

Návrh a realizace aplikace metod multikriteriálního hodnocení pro edukativní účely

Bc. Ladislav Langer

Diplomová práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav elektroniky a měření

Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Bc. Ladislav Langer
Osobní číslo:	A23606
Studijní program:	N1032A020003 Bezpečnostní technologie, systémy a management
Specializace:	Bezpečnostní technologie
Forma studia:	Kombinovaná
Téma práce:	Návrh a realizace aplikace metod multikriteriálního hodnocení pro edukativní účely
Téma práce anglicky:	Design and Implementation of the Application of Multi-Criteria Evaluation Methods for Educational Purposes

Zásady pro vypracování

1. Charakterizujte metody multikriteriální analýzy a zhodnoťte možnosti uplatnění dané multikriteriální metody v praxi, se zaměřením na oblast bezpečnosti.
2. Zpracujte literární rešerši softwarových nástrojů týkající se této problematiky.
3. Navrhněte a vytvořte obecný postup návrhu a realizace aplikace.
4. Zdokumentujte návrh konceptu aplikace pro vybrané metody multikriteriálního hodnocení variant.
5. Popište realizaci navržené aplikace.
6. Ověřte funkčnost aplikace na příkladu z oblasti bezpečnosti.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. FOTR, Jiří a ŠVECOVÁ, Lenka. Manažerské rozhodování: postupy, metody a nástroje rozhodování v dynamickém a nejistém prostředí. Čtvrté, přepracované vydání. Jesenice: Ekopress, 2022. ISBN 978-80-87865-76-7.
2. ANDERSON, David Ray; SWEENEY, Dennis J.; WILLIAMS, Thomas Arthur; WISNIEWSKI, Mik a PIERRON, Xavier. *An introduction to management science: quantitative approaches to decision making*. Third edition. Andover: Cengage Learning, 2017. ISBN 978-1-4737-2932-2.
3. SAATY, Thomas L., 2008. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*. March 18, 2008, vol. 2008, no. 1, s. 83-98. ISSN 1753-1454.
4. DOUMPOS, Michael; FIGUEIRA, José Rui Matos; GRECO, Salvatore a ZOPOUNIDIS, Constantin (ed.). New perspectives in multiple criteria decision making: innovative applications and case studies. Multiple criteria decision. Cham: Springer, 2019. ISBN 978-3-030-11481-7.
5. KULKARNI, Anand J. *Multiple Criteria Decision Making: Techniques, Analysis and Applications*. Singapore: Springer, 2022. ISBN 978-981-16-7414-3.
6. GWO-HSHIUNG, Tzeng a KAO-YI, Shen. *New Concepts and Trends of Hybrid Multiple Criteria Decision Making*. Boca Raton: CRC Press, 2017. ISBN 9781315166650.
7. FLANAGAN, David. *JavaScript: the definitive guide : master the world's most-used programming language*. Seventh edition. Beijing: O'Reilly, 2020. ISBN 978-1-491-95202-3.
8. MATTHES, Eric. *Python crash course: a hands-on, project-based introduction to programming*. 2nd edition. San Francisco: No Starch Press, 2019. ISBN 9781593279295.

Vedoucí diplomové práce: **Ing. David Šaur, PhD.**
Ústav matematiky

Datum zadání diplomové práce: **29. listopadu 2024**

Termín odevzdání diplomové práce: **30. května 2025**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



Ing. Milan Navrátil, Ph.D. v.r.
ředitel ústavu

Ve Zlíně dne 29. listopadu 2024

PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE

Beru na vědomí, že

- odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení diplomové práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – diplomovou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na diplomové práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Tato diplomová práce se zaměřuje na návrh a realizaci webové aplikace pro multikriteriální analýzu, která umožňuje uživatelům aplikovat vybrané metody hodnocení variant, jako jsou WSM, WPM, TOPSIS, ELECTRE a MABAC. Cílem je vytvořit edukativní nástroj pro vizualizaci rozhodovacích procesů s důrazem na bezpečnostní aplikace. Práce zahrnuje charakterizaci metod MCA, literární rešerši softwarových nástrojů a návrh obecného postupu vývoje aplikace. Funkčnost aplikace je ověřena na modelovém příkladu z oblasti bezpečnosti, přičemž výsledkem je modulární řešení umožňující analýzu rozhodovacích variant na základě různých metod multikriteriálního hodnocení.

Klíčová slova: multikriteriální analýza, rozhodovací metody, WSM, WPM TOPSIS, ELECTRE, MABAC, bezpečnostní analýza, softwarová aplikace, vizualizace rozhodování

ABSTRACT

This thesis focuses on the design and implementation of a web-based application for multi-criteria analysis, allowing users to apply selected evaluation methods such as WSM, WPM, TOPSIS, ELECTRE and MABAC. The goal is to create an educational tool for visualizing decision-making processes with an emphasis on security applications. The study includes an overview of MCA methods, a literature review of relevant software tools, and a general approach to application development. The functionality of the application is validated through a security-related case study, resulting in a modular solution for analysing decision-making variants using various multi-criteria evaluation methods.

Keywords: Multi-criteria analysis, decision-making methods, WSM, WPM, TOPSIS, ELECTRE, MABAC, security analysis, software application, decision visualization

Upřímně děkuji Ing. Davidu Šaurovi, Ph.D., vedoucímu této diplomové práce, za odborné vedení, cenné připomínky a trpělivou podporu, které významně přispěly k úspěšnému dokončení této práce.

OBSAH

ÚVOD.....	11
I TEORETICKÁ ČÁST	13
1 ZÁKLADY MULTIKRITERIÁLNÍHO HODNOCENÍ.....	14
1.1 HISTORIE A VÝVOJ MULTIKRITERIÁLNÍHO HODNOCENÍ.....	14
1.1.1 Počátky multikriteriálního hodnocení	14
1.1.2 Tři dekády vývoje multikriteriálního hodnocení (1970-2000): klíčové algoritmy a osobnosti	16
1.1.3 Nové algoritmy a metodiky multikriteriálního hodnocení po roce 2000	18
1.2 ZÁKLADNÍ POJMY A PRINCIPY	20
1.2.1 Definice základních pojmů	20
1.2.2 Princip multikriteriálního rozhodování	22
1.2.3 Model a rozhodovací funkce	22
1.3 VYBRANÉ METODY MULTIKRITERIÁLNÍ ANALÝZY	23
1.3.1 WSM	23
1.3.2 WPM	25
1.3.3 TOPSIS	27
1.3.4 ELECTRE	31
1.3.5 MABAC	35
1.4 HODNOCENÍ VYUŽITELNOSTI VYBRANÝCH METOD V BEZPEČNOSTNÍCH APLIKACÍCH.....	38
2 REŠERŠE DOSTUPNÝCH SOFTWAREVÝCH ŘEŠENÍ.....	41
2.1 OBLASTI ZKOUMÁNÍ DOSTUPNÝCH SOFTWAREVÝCH NÁSTROJŮ.....	41
2.2 PŘEHLED VYBRANÝCH SOFTWAREVÝCH NÁSTROJŮ	42
2.3 SHRNUÍ SOFTWAREVÝCH ŘEŠENÍ PRO MCDM A SYNTÉZA PRO NÁVRH VLASTNÍHO NÁSTROJE	46
II PRAKTICKÁ ČÁST.....	49
3 ÚVOD DO PRAKTICKÉ ČÁSTI – SOFTWAREVÁ IMPLEMENTACE METOD MULTIKRITERIÁLNÍ ANALÝZY	50
4 OBECNÝ POSTUP NÁVRHU APLIKACE	53
4.1 SPECIFIKACE POŽADAVKŮ	53
4.2 METODIKA NÁVRHU A VÝVOJE.....	55
5 KONCEPTUÁLNÍ NÁVRH A REALIZACE APLIKACE	56
5.1 ARCHITEKTURA APLIKACE	56
5.2 DATOVÝ MODEL	58
6 ALGORITMIZACE ANALYTICKÝCH VÝPOČTŮ VÍCEKRITERIÁLNÍHO HODNOCENÍ A STANOVENÍ VAH.....	60
6.1 NORMALIZACE HODNOT	60

6.2	ALGORITMUS WSM	61
6.3	ALGORITMUS WPM	61
6.4	ALGORITMUS TOPSIS.....	62
6.5	ALGORITMUS ELECTRE	64
6.6	ALGORITMUS MABAC	65
6.7	ALGORITMUS AHP PRO STANOVENÍ VAH	66
6.8	ALGORITMUS METODY ENTROPIE	68
7	OVĚŘENÍ FUNKČNOSTI APLIKACE.....	69
7.1	PŘÍPADOVÁ STUDIE – ZADÁNÍ.....	69
7.2	PŘÍPADOVÁ STUDIE – VSTUPNÍ DATA	70
7.3	PŘÍPADOVÁ STUDIE – HODNOCENÍ VARIANT	71
7.4	PŘÍPADOVÁ STUDIE – INTERPRETACE VÝSLEDKŮ A SHRUTÍ ANALÝZY	72
ZÁVĚR.....		73
SEZNAM POUŽITÉ LITERATURY		75
SEZNAM OBRÁZKŮ		78
SEZNAM TABULEK.....		79
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		80
SEZNAM PŘÍLOH.....		82

ÚVOD

V současném dynamickém prostředí organizace a instituce neustále čelí složitým rozhodovacím problémům, které vyžadují aplikaci sofistikovaných metod. Strukturování rozhodovacích procesů na základě metod multikriteriálního hodnocení (MCDM z angl. *Multi-criteria decision making*)¹ umožňuje efektivnější alokaci zdrojů a optimalizaci rozhodnutí, zejména ve vícerozměrných problémech s protichůdnými cíli. Tato diplomová práce se soustředí na aplikaci MCDM v kontextu bezpečnostních analýz, kde je rozhodování podmíněno nejen ekonomickými faktory, ale také aspekty spolehlivosti a míry kritičnosti potenciálních hrozeb.

Cílem práce je návrh a realizace softwarové aplikace, jež propojuje pět reprezentativních metod multikriteriálního hodnocení (WSM, WPM, TOPSIS, ELECTRE a MABAC) a vytváří tak ucelenou vzdělávací platformu pro vizualizaci rozhodovacích procesů. Aplikace dovoluje uživatelům definovat problém, stanovit váhy kritérií (přímo, pomocí párového porovnání, nebo entropie) a okamžitě porovnat výsledky různých algoritmů v jednotném prostředí.

Teoretická část práce pokrývá fundamentální koncepty MCDM, zahrnující historický vývoj, základní axiomatické principy a metodologické přístupy. V této souvislosti je analyzováno spektrum metod, od klasických kompenzačních modelů až po pokročilé outranking techniky. Dále je diskutováno, jakým způsobem lze tyto metody integrovat do softwarového řešení a jaké jsou jejich praktické implikace.

Důležitou součástí teoretické části je literární rešerše zaměřená na existující softwarové nástroje pro multikriteriální rozhodování, které hrají stále významnější roli v datech řízeném rozhodování. Rešerše porovnává vybrané komerční i open-source systémy z hlediska funkcionality, metodologických přístupů a omezení. Tímto způsobem umožňuje nejen zasadit nově navrženou aplikaci do kontextu současných řešení, ale také identifikovat mezery, které navržená aplikace překonává.

Praktická část dokumentuje návrh třívrstvé architektury aplikace MCAApp, postavené na platformě Anvil s backendem v Pythonu 3.10, responsivním rozhraním a datovým modelem kombinujícím relační tabulky s NoSQL prvky. Modularita kódu umožňuje snadné

¹ V odborné literatuře často také MCDA (z angl. *Multi-criteria decision analysis*, příp. *aiding*)

doplnění nových metod nebo analytických modulů, zatímco vestavěná autentizace založená na JWT a granulárních rolích zajišťuje bezpečnost dat.

Funkčnost aplikace je demonstrována případovou studií výběru domácího antivirového řešení, která potvrzuje správnost implementovaných algoritmů i přínos vizuální analýzy pro konečné rozhodnutí.

Závěrem práce je představena perspektiva dalšího rozvoje – integrace fuzzy modelů, neuronových sítí a hybridních přístupů kombinujících MCDM s umělou inteligencí – s cílem vytvořit univerzální open-source platformu, jež bude sloužit akademické sféře i praxi při řešení komplexních bezpečnostních úloh.

Poznámka: Pokud není u tabulek nebo obrázků uveden zdroj, jedná se o vlastní tvorbu autora. Vzhledem k tomu, že práce čerpá převážně ze zahraničních (anglicky psaných) odborných zdrojů, jsou v textu překlady názvů doplněny i o jejich původní znění v angličtině. Výjimku tvoří běžně používané a všeobecně známé názvy a zkratky, zejména z oblasti informačních technologií, které jsou ponechány bez doplnění originálního názvu.

Podrobnosti o dostupnosti zdrojového kódu, možnostech online přístupu k aplikaci MCAApp a instrukce pro testování jsou uvedeny v **příloze P I** této práce.

I TEORETICKÁ ČÁST

1 ZÁKLADY MULTIKRITERIÁLNÍHO HODNOCENÍ

Tato kapitola představuje teoretická východiska a koncepty multikriteriálního rozhodování (MCDM), přičemž klade důraz na základní principy, klíčové pojmy a klasifikaci metod využívaných v této oblasti. Úvodem je podrobně popsán historický vývoj MCDM, který sahá od prvních neformálních metod až po současné sofistikované postupy. Mezi významné osobnosti, jejichž práce položily základy multikriteriálního hodnocení, patří například Benjamin Franklin, Nicolas de Condorcet, Francis Edgeworth, Vilfredo Pareto a později také průkopníci matematického programování jako George Dantzig nebo Bernard Roy.

V další části kapitoly jsou systematicky definovány klíčové pojmy potřebné k pochopení principů multikriteriální analýzy. Patří sem zejména definice pojmů kritérium, váha kritéria, normalizace a hodnocení alternativ.

Následně jsou představeny vybrané metody multikriteriální analýzy, jež tvoří jádro této práce. Popsány jsou klasické metody, mezi které patří například WSM, WPM, TOPSIS, ELECTRE a MABAC. U jednotlivých metod jsou diskutovány jejich specifické vlastnosti, přednosti a omezení, a dále kritéria, podle kterých je vhodné volit jednotlivé metody s ohledem na charakter a povahu řešeného problému.

V závěrečné části kapitoly je pozornost zaměřena na využití metod multikriteriálního hodnocení v oblasti bezpečnosti. Jsou zde představeny konkrétní aplikace těchto metod v procesech krizového plánování, při hodnocení bezpečnostních rizik a v návrhu účinných bezpečnostních opatření. Kapitola tímto způsobem poskytuje komplexní základ pro pochopení multikriteriálního rozhodování v kontextu praktických bezpečnostních problémů.

1.1 Historie a vývoj multikriteriálního hodnocení

V této části první kapitoly, která se věnuje historickému vývoji algoritmů a metodologii MCDM práce čerpá z odborné publikace *Multiple Criteria Decision Making: From Early History to the 21st Century*. [1]

1.1.1 Počátky multikriteriálního hodnocení

Rozhodovací procesy zahrnující více kritérií se objevují již v hluboké historii, avšak jejich systematické studium jako samostatné vědecké disciplíny se začalo formovat až ve 20. století. První explicitní zmínky o metodickém přístupu k rozhodování lze nalézt u amerického politika a vědeckého myslitele Benjamina Franklina. Ve své korespondenci

s chemikem Josephem Priestleym Franklin popisuje jednoduchou metodu rozhodovací analýzy, při níž se argumenty pro a proti zapsané na list papíru podrobí subjektivnímu váhovému srovnání a postupně se eliminují faktory s nižším vlivem. Ačkoliv Franklin neformuloval formální matematický model, jeho metoda představovala jeden z prvních strukturovaných rozhodovacích postupů.

Souběžně s Franklinem se rozhodovacími procesy zabýval také Marquis de Condorcet, francouzský matematik a politický filozof. Ve svém díle *Esej o aplikaci analýzy na pravděpodobnost většinového rozhodování* (1785)² formuloval tzv. Condorcetův paradox, jenž dokazuje, že agregované rozhodnutí skupiny nemusí být transitivní, i když jednotliví voliči preferují alternativy v transitivním pořadí. Tento koncept měl klíčový dopad na teorii hlasování a skupinové rozhodování, což představuje jeden z fundamentálních aspektů moderního multikriteriálního rozhodování.

Další důležitý přínos k rozvoji tohoto oboru přinesl Francis Edgeworth zavedením konceptu indifferenčních křivek, které umožňují znázornit kompromis (tzv. trade-off) mezi různými kritérii. Tento přístup se stal základním nástrojem v ekonomii a operačním výzkumu. Na jeho práci navázal Vilfredo Pareto, který formalizoval koncept Paretovo optimum. Pareto definoval efektivní řešení jako takové, při němž není možné zlepšit jedno kritérium, aniž by se současně nezhoršilo jiné. Tento princip se stal pilířem moderní teorie multikriteriální optimalizace.

Ve 20. století se rozvoj multikriteriálního rozhodování propojuje s teorií užitku a matematickým programováním. John von Neumann a Oskar Morgenstern v roce 1944 ve své průlomové publikaci *Theory of Games and Economic Behavior* formulovali teorii očekávané užitečnosti, která umožňuje formální modelování rozhodování v podmínkách nejistoty. Tato teorie se stala zásadním výchozím bodem pro mnohé multikriteriální metody, protože poskytla nástroj pro kvantifikaci preferencí rozhodovatele.

Klíčovou roli v oblasti matematického programování sehrál George Dantzig, jenž v roce 1947 vyvinul simplexovou metodu jako efektivní algoritmus pro řešení lineárních optimalizačních úloh. Tento algoritmus umožnil řešení složitých optimalizačních problémů zahrnujících více kritérií. Paralelně s Dantzigem působil v Sovětském svazu Leonid

² Z fr. orig.: *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*

Kantorovich, jehož práce *Matematické metody organizace a plánování výroby*³ v oblasti ekonomického plánování položila základ pro další rozvoj víceúčelového programování.

