

Jihočeská univerzita v Českých Budějovicích
Ekonomická fakulta
Katedra aplikované matematiky a informatiky

Teze diplomové práce

Užitková funkce v rozhodovacím procesu

Vypracoval: Bc. Patrik Maryška

Vedoucí práce: doc. RNDr. Jana Klicnarová, Ph.D

České Budějovice

Rok 2022/2023

Klíčová slova: Aplikace pro konstrukci užitkové funkce peněz, Teorie užitku, Teorie rozhodování, Python

Anotace:

Tato diplomová práce se zabývá návrhem a implementací aplikace s odpovídajícími algoritmy a mechanismy pro konstrukci užitkové funkce peněz pro rozhodovatele s averzí k riziku. Práce zahrnuje shrnutí teoretických témat z Teorie rozhodování, Teorie užitku a zabývá se procesem konstrukce užitkové funkce peněz. Aplikace náhodně generuje otázky rovnoměrně do pěti intervalů. Jakmile je proces generování otázek dokončen, algoritmus pro hledání optimálních startovacích hodnot je aplikován. Tento algoritmus postupně porovnává SSE ve vypočítaném intervalu. Je zde řešen problém nelineární regrese a je používána Metoda nejmenších čtverců. Graf nalezené užitkové funkce je vykreslen a jsou zobrazeny i odpovídající nalezené parametry užitkové funkce společně s odpovídajícími daty. Práce dále se zaměřuje na ověření funkcionality aplikace řešením několika vybraných modelových příkladů (Petrohradský paradox, Powerball loterie a Loterie s oky kostky) pro dva rozdílné rozhodovatele. Autorovo navržená aplikace byla využita ke konstrukci užitkových funkcí peněz obou rozhodovatelů. Tyto užitkové funkce jsou následně použity pro výpočet jistotních ekvivalentů obou rozhodovatelů a výsledky jsou porovnány. Aplikace je implementována v programovacím jazyce Python a grafickém frameworku PyQt5.

Úvod a přehled literatury

Tato práce se snaží shrnout různé přístupy z teorie rozhodování a zaměřuje se na teorii užitkové funkce a její využití v rozhodovacím procesu. Klade si za cíl navrhnout software, který umožní rozhodovateli zkonstruovat si vlastní užitkovou funkci a následně ji využít v rozhodovacím procesu. Závěrem jsou uvedeny modelové příklady, kde jsou zhodnocena možná rozhodnutí v závislosti na zvoleném postupu – s využitím užitkové funkce či bez jejího využití.

Cíl práce

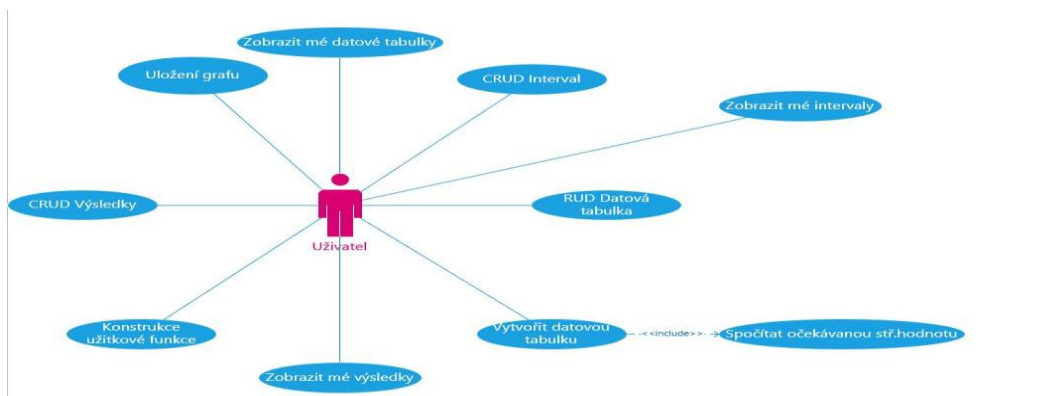
Cílem praktické části je vytvoření aplikace, která umožní rozhodovateli zkonstruovat si vlastní užitkovou funkci peněz a následně ji využít v rozhodovacím procesu. V rámci demonstrace aplikace je záměrem zkonstruovat užitkové funkce dvou různých rozhodovatelů prostřednictvím navržené aplikace a vyřešit zvolené modelové příklady s využitím těchto zkonstruovaných užitkových funkcí a porovnat výsledná řešení s řešeními bez použití užitkové funkce.

