


# Návrh diskrétní verze algoritmu diferenciální evoluce ve variantě DISH

Bc. Pavel Oškera

---

Diplomová práce  
2020

 Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

# Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Pavel Oškera**  
Osobní číslo: **A18254**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **Prezenční**  
Téma práce: **Návrh diskrétní verze algoritmu diferenciální evoluce ve variantě DISH**  
Téma práce anglicky: **Discrete Differential Evolution Design for DISH Algorithms**

### Zásady pro vypracování

1. Vypracujte literární řešení na dané téma.
2. Zvolte vhodné metody využitelné pro algoritmus diferenciální evoluce.
3. Implementujte algoritmus DISH ve Vámi zvoleném programovacím jazyce.
4. Implementujte zvolené metody z bodu 2. v algoritmu DISH.
5. Otestujte Vámi implementované verze algoritmu na vybraných benchmark problémech z diskrétní optimalizace.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam doporučené literatury:

1. ZELINKA, Ivan. Evoluční výpočetní techniky: principy a aplikace. Praha: BEN, 2009. ISBN 80-7300-218-3.
2. DAS, Swagatam, Sankha Subhra MULLICK a P.N. SUGANTHAN. Recent advances in differential evolution ? An updated survey. Swarm and Evolutionary Computation [online]. 2016, 27, 1-30 [cit. 2019-11-11]. DOI: 10.1016/j.swevo.2016.01.004. ISSN 22106502. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S2210650216000146>
3. KRUSE, Rudolf, Christian BORGELT, Christian BRAUNE, Sanaz MOSTAGHIM a Matthias STEINBRECHER. Computational Intelligence: A Methodological Introduction. Second Edition. London: Springer, 2016. ISBN 978-1-4471-7294-9.
4. BLUM, Christian a Günther R. RAIDL. Hybrid Metaheuristics: Powerful Tools for Optimization. Switzerland: Springer, 2016. ISBN 978-3-319-30882-1.
5. MOCKUS, Jonas, William EDDY, Audris MOCKUS, Linas MOCKUS a Gintaras REKLAITIS. Bayesian Heuristic approach to discrete and global optimization: Algorithms, visualization, software, and applications. Dordrecht: Springer, 1997. ISBN 978-1-4419-4767-3.
6. VIKTORIN, Adam, ŠENKERÍK, Roman, PLUHÁČEK, Michal, KADAVÝ, Tomáš, ZAMUDA, Aleš. Distance based parameter adaptation for Success-History based Differential Evolution. Swarm and Evolutionary Computation, 2019, 2019, 50, 1-17. ISSN 2210-6502.
7. KACPRZYK, Janusz a Witold PEDRYCZ. Springer handbook of computational intelligence. Dordrecht: Springer, 2015, lvi, 1633 s. ISBN 978-3-662-43504-5.

Vedoucí diplomové práce:

**Ing. Adam Viktorin**

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: 28. listopadu 2019  
Termín odevzdání diplomové práce: 15. května 2020



---

doc. Mgr. Milan Adámek, Ph.D.  
děkan

---

prof. Mgr. Roman Jašek, Ph.D.  
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Bc. Pavel Oškera v.r.  
podpis diplomanta

## **ABSTRAKT**

Cílem této práce bylo vytvořit diskretní variantu diferenciální evoluce ve variantě DISH. V teoretické části je nejprve popsáno, co to jsou evoluční algoritmy a k čemu se využívají. Následně je uvedena základní verze diferenciální evoluce, ze které výsledný algoritmus vychází a postupný vývoj od této základní verze až po verzi DISH. Jelikož diferenciální evoluce i její rozšířené verze jsou určeny pro spojité optimalizační úlohy, je uvedeno, jak lze algoritmus upravit, aby mohl řešit úlohy diskretní optimalizace. Teoretická část je pak uzavřena popisem jak a proč evoluční algoritmy testovat.

Praktická část obsahuje výsledky testování vzniklého algoritmu. Byly implementovány čtyři diskretní varianty, které byly otestovány na osmi testovacích funkcích. Pro porovnání úspěšnosti vzniklého algoritmu byly stejné funkce optimalizovány pomocí genetického algoritmu, který je už v základní verzi určen pro diskretní úlohy.

Klíčová slova: Diferenciální evoluce, DISH, Benchmarking, Diskretní optimalizace, Heuristická optimalizace

## **ABSTRACT**

The goal of this thesis was to create discrete variant of differential evolution in the DISH variant. The theoretical part describes what evolutionary algorithms are and what they are used for. Subsequently, the basic version of the differential evolution, on which the resulting algorithm is based, is described there along with the development from the basic version to the DISH version. Since differential evolution and its extended versions are for solving continuous optimization tasks, there is shown how the algorithm can be modified so it can solve discrete optimization tasks. This part is then concluded with description of why and how to test evolutionary algorithms.

The practical part contains results of testing of the final form of newly created algorithm. Four discrete variants were implemented, which were tested on eight test functions. To compare the success of the resulting algorithm the same functions were optimized using genetic algorithm, which used for discrete optimization tasks.

Keywords: Differential Evolution, DISH, Benchmarking, Discrete Optimization, Heuristic Optimization

Rád bych zde poděkoval vedoucímu mé diplomové práce panu Ing. Adamovi Viktorinovi za jeho trpělivost, ochotu a cenné rady při realizaci této práce. Dále bych chtěl poděkovat všem, kteří mi jakýmkoliv způsobem pomohli při tvorbě této práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

|  |           |
|--|-----------|
| <b>ÚVOD</b> .....  | <b>9</b>  |
| <b>I TEORETICKÁ ČÁST</b> .....   | <b>10</b> |
| <b>1 EVOLUČNÍ VÝPOČETNÍ TECHNIKY</b> .....   | <b>11</b> |
| 1.1    SPOLEČNÉ RYSY .....   | 12        |
| 1.2    OPTIMALIZACE .....  | 12        |
| 1.3    NO FREE LUNCH TEORÉM.....   | 13        |
| 1.4    ZÁKLADNÍ POJMY EVOLUČNÍCH ALGORITMŮ.....  | 13        |
| <b>2 DIFERENCIÁLNÍ EVOLUCE</b> .....   | <b>15</b> |
| 2.1    PRINCIP DIFERENCIÁLNÍ EVOLUCE .....   | 15        |
| 2.1.1    Mutace.....   | 16        |
| 2.1.2    Křížení.....  | 18        |
| 2.1.3    Selekce .....   | 20        |
| 2.2    VARIANTY DIFERENCIÁLNÍ EVOLUCE .....  | 20        |
| 2.2.1    DE s propagací řídicích parametru – jDE .....   | 20        |
| 2.2.2    DE s adaptací strategie – SaDE.....   | 21        |
| 2.2.3    Adaptivní DE s volitelným externím archivem – JADE.....                                   | 21        |
| 2.2.4    DE se sadou řídicích parametrů a strategií – EPSDE .....                                  | 22        |
| 2.2.5    JADE s adaptivitou parametrů založenou na historicky úspěšných<br>hodnotách – SHADE ..... | 22        |
| 2.2.6    SHADE s redukcí velikosti populace – L-SHADE.....   | 23        |
| <b>3 DIFERENCIÁLNÍ EVOLUCE VE VARIANTĚ DISH</b> .....  | <b>24</b> |
| 3.1    VZNIK VARIANTY DISH .....   | 24        |
| 3.2    ALGORITMUS DISH .....   | 25        |
| 3.2.2    Nastavení F a CR .....  | 26        |
| 3.2.3    Mutace.....   | 28        |
| 3.2.4    Křížení.....  | 29        |
| 3.2.5    Selekce a aktualizace archivu.....  | 29        |
| 3.2.6    Snížení velikosti populace.....   | 30        |
| 3.2.7    Aktualizace paměti $M_F$ a $M_{CR}$ .....   | 30        |
| 3.2.8    Aktualizace populace .....  | 31        |
| <b>4 METODY DISKRETIZACE</b> .....   | <b>32</b> |
| 4.1    ZAOKROUHLNÍ .....   | 32        |
| 4.1.1    Zaokrouhlení v populaci .....   | 32        |
| 4.1.2    Zaokrouhlení při ohodnocení .....   | 32        |
| 4.2    TRANSFORMACE ČÍSEL .....  | 32        |
| 4.2.1    Dopředná transformace .....   | 33        |
| 4.2.2    Zpětná transformace.....  | 33        |
| 4.3    DISKRÉTNÍ MUTACE .....  | 33        |

|           |  |           |
|-----------|--|-----------|
| <b>5</b>  | <b>TESTOVÁNÍ EVOLUČNÍCH ALGORITMŮ .....</b>                  | <b>35</b> |
| <b>6</b>  | <b>GENETICKÝ ALGORITMUS.....</b>                             | <b>36</b> |
| <b>II</b> | <b>PRAKTICKÁ ČÁST .....</b>                                  | <b>37</b> |
| <b>7</b>  | <b>WOLFRAM MATHEMATICA .....</b>                             | <b>38</b> |
| <b>8</b>  | <b>TESTOVÁNÍ DIFERENCIÁLNÍ EVOLUCE VE VARIANTĚ DISH.....</b> | <b>39</b> |
| 8.1       | NASTAVENÍ GENETICKÉHO ALGORITMU .....                        | 39        |
| 8.2       | TESTOVACÍ FUNKCE .....                                       | 40        |
| <b>9</b>  | <b>VÝSLEDKY ALGORITMU .....</b>                              | <b>43</b> |
| 9.1       | F1: ONEMAX.....  | 43        |
| 9.2       | F2: LINEÁRNÍ FUNKCE .....                                    | 46        |
| 9.3       | F3: SPHERE .....   | 48        |
| 9.4       | F4: SCHWEFELŮV PROBLÉM 1.2 .....                             | 50        |
| 9.5       | F5: SCHWEFELŮV PROBLÉM 2.26 .....                            | 52        |
| 9.6       | F6: SALOMONOVA FUNKCE .....                                  | 54        |
| 9.7       | F7: ACKLEYHO FUNKCE .....                                    | 56        |
| 9.8       | F8: GREIWANKOVA FUNKCE.....                                  | 58        |
| 9.9       | SHRNUTÍ VÝSLEDKŮ .....                                       | 60        |
|           | <b>ZÁVĚR .....</b>   | <b>61</b> |
|           | <b>SEZNAM POUŽITÉ LITERATURY.....</b>                        | <b>63</b> |
|           | <b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>               | <b>65</b> |
|           | <b>SEZNAM OBRÁZKŮ .....</b>                                  | <b>67</b> |
|           | <b>SEZNAM TABULEK.....</b>                                   | <b>68</b> |
|           | <b>SEZNAM PŘÍLOH.....</b>                                    | <b>69</b> |



## ÚVOD

Díky neustále se zvyšujícímu výkonu výpočetní techniky dochází k rozvoji evolučních výpočetních algoritmů. Tyto algoritmy jsou založeny na zjednodušené verzi evoluce, se kterou se setkáváme v přírodě. Úkolem evolučních algoritmů je nalézt optimální řešení reálného problému z různých oblastí. Aby byly schopny tento problém vyřešit, je potřeba, aby bylo možné problém popsat matematickým vztahem nebo účelovou funkcí. Oproti klasickým analytickým metodám výpočtu mají evoluční algoritmy řadu výhod, díky kterým umožňují řešit problémy, které by jinak byly neřešitelné.

Evolučních algoritmů existuje velké množství a každý je vhodný pro jiný typ problému. Jedním z nejrozšířenějších evolučních algoritmů je právě diferenciální evoluce. Ta vznikla již v roce 1995 a od té doby je neustále rozšiřována a zdokonalována. Nové verze přichází s dokonalejším nastavením parametrů, pomocnými paměťmi a samostatným učením.

Tato práce vychází z varianty diferenciální evoluce označené jako DISH. DISH se snaží odstranit problém konvergence k lokálnímu extrému přidáním ohodnocení vzdálenosti jednotlivých kandidátů řešení a tím upřednostnit prohledávání zvoleného prostoru ve snaze nalézt globální minimum/maximum.

Jelikož základní diferenciální evoluce i většina jejích odvozených verzí je určena pro výpočty ve spojitě oblasti, je tato práce zaměřena na nalezení vhodných metod pro převod tohoto algoritmu do diskrétní oblasti.

V teoretické části práce je nejprve uvedeno, co jsou to evoluční algoritmy a k čemu slouží. Jsou zde základní pojmy a informace, které budou v práci dále využívány i důvod, proč je třeba vyvíjet a testovat stále nové algoritmy. Následuje popis základní verze diferenciální evoluce a některých zvolených variant. Hlavní pozornost je pak věnována popisu varianty DISH a metodám diskretizace této varianty.

