním slouží pro definování typu kritéria, kde MAXIMIZING znamená maximalizační kritérium a MINIMIZING označuje minimalizační kritérium. Poslední pole Weight slouží pro definování váhy. Toto pole není povinné, pokud zůstane nevyplněné, kritérium bude mít hodnotu váhy 0 a je potřeba ji později upravit na kladnou hodnotu.



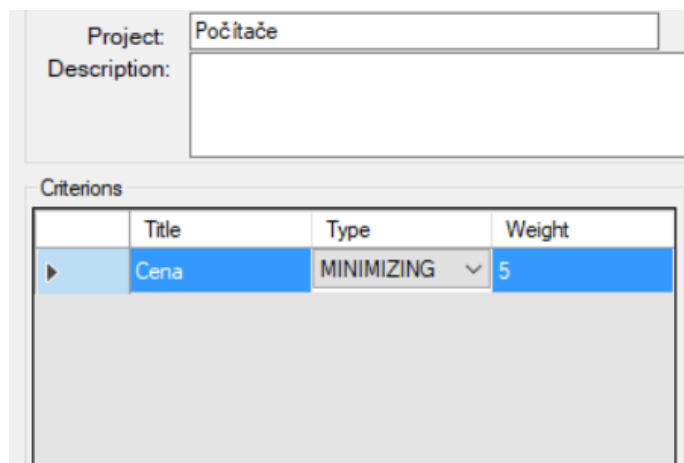
Obrázek 5: Criteria, přidání kritéria

Po přidání kritéria je možné toto kritérium upravit nebo smazat. Obě tyto akce lze provést v tabulce, ve které se kritéria zobrazují. Pro editaci názvu a váhy kritéria je potřeba na danou buňku tabulky kliknout dvakrát. Název kritéria musí být v rámci kritérií jedinečný. Pro editaci typu kritéria je potřeba kliknout na šipku v pravé části této buňky.



Obrázek 6: Criteria, úprava názvu kritéria

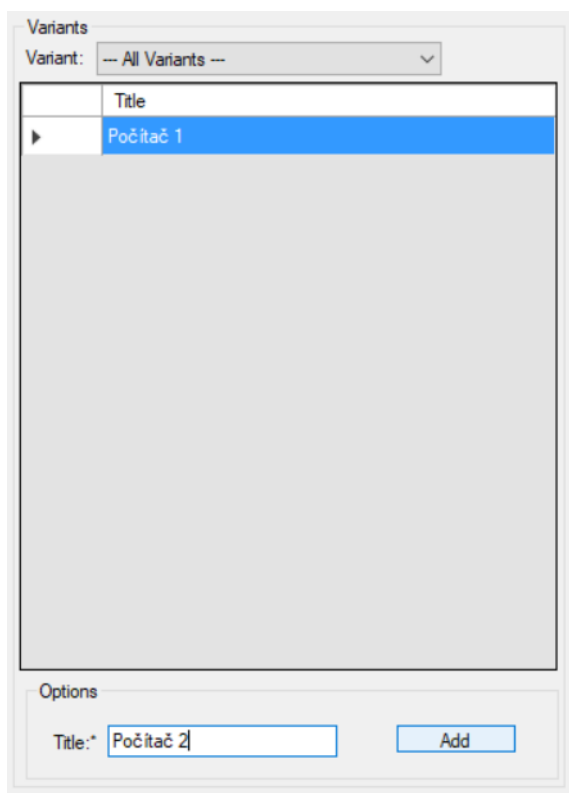
Pro smazání kritéria je potřeba označit celý požadovaný řádek kliknutím na první buňku v řádku. Následně je nutné stisknout klávesu Delete. Nepřípustné stavy např. záporná váha kritéria jsou řešeny vyskočením chybového okna.



Obrázek 7: Criteria, označený řádek pro smazání

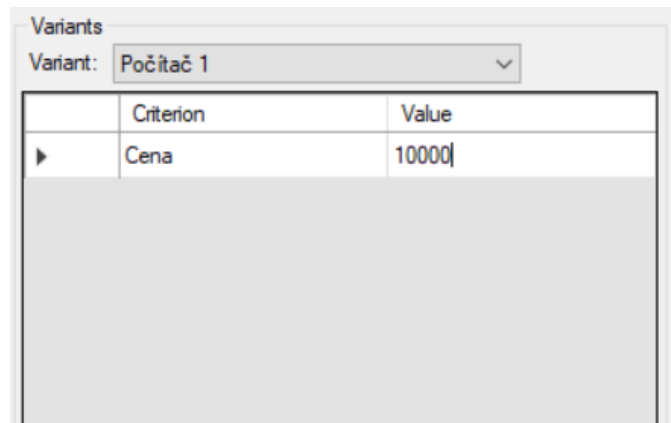
1.4 Část Variants

Část Variants slouží pro vytváření jednotlivých variant a jejich hodnocení v jednotlivých kritériích. Pro přidání varianty je nutné vyplnit pole Title, které slouží pro vyplnění názvu varianty. Toto pole je povinné pro přidání varianty a název varianty musí být v rámci variant jedinečný. Varianta se přidá stisknutím tlačítka Add.