Metodika

Je předpokládáno, že užitek ze zisku rovného nula, je roven nule. Důvodem je, že aplikace je určena pouze pro konstrukci užitkové funkce peněz – nikoli užitkové funkce například teploty, kde všichni lidé rozhodně nemají nulový užitek při 0 °C. V rámci peněz je tento předpoklad smysluplný. Dále je aplikace určena pro nezáporné hodnoty zisků.

Programovací jazyk pro implementaci aplikace byl zvolen Python, grafický framework PyQt5 byl použit pro tvorbu grafického rozhraní a dále byly využity matematické knihovny scipy, numpy a matplotlib.

Obrázek 1: Use case diagram



Zdroj: vlastní

Obrázek číslo 1 zobrazuje činnosti, které může uživatel v aplikaci provádět. Existují v aplikaci dva základní datové formáty – tabulky a intervaly. Pro konstrukci užitkové funkce peněz je možné použít jen jeden z datových formátů. Nejdůležitější funkcionalitou aplikace je konstrukce užitkové funkce peněz. V tomto případě je nutné, aby uživatel měl již vytvořené nějaké datové tabulky nebo intervaly na základě kterých proběhne proces konstrukce užitkové funkce peněz. Vzhledem k tomu, že se jedná o aplikaci, jejíž hlavní funkcionalitou je konstrukce užitkové funkce peněz, tak je stanoveno minimum na 0. Po zodpovězení všech požadovaných otázek probíhá vyhledávání optimálních parametrů

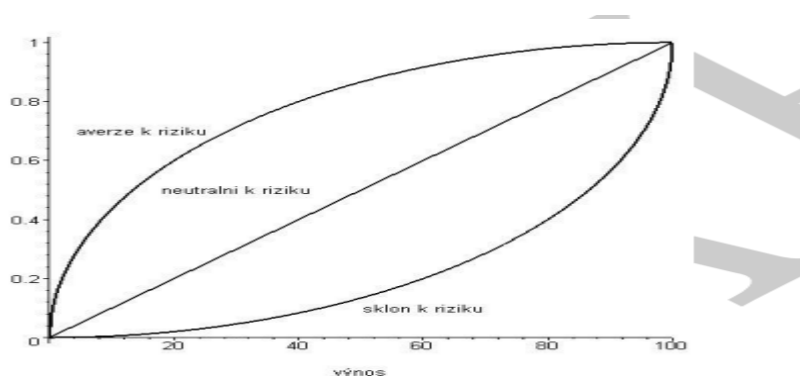
funkce, provedení nelineární regrese a zobrazení výsledků společně s grafem užtkové funkce peněz.

Rovnice (1) znázorňuje, v jakém tvaru je užtková funkce peněz. Tato rovnice je v aplikaci používána pro výpočty.

$$u = a * \exp\left(-\frac{x}{b}\right) + c \quad (1)$$

kde x je zisk, parametry a , b , c jsou parametry užtkové funkce. Platí, že parametr b musí být kladný, parametr a negativní. Na obrázku číslo 2 jsou zobrazeny užtkové funkce podle vztahu rozhodovatele k riziku. Aplikace je zaměřena jen na rozhodovatele s averzí k riziku, a proto výsledný graf užtkové funkce produkovaný autorovo navrženou aplikací je odpovídající tvaru užtkové funkce pro rozhodovatele s averzí k riziku.

Obrázek 2: Užtková funkce podle vztahu rozhodovatele k riziku



Zdroj: (Klicnarová, 2021)

V aplikaci jsou používány tři typy otázek – M_BASED, VALUE_BASED a PROBABILITY_BASED.

M_BASED otázka je v následujícím formátu:

Uveďte nejnižší takovou hodnotu zisku, kdy nejste schopni rozlišit, zda jste dostali tuto hodnotu či dvojnásobnou.

