

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A
INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2025

Noemi Dvořáková

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Porovnání vlivu různých ztrátových funkcí v symbolické regresi
Bakalářská práce

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Noemi Dvořáková**
Osobní číslo: **I22087**
Studijní program: **B0688A140009 Informační technologie**
Téma práce: **Porovnání vlivu různých ztrátových funkcí v symbolické regresi**
Zadávající katedra: **Katedra informačních technologií**

Zásady pro vypracování

Cílem této bakalářské práce je analyzovat vliv různých ztrátových funkcí na kvalitu symbolické regrese, která je realizována pomocí genetického programování. Student se zaměří na teoretické zpracování problematiky symbolické regrese, genetického programování a přehled ztrátových funkcí používaných v oblasti strojového učení. V praktické části navrhne experimenty, ve kterých budou různé ztrátové funkce aplikovány na vybrané jednorozměrné problémy a datasety používané v symbolické regresi. Na základě výsledků experimentů pak vyhodnotí, jaký vliv mají jednotlivé ztrátové funkce na kvalitu výsledných modelů.

Rozsah pracovní zprávy: **min. 30 stran**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

HYNEK, Josef. Genetické algoritmy a genetické programování. Praha: Grada, 2008. Průvodce (Grada). ISBN 978-80-247-2695-3.

MITCHELL, Melanie. An introduction to genetic algorithms. Cambridge, Mass.: MIT Press, c1996. ISBN 978-0262133166.

POLI, Riccardo, W. B. LANGDON, Nicholas F. MCPHEE a John R. KOZA. A field guide to genetic programming. [S.l.: Lulu Press], 2008. ISBN 978-1409200734.

Vedoucí bakalářské práce: **Ing. Jan Merta, Ph.D.**
Katedra softwarových technologií

Datum zadání bakalářské práce: **15. prosince 2024**

Termín odevzdání bakalářské práce: **16. května 2025**

prof. Ing. Petr Doležel, Ph.D. v.r.
děkan

L.S.

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

V Pardubicích dne 28. února 2025

Prohlašuji:

Práci s názvem Porovnání vlivu různých ztrátových funkcí v symbolické regresi jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

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

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 15. 05. 2025

Noemi Dvořáková v.r.

PODĚKOVÁNÍ

Ráda bych poděkovala doktoru Janu Mertovi za cenné rady a podporu při vedení mé bakalářské práce. Dále bych ráda poděkovala svým blízkým přítelkyním a své rodině za veškerou podporu kterou mi projevily během studia.

ANOTACE

Cílem této bakalářské práce je analyzovat vliv různých ztrátových funkcí na kvalitu symbolické regrese, která je realizována pomocí genetického programování. Student se zaměří na teoretické zpracování problematiky symbolické regrese, genetického programování a přehled ztrátových funkcí používaných v oblasti strojového učení. V praktické části navrhne experimenty, ve kterých budou různé ztrátové funkce aplikovány na vybrané jednorozměrné problémy a datasety používané v symbolické regresi. Na základě výsledků experimentů pak vyhodnotí, jaký vliv mají jednotlivé ztrátové funkce na kvalitu výsledných modelů.

KLÍČOVÁ SLOVA

Symbolická regrese, ztrátová funkce, genetické programování

TITLE

Comparison of the influence of different loss functions in symbolic regression

ANNOTATION

The aim of this bachelor's thesis is to analyze the impact of various loss functions on the quality of symbolic regression performed using genetic programming. The student will focus on the theoretical background of symbolic regression, genetic programming, and an overview of loss functions commonly used in the field of machine learning. In the practical part, experiments will be designed in which different loss functions will be applied to selected one-dimensional problems and datasets used in symbolic regression. Based on the results of these experiments, the influence of individual loss functions on the quality of the resulting models will be evaluated.

KEYWORDS

Symbolic regression, loss function, genetic programming

OBSAH

SEZNAM OBRÁZKŮ	9
SEZNAM ZKRATEK A ZNAČEK	11
ÚVOD	12
1 TEORETICKÁ ČÁST	13
1.1 Symbolická regrese	13
1.1.1 Ztrátové funkce	13
1.2 Genetické programování	19
1.2.1 Terminály	20
1.2.2 Funkce	20
1.2.3. Inicializace počáteční populace	21
1.2.4 Ohodnocení jedinců	23
1.2.5 Selektce	23
1.2.6 Křížení	24
1.2.7 Mutace	27
1.2.8 Ukončovací kritérium	27
2 EXPERIMENTÁLNÍ ČÁST	29
2.1 Implementace genetického programování	29
2.1.1 Testovací funkce	29
2.1.2 Použité metody a nastavení parametrů	30
2.2 Experimenty	30
2.2.1 Experiment ověřující stabilitu výsledků	31
2.2.2 Experimenty první konfigurace	32
2.2.3 Experimenty druhé konfigurace	42
2.2.4 Vyhodnocení všech experimentů	51
ZÁVĚR	52
POUŽITÁ LITERATURA	54
SEZNAM PŘÍLOH	57

SEZNAM OBRÁZKŮ

Obrázek 1: Vizualizace principu výpočtu MAE (zdroj: vlastní)	14
Obrázek 2: Vizualizace principu výpočtu MSE (zdroj: vlastní).....	15
Obrázek 3: Vizualizace principu Huberovy ztrátové funkce (zdroj: vlastní)	17
Obrázek 4: Vizualizace principu výpočtu GDL (zdroj: vlastní).....	18
Obrázek 5: Náhodně vygenerovaný syntaktický strom pomocí metody úplné [10].....	22
Obrázek 6: Náhodně vygenerovaný syntaktický strom pomocí metody růstové [10].....	23
Obrázek 7: Vizualizace křížení podstromů (zdroj: vlastní)	25
Obrázek 8: Vizualizace jednobodového křížení (zdroj: vlastní)	26
Obrázek 9: Vizualizace rovnoměrného křížení (zdroj: vlastní).....	27
Obrázek 10: Graf vývoje kumulativního průměru v závislosti na počtu provedených testů (zdroj: vlastní).....	31
Obrázek 11: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F1 při první konfiguraci (zdroj: vlastní).....	33
Obrázek 12: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F2 při první konfiguraci (zdroj: vlastní).....	34
Obrázek 13: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F3 při první konfiguraci (zdroj: vlastní).....	35
Obrázek 14: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F4 při první konfiguraci (zdroj: vlastní).....	36
Obrázek 15: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F5 při první konfiguraci (zdroj: vlastní).....	37
Obrázek 16: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F6 při první konfiguraci (zdroj: vlastní).....	38
Obrázek 17: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F7 při první konfiguraci (zdroj: vlastní).....	39
Obrázek 18: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F8 při první konfiguraci (zdroj: vlastní).....	40
Obrázek 19: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F1 při druhé konfiguraci (zdroj: vlastní)	42
Obrázek 20: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F2 při druhé konfiguraci (zdroj: vlastní)	44
Obrázek 21: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F3 při druhé konfiguraci (zdroj: vlastní)	45
Obrázek 22: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F4 při druhé konfiguraci (zdroj: vlastní)	46
Obrázek 23: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F5 při druhé konfiguraci (zdroj: vlastní)	47
Obrázek 24: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F6 při druhé konfiguraci (zdroj: vlastní)	48
Obrázek 25: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F7 při druhé konfiguraci (zdroj: vlastní)	49

Obrázek 26: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F8 při druhé konfiguraci (zdroj: vlastní)50

SEZNAM ZKRATEK A ZNAČEK

GDL – Gradient Difference Loss

HUBER – Huberova ztrátová funkce

MAE – Mean Absolute Error (průměrná absolutní chyba)

MSE – Mean Square Error (průměrná kvadratická chyba)

RMSE – Root Mean Square Error (odmocněná průměrná kvadratická chyba)

ÚVOD

Symbolická regrese představuje jednu z běžných úloh řešených pomocí genetického programování a dalších evolučních algoritmů. Cílem je nalézt matematický výraz, který by co nejlépe popisoval danou množinu dat. Jednou z hlavních výhod symbolické regrese je, že na rozdíl od jiných regresních metod nevyžaduje předem stanovený tvar modelu, což umožňuje evolučním algoritmům efektivně hledat a optimalizovat možné tvary výsledného modelu.

Při řešení symbolické regrese pomocí genetického programování je jednou z klíčových složek i volba ztrátové funkce, která určuje, jakým způsobem se měří kvalita navrhovaného modelu. V oblasti symbolické regrese se běžně používá řada různých ztrátových funkcí, z nichž každá má odlišné vlastnosti a může ovlivnit průběh evolučního procesu i samotné výsledky. Výběr vhodné ztrátové funkce tak může být zásadní pro dosažení co nejlepších modelů.

Tato práce se zabývá právě problematikou vlivu různých ztrátových funkcí na kvalitu řešení problému symbolické regrese realizované pomocí genetického programování. V teoretické části jsou nejprve představeny základní principy symbolické regrese, popsány jednotlivé ztrátové funkce a základní charakteristika genetického programování, včetně podrobného vysvětlení jeho jednotlivých částí.

Praktická část se zaměřuje na návrh a realizaci několika experimentů, ve kterých jsou různé ztrátové funkce aplikovány na vybrané jednorozměrné testovací funkce. Cílem těchto experimentů je zjistit, jak jednotlivé ztrátové funkce ovlivňují kvalitu výsledných modelů z hlediska dosažené chyby a stability řešení, porovnat vybrané ztrátové funkce z hlediska úspěšnosti a zjistit co ovlivňuje výběr vhodné ztrátové funkce. Výsledky experimentů by měly přispět k lepšímu pochopení role ztrátové funkce v procesu symbolické regrese a poskytnout doporučení pro jejich výběr.

1 TEORETICKÁ ČÁST

1.1 Symbolická regrese

Symbolická regrese je jednou z metod z oblasti regresní analýzy sloužící k nalezení matematické funkce, která nejlépe popisuje danou množinu dat. Na rozdíl od jiných regresních metod, jako například lineární nebo kvadratické regrese, které mají předem určenou strukturu a vztahy mezi proměnnými a jejichž algoritmus se pokouší nalézt nejlepší řešení úpravou pouze číselných koeficientů, u symbolické regrese není předem stanoven konkrétní tvar funkce. Namísto toho pracuje s předem stanovenou množinou terminálů a funkcí, sestavuje z nich správnou strukturu funkce a hledá vhodné parametry. Symbolická regrese je obvykle implementována pomocí evolučních algoritmů, přičemž nejrozšířenějším postupem je využití genetického programování. [1, 2]

Symbolická regrese dokáže odvodit rovnice funkcí čistě na základě dat, aniž by bylo nutné mít předem hluboké znalosti zkoumaného systému. Díky tomu nachází široké uplatnění ve vědě, technologii, ekonomii i sociálních vědách, přestože její výpočetní náklady jsou poměrně vysoké a výsledné rovnice mohou být složité. I proto je třeba znát a využívat efektivních programovacích strategií. [2]

1.1.1 Ztrátové funkce

Ztrátové funkce slouží k určení kvality nalezeného modelu. Daný model je hodnocen na základě množství dat, která dokáže správně zpracovat, a chyby, která může vyjadřovat například rozdíl mezi skutečnými daty a predikovanými hodnotami. Jelikož vyhodnocení probíhá na více datových bodech, chyba se často vyjadřuje průměrem. V kontextu symbolické regrese hrají klíčovou roli při řízení evolučního procesu, jelikož umožňují odlišit slibné modely s nižší chybovostí od méně přesných modelů. [2]

MAE

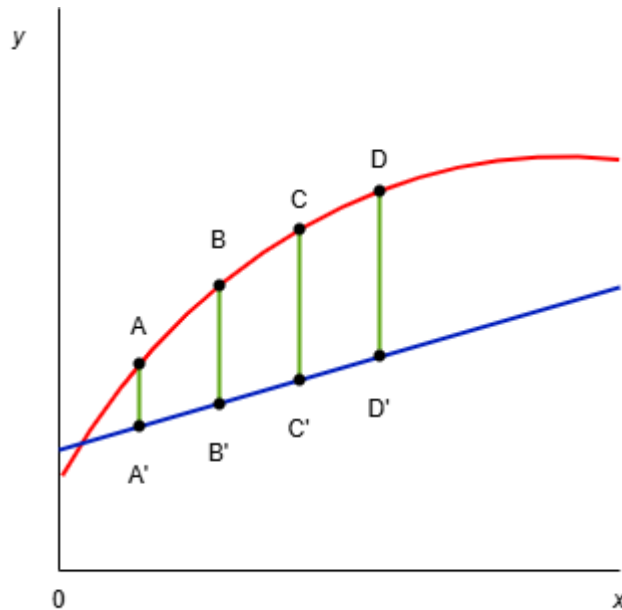
Průměrná absolutní chyba (Mean Absolute Error, zkráceně MAE) je jednou z nejznámějších a nejběžněji používaných ztrátových funkcí. Je založena na sečtení absolutních hodnot všech chyb, přičemž chyba je definována jako rozdíl mezi skutečnou a predikovanou hodnotou. Tento součet je následně vydělen celkovým počtem hodnot, čímž vznikne průměrná hodnota absolutních chyb. [3]

Tuto ztrátovou funkci lze tedy zapsat následujícím způsobem:

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i|$$

kde n označuje celkový počet pozorování a e_i je chyba u i -tého pozorování, tedy rozdíl mezi skutečnou a predikovanou hodnotou. [3]

Princip výpočtu hodnoty MAE lze ilustrovat na následujícím obrázku:



Obrázek 1: Vizualizace principu výpočtu MAE (zdroj: vlastní)

Červená křivka na obrázku představuje funkci vytvořenou ze skutečných dat, na níž jsou vyznačeny hodnoty označené body A až D. Modrá křivka znázorňuje predikovanou funkci s odpovídajícími hodnotami A' až D'. Zelené úsečky pak představují rozdíly mezi skutečnými (A až D) a predikovanými (A' až D') hodnotami. Tyto rozdíly se při výpočtu MAE převedou na absolutní hodnoty, sečtou se a jejich součet se vydělí počtem pozorování (v tomto případě čtyřmi), čímž vznikne výsledná průměrná absolutní chyba.

MSE

Jednou z dalších známých funkcí je průměrná kvadratická chyba (Mean Square Error, zkráceně MSE). Na rozdíl od MAE součet absolutních chyb umocňuje na druhou a až poté ji dělí celkovým počtem hodnot. [3]

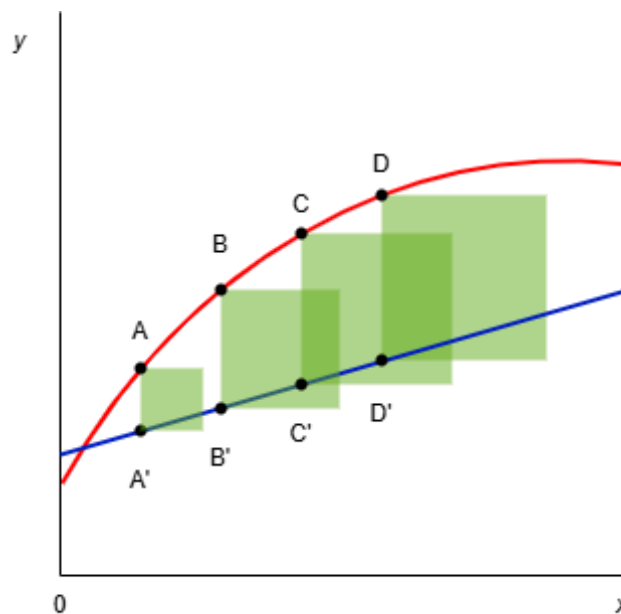
MSE lze tedy vyjádřit následovně:

$$MSE = \frac{1}{n} \sum_{i=1}^n |e_i|^2$$

kde n opět značí celkový počet pozorování a e_i reprezentuje chybu u i -tého pozorování neboli rozdíl mezi skutečnou a predikovanou hodnotou. [3]

Kvůli tomuto postupu mají větší odchylky výraznější vliv na celkovou hodnotu chyby než menší odchylky. Výsledná hodnota je tedy více penalizovaná za větší odchylky od původních dat. [3]

Výpočet MSE můžeme vizualizovat pomocí následujícího obrázku:



Obrázek 2: Vizualizace principu výpočtu MSE (zdroj: vlastní)

Na obrázku opět červená linie znázorňuje funkci skutečných hodnot s body označenými jako A až D. Modrá křivka představuje predikovanou funkci s body A' až D', které odpovídají predikovaným hodnotám. Při výpočtu MSE jsou rozdíly mezi skutečnými a predikovanými hodnotami umocněny na druhou, což je znázorněno zelenými čtverci. Plochy těchto čtverců se poté sečtou a tento součet se vydělí celkovým počtem pozorování (v tomto případě čtyřmi), čímž vznikne výsledná hodnota MSE.