Dalším důležitým milníkem byla formulace konceptu efektivních řešení. Tjalling Koopmans ve své studii *Analysis of Production as an Efficient Combination of Activities* (1951) zavedl pojem efektivního vektoru, což představovalo aplikaci Paretova optima v kontextu alokace zdrojů. Tato myšlenka se stala základem pro rozvoj vícekriteriálního matematického programování.

V polovině 20. století se vedle metod založených na užitkové teorii a optimalizaci objevila nová třída multikriteriálních metod, tzv. outranking přístupy. Průkopníkem tohoto směru byl Bernard Roy, jenž v roce 1968 představil první verzi metody ELECTRE (Élimination et Choix Traduisant la Réalité). Tato metoda umožňuje porovnávat alternativy na základě jejich vzájemné dominance, aniž by bylo nutné agregovat jednotlivá kritéria do jednoho užitkového skóre. ELECTRE se stala klíčovou outranking metodou a spolu s metodou PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations) položil základy tzv. francouzské školy multikriteriálního rozhodování.

1.1.2 Tři dekády vývoje multikriteriálního hodnocení (1970-2000): klíčové algoritmy a osobnosti

V 70. letech 20. století došlo k zásadnímu rozvoji metod a algoritmů MCDM. Klíčovým přínosem bylo rozpracování multiatributní užitkové teorie (MAUT), kterou systematizovali Ralph Keeney a Howard Raiffa ve své monografii *Decisions with Multiple Objectives: Preferences and Value Tradeoffs* (1976). Tato teorie poskytla matematický rámec pro kvantifikaci preferencí při rozhodování s více kritérii, čímž významně přispěla k formalizaci rozhodovací analýzy.

V oblasti interaktivních metod se prosadily přístupy, které umožňovaly rozhodovateli aktivně participovat na procesu hledání optimálního řešení. Významnými průkopníky byli Geoffrion, Dyer a Feinberg (1972), kteří vyvinuli metodu postupné aproximace efektivní množiny (Geoffrion-Dyer-Feinberg algorithm, GDF). Podobný přístup navrhli Zionts a Wallenius (1976), kteří kladli důraz na interaktivní dotazování rozhodovatele a průběžné zpřesňování preferencí. Současně Ralph E. Steuer zdokonalil algoritmy pro generování

³ Z rus. orig.: *Matematicheskie metody organizatsii i planirovaniya proizvodstva*

efektivních řešení ve vícekritériálním lineárním programování, což umožnilo jejich širší praktické nasazení.

Outranking metody, umožňující porovnávání alternativ bez nutnosti jejich agregace do jediné užitkové funkce, byly významně rozvinuty Bernardem Royem. V 70. letech navrhl ELECTRE II a III, které využívají párová srovnání alternativ a umožňují pracovat i s nekompletní komparabilitou variant.

Thomas L. Saaty představil Analytický hierarchický proces (AHP), který přinesl nový způsob strukturování složitých rozhodovacích problémů do hierarchických úrovní. Metoda AHP využívá párová srovnání a umožňuje kvantifikaci subjektivních preferencí rozhodovatele, čímž se stala široce aplikovanou v rozhodovací praxi.

V 80. letech pokračoval rozvoj MCDM metod, přičemž důraz byl kladen na softwarovou implementaci rozhodovacích modelů. Ralph E. Steuer v roce 1986 systematizoval algoritmy vícekritériálního matematického programování ve své knize *Multiple Criteria Optimization*, která se stala základní referencí pro akademické i aplikační využití.

Mezi nové algoritmické přístupy, které získaly významné postavení, patřila metoda PROMETHEE, kterou vyvinuli Brans, Vincke a Mareschal (1986). PROMETHEE zavádí zobecněné kritériální funkce pro srovnávání alternativ, což umožňuje větší flexibilitu při rozhodování. Dalším průlomem bylo využití evolučních algoritmů pro řešení multikritériálních optimalizačních problémů. J.D. Schaffer vyvinul Vector Evaluated Genetic Algorithm (VEGA) (1984), první víceúčelový genetický algoritmus, který aplikoval evoluční přístupy na hledání Pareto-optimálních řešení.

Paralelně se rozvíjely fuzzy přístupy k multikritériálnímu rozhodování, které umožňovaly zohlednit neurčitost a vágnost v rozhodovacích modelech. Saaty v 80. letech dále rozpracoval AHP, jehož implementace v softwarových nástrojích přispěla k širšímu využití této metody ve veřejném i soukromém sektoru.

V 90. letech došlo k výraznému zlepšení evolučních víceúčelových optimalizačních metod (EMO). Vyvinuly se pokročilé genetické algoritmy, jako MOGA (Multiobjective Genetic Algorithm), NSGA (Nondominated Sorting Genetic Algorithm) a PGA (Pareto Genetic Algorithm), které umožnily efektivní aproximaci Pareto-optimálních množin v komplexních rozhodovacích problémech.

Další významnou inovací byla metoda ELECTRE-TRI, která umožnila klasifikaci alternativ do předem definovaných kategorií na základě outranking principu. Ve stejné dekádě došlo k zásadnímu rozvoji webových aplikací rozhodovacích systémů, přičemž klíčovým milníkem byl WWW-NIMBUS (1995), první online implementace interaktivních víceúčelových modelů⁴.

V oblasti strojového učení se začaly prosazovat přístupy kombinující MCDM s teorií Rough sets, což vedlo k efektivnějšímu modelování preferencí při práci s neúplnými nebo nejistými daty. Tyto hybridní metody našly využití v diagnostických systémech, analýze rizik a predikčních modelech.

Postupná integrace MCDM algoritmů do standardních optimalizačních softwarů vedla k jejich širokému nasazení v průmyslu, veřejné správě a strategickém plánování. Metody MCDM se uplatnily v široké škále aplikací od optimalizace výrobních procesů, přes dopravní a urbanistické plánování, až po environmentální a energetická rozhodování.

Díky pokroku v algoritmech a výpočetních technologiích se multikriteriální rozhodování etablovalo jako klíčová součást moderní analytiky rozhodovacích procesů, přičemž jeho rozvoj pokračuje v interdisciplinárním propojení s datovou vědou, umělou inteligencí a kognitivními vědami.

1.1.3 Nové algoritmy a metodiky multikriteriálního hodnocení po roce 2000

V novém tisíciletí došlo k významnému rozšíření metod multikriteriálního rozhodování (MCDM), přičemž mnohé nové přístupy vycházely z klasických metod a dále je rozvíjely. Mezi nejvýznamnější patří metoda VIKOR (Visekriterijumska Optimizacija i Kompromisno Resenje), publikovaná v roce 2004, která se zaměřuje na kompromisní hodnocení alternativ. Další významné techniky, které se v tomto období rozvinuly, zahrnují MOORA (Multi-Objective Optimization by Ratio Analysis, 2006), COPRAS (Complex Proportional Assessment, 2004–2013) a EDAS (Evaluation Based on Distance from Average Solution, 2015). Tyto metody se zařadily mezi nejčastěji aplikované přístupy v oblasti inženýrství, strategického řízení a ekonomického modelování a rozšířily spektrum dostupných rozhodovacích metod.

⁴ Aplikace WWW-NIMBUS je k 4. 3. 2025 stále dostupná na webových stránkách University of Jyväskylä (<https://wwwnimbus.it.jyu.fi/>).

Vedle zavedení nových metod došlo také k významnému rozšíření existujících přístupů. Příkladem je Analytic Network Process (ANP), který byl představen Thomasem Saatym v roce 2004 jako rozšíření metody AHP. Na rozdíl od AHP, která předpokládá nezávislost kritérií, umožňuje ANP modelovat i vzájemné vztahy mezi kritérii a alternativami, čímž poskytuje robustnější analytický rámec pro složité rozhodovací problémy. [2]

Současně došlo k výraznému pokroku v oblasti evolučních výpočetních metod a vícekritériální optimalizace. Mezi nejdůležitější přístupy patří NSGA-II (Non-dominated Sorting Genetic Algorithm II), publikovaný Kalyanmoyem Debem et al. v roce 2002, který se stal nejrozšířenější metaheuristickou metodou pro řešení vícekritériálních optimalizačních úloh. Následně byly vyvinuty pokročilejší varianty jako NSGA-III, přičemž genetické algoritmy a evoluční strategie se stále častěji uplatňují v inženýrských aplikacích, dopravním plánování a průmyslové optimalizaci. Tyto metody umožňují efektivní nalezení Pareto-optimálních řešení a překonávají výpočetní omezení tradičních optimalizačních algoritmů.

Významným trendem v tomto období byla hybridizace MCDM s jinými optimalizačními metodami, zejména s fuzzy logikou, která umožňuje lépe modelovat neurčitost a subjektivní hodnocení v rozhodovacím procesu. Kombinace fuzzy přístupů s klasickými metodami jako AHP a TOPSIS vedla ke vzniku fuzzy variant těchto metod, které si našly široké uplatnění v oblasti manažerského rozhodování a environmentálních analýz. Další přístupy kombinují MCDM s heuristickými algoritmy, jako jsou mravenčí kolonie nebo simulované žíhání, což zlepšuje efektivitu prohledávání rozsáhlých rozhodovacích prostorů.

Od počátku 21. století rovněž roste zájem o propojení MCDM se strojovým učením, což vede k vývoji nových hybridních algoritmů kombinujících metody MCDM s modely umělé inteligence. Významným příkladem tohoto trendu je metoda FAMCDM (Fusion Approach of MCDM), která kombinuje více rozhodovacích metod pro hodnocení výkonu klasifikačních modelů strojového učení. Obdobně byly vyvinuty přístupy, kde se výsledky více rozhodovacích metod agregují pomocí statistických či machine learning technik s cílem dosažení konsenzuálního hodnocení.

Tyto inovace ukazují rostoucí integraci MCDM s datovou vědou, zejména v situacích, kde je k dispozici velké množství dat a je třeba automatizovat nastavení vah kritérií nebo predikovat preference rozhodovatele. Stále větší dostupnost výpočetních kapacit a pokročilých optimalizačních algoritmů umožňuje efektivnější aplikaci MCDM v široké

škále rozhodovacích problémů, včetně průmyslového inženýrství, dopravního plánování, finanční analýzy a zdravotnictví.

1.2 Základní pojmy a principy

Níže uvedené pojmy a principy tvoří teoretický rámec pro aplikaci MCDM a jsou využívány v dalších částech práce. Definice vycházejí z literatury *Multiple Criteria Decision Making: Techniques, Analysis and Applications*. [3]

1.2.1 Definice základních pojmů

- **Varianta** (někdy též alternativa či scenario) v kontextu MCDM je konkrétní navržená možnost řešení, mezi kterými se rozhodujeme. Typicky se označí jako množina

$$A = \{a_1, a_2, \dots, a_m\},$$

kde m je počet dostupných variant.

- **Kritérium** je hledisko, podle něhož hodnotíme jednotlivé alternativy. Kritéria mohou být *kvantitativní* měřitelná číslem (např. cena, doba zpracování, kapacita), nebo *kvalitativní*: obtížně měřitelná veličina (např. spokojenost, reputace, míra bezpečnosti).

Množinu kritérií značíme:

$$C = \{c_1, c_2, \dots, c_n\},$$

kde n je počet kritérií. Často se rozlišují *maximalizační* (vyšší hodnota je lepší) a *minimalizační* (nižší hodnota je lepší) kritéria.

- **Váha kritéria w_j** odráží relativní důležitost kritéria c_j . Váhy typicky normalizujeme tak, aby platilo:

$$\sum_{j=1}^n w_j = 1 \quad \text{a} \quad w_j \geq 0 \quad \text{pro všechna } j.$$

- Hodnoty kritérií pro každou alternativu sestavujeme do tzv. **rozhodovací matice**:

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix},$$

kde x_{ij} reprezentuje hodnotu kritéria c_j pro variantu a_j . Dimenze matice je tedy $m \times n$.

- **Normalizace** se používá na převod různých jednotek (např. čas, peníze, počet chyb atd.) na srovnatelnou stupnici $[0,1]$ resp. $[-1,1]$. Pro lineární *poměrovou normalizaci* (příklad pro maximalizační kritérium) se často používá vzorec:

$$r_{ij} = \frac{x_{ij}}{\max_{1 \leq i \leq m} x_{ij}},$$

pak můžeme pracovat s normalizovanou maticí $R = (r_{ij})$.

Hodnotíme-li podle maximalizačních a minimalizačních kritérií zároveň, můžeme použít universální lineární transformaci – *Min-max normalizaci*:

$$r_{ij} = \frac{x_{ij} - \min_i x_{ij}}{\max_i x_{ij} - \min_i x_{ij}} \quad \text{pro maximalizační kritéria,}$$

$$r_{ij} = \frac{\max_i x_{ij} - x_{ij}}{\max_i x_{ij} - \min_i x_{ij}} \quad \text{pro minimalizační kritéria.}$$

Kromě zmíněných lineárních typů normalizace existuje celá řada dalších přístupů. Volba konkrétní metody závisí na povaze dat, na požadavcích metody. Další často používané normalizace jsou:

1. *Vektorizace* (tzv. ℓ_2 -norma) – každý údaj x_{ij} se dělí odmocninou z celkového součtu druhých mocnin v daném sloupci:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}},$$

V tomto případě (maximalizačním) platí, že hodnoty r_{ij} se pohybují v intervalu $[0,1]$, ale interpretace je jiná: výsledkem je, že součet druhých mocnin ve sloupci je roven 1.

2. *Logaritmická normalizace* – používá se, pokud mají hodnoty kritéria různá měřítka řádově (např. od 10^1 po 10^5). Zpravidla se pak berou logaritmy původních dat, aby se snížil extrémní rozptyl:

$$r_{ij} = \frac{\ln(x_{ij})}{\sum_{i=1}^m \ln(x_{ij})}.$$

1.2.2 Princip multikriteriálního rozhodování

Postup MCDM lze shrnout do několika obecných kroků:

1. Definice cíle a identifikace variant – určíme, jakého cíle chceme dosáhnout, a vybereme možné způsoby, jimiž lze cíle dosáhnout.
2. Výběr kritérií – určíme, které faktory (kvantitativní a kvalitativní) budou hrát roli při hodnocení.
3. Určení vah kritérií – stanovíme či odhadneme relativní důležitost jednotlivých kritérií, někdy se aplikují metody párového srovnání, bodové škály, nebo entropie.
4. Hodnocení variant podle kritérií – shromáždíme potřebná data, popř. expertní odhady.
5. Aplikace vhodné MCDM metody – na základě zvoleného algoritmu či postupu integrujeme informace o kritériích a váhách do jedné výsledné preference či pořadí variant.
6. Interpretace výsledků a rozhodnutí – výsledek může mít formu pořadí variant podle preferencí, výběru té nejlepší, příp. identifikace kompromisních variant.

1.2.3 Model a rozhodovací funkce

V mnoha případech je cílem nalézt hodnotu tzv. *agregované užítkové funkce* $U(A_i)$ pro každou variantu. Jednoduchý lineární agregační model má např. tvar:

$$U(A_i) = \sum_{j=1}^n w_j f_j(x_{ij}),$$

kde f_j je *normalizační* funkce pro kritérium C_j . Ta zajistí, že se různé druhy veličin převedou na srovnatelné *užitky* (obvykle v intervalu $[0,1]$). Výsledné hodnoty $U(A_i)$ pak lze porovnat a určit pořadí variant.

$$\underbrace{x_{ij}}_{\text{fyzikální hodnota}} \xrightarrow{f_j} \underbrace{u_{ij} = f_j(x_{ij})}_{\text{užitek v } [0,1]} \xrightarrow{\text{váha } w_j} \underbrace{U(a_i)}_{\text{souhrnná hodnota}}$$

Tento lineární (vážený) součet je zdaleka nejrozšířenější díky jednoduchosti a srozumitelnosti.

1.3 Vybrané metody multikriteriální analýzy

V této podkapitole jsou představeny a stručně charakterizovány vybrané metody multikriteriální analýzy, které budou dále aplikovány v praktické části práce.

1.3.1 WSM

Popis této metody byl čerpán z článku *Sensitivity of Decisions to Probability Estimation Errors: A Reexamination*. [4]

WSM (z angl. *Weighted Sum Model*), známá také jako SAW (z angl. *Simple Additive Weighting*) metoda je založena na principu vážené sumy, kde se každé kritérium nejprve normalizuje a následně je vynásobeno svou váhou, která vyjadřuje jeho relativní důležitost. Výsledná hodnota umožňuje porovnat varianty a seřadit je podle dosaženého skóre.

Metoda WSM je vhodná pro případy, kdy lze jednotlivá kritéria normalizovat na stejnou škálu a přiřadit jim jasné váhy. Používá se v hodnocení projektů, analýze efektivity, investičním rozhodování či výběru nejvhodnější alternativy.

Klasifikace metody:

- Typ rozhodovacího problému: *MADM (hodnocení konečné množiny variant)*
- Způsob práce s kritérii: *kvantitativní metoda (vyžaduje numerické hodnoty)*
- Typ modelu: *deterministický (nepracuje s neurčitostí)*
- Agregační metoda: *lineární vážená suma*
- Preferenční struktura: *kompensační – slabá hodnota jednoho kritéria může být vyvážena silnou hodnotou jiného kritéria*

Matematický aparát:

Nechť máme m variant $(A_1, A_2, \dots, A_m)$ a n kritérií $(C_1, C_2, \dots, C_n)$. Každá varianta je hodnocena podle jednotlivých kritérií a každé kritérium má přiřazenou váhu.

WSM používá *Min-max normalizaci* a pro *vážený součtový model*: se výsledné hodnocení varianty A_j vypočítá podle vzorce:

$$S_j = \sum_{i=1}^n w_j r_{ij} ,$$

kde:

- S_j je celkové skóre varianty A_j ,
- w_j je váha kritéria C_i , přičemž platí $\sum w_i = 1$,
- r_{ij} je normalizovaná hodnota varianty A_j podle kritéria C_i .

