

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

FAKULTA INFORMATIKY A STATISTIKY



Nová metoda vícekritériálního hodnocení variant

Autor:	Adéla Paříková
Katedra:	Katedra ekonometrie
Obor:	Matematické metody v ekonomii
Vedoucí práce:	Prof. Dr. Ing. Petr Fiala, CSc.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci na téma „Nová metoda vícekritériálního hodnocení variant“ zpracovala samostatně. Veškerou použitou literaturu a další podkladové materiály uvádím v seznamu použité literatury.

V Praze dne 3. června 2013

.....

Adéla Paříková

Poděkování:

Ráda bych na tomto místě poděkovala prof. Dr. Ing. Petru Fialovi, CSc. za vedení mé bakalářské práce a za podnětné návrhy, které ji obohatily. Zároveň bych ráda poděkovala Bc. Petru Poláchovi za pomoc při vytvoření softwarové aplikace, bez níž bych tuto práci nemohla dokončit.

Abstrakt

Název práce: Nová metoda vícekritériálního hodnocení variant

Autor: Adéla Paříková

Katedra: Katedra ekonometrie

Vedoucí práce: Prof. Dr. Ing. Petr Fiala, CSc.

V této práci představím jednu z nových metod teorie rozhodování. Jedná se o diskrétní metodu řadící se do skupiny metod minimalizujících vzdálenost od ideální varianty. Nový algoritmus nabízí hodnocení variant nejen podle důležitosti hodnot kritérií ale i podle důležitosti rozdílu mezi jednotlivými hodnocenými variantami. Postup pracuje na základě generování všech hypotetických variant, které lze vytvořit kombinacemi hodnot jednotlivých kritérií pro všechny alternativy. Bude vám také nabídnuta modifikace této metody, která sice zjednodušuje původní postup, ale na druhou stranu bude využitelná v praktických aplikacích. V rámci tvorby této práce byla vytvořena softwarová aplikace, která tuto metodu využívá.

Klíčová slova: teorie rozhodování, diskrétní modely, vícekritériální hodnocení variant, nová metoda

Abstract

Title: New method of multi-criteria assessment analysis

Author: Adéla Paříková

Department: Department of Econometrics

Supervisor: Prof. Dr. Ing. Petr Fiala, CSc.

In this paper I will introduce one of new methods in the field of decision making theory. It is a discrete method from the group of methods, which minimize the distance from the ideal alternative. The new algorithm not only offers an assessment of the alternatives by the importance of the criteria values, but also by the importance of the difference between the evaluated options. The procedure works by generating all hypothetical variants that are created from combinations of the criteria values for all alternatives. You will also be offered a modification of this method. Though the new method simplifies the original procedure, on the other hand it can be used in practical applications. During writing this paper, there was a software application created, which uses this method to calculate decision problems.

Keywords: decision making theory, discrete models, multi-criteria evaluation, new method

Obsah

ÚVOD	1
1 TEORIE ROZHODOVACÍCH PROCESŮ	2
1.1 ZÁKLADNÍ POJMY	2
1.2 KRITÉRIA HODNOCENÍ.....	3
1.3 MODELOVÁNÍ PREFERENCÍ.....	4
1.4 METODY VOLBY VAH KRITÉRIÍ.....	4
1.5 METODY VÍCEKRITÉRIÁLNÍHO HODNOCENÍ VARIANT	7
1.5.1 Metody s aspiračními úrovněmi	7
1.5.2 Metody s ordinální informací	7
1.5.3 Metody s kardinální informací	8
1.5.4 Přístupy hodnocení na základě preferenční relace	10
2 NOVÁ METODA VÍCEKRITÉRIÁLNÍHO HODNOCENÍ VARIANT	13
2.1 ALGORITMUS	13
2.2 MODIFIKACE NOVÉ METODY	17
2.2.1 Modifikovaný algoritmus	18
2.3 UKÁZKOVÝ PŘÍKLAD	20
3 SROVNÁNÍ S JINÝMI DISKRÉTNÍMI METODAMI.....	23
3.1 PŘÍKLAD 1 (4x4).....	23
3.2 PŘÍKLAD 2 (6x5).....	25
3.3 PŘÍKLAD 3 (5x10).....	26
3.4 PŘÍKLAD 4 (9x10).....	28
3.5 PŘÍKLAD 5 (20x50).....	29
3.6 SHRNUTÍ	30
4 SOFTWARE.....	31
4.1 TECHNICKÉ PARAMETRY	31
4.2 UŽIVATELSKÁ PŘÍRUČKA	31
ZÁVĚR.....	34
LITERATURA	35
PŘÍLOHY	38
PŘÍLOHA 1	38
PŘÍLOHA 2	47

Úvod

Každý den lidí provází rozhodnutí o rozličných problémech. Někdy jednoduchá jako, kde si dají oběd nebo co si obléknou na sebe. Někdy ale stojí před rozhodnutím, které jim může zásadně ovlivnit život. V takových případech už se nedá rozhodnout hned a jen na základě minulých zkušeností. V případě firem jsou tato rozhodnutí ještě složitější, protože ovlivní větší množství lidí a třeba i budoucnost firmy. V této situaci se využívají různé matematické aparáty jako např. metody vícekritériálního rozhodování.

Tyto metody řeší jak diskrétní, tak spojitě rozhodovací problémy. Nová metoda, kterou v této práci představím, náleží mezi diskrétní metody neboli metody vícekritériálního hodnocení variant. Je to jedna z metod, která hledá vzdálenosti od ideální a bazální varianty. Je vhodné je použít u problémů, kde je více kritérií, které sledujeme, protože není-li sledován logický, dobře strukturovaný postup, může dojít ke zmatkům. Další obtíže mohou nastat při rozhodování v týmu, který je složen z odborníků z různých oborů. Každý z nich bude mít jiné priority.

V této práci se budu zabývat jednou z nových metod teorie rozhodování. Jejími autory jsou Mališa Žižovič, Nada Damljanovič, Vera Lazarevič a Nedeljko Deretič. Poprvé ji publikovali ve vědeckém bulletinu University Polytechniky Bukurešť (Universitatea Politehnica București) v roce 2011.

Téma jsem si vybrala, protože jsem o teorii rozhodování znala jen málo a toužila se o něm dozvědět více. Prozkoumání nové metody se zdálo jako dobrá příležitost.

V úvodu práce nejdříve nastíním teorii rozhodování a její základní terminologii a principy. Tato část seznámí čtenáře především se základními pojmy vícekritériálního hodnocení variant, s metodami odhadu vah kritérií a některými jinými metodami patřícími do této kategorie. V další kapitole představím algoritmus nové metody a předvedu jej na jednoduchém příkladu. Následně novou metodu srovnám s jinými často používanými metodami, abych získala objektivní obraz o výsledcích, které podává.

Cílem práce je zjistit, jak nová metoda funguje a při jakých rozhodovacích problémech by bylo vhodné ji aplikovat. Dále chci zjistit, jestli by se vůbec dala využít v praktických situacích nebo jestli zůstávala neznámá z nějakého konkrétního důvodu.

Jako nástroj pro podporu rozhodování bude použit Microsoft Office Excel 2007 s doplňkem Sanna 2009.

1 Teorie rozhodovacích procesů

Rozhodování je součástí každodenního života. Při řešení rozsáhlých problémů člověk není schopen vybrat nejlepší řešení pouze pomocí intuice a nabytých zkušeností. V takových případech nás situace nutí využívat matematické metody, pomocí kterých je možné získat empirické ukazatele. Ty nám při správném nastavení vstupních podmínek a správném vyhodnocení pomohou určit optimální řešení pro zpracováváný rozhodovací proces.

Vícekritériální analýza řeší rozhodovací problémy jako procesy výběru kompromisní varianty ze souboru všech možných variant. Podle toho, jak je zadána množina variant rozlišujeme dvě skupiny vícekritériálních úloh. První, charakterizovanou konečným počtem alternativ, nazýváme vícekritériální hodnocení variant (diskrétní modely) a druhou, pro kterou je typická nekonečně velká množina možných řešení, označujeme jako vícekritériální programování (spojité modely). Tato práce se bude věnovat pouze prvnímu typu problémů a to vzhledem k tomu, že nová metoda patří do této skupiny. Druhý typ tedy zůstane stranou.

Skupina úloh vícekritériálního hodnocení variant je popsána *explicitně seznamem variant, které jsou popsány ohodnocením podle jednotlivých kritérií, což vyjadřuje obsah informace. Informace však mohou mít různou formu. Cílem může být nalezení množiny "dobrých" variant, nalezení varianty, která by podle všech kritérií dosáhla co nejlepšího ohodnocení, nebo uspořádání všech variant.* [2, str. 47].

V procesu rozhodování figurují dva základní typy subjektů - rozhodovatel a analytik. Rozhodovatel problém identifikuje a využívá pak modelu a poskytovaných doporučení při rozhodování. Analytik zpracovává informace od rozhodovatele, vytváří model a na jeho základě doporučuje rozhodovateli řešení.

1.1 Základní pojmy

Jak už zde bylo uvedeno, úlohy vícekritériálního hodnocení variant jsou charakterizovány konečnou množinou variant, konečnou množinou hodnotících kritérií a vazbami mezi nimi.

Seznam variant $A = \{a_1, a_2, \dots, a_m\}$ a seznam kritérií $C = \{c_1, c_2, \dots, c_n\}$ lze zapsat do tzv. **kritériální matice**

$$Y = \begin{matrix} & \begin{matrix} c_1 & \dots & c_n \end{matrix} \\ \begin{matrix} a_1 \\ \vdots \\ a_m \end{matrix} & \begin{pmatrix} y_{11} & \dots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{m1} & \dots & y_{mn} \end{pmatrix} \end{matrix}$$

kde prvky kritériální matice y_{ik} , $i=1,2,\dots,m$, $k=1,2,\dots,n$, představují informace o hodnocení varianty a_i podle kritéria c_k . Na tento standardní tvar je možno převést každou úlohu vícekritériálního hodnocení variant.

Nedominovaná varianta je taková, ke které neexistuje žádná jiná varianta lépe hodnocená alespoň podle jednoho kritéria a současně hodnocená hůře podle zbývajících kritérií. Doplnkem množiny nedominovaných variant je množina **dominovaných variant**.

Ideální varianta $H=(H_1, H_2, \dots, H_m)$, hypotetická či reálně existující, dosahuje ve všech kritériích logicky nejlepší možné hodnoty. Protějškem jí je **varianta bazální** $D=(D_1, D_2, \dots, D_m)$, hypotetická či reálně existující, která dosahuje ve všech kritériích nejhorších hodnot.

Optimální varianta je taková, která je jediná nedominovaná. U většiny rozhodovacích procesů se ale s takovou variantou nesetkáme, proto je zaveden pojem kompromisní varianty. Pokud je nedominovaných variant více, vybereme z nich jednu, kterou považujeme za reprezentativní. Tuto variantu nazveme **variantou kompromisní**.

Kritéria, podle kterých jsou hodnoceny jednotlivé varianty, mohou být vyjádřena v různých jednotkách a různých měřítkách. Abychom na základě těchto kritérií mohli dané varianty srovnávat, potřebujeme je nějakým způsobem transformovat, a k tomu slouží **normalizace**. Při ní jsou prvky kritériální matice transformovány tak, aby nabývaly hodnot v intervalu $(0,1)$. Příklady metod normalizace:

$$\begin{aligned}
 1) \ r_{ik} &= \frac{y_{ik}}{H_k}, & i=1,2,\dots,m, \ k=1,2,\dots,n, \\
 2) \ r_{ik} &= \frac{y_{ik} - D_k}{H_k - D_k}, & i=1,2,\dots,m, \ k=1,2,\dots,n, \\
 3) \ r_{ik} &= \frac{y_{ik}}{\sqrt{\sum_{i=1}^m y_{ik}}}, & i=1,2,\dots,m, \ k=1,2,\dots,n, \\
 4) \ r_{ik} &= \frac{y_{ik}}{\sqrt{\sum_{i=1}^m (y_{ik})^2}}, & i=1,2,\dots,m, \ k=1,2,\dots,n.
 \end{aligned} \tag{1.1}$$

Prvky normalizační matice r_{ik} je možné uspořádat do normalizované kritériální matice **R**.

Volba příslušné normalizační metody pak závisí na konkrétním problému.

1.2 Kritéria hodnocení

Kritéria hodnocení mohou být popsána jako charakteristiky, podle nichž je možné určit, nakolik jednotlivé varianty splňují cíl. Výběr souboru kritérií provádí rozhodovatel v souladu s požadavky na *úplnost*, *neredundanci*, *měřitelnost*, *operacionalitu*, *dekomponovatelnost* a *minimální rozsah*. [3, str. 18] Úplný označujeme soubor, na jehož základě můžeme posoudit

varianty do všech důsledků. Požadavek neredundance zaručí jedinečnost kritérií. Měřitelnost a operacionalita zabezpečí, že všechna kritéria musí být nějakým způsobem měřitelná, srozumitelná a mít jasné definovaný obsah. O nemožnost rozložit kritérium na jednodušší celky se postará požadavek dekomponovatelnosti a předpoklad minimálního rozsahu omezuje počet kritérií na minimum.

Kritéria je možné dělit podle různých hledisek. Podle měřitelnosti rozlišujeme kritéria **kvantitativní** (vyjádřena číselně) a **kvalitativní** (vyjádřena slovním popisem). Podle jejich povahy pak kritéria maximalizační a minimalizační. U maximalizačních kritérií je žádoucí mít co nejvyšší možné hodnoty, u minimalizačních naopak. Při řešení úloh s různými typy kritérií je nutné všechna minimalizační převést na maximalizační. Jedna z možností jak toto provést je odečtení každého prvku ve sloupci minimalizačního kritéria od jeho nejhorší hodnoty (čili nejvyšší dosažené).

1.3 Modelování preferencí

Důležitou součástí modelu rozhodovací situace je i modelování preferencí rozhodovatele. Své představy o důležitosti kritérií může vyjádřit třemi způsoby:

1. aspiračními úrovněmi,
2. ordinálními informacemi,
3. kardinálními informacemi.

Vyjádření pomocí aspiračních úrovní vyžaduje od rozhodovatele udání minimální hodnoty, které by měly varianty v daném kritériu dosáhnout. Jestliže této hodnoty bude dosaženo, varianta je označena za akceptovatelnou.

Ordinální informací o kritériích rozumíme jejich uspořádání od nejdůležitějšího po to nejméně důležité.

Kardinální informace o kritériích je vyjádřena pomocí vah. Každému kritériu je přiřazena relativní váha tak, že čím důležitější kritérium je, tím vyšší váhu dostane. Takto stanoveny váhy v_k , pro $k = 1, 2, \dots, n$ můžeme zapsat pomocí vektoru vah kritérií

$$v = (v_1, v_2, \dots, v_n), \quad \sum_{k=1}^n v_k = 1, \quad v_k \geq 0.$$

1.4 Metody volby vah kritérií

Rozhodovatel často nedokáže určit hodnoty vah přímo, ale dokáže vyjádřit o nich jednodušší subjektivní informace. Tyto informace lze zpracovat pomocí metod konstruujících odhady vah.