RMSE

Odmocněná průměrná kvadratická chyba (Root Mean Square Error, zkráceně RMSE) je často preferovanou variantou MSE. Lze ji získat prvotním vypočítáním MSE a následným

odmocněním výsledné hodnoty. Tento krok slouží především k převedení hodnoty chyby zpět do stejných jednotek jako mají původní data, což usnadňuje interpretaci výsledků. RMSE si zachovává vlastnost MSE, kdy větší odchylky mají výraznější dopad na celkovou hodnotu chyby, ale poskytuje srozumitelnější výsledky. [4]

Matematicky je RMSE vyjádřena takto:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n |e_i|^2}$$

kde n opět představuje celkový počet pozorování a e_i odpovídá chybě u i -tého pozorování čili rozdíl mezi skutečnou a predikovanou hodnotou. [3]

Huberova ztrátová funkce

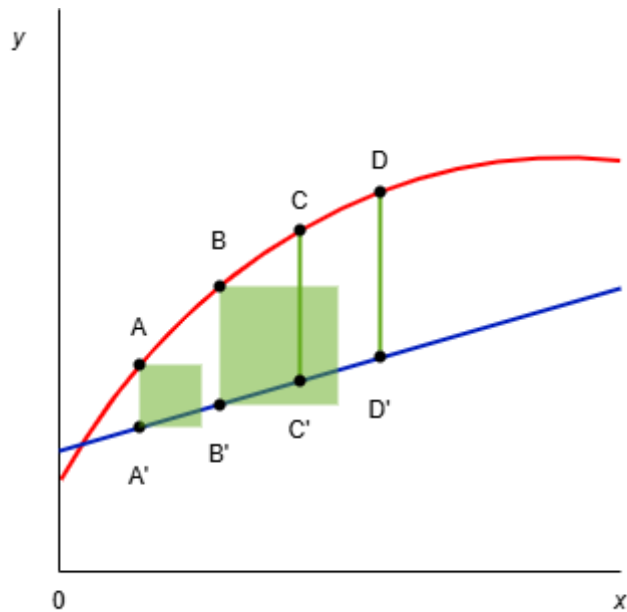
Huberova ztrátová funkce kombinuje výhody MAE a MSE tak, že vstupy s velkými odchylkami od skutečných dat jsou penalizovány lineárně, podobně jako je tomu u MAE, zatímco malé chyby jsou započítávány kvadraticky, podobně jako u MSE. Díky tomu je tato funkce odolnější vůči extrémním hodnotám. [5, 6]

Huberovu ztrátovou funkci lze tedy definovat následovně:

$$V_{\theta}(t) = \begin{cases} t^2, & \text{pro } |t| \leq \theta, \\ 2\theta|t| - \theta^2, & \text{jinak.} \end{cases}$$

Kde t je rozdíl mezi skutečnou a predikovanou hodnotou a θ určuje hodnotu, kdy přecházíme z kvadratického režimu do lineárního. [7]

Princip Huberovy ztrátové funkce lze znázornit na následujícím obrázku:



Obrázek 3: Vizualizace principu Huberovy ztrátové funkce (zdroj: vlastní)

Na obrázku opět vidíme červenou křivku znázorňující funkci skutečných hodnot, které jsou označeny body A až D, a modrou linii představující predikovanou funkci s odpovídajícími predikovanými hodnotami A' až D'. Protože Huberova ztrátová funkce v případě malých rozdílů mezi skutečnými a predikovanými hodnotami přechází do kvadratického režimu (analogicky k MSE), jsou rozdíly mezi body A, A' a B, B' znázorněny zelenými čtverci. Pomyslná hranice, ve vzorci značená jako θ , která určuje přechod z kvadratického na lineární režim, je v tomto případě větší než rozdíl mezi body B a B', ale menší než rozdíl mezi body C a C'. Pro rozdíly mezi C, C' a D, D' se tedy funkce chová lineárně (analogicky k MAE), což je na obrázku znázorněno pouze zelenými čarami.

GDL

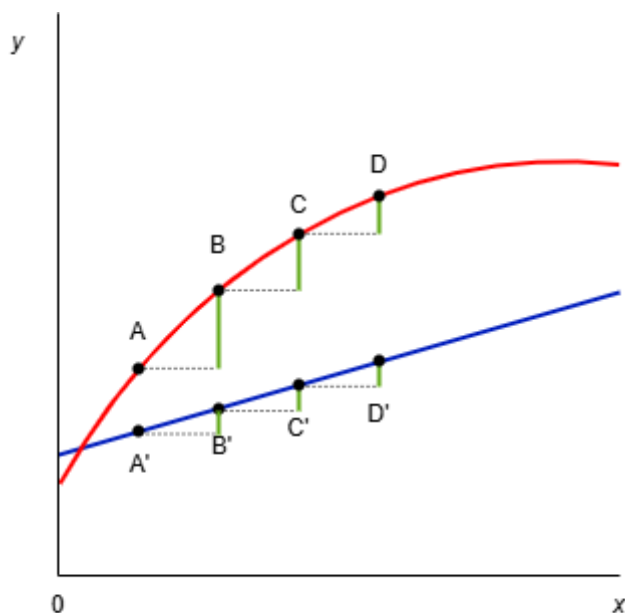
Méně známým přístupem je použití Gradient Difference Loss, zkráceně GDL, který se nezaměřuje přímo na hodnoty funkce, nýbrž na tvar funkce. Namísto výpočtu rozdílů mezi skutečnými a predikovanými hodnotami GDL nejprve spočítá rozdíl mezi sousedními hodnotami – jak u predikované funkce, tak u funkce tvořené skutečnými daty. Nakonec jsou tyto rozdíly porovnávány mezi sebou. [8]

Matematicky lze tento postup vyjádřit následovně:

$$GDL = \sum_{i=1}^n |(y_{i+1} - y_i) - (\hat{y}_{i+1} - \hat{y}_i)|$$

kde n opět značí celkový počet pozorování, hodnoty y_{i+1} a y_i představují sousední hodnoty ve funkci se skutečnými daty a hodnoty \hat{y}_{i+1} a \hat{y}_i reprezentují sousední hodnoty predikované funkce. [8]

Pro názornost lze princip výpočtu GDL ukázat na následujícím obrázku:



Obrázek 4: Vizualizace principu výpočtu GDL (zdroj: vlastní)

Červená křivka na obrázku opět reprezentuje funkci vytvořenou ze skutečných dat, jejíž hodnoty jsou označeny body A až D, a modrá křivka představuje predikovanou funkci s odpovídajícími hodnotami v bodech A' až D'.

Místo toho, aby byly porovnávány přímo hodnoty jednotlivých bodů, jako tomu bylo u předešlých ztrátových funkcí, GDL sleduje výškové rozdíly mezi sousedními body. Tyto rozdíly jsou na obrázku znázorněny zelenými úsečkami. Výpočet GDL spočívá v porovnání délky těchto čárkovaných úseků mezi skutečnou a predikovanou funkcí a v následném sečtení absolutních hodnot těchto rozdílů. V tomto příkladu by se tedy GDL počítala takto:

$$GDL = |(y_B - y_A) - (y_{B'} - y_{A'})| + |(y_C - y_B) - (y_{C'} - y_{B'})| + |(y_D - y_C) - (y_{D'} - y_{C'})|$$

Tento přístup umožňuje zkoumat změny gradientu (sklonu křivek), což odráží tvar funkce, ale nezohledňuje umístění funkce v grafu vzhledem k ose y . Proto se nejedná o klasickou ztrátovou funkci, ale spíše o regularizační člen, který se používá ve spojení s jinými ztrátovými funkcemi. [8]

1.2 Genetické programování

Genetické programování je metoda z kategorie evolučních algoritmů, která umožňuje počítačům samostatně nacházet řešení problémů bez nutnosti předem určovat jejich konkrétní podobu. [9] Vychází z genetických algoritmů inspirovaných Darwinovou teorií evoluce, která popisuje, jak se organismy vyvíjejí po mnoho generací prostřednictvím přirozeného výběru. Jedinci s výhodnějšími vlastnostmi mají vyšší šanci na přežití a reprodukci, a tím i ovlivnění vývoje svého druhu. [10] Pro identifikaci zdatnějších jedinců je třeba kvantifikovat kvalitu každého jedince, k čemuž se používá vyjádření pomocí hodnoty fitness (ohodnocení). [9, 10]

Stejně jako genetické algoritmy i genetické programování využívá genetických operací, jako je křížení (anglicky crossover) či mutace (anglicky mutation). [9, 10, 11] Tyto operace budou podrobně popsány v dalších podkapitolách, přesto si můžeme stručně představit jednoduchý evoluční algoritmus, kterým genetické algoritmy a genetické programování, řídí:

1. Náhodné vytvoření počáteční populace.
2. Vypočítání fitness (ohodnocení) každého jedince v populaci. Pokud je splněno ukončovací kritérium (například dosažení maximálního počtu generací či prahové hodnoty fitness), algoritmus se ukončí a jedinec s nejlepší hodnotou fitness je vybrán jako výsledné řešení.
3. Selektce (výběr) jedinců s pravděpodobností založenou na jejich hodnotách fitness.
4. Spárování vybraných jedinců a vytvoření jejich potomků za využití genetických operací, čímž vznikne nová generace, která nahradí tu stávající.
5. Návrat ke kroku 2. [9, 10, 12]

Genetické programování se od genetických algoritmů v několika ohledech liší. Jeden z nejzásadnějších rozdílů je v reprezentaci jedince. Zatímco v genetických algoritmech se nejčastěji využívá lineární reprezentace o pevně dané délce (používáno označení chromozom), v genetickém programování je využíváno reprezentace hierarchické v podobě syntaktického stromu. Ten se skládá z primitivních funkcí a terminálů, jež je třeba uživatelem předem definovat. [10, 11, 13]

Díky využití hierarchické reprezentace jedinců a schopnosti zvětšování (i případného zmenšování) jedinců představujících možná řešení daného problému, což vede k objevování komplikovanějších (popřípadě jednodušších) řešení, je možné pomocí genetického

programování generovat počítačové programy. [10, 13] Díky tomu má genetické programování širokou škálu využitelnosti a již bylo použito k řešení celé řady problémů, dokonce i k patentovaným vynálezům. Genetické programování je však technika velmi náročná na výpočetní čas, která se opírá o náhodné procesy, proto nelze nikdy zaručit nalezení optimálních výsledků. [9, 10]

1.2.1 Terminály

Terminály v genetickém programování reprezentují buď konstanty, které mohou nabývat různých hodnot, jako jsou například celá čísla, náhodně vygenerované konstanty či booleovské hodnoty, nebo proměnné zastupující externí vstupy programu. Ve struktuře syntaktického stromu se terminály vždy nacházejí v koncových uzlech (listech). [11]

Množina terminálů tedy může obsahovat:

1. Externí vstupy programu, obvykle reprezentované pojmenovanými proměnnými (např: x , y).
2. Funkce bez parametrů, např. funkce `rand()`, které při každém zavolání vrací náhodnou hodnotu, nebo funkce vracující senzorická data. Bezparametrické funkce mohou mít kromě návratu hodnoty i jiné vedlejší efekty, jako například změnu globálních datových struktur.
3. Konstanty, buď předem definované, nebo vzniklé během procesu mutace. Také lze využít tzv. efemérní náhodné konstanty, které při konstrukci stromu generují náhodnou hodnotu, jež narozdíl od funkce `rand()` přetrvává po celou dobu běhu programu. [9]

1.2.2 Funkce

Funkce se od terminálů liší především tím, že přijímají vstupní parametry. Počet těchto parametrů se označuje jako arita funkce. Funkce, jež mohou mít libovolnou aritu, se nacházejí v kořeni a vnitřních uzlech syntaktického stromu. [10]

Množina funkcí může obsahovat:

1. Aritmetické operace – například sčítání, odčítání, násobení apod.
2. Matematické funkce – například odmocnina, sinus, cosinus, logaritmus apod.
3. Booleovské operace – například AND, OR, NOT apod.

4. Podmíněné logické operace – například IF, IF-ELSE apod.
5. Funkce zajišťující opakované provádění kódu – například cykly typu while apod.
6. Specifické funkce přizpůsobené zkoumanému problému. [9]

Při definování množin funkcí a terminálů lze kombinovat různé z výše uvedených možností, avšak je důležité zajistit splnění určitých požadavků, aby výsledný program mohl fungovat správně. [10]

Požadavek uzavřenosti (closure)

Při plnění požadavku uzavřenosti je nutné zajistit, aby každá funkce z množiny funkcí byla schopna za každý svůj argument přijmout libovolnou hodnotu nebo datový typ, který je buď výsledkem kterékoli funkce z množiny funkcí, nebo samotným terminálem z množiny terminálů. [10]

Jedním z možných způsobů, jak tento požadavek splnit, je zachování konzistence typů, tedy zajištění, že všechny funkce přijímají i vracejí hodnoty stejného typu. Druhou možností je přizpůsobení operací křížení a mutace tak, aby respektovaly typy. [9]

Dále je nutné zajistit bezpečné vyhodnocení funkcí, aby nenastal stav, který způsobí chybu, typicky například dělení nulou. K vyřešení tohoto problému se obvykle používají chráněné verze funkcí, které v případě chyby vracejí výchozí hodnotu. [9]

Požadavek postačitelnosti (sufficiency)

Požadavek postačitelnosti vyjadřuje potřebu definování množin funkcí a terminálů tak, aby bylo zaručeno, že s jejich využitím lze nalézt řešení daného problému. Ačkoli někdy může být tento požadavek snadno splnitelný, v jiných případech vyžaduje hlubší porozumění problému a znalost teoretických poznatků či předchozích zkušeností, které by naznačovaly, jaké funkce a terminály jsou nezbytné pro nalezení řešení problému. [9, 10]

1.2.3. Inicializace počáteční populace

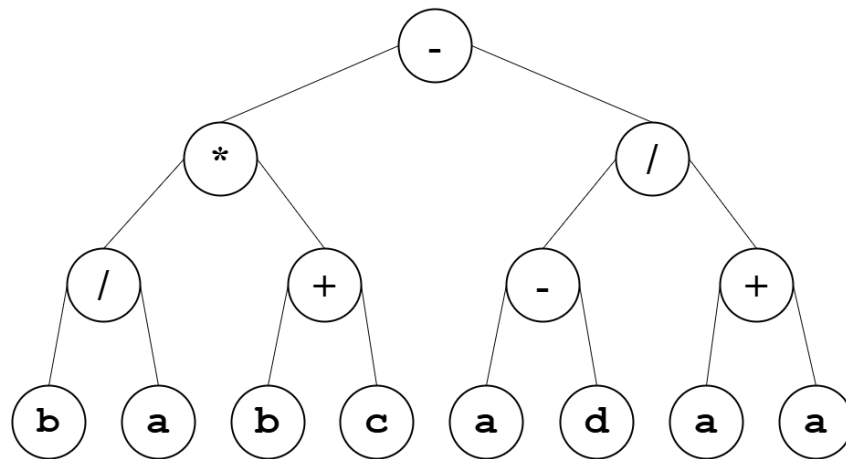
Jak již bylo zmíněno, jedinci v populaci jsou reprezentováni ve formě syntaktických stromů skládajících se z funkcí a terminálů vybraných z uživatelem definovaných množin. Jedinci počáteční populace jsou generováni náhodně, přičemž maximální hloubku stromu určuje uživatel. Existuje několik metod inicializace počáteční populace. Mezi základní z nich patří úplná (full) metoda, růstová (grow) metoda či metoda „půl napůl“ (ramped half-and-half). [10]

Metoda úplná (full)

Při generování syntaktických stromů metodou úplnou dochází k náhodnému výběru funkce z množiny funkcí pro každý vnitřní uzel stromu. Pro koncové uzly (listy), které se vždy nacházejí v maximální hloubce požadované uživatelem, jsou vybírány terminály z množiny terminálů. Výsledkem jsou syntaktické stromy s maximálním možným počtem prvků. [14, 15]