VALUE_BASED otázka je v následujícím formátu:

Pokud byste mohl/a hrát takovou hru, že s pravděpodobností p získáte x_1 a s pravděpodobností $(1-p)$ získáte x_2 . Jakou cenu jste ochoten/na za tuto hru zaplatit? Jaká hodnota Vás dělá indiferentním v této situaci? Tedy částka, od které preferujete raději výplatu před hraním zmíněné hry.

PROBABILITY_BASED otázka je v následujícím formátu:

Pokud byste mohl/a hrát takovou hru, kde první alternativa je, že s pravděpodobností p získáte zisk x_1 a zisk rovný x_2 s pravděpodobností $(1-p)$. Druhá alternativa je, že s jistotou získáte zisk Z . Jaká hodnota pravděpodobnosti p Vás dělá indiferentní/m v této hře? Do jaké maximální hodnoty pravděpodobnosti byste byl/a ochoten hrát zmíněnou hru a od které byste preferoval/a přímou výplatu Z ? Odpověď ve tvaru 0.NN.

Algoritmus v aplikaci pro generování otázek.

1. Generování uvítací zprávy.
2. Generování otázky k získání maximální hodnoty M – otázky typu M_BASED. Hodnota M je přidána do kolekce zisků.
3. Pokud kolekce zisků neobsahuje hodnotu 0, je přidána do kolekce zisků hodnota 0.

- Pokud obsahuje kolekce zisků již zisk 0, tento krok je přeskočen.
4. Seřazení kolekce zisků od nejmenšího po největší.
 5. Nejnížší hodnotě zisku je přiřazen užitek 0, nejvyšší hodnotě zisku je přiřazen užitek 1.
 6. Vytvoření tabulky z kolekce zisků a odpovídajících užiteků.
 7. Generování náhodných otázek. Otázky jsou generovány tak, aby bylo zaručeno rovnoměrné rozložení bodů do pěti intervalů.
 - 7.1 Pokud existují zisky bez přiřazených užiteků, je preferenčně vygenerována náhodně otázka typu PROBABILITY_BASED. Jinak je vygenerována náhodně otázka typu VALUE_BASED.
 - 7.2 Pokud je otázka typu PROBABILITY_BASED, tak na základě zodpovězené pravděpodobnosti je spočítán užitek pro zvolený bod.
 - 7.3 Pokud je otázka typu VALUE_BASED, tak na základě zodpovězené hodnoty je vytvořen nový bod a dopočítán užitek pro daný bod.
 8. Je provedena kontrola rovnoměrného rozložení bodů do intervalů v případě, že byla zadána datová tabulka jako vstup pro konstrukci užitekové funkce peněz. Pokud není dosaženo rovnoměrného rozložení, tak jsou generovány další otázky typu VALUE_BASED k dosažení rovnoměrného rozložení.
 9. Generování koncové zprávy.

Velmi obecný postup pro generování VALUE_BASED otázek v aplikaci je následující:

1. nalezení volných intervalů,
2. náhodné zvolení jednoho z volných intervalů,
3. nalezení vhodných kombinací spadajících do vybraného intervalu,
4. náhodné vybrání jedné z kombinací a navrácení hodnoty.

Rozhodovateli jsou náhodně generovány otázky ve vybraném množství, na které rozhodovatel odpovídá. Na základě těchto odpovědí jsou dopočítávány různé body užitekové funkce peněz rozhodovatele. Jakmile je získáno požadované množství bodů, tak je nutné tyto nalezené body proložit vhodnou funkcí, která nejlépe prokládá získané body – tou je užiteková funkce peněz zobrazená v rovnici (1). Cílem je nalézt takové hodnoty parametrů a , b , c funkce zobrazené rovnicí (1), aby tato funkce nejlépe proložila naše nalezené body. Je zde řešen problém nelineární regrese a je používána Metoda nejmenších čtverců, která je založena na minimalizaci rovnice (2).

$$SSE = \sum_{i=1}^N (y_i - (a \exp(-\frac{x_i}{b}) + c))^2 \quad (2)$$

kde y_i je nalezená hodnota užitku ze zisku x_i a N je počet známých bodů, které jsme získali.

Úloha může mít vícero extrémů, a proto je nutné spustit vyhledávání z většího množství výchozích bodů.