První z těchto metod je **metoda pořadí**. Při ní jsou kritéria nejdříve seřazena od nejdůležitějšího po nejméně důležité a poté jsou jim přiřazeny body n (pro nejdůležitější),

$n-1, \dots, 1$ (pro nejméně důležité). Obecně je k -tému kritériu přiřazeno číslo b_k . Váhy vypočítáme podle vzorce:

$$v_k = \frac{b_k}{\sum_{k=1}^n b_k}, \quad k = 1, 2, \dots, n.$$

Pro součet ve jmenovateli platí:

$$\sum_{k=1}^n b_k = \frac{n(n-1)}{2}.$$

Při **bodovací metodě** jsou kritéria ohodnocena kvantitativně body z nějakého předem daného intervalu. Čím důležitější kritérium, tím více bodů dostane. Váhy se stanoví podle stejného vzorce jako při metodě pořadí.

Třetí z metod, metoda párového srovnávání, využívá informace, které ze dvou kritérií je při párovém porovnání důležitější. Každá dvě kritéria jsou srovnána mezi sebou, celkový počet srovnání pak je:

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

Pro větší přehlednost je sestrojován tzv. Fullerův trojúhelník - podle toho se této metodě říká také Fullerova metoda. Než jsou kritéria uspořádána do trojúhelníku, je nutné je očíslovat pořadovými čísly $1, 2, \dots, n$. V trojúhelníkovém schématu jsou ve dvojřádcích uspořádány dvojice pořadových čísel tak, že se každá dvojice vyskytuje pouze jednou. Rozhodovatel pak zakroužkuje to důležitější kritérium z každé dvojice. Počet zakroužkování j -tého kritéria označme n_j . Váha kritéria se poté vypočítá podle vzorce:

$$v_k = \frac{n_i}{N}, \quad k = 1, 2, \dots, n.$$

Saatyho metoda (kvantitativní párové srovnávání) také srovnává páry kritérií a hodnocení pak ukládá do tzv. Saatyho matice $S = (s_{ij})$ kde $i, j = 1, 2, \dots, n$ podle systému:

$$(s_{ij}) = \begin{cases} 1 - \text{rovnocenná kritéria } i \text{ a } j \\ 3 - \text{slabě preferované kritérium } i \text{ před } j \\ 5 - i \text{ je silně preferováno před } j \\ 7 - \text{velmi silně preferované kritérium } i \text{ před } j \\ 9 - \text{absolutně preferované kritérium } i \text{ před } j \end{cases}$$

Hodnoty 2, 4, 6, 8 vyjadřují mezistupně. Pro prvky matice musí platit:

$$s_{ii} = 1, \quad i = 1, 2, \dots, n$$

$$s_{ji} = \frac{1}{s_{ij}}, \quad i = 1, 2, \dots, n$$

Prvky matice jsou interpretovány jako odhady podílu vah i – tého a j – tého kritéria:

$$s_{ij} \cong \frac{v_i}{v_j}, \quad i, j = 1, 2, \dots, n$$

Odhadovaný součet by se měl lišit od skutečného pouze minimálně, proto se snažíme minimalizovat součet čtverců rozdílů

$$F = \sum_{i=1}^n \sum_{j=1}^n (s_{ij} - \frac{v_i}{v_j})^2 \rightarrow \min,$$

za podmínky

$$\sum_{i=1}^n v_i = 1, \quad v_i \geq 0.$$

Při výpočtech kvadratického programování dochází k potížím, a proto se používá metoda geometrického průměru, neboli metoda nejmenších logaritmických čtverců. Snažíme se minimalizovat součet logaritmických čtverců

$$F = \sum_{i=1}^n \sum_{j>i} (\ln s_{ij} - (\ln v_i - \ln v_j))^2 \rightarrow \min,$$

za podmínky

$$\sum_{i=1}^n v_i = 1, \quad v_i \geq 0.$$

Řešením této úlohy pak je normalizovaný geometrický průměr řádků matice S :

$$v_i = \frac{\left[\prod_{j=1}^n s_{ij} \right]^{\frac{1}{n}}}{\sum_{i=1}^n \left[\prod_{j=1}^n s_{ij} \right]^{\frac{1}{n}}}, \quad i = 1, 2, \dots, n.$$

Matice musí být uspokojivě konzistentní. Měřítkem konzistence je odhad rozptylu σ^2 a musí pro něj platit:

$$\sigma^2 < 0,1 \text{ pro } k = 3,$$

$$\sigma^2 < 0,2 \text{ pro } k = 4, 5, 6, 7,$$

$$\sigma^2 < 0,3 \text{ pro } k > 7.$$

Platí $\sigma^2 = F/d$, kde F je hodnota kvadratické formy a d je počet nezávislých srovnání snížený o počet lineárně nezávislých váhových parametrů

$$d = \frac{k(k-1)}{2} - (k-1) = \frac{(k-1)(k-2)}{2}.$$

1.5 Metody vícekritériálního hodnocení variant

V současné době existuje velké množství metod na podporu rozhodování. Liší se od sebe nejen principem výpočtu a výpočetní složitostí, ale i situacemi, ve kterých se používají, nebo nároky kladenými na řešitele. Různé metody mohou také vést k různým cílům. Řešením může být nalezení množiny "dobrých" variant, nalezení varianty, která by podle všech kritérií dosáhla co nejlepšího hodnocení, nebo uspořádání všech variant.

Úlohy vícekritériální analýzy je možné dělit podle několika hledisek. Nejběžnější je dělení podle informace, se kterou metoda pracuje, a tato práce se jej bude držet.

Vzhledem k omezenému prostoru v této práci bude kladen důraz především na metody, se kterými budu "novou metodu" srovnávat. Ostatní metody budou pouze nastíněny s odkazem na literaturu, kde je možné se o nich dozvědět více.

1.5.1 Metody s aspiračními úrovněmi

U těchto metod se vychází z předpokladu, že rozhodovatel zadá hodnoty, kterých by měly varianty alespoň dosahovat při hodnocení podle jednotlivých kritérií, aby pro něj byly varianty akceptovatelné. Do této skupiny metod patří např. konjunktivní metoda, disjunktivní metoda nebo metoda PRIAM.

1.5.2 Metody s ordinální informací

Do této skupiny metod patří např. lexikografická metoda, permutační metoda nebo metoda ORESTE, kterou krátce nastíním.

1.5.2.1 Metoda ORESTE

Tato metoda vyžaduje jako vstup znalost pořadí kritérií a variant, přičemž stačí znalost jejich kvaziuspořádání. Metoda se skládá ze dvou částí. V první z nich určíme vzdálenosti každé varianty podle každého kritéria od fiktivního počátku a poté varianty podle určitých pravidel uspořádáme. Druhou částí je preferenční analýza. Tuto metodu můžeme rozdělit do šesti kroků:

1. Sestavení vektoru $q = (q_1, q_2, \dots, q_n)$ pořadových čísel jednotlivých kritérií a matice $P = (p_{ik})$, $i = 1, 2, \dots, m$, $k = 1, 2, \dots, n$, kde p_{ik} je pořadové číslo varianty a_i podle kritéria c_k . V případě indiference pro kritéria či varianty používáme průměrné pořadí - jestliže po prvních n variantách následuje m indiferentních variant, pak je jejich pořadové číslo $n + (1 + m) / 2$.
2. Vytvoření matice vzdáleností od fiktivního počátku $D = (d_{ik})$. Pro prvky d_{ik} platí:

$$d_{ik} = \left(\frac{1}{2}(p_{ik})^r + \frac{1}{2}(q_i)^r\right)^{1/r},$$

kde r je reálné číslo, jehož obvyklá hodnota je $r = 3$. Vzdálenost od fiktivního počátku je měřena pomocí Dujmovičovy metriky.

3. Vytvoření matice R_{ik} postupným ohodnocením seřazených hodnot d_{ik} pořadovými čísly r_{ik} , $i = 1, 2, \dots, m$, $k = 1, 2, \dots, n$. Z matice $R_{ik} = (r_{ik})$ poté spočítáme hodnoty

$$r_i = \sum_{k=1}^n r_{ik}$$

a ty seřadíme od nejmenší po největší, čímž získáme uspořádání variant.

4. Výpočet preferenčních intenzit $c_{ik} = \sum_{h \in I_{ik}} (r_{kh} - r_{ih})$, kde I_{ik} je množina kritérií, z jejichž hlediska je varianta a_i lepší než varianta a_k , neboli $a_i P a_k$. Maximální hodnota intenzity je $c^{\max} = k^2(p-1)$ a normalizovaná preferenční intenzita pak

$$c_{ik}^n = \frac{c_{ik}}{c^{\max}}$$

Tyto hodnoty se používají k identifikaci vztahů preference P, indiference I a nesrovnatelnosti N mezi variantami. Symboly α, β, γ označme jako prahy (parametry pro testy indiference a nesrovnatelnosti). Předpokládejme, že platí: $c_{ik}^n \geq c_{ki}^n$.

5. Test indiference. Jestliže platí $c_{ik}^n \leq \alpha$ a $c_{ik}^n - c_{ki}^n \leq \beta$ potom jsou varianty a_i a a_k navzájem indiferentní, neboli $a_i I a_k$. Pokud vztah neplatí, nastává vztah preference nebo nesrovnatelnosti.
6. Test nesrovnatelnosti. Pokud platí $c_{ki}^n / (c_{ik}^n - c_{ki}^n) \geq \gamma$, potom jsou varianty a_i a a_k navzájem nesrovnatelné, neboli $a_i N a_k$. V opačném případě varianta a_i preferuje variantu a_k .

Pro prahové hodnoty α, β, γ existují omezení, která by měla být respektována:

$$\alpha \leq \frac{1}{2(p-1)}, \quad \beta \leq \frac{1}{k(p-1)}, \quad \gamma \geq \frac{k-2}{4}$$

1.5.3 Metody s kardinální informací

Kardinální informaci o relativní důležitosti kritérií je možné vyjádřit pomocí vektoru vah kritérií

$$v = (v_1, v_2, \dots, v_n), \quad \sum_{k=1}^n v_k = 1, \quad v_k \geq 0.$$

Pro práci s kardinálními informacemi existují 3 výpočetní principy: princip maximalizace užitku, princip minimalizace vzdálenosti od ideální varianty a princip vyhodnocování variant na základě preferenční relace.

1.5.3.1 Metody využívající princip maximalizace užitku

Tato skupina metod je založena na možnosti vyčíslení užitku, který přináší výběr určité varianty. Čím vyšší užitek varianta přináší, tím je vhodnější. Z pohledu všech kritérií se varianta ohodnotí celkovou hodnotou užitku, která se získá agregací dílčích hodnot užitku s použitím vah kritérií.

Příklady této skupiny jsou metody WSA, SMART nebo AHP.

WSA

Weighted Sum Approach předpokládá lineární funkci užitku. Vytvoříme normalizovanou matici $R = (r_{ik})$ podle (1.1). Užitek varianty a_i je potom roven

$$u(a_i) = \sum_{k=1}^n v_k r_{ik}, \quad i = 1, 2, \dots, m, \quad k = 1, 2, \dots, n.$$

Varianta s největším užtkem je vybrána jako nejlepší.

1.5.3.2 Metody založené na minimalizaci vzdálenosti od ideální varianty

Ideální varianta byla definována výše jako varianta, která ve všech kritériích dosahuje nejlepších hodnot. Taková se většinou v úlohách nevyskytuje, a proto se jako "nejlepší" označuje ta, která má podle zvolené metriky nejmenší vzdálenost od ideální varianty.

Mezi zástupce této třídy patří např. metoda CP (Compromise Programming) nebo metoda TOPSIS, která bude využita v praktické části.

TOPSIS

Metoda TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) je založena na myšlence, že zvolená varianta by měla mít nejkratší vzdálenost od ideálního řešení a nejdelší vzdálenost od bazálního řešení na základě Eukleidovské metriky. Tato metoda poskytuje úplné uspořádání množiny všech variant, čili určí i tu nejlepší variantu.

Hodnoty všech variant přes všechna kritéria jsou uspořádány do kritériální matice Y , která byla definována výše. Tuto je potřeba normalizovat na matici R , pomocí vztahu:

$$r_{ik} = \frac{y_{ik}}{(\sum_{i=1}^m (y_{ik})^2)^{1/2}}, \quad i = 1, 2, \dots, m, \quad k = 1, 2, \dots, n.$$

Následuje sestavení vážené kritériální matice W tak, že je každý sloupec matice R násobený odpovídající vahou v_k . Z této matice vybereme nejvyšší a nejnižší hodnoty v každém sloupci a z nich sestavíme vektory ideální varianty podle vztahu $H_k = \max_i w_{ik}$ a bazální varianty $D_k = \min_i w_{ik}$.

Vzdálenost od ideální varianty vypočteme

$$d_i^+ = \sqrt{\sum_{k=1}^n (w_{ik} - H_k)^2}, \quad i = 1, 2, \dots, m.$$

Vzdálenost od bazální varianty vypočteme

$$d_i^- = \sqrt{\sum_{k=1}^n (w_{ik} - D_k)^2}, \quad i = 1, 2, \dots, m.$$

Pro výběr optimální varianty máme dvě možnosti:

1. podle relativního ukazatele vzdálenosti od bazální varianty

$$z_i = \frac{d_i^-}{d_i^+ + d_i^-}.$$

Tento ukazatel nabývá hodnot z intervalu $\langle 0, 1 \rangle$, kde 0 odpovídá bazální variantě a 1 ideální variantě. Relativní ukazatel vzdálenosti od ideální varianty je pro optimální variantu co nejvyšší.

2. podle relativního ukazatele vzdálenosti od optimální varianty

$$z_i^* = \frac{d_i^+}{d_i^+ + d_i^-}.$$

Tento ukazatel nabývá hodnot z intervalu $\langle 0, 1 \rangle$, kde 1 odpovídá bazální variantě a 0 ideální variantě. Relativní ukazatel vzdálenosti od ideální varianty je pro optimální variantu co nejnižší.

1.5.4 Přístupy hodnocení na základě preferenční relace

Významnou skupinou metod vícekritériálního hodnocení variant jsou metody založené na vyhodnocování podle preferenční relace. Tyto metody vycházejí z relací mezi dvojicemi variant vzhledem k jednotlivým kritériím. Agregáčnīm procedurami je pak možné získat párové relace mezi dvojicemi variant z hlediska všech kritérií. Tyto procedury se zakládají na porovnání stupňů preference, indiference, nesrovnatelnosti s prahovými hodnotami. Změnami prahových hodnot docílíme informace o citlivosti problému a jeho řešení. Vzhledem k značnému vlivu změn na řešení vyžaduje aplikace procedury jistou dávku zkušeností.

U těchto metod není vyžadována žádná forma normalizace, což je jejich velkou výhodou, vzhledem k tomu, že normalizace může ovlivnit výslednou volbu varianty.

Mezi nejznámější metody této skupiny patří metoda AGREPREF, třída metod ELECTRE, třída metod PROMETHEE (jimž se práce bude blíže věnovat) a metoda MAPPAC.

Metoda PROMETHEE

Preference Ranking Organization METHods for Enrichment Evaluations neboli metody PROMETHEE využívají pro vyjádření síly preference tzv. preferenční funkce definované

pro všechny dvojice variant pro každé kritérium $P: A \times A \rightarrow \langle 0,1 \rangle$. Mezní hodnoty intervalu funkčních hodnot:

$P(a_i, a_j) = 0$, indiference nebo preference a_j před a_i ,

$P(a_i, a_j) \sim 0$, slabá preference a_i před a_j ,

$P(a_i, a_j) \sim 1$, silná preference a_i před a_j ,

$P(a_i, a_j) = 1$, absolutní preference a_i před a_j .