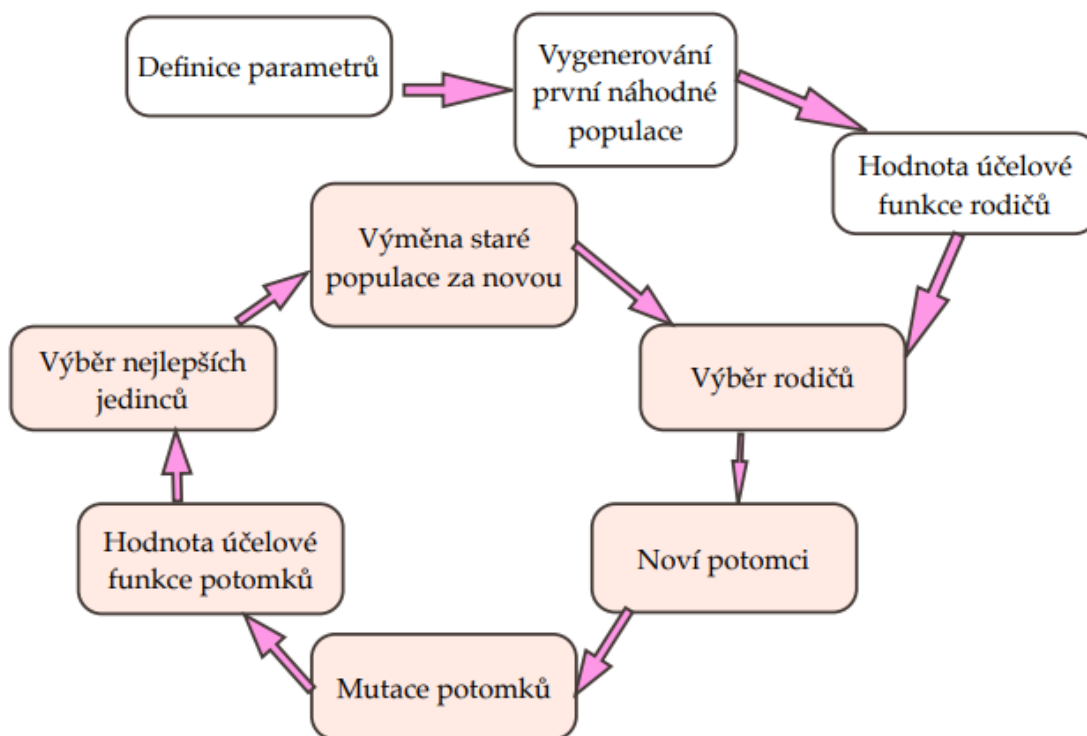
V praktické části jsou uvedeny výsledky, kterých dosáhly jednotlivé implementované diskrétní varianty na zvolených testovacích funkcích. Pro lepší představu o úspěšnosti algoritmu jsou výsledky jednotlivých diskrétních variant doplněny o výsledky genetického algoritmu, který je již v základní verzi navržen pro řešení úloh z oblasti diskrétní optimalizace.

## **I. TEORETICKÁ ČÁST**

## 1 EVOLUČNÍ VÝPOČETNÍ TECHNIKY

Evoluční výpočetní techniky v informatice jsou tvořeny skupinou algoritmů pro globální optimalizaci určitého problému a tvoří podobory pro oblast soft computingu a umělé inteligence. Algoritmy jsou inspirovány Darwinovými principy přirozené evoluce [14] a Mendelovým zákonem dědičnosti [15]. Optimalizační problém je zde definován pomocí účelové funkce, která udává kvalitu jednotlivých kandidátů řešení. Cílem algoritmů je nalezení globálního extrému této funkce [1].

Při evolučním výpočtu se vygeneruje počáteční populace, která se následně iterativně aktualizuje, čímž je simulován její vývoj (obr. 1). Při tvorbě nové generace jsou noví kandidáti vytvořeni na základě ohodnocení původní generace metodami křížení, mutace a následné selekce. Díky tomu se populace bude postupně vyvíjet a nalézat kvalitnější řešení podle zadané účelové funkce [1].



Obrázek 1. Evoluční výpočetní techniky – tvorba generace [1]

Evoluční výpočetní techniky mohou produkovat vysoce optimalizovaná řešení v širokém spektru problémů, což je činí populárními v počítačové informatice. Existuje mnoho variant a rozšíření, vhodných pro konkrétnější rodiny problémů a datových struktur. Lze je využít u

komplexních problémů, kde je velmi těžké nebo nemožné najít optimální řešení analytickou cestou [1].

## 1.1 Společné rysy

- **Jednoduchost** – algoritmy lze snadno naprogramovat, nevyužívají žádné složité funkce.
- **Hybridnost čísel** – algoritmy dokážou pracovat bez problému s libovolnými typy čísel (celá čísla, desetinná čísla nebo jen vybrané množiny čísel).
- **Rychlost** – díky své jednoduchosti jsou algoritmy většinou daleko rychlejší než klasické metody výpočtu.
- **Schopnost nalezení extrému** – algoritmy jsou navrženy tak, aby postupně konvergovaly k extrému funkce.
- **Vícenásobné řešení** – algoritmy lze upravit na takzvané „Niching algoritmy“. S touto úpravou je výsledkem vždy celá populace, kde jedno řešení je sice nejlepší, ale dostáváme celou množinu podobně kvalitních řešení, ze kterých si můžeme vybrat.
- **Vysoký počet dimenzí** – díky své jednoduchosti jsou algoritmy schopny zpracovávat problémy s velkým počtem vstupních parametrů a poměrně rychle nalézt extrémy.

## 1.2 Optimalizace

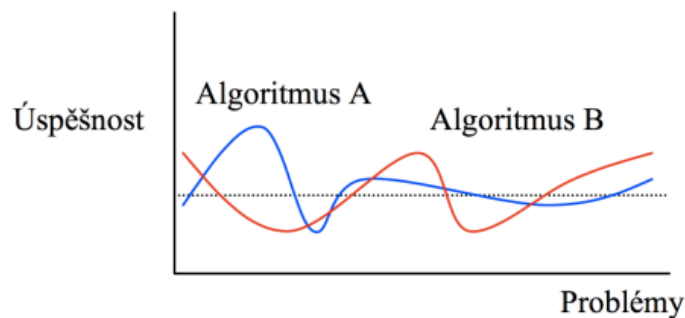
Snahou evolučních algoritmů je najít optimální řešení konkrétního problému. Ten je nejčastěji popsán pomocí účelové funkce  $f(x_1, x_2, \dots, x_n)$ , jejíž hodnotu se snažíme minimalizovat nebo maximalizovat. Vstupní proměnné  $x$  představují nastavitelné parametry daného systému, který se snažíme optimalizovat. Může se jednat o sestavování výrobního plánu pro podnik, plánování dopravy nebo plán pro národní hospodářství. Evoluční algoritmy lze použít ve velkém množství odvětví a díky své jednoduchosti dokáží řešit i velmi komplikované problémy.

Vstupní parametry mohou nabývat reálných nebo celočíselných hodnot podle toho, jaký problém se aktuálně řeší. Jelikož se jedná o reálný problém, vždy ovšem mohou nabývat hodnoty pouze z určitého intervalu.

Dalším omezením pro evoluční algoritmy může být omezení hodnoty účelové funkce. Pokud například plánujeme výrobu v podniku a snažíme se maximalizovat zisk, určitě bude podmínkou, aby byla hodnota účelové funkce kladná.

### 1.3 No Free Lunch teorém

U optimalizačních evolučních algoritmů existuje statisticky významná skutečnost, která se nazývá No Free Lunch teorém [11]. Ten říká, že neexistuje dokonalý algoritmus, který by dokázal vyřešit všechny problémy nejlépe. Každý algoritmus je vhodný pro řešení jiného typu úloh. To je ukázáno na obr. 2, kde jsou zobrazeny nejlepší výsledky dvou algoritmů na různých problémech. Právě z tohoto důvodu existuje velké množství evolučních algoritmů a jejich modifikací [1].



Obrázek 2. No Free Lunch teorém [1]

### 1.4 Základní pojmy evolučních algoritmů

- **Účelová funkce** – Je to funkce, která udává kvalitu nalezeného řešení. Nejčastěji je reprezentována pomocí geometrické „hyperplochy“ v  $(D+1)$  dimenzionálním prostoru. Kde  $D$  představuje počet optimalizovaných vstupních parametrů a jeden rozměr navíc udává kvalitu řešení.
- **Jedinec** – Představuje jedno z možných řešení. Každý jedinec je tvořen sadou vstupních parametrů, které se snažíme optimalizovat a hodnotou účelové funkce.
- **Populace** – Je soubor jedinců. Nejčastěji bývá znázorněna jako matice  $M \times N$  (obr. 3.), kde v každém sloupci je jeden jedinec a v jednotlivých řádcích má svoje

parametry.

|                 | $J_1$       | $J_2$       | $J_3$       | $J_4$      | ... | ... | ... | $J_M$       |
|-----------------|-------------|-------------|-------------|------------|-----|-----|-----|-------------|
| <b>Vhodnost</b> | <b>55,2</b> | <b>68,3</b> | <b>5,36</b> | <b>9,5</b> | ... | ... | ... | <b>0,89</b> |
| $P_1$           | 2,55        | 549,3       | -55,36      | 896,5      | ... | ... | ... | 1,89        |
| $P_2$           | 0,25        | 66,2        | 2           | -10        | ... | ... | ... | -2,2        |
| $P_3$           | -66,3       | 56          | 4           | 15,001     | ... | ... | ... | -83,66      |
| ...             | ...         | ...         | ...         | ...        | ... | ... | ... | ...         |
| $P_N$           | 259,3       | -10         | 22,22       | 536,22     | ... | ... | ... | -42,22      |

Obrázek 3. Příklad populace v evolučním algoritmu

- **Generace** – Jedná se o aktuální populaci, se kterou algoritmus pracuje. S každou iterací je vytvořena nová generace a stará je zapomenuta.
- **Elitismus** – Díky elitismu je do nové generace vybrán vždy lepší ze vzniklé dvojice rodič – potomek, což zaručuje zlepšování nalezeného řešení.
- **Křížení** – Je operace, která se využívá při tvorbě nového jedince. U Diferenciální evoluce se jedná o kombinaci parametrů rodiče a nově vzniklého „zmutovaného“ jedince.
- **Mutace** – V DE je to vektorová operace, pomocí které je vytvořen nově vzniklý „zmutovaný“ jedinec. Pro základní mutační strategii rand/1 se jedná o vážený rozdíl dvou náhodně zvolených jedinců, který je přičtený k dalšímu náhodnému jedinci. Existuje ovšem více mutačních strategií a u jiných algoritmů se může jednat o zcela jinou operaci.
- **Hranice prohledávaného prostoru** – Jelikož se nejčastěji jedná o optimalizaci určitého reálného problému, nemohou parametry nabývat libovolných hodnot, ale jen hodnot v určitém rozsahu (např.  $-10, 10$ ). Pokud při tvorbě nového jedince dojde k překročení této hranice, je potřeba vstupní parametr nějakým způsobem upravit, aby se vrátil do přijatelných mezí.

## 2 DIFERENCIÁLNÍ EVOLUCE

Diferenciální evoluce je evoluční algoritmus, který vznikl v roce 1995 [12]. V původní verzi algoritmus umožňuje řešit problémy definované účelovou funkcí s parametry v reálném prostoru. Hlavní myšlenka spočívá v diferenciální mutaci a křížení jedinců populace. Procesy mutace a křížení jsou prováděny jako jednoduché vektorové operace [1].

Oproti jiným evolučním algoritmům má diferenciální evoluce netradiční pořadí operací. U ostatních algoritmů většinou první dochází ke křížení rodičů a následné mutaci vzniklého potomka. Zde jsou operace prohozeny a první dochází k mutaci a až poté ke křížení [1].

### 2.1 Princip Diferenciální evoluce

Základní verze diferenciální evoluce má čtyři vstupní parametry:

- Ukončovací podmínka – Může se jednat o maximální počet generací, počet ohodnocení pomocí účelové funkce nebo třeba maximální doba pro výpočet.
- $NP$  – Velikost populace, doporučená hodnota závisí na počtu dimenzí daného problému, nejčastěji se pohybuje v rozmezí (20-100), ale pro extrémní případy může být i vyšší.
- $F$  – Mutační konstanta, původně byl doporučený rozsah stanoven na (0-2). Nejčastěji se ovšem používá hodnota (0,5-0,9).
- $CR$  – Práh křížení, pro většinu případů (0-1).

Algoritmus začíná vytvořením první náhodné populace. Poté se spustí hlavní cyklus algoritmu, který simuluje generační vývoj populace. V každém cyklu je z každého jedince pomocí operací mutace a křížení s ostatními jedinci vytvořen jeho potomek. Potomek je následně ohodnocen pomocí účelové funkce a je vybrán lepší z dvojice rodič – potomek, který postupuje do nové generace. Algoritmus je popsán níže následujícím pseudokódem [1].