Vygenerujeme-li tedy dva syntaktické stromy se stejnou maximální hloubkou, budou obsahovat stejný počet prvků. Pokud by maximální možná hloubka byla rovna jedné, pak by vygenerovaný strom obsahoval pouze jeden uzel – terminál. [14, 15]

Pro ilustraci je na následujícím obrázku uveden příklad náhodně vygenerovaného syntaktického stromu pomocí metody úplné s maximální hloubkou 3:

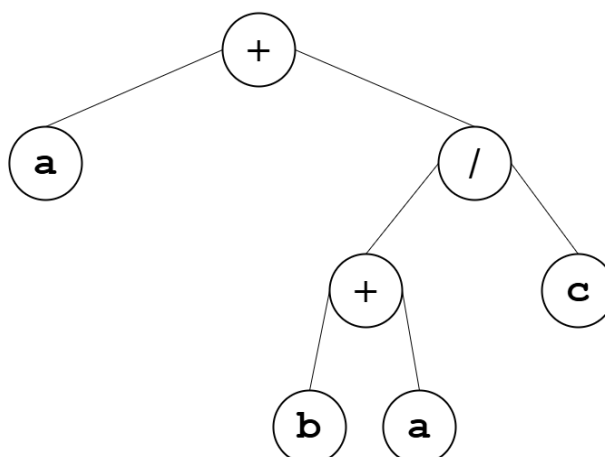


Obrázek 5: Náhodně vygenerovaný syntaktický strom pomocí metody úplné [10]

Metoda růstová (grow)

Růstová metoda se od úplné metody liší tím, že umožňuje ukončení větví v různých hloubkách stromu, nejen v maximální hloubce. Při generování počáteční populace touto metodou je u každého uzlu náhodně rozhodnuto, zda se bude jednat o terminál či funkci. V případě terminálu je jeden náhodně vybrán z množiny terminálů a větev stromu zde končí. V opačném případě je náhodně vybrána jedna funkce z množiny funkcí a danému prvku jsou přiřazeni potomci, jejichž počet odpovídá aritě funkce. Pro každého potomka se proces opakuje. Pokud se potomek nachází v maximální hloubce, je uzlu přiřazen náhodný terminál z množiny terminálů, čímž je zastaven růst větve. Díky tomu vznikají v populaci syntaktické stromy různých hloubek a velikostí, což přispívá k celkové rozmanitosti populace. [14, 15]

Pro ilustraci je na následujícím obrázku uveden příklad náhodně vygenerovaného syntaktického stromu pomocí metody růstové s maximální hloubkou 3:



Obrázek 6: Náhodně vygenerovaný syntaktický strom pomocí metody růstové [10]

Metoda ramped half-and-half

Metoda ramped half-and-half vznikla kombinací úplné a růstové metody. Polovina populace je vygenerována dle metody úplné, druhá polovina podle metody růstové. Tím vzniká populace jedinců s různou strukturou a velikostí, což napomáhá zachování větší variability ve výchozí populaci. [15]

1.2.4 Ohodnocení jedinců

Tento krok evolučního algoritmu spočívá v přiřazení fitness hodnoty každému jedinci populace. Tato hodnota odráží kvalitu jedince a napomáhá při výběru zdatnějších jedinců pro reprodukci a vznik nové generace. K určení fitness jsou využívány různé ohodnocující funkce, které je třeba navrhnout specificky pro konkrétní problém. [10]

Vzhledem ke stromové reprezentaci jedinců v genetickém programování je k ohodnocení jedinců nutné projít celý strom. To se obvykle provádí rekurzivním průchodem stromu do hloubky, přičemž vyhodnocení každého uzlu je odloženo do chvíle, kdy jsou známy hodnoty jeho potomků, pokud nějaké má. [9]

1.2.5 Selektce

Selekční mechanismy slouží k výběru jedinců ze stávající populace, kteří se budou podílet na vytváření svých potomků, tedy jedinců nové generace. Jedinci jsou vybíráni na základě jejich fitness ohodnocení – čím vyšší kvalitu jedinec vykazuje, tím je větší pravděpodobnost, že bude vybrán. I tady však platí, že existuje několik různých metod selektce. [16]

Ruletová selekce

Ruletová selekce je jedním z nejjednodušších a nejčastěji používaných mechanismů výběru v evolučních algoritmech. Její princip lze přirovnat otáčení ruletového kola, kde velikost jednotlivých sektorů odpovídá fitness hodnotě jedince. Čím lepší je fitness jedince, tím větší sektor mu náleží. Výběr pak probíhá náhodným určením bodu na kole a vybráním jedince, kterému odpovídá sektor, v němž se bod nachází. Tímto způsobem jsou upřednostňováni kvalitnější jedinci, kteří mají vyšší pravděpodobnost, že budou vybráni, zároveň však stále existuje možnost výběru méně vhodných jedinců, což napomáhá udržení diverzity v populaci. [17, 18]

Turnajová selekce

Turnajová selekce je selekční mechanismus inspirovaný přírodními principy soupeření jedinců mezi sebou o možnost reprodukce. Tento mechanismus spočívá v náhodném vybrání určitého počtu jedinců populace a porovnání jejich fitness hodnot. Jedinec s nejlepší fitness hodnotou vyhrává a je zařazen do výběru. [10, 18]

Čím větší je počet účastníků turnaje, tím větší je pravděpodobnost výběru jedinců s vyšší fitness hodnotou, což urychluje hledání řešení. Pokud je však tento počet příliš vysoký, může dojít k předčasnému nalezení nevhodného (suboptimálního) řešení. Některé varianty turnajové selekce proto ponechávají určitou šanci i méně zdatným účastníkům turnaje, čímž podporují genetickou rozmanitost populace. [10, 18]

1.2.6 Křížení

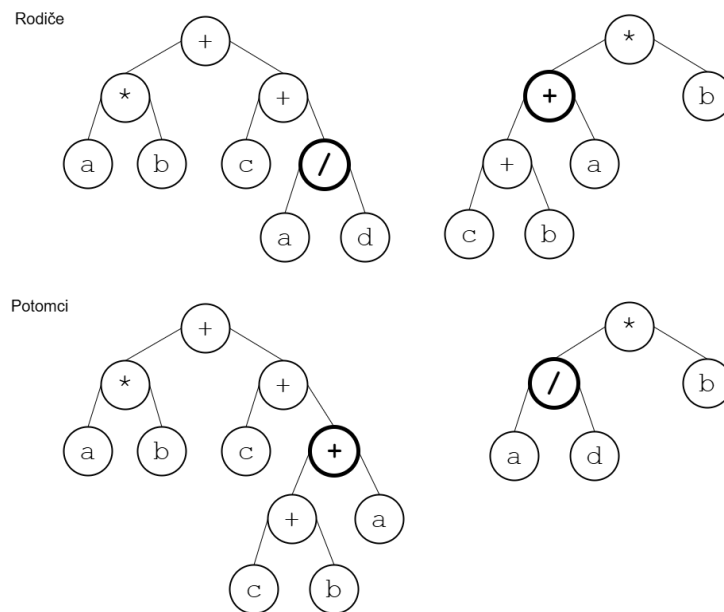
Křížení je primárním operátorem genetického programování sloužící k vytvoření nové generace. Jedinci zvoleni k vytváření nových potomků jsou nejprve náhodně spárováni do rodičovských dvojic, ze kterých následnou kombinací rodičů vzniknou potomci. [10, 19] Vzhledem k tomu, že genetické programování pracuje s jedinci reprezentovanými syntaktickými stromy proměnlivé délky a struktury, bylo nutné vyvinout speciální metody křížení odlišné od těch používaných v jiných evolučních algoritmech. [9, 10] Bylo provedeno několik výzkumů zaměřených na zefektivnění křížení což vedlo ke vzniku několika metod, z nichž některé zde budou představeny. [19]

Křížení podstromů

Jednou z nejčastěji používaných metod křížení v genetickém programování je křížení podstromů. Při užití tohoto přístupu se nejprve v každém rodičovském stromě náhodně vybere

jeden uzel označovaný jako bod křížení. Následně se podstromy s těmito uzly jako kořeny prohodí, čímž vznikají noví potomci. [9]

Příklad takového křížení je uveden na následujícím obrázku, kde jsou tučně zvýrazněny body křížení:



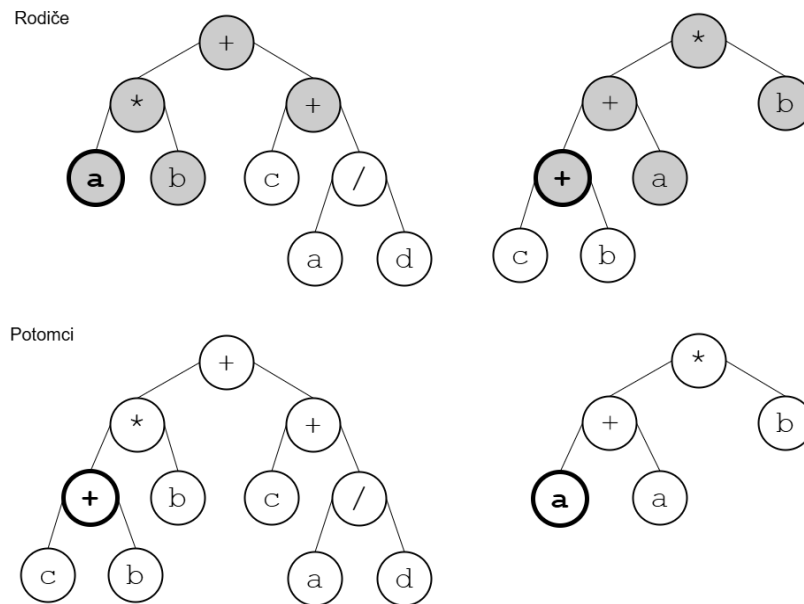
Obrázek 7: Vizualizace křížení podstromů (zdroj: vlastní)

Aby bylo možné zachovat původní stromy rodičů netknuté pro možné další křížení, pracuje se s jejich kopiemi. Tento typ křížení může generovat buď jednoho potomka, nebo oba nově vzniklé jedince. [9] Existuje také varianta, která preferuje výměnu vnitřních uzlů, přičemž obvykle platí poměr devadesát procent pro výběr vnitřního uzlu jako bodu křížení a deset procent pro výběr listu stromu. [19]

Jednobodové křížení

Jednobodové křížení spočívá ve výměně dvou podstromů, přičemž body křížení se nacházejí v každém rodiči na stejném místě. Avšak genetické programování umožňuje jedincům mít odlišnou délku a tvar, proto je nutné nejprve analyzovat rodičovské stromy od kořenových uzlů a určit oblast, ve které mají oba stromy stejný tvar. Bod křížení pak může být vybírán pouze z této oblasti. [9]

Příklad jednobodového křížení je uveden na následujícím obrázku, kde jsou šedivě vyznačeny uzly nacházející se v oblasti se stejnou strukturou obou rodičů a vybrané body křížení jsou zvýrazněny tučně:

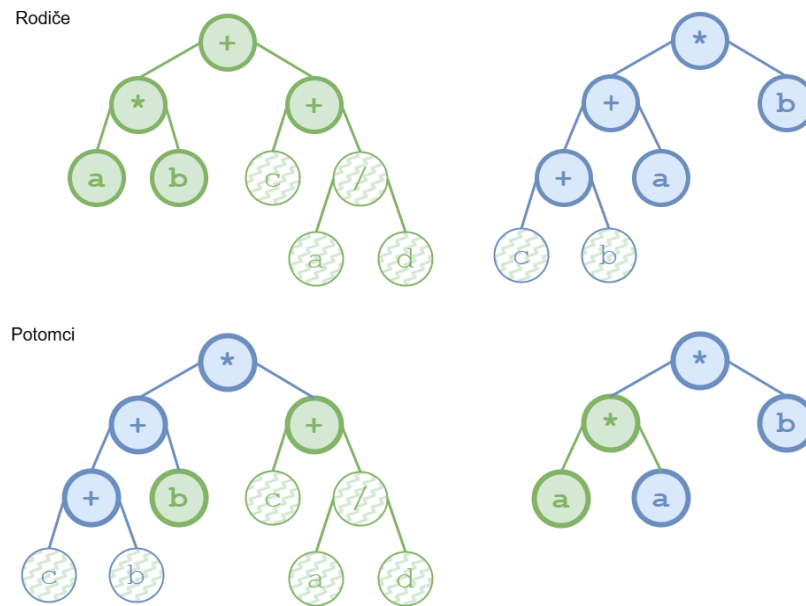


Obrázek 8: Vizualizace jednobodového křížení (zdroj: vlastní)

Rovnoměrné křížení

Rovnoměrné křížení, někdy též označované jako uniformní křížení, využívá podstatně odlišný postup, než jaký byl použit v předešlých případech. Potomek je vytvořen tak, že je nejprve vytyčena společná oblast rodičovských stromů, následně jsou procházeny všechny uzly společné oblasti rodičovských stromů a u každého je zvoleno, zda bude potomek obsahovat daný uzel od prvního či druhého rodiče. Pravděpodobnost na získání uzlu od jednoho rodiče je rovna padesáti procentům. Pokud je na pozici koncového uzlu společné oblasti zvolen uzel obsahující funkci, potomek zdědí celý podstrom. [9]

Příklad rovnoměrného křížení je znázorněn na následujícím obrázku, kde jsou tučně vyznačeny uzly spadající do společné oblasti rodičů a jednotlivé uzly potomků jsou barevně rozlišeny podle rodiče, od kterého byly převzaty:



Obrázek 9: Vizualizace rovnoměrného křížení (zdroj: vlastní)

1.2.7 Mutace

Mutace slouží k zabránění předčasné konvergence populace a ztrátě potenciálně užitečných vlastností tím, že do procesu vnáší náhodnou změnu. Na rozdíl od křížení není vždy považována za nezbytnou, proto je v genetických programech používána s nízkým procentuálním zastoupením – obvykle mutuje maximálně 10 % všech jedinců v populaci – nebo je vynechávána úplně. [9, 10]

Mutace podstromu

Metoda nazývaná mutace podstromu je nejběžněji používanou formou mutace. Jedná se o náhodný výběr uzlu ve stromu, označovaného jako bod mutace, a nahrazení celého podstromu, který se pod ním nachází, jiným nově vygenerovaným podstromem. V některých implementacích se tato metoda realizuje jako křížení s náhodně vygenerovaným stromem. [9]

Bodová mutace

Druhou často používanou metodou je bodová mutace. Ta je založena na náhodném výběru uzlu ve stromu a nahrazením jeho funkce či terminálem jinou funkcí či terminálem se stejnou aritou. Pokud žádná taková funkce neexistuje, uzel zůstává beze změny. Na rozdíl od mutace podstromu lze bodovou mutaci aplikovat na jeden strom opakovaně. Obvykle se strom prochází uzel po uzlu a s určitou pravděpodobností je rozhodováno, zda bude zmutován. [9]

1.2.8 Ukončovací kritérium

Ukončovací kritérium udává, kdy bude běh genetického programu ukončen. Ideálním případem je nalezení optimálního nebo dostatečně uspokojivého řešení, což lze vyjádřit

například dosažením určité hodnoty fitness. Genetické programování však nemůže zaručit nalezení takového řešení, proto by mohl nastat stav, kdy by program běžel nekonečně dlouho. Proto je nutné definovat dodatečnou podmínku, která zaručí jeho ukončení. Mezi běžně používaná kritéria patří dosažení maximálního počtu generací, překročení stanoveného výpočetního času či například nadměrné snížení diverzity populace. Tato kritéria mohou doplňovat primární podmínku nalezení optimálního řešení, nebo mohou být použita samostatně. [20]

2 EXPERIMENTÁLNÍ ČÁST

Tato kapitola se zaměřuje na popis experimentů provedených za účelem analýzy vlivu různých ztrátových funkcí při řešení problému symbolické regrese pomocí genetického programování. Experimenty se věnují konkrétně symbolické regresi s jednou vstupní proměnnou, kde je cílem nalézt funkční závislost $y = f(x)$. V rámci experimentálního zkoumání bylo testováno pět běžně používaných ztrátových funkcí v této oblasti – MAE, MSE, RMSE, Huberova ztrátová funkce a využití GDL spolu s MAE. Charakteristika těchto funkcí byla uvedena v kapitole 1.1.1 Ztrátové funkce.