Rozdíl d hodnocení variant podle kritéria c ovlivňuje hodnoty preferenční funkce

$$P(a_i, a_j) = P(c(a_i) - c(a_j)) = P(d), \\ P(d) = 0, \quad d \geq 0$$

Preferenci v obou směrech lze vyjádřit pomocí funkce

$$H(d) = \begin{cases} P(a_i, a_j), & d \geq 0, \\ P(a_j, a_i), & d \leq 0. \end{cases}$$

V rámci těchto metod je možné vyjádřit tzv. zobecněná kritéria pomocí šesti konkrétních typů jednodušších funkcí pro vyjadřování síly preference:

1. obyčejné kritérium

$$H(d) = \begin{cases} 0, & d = 0, \\ 1, & d \neq 0. \end{cases}$$

2. kvazikritérium

$$H(d) = \begin{cases} 0, & |d| \leq q, \\ 1, & |d| \geq q, \end{cases}$$

kde q je parametr, který udává šíři indifferenční oblasti.

3. kritérium s lineární preferencí

$$H(d) = \begin{cases} d/p, & |d| \leq q, \\ 1, & |d| \geq q, \end{cases}$$

kde p je parametr, který udává šíři oblasti lineární preference.

4. úrovněvé kritérium

$$H(d) = \begin{cases} 0, & |d| \leq q, \\ 1/2, & q < |d| \leq p, \\ 1, & |d| > p, \end{cases}$$

kde q je parametr, který udává šíři indifferenční oblasti, a p je parametr, který udává šíři oblasti pro střední úroveň.

5. kritérium s indifferenční oblastí a lineární preferencí

$$H(d) = \begin{cases} 0, & |d| \leq q, \\ \frac{|d| - q}{p - q}, & q < |d| \leq p, \\ 1, & |d| > p, \end{cases}$$

kde q je parametr, který udává šíři indifferenční oblasti, a p je parametr, který udává šíři oblasti lineární preference.

6. Gaussovo kritérium

$$H(d) = 1 - e^{-d^2/\sigma^2},$$

kde σ je směrodatná odchylka.

Z těchto zobecněných kritérií si rozhodovatel vybere to, které je vhodné podle reálné interpretace vlastního kritéria. Na základě vybraného zobecněného kritéria se vypočtou hodnoty preferenční funkce pro každé kritérium ($c = 1, 2, \dots, n$) a všechny uspořádané dvojice variant ($i, j = 1, 2, \dots, m$) jako $P_h = (a_i, a_j)$.

Vícekritériální preferenční index

$$\pi(a_i, a_j) = \sum_{c=1}^n v_c P_c(a_i, a_j)$$

vyjadřuje sílu varianty a_i před variantou a_j z hlediska všech kritérií. Jeho hodnoty leží v intervalu $\langle 0, 1 \rangle$:

$$\pi(a_i, a_j) \sim 0, \text{ slabá preference } a_i \text{ před } a_j,$$

$$\pi(a_i, a_j) \sim 1, \text{ silná preference } a_i \text{ před } a_j.$$

Vyhodnocovací relaci můžeme vytvořit na základě hodnot vícekritériálního preferenčního indexu. Pro každou variantu je definován vystupující tok

$$F^+(a_i) = v_c \sum_{a_j \in A} \pi(a_i, a_j),$$

který kvantifikuje na kolik je varianta a_i globálně preferována před ostatními. Vstupující tok

$$F^-(a_i) = v_c \sum_{a_j \in A} \pi(a_j, a_i)$$

naopak kvantifikuje na kolik jsou preferovány ostatní varianty před variantou a_i . Ideální varianta by měla vystupující tok roven 1 a vstupující tok roven 0. Rozdílem vystupujícího a vstupujícího toku je čistý tok

$$F(a_i) = F^+(a_i) - F^-(a_i).$$

2 Nová metoda vícekritériálního hodnocení variant

Autory nové metody jsou Mališa Žižovič, Nada Damljanovič, Vera Lazarevič a Nedeljko Deretič, kteří ji poprvé publikovali ve vědeckém bulletinu University Polytechniky Bukurešť (Universitatea Politehnica București) v roce 2011. [5]

2.1 Algoritmus

Metoda, kterou bych vám v této práci ráda představila, je jednou z metod založených na minimalizaci vzdáleností od ideální varianty. Velice se podobá např. TOPSIS, ale každá hledá vzdálenosti od ideální a bazální varianty jiným způsobem.

Uvažujeme množinu alternativ $A = (a_1, a_2, \dots, a_m)$, které jsou ohodnoceny podle množiny kritérií $C = (c_1, c_2, \dots, c_n)$. Předpokládáme, že rozhodovatel jasně stanoví, která kritéria jsou relevantní při ohodnocování alternativ. Dále předpokládáme, že kritéria při této metodě nemusí nutně být stejně důležitá a že rozhodovatel si je vědom důležitosti každého z nich a je schopen je v tomto směru ohodnotit. Každému kritériu $c_k, k \in (1, 2, \dots, n)$ je tak přiřazeno číslo $z_k \in (0, 1)$, které reprezentuje stupeň důležitosti kritéria a stanoví jej rozhodovatel. Kritéria mohou být jak maximalizační tak minimalizační, ale při výpočtu je třeba je všechny převést na maximalizační.

Také pro každé kritérium uvažujeme funkci vyjadřující důležitost rozdílu mezi dvěma alternativami. Rozdíl v ceně 500,- mezi televizemi, které stojí 10 000,- a 10 500,- není tak markantní jako mezi televizemi, které stojí 3 000,- a 3 500,-. Tyto funkce jsou stanoveny na základě domluvy mezi rozhodovatelem a řešitelem a pro vnější svět jsou v podstatě skryty. Tyto funkce označíme $\zeta_k : A \times A \rightarrow [0, \infty)$ pro každé kritérium $c_k, k \in (1, 2, \dots, n)$, kde ζ_k je nezáporné reálné číslo z intervalu (a_i, a_j) pro každý pár alternativ $a_i, a_j \in A$.

Pro usnadnění reprezentujeme alternativy a_i jako n -tice $a_i = (a_{i1}, a_{i2}, \dots, a_{in})$, pro každé $i \in (1, 2, \dots, m)$. Každý z prvků $a_{ij}, j \in (1, 2, \dots, n)$ je nezáporné reálné číslo vyjadřující stupeň spokojenosti s alternativou a_i v kritériu c_k .

V následujícím algoritmu budu pro jednoduchost a jasnost označovat alternativy z různých množin a a b .

Ted' vytvoříme dvě hypotetické množiny alternativ následujícím způsobem:

$$\bar{A} = \left\{ \left(\max_{i \in J} a_{ik} \right)_{k=1}^n \mid J \in \pi(\{1, 2, \dots, m\}) \setminus \emptyset \right\},$$

$$\underline{A} = \left\{ \left(\min_{i \in J} a_{ik} \right)_{k=1}^n \mid J \in \pi(\{1, 2, \dots, m\}) \setminus \emptyset \right\}.$$

Množina $\bar{A}$ je uzavřena vůči maximální operaci, tj. pro každé $a, b \in \bar{A}$ existuje

$$\max \{a, b\} = \left(\max \{a_k, b_k\} \right)_{k=1}^n \in \bar{A}.$$

Přirozené uspořádání v množině $\bar{A}$ je definováno jako:

$$a \leq_{\bar{A}} b \Leftrightarrow \max \{a, b\} = b, \text{ pro každé } a, b \in \bar{A}.$$

$(\bar{A}, \leq_{\bar{A}})$ je horní polosvaz s nejvyšším prvkem $\max A$, který označíme jako **1** a nazýváme ho ideální variantou. $\bar{A}$ je množina všech alternativ, mezi kterými se rozhodujeme, a všech lepších (hypotetických) alternativ, které můžeme vytvořit kombinacemi hodnot kritérií c_k pro všechny varianty až po ideální variantu **1**. Tyto jsou seřazeny do uzavřeného grafu, kde na spodní úrovni jsou alternativy, mezi kterými se rozhodujeme. Čím vyšší úroveň, tím lepší hypotetické alternativy až po nejvyšší úroveň, která je vyhrazena pro ideální variantu.

Množina $\underline{A}$ je uzavřena vůči minimální operaci, tj. pro každé $a, b \in \underline{A}$ existuje

$$\min \{a, b\} = \left(\min \{a_k, b_k\} \right)_{k=1}^n \in \underline{A}.$$

Přirozené uspořádání v množině $\underline{A}$ je definován jako:

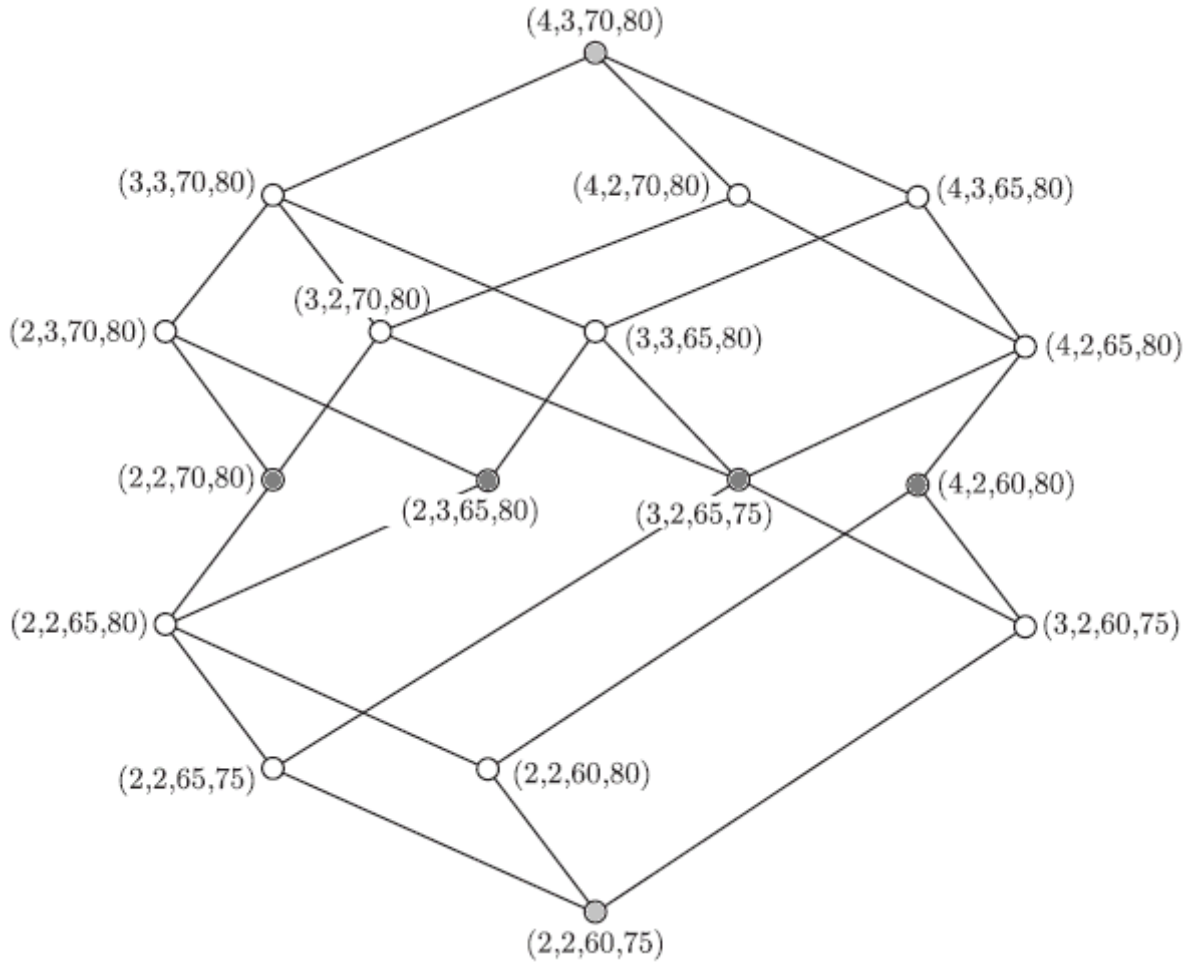
$$a \leq_{\underline{A}} b \Leftrightarrow \min \{a, b\} = a, \text{ pro každé } a, b \in \underline{A}.$$

$(\underline{A}, \leq_{\underline{A}})$ je dolní polosvaz s nejnižším prvkem $\min A$, který označíme jako **0** a nazýváme bazální variantou. $\underline{A}$ je množina všech alternativ, mezi kterými se rozhodujeme, a všech horších (hypotetických) alternativ, které můžeme vytvořit kombinacemi hodnot kritérií c_k pro všechny varianty až po bazální variantu **0**. Tyto jsou seřazeny do uzavřeného grafu, kde na spodní úrovni je bazální alternativa. Čím vyšší úroveň, tím lepší hypotetické alternativy až po nejvyšší úroveň, kde se nachází alternativy, mezi kterými se rozhodujeme.

Dalším krokem je vytvoření množiny L podle vztahu $L = \bar{A} \cup \underline{A}$. Na množině L částečně uspořádáme prvky a přitom zachováme uspořádání z množin $\bar{A}$ a $\underline{A}$ tímto způsobem: pro $a, b \in L$: když $a \leq_{\bar{A}} b$ pak $a \leq b$ a když $a \leq_{\underline{A}} b$ pak $a \leq b$. $(L, \leq)$ je částečně uspořádaná množina definovaná jako součet dolního a horního polosvazu.

$\bar{A}$ a $\underline{A}$ jsou vlastně 2 poloviny jednoho grafu, který zobrazuje množinu L , množinu všech možných variant, které se dají vytvořit kombinacemi hodnot všech kritérií přes všechny alternativy. Množina L je tvořena, abychom mohli lépe ohodnotit důležitost rozdílu mezi dvěma alternativami. Pro výpočet je klíčové mít v grafu varianty seřazeny tak, aby se alternativy o úroveň výš lišily pouze v jednom kritériu.

Příklad takového grafu můžete vidět na Obr. 2-1. Uprostřed jsou 4 alternativy (tmavě šedé body), mezi kterými si chce rozhodovatel vybrat tu nejlepší. Od nich nahoru jsou hypotetické varianty s lepšími hodnotami kritérií a v horním vrcholu grafu je ideální varianta. Směrem dolů jsou horší hypotetické varianty a dolní vrchol tvoří bazální varianta.



Obr. 2-1: Ukázka uzavřeného grafu hypotetických variant

Připomeňme, že prvek b z částečně uspořádané množiny L zahrnuje $a \in L$, což označíme $a < b$, pokud $a < b$ a $c \in L$ a platí $a \leq c \leq b$, pak $c = a$ nebo $c = b$.

Nechť $a, b \in L$ splňují $a < b$. Pak pro každé kritérium $c_k, k \in (1, 2, \dots, n)$ definujeme preferenci alternativy b před alternativou a s ohledem na kritérium c_k podle vztahu

$$\delta_k(a, b) = \zeta_k(a, b) \cdot \frac{b_k - a_k}{1_k - 0_k}. \quad (2.1)$$

ζ_k je funkce, která vyjadřuje důležitost rozdílu mezi dvěma alternativami. δ_k porovná odpovídající kritéria alternativ a funkce ζ_k může být nastavena např. jako

$$\zeta_k(a, b) = \begin{cases} 1, & \text{když } a_k \neq b_k, \\ 0, & \text{jinak.} \end{cases}$$

A čím více budeme chtít zvýraznit rozdíl a_k a b_k tím vyšší hodnotu dáme funkci $\zeta_k(a, b)$. V tomto je tato metoda jiná oproti ostatním ve své kategorii. Inovací je, že dokáže i zhodnotit důležitost velikosti rozdílu mezi hodnotami kritérií a_k a b_k .