---

**Algoritmus 1: Diferenciální evoluce**


---

```

1  inicializace populace  $P = (x_1, x_2, \dots, x_{NP})$ ,  $x_i$  je definován jako D-dimenzionální
vektor
2  while není dosažena ukončovací podmínka do
3    for  $i = 1$  až  $NP$  do
4      vytvoření zmutovaného jedince  $v_i$ 
5      zmutování jedince  $v_i$  do  $u_i$ 
6      if  $f(u_i) \leq f(x_i)$  then
7        vložit  $u_i$  do nové populace  $P_{new}$ 
8      else
9        vložit  $x_i$  do nové populace  $P_{new}$ 
10     end if
11   end for
12    $P := P_{new}$ 
13 end while

```

---

**2.1.1 Mutace**

Při tvorbě nového potomka v diferenciální evoluci jsou rodiče nejprve podrobeni mutaci. Jedná se o vektorovou operaci, která způsobí náhodnou změnu jeho vlastností. „Zmutovaný“ jedinec je pak oproti rodiči v prohledávaném prostoru posunut [1].

Pro mutaci se využívají různé strategie. Mění se zde na základě kolika jedinců je vytvořen potomek a jak jsou tito jedinci vybíráni [1].

Mutační strategie jsou zapsány vždy v následujícím tvaru:

$$m / n$$

- **m** – typ mutace
- **n** – počet jedinců potřebných pro mutaci

Základní strategií je strategie **DE/rand/1** (1). Ta funguje na následujícím principu:

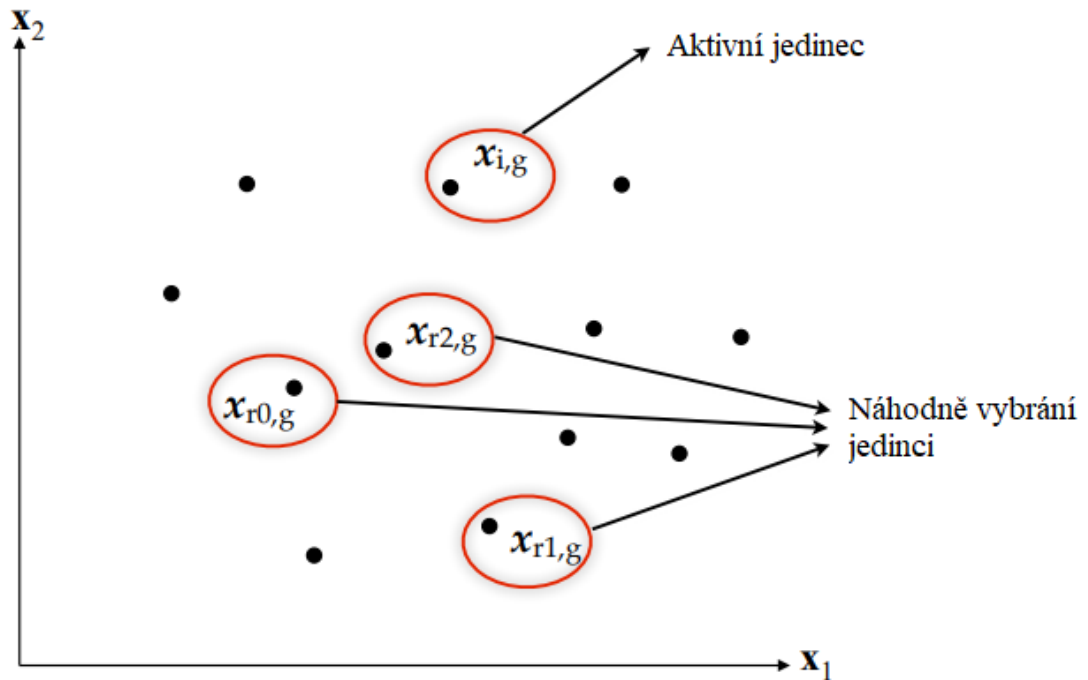
- Z populace jsou vybráni tři náhodní jedinci ( $x_{r0}$ ,  $x_{r1}$ ,  $x_{r2}$ ), kteří se liší od sebe a zároveň od aktivního jedince – rodiče ( $x_i$ ).

Vypočítá se „zmutovaný“ jedinec  $v_i$  pomocí vztahu:

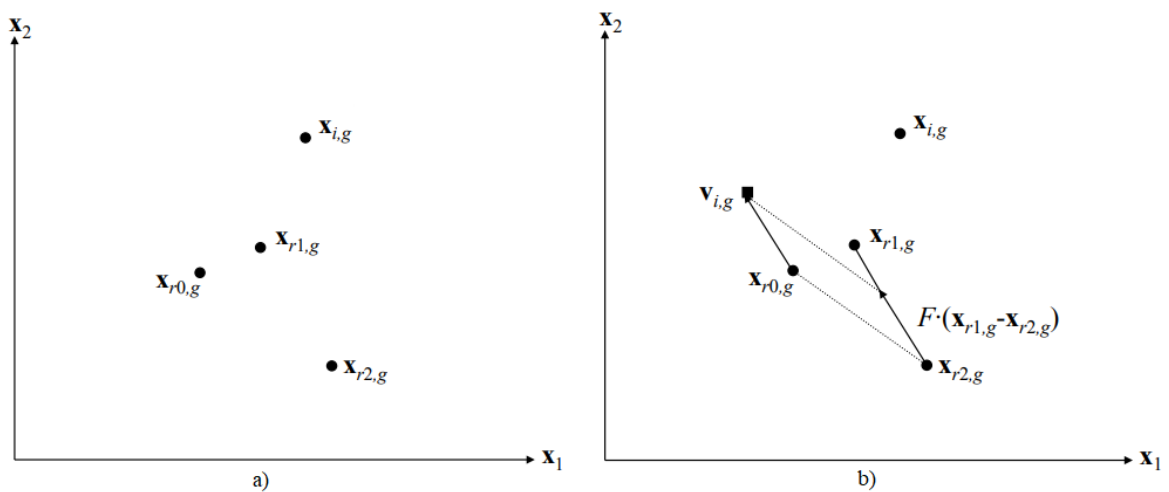
$$v_i = x_{r0} + F * (x_{r1} - x_{r2}) \quad (1)$$



Kde  $v_i$  je nově vzniklý „zmutovaný“ jedinec,  $F$  je mutační konstanta,  $(x_{r1} - x_{r2})$  tvoří diferenci mezi dvěma náhodnými jedinci a  $x_{r0}$  je náhodný jedinec (obr. 4.), ke kterému je vynásobená diference přičtena (obr. 5.) [1].



Obrázek 4. Populace ve dvou dimenzionálním problému a výběr jedinců pro strategii DE/rand/1



Obrázek 5. Na obrázku a) jsou znázorněni vybraní jedinci a na obrázku b) je znázorněno, jak je tvořen „zmutovaný“ potomek  $v_i$

### 2.1.1.1 Další mutační strategie

Tabulka 1. Mutační strategie [1]

| Označení             | Mutační vztah  |
|----------------------|--|
| DE/rand/1            | $v_i = x_{r0} + F * (x_{r1} - x_{r2})$ (2)   |
| DE/rand/2            | $v_i = x_{r1} + F(x_{r2} - x_{r3}) + F(x_{r4} - x_{r5})$ (3)                           |
| DE/best/1            | $v_i = x_{best} + F(x_{r1} - x_{r2})$ (4)  |
| DE/best/2            | $v_i = x_{best} + F(x_{r1} - x_{r2}) + F(x_{r3} - x_{r4})$ (5)                         |
| DE/current-to-best/1 | $v_i = x_i + K(x_{best} - x_i) + F(x_{r1} - x_{r2})$ (6)                               |
| DE/current-to-rand/1 | $v_i = x_i + K(x_{r1} - x_i) + F(x_{r2} - x_{r3})$ (7)                                 |
| DE/rand-to-best/1    | $v_i = x_i + F(x_{best} - x_{r1}) + F(x_{r2} - x_{r3})$ (8)                            |
| DE/rand-to-best/2    | $v_i = x_i + F(x_{best} - x_{r1}) + F(x_{r2} - x_{r3}) + F(x_{r4} - x_{r5})$ (9)       |
| DE/randrl/1          | $v_i = x_{rb} + F(x_{r2} - x_{r3})$ (10)<br>kde $x_{rb}$ je nejlepší z vybrané trojice |

### 2.1.2 Křížení

Při křížení v diferenciální evoluci dochází k sestavení nového jedince ze dvou základních vektorů – rodiče a nově vzniklého „zmutovaného“ potomka [1].

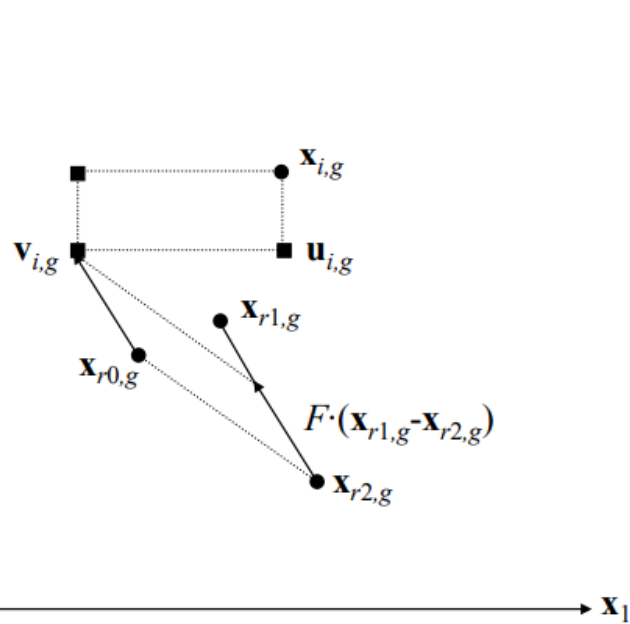
Existuje několik variant křížení, ale mezi nejčastěji používané se řadí binomiální (11) a exponenciální (13) křížení [1].

#### 2.1.2.1 Binomiální křížení

Tvorba výsledného jedince je závislá na parametru CR, který je jedním ze čtyř vstupních údajů pro diferenciální evoluci. Ten udává jestli bude do nově vzniklého jedince  $u_i$  zkopírován parametr z rodiče  $x_i$  nebo ze „zmutovaného“ potomka  $v_i$ .

Pro každý parametr výsledného jedince se vygeneruje pseudonáhodné číslo v rozmezí (0-1), které se porovná s hodnotou CR. Pokud je číslo menší nebo rovno CR, do výsledného jedince se zkopíruje hodnota z vektoru  $v_i$ . Jinak je zde zkopírována hodnota z rodiče  $x_i$  (obr. 6.). Aby se zajistilo, že nový jedinec bude mít alespoň jeden z parametrů od „zmutovaného“ jedince, je náhodně vybrán jeden index  $j$ , který se vždy zkopíruje z vektoru  $v_i$ .

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand(0,1) \leq CR \text{ or } j = rand(1,D) \\ x_{i,j}, & \text{jinak} \end{cases} \quad (11)$$



Obrázek 6. Binomiální křížení

Ve dvoudimenzionálním prostoru tedy mohou nastat tři možné varianty:

- Oba parametry pochází od „zmutovaného“ potomka.
- První parametr od potomka, druhý od rodiče.
- První parametr od rodiče, druhý od potomka.

### 2.1.2.2 Exponenciální křížení

Při exponenciálním křížení se nejprve pseudonáhodně určí ve vektoru pozice  $k$ , od které se bude začínat. Od této pozice algoritmus dojde do konce vektoru a pokračuje cyklicky od začátku, dokud není výsledný vektor nového jedince zaplněn.

Prvek, který je určen jako počáteční, je vždy zkopírován ze „zmutovaného“ potomka a u následujících prvku pravděpodobnost exponenciálně klesá.

$$CR^D - D * L * CR + D * L - 1 = 0 \quad (12)$$

Prvních  $L$  parametrů je tedy použito ze zmutovaného jedince a zbývající poskytuje rodič.

$$u_{i,j} = \begin{cases} u_{i,j}, & \text{for } j = k \bmod D, (k + 1) \bmod D, \dots, (k + L - 1) \bmod D \\ x_{i,j}, & \text{jinak} \end{cases} \quad (13)$$

### 2.1.3 Selekcce

Jakmile je sestaven nový jedinec, dochází k selekci mezi ním a jeho rodičem. Ta spočívá v ohodnocení nového potomka pomocí účelové funkce a porovnání s funkční hodnotou rodiče. Kvalitnější z nich následně postupuje do nové generace [1].

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G}, & \text{if } f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G}, & \text{jinak} \end{cases} \quad (14)$$

Z výběru nového jedince je vidět, že diferenciální evoluce je čistě elitistický algoritmus. Do nové generace se vždy dostane jen jedinec, který je stejně dobrý nebo lepší než jeho předchůdce. Při výběru je vybrán nový jedinec, i když je stejně kvalitní jako jeho rodič. To zaručí posun po dané hyperploše v případě, že je extrém obklopen rovnou plochou [1,2].