2.1 Implementace genetického programování

Pro provedení experimentů byl vytvořen vlastní program v prostředí .NET využívající principy genetického programování. Tento program, implementující vybrané metody genetického programování, byl použit jednotně pro ve všech experimentech, včetně stejného nastavení volitelných parametrů s výjimkou ohodnocující funkce, počtu jedinců populace a maximálního počtu generací, které se u jednotlivých konfigurací popsaných níže mění.

Program nejprve vygeneruje testovací data ve formě dvojic hodnot – souřadnic x a y – odpovídající předem definované testovací funkci, se kterými následně pracuje. Úkolem programu je na základě těchto dat nalézt funkci, která jim nejlépe odpovídá, aniž by znal tvar testovací funkce.

Program poté provádí na vygenerovaných datech symbolickou regresi za pomoci genetického programování. Po dokončení každé evoluce je zaznamenán nejlepší nalezený jedinec v rámci celé populace i informace o něm, přičemž je vždy jako finální ohodnocení vypočítána jeho hodnota MAE – bez ohledu na to, jaká ohodnocující funkce byla pro selekci v dané evoluci použita. Tato hodnota poté slouží jako jednotné měřítko pro porovnání kvality výsledných řešení mezi různými ztrátovými funkcemi použitými během experimentů.

2.1.1 Testovací funkce

Pro účely experimentů byla zvolena sada testovacích funkcí označovaná jako Nguyen benchmark. Tento soubor funkcí patří mezi často používané standardní benchmarky pro testování algoritmů řešících problém symbolické regrese a je dostupný například na stránce HeuristicLab [21]. V rámci experimentálního zkoumání bylo využito pouze prvních osmi testovacích funkcí zabývajících se symbolickou regresi s jednou vstupní proměnnou:

- *Nguyen F1* $= x^3 + x^2 + x$

- $Nguyen F2 = x^4 + x^3 + x^2 + x$
- $Nguyen F3 = x^5 + x^4 + x^3 + x^2 + x$
- $Nguyen F4 = x^6 + x^5 + x^4 + x^3 + x^2 + x$
- $Nguyen F5 = \sin(x^2) * \cos(x) - 1$
- $Nguyen F6 = \sin(x) + \sin(x + x^2)$
- $Nguyen F7 = \log(x + 1) + \log(x^2 + 1)$
- $Nguyen F8 = \sqrt{x}$

Testovací data jsou poté získána tak, že v daném rozsahu je vygenerováno 500 hodnot představujících souřadnice x a ke každému bodu x je následně dle dané testovací funkce dopočítána hodnota y .

2.1.2 Použité metody a nastavení parametrů

Pro vytvoření programu, který byl použit pro spuštění všech experimentů, bylo třeba vybrat vhodné metody genetického programování, například metody inicializace počáteční populace, selekce, křížení či mutace. Dále bylo nutné definovat množiny terminálů a funkcí a vhodně nastavit parametry algoritmu. Tato kapitola popisuje vybrané techniky, které zůstaly po celou dobu běhu experimentů neměnné.

Množina terminálů zahrnovala konstantní hodnoty 1, 2, 3, 5, 7 a 9 a vstupní proměnnou x . Množina funkcí se skládala ze čtyř základních matematických operátorů: sčítání, odčítání, násobení a dělení. Inicializace počáteční populace pak byla realizovaná pomocí metody růstové. Kořen každého stromu generovaného při inicializaci počáteční populace vždy obsahoval funkci, zatímco ostatní potomci měli 60% šanci pro výběr funkce a 40% šanci pro výběr terminálu. Maximální hloubka stromu byla omezena na 8 úrovní.

Selekce probíhala dle turnajové strategie výběru jedince, přičemž velikost turnaje byla nastavena na 2 jedince a pravděpodobnost pro výhru slabšího jedince se rovnala 10 %. Dále byla zvolena metoda křížení podstromů a metoda mutace podstromů s 10% pravděpodobností mutace nově vzniklého potomka.

2.2 Experimenty

Tato kapitola popisuje samotné experimenty, které byly rozděleny do několika sad testů, dále označovaných jako konfigurace. Jednotlivé konfigurace se liší v nastavení počtu jedinců v populaci a maximálního počtu generací v rámci jedné evoluce. Snížením těchto parametrů

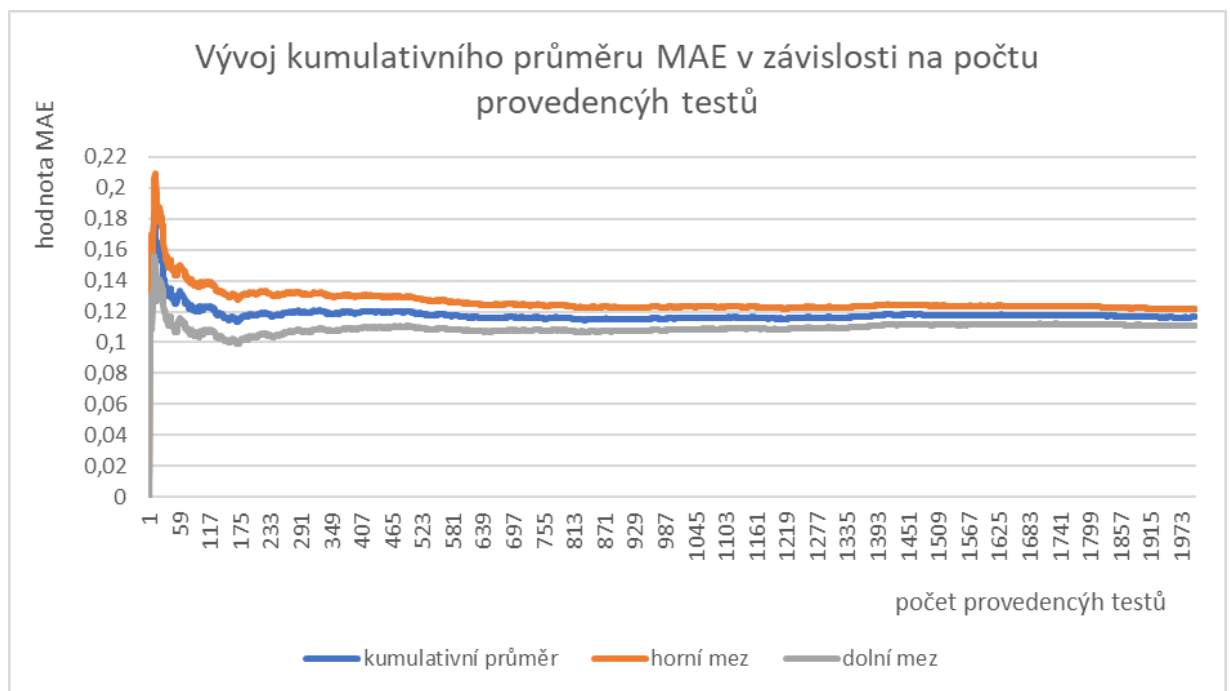
došlo nejen ke zkrácení výpočetního času, ale také ke zvýšení nároků na schopnost algoritmu nalézt optimální řešení v omezených podmínkách.

V rámci každé konfigurace byly provedeny experimenty se všemi osmi testovacími funkcemi. Pro každou z těchto funkcí bylo spuštěno 1000 testů pro každou z pěti zvolených ztrátových funkcí (MAE, MSE, RMSE, Huberova ztrátová funkce a využití GDL ve spojení s MAE). Celkem tedy bylo v rámci jedné konfigurace provedeno 40 000 testů.

2.2.1 Experiment ověřující stabilitu výsledků

Aby bylo možné ověřit, že 1000 opakování testu představuje dostatečný počet pro dosažení stabilních a spolehlivých výsledků, byl nejprve proveden experiment zaměřený na ustálení fitness hodnoty.

V tomto experimentu bylo provedeno 2000 testů při stejném nastavení (50 jedinců v populaci a maximálně 50 generací) s využitím pouze ztrátová funkce MAE. Na konci každého testu byla zaznamenána hodnota MAE nejúspěšnějšího nalezeného jedince. Výsledky byly poté zpracovány do následujícího grafu znázorňující vývoj kumulativního průměru MAE v závislosti na počtu uskutečněných testů.



Obrázek 10: Graf vývoje kumulativního průměru v závislosti na počtu provedených testů (zdroj: vlastní)

Na ose x je zobrazen počet provedených testů, na ose y pak průměrná hodnota MAE. Modrá linie reprezentuje kumulativní průměr hodnoty MAE, oranžová značí horní mez a šedá dolní mez intervalu spolehlivosti.

Z grafu je patrné, že se průměrná hodnota MAE stabilizuje přibližně po 800 provedených testech, což potvrzuje, že počet 1000 opakování stejného testu je dostatečný. Navýšení počtu testů by vedlo jen k minimálnímu zpřesnění výsledků, avšak za cenu výrazného nárůstu výpočetního času.

2.2.2 Experimenty první konfigurace

Tato kapitola popisuje sadu osmi experimentů, z nichž každý odpovídá jedné testovací funkci v rámci Nguyen benchmarku, provedených se stejným nastavením parametrů. Nastavení odpovídá popisu v kapitole 2.1.2 Použité metody a nastavení parametrů doplněné o nastavení počtu jedinců každé populace na 50 a nastavení maximálního možného počtu generací také na 50.

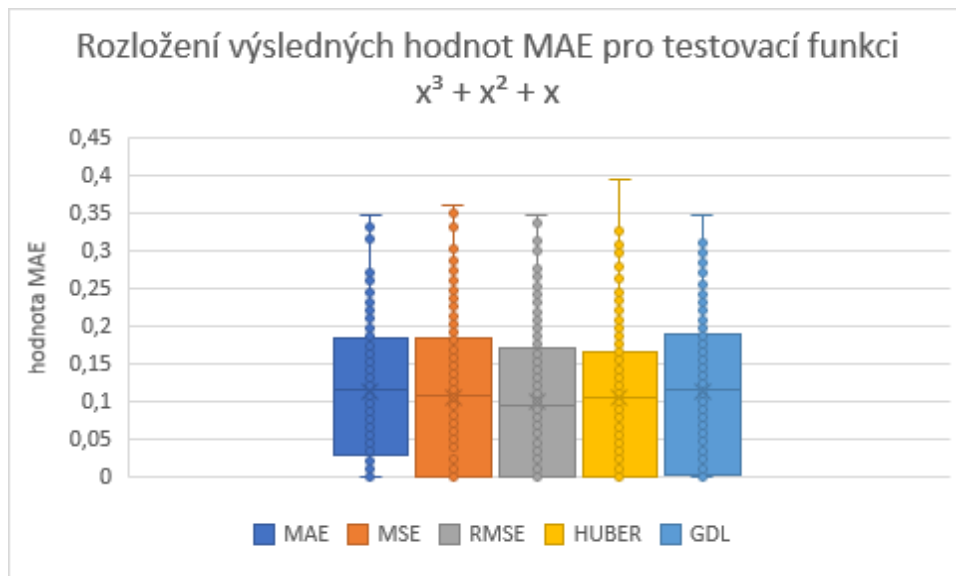
Experiment první konfigurace s testovací funkcí Nguyen F1

První experiment dané konfigurace byl založen na testovacích datech vygenerovaných na základě testovací funkce ve tvaru:

$$\text{Nguyen F1} = x^3 + x^2 + x$$

V rámci tohoto experimentu bylo provedeno 5000 testů, přičemž pro každou z pěti zkoumaných ztrátových funkcí (MAE, MSE, RMSE, Huberova ztrátová funkce a GDL ve spojení s MAE) bylo spuštěno 1000 opakování. U každého běhu byla výsledná fitness hodnota přepočtena na MAE, aby bylo možné objektivně porovnat kvalitu nalezených řešení mezi jednotlivými ztrátovými funkcemi.

Výsledky byly následně vizualizovány prostřednictvím krabicového diagramu, který znázorňuje rozložení hodnot MAE pro každou ztrátovou funkci. Každý sloupec v grafu představuje výsledek 1000 testů s danou funkcí. Graf tedy umožňuje přehledně porovnat, jak dobře si jednotlivé ztrátové funkce vedly.



Obrázek 11: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F1 při první konfiguraci (zdroj: vlastní)

Z grafu je patrné, že první kvartil (hranice, pod kterou leží 25 % hodnot) u všech testovaných ztrátových funkcí v tomto experimentu, s výjimkou MAE, dosahuje nulové hodnoty, což značí že algoritmus byl v mnoha případech schopen najít řešení s nulovou chybou. To ukazuje na vysokou úspěšnost evolučního algoritmu při hledání optimálních řešení symbolické regrese.

Z pohledu porovnání mediánů v rámci toho experimentu je nejuspěšnější ztrátovou funkcí RMSE, následovaná Huberovou ztrátovou funkcí, která má také nejnižše položený třetí kvartil, což naznačuje větší konzistenci ve většině případech. Na druhou stranu má Huberova ztrátová funkce delší horní fousek, což může naznačovat vyšší variabilitu v oblasti vyšších hodnot a možnou asymetrii rozdělení. Tyto hodnoty však nemají ve výsledném hodnocení zásadní vliv na celkové hodnocení.

Méně úspěšnými ztrátovými funkcemi tohoto experimentu jsou MAE, jejíž první kvartil nedosahuje nulové hodnoty jako je tomu u ostatních ztrátových funkcí, a především kombinace GDL spojené s MAE, která vykazuje nejen nejvýše položený medián, ale i největší rozptyl hodnot.

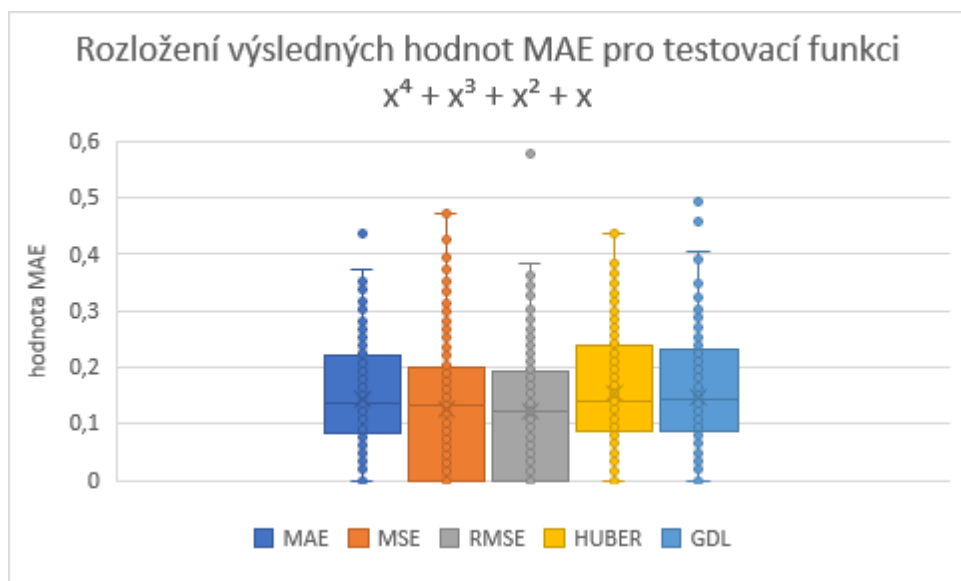
Je však důležité zdůraznit, že navzdory rozdílům mezi jednotlivými ztrátovými funkcemi dosahují výsledky v 75 % případů všech testů hodnoty MAE nižší než 0,2. To potvrzuje vysokou míru úspěšnosti evolučního algoritmu napříč všemi použitými ztrátovými funkcemi.

Experiment první konfigurace s testovací funkcí Nguyen F2

Testovací data použitá ve druhém experimentu první konfigurace byla vygenerována na základě testovací funkce ve tvaru:

$$\text{Nguyen F2} = x^4 + x^3 + x^2 + x$$

To je zároveň jediný rozdíl oproti předešlému experimentu. Stejně jako v předešlém experimentu i v tomto případě bylo provedeno 5000 testů – 1000 testů pro každou z pěti testovaných ztrátových funkcí. Výsledky těchto testů, opět převedené na MAE hodnoty, byly také zpracovány do formy krabicového diagramu, ve kterém lze porovnat úspěšnost jednotlivých ztrátových funkcí.