Pomocí (2.1) také normalizujeme. Na rozdíl od např. TOPSIS, kde se normování dělá v samostatném kroku, při této metodě je spojeno s dalším výpočtem.

Budeme používat následující značení:

$$\delta_k(a,b) \\ a \mapsto b.$$

Nechť $a, b \in L$ a necht' P je cesta v množině L z a do b a existují $p_1, p_2, \dots, p_j \in L$ taková, že

$$P: a = p_1 \xrightarrow{\delta_k(p_1, p_2)} p_2 \xrightarrow{\delta_k(p_2, p_3)} \dots \xrightarrow{\delta_k(p_{j-1}, p_j)} p_j = b.$$

p_j je cesta mezi dvěma body v grafu, mezi kterými cesta existuje.

Pak preference alternativy b před alternativou a s ohledem na kritérium c_k přes cestu P je definována jako

$$\delta_k^P(a,b) = \begin{cases} 0, & \text{když } \delta_k(p_i, p_{i+1}) = 0, \text{ pro } i \in \{1, 2, \dots, j\}, \\ \frac{1}{\sum_{i=1}^{j-1} \zeta_k(p_i, p_{i+1})} \cdot \sum_{i=1}^{j-1} \delta_k(p_i, p_{i+1}), & \text{jinak,} \end{cases}$$

a podle (2.1) dostaneme

$$\delta_k^P(a,b) = \begin{cases} 0, & \text{když } \delta_k(p_i, p_{i+1}) = 0, \text{ pro } i \in \{1, 2, \dots, j\}, \\ \frac{1}{\sum_{i=1}^{j-1} \zeta_k(p_i, p_{i+1})} \cdot \sum_{i=1}^{j-1} \zeta_k(p_i, p_{i+1}) \cdot \frac{p_{i+1,k} - p_{ik}}{\mathbf{1}_k - \mathbf{0}_k}, & \text{jinak.} \end{cases}$$

Začínáme-li s alternativou a , alternativy b lze v množině L dosáhnout několika různými cestami s odpovídajícími vahami preferencí tak, že definujeme preferenci alternativy b před alternativou a s ohledem na kritérium c_k následovně:

$$\pi_k(a,b) = \max_{P: a \rightarrow b} \delta_k^P(a,b),$$

a proto, preference alternativy b před alternativou a s ohledem na kritérium c_k může být popsána jako n-tice

$$\pi(a,b) = (\pi_k(a,b))_{k=1}^n.$$

Vzdálenost alternativy $a \in L$ od ideální varianty $\mathbf{1}$ je pak definována jako

$$\mathcal{D}_1(a) = \frac{1}{\sum_{k=1}^n z_k} \cdot \sum_{k=1}^n z_k \cdot \pi_k(a, \mathbf{1})$$

a vzdálenost alternativy $a \in L$ od bazální varianty $\mathbf{0}$ je definována jako

$$\mathcal{D}_0(a) = \frac{1}{\sum_{k=1}^n z_k} \cdot \sum_{k=1}^n z_k \cdot \pi_k(\mathbf{0}, a).$$

Ve výpočtu celkové vzdálenosti od ideální nebo bazální varianty už jsou zapojeny i váhy kritérií.

Je třeba zmínit, že celý tento postup lze použít jak na celé cestě P z a do b , tak i na její části p_j . Pokud chceme použít výpočet po částečných cestách p_j , stačí sečíst $\mathcal{D}_{1j}$ a $\mathcal{D}_{0j}$ za částečné cesty abychom dostali výsledná $\mathcal{D}_1$ a $\mathcal{D}_0$.

Výsledné pořadí alternativ dostaneme porovnáním $\mathcal{D}_1$ a $\mathcal{D}_0$ za každou variantu. Řekneme, že z alternativ $a, b \in A$ je alternativa a preferována před alternativou b , když a je blíže ideální alternativě a dále od bazální alternativy:

a je preferována před b když $\mathcal{D}_1(a) < \mathcal{D}_1(b)$ a $\mathcal{D}_0(a) > \mathcal{D}_0(b)$,
nebo $\mathcal{D}_1(a) = \mathcal{D}_1(b)$ a $\mathcal{D}_0(a) > \mathcal{D}_0(b)$,
nebo $\mathcal{D}_1(a) < \mathcal{D}_1(b)$ a $\mathcal{D}_0(a) = \mathcal{D}_0(b)$,

a je indiferentní od b když

$$\mathcal{D}_1(a) = \mathcal{D}_1(b) \text{ a } \mathcal{D}_0(a) = \mathcal{D}_0(b),$$

v ostatních případech jsou a a b neporovnatelné.

Tyto vztahy mohou rozhodovateli nabídnout graf, ve kterém jsou některé alternativy porovnatelné a některé ne. Tato informace může být užitečná v praktických aplikacích.

Pokud by rozhodovatel upřednostnil přesné pořadí alternativ, může využít vztah

$$\mathcal{D}(a) = \mathcal{D}_0(a) - \mathcal{D}_1(a), \quad \text{pro } a \in A.$$

Ale tento vztah je slabší na informaci a méně realistický, dochází totiž k započtení vzdálenosti od nejlepší a nejhorší varianty.

Celkové pořadí alternativ je definováno následovně

a je preferována před b když $\mathcal{D}(a) > \mathcal{D}(b)$,
 a je indiferentní od b když $\mathcal{D}(a) = \mathcal{D}(b)$.

2.2 Modifikace nové metody

Algoritmus nové metody je založen na generování všech možných kombinací alternativ při daných hodnotách kritérií. Toto u příkladu se 6 alternativami a 6 kritérii (všech různých hodnot) vytvoří 46 656 možných alternativ. S tak velkým množstvím mezivýsledků si lehce neporadí ani výpočetní technika. Při špatně zvoleném algoritmu programu, může počítač takovou úlohu kalkulovat i několik minut, pokud mu to na to bude stačit paměť. Z tohoto důvodu je tato metoda v praxi vlastně nepoužitelná.

Vytvořila jsem její modifikaci, která značně zjednoduší výpočet a tak může být použita v praktických aplikacích. Logika tvorby je založena na aplikaci algoritmu pouze na jedné hraně. Takto vypočtené vzdálenosti za jednotlivé hrany můžeme sečíst a získat tak celkovou

vzdálenost, ať už od ideální nebo bazální varianty. Tento postup odvozují z postupu vytváření stromu variant. Co lze spočítat za celou cestu, musí jít spočítat i za jednu její část.

To mě přivedlo na myšlenku, že parciální mezivýsledky jsou vlastně poměrné části za daný úsek celé cesty. Čili výpočtem vzdálenosti pouze mezi danou variantou a ideální (popř. bazální) musím dojít ke stejnému výsledku. Toto by výrazně zkrátilo celý postup a nebylo by potřeba generovat hypotetické varianty.

Tuto úvahu jsem ověřila na několika příkladech a výsledky se od původního algoritmu neliší při

$$\zeta_k(a, b) = \begin{cases} 1, & \text{když } a_k \neq b_k, \\ 0, & \text{jinak.} \end{cases}$$

Toto dokazuje, že při takto nastavené funkci ζ_k ji v algoritmu vůbec nepotřebujeme. Toto zjištění usnadňuje situaci s tvorbou modifikace. Při úpravě algoritmu pro výpočet za jednu hranu, se některé vzorce a mezikroky stávají nepotřebnými. Právě jako funkce ζ_k . Už v prvním kroku při výpočtu δ_k podle vzorce

$$\delta_k(a, b) = \zeta_k(a, b) \cdot \frac{b_k - a_k}{\mathbf{1}_k - \mathbf{0}_k}$$

můžeme ζ_k vynechat. Pokud $a_k = b_k$, tak v čitateli stejně dostaneme 0 a ζ_k není potřeba. A pokud $a_k \neq b_k$ a $\zeta_k = 1$, tak nám to hodnotu výrazu nemění.

Navíc při výpočtu δ_k^P za celou cestu vlastně počítáme pouze s jedním δ_k a pouze v případě, kdy $\zeta_k = 1$, takže podle vzorce

$$\delta_k^P(a, b) = \begin{cases} 0, & \text{když } \delta_k(a, b) = 0, \\ \frac{1}{\zeta_k(a, b)} \cdot \zeta_k(a, b) \cdot \frac{b_k - a_k}{\mathbf{1}_k - \mathbf{0}_k}, & \text{jinak,} \end{cases}$$

se $\zeta_k(a, b)$ vyruší. Ve výsledku se celé δ_k^P neliší od δ_k . Stejně můžeme vynechat i π_k , protože hledat maximum z jednoho čísla není potřeba.

Vynecháním funkce ζ_k se připravíme o možnost započtení důležitosti rozdílu mezi dvěma alternativami. Z mé zkušenosti ovšem soudím, že zavedení ζ_k výsledky nijak nemění, pokud není definováno složitě. A to už se zdá velice překombinované.

Tím, že vypustíme ζ_k a zbavíme se nutnosti vytvářet hypotetické varianty značně zjednodušíme původní postup. Nebude potřeba tolik mezikroků k výpočtu celkové vzdálenosti $\mathcal{D}$ a to umožní vytvořit efektivní počítačový program, který si poradí s jakýmkoli příkladem.

2.2.1 Modifikovaný algoritmus

Uvažujeme množinu alternativ $A = (a_1, a_2, \dots, a_m)$, které jsou ohodnoceny podle množiny kritérií $C = (c_1, c_2, \dots, c_n)$. Každému kritériu $c_k, k \in (1, 2, \dots, n)$ je přiřazeno číslo

$z_k \in (0,1)$, které reprezentuje stupeň důležitosti kritéria a stanoví jej rozhodovatel. Pro jednoduchost předpokládáme všechna kritéria maximalizační.

Vytvoříme ideální variantu **1** a bazální variantu **0**. Vzdálenosti tvoříme od naší alternativy k ideální variantě nebo od bazální varianty k srovnávané alternativě. Pro jednoduchost budeme zapisovat alternativu, ze které vycházíme, a a alternativu, do které přicházíme, b . Cesta P je definována jako cesta z alternativy a do b .

Pro každé kritérium $c_k, k \in (1, 2, \dots, n)$ definujeme preferenci alternativy b před alternativou a s ohledem na kritérium c_k podle vztahu

$$\delta_k(a, b) = \frac{b_k - a_k}{1_k - 0_k}. \quad (2.2)$$

A dalším krokem už je výpočet celkové vzdálenosti alternativy a od ideální varianty **1**. Tato se definuje jako

$$\mathcal{D}_1(a) = \frac{1}{\sum_{k=1}^n z_k} \cdot \sum_{k=1}^n z_k \cdot \delta_k(a, 1), \quad (2.3)$$

a vzdálenost alternativy a od bazální varianty **0** je definována jako

$$\mathcal{D}_0(a) = \frac{1}{\sum_{k=1}^n z_k} \cdot \sum_{k=1}^n z_k \cdot \delta_k(0, a). \quad (2.4)$$

Výsledné pořadí alternativ dostaneme porovnáním $\mathcal{D}_1$ a $\mathcal{D}_0$ za každou variantu. Řekneme, že z alternativ $a, b \in A$ je alternativa a preferována před alternativou b , když a je blíže ideální alternativě a dále od bazální alternativy:

a je preferována před b když $\mathcal{D}_1(a) < \mathcal{D}_1(b)$ a $\mathcal{D}_0(a) > \mathcal{D}_0(b)$,
nebo $\mathcal{D}_1(a) = \mathcal{D}_1(b)$ a $\mathcal{D}_0(a) > \mathcal{D}_0(b)$,
nebo $\mathcal{D}_1(a) < \mathcal{D}_1(b)$ a $\mathcal{D}_0(a) = \mathcal{D}_0(b)$,

a je indiferentní od b když

$$\mathcal{D}_1(a) = \mathcal{D}_1(b) \text{ a } \mathcal{D}_0(a) = \mathcal{D}_0(b),$$

v ostatních případech jsou a a b neporovnatelné.

Pokud by rozhodovatel upřednostnil přesné pořadí alternativ, může využít vztah

$$\mathcal{D}(a) = \mathcal{D}_0(a) - \mathcal{D}_1(a), \quad \text{pro } a \in A. \quad (2.5)$$

Celkové pořadí alternativ je definováno následovně

a je preferována před b když $\mathcal{D}(a) > \mathcal{D}(b)$,
 a je indiferentní od b když $\mathcal{D}(a) = \mathcal{D}(b)$.

2.3 Ukázkový příklad

Na tomto příkladu dokážu svou teorii, ale také ukážu modifikovaný algoritmus.

Uvažujeme množinu alternativ $A = (a_1, a_2, \dots, a_m)$, které jsou ohodnoceny podle množiny kritérií $C = (c_1, c_2, \dots, c_n)$.

Uvedeme názorný hypotetický příklad. Zadání obsahuje Tab. 2-1.

	c_1	c_2	c_3
A_1	5	1.6	60
A_2	4	1.4	70
A_3	6	1.5	65

Tab. 2-1: Zadání hypotetického příkladu

Váhy kritérií nastavme $z_1 = z_2 = z_3 = 1$, takže všechna kritéria budou mít stejnou důležitost. Všechna kritéria jsou maximalizační.

Ideální alternativa $\mathbf{1} = (6, 1.6, 70)$.

Bazální alternativa $\mathbf{0} = (4, 1.4, 60)$.

Graf hypotetických variant naleznete na Obr. 2-2.¹

Z $\mathbf{0}$ do A_1 vede několik cest. A na každé cestě předvedu jiný výpočet.

Nejdříve použijeme algoritmus pro výpočet jednotlivých hran na 1. cestě z $\mathbf{0} = (4, 1.4, 60)$ do $A_1 = (5, 1.6, 60)$.