Příklad 1: Výpočet WSM

Máme 3 varianty (A_1, A_2, A_3) a 3 kritéria (C_1, C_2, C_3), kde první dvě kritéria jsou maximalizační a třetí kritérium je minimalizační. Váhy kritérií jsou:

$$w_1 = 0,5, w_2 = 0,3, w_3 = 0,2$$

Tabulka 1 Příklad 1 – rozhodovací matice

Varianta	C ₁ (max)	C ₂ (max)	C ₃ (min)
A ₁	50	3	120
A ₂	40	5	150
A ₃	30	2	100

Tabulka 2 Příklad 1 – lineárně min-max normalizovaná matice

Varianta	C ₁ (max)	C ₂ (max)	C ₃ (min)
A ₁	$\frac{50 - 30}{50 - 30} = 1,00$	$\frac{3 - 2}{5 - 2} = 0,33$	$\frac{150 - 120}{150 - 100} = 0,60$
A ₂	$\frac{40 - 30}{50 - 30} = 0,50$	$\frac{5 - 2}{5 - 2} = 1,00$	$\frac{150 - 150}{150 - 100} = 0,00$
A ₃	$\frac{30 - 30}{50 - 30} = 0,00$	$\frac{2 - 2}{5 - 2} = 0,00$	$\frac{150 - 100}{150 - 100} = 1,00$

Výpočet finálního skóre

$$S_1 = (1,00 \times 0,5) + (0,33 \times 0,3) + (0,60 \times 0,2) = 0,720$$

$$S_2 = (0,50 \times 0,5) + (1,00 \times 0,3) + (0,00 \times 0,2) = 0,550$$

$$S_3 = (0,00 \times 0,5) + (0,00 \times 0,3) + (1,00 \times 0,2) = 0,200$$

Výsledné pořadí variant

1. $A_1(0,720)$ - nejlepší
2. $A_2(0,550)$
3. $A_3(0,200)$ – nejhorší

1.3.2 WPM

Matematický základ této metody byl čerpán z článku *Sensitivity of Decisions to Probability Estimation Errors: A Reexamination*. [4]

WPM (z angl. *Weighted Product Model*), se od WSM liší tím, že místo sčítání vážených hodnot využívá součin jednotlivých normalizovaných hodnot, přičemž každá je umocněna na hodnotu odpovídající její váze. Výsledkem je relativní poměr mezi variantami, nikoli absolutní hodnota užítku.

Metoda WPM se používá v analýze efektivity, výběru investičních možností či optimalizaci technologických procesů.

Klasifikace metody:

- Typ rozhodovacího problému: *MADM*
- Způsob práce s kritérii: *kvantitativní metoda*
- Typ modelu: *deterministický*
- Agregační metoda: *multiplikativní součin vážených normalizovaných hodnot*
- Preferenční struktura: *kompensační – méně než WSM, protože normalizované hodnoty jsou vzájemně závislé*

Matematický aparát:

Nechť máme m variant $(A_1, A_2, \dots, A_m)$ a n kritérií $(C_1, C_2, \dots, C_n)$. Každá varianta je hodnocena podle jednotlivých kritérií a každé kritérium má přiřazenou váhu.

WPM lze normalizovat tak, aby všechna kritéria byla maximalizační, tj. pro maximalizační kritéria jsou platné původní hodnoty x_{ij} a pro minimalizační kritéria se použijí převrácené hodnoty $\frac{1}{x_{ij}}$

Pro vážený součinnový model: se výsledné hodnocení varianty A_j vypočítá podle vzorce:

$$S_j = \prod_{i=1}^n r_{ij}^{w_i},$$

kde:

- S_j je celkové skóre varianty A_j ,
- w_i je váha kritéria C_i , přičemž platí $\sum w_i = 1$,
- r_{ij} je normalizovaná hodnota varianty A_j podle kritéria C_i .

Příklad 2: Výpočet WPM

Máme 3 varianty (A_1, A_2, A_3) a 3 kritéria (C_1, C_2, C_3), kde první dvě kritéria jsou maximalizační a třetí kritérium je minimalizační. Váhy kritérií jsou:

$$w_1 = 0,5, w_2 = 0,3, w_3 = 0,2$$

Tabulka 3 Příklad 2 – rozhodovací matice

Varianta	C_1 (max)	C_2 (max)	C_3 (min)
A_1	50	3	120
A_2	40	5	150
A_3	30	2	100

Tabulka 4 Příklad 2 – matice se standardizovanými kritérii

Varianta	C_1 (max)	C_2 (max)	C_3 (min)
A_1	50	3	$\frac{1}{120}$
A_2	40	5	$\frac{1}{150}$
A_3	30	2	$\frac{1}{100}$

Výpočet finálního skóre

$$S_1 = (50)^{0,5} \cdot (3)^{0,3} \cdot \left(\frac{1}{120}\right)^{0,2} = 3,773$$

$$S_2 = (40)^{0,5} \cdot (5)^{0,3} \cdot \left(\frac{1}{150}\right)^{0,2} = 3,763$$

$$S_3 = (30)^{0,5} \cdot (2)^{0,3} \cdot \left(\frac{1}{100}\right)^{0,2} = 2,684$$

Tabulka 5 Příklad 2 – výsledné pořadí variant

Varianta	WPM skóre	Relativní WPM	Pořadí
A ₁	3,773	1,000	1
A ₂	3,763	0,997	2
A ₃	2,684	0,7711	3

1.3.3 TOPSIS

Popis metody čerpá z knihy *Methods for multiple attribute decision making*. [5]

TOPSIS (z angl. *Technique for Order of Preference by Similarity to Ideal Solution*) vychází z myšlenky, že nejlepší varianta je ta, která má nejmenší vzdálenost od ideálního řešení (tj. nejlepší možné hodnoty všech kritérií) a zároveň největší vzdálenost od nejhoršího možného řešení (tj. nejhorší hodnoty všech kritérií). Na základě těchto vzdáleností se vypočítá relativní blízkost k ideálnímu řešení, což umožní seřazení variant.

Hlavní kroky metody TOPSIS:

1. Normalizace hodnot kritérií.
2. Vážená normalizace hodnot.
3. Určení ideálního a anti-ideálního řešení.
4. Výpočet euklidovské vzdálenosti variant od obou referenčních bodů.
5. Výpočet relativní blízkosti k ideálnímu řešení a seřazení variant.

Tato metoda je vhodná tam, kde je nutné hledat variantu co nejbližší optimálním hodnotám. Používá se v hodnocení investic, výběru technologických procesů, logistické optimalizaci či analýze výkonnosti podniků.

Klasifikace metody:

- Typ rozhodovacího problému: *MADM*
- Způsob práce s kritérii: *kvantitativní metoda*
- Typ modelu: *deterministický, nicméně existují fuzzy varianty*
- Agregáčnı metoda: *vzdálenost od ideálního a anti-ideálního řešení*
- Preferenční struktura: *kompensační – umožňuje trade-off mezi kritérii*

Matematický aparát:

Nechť máme m variant ($A_1, A_2 \dots, A_m$) a n kritérií ($C_1, C_2 \dots, C_n$). Každá varianta je hodnocena podle jednotlivých kritérií a každé kritérium má přiřazenou váhu.

TOPSIS používá *vektorovou normalizaci*, která zachovává metrickou konzistenci při výpočtu vzdáleností v n -rozměrném prostoru:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}},$$

kde x_{ij} je původní hodnota kritéria C_i , pro variantu A_j a r_{ij} je normalizovaná hodnota.

Vzdálenost varianty od *ideálního řešení* D_j^+ a *anti-ideálního řešení* D_j^- se vypočítá:

$$D_j^+ = \sqrt{\sum_{i=1}^n (v_{ij} - A_i^+)^2} \quad \text{a} \quad D_j^- = \sqrt{\sum_{i=1}^n (v_{ij} - A_i^-)^2},$$

kde *vážená normalizovaná hodnota* $v_{ij} = w_i r_{ij}$ a A_i^+ je *ideální varianta* resp. A_i^- je *anti-ideální varianta*. Ideální a anti-ideální varianta se určí:

$$A^+ = (\max v_{ij} \text{ pro maximalizační kritéria, } \min v_{ij} \text{ pro minimalizační kritéria})$$

$$A^- = (\min v_{ij} \text{ pro maximalizační kritéria, } \max v_{ij} \text{ pro minimalizační kritéria})$$

Relativní blízkost každé varianty k ideálnímu řešení se vypočítá:

$$C_j^* = \frac{D_j^-}{D_j^+ + D_j^-}$$

Varianty se seřadí podle hodnoty C_j^* sestupně, přičemž nejlepší varianta je ta s nejvyšší hodnotou.

Příklad 3: Výpočet TOPSIS

Máme 3 varianty (A_1, A_2, A_3) a 3 kritéria (C_1, C_2, C_3), kde první dvě kritéria jsou maximalizační a třetí kritérium je minimalizační. Váhy kritérií jsou:

$$w_1 = 0,5, w_2 = 0,3, w_3 = 0,2$$

Tabulka 6 Příklad 3 – hodnoty kritérií před normalizací

Varianta	C ₁ (max)	C ₂ (max)	C ₃ (min)
A ₁	50	3	120
A ₂	40	5	150
A ₃	30	2	100

Výpočet odmocniny ze součtu čtverců pro každé kritérium:

$$\sqrt{50^2 + 40^2 + 30^2} = 70,71$$

$$\sqrt{3^2 + 5^2 + 2^2} = 6,16$$

$$\sqrt{120^2 + 150^2 + 100^2} = 216,67$$

Tabulka 7 Příklad 3 – normalizované hodnoty

Varianta	r_{1j}	r_{2j}	r_{3j}
A ₁	$\frac{50}{70,71} = 0,707$	$\frac{3}{6,16} = 0,487$	$\frac{120}{216,67} = 0,554$
A ₂	$\frac{40}{70,71} = 0,566$	$\frac{5}{6,16} = 0,812$	$\frac{150}{216,67} = 0,692$
A ₃	$\frac{30}{70,71} = 0,424$	$\frac{2}{6,16} = 0,325$	$\frac{100}{216,67} = 0,462$

Tabulka 8 Příklad 3 - vážené normalizované hodnoty

Varianta	v_{1j}	v_{2j}	v_{3j}
A ₁	$0,5 \cdot 0,707 = 0,354$	$0,3 \cdot 0,487 = 0,146$	$0,2 \cdot 0,554 = 0,111$
A ₂	$0,5 \cdot 0,556 = 0,283$	$0,3 \cdot 0,812 = 0,244$	$0,2 \cdot 0,692 = 0,138$
A ₃	$0,5 \cdot 0,424 = 0,212$	$0,3 \cdot 0,325 = 0,098$	$0,2 \cdot 0,707 = 0,092$

Určení ideální a anti-ideální hodnoty:

$$A^+ = (\max v_{1j}, \max v_{2j}, \min v_{3j}) = (0,354, 0,244, 0,092)$$

$$A^- = (\min v_{1j}, \min v_{2j}, \max v_{3j}) = (0,212, 0,098, 0,138)$$

Vzdálenost varianty A_j od ideálního a anti-ideálního řešení:

$$D_1^+ = \sqrt{(0,354 - 0,354)^2 + (0,146 - 0,244)^2 + (0,111 - 0,092)^2} = 0,099$$

$$D_2^+ = \sqrt{(0,283 - 0,354)^2 + (0,244 - 0,244)^2 + (0,138 - 0,092)^2} = 0,085$$

$$D_3^+ = \sqrt{(0,212 - 0,354)^2 + (0,098 - 0,244)^2 + (0,092 - 0,092)^2} = 0,203$$

$$D_1^- = \sqrt{(0,354 - 0,212)^2 + (0,146 - 0,098)^2 + (0,111 - 0,138)^2} = 0,152$$

$$D_2^- = \sqrt{(0,283 - 0,212)^2 + (0,244 - 0,098)^2 + (0,138 - 0,138)^2} = 0,162$$

$$D_3^- = \sqrt{(0,212 - 0,212)^2 + (0,098 - 0,098)^2 + (0,092 - 0,138)^2} = 0,046$$

Výpočet relativní blízkosti k ideálnímu řešení:

$$C_1^* = \frac{0,152}{0,099 + 0,152} = 0,606$$

$$C_2^* = \frac{0,162}{0,085 + 0,162} = 0,658$$

$$C_3^* = \frac{0,046}{0,203 + 0,046} = 0,185$$

Výsledné pořadí variant:

1. A₂(0,658) - nejlepší
2. A₁(0,606)
3. A₃(0,185) – nejhorší

1.3.4 ELECTRE

Použití této metody je popsáno v článku *A Comprehensive Overview of the ELECTRE Method in Multi Criteria Decision-Making*. [6]

ELECTRE (z fran. *Élimination et Choix Traduisant la Réalité*) je rodina outrankingových metod multikriteriálního rozhodování, která se používá pro výběr, řazení nebo třídění variant. Každá z verzí ELECTRE má specifické vlastnosti, které jsou přizpůsobené různým typům rozhodovacích problémů (viz tab. 8). Práce dále popisuje pouze verzi ELECTRE I pro výběr nejlepší varianty.

Tabulka 9 Porovnání variant metody ELECTRE

Verze	Typ rozhodnutí	Použití vah	Použití prahů preference (p), indiference (q), veto (v) ⁵
I	Výběr nejlepší varianty (eliminace horších), výstupem je množina preferovaných variant	Ano	Ne (rozhoduje pouze na základě indexů souhlasu a nesouhlasu)
II	Seřazení variant od nejlepší po nejhorší	Ano	Ne (rozhoduje pouze na základě indexů souhlasu a nesouhlasu)
III	Seřazení variant s vágností (fuzzy přístup)	Ano	Ano (p, q, v)
IV	Výběr bez použití vah, výstupem je množina preferovaných variant	Ne	Ne
TRI	Kategorizace variant dle referenčních limitů	Ano	Ano (p, q, v)

Princip ELECTRE I spočívá v porovnávání každé dvojice variant (A_i, A_j) a určení, zda jedna varianta *překonává* (outranks) druhou na základě souhlasu většiny kritérií a absence silné neshody v jakémkoliv kritériu.

⁵ Preferenční prahy:

q (indifference threshold) – rozdíl mezi dvěma variantami je tak malý, že nejsou rozlišitelné

p (preference threshold) – minimální rozdíl, kdy lze říct, že jedna varianta je lepší než druhá

v (veto threshold) – pokud je rozdíl mezi variantami příliš velký, jedna varianta je automaticky vyřazena

Postup metody ELECTRE I (výběr nejlepší varianty):

1. Normalizace hodnot kritérií.
2. Výpočet *concordance* (indexu souhlasu) – určuje, do jaké míry varianta A_i překonává A_j .
3. Výpočet *discordance* (indexu nesouhlasu) – měří, do jaké míry některé kritérium silně nesouhlasí s tím, že A_i překonává A_j .
4. Určení *outrankingové relace* – pokud varianta A_i překonává A_j , pak je zařazena do doporučeného souboru variant.

ELECTRE I je vhodná pro složité rozhodovací problémy, kde nelze jednoduše agregovat všechna kritéria do jedné hodnoty. Používá se v strategickém rozhodování, plánování projektů, výběru dodavatelů a veřejných zakázkách.

Klasifikace metody:

- Typ rozhodovacího problému: *MADM (hodnocení konečné množiny variant)*
- Způsob práce s kritérii: *kvantitativní i kvalitativní metoda (umožňuje práci s jazykovými preferencemi)*
- Typ modelu: *deterministický (pozn. existují fuzzy varianty)*
- Agregační metoda: *outrankingová relace (porovnání variant mezi sebou)*
- Preferenční struktura: *nekompenzační – silně horší kritérium nelze vždy kompenzovat jinými kritérii*

Matematický aparát:

Nechť máme m variant $(A_1, A_2, \dots, A_m)$ a n kritérií $(C_1, C_2, \dots, C_n)$. Každá varianta je hodnocena podle jednotlivých kritérií a každé kritérium má přiřazenou váhu.

ELECTRE používá lineární *poměrovou* (z angl. *ratio*) *normalizaci*.

Concordance pro dvojici variant A_i a A_j je dána vzorcem:

$$C_{ij} = \sum_{k \in S_{ij}} w_k,$$

kde w_k je váha kritéria C_k a S_{ij} je množina kritérií, ve které platí, A_i je lepší než A_j .

Discordance je dána vzorcem:

$$D_{ij} = \max_{k \in N_{ij}} \frac{x_{kj} - x_{ki}}{\max x_k - \min x_k},$$

kde N_{ij} je množina kritérií, ve které platí, A_i je lepší než A_j .

Varianta A_i překonává variantu A_j , pokud platí:

$$C_{ij} \geq c^* \text{ a } D_{ij} \leq d^*,$$

kde c^* a d^* jsou prahové hodnoty.