## 2.2 Varianty diferenciální evoluce

Vzhledem k tomu, že diferenciální evoluce má několik různých mutačních strategií a možnost libovolně měnit vstupní parametry  $F$  a  $CR$ , existuje velké množství kombinací, jak algoritmus nastavit. Jelikož je hledání vhodného nastavení parametrů časově náročné, vznikly nové adaptivní verze algoritmu, které v průběhu algoritmu mohou podle potřeby měnit hodnoty  $F$  a  $CR$ , mutační strategii nebo velikost populace.

Díky tomu, že tyto algoritmy mají možnost své parametry adaptivně měnit, mohou je upravit tak, aby bylo v určité fázi upřednostňováno prohledávání prostoru před konvergencí k extrému. Další z nových možností je samostatné učení, kde algoritmus zpočátku zkouší parametry volit náhodně a podle úspěšnosti se učí, jaké hodnoty jsou pro daný problém vhodné.

### 2.2.1 DE s propagací řídicích parametru – jDE

Ve verzi **jDE** má každý jedinec kromě svých parametrů navíc uloženy i vlastní hodnoty  $F$  a  $CR$ . Před začátkem mutace jsou tyto parametry s 10% pravděpodobností náhodně změněny. Jestliže byl nový jedinec úspěšný, do nové generace postupuje i s novými hodnotami  $F$  a  $CR$ . V opačném případě jsou nové hodnoty zapomenuty a do nové generace postupuje rodič se svými původními hodnotami.

Verze **jDE** je navržena pouze pro mutační strategii **DE/rand/1** s náhodnou inicializací  $F$  a  $CR$  pro každého jedince. Při generování nových parametrů před mutací jsou nové hodnoty

vybírány při rovnoměrném rozložení pravděpodobnosti z intervalů  $F \in (0,1; 0,9)$  a  $CR \in (0;1)$ .

### 2.2.2 DE s adaptací strategie – SaDE

Tato verze algoritmu DE je již více propracovaná a sofistikovaná. Algoritmus během svého běhu má možnost vybírat ze čtyř možných strategií a podle úspěšnosti jednotlivých strategií se učí, která z nich je pro daný problém nejvhodnější. Každý jedinec v populaci má svoji vlastní mutační konstantu  $F$ . Práh křížení je naopak přiřazen podle zvolené strategie.

Strategie používané v SaDE:

- DE/rand/1/bin
- DE/rand/2/bin
- DE/rand-to-best/2/bin
- DE/current-to-rand/1/bin

Zpočátku mají všechny čtyři strategie stejnou pravděpodobnost použití  $p_k = 1/4$ . Algoritmus poté během svého běhu ukládá po určitou dobu úspěšnost jednotlivých strategií a po zadaném počtu generací upraví pravděpodobnost pro jejich generování podle vztahů:

$$p_k = \frac{S_k}{\sum_{j=1}^4 (S_j)} \quad (15)$$

$$S_k = \frac{succ_k}{succ_k + fail_k} + \varphi \quad (16)$$

Kde  $succ_k$  a  $fail_k$  jsou četnosti úspěšnosti/neúspěšnosti  $k$ -té strategie ve sledovaném období. Konstanta  $\varphi = 0,01$  se do výpočtu přidává, aby žádná pravděpodobnost neměla nulovou hodnotu.

Doporučené množství sledovaných generací pro učení je 50.

### 2.2.3 Adaptivní DE s volitelným externím archivem – JADE

Varianta JADE dodává do algoritmu DE tři vylepšení:

- Novou strategií: DE/current-to-pbest/1.
- Adaptivní nastavení pro parametry  $CR$  a  $F$  u každého jedince.

- Volitelný archiv pro starší vyřazené jedince. Pokud je použit, využívá se při výběru jedinců pro mutaci, kde jsou jedinci vybíráni nejen ze současné generace, ale i z archivu.

Díky přidání archivu dokáže algoritmus odstranit problémy s konvergencí k lokálnímu extrému.

Nová mutační strategie je dána vztahem:

$$\mathbf{u}_i = \mathbf{x}_i + F(\mathbf{x}_{pbest} - \mathbf{x}_i) + F(\mathbf{x}_{r1} - \mathbf{x}_{r2}) \quad (17)$$

Kde hodnota  $\mathbf{x}_{pbest}$  se vybírá z 100*p* % nejlepších jedinců populace, *p* je voleno z intervalu (0.05-0.20),  $\mathbf{x}_{r1}$  je náhodně vybíraný jedinec z aktuální populace,  $\mathbf{x}_{r2}$  je náhodný jedinec vybraný ze spojené aktuální populace a archivu. Jedinci  $\mathbf{x}_{r1}, \mathbf{x}_{r2}$  musí být odlišní od aktuálního jedince  $\mathbf{x}_i$  a zároveň jeden od druhého.

Archiv je na počátku prázdný a postupně jsou do něj přidáváni rodiče, kteří byli nahrazeni svými potomky. Velikost archivu je stejně velká jako je velikost populace a při jeho zaplnění je při dalším vložení nahrazen náhodný starší jedinec.

#### 2.2.4 DE se sadou řídicích parametrů a strategií – EPSDE

V této variantě DE má každý jedinec náhodně přiřazené hodnoty {*strategie*, *F*, *CR*}. Tyto hodnoty jsou vybírány z předem definované sady parametrů.

- Strategie: {DE/best/2/bin, DE/rand/1/bin, DE/current-to-rand/1/bin}
- $F = \{0,4; 0,5; 0,6; 0,7; 0,8; 0,9\}$
- $CR = \{0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9\}$

Algoritmus neobsahuje žádný učící mechanismus. Parametry pro nového potomka se přenáší z jeho rodiče. Pokud je nový jedinec horší, rodič přechází do nové generace a dochází u něho k vygenerování nové sady řídicích parametrů.

#### 2.2.5 JADE s adaptivitou parametrů založenou na historicky úspěšných hodnotách – SHADE

Jedná se o rozšíření JADE varianty algoritmu. SHADE verze vychází z anglického názvu „Success-History based Adaptive DE“. Pokud dojde k úspěšné tvorbě nového potomka, ukládají se zde kromě starších vyřazených jedinců i hodnoty *F* a *CR*. Mutační strategie zde

zůstala stejná jako v původní JADE verzi, jen se zde změnil výběr parametru  $p$ , který se nyní náhodně generuje z intervalu  $(p_{min}, 0,2)$ . Hodnota  $p_{min}$  se pak nejčastěji určuje podle velikosti populace vztahem  $p_{min} = 2/NP$ .

### 2.2.6 SHADE s redukcí velikosti populace – L-SHADE

L-SHADE je dalším rozšířením pro verzi SHADE. Ta přichází s lineární redukcí velikosti populace. Velikost populace je v závislosti na množství zbývajících ohodnocení po jednotlivých generacích snižována. Při snížení velikosti populace dojde k odstranění jedince s nejhorším ohodnocením. Snižování velikosti končí, pokud se velikost populace dostane na minimální počet jedinců potřebných pro tvorbu potomka, tedy na čtyři nebo na zadanou hodnotu.

Pro výpočet velikosti populace se využívá vztah:

$$NP = \text{round} \left[ NP_{min} + FES \left( \frac{NP_{init} - NP_{min}}{maxFES} \right) \right] \quad (18)$$

Kde  $NP_{min}$  je minimální velikost populace,  $NP_{init}$  je počáteční velikost populace,  $FES$  je počet proběhlých ohodnocení a  $maxFES$  je maximální počet ohodnocení.

### 3 DIFERENCIÁLNÍ EVOLUCE VE VARIANTĚ DISH

Tato varianta diferenciální evoluce je založena na myšlence prohledat co největší oblast prohledávaného prostoru a předejít tím předčasné konvergenci. Označení DISH vychází z anglického názvu algoritmu „**D**istance Based Parameter Adaptation for **S**uccess-**H**istory based Differential Evolution“. Předchozí algoritmy mají sklon k rychlé konvergenci k lokálnímu extrému a z toho důvodu mohou minout extrém globální. Algoritmus je založen na dříve popsaných verzích DE, které rozšiřuje o nové možnosti [2].

#### 3.1 Vznik varianty DISH

Diferenciální evoluce ve variantě DISH vychází z varianty L-SHADE, která je popsána v předchozí kapitole.

Následujícím rozšířením byla verze jSO, která do algoritmu přidává váhově založenou mutaci. Tato verze zpočátku omezuje hodnotu parametru  $F$  a s rostoucím počtem provedených ohodnocení se tato hodnota zvyšuje [2].

Posledním rozšířením, které přidává algoritmus DISH je ohodnocení vzdálenosti. Výpočet vzdálenosti je zde založen na Euklidovské vzdálenosti dvou bodů – rodiče a potomka. Vyšší vzdálenost je zde lepší, protože dochází k prohledávání větší oblasti. Tato hodnota je následně přepočítána na váhu a nahrazuje výpočet váhy používaný v jSO. Pomocí této váhy a uložených hodnot  $F$  a  $CR$  z úspěšných jedinců jsou následně dopočítány nové hodnoty  $F$  a  $CR$ . Tyto nové hodnoty  $F$  a  $CR$  jsou uloženy do odpovídajících pamětí a jsou používány v následujících generacích. [2].



### 3.2 Algoritmus DISH

---

#### Algoritmus 2: DISH

---

|   |  |
|---|--|
| <pre> 1  Nastavit <math>NP_{init}, NP_f, H</math> a ukončovací    podmínku 2  <math>NP = NP_{init}, G = 0, x_{best} = \{\}, k =</math>    <math>1, p_{min} = 2/NP, A = \emptyset</math> 3  Náhodně inicializovat populaci <math>P =</math>    <math>(x_{1,G}, \dots, x_{NP,G})</math> 4  Nastavit všechny hodnoty <math>M_F = 0,5</math>    a <math>M_{CR} = 0,8</math> 5  <math>P_{new} = \{\}, x_{best} =</math> nejlepší    z populace <math>P</math> 6  <b>while</b> není dosažena ukončovací    podmínka <b>do</b> 7    <math>S_F = \emptyset, S_{CR} = \emptyset</math> 8    <b>for</b> <math>i = 1</math> to <math>NP</math> <b>do</b> 9      <math>r = U[1, H]</math> 10     <b>if</b> <math>r = H</math> <b>then</b> 11       <math>M_{F,r} = 0,9</math> 12       <math>M_{CR,r} = 0,9</math> 13     <b>end if</b> 14     <math>CR_{i,G} = N(M_{CR,r}; 0,1)</math> 15     <b>if</b> <math>CR_{i,G} &lt; 0</math> <b>then</b> 16       <math>CR_{i,G} = 0</math> 17     <b>else if</b> <math>CR_{i,G} &gt; 1</math> <b>then</b> 18       <math>CR_{i,G} = 1</math> 19     <b>end if</b> 20     <math>F_{i,G} = C(M_{F,r}; 0,1)</math> 21     <b>if</b> <math>G &lt; 0,6G_{MAX}</math> a <math>F_{i,G} &gt; 0,7</math> <b>then</b> 22       <math>F_{i,G} = 0,7</math> 23     <b>end if</b> 24     <b>if</b> <math>G &lt; 0,25G_{MAX}</math> <b>then</b> 25       <math>CR_{i,G} = \max(CR_{i,G}; 0,7)</math> 26     <b>else if</b> <math>G &lt; 0,5G_{MAX}</math> <b>then</b> 27       <math>CR_{i,G} = \max(CR_{i,G}; 0,6)</math> 28     <b>end if</b> 29     <math>x_{i,G} = P_i</math> 30     <math>v_{i,G}</math> pomocí mutace 31     <math>u_{i,G}</math> pomocí křížení 32     <b>if</b> <math>f(u_{i,G}) \leq f(x_{i,G})</math> <b>then</b> 33       <math>x_{i,G+1} = u_{i,G}</math> 34     <b>else</b> 35       <math>x_{i,G+1} = x_{i,G}</math> 36     <b>end if</b> 37     <b>if</b> <math>f(u_{i,G}) \leq f(x_{i,G})</math> <b>then</b> </pre> | <pre> 38       <math>x_{i,G} \rightarrow A</math> 39       <math>F_i \rightarrow S_F, CR_i \rightarrow S_{CR}</math> 40     <b>end if</b> 41     <b>if</b> <math> A  &gt; NP</math> <b>then</b> 42       Smazat náhodně <math> A  - NP</math>    jedinců z <math>A</math> 43     <b>end if</b> 44     <math>x_{i,G+1} \rightarrow P_{new}</math> 45   <b>end for</b> 46   Výpočet <math>NP_{new}</math> 47   <b>if</b> <math>NP_{new} &lt; NP</math> 48     Seřadit jedince v <math>P</math> podle    účelové hodnoty a odstranit <math>NP - NP_{new}</math>    nejhorších 49   <b>end if</b> 50   <math>NP = NP_{new}</math> 51   <b>if</b> <math> A  &gt; NP</math> <b>then</b> 52     Smazat náhodně <math> A  - NP</math>    jedinců z <math>A</math> 53   <b>end if</b> 54   <b>if</b> <math>S_F \neq \emptyset</math> a <math>S_{CR} \neq \emptyset</math> <b>then</b> 55     Aktualizace paměti <math>M_F</math> a    <math>M_{CR}, k++</math> 56     <b>if</b> <math>k &gt; H</math> <b>then</b> 57       <math>k = 1</math> 58     <b>end if</b> 59   <b>end if</b> 60   <math>P = P_{new}, P_{new} = \{\}, x_{best} =</math>    nejlepší z populace <math>P, G++</math> 61 <b>end while</b> 62 <b>return</b> <math>x_{best}</math> jako nejlepší řešení </pre> |
|---|--|

### 3.2.1 Inicializace

Při spuštění algoritmu dochází nejprve k nastavení potřebných parametrů a inicializaci proměnných.