1. cesta vede přes $\mathbf{0} = (4, 1.4, 60) \rightarrow (4, 1.5, 60) \rightarrow (4, 1.6, 60) \rightarrow A_1 = (5, 1.6, 60)$

A nejdříve celý postup provedeme pro $\mathbf{0} = (4, 1.4, 60) \rightarrow (4, 1.5, 60)$.

$$c_1: 4, 4 \quad \delta_1(4, 4) = \zeta_k(a, b) \cdot \frac{b_k - a_k}{\mathbf{1}_k - \mathbf{0}_k} = 0 \cdot \frac{4 - 4}{6 - 4} = 0$$

$$c_2: 1.4, 1.5 \quad \delta_2(1.4, 1.5) = 1 \cdot \frac{1.5 - 1.4}{1.6 - 1.4} = \frac{1}{2}$$

$$c_3: 60, 60 \quad \delta_3(60, 60) = 0 \cdot \frac{60 - 60}{70 - 60} = 0$$

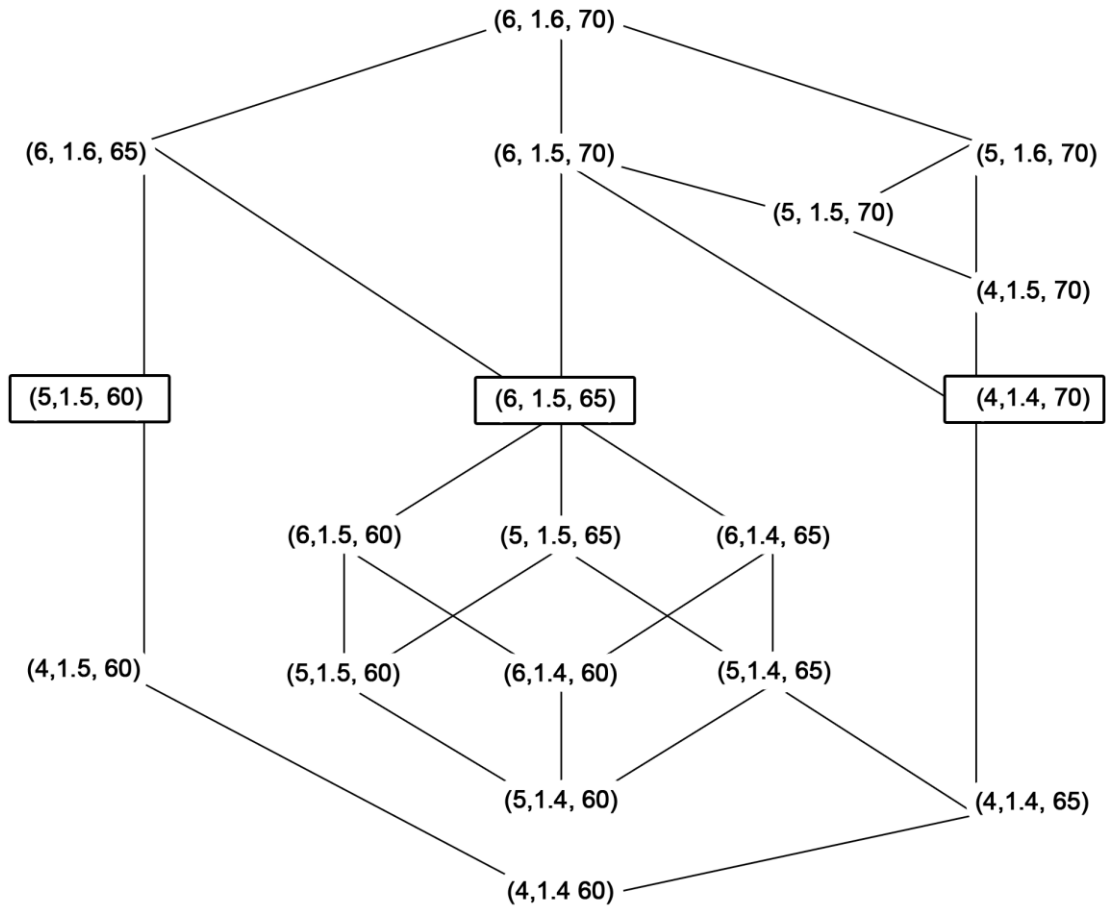
$$\delta_1^P(\mathbf{0}, a_1) = 0$$

$$\delta_2^P(\mathbf{0}, a_1) = \frac{1}{2}$$

$$\delta_3^P(\mathbf{0}, a_1) = 0$$

$$\mathcal{D}_0(a_1) = \frac{1}{1+1+1} \cdot (1 \cdot 0 + 1 \cdot \frac{1}{2} + 1 \cdot 0) = \frac{1}{6}$$

¹ Z estetických důvodů obsahuje graf pouze některé alternativy. Vynechány byly varianty, ze kterých vedly pouze 2 hrany.



Obr. 2-2: Uzavřený graf hypotetických alternativ k ukázkovému příkladu ¹

Pro $(4, 1.5, 60) \rightarrow (4, 1.6, 60)$:

$$\delta_1(4, 4) = 0 \cdot \frac{4-4}{6-4} = 0$$

$$\delta_2(1.4, 1.6) = 1 \cdot \frac{1.6-1.5}{1.6-1.4} = \frac{1}{2}$$

$$\delta_3(60, 60) = 0 \cdot \frac{60-60}{70-60} = 0$$

$$\delta_1^P(\mathbf{0}, a_2) = 0$$

$$\delta_2^P(\mathbf{0}, a_2) = \frac{1}{2}$$

$$\delta_3^P(\mathbf{0}, a_2) = 0$$

$$\mathcal{D}(a_2) = \frac{1}{1+1+1} \cdot (1 \cdot 0 + 1 \cdot \frac{1}{2} + 1 \cdot 0) = \frac{1}{6}$$

Pro $(4, 1.6, 60) \rightarrow (5, 1.6, 60)$:

$$\delta_1(4, 5) = 1 \cdot \frac{5-4}{6-4} = \frac{1}{2}$$

$$\delta_2(1.6, 1.6) = 0 \cdot \frac{1.6-1.6}{1.6-1.4} = 0$$

$$\delta_3(60, 60) = 0 \cdot \frac{60-60}{70-60} = 0$$

$$\delta_1^P(\mathbf{0}, a_3) = \frac{1}{2}$$

$$\delta_2^P(\mathbf{0}, a_3) = 0$$

$$\delta_3^P(\mathbf{0}, a_3) = 0$$

$$\mathcal{D}_0(a_3) = \frac{1}{1+1+1} \cdot (1 \cdot \frac{1}{2} + 1 \cdot 0 + 1 \cdot 0) = \frac{1}{6}$$

A za celou cestu vypadá $\mathcal{D}_0$:

$$\mathcal{D}_0(A_1) = \mathcal{D}_0(a_1) + \mathcal{D}_0(a_2) + \mathcal{D}_0(a_3) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}.$$

2. cesta vede přes $\mathbf{0} = (4, 1.4, 60) \rightarrow (5, 1.4, 60) \rightarrow (5, 1.5, 60) \rightarrow A_1 = (5, 1.6, 60)$.
Nyní vyzkoušíme způsob výpočtu z $\mathbf{0} = (4, 1.4, 60)$ přímo do $A_1 = (5, 1.6, 60)$.

$$c_1: \delta_1(4, 5) = 1 \cdot \frac{5-4}{6-4} = \frac{1}{2} \quad c_2: \delta_2(1.4, 1.6) = 1 \cdot \frac{1.6-1.4}{1.6-1.4} = 1 \quad c_3: \delta_3(60, 60) = 0 \cdot \frac{60-60}{70-60} = 0$$

$$\delta_1^p(\mathbf{0}, A_1) = \frac{1}{2}$$

$$\delta_2^p(\mathbf{0}, A_1) = 1$$

$$\delta_3^p(\mathbf{0}, A_1) = 0$$

$$\mathcal{D}_0(A_1) = \frac{1}{1+1+1} \cdot (1 \cdot \frac{1}{2} + 1 \cdot 1 + 1 \cdot 0) = \frac{1}{2}$$

3. cesta vede přes $\mathbf{0} = (4, 1.4, 60) \rightarrow (4, 1.5, 60) \rightarrow (5, 1.5, 60) \rightarrow A_1 = (5, 1.6, 60)$.
Opakování výpočtu už není nutné. Výsledné $\mathcal{D}_0$ vychází opět

$$\mathcal{D}_0 = \frac{1}{2}.$$

Stejným způsobem vypočítáme i vzdálenost k ideální variantě. Tyto výpočty je třeba provést i pro alternativy A_2 a A_3 .

Výsledná $\mathcal{D}$ jsou uvedena v Tab. 2-2.

	A_1	A_2	A_3
$\mathcal{D}_1$	1/2	2/3	1/3
$\mathcal{D}_0$	1/2	1/3	2/3
$\mathcal{D}$	0	-1/3	1/3

Tab. 2-2: Řešení ukázkového příkladu pomocí nové metody

Nejlepší variantou je A_3 vzhledem k nejvyššímu $\mathcal{D}$ podle (2.5).

² Výpočet vychází stejně přes všechny cesty, protože nebyla zadána funkce ζ_k . Kdyby tato funkce byla přidána do algoritmu, není pravidlem, že vzdálenost přes všechny cesty musí vyjít stejně.

3 Srovnání s jinými diskrétními metodami

V této části práce bude nová metoda³ srovnána s jinými hojně používanými metodami řešícími diskrétní modely rozhodování. Srovnání bude provedeno na několika příkladech, aby byla nová metoda otestována v různých podmínkách. Z metod vícekritériálního hodnocení variant používajících kardinální informaci využijeme WSA, TOPSIS a PROMETHEE. A pro obecnější srovnání využijeme metodu ORESTE, jako zástupce metod s ordinální informací.

Všechny metody jsou schopny nám poskytnout celkové pořadí alternativ. Pro lepší orientaci uvedu v této kapitole vždy jen graf pořadí alternativ. Přesné výsledky příkladů podle všech metod jsou k dispozici v příloze Příloha 1.

V algoritmech používaných metod jsou rozdíly. WSA sleduje užitek, TOPSIS vzdálenosti od ideální a bazální alternativy, PROMETHEE pracuje na bázi preferencí mezi dvojicemi alternativ a ORESTE s pořadím alternativ. Kromě metody PROMETHEE a ORESTE žádná nepotřebuje jiné vstupní informace než množinu alternativ $A = (a_1, a_2, \dots, a_m)$, které jsou ohodnoceny podle množiny kritérií $C = (c_1, c_2, \dots, c_n)$. Pro zjednodušení budeme předpokládat stejné váhy všech kritérií a maximalizační kritéria.

PROMETHEE vyžaduje dodatečné informace o kritériích, jak je popsáno výše (str. 11). Příklady zde použité jsou čistě teoretické a nemají žádnou reálnou předlohu. Nelze tedy ohodnotit typ kritéria na základě jeho reálného významu. Proto označíme všechna kritéria jako obecná. Bohužel to znamená nedostatečné využití potenciálu metody PROMETHEE, avšak dostačující pro náš účel.

Při metodě ORESTE byla reálné číslo r přiřazena jeho obvyklá hodnota $r = 3$.

V následujících podkapitolách bude zadáno několik příkladů od nejjednoduššího (rozměru 4x4) po nejsložitější (rozměru 20x50). Budou řešeny výše zmíněnými metodami a zobrazeno jejich pořadí dosažené na základě každé metody.

3.1 Příklad 1 (4x4)

	c_1	c_2	c_3	c_4
A_1	2	2	70	80
A_2	2	3	65	80
A_3	3	2	65	75
A_4	4	2	60	80

Tab. 3-1: Zadání příkladu 1

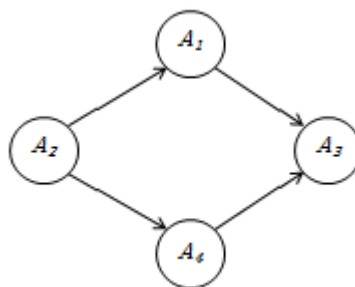
³ Srovnána bude i modifikace nové metody. Při nastavení kritérií na stejnou váhu vychází metody vždy shodně.

Novou metodou získáme řešení obsažené v tabulce Tab. 3-2.

	A_1	A_2	A_3	A_4
$\mathcal{D}_1$	0.5	0.375	0.75	0.5
$\mathcal{D}_0$	0.5	0.625	0.25	0.5
$\mathcal{D}$	0	0.25	-0.5	0

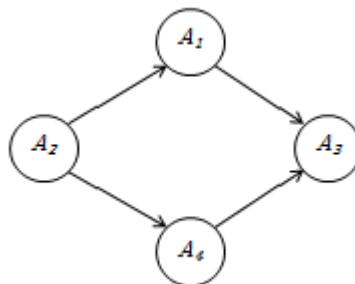
Tab. 3-2: Řešení příkladu 1 pomocí nové metody

A pořadí zobrazené na Obr. 3-1.

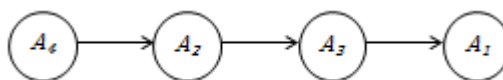


Obr. 3-1: Řešení příkladu 1 novou metodou

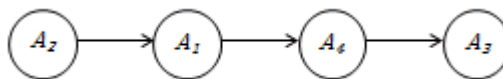
Na Obr. 3-2 - Obr. 3-5 lze vidět pořadí podle ostatních metod.



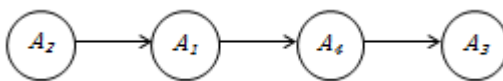
Obr. 3-2: Řešení příkladu 1 metodou WSA



Obr. 3-3: Řešení příkladu 1 metodou TOPSIS



Obr. 3-4: Řešení příkladu 1 metodou ORESTE



Obr. 3-5: Řešení příkladu 1 metodou PROMETHEE

Podle většiny metod je nejlepší variantou A_2 a nejhorší variantou A_3 . A_1 je většinou druhou nejlepší před A_4 .

Pořadí podle WSA a nové metody vychází shodně. Je zajímavé, že ani jedna nedokáže rozlišit pořadí A_1 a A_4 .

Výsledky podle TOPSIS se výrazně liší od ostatních. Rozhodovatel by vybral A_4 před A_2 , A_3 a určitě by nezvolil A_1 hodnocenou nejhůře.

3.2 Příklad 2 (6x5)

	c_1	c_2	c_3	c_4	c_5
A_1	100	4	32	50	4,5
A_2	95	3	35	44	4,4
A_3	90	6	24	40	4,0
A_4	70	10	30	45	4,0
A_5	150	2	33	50	5,0
A_6	120	5	32	40	6,0

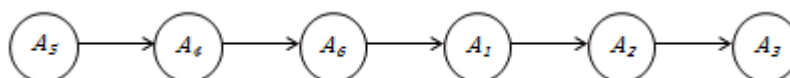
Tab. 3-3: Zadání příkladu 2

Novou metodou získáme řešení obsažené v Tab. 3-4.

	A_1	A_2	A_3	A_4	A_5	A_6
$\mathcal{D}_1$	0.47955	0.59250	0.85000	0.59091	0.33636	0.45455
$\mathcal{D}_0$	0.52045	0.40750	0.15000	0.40909	0.66363	0.54545
$\mathcal{D}$	0.04091	-0.18500	-0.70000	0.18182	0.32727	0.09091

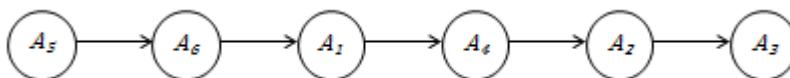
Tab. 3-4: Řešení příkladu 2 pomocí nové metody

A pořadí zobrazené na Obr. 3-6.

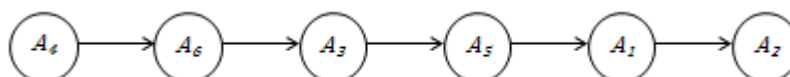


Obr. 3-6: Řešení příkladu 2 novou metodou

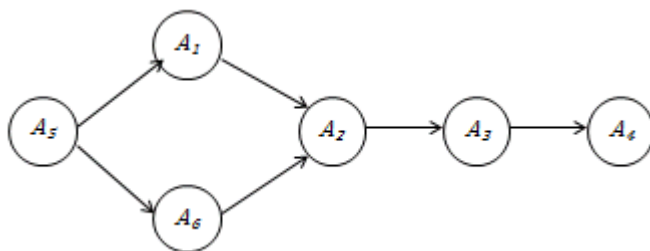
Na Obr. 3-7 - Obr. 3-10 lze vidět pořadí podle ostatních metod.



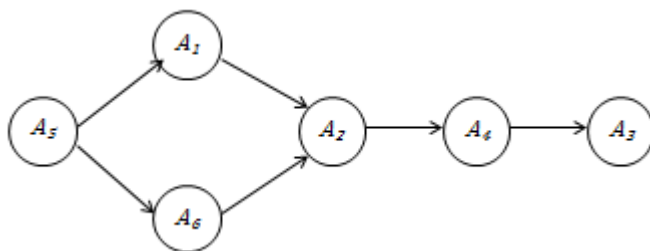
Obr. 3-7: Řešení příkladu 2 metodou WSA



Obr. 3-8: Řešení příkladu 2 metodou TOPSIS



Obr. 3-9: Řešení příkladu 2 metodou ORESTE



Obr. 3-10: Řešení příkladu 2 metodou PROMETHEE

Podle většiny metod je nejlepší variantou A_5 a nejhorší variantou A_3 . Zbytek alternativ už nevyhází nijak jednoznačně.