Příklad 4: Výpočet ELECTRE I

Máme 3 varianty (A_1, A_2, A_3) a 3 kritéria (C_1, C_2, C_3), kde první dvě kritéria jsou maximalizační a třetí kritérium je minimalizační. Váhy kritérií jsou:

$$w_1 = 0,5, w_2 = 0,3, w_3 = 0,2$$

Tabulka 10 Příklad 4 - hodnoty kritérií před normalizací

Varianta	C ₁ (max)	C ₂ (max)	C ₃ (min)
A ₁	50	3	120
A ₂	40	5	150
A ₃	30	2	100

Tabulka 11 Příklad 4 – poměrově normalizované hodnoty

Varianta	C ₁ (max)	C ₂ (max)	C ₃ (min)
A ₁	50 / 50 = 1,00	3 / 5 = 0,60	100 / 120 = 0,83
A ₂	40 / 50 = 0,80	5 / 5 = 1,00	100 / 150 = 0,67
A ₃	30 / 50 = 0,60	2 / 5 = 0,40	100 / 100 = 1,00

Tabulka 12 Příklad 4 – matice indexu souhlasu

Varianta	A ₁	A ₂	A ₃
A ₁	-	$0,5 + 0,2 = 0,7$	$0,5 + 0,3 = 0,8$
A ₂	0,3	-	$0,5 + 0,3 = 0,8$
A ₃	0,2	0,2	-

Maximální a minimální hodnoty, v kritériích:

$$C_1(max): \quad max = 1,00, \quad min = 0,60 \quad \Rightarrow \quad rozdíl = 1,00 - 0,60 = 0,40$$

$$C_2(max): \quad max = 1,00, \quad min = 0,40 \quad \Rightarrow \quad rozdíl = 1,00 - 0,40 = 0,60$$

$$C_3(min): \quad max = 1,00, \quad min = 0,67 \quad \Rightarrow \quad rozdíl = 1,00 - 0,67 = 0,33$$

Tabulka 13 Příklad 4 – matice indexu nesouhlasu

Varianta	A ₁	A ₂	A ₃
A ₁	-	$0,4 / 0,6 = 0,67$	$0,17 / 0,6 = 0,28$
A ₂	0,33	-	$0,33 / 0,6 = 0,56$
A ₃	0,67	1	-

Určení outranking relace:

Pokud nastavíme prahové hodnoty:

- $c^* = 0,7$ (minimální požadovaná úroveň souhlasu)
- $d^* = 0,5$ (maximální akceptovatelná úroveň nesouhlasu)

pak:

- A₁ nepřevyšuje A₂ (index nesouhlasu $0,67 > d^*$)
- A₁ převyšuje A₃
- A₂ nepřevyšuje A₁ (index souhlasu $0,3 < c^*$)
- A₂ nepřevyšuje A₃ (index nesouhlasu $0,56 > d^*$)

- A_3 nepřevyšuje A_1 (index souhlasu $0,2 < c^*$ a index nesouhlasu $0,67 > d^*$)
- A_3 nepřevyšuje A_2 (index souhlasu $0,2 < c^*$ a index nesouhlasu $1,00 > d^*$)

Výsledné pořadí variant:

1. A_1 (převyšuje jednu variantu $A_3 \Rightarrow$ skóre 1) - nejlepší
2. A_2 (skóre 0)
3. A_3 (je převýšen jednou variantou $A_1 \Rightarrow$ skóre - 1) - nejhorší

1.3.5 MABAC

Koncept metody je popsán v článku *The selection of transport and handling resources in logistics centers using Multi-Attributive Border Approximation area Comparison*. [7]

Metoda MABAC (z angl. *Multi-Attributive Border Approximation Area Comparison*) je založena na konceptu hraniční aproximační oblasti (BAA), která představuje mezní hodnotu mezi lepšími a horšími variantami. Hodnoty kritérií jsou porovnávány s touto oblastí a na základě toho je určeno konečné pořadí variant.

Klasifikace metody:

- Typ rozhodovacího problému: *MADM*
- Způsob práce s kritérii: *kvantitativní metoda, ale lze upravit pro kvalitativní data*
- Typ modelu: *deterministický, nicméně existují fuzzy varianty*
- Agregační metoda: *vzdálenost od hraniční aproximační oblasti*
- Preferenční struktura: *kompensační*

Matematický aparát:

Nechť máme m variant ($A_1, A_2, \dots, A_m$) a n kritérií ($C_1, C_2, \dots, C_n$). Každá varianta je hodnocena podle jednotlivých kritérií a každé kritérium má přiřazenou váhu.

MABAC pro normalizaci hodnot rozhodovací matice používá různé vzorce podle typu kritéria.

Hraniční hodnota G_i pro kritérium C_i se vypočítá jako geometrický průměr všech variant pro dané kritérium:

$$G_i = \left(\prod_{j=1}^m v_{ij} \right)^{\frac{1}{m}}$$

Hodnota každé varianty se určuje jako rozdíl mezi její váženou normalizovanou hodnotou a hraniční aproximační oblastí, tedy $Q_{ij} = v_{ij} - G_i$, kde Q_{ij} představuje skóre varianty vzhledem k hraniční aproximační oblasti. Čím vyšší je Q_{ij} , tím je varianta lepší, protože se nachází dále od hraniční oblasti směrem k optimálním hodnotám kritérií.

Celkové skóre varianty A_j je dáno součtem přes všechna kritéria:

$$S_j = \sum_{i=1}^n Q_{ij}$$

Příklad 5: Výpočet MABAC

Máme 3 varianty (A_1, A_2, A_3) a 3 kritéria (C_1, C_2, C_3), kde první dvě kritéria jsou maximalizační a třetí kritérium je minimalizační. Váhy kritérií jsou:

$$w_1 = 0,5, w_2 = 0,3, w_3 = 0,2$$

Tabulka 14 Příklad 5 - hodnoty kritérií před normalizací

Varianta	C ₁ (max)	C ₂ (max)	C ₃ (min)
A ₁	50	3	120
A ₂	40	5	150
A ₃	30	2	100

Tabulka 15 Příklad 5 – lineárně Min-max normalizovaná matice

Varianta	C ₁ (max)	C ₂ (max)	C ₃ (min)
A ₁	$\frac{50 - 30}{50 - 30} = 1,00$	$\frac{3 - 2}{5 - 2} = 0,33$	$\frac{150 - 120}{150 - 100} = 0,60$
A ₂	$\frac{40 - 30}{50 - 30} = 0,50$	$\frac{5 - 2}{5 - 2} = 1,00$	$\frac{150 - 150}{150 - 100} = 0,00$
A ₃	$\frac{30 - 30}{50 - 30} = 0,00$	$\frac{2 - 2}{5 - 2} = 0,00$	$\frac{150 - 100}{150 - 100} = 1,00$

Tabulka 16 Příklad 5 - vážené normalizované hodnoty⁶

Var.	v_{1j}	v_{2j}	v_{3j}
A ₁	$0,5 \cdot (1,00 + 1) = 1,00$	$0,3 \cdot (0,33 + 1) = 0,40$	$0,2 \cdot (0,60 + 1) = 0,32$
A ₂	$0,5 \cdot (0,50 + 1) = 0,75$	$0,3 \cdot (1,00 + 1) = 0,60$	$0,2 \cdot (0,00 + 1) = 0,20$
A ₃	$0,5 \cdot (0,00 + 1) = 0,50$	$0,3 \cdot (0,00 + 1) = 0,30$	$0,2 \cdot (1,00 + 1) = 0,40$

Určení hraniční aproximační oblasti:

$$G_1 = (1,00 \cdot 0,75 \cdot 0,50)^{\frac{1}{3}} = 0,721$$

$$G_2 = (0,40 \cdot 0,60 \cdot 0,30)^{\frac{1}{3}} = 0,416$$

$$G_3 = (0,32 \cdot 0,20 \cdot 0,40)^{\frac{1}{3}} = 0,295$$

Tabulka 17 Příklad 5 matice vzdáleností od hraniční aproximační oblasti

Var.	C ₁ (max)	C ₂ (max)	C ₃ (min)
A ₁	$1,00 - 0,721 = 0,279$	$0,40 - 0,416 = -0,016$	$0,32 - 0,295 = 0,025$
A ₂	$0,75 - 0,721 = 0,029$	$0,60 - 0,416 = 0,184$	$0,20 - 0,295 = 0,095$
A ₃	$0,50 - 0,721 = -0,221$	$0,30 - 0,416 = -0,116$	$0,40 - 0,295 = 0,105$

Tabulka 18 Příklad 5 pořadí variant

Varianta	Celková hodnota kritériální funkce (S_i)	Pořadí variant
A ₁	$0.279 + (-0.016) + 0.025 = 0.288$	1
A ₂	$0.029 + 0.184 + (-0.095) = 0.118$	2
A ₃	$-0.221 + (-0.116) + 0.295 = -0.232$	3

⁶ Výpočet geometrického průměru pro BAA by mohl být *nestabilní* nebo méně smysluplný při práci s hodnotami blízkými nule. Přičtení „+1“ ve vzorci MABAC $v_{ij} = w_j \cdot (n_{ij} + 1)$ také jasněji odliší horní a dolní aproximační oblast.

1.4 Hodnocení využitelnosti vybraných metod v bezpečnostních aplikacích

Níže uvedené tabulky (tab. 19 a tab. 20) shrnují porovnání pěti vybraných metod multikriteriální analýzy (WSM, WPM, TOPSIS, ELECTRE I, MABAC) v kontextu jejich použitelnosti pro bezpečnostní aplikace.

Tabulka 19 Souhrn vybraných metod MCDM

Hodnocení	WSM	WPM	TOPSIS	ELECTRE I	MABAC
Princip rozhodování	Vážená suma	Vážený součin	Vzdálenost od ideálu	Outranking (párové porovnání)	Vzdálenost od hraniční oblasti
Transparentnost výsledků	Vysoká	Střední	Vysoká	Nížší	Vysoká
Práce s neurčitostí	Nízká	Nízká	Střední	Vysoká	Střední
Schopnost kompenzace kritérií	Vysoká	Vysoká	Vysoká	Nízká (omezená)	Vysoká
Vhodná datová základna	Kvantitativní (přesná data)	Kvantitativní (poměrová data)	Kvantitativní (přesná data)	Kvantitativní a kvalitativní (expertní hodnocení)	Kvantitativní (přesná data)
Interpretace výsledků	Velmi snadná	Snadná (poměrová interpretace)	Snadná (intuitivní)	Náročnější (outranking)	Snadná (vzdálenosti od hranice)
Hlavní výhody	Jednoduchost, jasné výsledky	Relativní porovnání kritérií	Jasná interpretace výsledků	Práce s neurčitostí, vhodnost pro složité problémy	Vyváženost, jednoduchá interpretace
Hlavní nevýhody	Citlivost na normalizaci, vyžaduje přesná data	Omezená interpretace (pouze relativní poměry)	Citlivost na normalizaci	Složitější interpretace výsledků, vyšší náročnost výpočtu	Závislost na správné normalizaci

Metody se liší svým základním přístupem k rozhodování – některé využívají přímou agregaci hodnot kritérií (WSM, WPM), jiné posuzují vzdálenosti od ideálních hodnot (TOPSIS, MABAC), případně se zaměřují na párová porovnávání a outrankingové relace (ELECTRE I).

Z hlediska transparentnosti a jednoduchosti interpretace výsledků se jako nejvhodnější jeví metody WSM, TOPSIS a MABAC, které umožňují přehledné a snadno interpretovatelné výsledky. Metoda ELECTRE I naopak poskytuje robustní výsledky při vágních preferencích a komplexních problémech, nicméně interpretace výsledků je náročnější a vyžaduje větší zkušenosti rozhodovatelů.

Pokud jde o práci s neurčitostí a nejistotou, největší schopnosti nabízí právě metoda ELECTRE I, vhodná pro expertní hodnocení a krizové situace s neurčitými daty. Metody jako WSM a WPM jsou výrazně závislé na přesnosti vstupních dat a nejsou schopny efektivně pracovat s neurčitostí. Metody TOPSIS a MABAC jsou v této oblasti flexibilnější, nicméně rovněž vyžadují relativně přesnou kvantitativní datovou základnu.

Níže uvedená tabulka (tab. 20) doplňuje tyto informace a přináší praktický pohled na konkrétní situace v bezpečnostní praxi, pro které jsou jednotlivé metody vhodné. Umožňuje tak lepší orientaci při volbě vhodné metody podle specifických požadavků řešeného bezpečnostního problému.

Tabulka 20 Porovnání využitelnosti metod MCDM v bezpečnostních aplikacích

Typ bezpečnostní aplikace	Konkrétní příklady bezpečnostních úloh	WSM	WPM	TOPSIS	ELCR. I	MABAC
Hodnocení bezpečnostních rizik	Analýza a prioritizace hrozeb, výběr oblastí s nejvyšší úrovní rizika					
Výběr technologií a vybavení	Výběr kamerových systémů, protipožárních zařízení, zabezpečení vstupních bodů					
Hodnocení variant krizových opatření	Porovnání scénářů evakuace, výběr havarijních opatření					

Plánování evakuace a optimalizace cest	Návrh tras nouzové evakuace, výběr evakuačních koridorů, rozmístění únikových cest					
Hodnocení scénářů ohrožení infrastruktury	Hodnocení variant ochrany kritické infrastruktury, výběr preventivních opatření					
Strategické krizové řízení za nejistoty	Expertní posouzení krizových scénářů, rozhodování v případě nedostatku přesných dat					
Lokalizace bezpečnostních prvků	Optimalizace rozmístění kamer, hlídek nebo detektorů, výběr lokality skladů materiálu					
Rychlé rozhodování v krizových situacích	Okamžité posouzení variantních řešení v situacích s časovým omezením					
Bezpečnostní rozhodnutí založená na expertních odhadech	Hodnocení hrozeb na základě expertního panelu, porovnání opatření při nejasných preferencích					

Legenda: ● *Nejvhodnější metoda* | ● *Velmi vhodná metoda* | ● *Středně vhodná metoda* | ● *Omezeně vhodná metoda*

Dílčí závěr:

Kapitola poskytla přehled historického vývoje a základních principů multikriteriálního rozhodování. Byly objasněny klíčové pojmy a popsány vybrané metody MCA, včetně jejich výhod a omezení. Zvláštní důraz byl kladen na praktickou využitelnost těchto metod v oblasti bezpečnostních aplikací, čímž kapitola vytvořila teoretický rámec pro praktickou část práce.

2 REŠERŠE DOSTUPNÝCH SOFTWAREVÝCH ŘEŠENÍ

Tato kapitola se zaměřuje na analýzu dostupných softwarových nástrojů pro multikriteriální rozhodování z hlediska jejich funkční šířky, uživatelské přívětivosti, technické dostupnosti a edukačního potenciálu. Cílem není pouze poskytnout čtenáři přehled aktuální nabídky a usnadnit orientaci v různých licenčních modelech, technických platformách a metodologických přístupech, ale také využít získané poznatky jako podklad pro návrh a realizaci vlastního softwarového řešení s důrazem na jeho vzdělávací využití.

2.1 Oblasti zkoumání dostupných softwarových nástrojů

Pro vytvoření uceleného obrazu o současných softwarových prostředcích pro multikriteriální rozhodování je nezbytné systematicky zmapovat jednotlivé typy těchto nástrojů. Teprve obsáhnutím všech relevantních typů řešení lze zajistit dostatečně reprezentativní přehled. Rešerše pokrývá tyto relevantní kategorie:

a) Podle typu licencování

Z pohledu licencování jsou nástroje rozdělovány na:

- *Open-source nástroje*: Zdrojový kód je veřejně dostupný, nástroj může být upravován a dále rozvíjen komunitou. Uživatelé nejsou omezováni licenčními poplatky.
- *Komerční nástroje*: Jsou poskytovány na základě placené licence, často s podporou od vývojáře, vyšší úrovní zabezpečení a širší funkcionalitou.

b) Podle formy a technologického zázemí

Softwarová řešení se liší i formou technického provedení:

- *Desktopové aplikace*: Jsou instalovány lokálně na zařízení uživatele. Nabízejí obvykle vyšší výkon, ale nižší flexibilitu.
- *Webové aplikace*: Jsou přístupné přes internetový prohlížeč. Uživatel nemusí provádět instalaci, výhodou je snadná dostupnost a podporu více platforem.
- *Programovací knihovny*: Jsou určeny pro pokročilé uživatele, kteří provádějí analýzy v prostředí jako Python, R nebo MATLAB. Nabízejí vysokou flexibilitu a možnost integrace do vlastních systémů.

c) Podle účelu a doménového zaměření

Nástroje lze dále rozlišit podle míry univerzálnosti:

- *Obecné nástroje*: Umožňují aplikaci metod napříč různými doménami bez specializace. Jsou určeny široké skupině uživatelů (výzkumníkům, studentům, analytikům).
- *Doménově specifické nástroje*: Jsou navrhovány pro konkrétní oblast rozhodování, např. dopravu, environmentální plánování nebo krizové řízení. Obsahují často předdefinované šablony a datové struktury, které odpovídají dané oblasti.

Tato klasifikace slouží jako rámec pro následující přehled konkrétních softwarových nástrojů, které jsou popsány v další podkapitole.

2.2 Přehled vybraných softwarových nástrojů

V této podkapitole je uveden souhrnný přehled dvanácti reprezentativních nástrojů, které jsou v literatuře často zmiňovány pro podporu multikriteriálního rozhodování. Pro každý nástroj jsou uvedeny klíčové informace o autorovi či společnosti, licenčním modelu, podporovaných metodách, charakteristických vlastnostech a hlavních omezeních.