Dále je nutné najít vhodné počáteční parametry a , b , c zobrazené v rovnici (1). Je předpokládáno, že zisk rovný nule odpovídá užitku nula. Díky tomuto předpokladu je možné dosadit do rovnice (1) zisk rovný nule a užitek rovný nule. Po úpravě této rovnice za zmíněných podmínek platí následující vztah, který je využíván pro odhad **parametru c** : kde c i a jsou parametry funkce zobrazené v rovnici (23).

$$c = -a \quad (3)$$

Parametr a je nastavován v intervalu $\langle 1, 1.2 \rangle$. Důvodem je, že užitek dosahuje maximálně hodnoty rovné jedna a lze předpokládat, že funkce může tuto hodnotu nepatrně překročit. K výpočtu **parametru b** je nutné získat maximální hodnotu M uživatele pomocí otázky popsané v kapitole 6.3.2.1 (M_BASED otázka). Na základě získané maximální hodnoty M jsou počítány odhady parametrů pro horní a dolní hranici parametru b.

$$\text{mindolníhraniceb} = \frac{M}{5} \quad (4)$$

$$\text{maxhorníhraniceb} = \frac{M}{3} \quad (5)$$

M je maximální hodnota získaná prostřednictvím otázky na maximální hodnotu M. Tato hodnota ukazuje, kde je již užitečná funkce celkem ustálená. Rozdíl mezi M a 2M je velmi malý. Logika volby minimální a maximální hodnoty pro parametr b je založena na tom, že hledáme takové hodnoty, aby v bodě M nebyla hodnota užitečné funkce příliš brzy u jedné (výrazně menší hodnoty než M mají být pro rozhodovatele rozlišitelné).

Obrázek 3: Algoritmus pro získání optimální startovací hodnoty parametru b

```
def estimateInitialParamsWithComparison(self, resultUpperBoundary, resultLowerBoundary, levels):
    if levels==0:
        return min(resultLowerBoundary, resultUpperBoundary)

    # Pokud jsou nalezené parametry stejné u dolní i horní hranice tak končíme vyhledávání a vracíme nalezené parametry
    if round(resultUpperBoundary.result.x[0],2) == round(resultLowerBoundary.result.x[0], 2)
    and round(resultUpperBoundary.result.x[1],2) == round(resultLowerBoundary.result.x[1], 2)
    and round(resultUpperBoundary.result.x[2],2) == round(resultLowerBoundary.result.x[2],2):
        return resultUpperBoundary
    else:
        newNodeEstimate = (resultUpperBoundary.bEstimate+resultLowerBoundary.bEstimate)/2
        print(newNodeEstimate)
        newNodeResult = self.combineParameters(newNodeEstimate)
        if resultUpperBoundary.SSE <= resultLowerBoundary.SSE:
            return self.estimateInitialParamsWithComparison(resultUpperBoundary, newNodeResult, levels-1)
        else:
            return self.estimateInitialParamsWithComparison(newNodeResult, resultLowerBoundary, levels-1)
```