Pořadí podle ORESTE a PROMETHEE vychází téměř shodně. Vyměněny jsou pouze alternativy na posledních místech. Je zajímavé, že ani jedna nedokáže rozlišit pořadí A_1 a A_6 .

Výsledky podle TOPSIS se výrazně liší od ostatních. Rozhodovatel by vybral A_4 a určitě by nezvolil A_2 hodnocenou nejhuře. Nejlepší varianta podle ostatních A_5 je až na 4. místě.

Velice podobně opět vyšly pořadí podle WSA a nové metody. Liší se pouze v ohodnocení A_4 .

3.3 Příklad 3 (5x10)

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
A_1	100	4	32	50	4.5	3	32	250	1.8	5
A_2	95	3	35	44	4.4	2	36	200	1.5	5.5
A_3	90	6	24	40	4.0	4	35	280	1.6	5.8
A_4	70	10	30	45	4.0	3	33	265	1.9	5.6
A_5	150	2	33	50	5.4	2	32	230	1.5	6

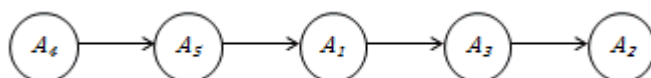
Tab. 3-5: Zadání příkladu 3

Novou metodou získáme řešení obsažené v Tab. 3-6.

	A_1	A_2	A_3	A_4	A_5
$\mathcal{D}_1$	0.5416	0.6377	0.5450	0.4792	0.4807
$\mathcal{D}_0$	0.4584	0.3623	0.4550	0.5208	0.5193
$\mathcal{D}$	-0.0831	-0.2754	-0.0900	0.0416	0.0386

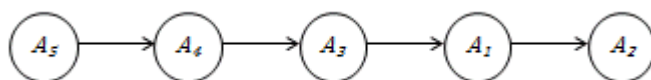
Tab. 3-6: Řešení příkladu 3 pomocí nové metody

A pořadí zobrazené na Obr. 3-11.

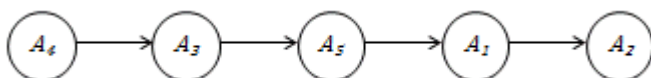


Obr. 3-11: Řešení příkladu 3 novou metodou

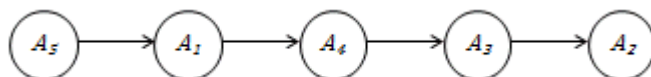
Na Obr. 3-12 - Obr. 3-15 lze vidět pořadí podle ostatních metod.



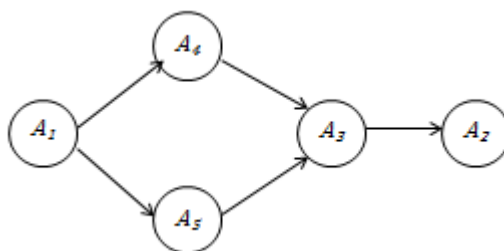
Obr. 3-12: Řešení příkladu 3 metodou WSA



Obr. 3-13: Řešení příkladu 3 metodou TOPSIS



Obr. 3-14: Řešení příkladu 3 metodou ORESTE



Obr. 3-15: Řešení příkladu 3 metodou PROMETHEE

Řešení tohoto příkladu už není tak jednoznačné jako u dvou předchozích. Metody se shodnou pouze na nejhorší variantě A_2 .

PROMETHEE klade na 1. místo variantu A_1 podle ostatních spíše horší.

Výsledky podle TOPSIS se výrazně liší od ostatních. Rozhodovatel by vybral A_4 a určitě by nezvolil A_2 hodnocenou nejhůře. Nejlepší varianta podle ostatních A_5 je až na 4. místě.

Velice podobně opět vyšly pořadí podle WSA a nové metody. Jsou vyměněny alternativy na 1. a 2. místě a na 3. a 4. místě.

3.4 Příklad 4 (9x10)

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
A_1	100	4	32	50	4.5	3	32	250	1.8	5
A_2	95	3	35	44	4.4	2	36	200	1.5	5.5
A_3	90	6	24	40	4.0	4	35	280	1.6	5.8
A_4	70	10	30	45	4.0	3	33	265	1.9	5.6
A_5	150	2	33	50	5.4	2	32	230	1.5	6
A_6	80	8	36	46	5.2	2	36	270	1.8	4.9
A_7	90	7	32	51	5.3	3	34	263	1.7	5.3
A_8	105	6	28	42	4.9	5	33	260	1.4	5.4
A_9	140	2	29	43	4.6	2	36	270	1.6	5.8

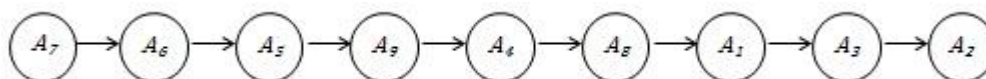
Tab. 3-7: Zadání příkladu 4

Novou metodou získáme řešení obsažené v Tab. 3-8.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
$\mathcal{D}_1$	0.559	0.625	0.562	0.501	0.477	0.405	0.395	0.545	0.491
$\mathcal{D}_6$	0.441	0.375	0.438	0.499	0.523	0.595	0.605	0.455	0.509
$\mathcal{D}$	-0.119	-0.250	-0.123	-0.003	0.047	0.191	0.211	-0.090	0.017

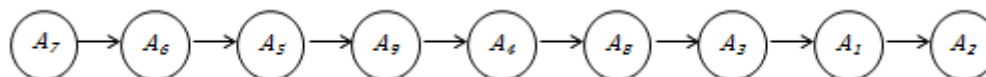
Tab. 3-8: Řešení příkladu 4 pomocí nové metody

A pořadí zobrazené na Obr. 3-16.

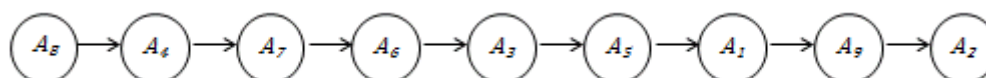


Obr. 3-16: Řešení příkladu 4 novou metodou

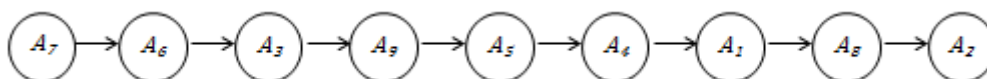
Na Obr. 3-17 - Obr. 3-20 lze vidět pořadí podle ostatních metod.



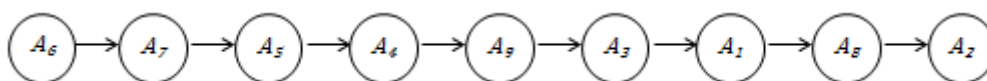
Obr. 3-17: Řešení příkladu 4 metodou WSA



Obr. 3-18: Řešení příkladu 4 metodou TOPSIS



Obr. 3-19: Řešení příkladu 4 metodou ORESTE



Obr. 3-20: Řešení příkladu 4 metodou PROMETHEE

Všechny metody se shodnou na nejhorší variantě A_2 . Nejlepšími variantami jsou podle většiny A_6 a A_7 , které se střídají na prvních pozicích všech metod kromě TOPSIS.

TOPSIS určila jako nejlepší alternativu A_8 , ale podle ostatních je to jedna z horších variant.

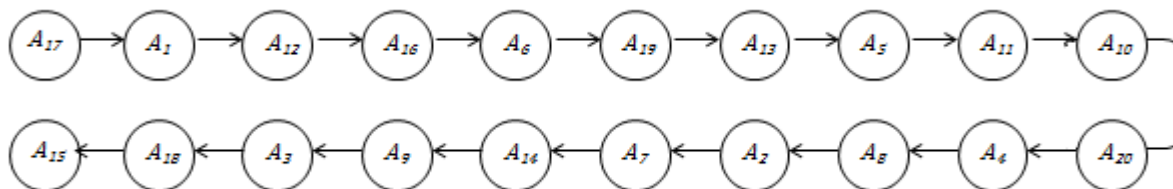
Pořadí podle WSA a nové metody se liší pouze výměnou A_1 a A_3 na 7. a 8. místě.

PROMETHEE a ORESTE se shodují na třech posledních místech a na prvních 2 mají vyměněné A_6 a A_7 .

3.5 Příklad 5 (20x50)

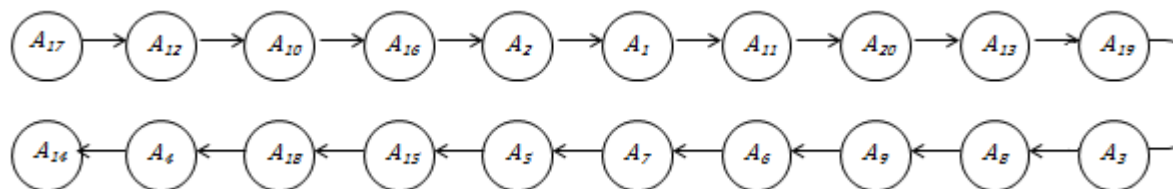
Zadání příkladu 5 je uvedeno v příloze v Tab. 17 vzhledem k jeho rozsáhlosti. Z toho samého důvodu zde nebude uvedeno ani přesné řešení pomocí nové metody, pouze výsledné pořadí variant. Konkrétní hodnoty obsahuje Tab. 18 v příloze.

A pořadí zobrazené na Obr. 3-21.

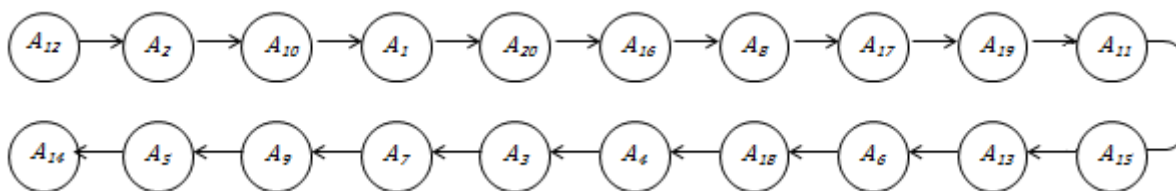


Obr. 3-21: Řešení příkladu 5 novou metodou

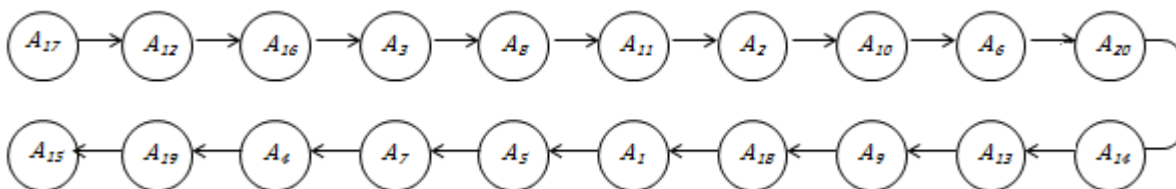
Na Obr. 3-22 - Obr. 3-25 lze vidět pořadí podle ostatních metod.



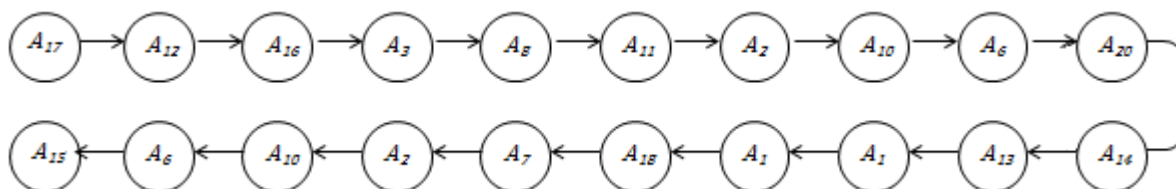
Obr. 3-22: Řešení příkladu 5 metodou WSA



Obr. 3-23: Řešení příkladu 5 metodou TOPSIS



Obr. 3-24: Řešení příkladu 5 metodou ORESTE



Obr. 3-25: Řešení příkladu 5 metodou PROMETHEE

Podle většiny metod se jeví nejlepší variantou A_{17} , naopak nejhoršími A_{14} a A_{15} .

TOPSIS se s ostatními shodne na posledním místě, ale jinak nabízí úplně odlišné pořadí. Na 1. místo staví A_{12} a A_{17} až na 8. místo.

Při takto rozsáhlém rozhodovacím problému už se WSA a nová metoda moc nepřekrývají. Shodnou se pouze na 1., 4. a 15. místě.

Naproti tomu, ORESTE a PROMETHEE se shodují od 1. po 12. místo a na posledním.

3.6 Shrnutí

Nová metoda se ukázala jako velmi podobná WSA, alespoň co se výsledků týče. V praktických aplikacích by se to mohlo jevit jako problém. Budeme-li počítat příklad o třech variantách, z nichž budou dvě s krajními hodnotami a jedna vyrovnaná, např. (10, 1), (1, 10), (5, 5), nová metoda nevybere vyrovnanou variantu. Rozhodovateli nabídne jako nejvýhodnější jednu z marginálních a nerozliší kterou.

TOPSIS se svými výsledky značně liší od všech ostatních použitých metod.

Pro mě překvapivé zjištění byly i výsledky ORESTE a PROMETHEE. Ve většině příkladů dosahovaly shodných výsledků.

4 Software

Pro účely výzkumu v této práci byl vytvořen program, který implementuje algoritmus modifikované metody. Díky tomuto programu bylo možné otestovat metodu v širokých spektrech parametrů a odhalit tak její silné a slabé stránky.

4.1 Technické parametry

Aplikace je vytvořena v jazyce C#.Net. Součástí přílohy Příloha 2 je zdrojový kód programu.

Má tyto minimální požadavky na systém:

- Windows 7
- .Net Framework 4.5 (platforma, kterou obsahují všechny aktualizované operační systémy Windows 7 a novější)

Program funguje na bázi modifikovaného algoritmu. Nejdříve vypočítá δ_k všech kritérií podle (2.2) a z nich následně vzdálenosti od zadaných alternativ k ideální variantě podle vzorce (2.3) a od bazální varianty k zadaným alternativám podle vzorce (2.4).

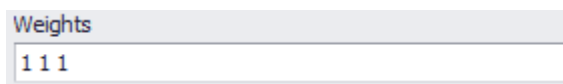
Součástí této práce je CD, které obsahuje složku *New method*. V této složce je soubor *NewMethod.exe*, což je spustitelný soubor s programem. Program není nutné instalovat, stačí jej nahrát na pevný disk a spustit.

4.2 Uživatelská příručka

Uživatelské rozhraní aplikace je velmi jednoduché a intuitivní. Základní obrazovku aplikace můžete vidět na Obr. 4-2.