Expert Choice Desktop 11.5 (Desktop aplikace) [8]

- Dostupné: www.expertchoice.com
- Autor (společnost): Expert Choice Inc., Arlington (VA, USA), založena 1983 E. Formanem a T. Saatyem
- Licence: plně komerční (per-seat a síťová); zvýhodněné akademické balíčky
- Metody: AHP; podpora ANP realizována prostřednictvím samostatného modulu SuperDecisions (tento modul je i samostatná aplikace)
- Charakteristika: stromové GUI pro tvorbu hierarchie a párová porovnání, automatická kontrola konzistence, what-if scénáře a citlivostní analýza, modul Comparion pro kolaborativní hodnocení více účastníků a přípravu on-line reportů
- Omezení: implementuje výhradně AHP, dostupné pouze pro Windows; vyšší pořizovací cena a strmější křivka učení pro pokročilé funkce

SuperDecisions v3 (Desktop aplikace) [9]

- Dostupné: www.superdecisions.com
- Instituce: Creative Decisions Foundation (T. Saaty)
- Licence: základ zdarma, pokročilé moduly placené
- Metody: AHP, ANP
- Charakteristika: umožňuje modelovat kruhové závislosti a zpětné vazby mezi klíčovými prvky; zabudovaná citlivostní analýza a skupinové hlasování
- Omezení: Windows; strmější křivka učení při tvorbě sítí

Visual PROMETHEE (Desktop aplikace) [10]

- Dostupné: bertrand.mareschal.web.ulb.be/promethee.html
- Autor, instituce: Bertrand Mareschal, Université Libre de Bruxelles; (komerční distribuce Visual PROMETHEE Team)
- Licence: placená, existuje bezplatná Academic Edition
- Metody: PROMETHEE I–VI + GAIA vizualizace
- Charakteristika: robustní desktopová implementace outrankingové rodiny s interaktivní GAIA rovinou a analýzou citlivosti
- Omezení: podporuje pouze metody PROMETHEE; jen Windows; plná funkčnost závisí na placené licenci

Decision Lab 2025 (Desktop / webová aplikace) [11]

- Dostupné: www.decisionlab.se
- Vývojář: DecisionLab AB, Švédsko
- Licence: komerční (předplatné, zkušební verze)
- Metody: PROMETHEE I–II, GAIA, rozšiřitelné moduly pro portfoliové analýzy
- Charakteristika: modernizovaný nástupce původního Decision Lab; důraz na kolaborativní workflow, sdílení scénářů a cloudovou synchronizaci
- Omezení: metodicky omezeno na rodinu PROMETHEE; vyšší cenová hladina u podnikových plánů

D-Sight Online (Webová aplikace) [12]

- Dostupné: *www.d-sight.com*
- Společnost: D-Sight SA, Brusel
- Licence: komerční SaaS (měsíční / roční předplatné)
- Metody: AHP, PROMETHEE, VIKOR, portfoliové výběry
- Charakteristika: plně webová platforma s podporou auditů, dashboardy a snadným exportem reportů
- Omezení: pouze předplatné

Web-HIPRE v2.0 (Webová aplikace) [13]

- Dostupné: *hipre.aalto.fi*
- Autoři: Systems Analysis Laboratory, Aalto University (R. P. Hämäläinen et al.)
- Licence: free-for-academic use
- Metody: AHP, SMART, SWING, SMARTER (rank-based)
- Charakteristika: ve webovém prohlížeči lze přepínat mezi metodami pro stanovení vah a okamžitě porovnávat výsledky, modely se ukládají lokálně a lze je sdílet
- Omezení: zaměřeno na hierarchické/aditivní modely; pro rozsáhlé datové sady nepoužitelné

PriEsT (Desktop aplikace) [14]

- Dostupné: *sourceforge.net/projects/priority*
- Autoři: Sajid Siraj, Leonid Mikhailov & James Keane
- Licence: GPL v3 (open-source, Java)
- Metody: AHP (více algoritmů odvození vah)
- Charakteristika: lehká multiplatformní aplikace s kontrolou konzistence, evoluční optimalizací a exportem XML
- Omezení: podporuje výhradně AHP; poslední oficiální vydání 2017.

1000minds (web – SaaS) [15]

- Dostupné: www.1000minds.com
- Společnost: 1000minds Ltd., Nový Zéland
- Licence: komerční, akademické slevy
- Metoda: PAPRIKA (stanovení vah pomocí párového srovnání)
- Charakteristika: intuitivní rozhraní připomínající anketu, nástroj generuje váhové sady použitelné v ekonomických modelech
- Omezení: k dispozici jediná metoda, komplexní scénáře mohou vyžadovat mnoho párových otázek

AHP-OS (Webová aplikace) [16]

- Dostupné: bpmmsg.com/ahp/
- Autor: Klaus D. Goepel (spol. BPMMSG)
- Licence: bezplatné – zdrojový kód na GitHubu
- Metody: individuální i skupinový AHP,
- Charakteristika: vícejazyčné UI, automatická exportní funkce
- Omezení: omezeno pouze na AHP; chybí pokročilejší vizualizace.

Visual PROMETHEE-Cloud (Webová aplikace) [17]

- Dostupné: promethee.pom.uni-due.de/
- Autoři: Erik Pohl & Jutta Geldermann, Universität Duisburg-Essen
- Licence: zdarma (open-access beta)
- Metody: PROMETHEE I–II, GAIA
- Charakteristika: plně webová implementace, ukládání scénářů a sdílení odkazem
- Omezení: beta-verze – menší sada funkcí než desktopová varianta; omezené možnosti citlivostní analýzy

pyMCDM (Python knihovna) [18]

- Dostupné: pypi.org/project/pymcdm/

- Autoři: shekhand & kiziub (GitHub)
- Licence: MIT (open-source, PyPI)
- Metody: 40+ (MAUT, TOPSIS, VIKOR, ELECTRE III/IV, CODAS, ...), normalizační, vizualizační moduly a podpora řady metod stanovení vah
- Charakteristika: baterie metod sjednocená v jednom API; ideální pro vědecké experimenty v Jupyteru a integraci do vlastních systémů
- Omezení: vyžaduje znalost Pythonu; pro interaktivní GUI je nutné externí rozhraní

scikit-criteria (Python knihovna) [19]

- Dostupné: scikit-criteria.quatrope.org/
- Autor: Antonio Horta (Cybercrafter®)
- Licence: Apache 2.0
- Metody: TOPSIS, VIKOR, PROMETHEE-II, ELECTRE I/II, WSM/WPM aj., nástroje pro určování vah (AHP, entropie, ...)
- Charakteristika: syntaxí a duchem navazuje na scikit-learn; plná kompatibilita s pandas/NumPy
- Omezení: menší počet outrankingových metod v porovnání s pyMCDM; slabší grafická podpora

2.3 Shrnutí softwarových řešení pro MCDM a syntéza pro návrh vlastního nástroje

Visual PROMETHEE a Decision Lab 2025 zastupují moderní, uživatelsky přívětivé desktopové aplikace s důrazem na outrankingovou rodinu PROMETHEE. Visual PROMETHEE zaujme interaktivní GAIA rovinou, snadnou citlivostní analýzou a akademickou licencí; jeho limitem je ovšem single-platformní (Windows) charakter a uzavřenost vůči jiným metodám. Decision Lab 2025 tento nedostatek částečně kompenzuje cloudovou synchronizací a modulárními rozšířeními, avšak cenově míří spíše na podnikové zákazníky.

V segmentu hierarchických metod dominuje Expert Choice Desktop 11.5 a SuperDecisions v3. Expert Choice zůstává zlatým standardem pro AHP díky intuitivnímu stromovému GUI a what-if scénářům, ale nepodporuje jiné MCDM postupy. SuperDecisions rozšiřuje metodiku na Analytic Network Process, přidává skupinové hlasování, ovšem za cenu strmější křivky učení a stejně jako předchůdce vyžaduje systém Windows.

Stále významnější část uživatelů dává přednost nástrojům, jež fungují přímo v prohlížeči a umožňují týmovou práci v reálném čase. D-Sight Online kombinuje AHP, PROMETHEE a VIKOR v jednom rozhraní, generuje přehledné dashboardy a auditní reporty na jedno kliknutí. Vyžaduje však neustálé připojení ke cloudové službě a placené předplatné.

1000minds popularizuje metodu PAPRIKA díky anketnímu stylu dotazování, který je srozumitelný i laikům. Nástroj si našel cestu hlavně do zdravotnického výzkumu a policy-makingu. Silně se však opírá o jedinou metodu a u komplexních problémů může počet párových otázek rychle narůstat.

Jednoduchým webovým alternativám vévodí Web-HIPRE v2.0 a AHP-OS. Všechny jsou bezplatné, rychle dostupné a vhodné pro výuku, nicméně každá je metodicky úzce zaměřena a postrádá sofistikovanou vizualizaci.

Díky otevřenému kódu a komunitnímu vývoji nabízejí pyMCDM nebo scikit-criteria nejširší metodické portfolio. Knihovna pyMCDM aktuálně integruje přes čtyřicet algoritmů v jednotném API, scikit-criteria zrcadlí ekosystém scikit-learn. Hlavní výhodou je možnost přímé integrace do datových workflow (pandas, NumPy) a automatizace zpracování dat. Bariérou zůstává nutná znalost programování a absence grafického rozhraní – interaktivní dashboard si uživatel musí doprogramovat.

Knihovny doplňují dvě menší desktopové aplikace: PriEsT (open-source Java) a Visual PROMETHEE-Cloud beta. PriEsT dokáže vyhodnocovat konzistenci AHP matic a experimentovat s evolučními algoritmy, ale vývoj ustal v roce 2017. Cloudová varianta Visual PROMETHEE se naopak rychle rozvíjí, zatím však nenabízí plnou funkcionalitu desktopového předchůdce.

Z komparace vyplývají tři klíčové trendy:

- **Konvergence do cloudu** – kolaborativní SaaS řešení (D-Sight, Decision Lab 2025, 1000minds) potvrzují poptávku po sdílených scénářích a rychlé dostupnosti.

- **Rozšiřování open-source ekosystému** – knihovny pyMCDM a scikit-criteria posunuly hranici počtu implementovaných metod i jednoduchosti jejich volání. Chybí jim však uživatelské rozhraní, které by tyto možnosti zpřístupnilo širšímu publiku.
- **Metodická specializace** – řada nástrojů (SuperDecisions, 1000minds, AHP-OS) podporuje jedinou metodu a postrádá vzájemné porovnání výsledků různými algoritmy.

Tyto poznatky definují požadavky pro návrh nového řešení, které zahrnuje webovou dostupnost, otevřený modulární základ využívající robustní Python knihovny, interaktivní vizualizace a srovnání výsledků napříč metodami.

Dílčí závěr:

Rešerše dvanácti reprezentativních softwarových řešení ukazuje, že žádné z nich současně nesplňuje všechny požadavky na otevřenost, metodickou šíři. Tyto mezery definují prostor pro nové řešení: modulární, webově dostupný nástroj vybudovaný na otevřeném Python ekosystému, který kombinuje široké metodické portfolio s interaktivní. Těmto cílům se věnuje následující praktická část práce, jež popisuje návrh a implementaci vlastního MCA nástroje.

II PRAKTICKÁ ČÁST

3 ÚVOD DO PRAKTICKÉ ČÁSTI – SOFTWAREVÁ IMPLEMENTACE METOD MULTIKRITERIÁLNÍ ANALÝZY

Cílem praktické části diplomové práce bylo navrhnout a realizovat webovou aplikaci podporující vícekritériální rozhodování. Výsledkem je aplikace MCApp, která je určena zejména pro vzdělávací účely a umožňuje uživatelům intuitivní formou analyzovat rozhodovací problémy za použití více metod multikritériálního hodnocení.

Aplikace MCApp je navržena jako plně webové řešení, přístupné z libovolného moderního internetového prohlížeče bez nutnosti instalace. Uživatelské prostředí je koncipováno jako průvodce, který uživatele krok za krokem provádí definicí rozhodovacího problému, zadáním kritérií a variant, stanovením vah kritérií a následnou analýzou výsledků.

Proces práce v aplikaci je rozdělen do několika fází:

- Definice rozhodovacího problému a jeho základních parametrů,
- Zadání hodnotících kritérií a volba metody pro stanovení jejich vah (přímé zadání, AHP metoda, metoda entropie),
- Přidání jednotlivých variant včetně jejich popisu,
- Vyplnění hodnotící matice,
- Výběr metody multikritériálního hodnocení a zobrazení výsledků prostřednictvím pokročilých vizualizací,
- Export výsledků do formátů PDF a Excel.

Aplikace MCApp implementuje pět metod multikritériálního hodnocení: WSM, WPM, TOPSIS, ELECTRE a MABAC. Výsledky analýz jsou doplněny o interaktivní grafy (sloupcové grafy, radarové grafy, tepelné mapy, síťové grafy) a analýzu citlivosti, která umožňuje simulovat dopady změn vah kritérií.

Vývoj aplikace probíhal na platformě *Anvil* s využitím jazyka *Python 3.10*. Klientská část je realizována pomocí *Skulpt* (kompilace Pythonu do JavaScriptu) a využívá Material Design komponenty s responsivním rozvržením (CSS Grid, Flexbox).

Serverová část je založena na *Anvil Server* s datovým uložištěm *Anvil Data Tables* (NoSQL model) a komunikace mezi klientem a serverem probíhá přes interní API nad WebSockets. Data jsou uchovávána ve strukturovaném JSON formátu, což umožňuje jejich flexibilní rozšiřování.

Další technologie použité v aplikaci:

- **Vizualizace dat:** Plotly (pro generování interaktivních grafů),
- **Export dokumentů:** *anvil.pdf* (PDF) a *XlsxWriter* (Excel),
- **Bezpečnost:** autentizace pomocí JWT tokenů, oddělení dat mezi uživateli,
- **Zpracování chyb:** vlastní mechanismus validace a zachytávání výjimek.

Aplikace je hostována v cloudovém prostředí VPS na infrastruktuře Akamai (Linode), případně na AWS eu-west-2 a je kompatibilní se všemi hlavními prohlížeči (Chrome, Firefox, Edge, Safari).

Uživatelská správa je řešena pomocí dvou rolí:

- **Standardní uživatel** může vytvářet a spravovat vlastní analýzy,
- **Administrátor** má rozšířená oprávnění pro správu všech analýz a uživatelských účtů.

Díky modulární architektuře je aplikace připravena na budoucí rozšíření o nové metody hodnocení a další funkce.

Při vývoji aplikace byly využívány dostupné oficiální zdroje, dokumentace a podpůrné materiály vztahující se k použitým technologiím a knihovnám. Hlavním východiskem pro návrh architektury a implementaci uživatelského rozhraní byla oficiální *dokumentace platformy Anvil* [20], která poskytuje komplexní popis komponent, serverových funkcí, práce s databází i způsobu komunikace mezi klientem a serverem pomocí dekorátorů a WebSockets rozhraní.

Pro vizualizaci výsledků byla využita *knihovna Plotly*, přičemž informace o použití knihovny byly čerpány z oficiálních referenčních příkladů a online dokumentace dostupné na *plotly.com* [21], která poskytla přehled dostupných typů grafů a jejich parametrizace v Pythonu.

Export výsledků do formátů PDF a Excel byl realizován pomocí modulů *anvil.pdf* [22] a *XlsxWriter* [23], přičemž vývoj vycházel z příkladů dostupných ve vývojářské sekci platformy *Anvil* a oficiální dokumentace daných knihoven.

V oblasti zpracování dat a obsluhy chyb byla využita standardní knihovna jazyka Python, zejména moduly *functools*, *logging*, *datetime* a *typing*. Jejich využití vycházelo z dokumentace Pythonu dostupné na *docs.python.org* [24].

Struktura aplikace a volba návrhových vzorů (např. Singleton, Factory, Observer) vycházela z obecně uznávaných principů návrhu softwaru, Pro účely návrhu efektivního kódu byly využity poznatky z učebnice *JavaScript: the definitive guide : master the world's most-used programming language* [25], která je zaměřena na objektově orientované programování a návrhové vzory pro webové aplikace.

4 OBECNÝ POSTUP NÁVRHU APLIKACE

Návrh a realizace aplikace pro multikriteriální hodnocení variant probíhaly strukturovaně a byly založeny na standardních postupech softwarového inženýrství, které jsou popsány v knize *Python crash course: a hands-on, project-based introduction to programming* [26], a ze kterých tato práce čerpá. Níže uvedené kroky reflektují obecný proces od specifikace požadavků po konečné nasazení a ověření aplikace.

4.1 Specifikace požadavků

Specifikace požadavků byla stanovena na základě konzultací s cílovými uživateli a analýzy metod uvedených v teoretické části této práce. Hlavními požadavky na aplikaci byly:

- Podpora různých metod multikriteriálního hodnocení
- Možnost stanovení vah kritérií několika způsoby
- Administrace uživatelů a jejich analýz
- Export výsledků analýz ve formátech PDF a Excel

Tyto základní požadavky byly dále detailně rozpracovány pro účely návrhu a implementace aplikace, což přehledně shrnuje tabulka 21.

Rozšířená specifikace požadavků vychází z literární rešerše a analýzy existujících řešení, která ukázala, že dostupné aplikace obvykle postrádají integraci více metod nebo flexibilní správu uživatelů. Z této analýzy vyplynuly následující mezery, které se staly východiskem pro návrh nové aplikace:

- Absence intuitivního webového rozhraní dostupného bez instalace,
- nedostatečná integrace různých metod v jednom nástroji,
- omezené možnosti vizualizace a srovnání výsledků různých metod,
- nedostatečná podpora pro český jazyk a nevhodnost stávajících řešení pro výukové účely,
- vysoké licenční náklady u komerčních řešení.

Tyto nedostatky byly vzaty v úvahu při tvorbě detailní specifikace požadavků a následném návrhu řešení, které je zaměřeno na jejich odstranění a poskytnutí komplexní aplikace splňující potřeby cílových uživatelů.

Tabulka 21 Specifikace požadavků na aplikaci

Kategorie požadavku	Podkategorie	Specifikace požadavků
Funkční požadavky:	Implementace vybraných metod MCA	1. WSM 2. WPM 3. TOPSIS 4. ELECTRE 5. MABAC
	Implementace metod pro stanovení vah kritérií	1. Přímého zadání vah 2. Párové porovnání (metodika AHP) 3. Automatický výpočet (metoda entropie)
	Správa dat	Ukládání a načítání vytvořených analýz Správa uživatelských účtů a přístupových práv Podpora pro kopírování (klonování) existujících analýz
	Vizualizace výsledků	Grafické zobrazení výsledků analýz Srovnání výsledků různých metod Analýza citlivosti na změnu vah kritérií
	Export dat	Export výsledků do formátu PDF Export souhrnného výpočtu do formátu Excel
Nefunkční požadavky:	Použitelnost	Intuitivní uživatelské rozhraní Průvodce (wizard) pro asistenci uživatelům Responsivní design pro různá zařízení
	Výkon	Efektivní algoritmy pro výpočty Rychlá odezva i při větším počtu kritérií a variant
	Rozšiřitelnost	Modulární architektura pro snadné přidání nových metod Oddělení datové a prezentační vrstvy
	Bezpečnost	Autentizace a autorizace uživatelů Ochrana citlivých dat

4.2 Metodika návrhu a vývoje

Návrh a realizace aplikace probíhaly s důrazem na flexibilitu, možnost rychlého přizpůsobení požadavkům a postupné rozšiřování funkcionalit. Pro tyto účely byla zvolena kombinace iterativního a inkrementálního přístupu, která umožnila nejen efektivní rozvoj aplikace, ale i průběžné začleňování zpětné vazby.