Zdroj: vlastní

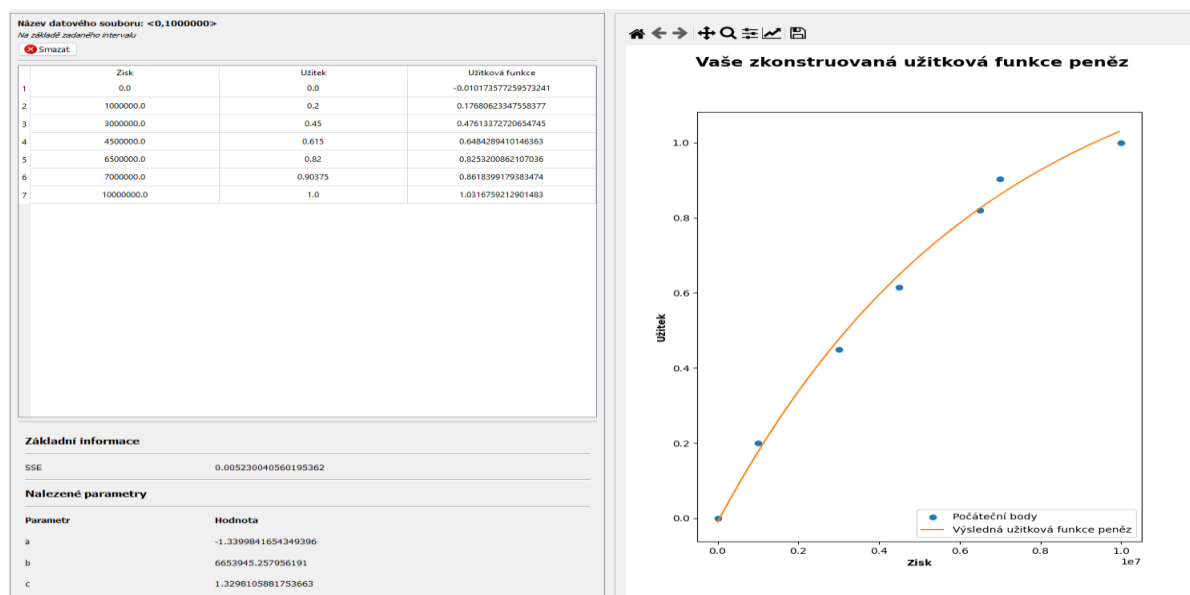
Obrázek číslo 3 zobrazuje kód a algoritmus pro získávání optimální startovací hodnoty parametru b do nelineární regrese (**resultUpperBoundary** v první iteraci = maxhorníhraniceb, **resultLowerBoundary** v první iteraci = minhorníhraniceb). Pokud nalezené parametry užitečné funkce pro horní hranici parametru b jsou stejné jako pro dolní hranici parametru b, je vyhledávání ukončeno a je vrácen výsledek. Pokud nalezené parametry nejsou ekvivalentní, je vytvořena nová hodnota odhadu parametru b, která je reprezentována proměnnou **newNodeEstimate**. Tato proměnná je počítána jako součet horní a dolní hranice, které byly zadány jako vstupní parametry do této metody a tento součet je vydělen dvěma. Je hledán nejlepší výsledek pro nový odhad parametru b (proměnná **newNodeEstimate**). Dále jsou porovnávány SSE jednotlivých hranic. Pokud SSE pro horní hranici nižší nebo rovno SSE pro dolní hranici, je volána stejná metoda **estimateInitialParamsWithComparison** rekurzivně – s následujícími parametry: nová horní hranice je horní hranice vložená jako vstupní parametr do této metody (**resultUpperBoundary**), dolní hranice je výsledek získaný pomocí nového odhadu parametru b (proměnná **newNodeResult**). Algoritmus se vrací opět na začátek s jinými vstupními parametry. Pokud SSE pro dolní hranici je nižší než SSE pro horní hranici, tak je volána rekurzivně stejná metoda. Vstupními parametry jsou: horní hranici je výsledek získaný pomocí nového odhadu parametru b (**newNodeResult**), nová dolní hranice je dolní hranice vložená do této metody jako vstupní parametr. Algoritmus se vrací na začátek s jinými zadanými vstupními parametry.

Výsledky

Pro řešení modelových příkladů byly zkonstruovány dvě užitečné funkce peněz dvou různých rozhodovatelů. Ke zkonstruování těchto užitečných funkcí byla využita autorem navržená