Obrázek 8: Variants, přidání varianty

Editovat a mazat varianty lze stejným způsobem jako kritéria. Pro ohodnocení varianty v jednotlivých kritériích je nutné vybrat danou variantu z rozbalovacího seznamu variant v horní části. Po vybrání varianty se zobrazí tabulka se seznamem kritérií a jejich hodnot pro danou variantu. Úprava hodnot probíhá dvojklikem na danou buňku. V této tabulce nelze upravovat první sloupec a nelze řádky mazat. Pro zobrazení seznamu variant je nutné vybrat v rozbalovacím seznamu možnost --- All Variants ---.

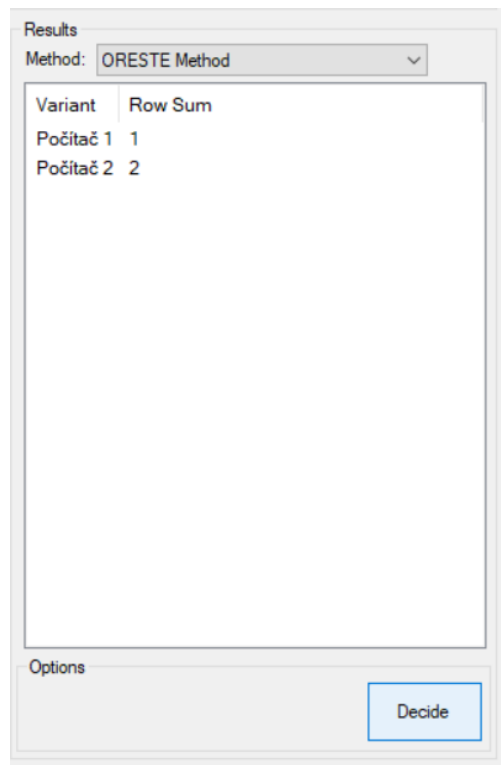


	Criterion	Value
▶	Cena	10000

Obrázek 9: Variants, ohodnocení varianty

1.5 Část Results

Poslední část formulářového okna aplikace slouží pro zobrazování výsledků rozhodování. Pro učinění rozhodnutí je nutné, aby rozhodovací projekt obsahoval alespoň jedno kritérium s nenulovou váhou a alespoň jednu variantu. Pro učinění rozhodnutí se z rozbalovacího seznamu v horní části vybere požadovaná metoda a poté se stiskne tlačítko Decide. V tabulce s výsledky se zobrazí utříděný seznam variant. V případě metody váženého součtu, ORESTE a TOPSIS se navíc zobrazí rozhodující hodnoty, podle kterých je seznam variant seřazený.