Vývojový proces se inspiroval principy agilní metodiky Scrum⁷, přičemž tento proces byl upraven pro potřeby individuálního vývoje a specifík akademického projektu. Celý vývoj probíhal v několika fázích:

Inicializační fáze:

- Definování hlavních cílů projektu a stanovení klíčových funkčních a nefunkčních požadavků na základě analýzy (viz kapitola 2.1),
- návrh datového modelu a architektury aplikace s důrazem na modulární stavbu a oddělení datové a prezentační vrstvy.

Iterativní vývoj:

- Implementace základních funkcionalit po jednotlivých modulech, např. načítání dat, výpočty, vizualizace,
- pravidelná kontrola výsledků a zpětná vazba od konzultantů a testujících uživatelů.

Inkrementální rozšiřování:

- Postupné doplňování metod multikriteriálního hodnocení,
- rozšíření uživatelského rozhraní o nové interaktivní prvky,
- vylepšování výstupů – zejména vizualizací a možnostmi exportu výsledků.

Validace a testování:

- Důkladná kontrola matematické správnosti implementovaných algoritmů včetně porovnání s ručními výpočty,
- uživatelské testování s cílem ověřit srozumitelnost rozhraní a intuitivnost ovládání,

⁷ Scrum je agilní metodika řízení projektů, která využívá iterativní přístup k vývoji, krátké vývojové cykly (sprinty) s důrazem na průběžnou zpětnou vazbu.[9]

5 KONCEPTUÁLNÍ NÁVRH A REALIZACE APLIKACE

Kapitola popisuje konceptuální návrh aplikace a její datové struktury.

5.1 Architektura aplikace

Aplikace pro multikriteriální hodnocení je navržena jako webová aplikace založená na třívrstvé architektuře [27], která odděluje prezentační, aplikační a datovou vrstvu. Tento přístup poskytuje výhody v podobě škálovatelnosti, udržitelnosti a efektivního vývoje.

Prezentační vrstva zajišťuje interakci s uživatelem a je implementována pomocí technologií HTML5, CSS3 a JavaScript. Využívá responzivní design pro přizpůsobení různým zařízením a obrazovkám.

Hlavní komponenty:

- Uživatelské rozhraní – intuitivní, přehledné a snadno použitelné rozhraní s důrazem na přístupnost.
- Formuláře pro zadávání dat – průvodce pro strukturované zadávání kritérií, variant a jejich hodnocení.
- Vizualizační komponenty – grafy a tabulky pro prezentaci výsledků hodnocení.
- Navigační prvky – menu, tlačítka a další navigační prvky pro snadný pohyb v aplikaci.

Aplikační vrstva obsahuje business logiku aplikace a je implementována v jazyce Python. Tato vrstva zpracovává požadavky z prezentační vrstvy, provádí výpočty spojené s multikriteriálním hodnocením a komunikuje s datovou vrstvou.

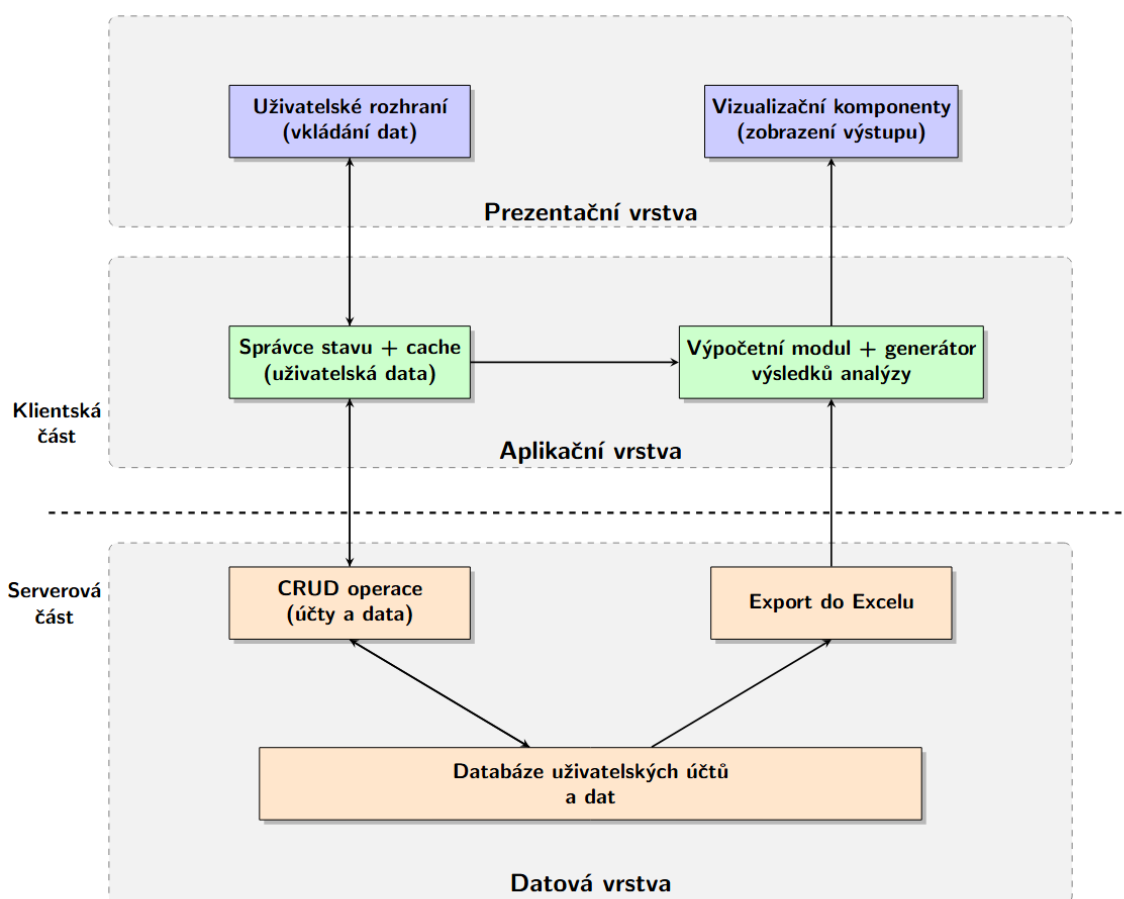
Hlavní komponenty:

- Výpočetní modul – zpracovává výpočty jednotlivých implementovaných metod MCA.
- Správce stavu – třída pro centralizovanou správu stavu aplikace.
- Servisní vrstva – soubor funkcí pro zpracování dat a komunikaci s datovou vrstvou.
- Validací komponenty – zajišťují kontrolu vstupních dat před jejich zpracováním.

Datová vrstva je zodpovědná za perzistenci dat a je realizována prostřednictvím serverové databáze. Pro efektivní ukládání dat analýz je využita kombinace tabulek a strukturovaných dat v JSON formátu⁸.

Hlavní komponenty:

- Relační databáze – tabulka pro uživatele a tabulka pro analýzy, která
- Správa uživatelů – funkce pro autentizaci a autorizaci účtů na principu přidělených rolí.
- CRUD (z angl. *Create / Read / Update / Delete*) operace – funkce pro práci s uživatelskými účty a jejich daty.
- Modul pro export dat do Excelu



Obrázek 1 Interakce mezi vrstvami

⁸ JSON (*JavaScript Object Notation*) je strukturovaný textový formát pro výměnu dat, který je srozumitelný pro lidi i počítače. Používá se velmi často pro přenos dat mezi klientem (např. webovou aplikací) a serverem, nebo pro ukládání konfigurací.

Komunikace mezi vrstvami je navržena tak, aby každá z nich řešila svou specializovanou odpovědnost co nejjednodušeji a zároveň si předávala jen ty informace, které opravdu potřebuje. Díky tomu zůstává aplikace bezpečná a snadno škálovatelná. Konkrétně:

Prezentační ↔ Aplikační vrstva (klient)

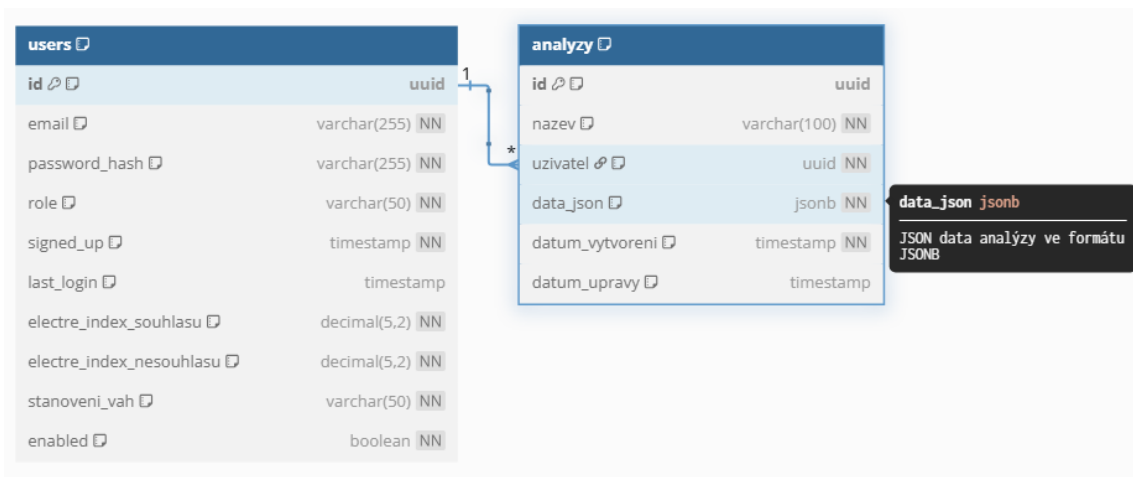
Uživatelské rozhraní (Formuláře, grafy, tabulky) zavolá metodu například z klientského modulu Správce stavu, nebo Výpočetní modul jako běžnou Python-funkci. Komponenty si předávají data, výsledky výpočtů i informace o stavu (cache), a to okamžitě v paměti prohlížeče. Díky tomu je UI extrémně responzivní – změny filtru nebo vstupní hodnoty vidí uživatel téměř okamžitě, bez čekání na síťové požadavky. Současně se minimalizuje přenosové zatížení, protože se opakovaně nenačítají celá data ze serverové databáze.

Aplikační ↔ Datová vrstva (server)

Když klient potřebuje trvale uložit nebo vyčíst data (např. přihlášení uživatele, uložení analýzy, generování Excelu), volá ji pomocí dekorované funkce v serverovém modulu. Komunikace probíhá přes zabezpečený HTTP/WebSocket kanál. Serverová vrstva chrání přístup k databázi, logiku CRUD operací před přímým přístupem z prohlížeče. To zajišťuje integritu dat a dodržení oprávnění uživatelů.

5.2 Datový model

Datový model aplikace využívá hybridní přístup kombinující relační databázový model s NoSQL prvky (JSON data).



Obrázek 2 Schéma databáze

Relační model je implementován prostřednictvím databázových tabulek (viz obr. 2) a pro ukládání dat spojených s analýzou je využita JSON struktura, která umožňuje podporu různých metod multikriteriálního hodnocení a snadné rozšiřování o další metody MCDM v budoucnosti.

data_json objekt:

```
{
  "nazev": "Název analýzy",
  "popis_analyzy": "Detailní popis analýzy",
  "kriteria": {
    "Kritérium 1": {
      "typ": "max",
      "vaha": 0.4
    },
    "Kritérium 2": {
      "typ": "min",
      "vaha": 0.6
    }
  },
  "varianty": {
    "Varianta A": {
      "popis_varianty": "Popis varianty A",
      "Kritérium 1": 10,
      "Kritérium 2": 5
    },
    "Varianta B": {
      "popis_varianty": "Popis varianty B",
      "Kritérium 1": 7,
      "Kritérium 2": 3
    }
  }
}
```

6 ALGORITMIZACE ANALYTICKÝCH VÝPOČTŮ VÍCEKRITERIÁLNÍHO HODNOCENÍ A STANOVENÍ VAH

Tato kapitola se zabývá algoritmizací metod vícekritériální analýzy implementovaných v aplikaci. Je zde vysvětlen algoritmus každé metody. Matematický aparát jednotlivých metod je detailně popsán v kapitole 1.3 teoretické části, zde se tedy zaměříme primárně na postup výpočtu a klíčové aspekty implementace.

6.1 Normalizace hodnot

Společným krokem pro většinu metod je normalizace hodnot kritériální matice, implementovaná funkcí *normalizuj_matici_minmax()*:

```
def normalizuj_matici_minmax(matice, typy_kriterii, varianty, kriteria):
    """Provede min-max normalizaci hodnot matice."""
    norm_matice = []
    for i in range(len(matice)):
        norm_radek = []
        for j in range(len(matice[0])):
            sloupec = [row[j] for row in matice]
            min_val = min(sloupec)
            max_val = max(sloupec)

            # Ošetření případu, kdy jsou všechny hodnoty stejné
            if max_val == min_val:
                norm_hodnota = 1.0
            else:
                # Pro MIN kritéria obrátíme normalizaci
                if typy_kriterii[j].lower() in ("min", "cost"):
                    norm_hodnota = (max_val - matice[i][j]) / (max_val - min_val)
                else:
                    norm_hodnota = (matice[i][j] - min_val) / (max_val - min_val)

            norm_radek.append(norm_hodnota)
        norm_matice.append(norm_radek)

    return {
        'nazvy_variant': varianty,
        'nazvy_kriterii': kriteria,
        'normalizovana_matice': norm_matice
    }
```

Tato funkce transformuje všechny hodnoty do intervalu $[0, 1]$, kde 1 představuje nejlepší hodnotu a 0 nejhorší, nezávisle na typu kritéria. Pro maximalizační kritéria je použit standardní min-max vzorec, zatímco pro minimalizační kritéria je vzorec obrácen, aby nejvyšší hodnota odpovídala nejlepšímu výsledku.

6.2 Algoritmus WSM

Algoritmus pracuje v těchto krocích:

1. **Normalizace hodnot** – Převedení všech hodnot na srovnatelnou škálu pomocí min-max normalizace.
2. **Výpočet vážených hodnot** – Vynásobení normalizovaných hodnot vahami příslušných kritérií.
3. **Výpočet celkového skóre** – Sečtení vážených hodnot pro každou variantu.
4. **Seřazení variant** – Varianty se seřadí sestupně podle celkového skóre.

Implementace WSM spočívá ve výpočtu vážených hodnot a jejich následném součtu:

```
# Výpočet vážených hodnot
vazene_hodnoty = vypocitej_vazene_hodnoty(norm_matice, vahy)

# Sečtení řádků (variant) pro získání skóre
skore = []
for i, vazene_radek in enumerate(vazene_hodnoty):
    skore.append((varianty[i], sum(vazene_radek)))

# Seřazení podle skóre sestupně
serazene = sorted(skore, key=lambda x: x[1], reverse=True)
```

Funkce `vypocitej_vazene_hodnoty()` jednoduše násobí každou normalizovanou hodnotu odpovídající vahou:

```
def vypocitej_vazene_hodnoty(matice, vahy):
    vazene_matice = []
    for radek in matice:
        vazene_radek = [hodnota * vahy[i] for i, hodnota in enumerate(radek)]
        vazene_matice.append(vazene_radek)
    return vazene_matice
```

Tím získáme pro každou variantu seznam vážených hodnot pro jednotlivá kritéria. Tyto hodnoty následně sečteme, čímž dostaneme celkové skóre varianty. Varianty poté seřadíme podle tohoto skóre sestupně, aby nejlepší varianta byla na prvním místě.

6.3 Algoritmus WPM

Algoritmus pracuje v těchto krocích:

1. **Úprava hodnot** – Zpracování hodnot pro minimalizační kritéria (převrácení hodnot).
2. **Výpočet produktů** – Pro každou variantu je vypočten produkt hodnot umocněných na příslušné váhy.

3. Seřazení variant – Varianty se seřadí sestupně podle výsledných hodnot produktů

Metoda WPM má klíčovou odlišnost od WSM – místo sčítání vážených hodnot používá jejich násobení po umocnění na příslušné váhy. Kritická část implementace:

```
# Inicializace výsledných hodnot na 1 pro násobení
skore = [1.0] * len(varianty)

for i in range(len(varianty)):
    for j in range(len(kriteria)):
        hodnota = matice[i][j]

        # Kontrola, že hodnoty nejsou nulové nebo záporné
        if hodnota <= 0:
            hodnota = 0.001 # Malá kladná hodnota

        # Pro minimalizační kritéria používáme 1/hodnota
        if typy_kriterii[j].lower() in ("min", "cost"):
            hodnota = 1 / hodnota

        # Umocníme hodnotu na váhu a vynásobíme dosavadní skóre
        skore[i] *= hodnota ** vahy[j]
```

Důležité aspekty implementace:

- Pro minimalizační kritéria používáme převrácenou hodnotu (1/hodnota)
- Ošetření nulových nebo záporných hodnot nahrazením malou kladnou konstantou
- Každá hodnota je umocněna na příslušnou váhu a vynásobena s dosavadním skóre

Tento přístup je robustnější vůči extrémním hodnotám než metoda WSM, protože pracuje s poměry hodnot místo jejich absolutních rozdílů.

6.4 Algoritmus TOPSIS

Algoritmus pracuje v těchto krocích:

1. **Vektorizace kritériální matice** – Normalizace hodnot pomocí Euklidovské normy.
2. **Výpočet vážené normalizované matice** – Vynásobení normalizovaných hodnot příslušnými vahami.
3. **Určení ideálního a anti-ideálního řešení** – Nalezení nejlepších a nejhorších hodnot pro každé kritérium.
4. **Výpočet vzdáleností od ideálního a anti-ideálního řešení** – Výpočet Euklidovské vzdálenosti každé varianty.
5. **Výpočet relativní blízkosti k ideálnímu řešení** – Pro každou variantu.