Obrázek 12: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F2 při první konfiguraci (zdroj: vlastní)

Z diagramu je patrné, že rozložení hodnot jednotlivých ztrátových funkcí je o něco vyšší než v minulém experimentu, zejména funkcí MAE, GDL a Huberovy ztrátové funkce, u kterých se první kvartil už nerovná nulové hodnotě a třetí kvartil se nachází nad hodnotou 0,2.

RMSE a MSE prokazují v rámci tohoto experimentu lepší výsledky, což dokládají nejen nulové hodnoty jejich prvních kvartilů, ale jejich nižší mediány i průměrné hodnoty MAE. Z hlediska porovnání mediánů je tedy nejúspěšnější ztrátovou funkcí tohoto experimentu RMSE následovaná MSE, která má mimo jiné také nejdelší úsek odlehlých hodnot, dále jsou v pořadí funkce MAE, Huberova ztrátová funkce a GDL.

V tomto experimentu jsou tedy patrnější rozdíly mezi ztrátovými funkcemi RMSE a MSE oproti MAE, Huberově ztrátové funkci a GDL kombinované s MAE, ačkoli naprostá většina

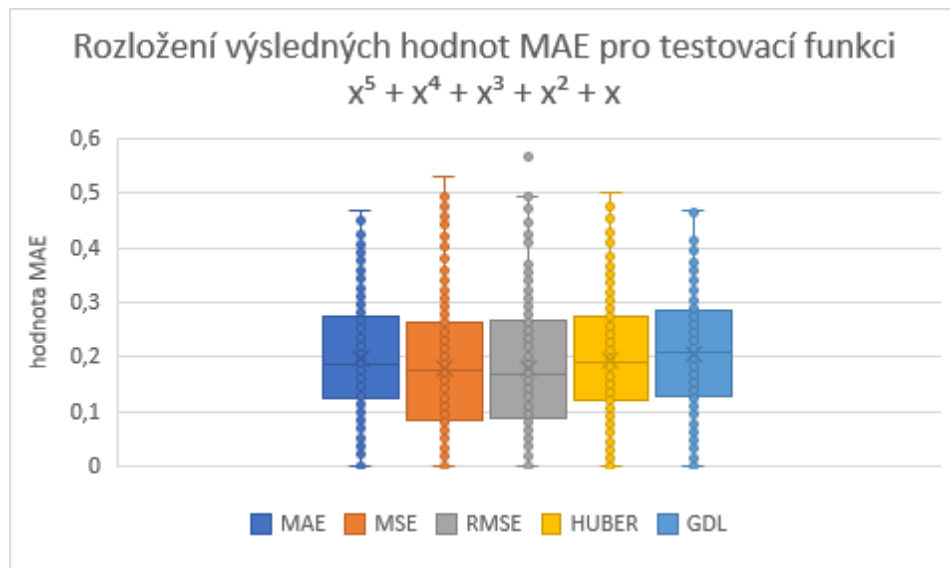
výsledků u všech funkcí zůstává pod hodnotou 0,3, což značí vysokou úspěšnost všech testovaných ztrátových funkcí.

Experiment první konfigurace s testovací funkcí Nguyen F3

Stejně jako v předchozích experimentech bylo i tentokrát provedeno 1000 testů pro každou z pěti testovaných ztrátových funkcí. Jediným zásadním rozdílem je opět použití testovací funkce, na jejímž základě byla generována testovací data:

$$\text{Nguyen F3} = x^5 + x^4 + x^3 + x^2 + x$$

Rozložení výsledných fitness hodnot převedených na hodnoty MAE pro každou ztrátovou funkci bylo opět znázorněno formou krabicového grafu.



Obrázek 13: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F3 při první konfiguraci (zdroj: vlastní)

Z grafu je patrné, že ztrátové funkce MSE a RMSE v tomto experimentu opět dosahují o něco lepších výsledků než ostatní ztrátové funkce, což dokazují nejen jejich první kvartily, které sice již nedosahují nulové hodnoty, ale jako jediné zůstávají pod hranicí 0,1, nýbrž i jejich nižší mediány a průměrné hodnoty MAE.

Třetí nejlepší funkcí z pohledu mediánu je MAE následovaná Huberovou ztrátovou funkcí. Nejhorších výsledků v tomto experimentu dosáhla GDL zkombinovaná s MAE, jejíž medián jako jediný přesahuje hodnotu 0,2. Stejně tak její první i třetí kvartil vykazují vyšší hodnoty ve srovnání s ostatními ztrátovými funkcemi.

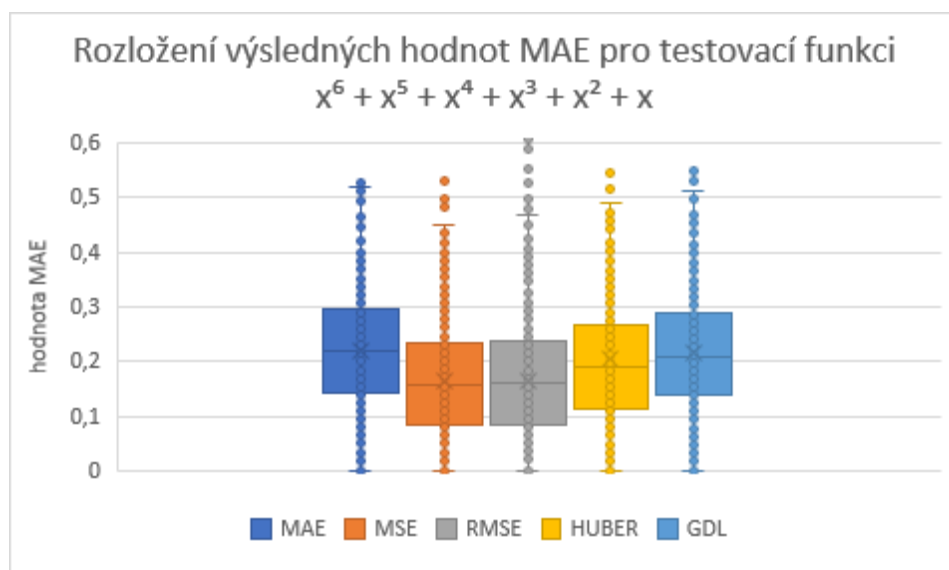
Jak lze vidět z grafu, v tomto experimentu už žádný první kvartil neodpovídá nulové hodnotě, jako tomu bylo v předešlých dvou experimentech. Jedním z důvodů může být vyšší složitost funkce, která byla použita k vygenerování testovacích dat. Na druhou stranu se všechny třetí kvartily stále drží pod hranicí odpovídající 0,3 hodnoty MAE.

Experiment první konfigurace s testovací funkcí Nguyen F4

V rámci tohoto experimentu bylo opět provedeno 1000 testů pro každou z vybraných ztrátových funkcí, celkem tedy 5000 testů. Jediným rozdílem oproti předchozímu experimentu je použití o něco složitější testovací funkce pro vygenerování testovacích dat:

$$Nguyen\ F4 = x^6 + x^5 + x^4 + x^3 + x^2 + x$$

Po provedení všech testů byly jejich výsledky zpracovány do následujícího krabicového grafu ukazujícího rozložení MAE hodnot nejlepších jedinců z jednotlivých testů rozříděných dle použité ztrátové funkce:



Obrázek 14: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F4 při první konfiguraci (zdroj: vlastní)

Stejně jako v předešlých experimentech lze z grafu pozorovat, že MSE a RMSE dosahovaly celkově lepších výsledků než zbylé testované ztrátové funkce. Důkazem toho jsou nejen nižší první kvartily, které se stále jako jediné drží pod hranicí 0,1, ale i jejich nižší hodnoty třetích kvartilů, mediánů i průměrů hodnoty MAE. Na základě výsledků lze tyto dvě funkce i přes drobné rozdíly v tomto experimentu považovat za vyrovnané, tedy stejně úspěšné.

Oproti tomu mezi ostatními testovanými funkcemi jsou patrné výraznější rozdíly. Z hlediska hodnot kvartilů i mediánů lze jako třetí nejúspěšnější funkci experimentu považovat

Huberovu ztrátovou funkci následovanou GDL. Nejhůře v tomto experimentu dopadla MAE, jejíž třetí kvartil se drží těsně pod hodnotou 0,3, což naznačuje zatím nejhorší pozorovanou výkonnost.

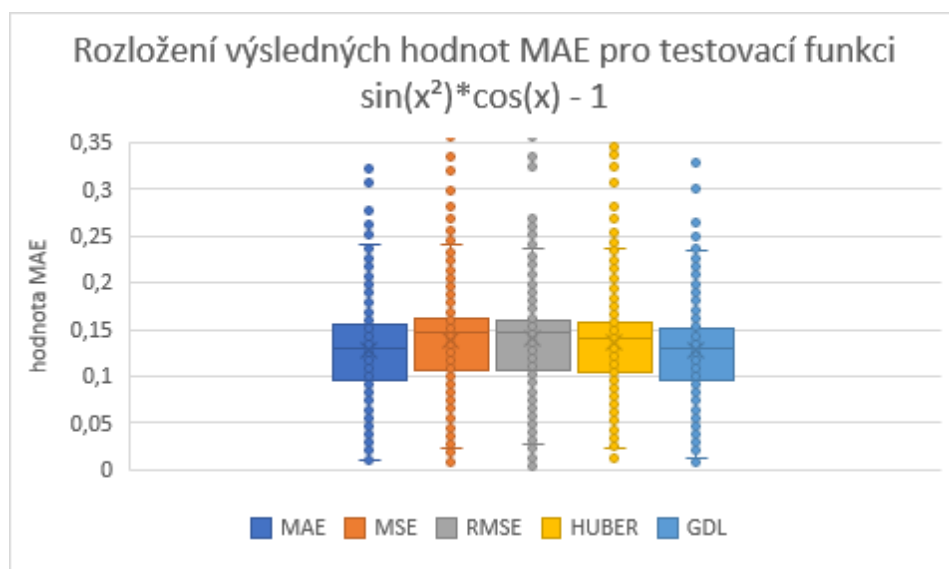
Experiment první konfigurace s testovací funkcí Nguyen F5

I v tomto experimentu bylo provedeno 5000 testů, tedy 1000 testů pro každou z vybraných ztrátových funkcí. Oproti předešlým experimentům však byla k vygenerování testovacích využita poměrně odlišná testovací funkce:

$$Nguyen\ F5 = \sin(x^2) * \cos(x) - 1$$

Tato funkce již nepracuje pouze se sčítáním mocnin proměnné x , ale zahrnuje i goniometrické funkce. Program využitý pro experimenty má ale k dispozici pouze základní aritmetické operace zahrnující pouze sčítání, odčítání, násobení a dělení. Testovací data však byla vygenerována pro hodnoty x v rozsahu od -1 do 1, je tedy reálné v tomto omezeném rozsahu dosáhnout relativně přesných výsledků za využití pouze dostupných operací.

Výsledky, opět ve formě fitness hodnoty převedené na MAE nejlepšího jedince z každého testu, jsou znázorněny v následujícím krabicovém diagramu, kde jsou rozděleny dle použité ztrátové funkce.



Obrázek 15: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F5 při první konfiguraci (zdroj: vlastní)

Z grafu je na první pohled patrné, že rozložení výsledků všech testovaných ztrátových funkcí je kompaktnější než v předešlých experimentech, jinými slovy rozdíl mezi první a třetím kvantilem je menší.

Také z pohledu srovnání jednotlivých funkcí došlo v tomto experimentu k patrné změně jejich pořadí dle úspěšnosti. Nejlepších výsledků dosáhla ztrátová funkce GDL, což dokládá nejen nejnižší medián, ale také nejnižší hodnoty prvního i třetího kvartilu. Hned za ní následuje MAE s podobnými výsledky.

Naopak MSE a RMSE, které se v předchozích experimentech umísťovaly na nejlepších pozicích, vykazují tentokrát nejvyšší hodnoty mediánů. Jejich výsledky jsou navzájem téměř shodné, přičemž Huberova ztrátová funkce se od nich odlišuje pouze mírně nižší hodnotou mediánu.

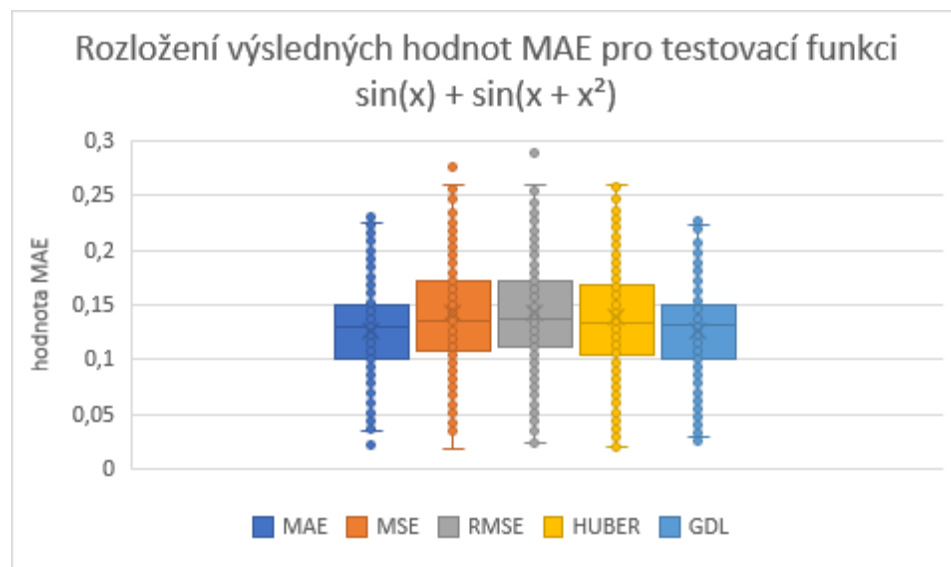
Tato změna v pořadí může být důsledkem odlišné podoby testovací funkce, která jako první v sérii experimentů obsahuje i goniometrické prvky.

Experiment první konfigurace s testovací funkcí Nguyen F6

V rámci tohoto experimentu, bylo provedeno 1000 testů pro každou z pěti vybraných ztrátových funkcí. Data použitá k testování byla vygenerována za použití následující testovací funkce, která stejně jako v předchozím experimentu využívá goniometrických funkcí:

$$Nguyen\ F6 = \sin(x) + \sin(x + x^2)$$

Výsledky byly zpracovány stejným způsobem jako v předchozích experimentech a vizualizovány pomocí krabicového diagramu:



Obrázek 16: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F6 při první konfiguraci (zdroj: vlastní)

I z tohoto grafu je opět patrná změna pořadí úspěšnosti testovaných ztrátových funkcí oproti výsledkům z prvních čtyř provedených experimentů v rámci této konfigurace. Nejlepší výsledky v tomto případě vykazuje funkce MAE, díky nejnižší hodnotě prvního kvartilu i mediánu. Těsně za ní následuje GDL s velmi podobným rozložením hodnot.

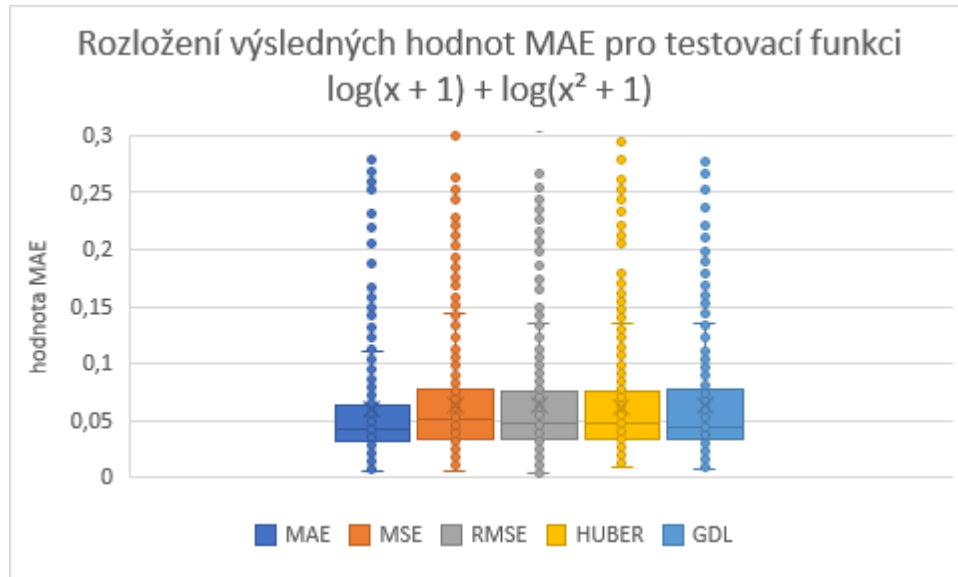
Huberova ztrátová funkce dosahuje o něco horších výsledků, zejména kvůli vyšší hodnotě třetího kvartilu. Hned po ní se umístila MSE následovaná RMSE s nejméně položeným prvním kvantilem.