- 1 Nastavit  $NP_{init}$ ,  $NP_f$ ,  $H$  a ukončovací podmínku
- 2  $NP = NP_{init}$ ,  $G = 0$ ,  $x_{best} = \{\}$ ,  $k = 1$ ,  $p_{min} = 2/NP$ ,  $A = \emptyset$
- 3 Náhodně inicializovat populaci  $P = (x_{1,G}, \dots, x_{NP,G})$
- 4 Nastavit všechny hodnoty  $M_F = 0,5$  a  $M_{CR} = 0,8$
- 5  $P_{new} = \{\}$ ,  $x_{best} = \text{nejlepší z populace } P$

Obrázek 7. Inicializace parametrů [2,3]

Proměnná  $NP$  představuje velikost populace,  $NP_{init}$  je počáteční velikost a doporučuje se nastavit ji pomocí vztahu:

$$NP_{init} = 20\sqrt{D} \log D \quad (19)$$

$NP_f$  představuje minimální velikost populace a nejmenší hodnotu, kterou lze použít je  $NP_f = 4$ , protože potřebujeme 4 rozdílné jedince pro provedení mutace.  $H$  označuje velikost paměti  $M_F$  a  $M_{CR}$ , ve kterých jsou uloženy úspěšné hodnoty  $F$  a  $CR$  z předchozích generací [2,3].

Na 2. řádku dochází k nastavení velikosti počáteční populace  $NP$ , nastavení indexů pro generace  $G$  a aktuální pozici v paměti  $k$ . Je zde vytvořena proměnná  $x_{best}$  pro ukládání nejlepšího nalezeného řešení a paměť pro ukládání starších vyřazených jedinců – archiv  $A$ . Následně dochází k vytvoření první náhodné populace  $P$  a nastavení inicializačních hodnot pro všechny prvky v pamětech  $M_F = 0,5$  a  $M_{CR} = 0,8$ . Populace je nejčastěji vygenerována pomocí pseudo-náhodného generování čísel v rozsahu specifikovaném pro dané optimalizované parametry [2,3].

V poslední části dochází k ohodnocení vytvořené populace, uložení nejlepšího jedince do  $x_{best}$  a vytvoření prázdného pole pro novou generaci  $P_{new}$  [2,3].

### 3.2.2 Nastavení F a CR

Nyní dochází ke spuštění hlavní smyčky algoritmu.

```

6   while není dosažena ukončovací podmínka do
7      $S_F = \emptyset, S_{CR} = \emptyset$ 
8     for  $i = 1$  to  $NP$  do
9        $r = U[1, H]$ 
10      if  $r = H$  then
11         $M_{F,r} = 0,9$ 
12         $M_{CR,r} = 0,9$ 
13      end if

```

Obrázek 8. Načtení paměti [2,3]

Na začátku každé generace dochází k vynulování pomocné paměti  $S_F$  a  $S_{CR}$ , které slouží pro ukládání úspěšných hodnot  $F$  a  $CR$  v rámci jedné generace. Na základě těchto dvou pomocných pamětí jsou na konci každé generace aktualizovány odpovídající buňky paměti  $M_F$  a  $M_{CR}$  [2,3].

Pro každého jedince se poté vygeneruje náhodné celé číslo  $r$ , které představuje index v pamětech  $M_F$  a  $M_{CR}$ . Pokud dojde k vybrání posledního prvku v pamětech, dochází k jeho nahrazení hodnotou  $M_F = M_{CR} = 0,9$ , jinak jsou použity hodnoty z odpovídající paměti [2,3].

```

14       $CR_{i,G} = N(M_{CR,r}; 0,1)$ 
15      if  $CR_{i,G} < 0$  then
16         $CR_{i,G} = 0$ 
17      else if  $CR_{i,G} > 1$  then
18         $CR_{i,G} = 1$ 
19      end if

```

Obrázek 9. Generování  $CR$  [2,3]

Nejprve je vygenerována nová hodnota  $CR$  pomocí normálního rozdělení se střední hodnotou odpovídající hodnotě  $M_{CR}$  a rozptylem rovným  $0,1$ . Pokud by vygenerovaná hodnota  $CR$  byla mimo rozsah  $CR \in \langle 0,1 \rangle$  dojde k nastavení  $CR$  na odpovídající hraniční hodnoty  $CR = 0$  nebo  $CR = 1$  [2,3].

```

20       $F_{i,G} = C(M_{F,r}; 0,1)$ 
21      if  $G < 0,6G_{MAX}$  a  $F_{i,G} > 0,7$  then
22         $F_{i,G} = 0,7$ 
23      end if

```

Obrázek 10. Generování  $F$  [2,3]

Pro generování hodnoty  $F$  se využívá náhodné číslo generované pomocí Cauchyho rozdělení a stejně jako u generování  $CR$  je zde hodnota paměti použita jako střední hodnota [2,3].

$$F_i = C[M_F; 0,1] \quad (20)$$

Pro zpomalení konvergence k lokálním extrémům je hodnota  $F$  v prvních 60 % generací nastavována maximálně na hodnotu  $F = 0,7$  [2,3].

```

24     if  $G < 0,25G_{MAX}$  then
25          $CR_{i,G} = \max(CR_{i,G}; 0,7)$ 
26     else if  $G < 0,5G_{MAX}$  then
27          $CR_{i,G} = \max(CR_{i,G}; 0,6)$ 
28     end if

```

Obrázek 11. Úprava  $CR$  [2,3]

Posledním krokem je úprava  $CR$ . Tento parametr je pro prvních 25 % generací nastaven minimálně na  $CR = 0,7$  a poté na dalších 25 % minimálně na  $CR = 0,6$ . Díky tomu jsou parametry pro nové potomky vybírány spíše ze „zmutovaných“ jedinců a dochází tak k většímu průzkumu prohledávané hyperplochy [2,3].

### 3.2.3 Mutace

Mutace ve variantě DISH využívá stejné mutační schéma jako varianta jSO, tedy:

$$\mathbf{v}_i = \mathbf{x}_i + F_W(\mathbf{x}_{pbest} - \mathbf{x}_i) + F(\mathbf{x}_{r1} - \mathbf{x}_{r2}) \quad (21)$$

Kde  $\mathbf{v}_i$  je nově vzniklý „zmutovaný“ jedinec,  $\mathbf{x}_i$  je jedinec z aktuální populace – rodič. Jedinec  $\mathbf{x}_{r1}$  je vybrán náhodně z aktuální populace,  $\mathbf{x}_{r2}$  se vybírá náhodně ze sloučené aktuální populace a archivu [2,3].

Hodnota  $F_W$  je počítána podle vztahu:

$$F_W = \begin{cases} 0,7F, & FES < 0,2 \max FES \\ 0,6F, & FES < 0,4 \max FES \\ 1,2F, & jinak \end{cases} \quad (22)$$

Kde  $FES$  je počet provedených ohodnocení funkce a  $\max FES$  je maximální povolený počet ohodnocení [2,3].

Jedinec  $\mathbf{x}_{pbest}$  je vybrán náhodně z  $NP \cdot p$  nejlepších jedinců aktuální populace. Hodnota  $p$  je počítána podle vztahu:

$$p = p_{min} + \frac{FES}{\max FES} (p_{max} - p_{min}) \quad (23)$$

Kde  $p_{max} = 0,25$ ,  $p_{min}$  se počítá jako  $p_{min} = 2/NP$  a poměr  $FES/\max FES$  udává kolik ohodnocení účelové funkce už proběhlo [2,3].

### 3.2.4 Křížení

Varianta DISH využívá binomiální křížení.

Nejprve je náhodně vybrán jeden parametr, který se překopíruje vždy z „mutanta“. Pro ostatní parametry platí klasické binomiální křížení, kde je vygenerováno náhodně číslo, které je porovnáno s hodnotou  $CR$ . Pokud je číslo menší než hodnota  $CR$ , je použit parametr z „mutanta“, jinak je dosazena hodnota z rodiče [2,3].

$$\mathbf{u}_{j,i} = \begin{cases} \mathbf{v}_{j,i}, & U[0,1] \leq CR, \text{ nebo } j = j_{rand} \\ \mathbf{x}_{j,i}, & \text{jinak} \end{cases} \quad (24)$$

### 3.2.5 Selektce a aktualizace archivu

Selektce je v pseudokódu popsána řádky 32-36 a jedná se o ohodnocení potomka účelovou funkcí a porovnání s funkční hodnotou rodiče. Následně je vybrán lepší z nich [2,3].

```

37     if  $f(u_{i,G}) \leq f(x_{i,G})$  then
38          $x_{i,G} \rightarrow A$ 
39          $F_i \rightarrow S_F, CR_i \rightarrow S_{CR}$ 
40     end if
41     if  $|A| > NP$  then
42         Smazat náhodně  $|A| - NP$  jedinců z  $A$ 
43     end if

```

Obrázek 12. Aktualizace archivu [2,3]

Pokud dojde k výběru potomka a do nové generace se tedy dostává nový jedinec, dochází k uložení rodiče do archivu, pokud je archiv použit. Velikost archivu se doporučuje stejná jako je velikost aktuální populace. Když je zde volné místo, je rodič vložen zde. Je-li už archiv zaplněn, dochází k náhodnému přepsání některého ze starších jedinců. Zároveň s uložení rodiče do archivu jsou uloženy hodnoty  $F$  a  $CR$ , které vedly ke tvorbě nového jedince do odpovídajících pomocných pamětí  $S_F$  a  $S_{CR}$  [2,3].

### 3.2.6 Snížení velikosti populace

```

46  Výpočet  $NP_{new}$ 
47  if  $NP_{new} < NP$ 
48      Seřadit jedince v P podle účelové hodnoty a odstranit  $NP - NP_{new}$  nejhorších
49  end if
50   $NP = NP_{new}$ 
51  if  $|A| > NP$  then
52      Smazat náhodně  $|A| - NP$  jedinců z A
53  end if

```

Obrázek 13. Změna velikosti populace [2,3]

Po výpočtu celé generace dochází k aktualizaci velikosti populace. Velikost nové populace je vypočítána vztahem:

$$NP_{new} = round \left( NP_{init} - \frac{FES}{maxFES} * (NP_{init} - NP_f) \right) \quad (25)$$

Kde  $NP_{new}$  představuje velikost nové populace,  $NP_{init}$  udává velikost populace na počátku,  $NP_f$  udává koncovou velikost populace a poměr  $FES/maxFES$  udává kolik ohodnocení účelové funkce už proběhlo [2,3].

Pokud dojde ke snížení velikosti populace, aktuální populace je seřazena podle hodnoty účelové funkce a jsou odstraněni nejhorší jedinci. Zároveň musí dojít k úpravě velikosti archivu. Pokud tento stav nastane, jsou z archivu odstraněny náhodné prvky tak, aby jeho velikost odpovídala velikosti populace [2,3].

### 3.2.7 Aktualizace paměti $M_F$ a $M_{CR}$

```

54  if  $S_F \neq \emptyset$  a  $S_{CR} \neq \emptyset$  then
55      Aktualizace paměti  $M_F$  a  $M_{CR}$ ,  $k++$ 
56      if  $k > H$  then
57           $k = 1$ 
58      end if
59  end if

```

Obrázek 14. Aktualizace paměti [2,3]

Pokud během generace došlo k uložení hodnot do pomocných pamětí  $S_F$  a  $S_{CR}$ , dochází k aktualizaci hodnoty v odpovídajících pamětech  $M_F$  a  $M_{CR}$  [2,3].

Z hodnot uložených v pomocných pamětech  $S_F$  a  $S_{CR}$  je vypočítán vážený Lehmerův průměr:

$$M_F = \frac{\sum_{k=1}^{|S|} w_k \cdot S_{F,k}^2}{\sum_{k=1}^{|S|} w_k \cdot S_{F,k}} \quad (26)$$

$$M_{CR} = \frac{\sum_{k=1}^{|S|} w_k \cdot S_{CR,k}^2}{\sum_{k=1}^{|S|} w_k \cdot S_{CR,k}} \quad (27)$$

$$w_k = \frac{\sqrt{\sum_{j=1}^D (u_{k,j} - x_{k,j})^2}}{\sum_{m=1}^{|S|} \sqrt{\sum_{j=1}^D (u_{m,1} - x_{m,1})^2}} \quad (28)$$

Kde vztahy (26) a (27) představují vážený Lehmerův průměr a vztah (28) představuje výpočet váhy na základě poměru vzdálenosti nového jedince od jeho rodiče vůči součtu vzdáleností všech takto vzniklých dvojic v dané generaci [2,3].