6. Seřazení variant – Podle klesající hodnoty relativní blízkosti.

Na rozdíl od WSM a WPM, TOPSIS používá Euklidovskou normalizaci namísto min-max normalizace:

```
# Vektorizace rozhodovací matice (normalizace pomocí Euklidovské normy)
norm_matice = []
for i in range(len(varianty)):
    radek = []
    for j in range(len(kriteria)):
        # Vypočítáme Euklidovskou normu pro každé kritérium
        sloupec = [matice[k][j] for k in range(len(varianty))]
        norma = (sum(x**2 for x in sloupec)) ** 0.5

        # Normalizace hodnoty
        hodnota = matice[i][j] / norma if norma != 0 else 0
        radek.append(hodnota)
    norm_matice.append(radek)
```

Dalším krokem TOPSIS je určení ideálního a anti-ideálního řešení:

```
# Určení ideálního a anti-ideálního řešení
ideal = []
anti_ideal = []
for j in range(len(kriteria)):
    sloupec = [vazena_matice[i][j] for i in range(len(varianty))]

    if typy_kriterii[j].lower() in ("max", "benefit"):
        ideal.append(max(sloupec))
        anti_ideal.append(min(sloupec))
    else: # min kritéria
        ideal.append(min(sloupec))
        anti_ideal.append(max(sloupec))
```

Následně se počítá vzdálenost každé varianty od ideálního a anti-ideálního řešení:

```
# Výpočet vzdáleností od ideálního a anti-ideálního řešení
for i in range(len(varianty)):
    sum_ideal = 0
    sum_anti_ideal = 0
    for j in range(len(kriteria)):
        sum_ideal += (vazena_matice[i][j] - ideal[j]) ** 2
        sum_anti_ideal += (vazena_matice[i][j] - anti_ideal[j]) ** 2
    dist_ideal.append(sum_ideal ** 0.5)
    dist_anti_ideal.append(sum_anti_ideal ** 0.5)
```

Finální výpočet relativní blízkosti k ideálnímu řešení:

```
# Výpočet relativní blízkosti k ideálnímu řešení
relativni_blizkost = []
for i in range(len(varianty)):
    if dist_ideal[i] + dist_anti_ideal[i] == 0:
        relativni_blizkost.append(0) # Ošetření dělení nulou
    else:
        relativni_blizkost.append(dist_anti_ideal[i] / (dist_ideal[i] +
        dist_anti_ideal[i]))
```

Čím blíže je hodnota k 1, tím lepší je varianta (blíží k ideálnímu a vzdálenější od anti-ideálního řešení).

6.5 Algoritmus ELECTRE

Algoritmus pracuje v těchto krocích:

1. **Normalizace hodnot** – Standardizace hodnot pro srovnatelnost kritérií.
2. **Výpočet matice souhlasu (concordance)** – Pro každou dvojici variant se počítá index souhlasu.
3. **Výpočet matice nesouhlasu (discordance)** – Pro každou dvojici variant se počítá index nesouhlasu.
4. **Výpočet matice převahy (outranking)** – Kombinace matic souhlasu a nesouhlasu pomocí prahových hodnot.
5. **Výpočet Net Flow** – Určení čisté převahy pro každou variantu.
6. **Seřazení variant** – Podle klesající hodnoty Net Flow.

Metoda ELECTRE je komplexnější než předchozí metody a vyžaduje výpočet několika matic. Klíčová část je výpočet matice souhlasu:

```
# Pro každou dvojici variant i a j
for i in range(pocet_variant):
    for j in range(pocet_variant):
        if i != j: # Vynecháme diagonálu
            # Souhlasná množina - ve kterých je var. i alespoň tak dobrá jako j
            concordance_sum = 0

            for k in range(len(vahy)):
                # Pokud je varianta i alespoň tak dobrá jako j v kritériu k
                if norm_matice[i][k] >= norm_matice[j][k]:
                    concordance_sum += vahy[k]

            concordance_matrix[i][j] = concordance_sum
```

a výpočet matice nesouhlasu:

```
# Pro každou dvojici variant i a j
for i in range(pocet_variant):
    for j in range(pocet_variant):
        if i != j: # Vynecháme diagonálu
            # Nesouhlasná množina - max. normalizovaný rozdíl ve prospěch j nad i
            max_diff = 0

            for k in range(len(norm_matice[0])):
                # Rozdíl mezi j a i v kritériu k (pokud j je lepší než i)
                diff = max(0, norm_matice[j][k] - norm_matice[i][k])
                max_diff = max(max_diff, diff)

            discordance_matrix[i][j] = max_diff
```

Z těchto matic se následně vytvoří matice převahy, která indikuje, zda varianta i převyšuje variantu j :

```
# Pro každou dvojici variant i a j
for i in range(pocet_variant):
    for j in range(pocet_variant):
        if i != j: # Vynecháme diagonálu
            # Varianta i převyšuje variantu j, pokud:
            # 1. Index souhlasu je větší nebo roven prahu
            # 2. Index nesouhlasu je menší nebo roven prahu
            if (concordance_matrix[i][j] >= index_souhlasu and
                discordance_matrix[i][j] <= index_nesouhlasu):
                outranking_matrix[i][j] = 1
```

Finální hodnocení variant je založeno na Net Flow:

```
for i in range(pocet_variant):
    # Počet variant, které i převyšuje
    vychazejici = sum(outranking_matrix[i])

    # Počet variant, které převyšují i
    prichazejici = sum(outranking_matrix[j][i] for j in range(pocet_variant))

    # Net Flow = vychazejici - prichazejici
    net_flow = vychazejici - prichazejici
    net_flows.append((varianty[i], net_flow))
```

6.6 Algoritmus MABAC

Algoritmus pracuje v těchto krocích:

1. **Normalizace hodnot** – Převedení všech hodnot na srovnatelnou škálu pomocí min-max normalizace.
2. **Výpočet vážené normalizované matice** – Aplikace vah a specifická transformace pro MABAC: $v_{ij} = w_j \cdot (r_{ij} + 1)$.
3. **Výpočet hraničních hodnot** – Určení hraniční aproximační oblasti pro každé kritérium pomocí geometrického průměru.
4. **Výpočet vzdáleností od hraničních hodnot** – Určení, jak daleko jsou varianty od hraniční oblasti.
5. **Výpočet celkového skóre** – Sečtení těchto vzdáleností pro každou variantu.
6. **Seřazení variant** – Varianty se seřadí sestupně podle celkového skóre.

Specifický krok metody MABAC je výpočet hraničních hodnot pomocí geometrického průměru:

```
# Výpočet hraničních hodnot pro každé kritérium (G)
g_values = []
for j in range(len(kriteria)):
    # Výpočet hraničního aproximačního prostoru pro každé kritérium
    g = 1
    for i in range(len(varianty)):
        g *= vazena_matice[i][j]
    g = g ** (1/len(varianty)) # Geometrický průměr
    g_values.append(g)
```

Následně se počítá vzdálenost každé varianty od hraniční oblasti:

```
# Výpočet vzdáleností od hraničního aproximačního prostoru (Q)
q_matrix = []
for i in range(len(varianty)):
    q_row = []
    for j in range(len(kriteria)):
        q = vazena_matice[i][j] - g_values[j]
        q_row.append(q)
    q_matrix.append(q_row)
```

Celkové hodnocení varianty je pak součtem těchto vzdáleností:

```
# Výpočet celkového hodnocení pro každou variantu
skore = []
for i, varianta in enumerate(varianty):
    celkove_skore = sum(q_matrix[i])
    skore.append((varianta, celkove_skore))
```

Kladné hodnoty vzdálenosti indikují, že varianta je nad hraniční oblastí (lepší), zatímco záporné hodnoty ukazují, že je pod hraniční oblastí (horší).

6.7 Algoritmus AHP pro stanovení vah

Algoritmus pracuje v těchto krocích:

- **Sestavení matice párového porovnání** – Uživatelé porovnávají každou dvojici kritérií a určují jejich relativní důležitost.
- **Výpočet vah kritérií** – Pomocí metody geometrického průměru.
- **Kontrola konzistence** – Ověření, zda jsou párová porovnání vnitřně konzistentní.

Hlavní kroky implementace AHP:

```

# 1. Vytvoření matice porovnání z uživatelských vstupů
matice = []
for i in range(pocet_kriterii):
    radek = []
    for j in range(pocet_kriterii):
        if i == j:
            # Diagonála matice
            radek.append(1)
        elif i < j:
            # Hodnota z párového porovnání
            hodnota = self.ahp_hodnoty.get((nazvy_krit[i], nazvy_krit[j]), 1)
            radek.append(hodnota)
        else: # i > j
            # Reciproční hodnota
            hodnota = self.ahp_hodnoty.get((nazvy_krit[j], nazvy_krit[i]), 1)
            radek.append(1/hodnota)
    matice.append(radek)

# 2. Výpočet geometrických průměrů řádků
geom_prumery = []
for radek in matice:
    soucin = 1
    for hodnota in radek:
        soucin *= hodnota
    geom_prumer = soucin ** (1/pocet_kriterii)
    geom_prumery.append(geom_prumer)

# 3. Normalizace na součet 1
soucet_prumeru = sum(geom_prumery)
normalizovane_vahy = [prumer/soucet_prumeru for prumer in geom_prumery]

```

Pro ověření se vypočítává index konzistence (CI) a poměr konzistence (CR):

```

# 1. Vytvoření matice porovnání z uživatelských vstupů
matice = []
for i in range(pocet_kriterii):
    radek = []
    for j in range(pocet_kriterii):
        if i == j:
            # Diagonála matice
            radek.append(1)
        elif i < j:
            # Hodnota z párového porovnání
            hodnota = self.ahp_hodnoty.get((nazvy_krit[i], nazvy_krit[j]), 1)
            radek.append(hodnota)
        else: # i > j
            # Reciproká hodnota
            hodnota = self.ahp_hodnoty.get((nazvy_krit[j], nazvy_krit[i]), 1)
            radek.append(1/hodnota)
    matice.append(radek)

# 2. Výpočet geometrických průměrů řádků
geom_prumery = []
for radek in matice:
    soucin = 1
    for hodnota in radek:
        soucin *= hodnota
    geom_prumer = soucin ** (1/pocet_kriterii)
    geom_prumery.append(geom_prumer)

# 3. Normalizace na součet 1
soucet_prumeru = sum(geom_prumery)
normalizovane_vahy = [prumer/soucet_prumeru for prumer in geom_prumery]

```

Hodnota $CR \leq 0.1$ indikuje dostatečnou konzistenci hodnocení.

6.8 Algoritmus metody entropie

Algoritmus pracuje v těchto krocích:

- **Normalizace hodnot** – Transformace původních hodnot na podíly.
- **Výpočet entropie kritérií** – Výpočet míry neurčitosti pro každé kritérium.
- **Výpočet míry diverzity** – Výpočet doplňku entropie.
- **Normalizace vah** – Stanovení konečných vah kritérií.

Hlavní kroky implementace metody entropie:

```
# 1. Normalizace matice pro každé kritérium (sloupec)
norm_matice = []
for radek in matice:
    norm_radek = []
    for j in range(len(kriteria)):
        sloupec = [matice[i][j] for i in range(len(varianty))]
        soucet_sloupec = sum(sloupec)
        if soucet_sloupec <= 0:
            norm_hodnota = 1.0 / len(varianty) # Rovnoměrné rozdělení při nulách
        else:
            norm_hodnota = radek[j] / soucet_sloupec
        norm_radek.append(norm_hodnota)
    norm_matice.append(norm_radek)

# 2. Výpočet entropie pro každé kritérium
entropie = []
k = 1 / math.log(len(varianty)) # Konstanta pro normalizaci entropie

for j in range(len(kriteria)):
    e_j = 0
    for i in range(len(varianty)):
        # Vynecháme nulové hodnoty při výpočtu entropie (log(0) je nedefinováno)
        if norm_matice[i][j] > 0:
            e_j -= norm_matice[i][j] * math.log(norm_matice[i][j])
    e_j *= k
    entropie.append(e_j)

# 3. Výpočet míry diverzity (1 - entropie)
diverzita = [1 - e for e in entropie]

# 4. Normalizace diverzity na váhy (suma vah = 1)
suma_diverzity = sum(diverzita)
if suma_diverzity <= 0:
    # Pokud je suma nulová, použijeme rovnoměrné váhy
    vahy = {krit: 1.0 / len(kriteria) for krit in kriteria.keys()}
else:
    vahy = {}
    for i, krit in enumerate(kriteria.keys()):
        vahy[krit] = diverzita[i] / suma_diverzity
```

Metoda přiřazuje větší váhy kritériím, která vykazují větší variabilitu v hodnotách (tedy nesou více informace).

7 OVĚŘENÍ FUNKČNOSTI APLIKACE

Cílem této kapitoly je prakticky demonstrovat správnou funkčnost a uživatelskou přívětivost vyvinuté webové aplikace MCAApp při řešení reálného rozhodovacího problému z oblasti bezpečnosti.

Jako modelový příklad byla zvolena úloha výběru domácího antivirového řešení, neboť představuje typickou bezpečnostní potřebu koncových uživatelů, k hodnoceným produktům existují veřejně dohledatelné a kvantifikované údaje a kritéria zahrnují jak technické, tak ekonomické aspekty, čímž umožňují plnohodnotné využití více metod multikriteriální analýzy.

7.1 Případová studie – zadání

V rámci analýzy je posuzováno osm komerčně dostupných bezpečnostních balíčků (Bitdefender Total Security, Kaspersky Plus, ESET Internet Security, Microsoft Defender, Avast Premium Security, Norton 360 Deluxe, Trend Micro Maximum Security a McAfee Total Protection). Každá varianta je hodnocena podle pěti kritérií:

Tabulka 22 Případová studie – zvolená kritéria

Kritérium	Typ	Stručná charakteristika	Zdroj dat
Protection_score	max	Úroveň detekce a blokování škodlivého kódu (0–6 bodů)	AV-TEST GmbH, test Windows 10 Home 02/2025
Performance_score	max	Dopad na systémové prostředky (0–6 bodů)	AV-TEST GmbH, test Windows 10 Home 02/2025
Usability_score	max	Míra falešných poplachů a uživatelská vstřícnost (0–6 bodů)	AV-TEST GmbH, test Windows 10 Home 02/2025
Annual_cost_CZK	min	Cena roční elektronické licence v Kč	Oficiální e-shop výrobce nebo Alza.cz (květen 2025)
Feature_count	max	Počet významných doplňkových funkcí (VPN, správce hesel, rodičovská kontrola apod.)	Produktové listy výrobců

(Data čerpána z [28])

Hodnocené skóre 0–6 je převzato z aktuálního laboratorního testu AV-TEST (Windows 10 Home, únor 2025) [28]; cenové údaje byly zjištěny k 2. květnu 2025, přepočet cizoměnových částek byl proveden kurzem 1 USD / EUR = 25 Kč. Při stanovení počtu

doplňkových funkcí se vycházelo z oficiálních produktových přehledů jednotlivých výrobců.

Pro ověření funkcionalit aplikace byly v MCAApp provedeny tyto kroky:

1. **Import rozhodovací matice** ve formátu JSON a kontrola správnosti dat.
2. **Stanovení vah kritérií** bylo provedeno expertním odhadem
 - a. Protection_score: 0,30
 - b. Performance_score: 0,25
 - c. Usability_score: 0,15
 - d. Annual_Cost_CZK: 0,20
 - e. Feature_count: 0,10
3. **Výpočet výsledků** pomocí pěti implementovaných metod – WSM, WPM, TOPSIS a ELECTRE I, MABAC –a následné srovnání pořadí variant.
4. **Export výstupů** do Excelu

7.2 Případová studie – vstupní data

Vstupní data pro vícekritériální analýzu tvoří rozhodovací matice a výše uvedené váhy jednotlivých kritérií.

Tabulka 23 Případová studie – rozhodovací matice

Varianta / Kritérium,	Feature count	Annual cost CZK	Usability score	Protection score	Performance score
Kaspersky Plus	4,0	890	6,0	6,0	5,5
Microsoft Defender	2,0	0	6,0	5,5	3,5
Norton 360 Deluxe	6,0	1159	5,5	6,0	4,0
Avast Premium Security	5,0	729	6,0	6,0	4,5
ESET Internet Security	3,0	1438	6,0	6,0	5,0
McAfee Total Protection	4,0	1199	6,0	6,0	6,0
Bitdefender Total Security	5,0	424	6,0	6,0	4,5
Trend Micro Maximum Security	4,0	279	6,0	5,5	4,0

(Data čerpána z [28])

Takto zvolená případová studie umožňuje prověřit veškeré klíčové funkce aplikace – od zadávání heterogenních dat, přes výběr a kombinaci metod MCDA, až po prezentaci a export výsledků.