Experiment první konfigurace s testovací funkcí Nguyen F7

Předposlední experiment v rámci první konfigurace k vygenerování testovacích dat, v rozsahu proměnné x od 0 do 2, využívá následující testovací funkci:

$$Nguyen\ F7 = \log(x + 1) + \log(x^2 + 1)$$

Tato testovací funkce je zároveň jediným významným rozdílem oproti ostatním experimentům této konfigurace. I v tomto experimentu bylo provedeno dohromady 5000 testů – 1000 testů pro každou testovanou ztrátovou funkci. Výsledky těchto testů byly zpracovány do následujícího krabicového grafu.



Obrázek 17: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F7 při první konfiguraci (zdroj: vlastní)

V tomto grafu lze pozorovat několik odlišností nejen z pohledu pořadí ztrátových funkcí oproti první polovině experimentů této konfigurace, ale také oproti předešlým dvěma experimentům. Prvním patrným rozdílem je výrazně nižší rozložení hodnot – zatímco v experimentech využívajících testovacích funkcí Nguyen F5 a F6 se první kvartily všech

ztrátových funkcí nacházely okolo hodnoty 0,1, v tomto experimentu třetí kvartily všech funkcí nedosahují ani hodnoty 0,08. Taková úspěšnost zatím nebyla v jiném experimentu zaznamenána.

Dalším rozdílem je relativní vyrovnanost všech funkcí. Největší odchylku vykazuje MAE, která prokazuje nejvyšší úspěšnost díky nejnižším hodnotám třetího kvartilu a mediánu. Medián s druhou nejnižší hodnotou patří GDL, zatímco nejvyšší medián má MSE, rozdíl mezi nimi však není příliš velký. Výsledky RMSE a Huberovy ztrátové funkce jsou srovnatelné.

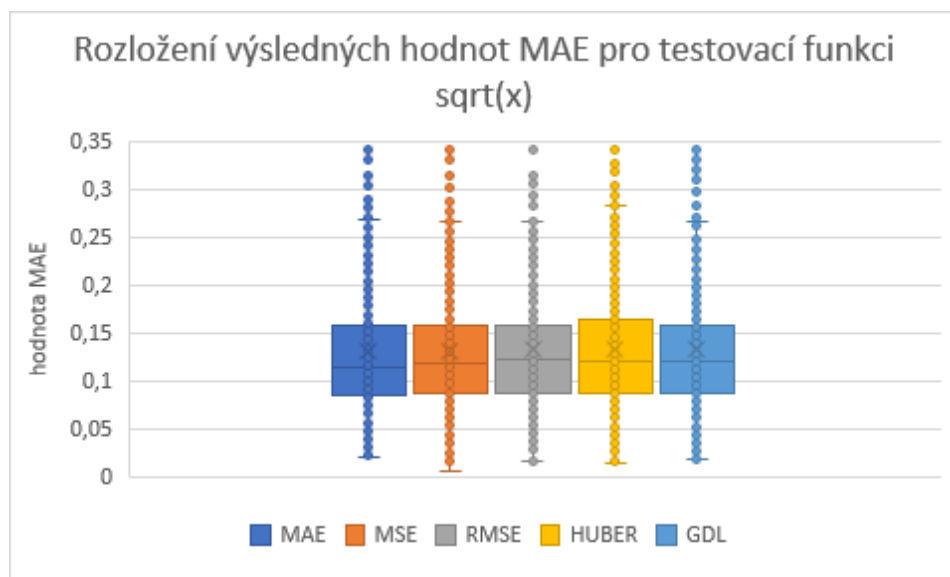
Celkově nižší hodnoty všech výsledků experimentu jsou pravděpodobně způsobeny tvarem testovací funkce použité k vygenerování dat, jejichž složením vznikne méně zakřivený graf, než tomu bylo u předešlých experimentů.

Experiment první konfigurace s testovací funkcí Nguyen F8

Poslední experiment první konfigurace využívá data vygenerovaná pro hodnoty proměnné x v rozsahu od 0 do 4. Tvar funkce, která je k vygenerování dat používána, lze zapsat následovně:

$$Nguyen\ F8 = \sqrt{x}$$

V rámci experimentování bylo provedeno celkem 5000 testů. Pro každou z pěti testovaných funkcí bylo tedy provedeno 1000 testů, jejichž výsledky v podobě hodnoty MAE nejlepšího z jedinců z celého testu byly opět zpracovány do následujícího krabicového grafu.



Obrázek 18: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F8 při první konfiguraci (zdroj: vlastní)

Grafické zpracování výsledků tohoto experimentu naznačuje relativní vyrovnanost všech testovaných ztrátových funkcí, vzhledem k umístění prvních a třetích kvartilů. Jediné výjimky tvoří o něco výše položený třetí kvartil Huberovy ztrátové funkce a nepatrně nižší hodnota prvního kvartilu u MAE.

Z pohledu mediánů prokazuje v tomto experimentu nejlepší výsledky MAE, následovaná MSE. Třetí v pořadí je GDL, těsně po ní Huberova ztrátová funkce. Nejvyšší medián patří RMSE, ačkoli rozdíly mezi mediány nejsou příliš výrazné.

Shrnutí výsledků první konfigurace

V rámci první konfigurace bylo provedeno osm samostatných experimentů, z nichž každý testoval výkonnost pěti různých ztrátových funkcí (MAE, MSE, RMSE, Huberovy ztrátové funkce a GDL spojené s MAE) na jedné konkrétní testovací funkci. Každý experiment zahrnoval 5000 testů, celkově tedy bylo provedeno 40 000 testů. Výsledky byly vyhodnoceny na základě hodnoty MAE nejlepšího jedince každého testu.

Při porovnání výsledků všech experimentů lze pozorovat některé opakující se jevy v úspěšnosti jednotlivých ztrátových funkcí. V první polovině experimentů, což byly experimenty využívající testovací funkce Nguyen F1 až Nguyen F4, které jsou založeny na součtu mocnin proměnné x , vykazovaly výrazně lepší výsledky oproti ostatním ztrátovým funkcím MSE a RMSE.

Naopak ve druhé polovině experimentů, kde již k vygenerování testovacích dat byly použity funkce obsahující i goniometrické členy, prokazovaly lepší výsledky ztrátové funkce MAE a GDL. Huberova ztrátová funkce se ve většině experimentů držela v prostředním pořadí úspěšnosti testovaných funkcí.

Lze tedy předpokládat, že volba vhodné ztrátové funkce pro nalezení řešení problému symbolické regrese pomocí genetického programování do značné míry závisí na charakteru problému, tedy na složitosti cílové funkce. První konfigurace naznačuje, že pro jednodušší cílové funkce může být vhodnější volbou MSE či RMSE, zatímco ve složitějších případech by bylo vhodné použít MAE či GDL. Huberova ztrátová funkce se ukazuje jako stabilní kompromis.

2.2.3 Experimenty druhé konfigurace

V této kapitole je popsáno dalších osm experimentů, které využívaly stejné testovací funkce jako experimenty v rámci první konfigurace. Stejně tak byly opět testovány tytéž ztrátové funkce – MAE, MSE, RMSE, Huberova ztrátová funkce a GDL ve spojení s MAE.

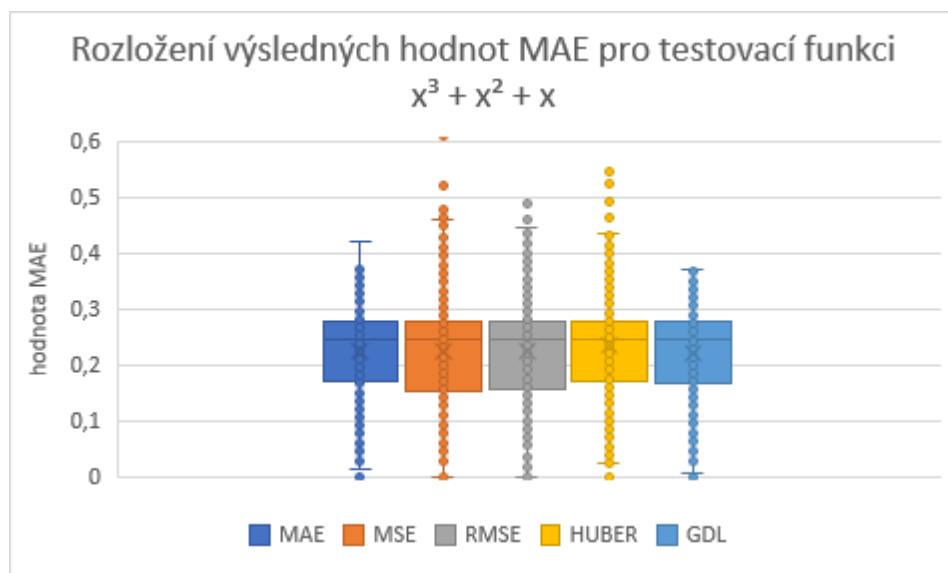
Za účelem dosažení výraznějších rozdílů mezi testovanými ztrátovými funkcemi, zejména u experimentů využívajících testovací funkce Ngueyn F7 a Nguyen F8, u nichž při použití první konfigurace nebyly rozdíly v použití různých ztrátových funkcí tak patrné, byly v této konfiguraci zpřísněny podmínky evolučního algoritmu. Konkrétně došlo ke snížení počtu členů populace na 20 a maximálnímu počtu generací na 30.

Experiment druhé konfigurace s testovací funkcí Nguyen F1

V rámci prvního experimentu druhé konfigurace bylo opět provedeno 5000 testů, přičemž pro každou z pěti zkoumaných ztrátových funkcí bylo spuštěno 1000 testů. Stejně jako první experiment první konfigurace, i v tomto byla pro vygenerování testovacích dat využita funkce ve tvaru:

$$Nguyen F1 = x^3 + x^2 + x$$

Výsledná fitness hodnota nejlepších jedinců všech testů byla stejně jako v první konfiguraci převedena na hodnotu MAE. Tyto výsledky poté byly rozděleny dle použité ztrátové funkce a vizualizovány pomocí následujícího krabicového grafu.



Obrázek 19: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F1 při druhé konfiguraci (zdroj: vlastní)

Grafické zpracování výsledků naznačuje relativní vyrovnanost všech zkoumaných ztrátových funkcí v rámci tohoto experimentu, zejména při porovnání třetích kvartilů a hodnot mediánů. Přesto je možné mezi jednotlivými ztrátovými funkcemi pozorovat určité rozdíly, především v hodnotách prvních kvartilů.

Nejlepších výsledků v tomto experimentu z pohledu prvních kvartilů dosáhla MSE následovaná ztrátovou funkcí RMSE. Za nimi se umístila GDL kombinovaná s MAE, u níž je patrný delší spodní fousek oproti hornímu, což naznačuje asymetrii rozdělení hodnot, tedy o něco větší počet úspěšnějších výsledků. Poté se umístila MAE, zatímco nejhorší výsledky tentokrát prokazovala Huberova ztrátová funkce s nejvýše položeným prvním kvantilem.

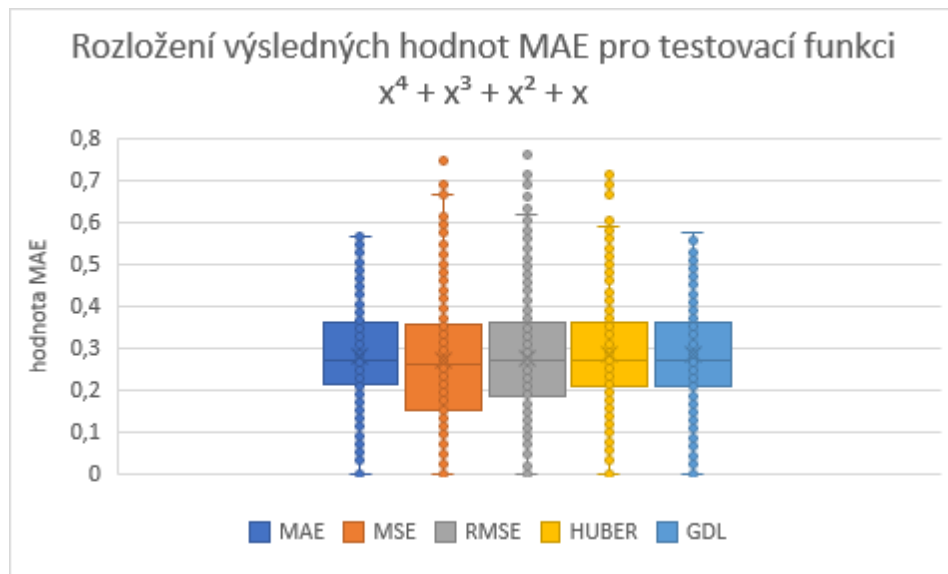
Je však důležité si povšimnout, že na rozdíl od stejného experimentu v první konfiguraci, kde většina prvních kvartilů dosahovala nulových hodnot, se nyní všechny hodnoty – včetně prvních a třetích kvartilů i mediánů – nacházejí výše. Tento posun lze pravděpodobně přičíst přísnějším podmínkám evolučního algoritmu v rámci druhé konfigurace.

Experiment druhé konfigurace s testovací funkcí Nguyen F2

Ve druhém experimentu druhé konfigurace byla pro testování použita data vygenerovaná pomocí testovací funkce následujícího tvaru:

$$Nguyen\ F2 = x^4 + x^3 + x^2 + x$$

Opět bylo v rámci tohoto experimentu provedeno 5000 testů, tedy 1000 testů pro každou z testovaných ztrátových funkcí. Výsledné hodnoty fitness nejlepších jedinců převedené na hodnoty MAE, byly zpracovány do následujícího krabicového grafu.



Obrázek 20: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F2 při druhé konfiguraci (zdroj: vlastní)

Graf rozložení výsledků v tomto experimentu opět ukazuje určitou vyrovnanost některých ztrátových funkcí, přestože lze pozorovat několik patrných rozdílů. Nejvýrazněji se od ostatních funkcí odlišuje MSE, která dosáhla nejen nejnižšího mediánu a třetího kvartilu, ale především nejnižší hodnoty prvního kvartilu.

Za MSE následuje RMSE, která má rovněž nižší první kvartil než ostatní ztrátové funkce. Medián i třetí kvartil RMSE jsou mírně nižší než u ostatních funkcí, i když v grafu nejsou tyto rozdíly příliš patrné.

Zbývající tři ztrátové funkce – Huberova, GDL (ve spojení s MAE) a samotná MAE – dosahují velmi podobných hodnot. Huberova a GDL vykazují téměř shodné hodnoty prvního kvartilu, přičemž první kvartil Huberovy funkce je nepatrně nižší. MAE má pak ze všech tří nejvyšší první kvartil. Mediány a třetí kvartily těchto funkcí jsou prakticky totožné.

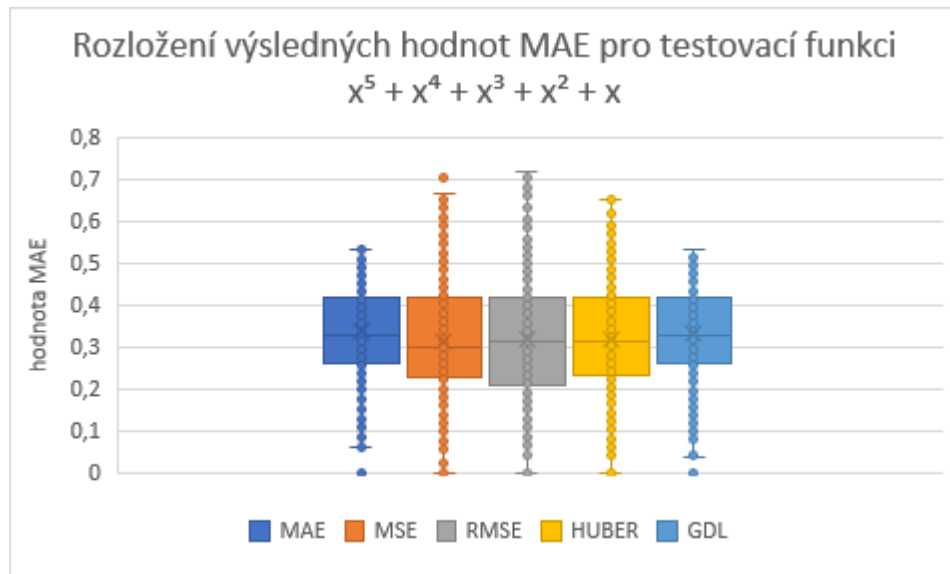
Je rovněž nutné upozornit, že oproti stejnému experimentu první konfigurace se i tentokrát všechny hodnoty – včetně prvních a třetích kvartilů i mediánů – nacházejí ve vyšších úrovních. Tento posun lze opět přičíst zpřísněným podmínkám evolučního algoritmu ve druhé konfiguraci.