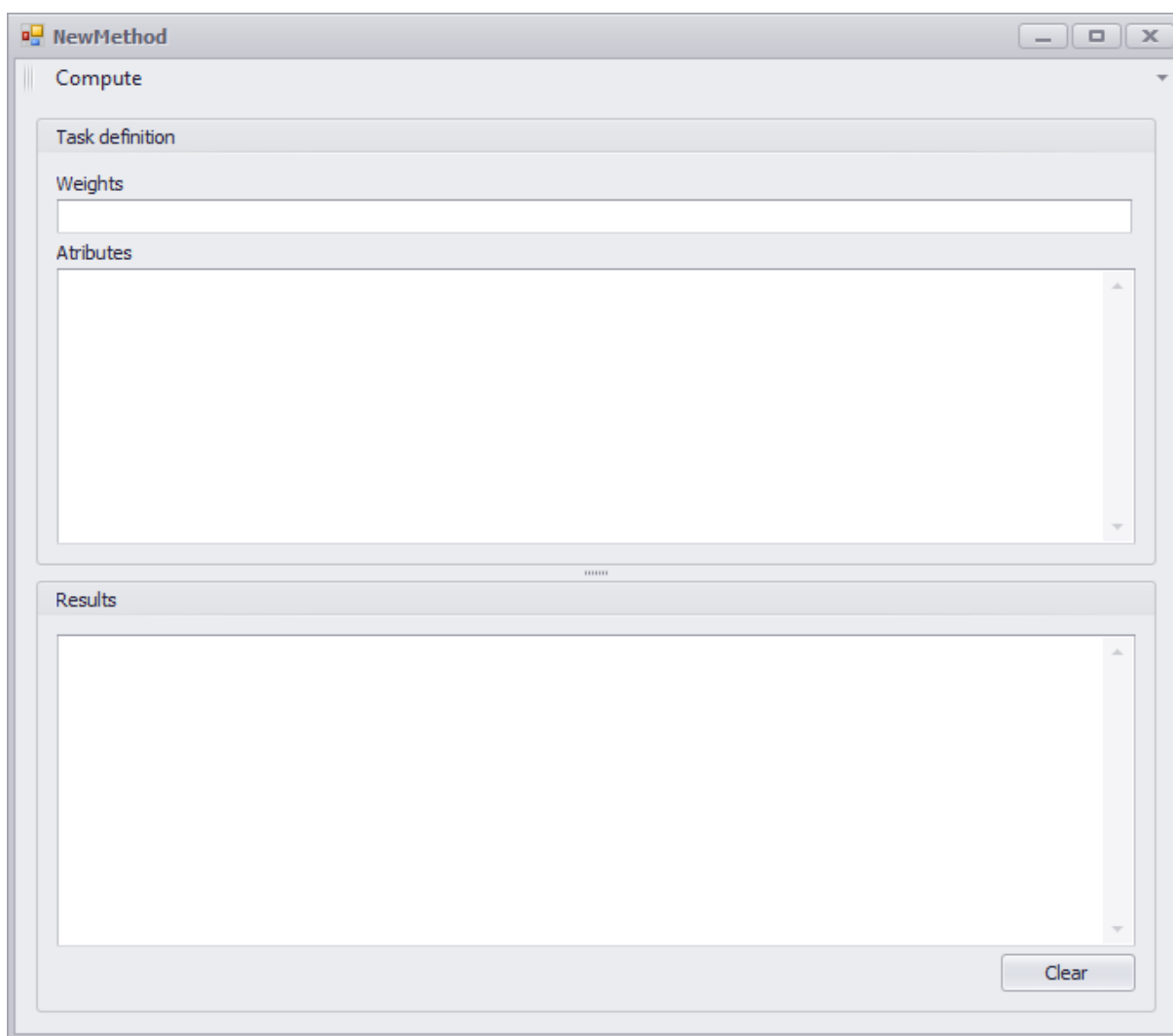
Všechny vstupní údaje se zadávají do panelu **Task definition**.

Váhy kritérií uživatel zapisuje do pole **Weights**. Kritéria jsou zapisována jako vektor bez závorek, čili hodnoty jednotlivých kritérií po řadě za sebou a mezi nimi jsou mezery. Na Obr. 4-1 je zadání vah ukázkového příkladu (Tab. 3-1 z kapitoly 3).

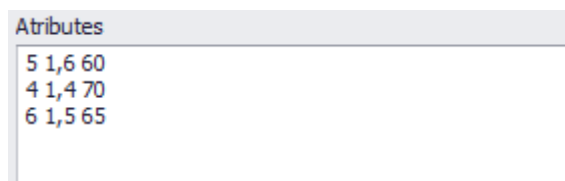


Obr. 4-1: Zadání vah v aplikaci

Do pole **Attributes** jsou zapisovány konkrétní hodnoty kritérií pro jednotlivé alternativy. Varianty jsou zadávány na samostatných řádcích. Kritéria pro jednu alternativu jsou v řádku po řadě za sebou. Mezi všechny hodnoty jsou zapisovány mezery stejně jako u kritérií. Příklad zápisu je na Obr. 4-3 (opět zadání ukázkového příkladu z Tab. 3-1 z kapitoly 3).

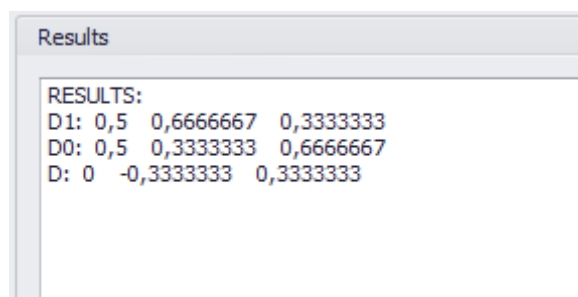


Obr. 4-2: Uživatelské rozhraní aplikace



Obr. 4-3: Zadání alternativ v aplikaci

Pro získání výsledku slouží tlačítko **Compute**. Výsledky jsou zobrazeny v panelu **Results**, jak je vidět na Obr. 4-4.



Obr. 4-4: Zobrazení výsledků v aplikaci

D1 jsou vzdálenosti variant k ideálu, D0 vzdálenosti k bazálu a D je celková vzdálenost. V řádku jsou vždy výsledky pro jednotlivé varianty za sebou podle pořadového čísla. Takže vzdálenost A_1 k **1** je 0.5, vzdálenost A_2 k **1** je 0.6667, atd.

Výsledky se v okně Results ukládají za všechny kalkulace až do stisknutí tlačítka **Clear**, které okno Results vyčistí.

Zadávání desetinných čísel funguje dle lokálního nastavení. Např. při nastavení operačního systému na desetinné čárky bude tato aplikace používat desetinné čárky.

Při ukončení programu a opětovném spuštění zůstává uloženo poslední zadání, což může být užitečné při řešení rozsáhlejších rozhodovacích problémů.

Závěr

V úvodu práce jsem se věnovala teorii rozhodování a její základní terminologii a principům. Část byla zaměřena především na základní pojmy vícekritériálního hodnocení variant tak, aby měl čtenář veškeré informace a podklady pro správné porozumění práci.

Jejím cílem bylo představit novou, dosud prakticky neznámou metodu, její algoritmus a fungování. Metoda používá kardinální informace o datech a řadí se mezi metody založené na minimalizaci vzdálenosti od ideální varianty. Metoda klade důraz nejen na váhu kritérií, ale také na důležitost rozdílu mezi dvěma variantami. Pracuje s hypotetickými variantami vytvořenými z hodnot kritérií původních variant. Kdy ze všech variant, ať už hypotetických nebo reálných je vytvořen svaz variant. Ve svazu jsou varianty srovnány od nejhorší (bazální) po nejlepší (ideální). A z něj se pak počítají vzdálenosti k ideální a bazální variantě. Během studia této metody jsem zjistila, že generování tohoto svazu je velice časově náročné, a proto není možné ji používat v praxi.

Toto zjištění mě vedlo k sestavení modifikace nové metody, která nevytváří svaz variant a není tak náročná na výpočet ani čas. Algoritmus byl sice ochuzen o funkci, která přidávala informaci o důležitosti rozdílu mezi dvěma variantami, ale ta se ve výsledku neprojeví, pokud je zadána jednoduše. Složitá funkce naopak ještě více komplikuje výpočet. Z těchto důvodů si myslím, že je v původní metodě spíše zbytečná. Modifikace nové metody bude snadno využitelná v reálných rozhodovacích situacích.

Novou metodu i její modifikaci jsem podrobila srovnání s jinými metodami vícekritériálního hodnocení variant, konkrétně WSA, TOPSIS, ORESTE a PROMETHEE. Její výsledky se od ostatních lišily, ale zanedbatelně. Když neberu v úvahu TOPSIS, jejíž výsledky byly vždy velice odlišné, metody se většinou shodly na prvních a posledních místech. Zajímavostí je značná podobnost výsledků nové metody s WSA. Mohlo by to znamenat, že podobně jako WSA nová metoda není schopna vybrat vyrovnané varianty a bude mít tendenci se přiklánět k nevyrovnaným.

Pro účely výzkumu v této práci byl vytvořen program, který implementuje algoritmus modifikované metody. Díky tomuto programu bylo možné otestovat metodu v širokých spektrech parametrů a odhalit tak její silné a slabé stránky. I když nebylo cílem práce takový program vytvořit, přikládám jej, aby si novou metodu mohl vyzkoušet kdokoli.

Při vytváření této práce jsem si prohloubila znalosti teorie rozhodování, což bylo hlavním cílem při výběru tématu. Dle mého názoru jsem splnila zadání a představila metodu, která by se mohla stát součástí různých programových vybavení.

Literatura

- [1] FIALA, P., JABLONSKÝ, J., MAŇAS, M.: *Vícekritériální rozhodování*. Praha: Vysoká škola ekonomická, 1996, 316 s. ISBN 80-7079-748-7.
- [2] FIALA, P.: *Modely a metody rozhodování*. 2. přeprac. vyd. V Praze: Oeconomica, 2008, 292 s. ISBN 978-80-245-1345-4.
- [3] FOTR, J., PÍŠEK, M.: *Exaktní metody ekonomického rozhodování*. Praha: Academia, 1986, 168 s.
- [4] MEZINÁRODNÍ ÚSTAV APLIKOVANÉ SYSTÉMOVÉ ANALÝZY, RAIFFA, H., ed., BELL, D. E., ed. a KEENEY, R. L., ed.: *Conflicting objectives in decisions*. Chichester: John Wiley & Sons, 1977. x, 442 s. International series on applied systems analysis; 1. A Wiley-Interscience publication. ISBN 0-471-99506-1.
- [5] ŽIŽOVIĆ, M., DAMLIJANOVIĆ, N., LAZAREVIĆ, V., DERETIĆ, N.: *New Method for Multicriteria Analysis*. U.P.B. Scientific Bulletin, Series A, Vol. 73, Iss. 2, 2011, str. 13-22.

Internetové zdroje

- [6] SEKNIČKOVÁ, J.: Vícekritériální hodnocení variant - VHV [online, cit. 25.3.2013]. Dostupný na WWW:
<http://jana.kalcev.cz/vyuka/kestazeni/EKO422-Vahy.pdf> .
- [7] SEKNIČKOVÁ, J.: Informace o aspiračních úrovních kritérií [online, cit. 25.3.2013]. Dostupný na WWW:
<http://jana.kalcev.cz/vyuka/kestazeni/EKO422-Aspiracni.pdf>
- [8] SEKNIČKOVÁ, J.: Ordinální informace o kritériích [online, cit. 25.3.2013]. Dostupný na WWW:
<http://jana.kalcev.cz/vyuka/kestazeni/EKO422-Ordinalni.pdf>
- [9] Preference ranking organization method for enrichment evaluation [online, cit. 26.3.2013]. Dostupný z WWW:
http://en.wikipedia.org/wiki/Preference_ranking_organization_method_for_enrichment_evaluation.

Ostatní zdroje

- [10] KOVAŘÍKOVÁ, L.: *Hodnocení investičních možností PRE do malých vodních elektráren*. Diplomová práce, Vysoká škola ekonomická v Praze, 2011. Vedoucí práce Jana Kalčevová.

Seznam obrázků

Obr. 2-1: Ukázka uzavřeného grafu hypotetických variant	15
Obr. 2-2: Uzavřený graf hypotetických alternativ k ukázkovému příkladu	21
Obr. 3-1: Řešení příkladu 1 novou metodou	24
Obr. 3-2: Řešení příkladu 1 metodou WSA	24
Obr. 3-3: Řešení příkladu 1 metodou TOPSIS	24
Obr. 3-4: Řešení příkladu 1 metodou ORESTE	24
Obr. 3-5: Řešení příkladu 1 metodou PROMETHEE	24
Obr. 3-6: Řešení příkladu 2 novou metodou	25
Obr. 3-7: Řešení příkladu 2 metodou WSA	25
Obr. 3-8: Řešení příkladu 2 metodou TOPSIS	25
Obr. 3-9: Řešení příkladu 2 metodou ORESTE	26
Obr. 3-10: Řešení příkladu 2 metodou PROMETHEE	26
Obr. 3-11: Řešení příkladu 3 novou metodou	27
Obr. 3-12: Řešení příkladu 3 metodou WSA	27
Obr. 3-13: Řešení příkladu 3 metodou TOPSIS	27
Obr. 3-14: Řešení příkladu 3 metodou ORESTE	27
Obr. 3-15: Řešení příkladu 3 metodou PROMETHEE	27
Obr. 3-16: Řešení příkladu 4 novou metodou	28
Obr. 3-17: Řešení příkladu 4 metodou WSA	28
Obr. 3-18: Řešení příkladu 4 metodou TOPSIS	28
Obr. 3-19: Řešení příkladu 4 metodou ORESTE	29
Obr. 3-20: Řešení příkladu 4 metodou PROMETHEE	29
Obr. 3-21: Řešení příkladu 5 novou metodou	29
Obr. 3-22: Řešení příkladu 5 metodou WSA	29
Obr. 3-23: Řešení příkladu 5 metodou TOPSIS	30
Obr. 3-24: Řešení příkladu 5 metodou ORESTE	30
Obr. 3-25: Řešení příkladu 5 metodou PROMETHEE	30
Obr. 4-1: Zadání vah v aplikaci	31
Obr. 4-2: Uživatelské rozhraní aplikace	32
Obr. 4-3: Zadání alternativ v aplikaci	32
Obr. 4-4: Zobrazení výsledků v aplikaci	33

Seznam tabulek

Tab. 2-1: Zadání hypotetického příkladu	20
Tab. 2-2: Řešení ukázkového příkladu pomocí nové metody	22
Tab. 3-1: Zadání příkladu 1	23
Tab. 3-2: Řešení příkladu 1 pomocí nové metody	24
Tab. 3-3: Zadání příkladu 2	25
Tab. 3-4: Řešení příkladu 2 pomocí nové metody	25
Tab. 3-5: Zadání příkladu 3	26
Tab. 3-6: Řešení příkladu 3 pomocí nové metody	27
Tab. 3-7: Zadání příkladu 4	28
Tab. 3-8: Řešení příkladu 4 pomocí nové metody	28
Tab. 1: Řešení příkladu č. 1 pomocí metody WSA	38
Tab. 2: Řešení příkladu č. 1 pomocí metody TOPSIS	38
Tab. 3: Řešení příkladu č. 1 pomocí metody ORESTE	38
Tab. 4: Řešení příkladu č. 1 pomocí metody PROMETHEE	38
Tab. 5: Řešení příkladu č. 2 pomocí metody WSA	38
Tab. 6: Řešení příkladu č. 2 pomocí metody TOPSIS	39
Tab. 7: Řešení příkladu č. 2 pomocí metody ORESTE	39
Tab. 8: Řešení příkladu č. 2 pomocí metody PROMETHEE	39
Tab. 9: Řešení příkladu č. 3 pomocí metody WSA	39
Tab. 10: Řešení příkladu č. 3 pomocí metody TOPSIS	39
Tab. 11: Řešení příkladu č. 3 pomocí metody ORESTE	39
Tab. 12: Řešení příkladu č. 3 pomocí metody PROMETHEE	40
Tab. 13: Řešení příkladu č. 4 pomocí metody WSA	40
Tab. 14: Řešení příkladu č. 4 pomocí metody TOPSIS	40
Tab. 15: Řešení příkladu č. 4 pomocí metody ORESTE	40
Tab. 16: Řešení příkladu č. 4 pomocí metody PROMETHEE	40
Tab. 17: Zadání příkladu č. 5	41
Tab. 18: Řešení příkladu č. 5 pomocí nové metody	44
Tab. 19: Řešení příkladu č. 5 pomocí metody WSA	44
Tab. 20: Řešení příkladu č. 5 pomocí metody TOPSIS	44
Tab. 21: Řešení příkladu č. 5 pomocí metody ORESTE	45
Tab. 22: Řešení příkladu č. 5 pomocí metody PROMETHEE	45