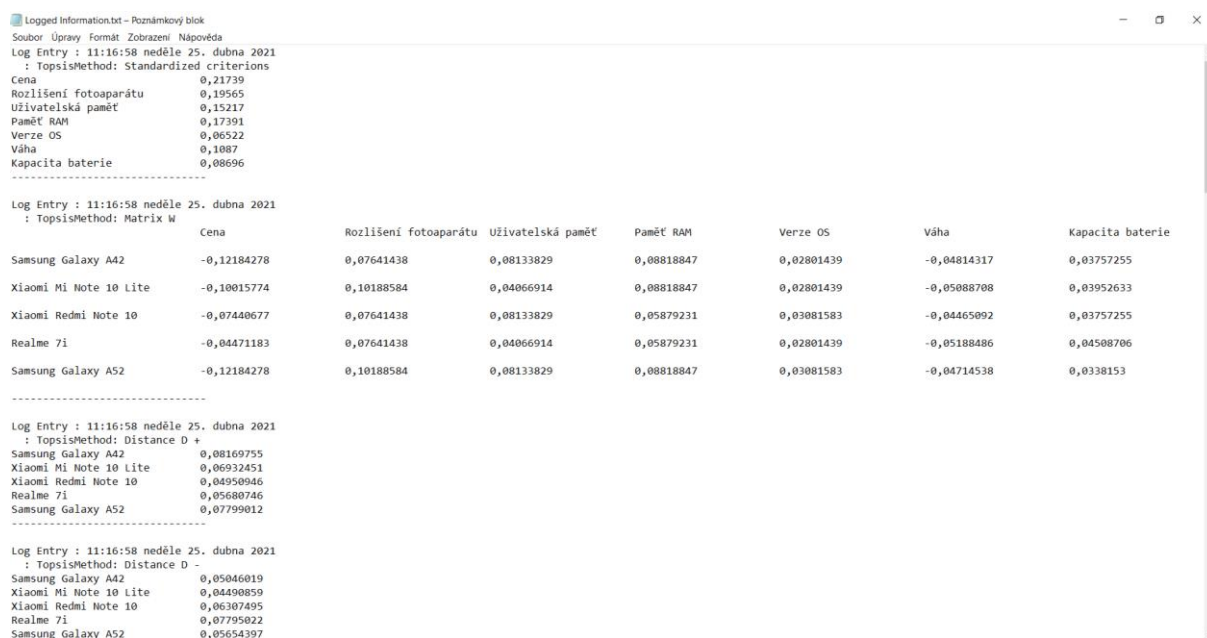
Variant	Row Sum
Počítač 1	1
Počítač 2	2

Obrázek 10: Results, zobrazení výsledků

2 LOGOVACÍ SOUBOR

Aplikace po každém spuštění aplikace automaticky vytvoří logovací soubor, který slouží pro uchování průběžných výsledků rozhodování. Pokud soubor již existuje po předchozím spuštění, jeho obsah je přepsán, pokud tedy uživatel chce průběžné výsledky zachovat, je nutné si je uložit na jiné místo. Logovací soubor se vytváří v adresáři, kde se nachází soubor, ze kterého byla aplikace spuštěna.

Logovací soubor je textový soubor a má název Logged Information. Všechny záznamy obsahují informaci o datumu a času zápisu a metodě rozhodování. Dále pak obsahují číselné údaje průběžných výpočtů v závislosti na použité metodě. Jednotlivé záznamy jsou oddělené několika spojovníky psaných za sebou.



```
Logged Information.txt - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
Log Entry : 11:16:58 neděle 25. dubna 2021
: TopsisMethod: Standardized criterions
Cena 0,21739
Rozlišení fotoaparátu 0,19565
Uživatelská paměť 0,15217
Paměť RAM 0,17391
Verze OS 0,06522
Váha 0,1087
Kapacita baterie 0,08696
-----
Log Entry : 11:16:58 neděle 25. dubna 2021
: TopsisMethod: Matrix W
Cena Rozlišení fotoaparátu Uživatelská paměť Paměť RAM Verze OS Váha Kapacita baterie
Samsung Galaxy A42 -0,12184278 0,07641438 0,08133829 0,08818847 0,02801439 -0,04814317 0,03757255
Xiaomi Mi Note 10 Lite -0,10015774 0,10188584 0,04066914 0,08818847 0,02801439 -0,05088708 0,03952633
Xiaomi Redmi Note 10 -0,07440677 0,07641438 0,08133829 0,05879231 0,03081583 -0,04465092 0,03757255
Realme 7i -0,04471183 0,07641438 0,04066914 0,05879231 0,02801439 -0,05188486 0,04508706
Samsung Galaxy A52 -0,12184278 0,10188584 0,08133829 0,08818847 0,03081583 -0,04714538 0,0338153
-----
Log Entry : 11:16:58 neděle 25. dubna 2021
: TopsisMethod: Distance D +
Samsung Galaxy A42 0,08169755
Xiaomi Mi Note 10 Lite 0,06932451
Xiaomi Redmi Note 10 0,04950946
Realme 7i 0,05680746
Samsung Galaxy A52 0,07799012
-----
Log Entry : 11:16:58 neděle 25. dubna 2021
: TopsisMethod: Distance D -
Samsung Galaxy A42 0,05046019
Xiaomi Mi Note 10 Lite 0,04490859
Xiaomi Redmi Note 10 0,06307495
Realme 7i 0,07795022
Samsung Galaxy A52 0,05654397
```

Obrázek 11: Obsah logovacího souboru

ZÁVĚR

Tento uživatelský manuál k aplikaci Decision Support System poskytuje návod k jejímu používání. Slovně i graficky popisuje možnosti aplikace a poskytuje tak dostatečné informace k pochopení používání aplikace. Popisuje také logovací soubor, který slouží pro logování průběžných výpočtů.