Experiment druhé konfigurace s testovací funkcí Nguyen F3

V dalším experimentu této konfigurace měla testovací funkce, která byla použita pro vygenerování testovacích dat, následující tvar:

$$Nguyen\ F3 = x^5 + x^4 + x^3 + x^2 + x$$

I v tomto experimentu bylo provedeno 1000 testů pro každou testovanou ztrátovou funkci. Celkem tedy proběhlo 5000 testů. Výsledky všech těchto testů byly následně zpracovány a zobrazeny v podobě následujícího krabicového diagramu.



Obrázek 21: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F3 při druhé konfiguraci (zdroj: vlastní)

I v tomto experimentu je možné pozorovat určitou vyrovnanost testovaných ztrátových funkcí z pohledu porovnání třetích kvartilů. Přesto se výsledky tohoto experimentu poněkud liší.

Zajímavostí je výrazně nižší medián MSE, přestože nejnižší hodnotu prvního kvartilu vykazuje RMSE. Druhý nejnižší první kvartil patří MSE, třetí Huberově ztrátové funkci, která má zároveň nepatrně nižší medián než RMSE. MAE a GDL vykazují stejné hodnoty mediánů, avšak GDL má nejnvýše postavený první kvartil.

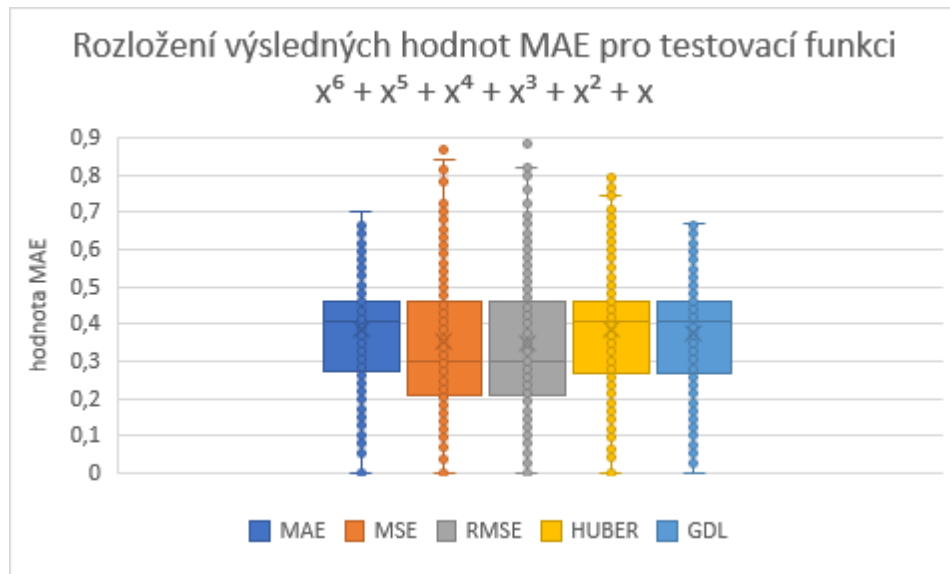
Stejně jako u předešlých dvou experimentech i tento v porovnání se stejným experimentem první konfiguraci vykazuje vyšší všechny hodnoty u všech ztrátových funkcí, což lze vysvětlit přísnějšími podmínkami použitými v evolučním algoritmu druhé konfigurace.

Experiment druhé konfigurace s testovací funkcí Nguyen F4

Čtvrtý experiment druhé konfigurace je také— posledním experimentem využívající k vygenerování testovacích dat testovací funkci tvořenou pouze mocninami proměnné x . Tvar použité testovací funkce tedy vypadá následovně:

$$Nguyen F4 = x^6 + x^5 + x^4 + x^3 + x^2 + x$$

I v tomto případě bylo provedeno 5000 testů, kde pro každých 1000 testů byla pro ohodnocení jedinců použita jiná ztrátová funkce. Rozložení hodnot MAE napříč ztrátovými funkcemi bylo znázorněno v následujícím grafu.



Obrázek 22: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F4 při druhé konfiguraci (zdroj: vlastní)

Graf ukazuje, že i v rámci tohoto experimentu vykazují třetí kvartily všech ztrátových funkcí velmi podobné hodnoty. Ačkoli při porovnání konkrétních hodnot GDL vykazuje nejnižší třetí kvartil a Huberova ztrátová funkce nejvyšší, rozdíly jsou téměř neznamatelné.

Výrazně lepších výsledků v tomto experimentu dosáhly RMSE a MSE, jejichž první kvartily dosahují stejných hodnot, avšak MSE má o něco nižší medián. Třetí v pořadí z pohledu prvního kvartilu je GDL následovaná Huberovou ztrátovou funkcí. Nejvyšší první kvartil patří MAE, jejíž medián je stejné hodnoty jako medián GDL. Huberova ztrátová funkce má o něco nižší medián.

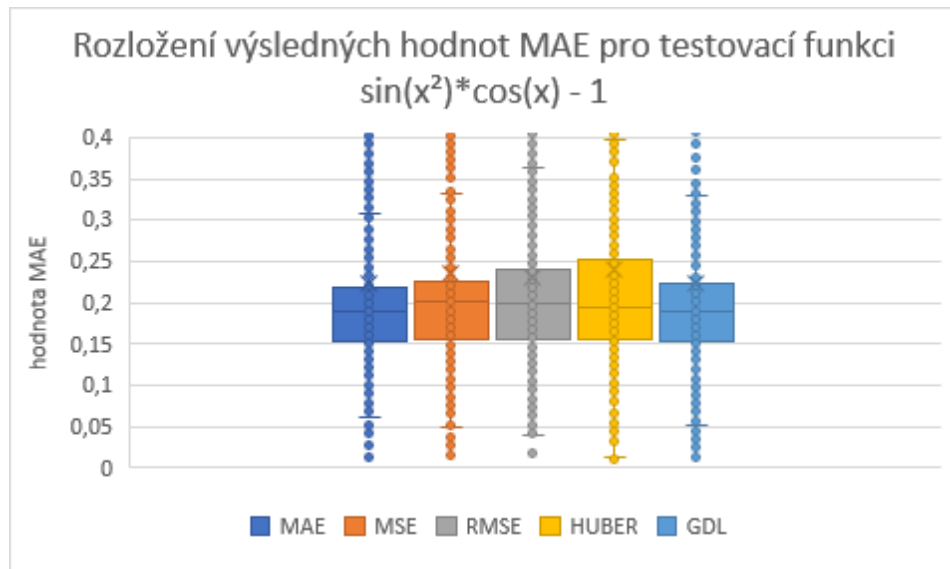
Je však třeba dodat, že rozdíly mezi MSE a RMSE a také mezi MAE, GDL a Huberovou ztrátovou funkcí jsou velmi malé až zanedbatelné. Za podstatné lze tedy v tomto experimentu považovat především výsledky MSE a RMSE, které se ukázaly jako úspěšnější ztrátové funkce než všechny ostatní.

Experiment druhé konfigurace s testovací funkcí Nguyen F5

V tomto experimentu již byla pro vygenerování testovacích dat využita testovací funkce obsahující nejen mocniny proměnné x , ale i goniometrické funkce:

$$Nguyen\ F5 = \sin(x^2) * \cos(x) - 1$$

Stejně jako u ostatních experimentů byly i v tomto případě všechny zkoumané ztrátové funkce otestovány na 1000 samostatných testech. Dohromady tedy bylo spuštěno 5000 testů v rámci tohoto experimentu. Získané hodnoty byly následně vizualizovány pomocí krabicového diagramu.



Obrázek 23: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F5 při druhé konfiguraci (zdroj: vlastní)

V rámci tohoto experimentu již nejsou třetí kvartily srovnatelné, jako tomu bylo u předešlých experimentů této konfigurace. Naopak hodnoty prvních kvartilů jsou si poměrně podobné, ačkoli jsou z grafu stále patrné rozdíly – nejnižší hodnotu prvního kvartilu vykazuje MAE následovaná GDL, nejhorší hodnota prvního kvartilu patří k Huberově ztrátové funkci.

MAE v tomto experimentu prokazuje nejlepší výsledky i z pohledu hodnoty třetího kvartilu i mediánu, v obou těsně následovaná GDL. Třetí nejnižší hodnotu mediánu má Huberova ztrátová funkce, ačkoli ta má také nejvýše posunutý třetí kvartil. Podobně i RMSE je čtvrtá v pořadí z hlediska mediánu, i když celkový rozptyl hodnot má větší než MSE.

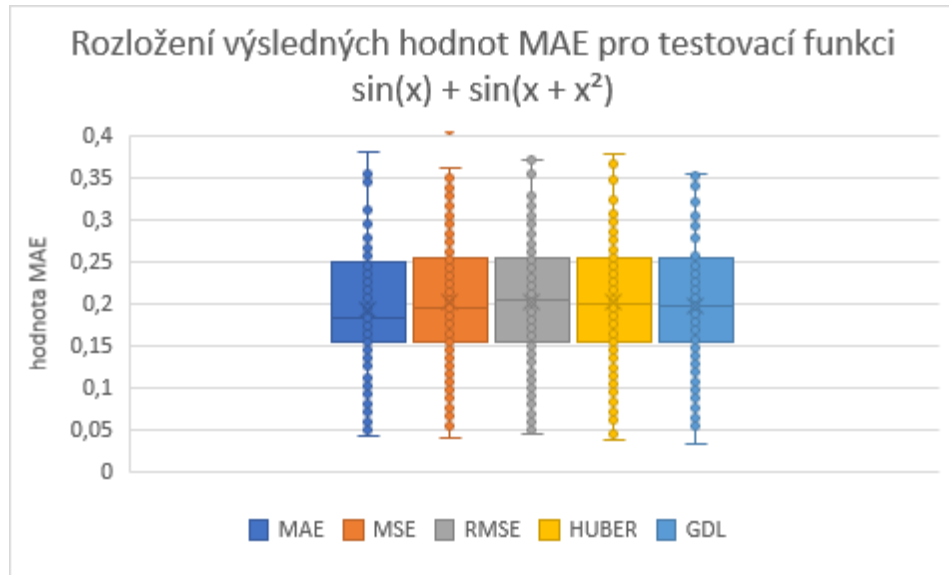
Experiment druhé konfigurace s testovací funkcí Nguyen F6

Podobně jako v minulém experimentu i v tomto byla pro vygenerování testovacích dat použita funkce využívající nejen mocnin proměnné x , ale i goniometrické funkce:

$$Nguyen\ F6 = \sin(x) + \sin(x + x^2)$$

Stejně tak celkový počet testů byl i tentokrát 5000, rovnoměrně rozdělených mezi pět testovaných ztrátových funkcí. Výsledky všech testů ve formě hodnot MAE, rozdělené podle

zkoumaných ztrátových funkcí, byly po zpracování znázorněny v následujícím krabicovém grafu.



Obrázek 24: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F6 při druhé konfiguraci (zdroj: vlastní)

V rámci tohoto experimentu graf ukazuje relativní rovnost prvních a třetích kvartilů všech testovaných ztrátových funkcí. A však z výsledků vyplývají rozdíly mezi hodnotami mediánů.

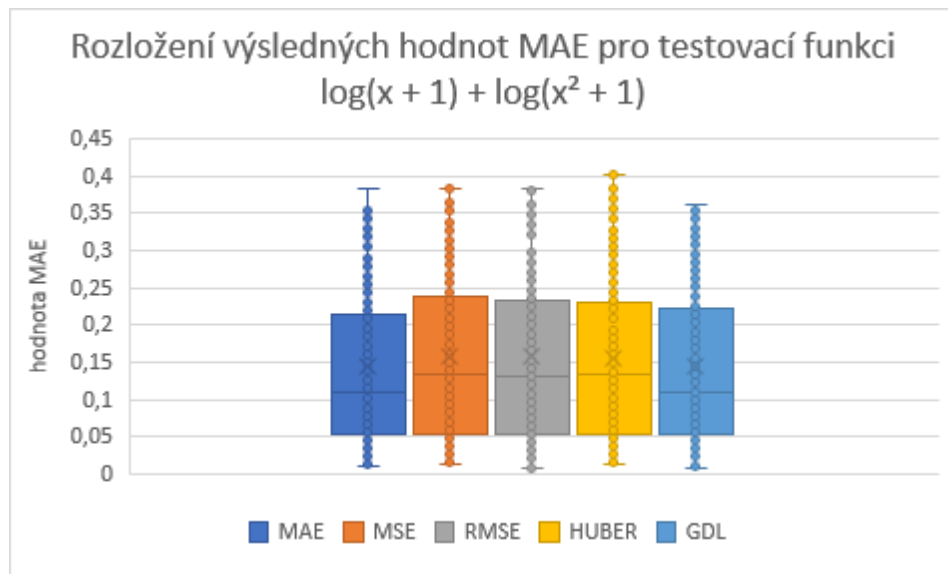
Při porovnání hodnot mediánů je nejlepší ztrátovou funkcí tohoto experimentu MAE. Druhou v pořadí je MSE následovaná GDL. Předposlední je Huberova ztrátová funkce a nejvyšší hodnota mediánu v rámci tohoto experimentu patří RMSE.

Experiment druhé konfigurace s testovací funkcí Nguyen F7

Předposlední experiment druhé konfigurace, v rámci něhož bylo opět provedeno 5000 testů, využíval pro generování testovacích dat funkci následujícího tvaru:

$$Nguyen F7 = \log(x + 1) + \log(x^2 + 1)$$

Podobně jako u všech předešlých experimentů byly i zde výsledky v podobě fitness hodnoty nejlepších jedinců z každého testu převedeny na hodnotu MAE. Tyto hodnoty poté byly vizualizovány pomocí následujícího krabicového diagramu.



Obrázek 25: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F7 při druhé konfiguraci (zdroj: vlastní)

V předposledním experimentu druhé konfigurace jsou patrné větší rozdíly ve výsledcích jednotlivých testovaných ztrátových funkcí, než tomu bylo v podobném experimentu provedeném v rámci první konfigurace. V předchozím experimentu totiž byly výsledky poměrně vyrovnané a všechny hodnoty třetích kvartilů nepřekročili hranici 0,08. Proto byla navržena druhá konfigurace s přísnějšími podmínkami evolučního algoritmu, jehož výsledky prokazují nižší úspěšnost jednotlivých ztrátových funkcí a patrnější rozdíly mezi nimi.