Přílohy

Příloha 1

Řešení příkladů 1 - 4 z kapitoly 3 pomocí metod WSA, TOPSIS, ORESTE, PROMETHEE

Tab. 1: Řešení příkladu č. 1 pomocí metody WSA

	A_1	A_2	A_3	A_4
u	0.5	0.625	0.25	0.5

Tab. 2: Řešení příkladu č. 1 pomocí metody TOPSIS

	A_1	A_2	A_3	A_4
d_i^+	0.10272	0.08757	0.07089	0.05784
d_i^-	0.02078	0.05596	0.04457	0.08740
c_i	0.16823	0.38988	0.38600	0.60178

Tab. 3: Řešení příkladu č. 1 pomocí metody ORESTE

	A_1	A_2	A_3	A_4
r_i	32	29.5	40.5	34

Tab. 4: Řešení příkladu č. 1 pomocí metody PROMETHEE

	A_1	A_2	A_3	A_4
F	0.08333	0.16667	-0.25000	0.00000

Tab. 5: Řešení příkladu č. 2 pomocí metody WSA

	A_1	A_2	A_3	A_4	A_5	A_6
u	0.52045	0.40750	0.15000	0.40909	0.70364	0.54545

Tab. 6: Řešení příkladu č. 2 pomocí metody TOPSIS

	A_1	A_2	A_3	A_4	A_5	A_6
d_i^+	0.09874	0.11375	0.08824	0.07168	0.11665	0.07859
d_i^-	0.04696	0.03875	0.06000	0.11748	0.07191	0.07038
c_i	0.32229	0.25411	0.40478	0.62108	0.38138	0.47247

Tab. 7: Řešení příkladu č. 2 pomocí metody ORESTE

	A_1	A_2	A_3	A_4	A_5	A_6
r_i	64.5	80	90.5	91.5	54	64.5

Tab. 8: Řešení příkladu č. 2 pomocí metody PROMETHEE

	A_1	A_2	A_3	A_4	A_5	A_6
F	0.20	-0.04	-0.52	-0.24	0.4	0.2

Tab. 9: Řešení příkladu č. 3 pomocí metody WSA

	A_1	A_2	A_3	A_4	A_5
u	0.45844	0.37482	0.47500	0.53580	0.54432

Tab. 10: Řešení příkladu č. 3 pomocí metody TOPSIS

	A_1	A_2	A_3	A_4	A_5
d_i^+	0.05574	0.07037	0.04732	0.04142	0.07117
d_i^-	0.03233	0.02240	0.04764	0.06704	0.04153
c_i	0.36708	0.24148	0.50169	0.61811	0.36851

Tab. 11: Řešení příkladu č. 3 pomocí metody ORESTE

	A_1	A_2	A_3	A_4	A_5
r_i	233	275.5	241	235.5	230

Tab. 12: Řešení příkladu č. 3 pomocí metody PROMETHEE

	A_1	A_2	A_3	A_4	A_5
F	0.075	-0.2	0.025	0.05	0.05

Tab. 13: Řešení příkladu č. 4 pomocí metody WSA

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
u	0.432	0.383	0.457	0.510	0.548	0.583	0.607	0.460	0.527

Tab. 14: Řešení příkladu č. 4 pomocí metody TOPSIS

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
d_i^+	0.045	0.056	0.037	0.036	0.057	0.042	0.034	0.031	0.057
d_i^-	0.024	0.017	0.034	0.049	0.031	0.040	0.036	0.043	0.026
c_i	0.351	0.229	0.484	0.575	0.352	0.485	0.510	0.580	0.315

Tab. 15: Řešení příkladu č. 4 pomocí metody ORESTE

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
r_i	488.5	537	431.5	486	447	376	370.5	519.5	445

Tab. 16: Řešení příkladu č. 4 pomocí metody PROMETHEE

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
F	-0.075	-0.2	-0.0625	0.025	0.0375	0.2125	0.2	-0.15	0.0125

Zadání a řešení příkladu č. 5 z kapitoly 3 pomocí všech použitých metod

Tab. 17: Zadání příkladu č. 5

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
C_1	12	15	19	13	14	16	9	20	21	26	18	17	16	13	14	10	24	22	11	16
C_2	5	6	5.5	5.6	5.4	5.3	5.9	5.6	5.55	5.3	5.65	5.8	5.95	6	5.1	5.25	5.2	5.3	5.8	5.7
C_3	52	56	58	49	46	48	50	51	56	54	57	60	48	47	52	53	55	54	47	51
C_4	0.12	0.23	0.14	0.16	0.18	0.11	0.14	0.14	0.18	0.21	0.19	0.14	0.16	0.18	0.22	0.23	0.21	0.18	0.17	0.13
C_5	120	125	160	110	185	210	135	140	180	190	175	130	135	140	200	205	130	140	150	160
C_6	16	18	12	13	14	15	18	16	12	17	21	11	15	16	17	13	18	16	20	20
C_7	6	8	9	4	5	6	7	8	2	5	9	7	6	4	5	6	8	2	3	6
C_8	15	12	18	13	14	15	16	17	19	14	13	15	14	13	14	14	17	16	15	14
C_9	120	130	150	140	120	130	150	140	130	150	160	150	140	130	120	150	160	130	120	150
C_{10}	500	550	560	540	530	520	550	560	540	520	530	550	540	560	550	550	510	590	530	540
C_{11}	1000	1050	1060	1070	1050	1060	1040	1070	1050	1040	1030	1010	1050	1080	1040	1060	1070	1050	1000	1030
C_{12}	16	17	18	14	15	14	16	17	18	13	12	10	18	19	15	16	17	18	13	15
C_{13}	432	435	439	442	435	436	439	432	429	428	445	439	438	435	432	444	428	428	436	438
C_{14}	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7

C_{15}	1	1	0.5	0	0.5	1	0	0	0.5	0.5	1	1	0.5	0	0	0.5	0	0.5	1	0
C_{16}	78	75	74	79	80	81	78	79	75	76	77	77	72	83	81	82	78	79	80	81
C_{17}	5	6	6	7	8	5	6	8	7	5	6	8	5	6	8	7	6	8	7	5
C_{18}	15	13	18	16	15	18	17	12	15	16	13	14	15	16	13	18	17	16	14	15
C_{19}	20	21	22	20	21	22	20	24	22	23	20	21	25	24	23	21	22	21	22	25
C_{20}	7.5	4	3.5	5.5	6	4.5	3.5	4.5	6	5	3	5.5	6.5	2.5	5	4	6.5	3	4.5	5
C_{21}	100	110	120	130	100	120	130	120	130	140	100	100	140	130	120	130	130	140	100	110
C_{22}	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
C_{23}	16	15	18	20	19	21	23	20	17	16	15	23	20	14	18	13	15	17	16	15
C_{24}	150	160	150	130	140	150	130	170	180	130	150	160	140	130	140	140	150	160	180	160
C_{25}	800	820	840	820	800	840	840	860	820	840	820	800	800	820	840	860	880	820	840	820
C_{26}	5	6	7	8	9	10	5	6	7	8	9	10	5	6	7	8	9	10	5	6
C_{27}	10.1	9.5	9.6	10.1	9.7	9.9	9.9	10.0	9.5	9.8	10.2	10.1	9.7	9.9	10	10.2	10.3	10.1	10	9.7
C_{28}	80	82	79	80	83	84	78	79	80	82	81	84	78	76	81	83	79	79	81	83
C_{29}	5.5	5.6	5.3	5.4	5.5	5.7	5.4	5.3	5.2	5.8	5.5	5.6	5.7	5.2	5.5	5.3	5.9	5.1	5.6	5.4
C_{30}	1	0.5	0	1	0	0	1	0.5	0.5	1	0	1	0.5	0	1	0	0.5	0.5	1	1
C_{31}	150	170	130	140	160	180	150	120	130	140	150	160	170	180	130	140	160	120	150	180
C_{32}	999	1000	998	996	1001	1005	997	996	994	995	994	1003	1005	999	998	994	996	1000	1002	1004

C_{33}	1005	1003	997	995	1005	1001	1000	999	997	996	1001	995	1003	1005	999	999	1004	1001	1002	998
C_{34}	10	10.2	10.3	9.9	9.8	10.1	10.3	10.4	9.8	9.6	9.7	10.1	10.2	10.4	9.9	10.0	10.1	9.9	9.8	10.0
C_{35}	18	17	16	13	14	13	14	16	9	10	24	22	11	16	12	15	19	13	14	16
C_{36}	5.65	5.8	5.95	6	5.1	5.6	5.4	5.3	5.9	5.25	5.2	5.3	5.8	5.7	5	6	5.5	5.6	5.4	5.3
C_{37}	57	60	48	47	52	49	46	48	50	53	55	54	47	51	52	56	58	49	46	48
C_{38}	0.19	0.14	0.16	0.18	0.22	0.16	0.18	0.11	0.14	0.23	0.21	0.18	0.17	0.13	0.12	0.23	0.14	0.16	0.18	0.11
C_{39}	175	130	135	140	200	110	185	210	135	205	130	140	150	160	120	125	160	110	185	210
C_{40}	21	11	15	16	17	13	14	15	18	13	18	16	20	20	16	18	12	13	14	15
C_{41}	9	7	6	4	5	4	5	6	7	6	8	2	3	6	6	8	9	4	5	6
C_{42}	13	15	14	13	14	13	14	15	16	14	17	16	15	14	15	12	18	13	14	15
C_{43}	160	150	140	130	120	140	120	130	150	150	160	130	120	150	120	130	150	140	120	130
C_{44}	530	550	540	560	550	540	530	520	550	550	510	590	530	540	500	550	560	540	530	520
C_{45}	1030	1010	1050	1080	1040	1070	1050	1060	1040	1060	1070	1050	1000	1030	1000	1050	1060	1070	1050	1060
C_{46}	12	10	18	19	15	14	15	14	16	16	17	18	13	15	16	17	18	14	15	14
C_{47}	445	439	438	435	432	442	435	436	439	444	428	428	436	438	432	435	439	442	435	436
C_{48}	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7	1.8	1.7
C_{49}	1	1	0.5	0	0	0	0.5	1	0	0.5	0	0.5	1	0	1	1	0.5	0	0.5	1
C_{50}	77	77	72	83	81	79	80	81	78	82	78	79	80	81	78	75	74	79	80	81

Tab. 18: Řešení příkladu č. 5 pomocí nové metody

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
$\mathcal{D}_1$	0.3382111	0.4096388	0.4200993	0.4057877	0.3895574	0.367249	0.4127319	0.4062027	0.4185743	0.3900307
$\mathcal{D}_0$	0.4217889	0.3503612	0.3399007	0.3542123	0.3704426	0.3927511	0.3472681	0.3537973	0.3414257	0.3699692
$\mathcal{D}$	0.0835778	-0.0592776	-0.0801985	-0.0515754	-0.0191148	0.0255022	-0.0654639	-0.0524054	-0.0771486	-0.0200615

	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
$\mathcal{D}_1$	0.3898889	0.3398373	0.3853257	0.4176407	0.4338762	0.3647096	0.308623	0.4325546	0.3773674	0.3909621
$\mathcal{D}_0$	0.370111	0.4201627	0.3746743	0.3423593	0.3261239	0.3952904	0.451377	0.3274453	0.3826326	0.3690379
$\mathcal{D}$	-0.0197779	0.0803255	-0.0106514	-0.0752814	-0.1077523	0.0305808	0.142754	-0.1051093	0.0052652	-0.0219242

Tab. 19: Řešení příkladu č. 5 pomocí metody WSA

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
u	0.51657	0.52535	0.49381	0.43968	0.46141	0.48829	0.47223	0.49181	0.48852	0.53443

	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
u	0.51611	0.54881	0.51381	0.43263	0.45641	0.52831	0.59505	0.45099	0.49561	0.51433

Tab. 20: Řešení příkladu č. 5 pomocí metody TOPSIS

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
d_i^+	0.01071	0.00955	0.01316	0.01392	0.01438	0.01338	0.01403	0.01130	0.01393	0.00943
d_i^-	0.01522	0.01483	0.01015	0.01124	0.00912	0.01214	0.01042	0.01320	0.00976	0.01392
c_i	0.58700	0.60831	0.43552	0.44678	0.38801	0.47576	0.42633	0.53885	0.41202	0.59608

	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
d_i^+	0.01348	0.00954	0.01238	0.01543	0.01315	0.01129	0.01217	0.01332	0.01240	0.01126
d_i^-	0.01297	0.01504	0.01134	0.00950	0.01207	0.01360	0.01228	0.01077	0.01210	0.01379
c_i	0.49043	0.61173	0.47812	0.38108	0.47842	0.54624	0.50222	0.44697	0.49386	0.55039

Tab. 21: Řešení příkladu č. 5 pomocí metody ORESTE

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
r_i	23614.5	25677.5	26353.5	22976.5	23404	25355	23298	25979.5	24375.5	25419

	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
r_i	25826	27504	24578	24625.5	22065.5	26948	30981	23646.5	22920	24952.5

Tab. 22: Řešení příkladu č. 5 pomocí metody PROMETHEE

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
F	-0.05579	0.02421	0.04211	-0.08211	-0.07053	0.01158	-0.06842	0.03579	-0.02632	0.0221

	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
F	0.03474	0.10421	-0.01895	-0.01579	-0.12000	0.07684	0.24526	-0.05789	-0.08211	-0.00105

Příloha 2

Zdrojový kód programu

Část jádra algoritmu.

```

/// <summary>
/// Method computes all particular deltas through all dimension between 2 nodes in structure
/// </summary>

internal float GetEdgeDelta(Node n1, Node n2)
{
    float[] deltas = new float[def.NumberOfCriterias];
    var i = 0;
    n1.criteria.ForEach(crit =>
    {
        deltas[i] = GetParticularDelta(n1.criteria[i], n2.criteria[i], i);
        i++;
    });
    return ComputeD(deltas);
}

/// <summary>
/// This method computes particular delta in one dimension. Applies Zeta functions to whole
/// algorithm.
/// </summary>

internal float GetParticularDelta(float num1, float num2, int dimension)
{
    return (num1 == num2) ? def.Zeta0 : def.Zeta1 * ((num2 - num1) /
    (MaxNode.criteria[dimension] - MinNode.criteria[dimension]));
}

/// <summary>
/// Computes D for whole vector
/// </summary>
internal float ComputeD(float[] deltas)
{
    var dim = 0;
    float result = 0;
    deltas.ToList().ForEach(delta => result += delta * def.criteriumWeight[dim++]);
    return result = result / def.criteriumWeight.Sum();
}

```

CD s aplikací

Součástí práce je i přiložené CD, které obsahuje program.