Vypočtená hodnota  $F$  a  $CR$  je poté uložena na  $k$ -tý index v odpovídající paměti. Po zapsání hodnot na poslední pozice v pamětech, dochází k přesunu na první prvek a jejich cyklickému přepisování [2,3].

### 3.2.8 Aktualizace populace

Posledním krokem algoritmu je nahrazení staré populace za novou a aktualizace nejlepšího nalezeného řešení  $x_{best}$  [2,3].

## 4 METODY DISKRETIZACE

Algoritmus diferenciální evoluce je v základní verzi navržen pro řešení spojitých optimalizačních úloh. Pokud jej chceme použít pro řešení diskretních problémů, je třeba algoritmus upravit [4,5].

V případě, že vygenerujeme jedince s diskretními parametry, mutační mechanismus diferenciální evoluce způsobí přeměnu diskretních hodnot na spojitě. K tomu dochází z důvodu násobení mutační konstantou, která má tvar desetinného čísla. Proto je potřeba provést potřebné úpravy během mutačního procesu [4,5].

### 4.1 Zaokrouhlení

Nejjednodušším způsobem, jak zachovat diskretní řešení po mutaci, je vzniklé hodnoty zaokrouhlit. Toto řešení je sice jednoduché ale zanáší do evolučního procesu určitou chybu, která může ovlivnit průběh výpočtu i výsledné nalezené řešení. Při použití tohoto řešení diskretizace máme dvě možnosti, kde zaokrouhlení provést.

#### 4.1.1 Zaokrouhlení v populaci

Prvním možným řešením je zaokrouhlit hodnoty ihned po provedení mutace. Tohle řešení ovšem omezí prohledávaný prostor pouze na diskretní množinu pozic a algoritmus ztratí část své robustnosti. Může zde nastat situace, kde při pomalém posunu v prohledávaném prostoru (v rámci desetin) nás bude zaokrouhlování vracet pořád do výchozího bodu [8].

#### 4.1.2 Zaokrouhlení při ohodnocení

Druhým řešením, jak provádět zaokrouhlování, je nechat jedince v populaci ve spojitě podobě a zaokrouhlení provádět pouze při dosazení do účelové funkce. Tento nepatrný rozdíl způsobí, že algoritmus neztratí na své diverzibilitě ani robustnosti a i při pomalém posunu v prohledávaném prostoru může dojít po několika generacích ke změně hodnoty [8].

### 4.2 Transformace čísel

Při použití této metody dochází ke změně celočíselných hodnot na reálné a po provedení mutace v reálné oblasti dochází ke zpětné transformaci na celočíselné hodnoty [4].



#### 4.2.1 Dopředná transformace

Dopředná transformace je použita před mutací pro získání reálných čísel. Je formulována jako

$$x'_i = -1 + \frac{x_i h 5}{10^3 - 1} \quad (29)$$

Kde  $x'_i$  je transformovaná hodnota,  $x_i$  je vstupní celočíselná hodnota a  $h = 100$  je konstanta, která určuje přesnost vygenerované proměnné [4].

#### 4.2.2 Zpětná transformace

Po uskutečnění mutace je provedena zpětná transformace, kde jsou hodnoty navraceny do celočíselné oblasti.

$$\text{int}[x'_i] = \frac{(1 + x'_i)(10^3 - 1)}{5h} \quad (30)$$

Po zpětné transformaci dojde k zaokrouhlení čísla na nejbližší celočíselnou hodnotu [4].

### 4.3 Diskrétní mutace

Při použití diskrétní mutace se místo násobení mutační konstantou  $F$  vygeneruje náhodné číslo  $r$ , které se porovná s mutační konstantou. Pro mutační schéma DE/rand/1:

$$\mathbf{u}_i = \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (31)$$

je postup následující [5].

Nejprve je vypočítána diference mezi náhodnými jedinci  $\mathbf{d} = (\mathbf{x}_{r2} - \mathbf{x}_{r3})$ . Následně se vygeneruje pro každou dimenzi náhodné číslo  $r$  v rozsahu  $r = \langle 0,1 \rangle$ , které se porovná s mutační konstantou  $F$ . Pokud je náhodné číslo menší než mutační konstanta, hodnota v dané dimenzi zůstává. V opačném případě je hodnota vynulována [5].

$$\delta_i = \begin{cases} d_i, & \text{if } \text{rand}(0,1) < F \\ 0, & \text{jinak} \end{cases} \quad (32)$$

Tím vznikne dočasný vektor  $\delta$ , který se přičte k náhodnému jedinci  $\mathbf{x}_{r1}$ , čímž vznikne výsledný „zmutovaný“ jedinec  $\mathbf{u}_i$  [5].

Tato metoda přichází i s řešením „za-hraničních“ stavů, kdy nový jedinec může být mimo rozsah povolených hodnot. Pro ošetření je zde použita cyklická metoda, kde po maximální povolené hodnotě následuje nejnižší povolená hodnota [5].



## 5 TESTOVÁNÍ EVOLUČNÍCH ALGORITMŮ

Aby bylo možné určit, jestli a jak dobře algoritmus funguje, využívají se testovací funkce. Tyto funkce mají známý předpis a většinou u nich předem známe polohu globálního extrému. Podle výsledku tedy dokážeme říct, jestli algoritmus dokázal tento extrém najít. Tyto funkce jsou často sestavené tak, aby obsahovaly různé patologie a nelineárnosti [6].

Díky zjištěným výsledkům můžeme najít slabá místa navrženého algoritmu, které nás můžou navést, jak algoritmus vylepšit [6].

Další důležitou úlohu hrají testovací funkce při hledání správného nastavení parametrů. Kde na jedné vybrané funkci zkoušíme algoritmus několikrát pouštět s různým nastavením parametrů a podle získaných výsledků určujeme, která kombinace byla nejlepší [6].

Samotné testování je většinou prováděno na jedné funkci několikrát, abychom měli dostatečný počet dat pro objektivní posouzení. Důvodem pro opakované spouštění je náhodná složka evolučních algoritmů, která způsobuje unikátní běhy a z jednoho průběhu tedy nelze objektivně posoudit, jak je algoritmus úspěšný.

## 6 GENETICKÝ ALGORITMUS

Genetické algoritmy patří mezi jedny z nejznámějších evolučních algoritmů. Princip, na jakém fungují, je inspirován biologickým křížením a mutací genů, která nastává při reprodukci. Většina termínů, které jsou při GA používány jsou tedy převzaty z biologické evoluce. Základní myšlenkou je, že při vzniku nového potomka jsou použity geny rodičů, které se určitým způsobem smíchají a občas náhodně zmutují [1,13].

| v přírodě      | v genetických algoritmech   |
|----------------|---|
| jedinec        | řetězec symbolů, např. $ind = (1001)$   |
| přírodní výběr | výběr podle hodnotící funkce $f(ind)$   |
| křížení        | kombinace dvou řetězců<br>např. $(000   1) + (101   0) \rightarrow (0000) + (1011)$ |
| mutace         | náhodná záměna 0 a 1 v řetězci<br>např. $(0010) \rightarrow (1010)$                 |

Obrázek 16. Genetický algoritmus – princip [13]

Na obrázku 16. jsou zachyceny základní operace, které jsou u genetického algoritmu použity. Každý jedinec je tvořen řetězcem symbolů. Nejčastěji se jedná o čísla v binární podobě. Z populace jsou při křížení vybráni dva jedinci a z nich se stávají rodiče. Jedinci s lepší hodnotou účelové funkce mají větší pravděpodobnost, což simuluje přírodní výběr – přežijí pouze nejsilnější. Tito rodiče jsou následně zkříženi, jak je naznačeno na obrázku 16. Křížení může být jedno nebo více bodové. Posledním krokem při tvorbě nových potomků je mutace. Ta vyžaduje, aby byli jedinci zachyceni v binární podobě. Dochází při ní totiž k náhodnému překlopení jednoho nebo více bitů. Poté už následuje selekce, která je podobná jako u diferenciální evoluce [1,13].

## **II. PRAKTICKÁ ČÁST**

## 7 WOLFRAM MATHEMATICA

Algoritmus byl implementován v prostředí Mathematica, kde byl následně testován.

Mathematica je výkonné vývojové prostředí, využívané hlavně ve vývoji a testování nových algoritmů. Mathematica je široce využívána ve vědecké oblasti z důvodu své technické zdatnosti a snadného používání. Poskytuje uživateli jedno integrované prostředí, které je již 30 let neustále rozšiřováno o nové funkce. Prostředí aktuálně obsahuje téměř 5000 vestavěných funkcí, které pokrývají všechny technické oblasti. Mathematica je navržena tak, aby poskytovala možnost pracovat s robustními a efektivními algoritmy schopnými zvládnout rozsáhlé problémy [9].

Psaní kódu v prostředí Mathematica je velmi intuitivní díky anglickým názvům funkcí a vestavěné nápovědě, která kromě popisu jednotlivých funkcí nabízí i interaktivní příklady jejich použití. Mathematica využívá interpretovaný jazyk a kódy se vyznačují vysokou mírou abstrakce [9].

Při zpracování velkého počtu dat je Mathematica ideálním nástrojem, jelikož dokáže v jednom notebooku kombinovat spustitelný kód i dynamicky vykreslované grafy [9].

## 8 TESTOVÁNÍ DIFERENCIÁLNÍ EVOLUCE VE VARIANTĚ DISH

Pro testování bylo vybráno 8 benchmarkových funkcí, na kterých byl algoritmus několiknásobně spuštěn s následujícími parametry:

- Dimenze: 10/30/100
- Počáteční populace: 80/200/500
- Konečná populace: 4
- Velikost paměti: 5
- Počet ohodnocení účelovou funkcí: 1000\*Dimenze
- Prohledávaný prostor hodnot: (-100,100)
- Počet opakování pro každou kombinaci účelové funkce a diskretní verze: 30

Velikost počáteční populace byla nastavena podle vztahu (34), který se využívá u spojitě verze DISH algoritmu.

$$NP_{init} = 25 \log(D) * \sqrt{D} \quad (34)$$

Byly použity čtyři diskretní verze:

1. Spojitá mutace, zaokrouhlení při dosazení do účelové funkce (Zk-Fce)
2. Spojitá mutace, zaokrouhlení v populaci (Zk-Pop)
3. Transformace čísel (Transf.)
4. Diskretní mutace (DM)

### 8.1 Nastavení genetického algoritmu

Genetický algoritmus, jehož výsledky jsou uvedeny spolu s testovanými verzemi DISH byl spuštěn s následujícím nastavením

- Dimenze: 10/30/100
- Populace: 30/50/100
- Počet ohodnocení účelovou funkcí: 1000\*Dimenze
- Pravděpodobnost křížení: 0,9
- Jednobodové křížení

- Pravděpodobnost mutace: 0,01
- Prohledávaný prostor hodnot: (-100,100)
- Počet opakování pro každou účelovou funkci: 30

## 8.2 Testovací funkce

Vybrané testovací funkce, jejich předpis, hodnota a poloha minima:

- F1: OneMax:

$$f(x) = \sum_{i=1}^D (x_i) \quad (35)$$

Tabulka 2. Hodnota a poloha minima funkce F1

|                                 |                           |
|---------------------------------|---------------------------|
| Minimum (hodnota) {10, 30, 100} | {-1 000, -3 000, -10 000} |
| Minimum (poloha)                | {-100,-100, ... ,-100}    |

- F2: Lineární funkce

$$f(x) = \sum_{i=1}^D (i * x_i) \quad (37)$$

Tabulka 3. Hodnota a poloha minima funkce F2

|                                 |                             |
|---------------------------------|-----------------------------|
| Minimum (hodnota) {10, 30, 100} | {-5 500, -46 500, -505 000} |
| Minimum (poloha)                | {-100,-100, ... ,-100}      |

- F3: Sphere

$$f(x) = \sum_{i=1}^D (x_i^2) \quad (38)$$

Tabulka 4. Hodnota a poloha minima funkce F3

|                                 |               |
|---------------------------------|---------------|
| Minimum (hodnota) {10, 30, 100} | {0,0,0}       |
| Minimum (poloha)                | {0,0, ... ,0} |



- F4: Schwefelův problém 1.2

$$f(x) = \sum_{i=1}^D \left( \sum_{j=1}^i (x_j) \right)^2 \quad (39)$$

Tabulka 5. Hodnota a poloha minima funkce F4

|                                 |               |
|---------------------------------|---------------|
| Minimum (hodnota) {10, 30, 100} | {0,0,0}       |
| Minimum (poloha)                | {0,0, ..., 0} |

- F5: Schwefelův problém 2.26

$$f(x) = \sum_{i=1}^D (-x_i * \sin(\sqrt{|x_i|})) \quad (40)$$

Tabulka 6. Hodnota a poloha minima funkce F5

|                                 |                               |
|---------------------------------|-------------------------------|
| Minimum (hodnota) {10, 30, 100} | {-989,21; -2967,64; -9892,14} |
| Minimum (poloha)                | {-99,-99, ..., -99}           |

- F6: Salomonova funkce

$$f(x) = 1 - \cos \left( 2\pi \sqrt{\sum_{i=1}^D x_i^2} \right) + 0.1 \sqrt{\sum_{i=1}^D x_i^2} \quad (36)$$

Tabulka 7. Hodnota a poloha minima funkce F6

|                                 |               |
|---------------------------------|---------------|
| Minimum (hodnota) {10, 30, 100} | {0,0,0}       |
| Minimum (poloha)                | {0,0, ..., 0} |

- F7: Ackleyho funkce

$$f(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e \quad (41)$$

Tabulka 8. Hodnota a poloha minima funkce F7

|                                 |               |
|---------------------------------|---------------|
| Minimum (hodnota) {10, 30, 100} | {0,0,0}       |
| Minimum (poloha)                | {0,0, ..., 0} |

- F8: Greiwankova funkce

$$f(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (42)$$

Tabulka 9. Hodnota a poloha minima funkce F8

|                                 |               |
|---------------------------------|---------------|
| Minimum (hodnota) {10, 30, 100} | {0,0,0}       |
| Minimum (poloha)                | {0,0, ... ,0} |

Byly zvoleny funkce, které lze použít pro testování spojitých i diskretních algoritmů. Funkce OneMax a Lineár jsou lineární funkce, které mají pouze jedno minimum, které se nalézá na spodním okraji prohledávané oblasti. Dále byly zvoleny funkce Sphere a Schwefelův problém 1.2, které mají pouze jeden extrém umístěný v bodě  $[0, \dots, 0]$  [6,7,10].