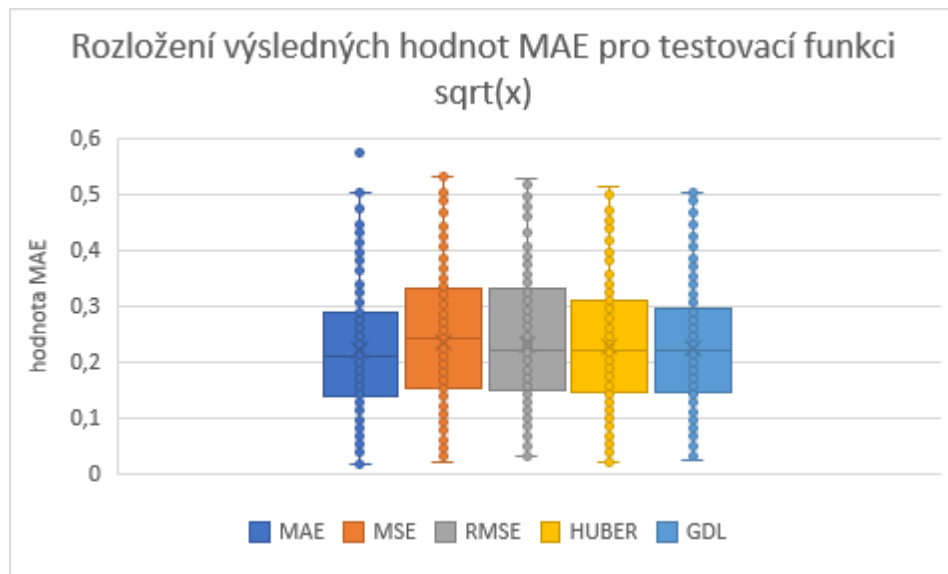
Ačkoli všechny první kvartily dosahují téměř stejných hodnot, nejmenší hodnota mediánu patří MAE, která má zároveň i nejnižší položený třetí kvartil. Druhou ztrátovou funkcí v pořadí z hlediska porovnání hodnot mediánů i třetích kvartilů je GDL. Funkce se třetím nejnižším mediánem je RMSE, ačkoli má o něco vyšší třetí kvartil než Huberova ztrátová funkce, která má čtvrtý nejvyšší medián. Nejhorší výsledky tohoto experimentu ukazuje MSE s nejvyšší hodnotou mediánu i třetího kvartilu.

Experiment druhé konfigurace s testovací funkcí Nguyen F8

Experiment využívající testovací funkci následujícího tvaru pro vygenerování testovacích dat je zároveň posledním experimentem druhé konfigurace.

$$Nguyen\ F8 = \sqrt{x}$$

I v rámci tohoto experimentu bylo provedeno 1000 testů pro každou z testovaných ztrátových funkcí. Dohromady tedy bylo spuštěno 5000 testů. Výsledky těchto testů byly poté zpracovány a znázorněny v následujícím krabicovém grafu.



Obrázek 26: Graf rozložení výsledných hodnot MAE pro testovací funkci Nguyen F8 při druhé konfiguraci (zdroj: vlastní)

I ve druhém experimentu, na který byla primárně zaměřena tato konfigurace, jsou patrnější rozdíly mezi výsledky zkoumaných ztrátových funkcí. Stejně jako tomu bylo u všech experimentů této konfigurace, i tady jsou hodnoty prvních a třetích kvartilů i mediánů vyšší, než tomu bylo u experimentů první konfigurace.

V tomto experimentu nejlepší výsledky prokazuje MAE, jež má celkové rozložení hodnot nižší než kterákoli jiná testovaná ztrátová funkce. Za druhou nejúspěšnější funkci tohoto experimentu lze považovat GDL. Následuje Huberova ztrátová funkce s o něco vyššími hodnotami mediánu a prvního i třetího kvartilu. Jako předposlední se v tomto experimentu umístila RMSE a nejhorší úspěšnost prokázala MSE s výrazně vyšším mediánem.

Shrnutí výsledků druhé konfigurace

Podobně jako v rámci první konfigurace i v této bylo provedeno osm samostatných experimentů, zkoumajících výkonnost pěti různých ztrátových funkcí (MAE, MSE, RMSE, Huberovy ztrátové funkce a GDL spojené s MAE). Během každého experimentu bylo spuštěno přesně 5000 testů, dohromady tedy bylo provedeno 40 000 testů. Výsledky ve formě fitness hodnot nejlepších jedinců ze všech testů převedených na hodnotu MAE byly vždy graficky zobrazeny.

Vzhledem k přísnějším podmínkám druhé konfigurace je pravděpodobné, že vyrovnanosti výsledků některých experimentů mohly být způsobeny nedostatečnou šancí na projevení všech aspektů jednotlivých ztrátových funkcí. I přes větší podobnosti hodnot bylo však možné

pozorovat určitou dominanci MSE a RMSE v prvních čtyřech experimentech, jež využívali k vygenerování testovacích dat testovací funkce složené ze součtu mocnin proměnné x .

Naopak ve druhé polovině experimentů, kde k vygenerování testovacích dat byly použity složitější funkce obsahující goniometrické tvary, převažovali s lepšími výsledky MAE a GDL. Huberova ztrátová funkce se nikdy nedostala na první dvě příčky a držela se spíše na středních pozicích.

I v tomto případě výsledky naznačují, že volba vhodné ztrátové funkce pro řešení symbolické regrese pomocí genetického programování z určité části závisí na charakteru a míře složitosti problému. Dle výsledků této konfigurace lze předpokládat, že pro jednodušší cílové funkce by mohlo být vhodnější zvolit MSE či RMSE, naopak ve složitějších případech by mohlo být výhodnější pracovat s MAE či s GDL.

2.2.4 Vyhodnocení všech experimentů

V rámci této práce bylo dohromady provedeno 80 000 testů rozdělených do dvou rozdílných konfigurací po osmi experimentech. Během nich byl zkoumán vliv pěti ztrátových funkcí (MAE, MSE, RMSE, Huberovy ztrátové funkce a GDL spojené s MAE) na řešení symbolické regrese pomocí genetického programování s využitím osmi testovacích funkcí ze sady Nguyen benchmark pro různé experimenty.

Při porovnání výsledků první i druhé konfigurace je patrné, že v experimentech využívajících ke generování dat testovací funkce Nguyen F1 až Nguyen F4, což jsou testovací funkce složené pouze z mocnin proměnné x , vždy dominovali ztrátové funkce MSE a RMSE. Naopak při použití složitějších testovacích funkcí obsahujících goniometrické členy se ukázaly jako úspěšnější ztrátové funkce MAE a GDL.

Na základě těchto výsledků se dá předpokládat, že v případě volby vhodné ztrátové funkce při řešení symbolické regrese pomocí genetického programování záleží do určité míry na složitosti předpokládané cílové funkce. V případě řešení jednodušších problémů se ukazuje jako účinnější MSE či RMSE, kdežto pro nalezení složitějších cílových funkcí by bylo vhodnější zvolit MAE, či ji zkombinovat s GDL.

ZÁVĚR

Cílem této práce bylo analyzovat vliv různých ztrátových funkcí na kvalitu řešení problému symbolické regrese pomocí genetického programování. Práce se zaměřila na základní popis symbolické regrese, principů genetického programování a vlastností vybraných ztrátových funkcí. V praktické části byly následně navrženy a realizovány experimenty, které měly za úkol ověřit, jak se volba konkrétní ztrátové funkce promítá do výsledků procesu symbolické regrese.

V rámci experimentálního zkoumání byly navrženy dvě konfigurace experimentů. První konfigurace využívala standardní nastavení parametrů evolučního algoritmu a jejím cílem bylo provést základní srovnání pěti zvolených ztrátových funkcí na osmi testovacích jednorozměrných funkcích ze sady Nguyen benchmark. Výsledky první konfigurace ukázaly, že pro jednodušší testovací funkce založené pouze na mocninách proměnné x dosahovaly nejlepších výsledků ztrátové funkce MSE a RMSE. Naproti tomu při řešení složitějších testovacích funkcí, obsahujících i goniometrické členy, vykazovaly lepší výkonnost ztrátové funkce MAE a GDL. Huberova ztrátová funkce se ve většině případů pohybovala ve střední části pořadí, čímž potvrdila svou roli kompromisního řešení.

Na základě výsledků první konfigurace byla navržena konfigurace druhá, která zpřísnila podmínky evolučního algoritmu. Snížením počtu jedinců v populaci a maximálního počtu generací došlo k zajištění náročnějších podmínek pro nalezení optimálního řešení. Cílem tohoto kroku bylo zejména prohloubení rozdílů mezi jednotlivými ztrátovými funkcemi a posouzení jejich chování v náročnějších podmínkách. Přestože výsledky druhé konfigurace byly ve srovnání s první konfigurací celkově o něco horší, vzhledem k přísnějším podmínkám evolučního algoritmu, základní vzorec úspěšnosti jednotlivých ztrátových funkcí zůstal zachován. I zde se MSE a RMSE ukázaly jako vhodnější při řešení jednodušších funkcí, zatímco MAE a GDL prokázaly své přednosti u složitějších úloh. Huberova ztrátová funkce si i nadále udržela střední pozici.

V průběhu experimentů se potvrdilo, že výběr ztrátové funkce skutečně ovlivňuje nejen kvalitu nalezených řešení, ale také stabilitu výsledků. Ztrátové funkce MSE a RMSE byly velmi efektivní při hledání jednoduchých modelů s nízkou chybou, avšak v některých případech vykazovaly vyšší variabilitu výsledků. Naproti tomu MAE a GDL byly úspěšnější v případě složitějších modelů, přičemž jejich výsledky byly často konzistentnější. Nevýhodou

MAE a GDL může být nižší citlivost na malé odchylky, což se projevovalo u jednodušších funkcí, kde nedosahovaly tak nízkých hodnot jako MSE a RMSE.

Výsledky této práce ukazují, že při řešení symbolické regrese pomocí genetického programování je vhodné volit ztrátovou funkci s ohledem na charakter problému. Pro jednodušší funkce s hladkým průběhem se jako efektivnější ukazují MSE a RMSE, zatímco ve složitějších úlohách, které obsahují například goniometrické prvky, mohou lepší výsledky přinést MAE nebo GDL.

Do budoucna by bylo možné navázat na tuto práci například rozšířením experimentů za použití jiných testovacích funkcí nebo zkoumáním chování ztrátových funkcí na vícerozměrných úlohách. Zajímavým směrem by mohlo být rovněž ověření výsledků na reálných datech nebo za použití jiných evolučních algoritmů, čímž by bylo možné lépe posoudit univerzálnost zjištěných poznatků.

POUŽITÁ LITERATURA

- [1] WANG, Yiqun; WAGNER, Nicholas a RONDINELLI, James M. Symbolic regression in materials science. Online. *MRS Communications*. 2019, vol. 9, no. 3, s. 793-805. Dostupné z: <https://doi.org/10.1557/mrc.2019.85>. [cit. 2025-04-03].
- [2] ANGELIS, Dimitrios; SOFOS, Filippos a KARAKASIDIS, Theodoros E. Artificial Intelligence in Physical Sciences: Symbolic Regression Trends and Perspectives. Online. *Archives of Computational Methods in Engineering*. 2023, vol. 30, no. 6, s. 3845-3865. Dostupné z: <https://doi.org/10.1007/s11831-023-09922-z>. [cit. 2025-04-03].
- [3] WILLMOTT, Cort J. a MATSUURA, Kenji. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Online. *Climate Research*. 2005, vol. 30, no. 1, s. 79-82. Dostupné z: <https://www.jstor.org/stable/24869236>. [cit. 2025-04-03].
- [4] HODSON, Timothy O. Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. Online. *Geoscientific Model Development*. 2022, vol. 15, no. 14, s. 5481-5487. Dostupné z: <https://doi.org/10.5194/gmd-15-5481-2022>. [cit. 2025-04-03].
- [5] XIE, Wenxuan; WU, Jiali a SHENG, Yuhong. RETRACTED: Uncertain regression model based on Huber loss function. Online. *Journal of Intelligent & Fuzzy Systems*. 2023, vol. 45, no. 1, s. 1169-1178. ISSN 10641246. Dostupné z: <https://doi.org/10.3233/JIFS-223641>. [cit. 2025-04-03].
- [6] BALASUNDARAM, S. a PRASAD, Subhash Chandra. Robust twin support vector regression based on Huber loss function. Online. *Neural Computing and Applications*. 2020, vol. 32, no. 15, s. 11285-11309. ISSN 0941-0643. Dostupné z: <https://doi.org/10.1007/s00521-019-04625-8>. [cit. 2025-04-03].
- [7] TONG, Hongzhi. Functional linear regression with Huber loss. Online. *Journal of Complexity*. 2023, vol. 74. ISSN 0885064X. Dostupné z: <https://doi.org/10.1016/j.jco.2022.101696>. [cit. 2025-04-03].
- [8] LIU, Ruixia; HU, Huichen; ZHANG, Shuaishuai; DENG, Yanjun; LIU, Zhaoyang et al. An ECG denoising technique based on AHIN block and gradient difference max loss. Online. *Journal of Electrocardiology*. 2024, vol. 86. ISSN 00220736. Dostupné z: <https://doi.org/10.1016/j.jelectrocard.2024.153761>. [cit. 2025-04-03].
- [9] POLI, Riccardo; LANGDON, W. B.; MCPHEE, Nicholas F. a KOZA, John R. *A field guide to genetic programming*. [S.l.: Lulu Press], 2008. ISBN 978-1-4092-0073-4.

- [10] HYNEK, Josef. *Genetické algoritmy a genetické programování*. Průvodce. Praha: Grada, 2008. ISBN 978-80-247-2695-3.
- [11] KOZA, John R. Genetic programming as a means for programming computers by natural selection. Online. *Statistics and Computing*. 1994, vol. 4, no. 2, s. 87-112. ISSN 0960-3174. Dostupné z: <https://doi.org/10.1007/BF00175355>. [cit. 2025-04-03].
- [12] SETTE, S. a BOULLART, L. Genetic programming: principles and applications. Online. *Engineering Applications of Artificial Intelligence*. 2001, vol. 14, no. 6, s. 727-736. ISSN 09521976. Dostupné z: [https://doi.org/10.1016/S0952-1976\(02\)00013-1](https://doi.org/10.1016/S0952-1976(02)00013-1). [cit. 2025-04-03].
- [13] MITCHELL, Melanie. *An introduction to genetic algorithms*. 2nd print. A Bradford Book. Cambridge, Mass.: MIT Press, 1996. ISBN 978-0-262-13316-6.
- [14] SCHWEIM, Dirk; WITTENBERG, David a ROTHLAUF, Franz. On sampling error in genetic programming. Online. *Natural Computing*. 2022, vol. 21, no. 2, s. 173-186. ISSN 1567-7818. Dostupné z: <https://doi.org/10.1007/s11047-020-09828-w>. [cit. 2025-04-03].
- [15] WALKER, Matthew. *Introduction to Genetic Programming*. Massey University, 2001. Dostupné také z: https://icog-labs.com/wp-content/uploads/2014/07/Introduction_to_GP_Matthew-Walker.pdf.
- [16] MILLER, Brad L. a GOLDBERG, David E. Genetic Algorithms, Tournament Selection, and the Effects of Noise. Online. *Complex Systems*. 1995, vol. 9, no. 3, s. 193-212. Dostupné z: https://www.complex-systems.com/abstracts/v09_i03_a02/. [cit. 2025-04-03].
- [17] LIPOWSKI, Adam a LIPOWSKA, Dorota. Roulette-wheel selection via stochastic acceptance. Online. *Physica A: Statistical Mechanics and its Applications*. 2012, vol. 391, no. 6, s. 2193-2196. ISSN 03784371. Dostupné z: <https://doi.org/10.1016/j.physa.2011.12.004>. [cit. 2025-04-03].
- [18] PHAM, Vu Hong Son; NGUYEN DANG, Nghiep Trinh a NGUYEN, Van Nam. Enhancing engineering optimization using hybrid sine cosine algorithm with Roulette wheel selection and opposition-based learning. Online. *Scientific Reports*. 2024, vol. 14, no. 1, s. 694. ISSN 2045-2322. Dostupné z: <https://doi.org/10.1038/s41598-024-51343-w>. [cit. 2025-04-03].
- [19] UY, Nguyen Quang; HOAI, Nguyen Xuan; O'NEILL, Michael; MCKAY, R. I. a GALVÁN-LÓPEZ, Edgar. Semantically-based crossover in genetic programming: application to real-valued symbolic regression. Online. *Genetic Programming and Evolvable Machines*. 2011, vol. 12, no. 2, s. 91-119. ISSN 1389-2576. Dostupné z: <https://doi.org/10.1007/s10710-010-9121-2>. [cit. 2025-04-03].

- [20] EIBEN, A.E. a SMITH, J.E. *Introduction to Evolutionary Computing*. Online. Natural Computing Series. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. ISBN 978-3-662-44873-1. Dostupné z: <https://doi.org/10.1007/978-3-662-44874-8>. [cit. 2025-04-03].
- [21] HEURISTICLAB. *HeuristicLab A Paradigm-Independent and Extensible Environment for Heuristic Optimization*. Online. Heuristiclab. 2012. Dostupné z: https://dev.heuristiclab.com/trac.fcgi/blog/gkronber/symbolic_regression_benchmark. [cit. 2025-04-12].

SEZNAM PŘÍLOH

Příloha A: Zdrojové kódy programu použitého pro spouštění experimentů