7.3 Případová studie – hodnocení variant

Níže uvedená tabulka uvádí srovnání jednotlivých metod:

Tabulka 24 Případová studie – srovnání metod

Varianta / metoda (pořadí a skóre)	WSM	WPM	TOPSIS	ELECTRE	MABAC
Kaspersky Plus	2 (0,776)	5 (1,013)	5 (0,454)	2 (2)	2 (0,206)
Microsoft Defender	8 (0,350)	1 (12,73)	2 (0,661)	6 až 7 (-2)	8 (-0,220)
Norton 360 Deluxe	6 (0,489)	7 (0,912)	7 (0,297)	6 až 7 (-2)	6 (-0,082)
Avast Premium Security	4 (0,724)	4 (1,025)	4 (0,501)	3 až 4 (1)	4 (0,153)
ESET Internet Security	5 (0,625)	8 (0,873)	8 (0,209)	8 (-4)	5 (0,055)
McAfee Total Protection	1 (0,783)	6 (0,975)	6 (0,361)	3 až 4 (1)	1 (0,213)
Bitdefender Total Security	3 (0,766)	3 (1,143)	3 (0,660)	1 (4)	3 (0,196)
Trend Micro Maximum Security	7 (0,411)	2 (1,149)	1 (0,663)	5 (0)	7 (-0,159)

Tabulka 25 Případová studie – konsolidované pořadí

Celkové pořadí	Varianta	Průměrné pořadí
1.	Bitdefender Total Security	2,6
2.	Kaspersky Plus	3,2
3.	McAfee Total Protection	3,6
4.	Avast Premium Security	3,8
5.	Trend Micro Maximum Security	4,4
6.	Microsoft Defender	5,0
7.	Norton 360 Deluxe	6,6
8.	ESET Internet Security	6,8

7.4 Případová studie – interpretace výsledků a shrnutí analýzy

Interpretace a doporučení v hlavních bodech

- Bitdefender Total Security dosahuje stabilně vysokých výsledků u všech metod a zároveň má velmi příznivou cenu (424 Kč) a bohatou funkční výbavu → doporučená volba pro domácí uživatele požadující komplexní ochranu.
- Kaspersky Plus je těsně druhý – ochrana a výkon na maximum, cena pod 900 Kč; vhodný tam, kde uživatelům nevadí menší počet doplňků.
- McAfee Total Protection benefituje z nejvyššího výkonu a plné sady funkcí, ale vyšší cena (1199 Kč) a nižší relativní váha funkcí jej posouvají na třetí místo.
- Trend Micro MS a Microsoft Defender profitují z velmi nízkých nákladů, proto je některé metody preferují; celkově jim však škodí slabší výkon či ochrana.
- ESET a Norton doplácí hlavně na vyšší cenu (ESET) nebo horší výkon (Norton).

Metody s důrazem na cenu (WPM, TOPSIS) favorizují bezplatného Microsoft Defender nebo levný Trend Micro. Pokud by se váha nákladů snížila pod 10 %, všechny metody by se shodly na Bitdefenderu jako jasném vítězi, což potvrzuje robustnost tohoto řešení.

Shrnutí analýzy

Na základě multikriteriálního vyhodnocení osmi předních domácích bezpečnostních balíčků s laboratorními skóre AV-TEST 02/2025 je Bitdefender Total Security nejvyváženější volbou, která kombinuje špičkovou ochranu a výkon, rozumnou cenu a široké spektrum funkcí.

ZÁVĚR

Diplomová práce naplnila svůj primární cíl – navrhnout a realizovat webovou aplikaci MCAApp, která uživatelům umožňuje aplikovat pět reprezentativních metod multikriteriální analýzy (WSM, WPM, TOPSIS, ELECTRE a MABAC) na rozhodovací. Kombinací teoretické studie, podrobné rešerše existujících softwarových řešení a systematického softwarového inženýrství vznikl modulární nástroj, jenž překračuje limity většiny dostupných aplikací:

- **metodická šíře** – simultánní implementace kompenzačních, outrankingových i hraničních přístupů dovoluje triangulovat výsledky a odhalit citlivost pořadí variant;
- **uživatelská přívětivost** – průvodce provádí nezkušeného uživatele všemi kroky MCDM postupu, přičemž interaktivní vizualizace zajišťují okamžitou interpretační zpětnou vazbu;
- **webová dostupnost** – absence instalačních požadavků a
- responsivní rozhraní maximalizují přístupnost pro vzdělávací účely;
- **didaktická hodnota** – možnost paralelně spouštět různé algoritmy a sledovat divergenci výsledků napomáhá hlubšímu pochopení principů vícekritériálního rozhodování.

Empirická případová studie výběru domácího antivirového balíku potvrdila nejen matematickou správnost implementovaných algoritmů, ale rovněž schopnost aplikace generovat prakticky využitelné výstupy v reálném bezpečnostním scénáři.

Z hlediska softwarového inženýrství aplikace splnila všechny definované funkční i nefunkční požadavky:

- **architektura třívrstvé webové aplikace** vykazuje dobré separační vlastnosti, umožňuje horizontální škálování a usnadňuje budoucí rozšiřování;
- **bezpečnostní opatření** (JWT autentizace, granulární role, šifrovaný přenos) reflektují standardy OWASP a posilují důvěryhodnost nástroje v kontextu analytiky citlivých dat;
- **exportní mechanismy** (PDF, XLSX) zaručují interoperabilitu s běžnými nástroji řízení a reportingu.

Vyvinutá aplikace MCAApp je komplexní a otevřeně rozšiřitelná platforma pro multikriteriální analýzu se zaměřením na pedagogické využití. Spojuje pevné teoretické základy MCDM s moderními webovými technologiemi a poskytuje vysoce funkční nástroj, jenž splňuje profesní standardy, očekávání uživatelů i přísná bezpečnostní kritéria. Práce tímto posiluje softwarovou podporu rozhodovacích procesů.

SEZNAM POUŽITÉ LITERATURY

- [1] KÖKSALAN, Murat; WALLENIOUS, Jyrki a ZIONTS, Stanley, 2011. *Multiple criteria decision making: from early history to the 21st century*. Singapore: World Scientific Publishing. ISBN 978-981-4335-58-4.
- [2] SAATY, Thomas L., 2008. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*. March 18, 2008, vol. 2008, no. 1, s. 83-98. ISSN 1753-1454.
- [3] KULKARNI, Anand J., 2022. *Multiple Criteria Decision Making: Techniques, Analysis and Applications*. Singapore: Springer. ISBN 978-981-16-7414-3.
- [4] FISHBURN, Peter C., Allan H. MURPHY a Herbert H. ISAACS, 1968. Sensitivity of Decisions to Probability Estimation Errors: A Reexamination. *Operations Research*. Č. 16(2), s. 254–267. ISSN 0030-364X.
- [5] HWANG, Ching-Lai a Kwangsun YOON, 1981. Methods for multiple attribute decision making. In: *Multiple attribute decision making*. Berlin, Heidelberg: Springer, s. 58–191. ISBN 978-3-540-10558-9.
- [6] TAHERDOOST, Hamed a Maryam MADANCHIAN, 2023. A Comprehensive Overview of the ELECTRE Method in Multi Criteria Decision-Making. *Journal of Management Science & Engineering Research*. Roč. 6, č. 2, s. 5–16. Dostupné z: <https://doi.org/10.30564/jmser.v6i2.5637>. [cit. 2025-02-17].
- [7] PAMUČAR, Dragan a Goran ČIROVIĆ, 2015. The selection of transport and handling resources in logistics centers using Multi-Attributive Border Approximation area Comparison (MABAC). *Expert Systems with Applications*. Roč. 42, č. 6, s. 3016–3028. ISSN 0957-4174. Dostupné z: <https://doi.org/10.1016/j.eswa.2014.11.057>. [cit. 2025-02-20].
- [8] EXPERT CHOICE INC. *Expert Choice Solutions* [online]. Arlington (VA): Expert Choice Inc., [s. a.] Dostupné z: <https://www.expertchoice.com/>. [cit. 2025-05-04].
- [9] CREATIVE DECISIONS FOUNDATION. *SuperDecisions* [online]. Pittsburgh: CDF, 2025. Dostupné z: <https://www.superdecisions.com/>. [cit. 2025-05-04].

- [10] MARESCHAL, Bertrand. *Visual PROMETHEE – Academic Edition* [online]. Brussels: Université Libre de Bruxelles, 2024. Dostupné z: <https://bertrand.mareschal.web.ulb.be/promethee.html>. [cit. 2025-05-04].
- [11] DECISION LAB AB. *Decision Lab 2025* [online]. Stockholm: Decision Lab AB, 2025. Dostupné z: <https://www.decisionlab.se/>. [cit. 2025-05-04].
- [12] D-SIGHT SA. *D-Sight Online* [online]. Brussels: D-Sight SA, 2025. Dostupné z: <https://www.d-sight.com/>. [cit. 2025-05-04].
- [13] HÄMÄLÄINEN, R. P. et al. *Web-HIPRE (v2.0)* [online]. Espoo: Aalto University, 2023. Dostupné z: <https://hipre.aalto.fi/>. [cit. 2025-05-04].
- [14] SIRAJ, Sajid; MIKHAILOV, Leonid; KEANE, James A. PriEsT: an interactive decision support tool to estimate priorities from pairwise comparison judgments. *International Transactions in Operational Research* [online]. 2015, vol. 22, no. 2, pp. 217–235. ISSN 1475-3995. Dostupné z: <https://doi.org/10.1111/itor.12054>. [cit. 2025-05-04].
- [15] 1000MINDS LTD. *1000minds – Decision-making & Conjoint Analysis Software* [online]. Dunedin: 1000minds Ltd., 2025. Dostupné z: <https://www.1000minds.com/>. [cit. 2025-05-04].
- [16] GOEPEL, Klaus D. Implementation of an Online Software Tool for the Analytic Hierarchy Process (AHP-OS). *International Journal of the Analytic Hierarchy Process* [online]. 2018, vol. 10, no. 3, pp. 469–487. ISSN 2151-8181. Dostupné z: <https://doi.org/10.13033/ijahp.v10i3.590>. [cit. 2025-05-04].
- [17] POHL, Erik; GELDERMANN, Jutta. *PROMETHEE-Cloud* [online]. Duisburg: Universität Duisburg-Essen, 2024. Dostupné z: <https://promethee.pom.uni-due.de/>. [cit. 2025-05-04].
- [18] KIZIELEWICZ, Bartłomiej; SHEKHOVTSOV, Andrii; SAŁABUN, Wojciech. pymcdm — The universal library for solving multi-criteria decision-making problems. *SoftwareX* [online]. 2023, vol. 22, art. 101368. ISSN 2352-7110. Dostupné z: <https://doi.org/10.1016/j.softx.2023.101368>. [cit. 2025-05-04].
- [19] CABRAL, Juan B.; LUCZYW0, Nadia Ayelen; ZANAZZI, José Luis. Scikit-Criteria: Colección de Métodos de Análisis Multi-Criterio Integrado al Stack Científico de Python. In: *XLV Jornadas Argentinas de Informática e Investigación Operativa*

- (45 JAIIO) – *XIV Simposio Argentino de Investigación Operativa (SIO)*. Buenos Aires, 2016, s. 59–66 [online]. Dostupné z: <http://45jaiio.sadio.org.ar/sites/default/files/Sio-23.pdf>. [cit. 2025-05-04].
- [20] MATTHES, Eric. *Python crash course: a hands-on, project-based introduction to programming*. 2nd edition. San Francisco: No Starch Press, [2019]. ISBN 9781593279295.
- [21] ANVIL. *Anvil Documentation* [online]. [s.l.]: Anvil Works, [neuvedeno]. Dostupné z: <https://anvil.works/docs>. [cit. 2025-04-29]
- [22] PLOTLY TECHNOLOGIES INC. *Plotly Python Graphing Library* [online]. [s.l.]: Plotly Technologies Inc., [neuvedeno]. Dostupné z: <https://plotly.com/python/>. [cit. 2025-04-29].
- [23] XLSXWRITER DEVELOPERS. *XlsxWriter – A Python module for creating Excel XLSX files* [online]. [s.l.]: XlsxWriter Developers, [neuvedeno]. Dostupné z: <https://xlsxwriter.readthedocs.io/>. [cit. 2025-04-29].
- [24] ANVIL. *Generating PDFs in Anvil apps (anvil.pdf)* [online]. [s.l.]: Anvil Works, [neuvedeno]. Dostupné z: <https://anvil.works/docs/media/pdf>. [cit. 2025-04-29].
- [25] FLANAGAN, David. *JavaScript: the definitive guide : master the world's most-used programming language*. Seventh edition. Beijing: O'Reilly, 2020. ISBN 978-1-491-95202-3.
- [26] PYTHON SOFTWARE FOUNDATION. *The Python Standard Library: documentation for Python 3* [online]. Wilmington: Python Software Foundation, 2024. Dostupné z: <https://docs.python.org/3/library/>. [cit. 2025-04-29].
- [27] MICROSOFT. *Three-tier architecture style* [online]. [Redmond]: Microsoft Corporation, [neuvedeno]. Dostupné z: <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier>. [cit. 2025-04-29].
- [28] AV-TEST. *Windows 10 February 2025 – The best Windows antivirus software for home users* [online]. Magdeburg: AV-TEST GmbH, 2025. Dostupné z: <https://www.av-test.org/home-windows/>. [cit. 2025-05-02].

SEZNAM OBRÁZKŮ

Obrázek 1 Interakce mezi vrstvami	57
Obrázek 2 Schéma databáze	58

SEZNAM TABULEK

Tabulka 1 Příklad 1 – rozhodovací matice	24
Tabulka 2 Příklad 1 – lineárně min-max normalizovaná matice	24
Tabulka 3 Příklad 2 – rozhodovací matice	26
Tabulka 4 Příklad 2 – matice se standardizovanými kritérii	26
Tabulka 5 Příklad 2 – výsledné pořadí variant	27
Tabulka 6 Příklad 3 – hodnoty kritérií před normalizací	29
Tabulka 7 Příklad 3 – normalizované hodnoty	29
Tabulka 8 Příklad 3 - vážené normalizované hodnoty	30
Tabulka 9 Porovnání variant metody ELECTRE	31
Tabulka 10 Příklad 4 - hodnoty kritérií před normalizací	33
Tabulka 11 Příklad 4 – poměrově normalizované hodnoty	33
Tabulka 12 Příklad 4 – matice indexu souhlasu	34
Tabulka 13 Příklad 4 – matice indexu nesouhlasu	34
Tabulka 14 Příklad 5 - hodnoty kritérií před normalizací	36
Tabulka 15 Příklad 5 – lineárně Min-max normalizovaná matice.....	36
Tabulka 16 Příklad 5 - vážené normalizované hodnoty	37
Tabulka 17 Příklad 5 matice vzdáleností od hraniční aproximační oblasti	37
Tabulka 18 Příklad 5 pořadí variant	37
Tabulka 19 Souhrn vybraných metod MCDM	38
Tabulka 20 Porovnání využitelnosti metod MCDM v bezpečnostních aplikacích	39
Tabulka 21 Specifikace požadavků na aplikaci	54
Tabulka 22 Případová studie – zvolená kritéria.....	69
Tabulka 23 Případová studie – rozhodovací matice	70
Tabulka 24 Případová studie – srovnání metod.....	71
Tabulka 25 Případová studie – konsolidované pořadí	71

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AHP	Analytický hierarchický proces
ANP	Analytický síťový proces
API	Programové rozhraní aplikace
AV	Antivir
AVTEST	Laboratoř AV-TEST pro testování antivirů
AWS	Amazon Web Services
CSS	Kaskádové styly
ELECTRE	Eliminace a volba vyjadřující realitu
EMO	Evoluční víceúčelová optimalizace
GAIA	Geometrická analýza pro interaktivní podporu
GDF	Algoritmus Geoffrion-Dyer-Feinberg
HTML5	Hypertextový značkovací jazyk verze 5
IS	Informační systém
JSON	JavaScript Object Notation (formát zápisu)
JWT	Webový token ve formátu JSON
MAUT	Teorie multiatributní užitečnosti
MCA	Multikriteriální analýza
MCDM	Multikriteriální rozhodování
MADM	Multiatribuční rozhodování
MABAC	Porovnání hraniční aproximační oblasti s více atributy
MOGA	Víceúčelový genetický algoritmus
NSGA	Genetický algoritmus s tříděním nedominovaných řešení
OWASP	Open web projekt pro bezpečnost webových aplikací
PDF	Dokumentový formát
PGA	Pareto genetický algoritmus

PROMETHEE	Metoda organizace pořadí preferencí
SaaS	Software jako služba
TOPSIS	Metoda pořadí preferencí podle podobnosti s ideálním řešením
VEGA	Vektorově vyhodnocovaný genetický algoritmus
VPS	Virtuální privátní server
WPM	Vážený součinnový model
WSM	Vážený součtový model
XLSX	Microsoft Excel Open XML Spreadsheet (formát sešitu)
XML	eXtensible Markup Language – značkovací jazyk

SEZNAM PŘÍLOH

Příloha P I: Dostupnost zdrojového kódu a online přístup k aplikaci MCAApp

PŘÍLOHA P I: DOSTUPNOST ZDROJOVÉHO KÓDU A ONLINE PŘÍSTUP K APLIKACI MCAPP

V souladu s požadavky na transparentnost, opakovatelnost výsledků a podporu dalšího rozvoje aplikace je k diplomové práci přiložen kompletní zdrojový kód softwarového řešení MCApp ve formě archivovaného ZIP souboru, který je součástí elektronického odevzdání této práce.

Alternativně je zdrojový kód veřejně přístupný také prostřednictvím platformy GitHub na adrese:

 <https://github.com/langerlad/mcaplikace>

Tento veřejný repositář obsahuje:

- kompletní kódovou základnu aplikace,
- instrukce pro lokální spuštění (včetně požadavků na prostředí),
- dokumentaci architektury a datového modelu.

Pro účely hodnocení diplomové práce je navíc k dispozici i online nasazená verze aplikace MCApp, hostovaná v prostředí Anvil:

 <https://mca-utb.anvil.app/>

K testování funkcionalit aplikace lze využít následující demonstrační účet:

- **E-mail:** demo@ucet
- **Heslo:** demo_demo

Aplikace rovněž umožňuje volnou registraci nových uživatelských účtů pro individuální práci.

Tento hosting je dočasný a primárně určený k účelům oponentury a obhajoby diplomové práce. Jeho garantovaná dostupnost je zajištěna do **31. října 2025**. Následně bude aplikace migrována na fakultní infrastrukturu nebo jiné fakultou určené úložiště pro dlouhodobou archivaci a využití ve výuce.

V případě potřeby lze prostřednictvím kontaktních údajů autora získat podporu při instalaci, konfiguraci nebo rozšíření aplikace MCApp.