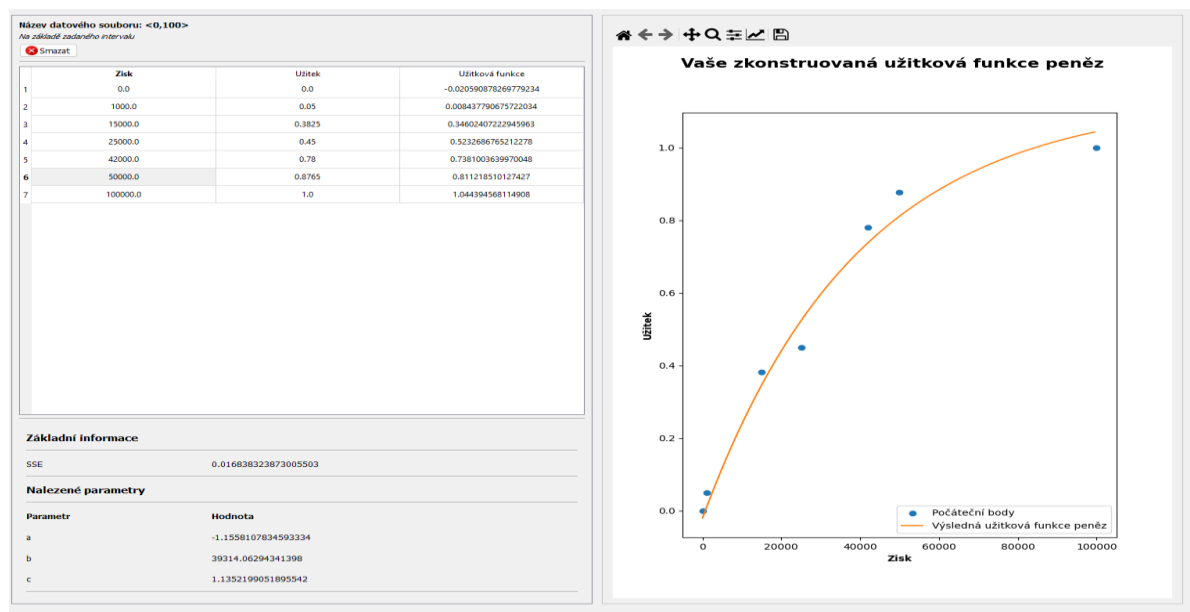
aplikace. Vygenerované užtkové funkce dvou různých rozhodovatelů jsou zobrazeny na obrázcích číslo 4 a 5. Na těchto obrázcích je možné vidět i grafické rozhraní aplikace a výstupní obrazovku po ukončení procesu konstrukce užtkové funkce peněz. Každý rozhodovatel odpověděl na pět náhodných otázek a jednu otázku ohledně maximální hodnoty M daného rozhodovatele. První rozhodovatel zvolil maximální hodnotu M 10^7 a jeho užtková funkce je zobrazena na obrázku číslo 4. Druhý rozhodovatel zvolil maximální hodnotu $M=10^5$ a jeho užtková funkce je zobrazena na obrázku číslo 5. První rozhodovatel tedy může být člověk bohatší, s vyšším kapitálem, zatímco druhý rozhodovatel se nechce pouštět do většího rizika. U všech příkladů je cílem spočítat jistotní ekvivalenty dvou různých rozhodovatelů na základě sestrojených užtkových funkcí peněz. Zvolené příklady jsou následující: Petrohradský paradox, Powerball loterie a Loterie s oky kostky (je házeno kostkou a je vypláceno 10^k , kde k je počet ok kostky).

Obrázek 4: Užtková funkce prvního rozhodovatele



Zdroj: vlastní

Obrázek 5: Užtková funkce druhého rozhodovatele



Zdroj: vlastní

Tabulka 1 – Výsledky řešených příkladů

Modelový příklad	$\hat{x}$ – 1. rozhodovatel	$\hat{x}$ – 2. rozhodovatel	Bayesovo pravidlo
Petrohradský paradox	11,8878	8,1872	Nekonečno
Powerball loterie	0,5045	0,4397	0,5065
Loterie s oky kostky	3033,3304	889,0469	3179,8

Zdroj: vlastní

V tabulce číslo 1 jsou uvedené zkompleťované výsledky pro všechny řešené modelové příklady. Na základě získaných výsledků je možné konstatovat, že jistotní ekvivalenty obou rozhodovatelů jsou pro každý modelový příklad menší než očekávaná střední hodnota související hry. Dále je evidentní, že jistotní ekvivalenty obou rozhodovatelů nejsou příliš rozdílné pro první dva modelové příklady (Petrohradský paradox a Powerball loterie) a to i přesto, že první rozhodovatel je bohatší a s menší averzí k riziku než rozhodovatel druhý. Petrohradský paradox je velmi extrémní, a proto jistotní ekvivalenty obou rozhodovatelů nejsou příliš rozdílné. Nicméně je v tomto případě možné stanovit jistotní ekvivalenty obou rozhodovatelů i přesto, že očekává střední hodnota této hry je rovna nekonečnu. U Powerball loterie jsou nepříliš rozdílné výsledky jistotních ekvivalentů pro oba rozhodovatele dány tím, že se jedná o hru, kde s vysokou pravděpodobností není žádná výhra a také platí, že se zvyšující se možnou výhrou velmi rychle klesá pravděpodobnost výhry. Třetí modelový příklad (Loterie s oky kostky) nejlépe demonstruje rozdíl mezi bohatým a chudším rozhodovatelem, neboť jistotní ekvivalenty obou rozhodovatelů jsou velmi rozdílné.