Tyto funkce pak byly doplněny o funkce obsahující více extrémů, ale pouze jeden globální. Konkrétně se jedná o funkce Salomonovu, Ackleyho, Greiwankovu a Schwefelův problém 2.26 [6,7,10].

## 9 VÝSLEDKY ALGORITMU

V následující části jsou uvedeny výsledky testovaných algoritmů pro každou z testovaných funkcí spolu se statistickým vyhodnocením.

### 9.1 F1: OneMax

Tabulka 10. Statistické vyhodnocení pro F1

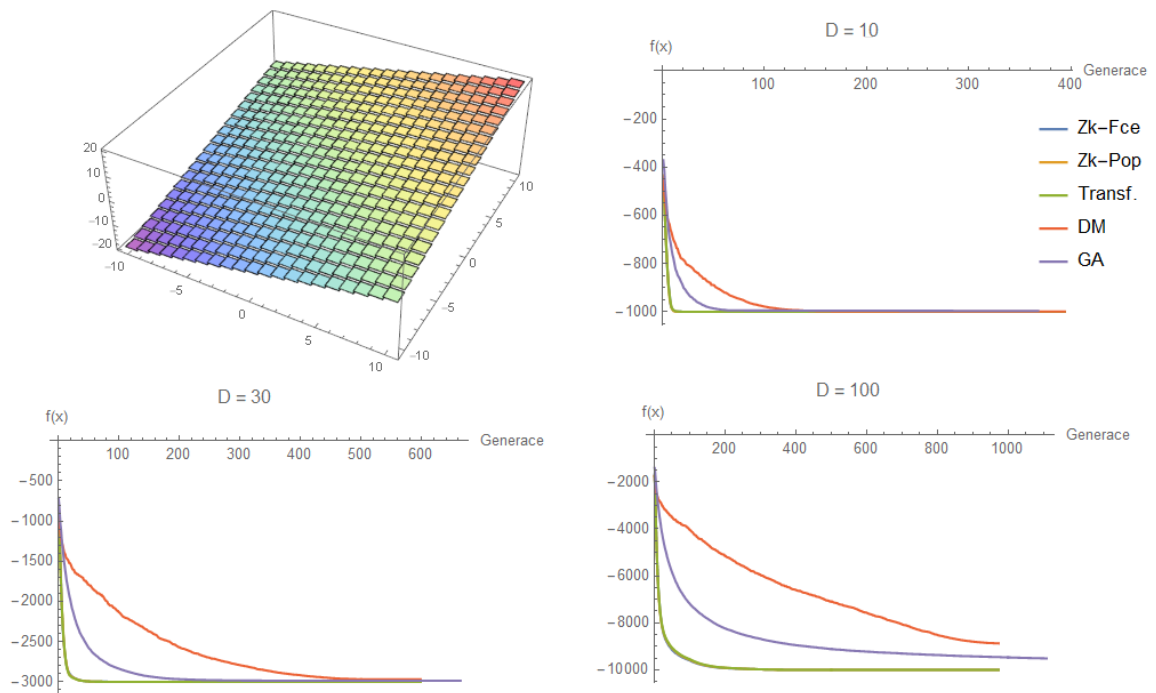
| Dimenze             | 10     | Minimum: -1000  |         |       |        | 2x maxFES |
|---------------------|--------|-----------------|---------|-------|--------|-----------|
| Verze               | Zk-Fce | Zk-Pop          | Transf. | DM    | GA     | DM        |
| Nejlepší výsledek   | -1000  | -1000           | -1000   | -1000 | -1000  | -1000     |
| Nejhorší výsledek   | -1000  | -1000           | -1000   | -995  | -993   | -997      |
| Průměrný výsledek   | -1000  | -1000           | -1000   | -998  | -996,9 | -999,7    |
| Směrodatná odchylka | 0      | 0               | 0       | 1,311 | 1,479  | 0,639     |
| Medián              | -1000  | -1000           | -1000   | -999  | -997   | -1000     |
| Dimenze             | 30     | Minimum: -3000  |         |       |        | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop          | DM      | DM    | GA     | DM        |
| Nejlepší výsledek   | -3000  | -3000           | -3000   | -2987 | -2991  | -3000     |
| Nejhorší výsledek   | -3000  | -3000           | -3000   | -2940 | -2981  | -2984     |
| Průměrný výsledek   | -3000  | -3000           | -3000   | -2971 | -2988  | -2996     |
| Směrodatná odchylka | 0      | 0               | 0       | 10,29 | 2,335  | 3,354     |
| Medián              | -3000  | -3000           | -3000   | -2972 | -2988  | -2996     |
| Dimenze             | 100    | Minimum: -10000 |         |       |        | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop          | Transf. | DM    | GA     | DM        |
| Nejlepší výsledek   | -10000 | -10000          | -10000  | -9219 | -9575  | -9895     |
| Nejhorší výsledek   | -9998  | -10000          | -10000  | -8252 | -9436  | -9402     |
| Průměrný výsledek   | -10000 | -10000          | -10000  | -8871 | -9511  | -9623     |
| Směrodatná odchylka | 0,402  | 0               | 0       | 228   | 34,82  | 113,2     |
| Medián              | -10000 | -10000          | -10000  | -8890 | -9508  | -9620     |

Z výsledků je vidět, že obě varianty zaokrouhlení a transformace čísel podávají stejně dobré výsledky pro všechna zvolená nastavení dimenzí. Verze s diskretní mutací je na tom podstatně hůře. Při deseti dimenzionálním problému ještě zvládá nalézt minimum funkce, ale při rostoucím počtu dimenzí se její výsledky zhoršují.

Při porovnání s genetickým algoritmem první 3 varianty DE podávají lepší výsledky než GA. Poslední varianta s diskretní mutací vrací o něco lepší výsledky pro  $D = 10$ , ovšem pro vyšší dimenze je už lepší genetický algoritmus.

V posledním sloupci byla testována varianta s diskretní mutací, která měla téměř u všech funkcí podstatně horší výsledky, s dvojnásobným počtem ohodnocení. Z výsledků se dá říct,

že při větším počtu ohodnocení se výsledky přiblížily ostatním variantám, ale i tak je pořád horší než ostatní testované verze.



Obrázek 17. F1: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty  
Z průběhu nejlepší nalezené hodnoty je vidět, že verze s diskrétní mutací má nejpomalejší konvergenci ke globálnímu minimu. Rozdílný počet generací mezi DE a GA je způsoben rozdílem ve velikosti populace. DE má z počátku větší populaci, která se ovšem v průběhu generací postupně snižuje. Ukončovací podmínka byla pro oba algoritmy stejná – maximální počet ohodnocení účelové funkce.

Tabulka 11. Wilcoxonův Rank-Sum test pro F1

| Dimenze | 10       |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
| Verze   | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,00E+00 | 1,00E+00 | 4,72E-06 | 3,46E-12 |
| Zk-Pop  | 1,00E+00 | -        | 1,00E+00 | 4,72E-06 | 3,46E-12 |
| Transf  | 1,00E+00 | 1,00E+00 | -        | 4,72E-06 | 3,46E-12 |
| DM      | 5,17E-06 | 5,17E-06 | 5,17E-06 | -        | 1,90E-06 |
| Dimenze | 30       |          |          |          |          |
| Verze   | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,00E+00 | 1,00E+00 | 1,06E-12 | 1,00E-12 |
| Zk-Pop  | 1,00E+00 | -        | 1,00E+00 | 1,06E-12 | 1,00E-12 |
| Transf  | 1,00E+00 | 1,00E+00 | -        | 1,06E-12 | 1,00E-12 |
| DM      | 1,19E-12 | 1,19E-12 | 1,19E-12 | -        | 5,15E-10 |
| Dimenze | 100      |          |          |          |          |
| Verze   | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,60E-01 | 1,60E-01 | 2,11E-12 | 2,11E-12 |
| Zk-Pop  | 1,40E-01 | -        | 1,00E+00 | 1,08E-12 | 1,07E-12 |
| Transf  | 1,40E-01 | 1,00E+00 | -        | 1,08E-12 | 1,07E-12 |
| DM      | 2,36E-12 | 1,21E-12 | 1,21E-12 | -        | 3,01E-11 |

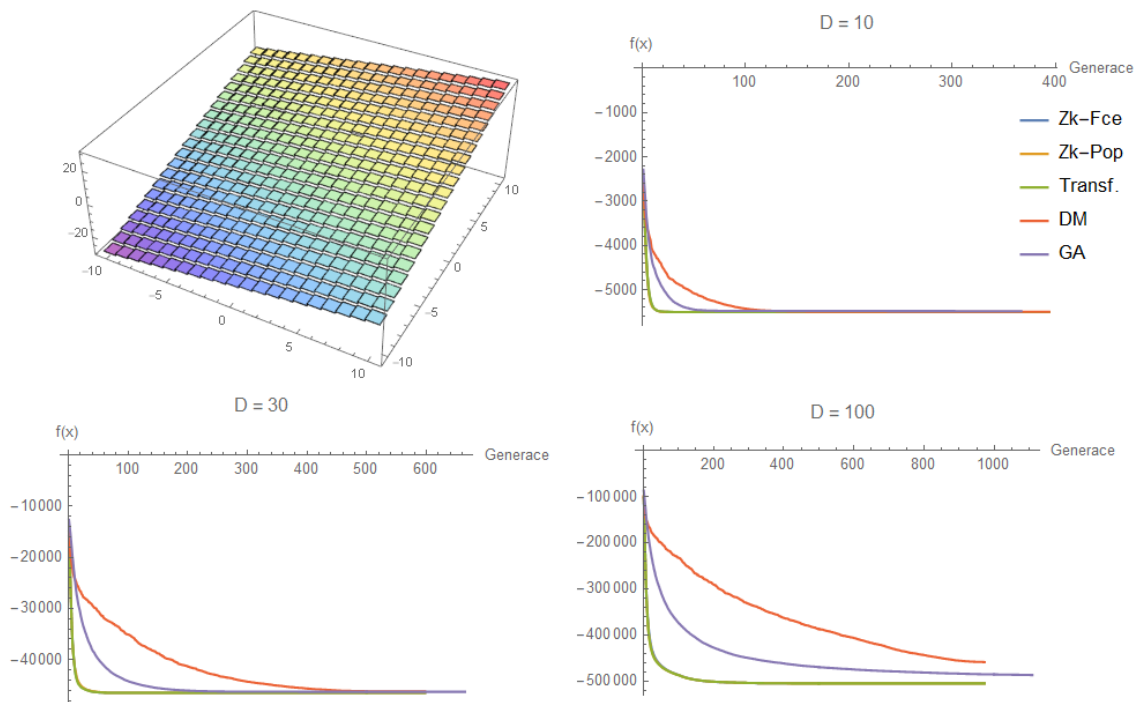
Výsledky jednotlivých variant byly statisticky porovnány pomocí Wilcoxonova Rank-Sum testu. Ten slouží k porovnání dvojice algoritmů a zjištění, jak jsou si blízké. Jako hranice byla stanovena hodnota 0,05, tedy maximálně 5% šance nesprávného zamítnutí nulové hypotézy, která říká, že výsledky algoritmů jsou stejné. Testované varianty s výsledkem testu menší než 0,05 jsou tedy výrazně lepší nebo výrazně horší.