Závěr

Závěrem lze říct, že aplikace byla úspěšně implementována a použita na modelové příklady. Autor se domnívá, že pomocí této aplikace lze řešit konkrétní rozhodovací problémy. Nicméně kvalita zkonstruované funkce závisí na množství získaných bodů při procesu generování otázek. Lze také konstatovat, že otázky jsou generovány rovnoměrně do pěti intervalů, nicméně pokud rozhodovatelem vytvořený interval, na kterém se generují otázky, je velmi velký, tak je sice dosaženo rovnoměrného rozložení do pěti dílčích intervalů, ale jednotlivé dílčí intervaly mohou být celkem velké. Další nevýhodou může být, že obecně generované otázky mohou být celkem složité a pro laika těžko zodpověditelné. Potencionální vylepšení aplikace by mohlo být dosaženo implementací konstrukce uživatelské funkce prostřednictvím alternativní uživatelské funkce, u které je vyžadována jen odpověď na jednu otázku a je tedy mnohem rychlejší, pokud rozhodovatel nechce podstupovat poměrně dlouhý proces, ve kterém je nutné odpovídat na větší množství otázek. Rozhodovatel by si pak mohl v rámci aplikace sám určit, jaký způsob konstrukce uživatelské funkce chce použít. Další možné vylepšení by byla konstrukce uživatelské funkce nejen peněz, ale i pro větší spektrum možností. Aplikace bude využita například k výuce předmětů, které se zabývají uživatelskou funkcí na Ekonomické fakultě Jihočeské univerzity.

Seznam literatury použité v tezích

Kniha

Klicnarová, J. (2021). Rozhodovací modely. Jihočeská univerzita

Další bibliografické údaje

Type of thesis: master

Author: Bc. Patrik Marýška

Academic Year: 2022/2023

Department: Department of Applied Mathematics and Informatics, Faculty of Economics, University of South Bohemia in Ceske Budejovice, Czech Republic

Thesis supervisor: doc. RNDr. Jana Klicnarová, Ph.D

Number of pages: 9

Language of thesis: CZ

Title of Thesis: Utility function in the decision-making process

Annotation:

This diploma thesis deals with the process of creating an application with appropriate algorithms and mechanisms for constructing utility function for money for a given decision maker. Related theoretical topics from the decision theory, utility theory and especially the process of construction of the utility function for money are explained. Application randomly generates questions in given intervals for the decision maker. After the process of the generating question is finished, algorithm for finding optimal starting values is applied. This algorithm gradually compares SSE in the calculated interval. Nonlinear regression is used to fit the data gathered in previous steps. The graph of the found utility function is drawn and found parameters of the utility function are displayed together with the corresponding data. The paper goes on to evaluate the functionality of the application by solving model examples (St.Petersburg paradox, Powerball lottery and Cube lottery) for two different decision makers. These decision makers used designed application for constructing their utility functions for money. These two utility functions are utilized for calculating certainty equivalents for both decision makers and the results are compared. Application is implemented in Python and PyQt5.

Keywords: Application software for construction of the utility function for money, Utility theory, Decision theory, Python

Přílohy vázané v práci:

- **utilityconstructor.zip** obsahuje implementovanou aplikaci. Pro spuštění aplikace je nutné vyextrahovat na libovolné místo na disku a následně nalézt soubor main.exe, který aplikaci spouští. Aplikaci v tomto formátu lze spustit jen na OS Windows.
- **modelovepriklady.xlsx** obsahuje řešené modelové příklady pro oba rozhodovatele v MS Excel.