U první funkce lze tedy říci, že verze s diskretní mutací a genetický algoritmus jsou výrazně odlišné a pokud se podíváme na výsledky nalezených hodnot je vidět, že jsou výrazně horší.

## 9.2 F2: Lineární funkce

Tabulka 12. Statistické vyhodnocení pro F2

| Dimenze             | 10      | Minimum: -5500   |         |         |         | 2x maxFES |
|---------------------|---------|------------------|---------|---------|---------|-----------|
| Verze               | Zk-Fce  | Zk-Pop           | DM      | DM      | GA      | DM        |
| Nejlepší výsledek   | -5500   | -5500            | -5500   | -5500   | -5500   | -5500     |
| Nejhorsí výsledek   | -5500   | -5500            | -5500   | -5474   | -5462   | -5484     |
| Průměrný výsledek   | -5500   | -5500            | -5500   | -5494   | -5482   | -5499     |
| Směrodatná odchylna | 0       | 0                | 0       | 7,059   | 8,192   | 3,168     |
| Medián              | -5500   | -5500            | -5500   | -5495   | -5483   | -5500     |
| Dimenze             | 30      | Minimum: -46500  |         |         |         | 2x maxFES |
| Verze               | Zk-Fce  | Zk-Pop           | DM      | DM      | GA      | DM        |
| Nejlepší výsledek   | -46500  | -46500           | -46500  | -46420  | -46370  | -46500    |
| Nejhorsí výsledek   | -46500  | -46500           | -46500  | -45900  | -46160  | -46410    |
| Průměrný výsledek   | -46500  | -46500           | -46500  | -46200  | -46300  | -46470    |
| Směrodatná odchylna | 0       | 0                | 0       | 107,1   | 49,62   | 22,25     |
| Medián              | -46500  | -46500           | -46500  | -46220  | -46310  | -46470    |
| Dimenze             | 100     | Minimum: -505000 |         |         |         | 2x maxFES |
| Verze               | Zk-Fce  | Zk-Pop           | DM      | DM      | GA      | DM        |
| Nejlepší výsledek   | -505000 | -505000          | -505000 | -474500 | -490100 | -496900   |
| Nejhorsí výsledek   | -504900 | -505000          | -505000 | -435200 | -484800 | -470700   |
| Průměrný výsledek   | -505000 | -505000          | -505000 | -458800 | -486900 | -489400   |
| Směrodatná odchylna | 21,66   | 0                | 0       | 10440   | 1459    | 5080      |
| Medián              | -505000 | -505000          | -505000 | -462000 | -486500 | -490400   |



Obrázek 18. F2: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty

Tabulka 13. Wilcoxonův Rank-Sum test pro F2

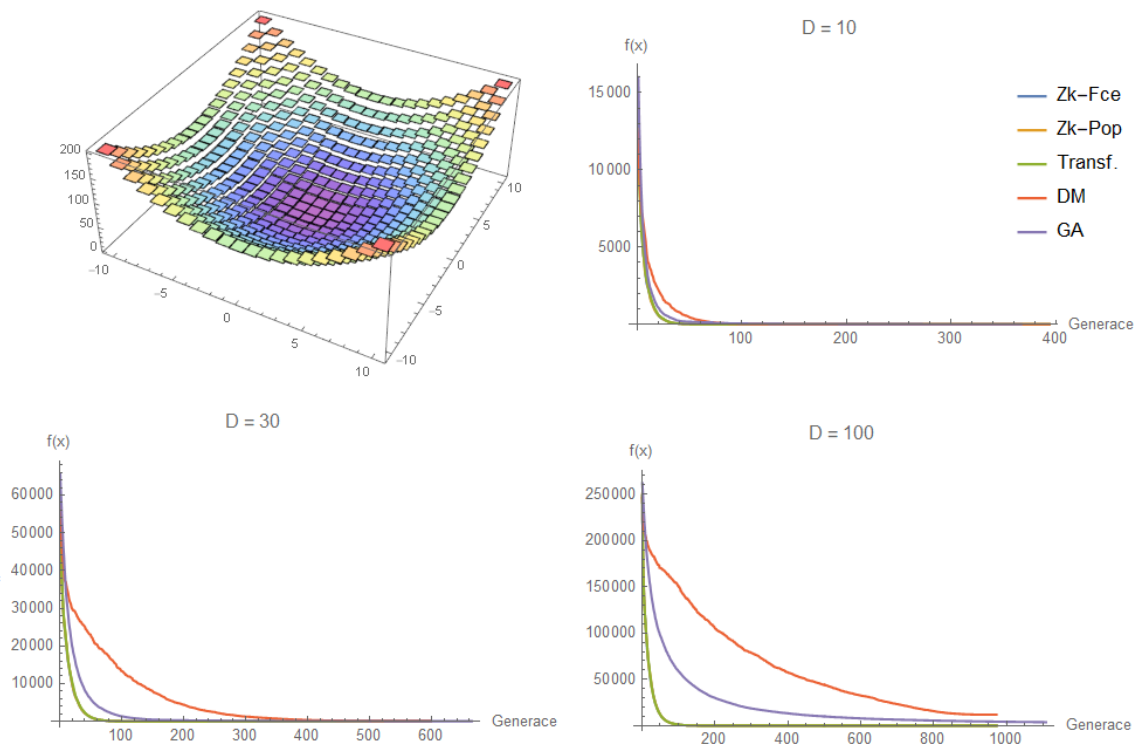
| Dimenze | 10       |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,00E+00 | 1,00E+00 | 1,18E-07 | 4,00E-12 |
| Zk-Pop  | 1,00E+00 | -        | 1,00E+00 | 1,18E-07 | 4,00E-12 |
| Transf  | 1,00E+00 | 1,00E+00 | -        | 1,18E-07 | 4,00E-12 |
| DM      | 1,30E-07 | 1,30E-07 | 1,30E-07 | -        | 7,41E-07 |
| Dimenze | 30       |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,00E+00 | 1,00E+00 | 1,08E-12 | 1,07E-12 |
| Zk-Pop  | 1,00E+00 | -        | 1,00E+00 | 1,08E-12 | 1,07E-12 |
| Transf  | 1,00E+00 | 1,00E+00 | -        | 1,08E-12 | 1,07E-12 |
| DM      | 1,21E-12 | 1,21E-12 | 1,21E-12 | -        | 1,74E-05 |
| Dimenze | 100      |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 6,60E-05 | 6,60E-05 | 1,61E-11 | 1,61E-11 |
| Zk-Pop  | 6,60E-05 | -        | 1,00E+00 | 1,08E-12 | 1,08E-12 |
| Transf  | 6,60E-05 | 1,00E+00 | -        | 1,08E-12 | 1,08E-12 |
| DM      | 1,78E-11 | 1,21E-12 | 1,21E-12 | -        | 3,01E-11 |

Statistické výsledky i průběhy hodnot jsou podobné jako u funkce F1. Opět se ukázaly jako nejlepší první 3 testované verze DE. Při sto dimenzionálním problému začíná první verze, tedy verze se zaokrouhlováním při dosažení do účelové funkce, podávat nepatrně horší výsledky. To potvrzuje i Wilcoxonův Rank-Sum test, který při nejvyšší testované dimenzi už považuje za odlišnou i první verzi se zaokrouhlováním při dosažení do účelové funkce.

## 9.3 F3: Sphere

Tabulka 14. Statistické vyhodnocení pro F3

| Dimenze             | 10     | Minimum: 0 |         |       |       | 2x maxFES |
|---------------------|--------|------------|---------|-------|-------|-----------|
| Verze               | Zk-Fce | Zk-Pop     | DM      | DM    | GA    | DM        |
| Nejlepší výsledek   | 0      | 0          | 0       | 0     | 1     | 0         |
| Nejhůrší výsledek   | 0      | 0          | 0       | 5     | 7     | 1         |
| Průměrný výsledek   | 0      | 0          | 0       | 0,566 | 3,167 | 0,033     |
| Směrodatná odchylka | 0      | 0          | 0       | 1,073 | 1,577 | 0,182     |
| Medián              | 0      | 0          | 0       | 0     | 3     | 0         |
| Dimenze             | 30     | Minimum: 0 |         |       |       | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0      | 0          | 0       | 12    | 8     | 2         |
| Nejhůrší výsledek   | 1      | 1          | 0       | 249   | 79    | 14        |
| Průměrný výsledek   | 0,033  | 0,033      | 0       | 104   | 25    | 6,5       |
| Směrodatná odchylka | 0,182  | 0,182      | 0       | 58,51 | 15,32 | 3,246     |
| Medián              | 0      | 0          | 0       | 102,5 | 20    | 6         |
| Dimenze             | 100    | Minimum: 0 |         |       |       | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0      | 1          | 1       | 6086  | 2700  | 1402      |
| Nejhůrší výsledek   | 13     | 6          | 7       | 20690 | 5174  | 5858      |
| Průměrný výsledek   | 5,967  | 3,267      | 3,467   | 11800 | 3751  | 2751      |
| Směrodatná odchylka | 2,988  | 1,337      | 1,961   | 3087  | 595,8 | 1273      |
| Medián              | 5,5    | 3          | 3,5     | 11340 | 3641  | 2366      |



Obrázek 19. F3: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty



Tabulka 15. Wilcoxonův Rank-Sum test pro F3

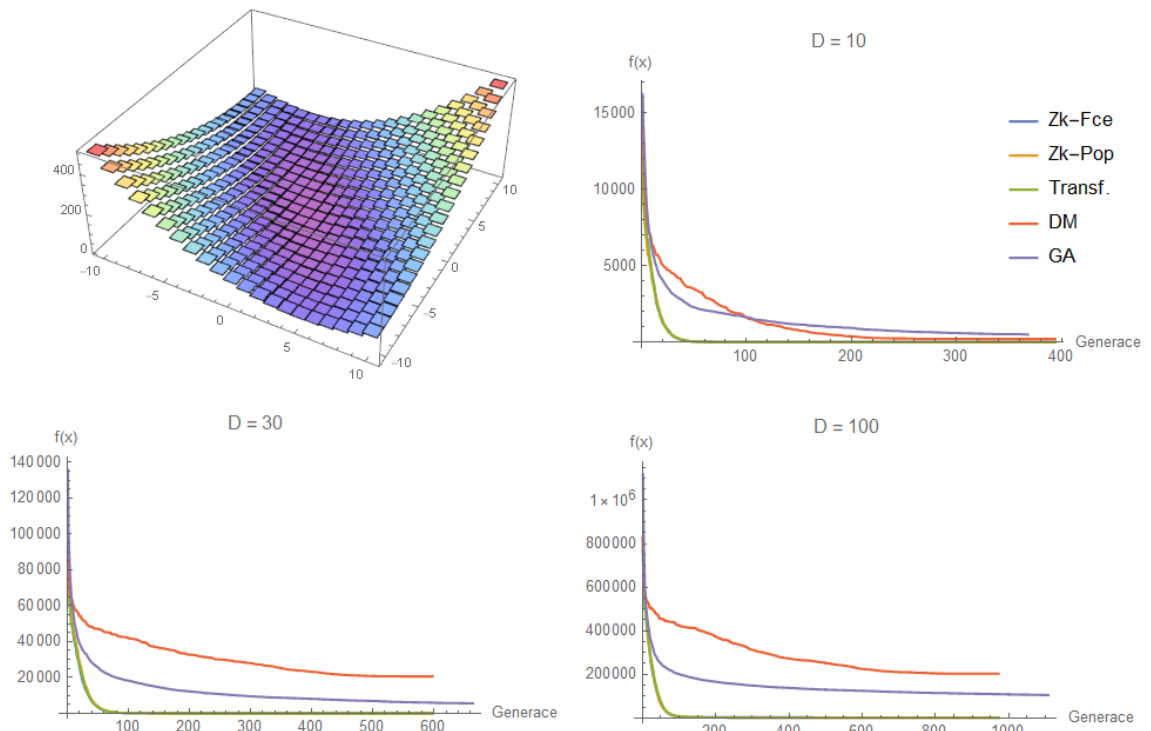
| Dimenze | 10       |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,00E+00 | 1,00E+00 | 5,98E-04 | 9,18E-13 |
| Zk-Pop  | 1,00E+00 | -        | 1,00E+00 | 5,98E-04 | 9,18E-13 |
| Transf  | 1,00E+00 | 1,00E+00 | -        | 5,98E-04 | 9,18E-13 |
| DM      | 6,51E-04 | 6,51E-04 | 6,51E-04 | -        | 4,36E-09 |
| Dimenze | 30       |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 9,80E-01 | 3,30E-01 | 1,53E-12 | 1,48E-12 |
| Zk-Pop  | 9,80E-01 | -        | 3,30E-01 | 1,53E-12 | 1,48E-12 |
| Transf  | 3,00E-01 | 3,00E-01 | -        | 1,07E-12 | 1,04E-12 |
| DM      | 1,71E-12 | 1,71E-12 | 1,20E-12 | -        | 4,27E-09 |
| Dimenze | 100      |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 4,10E-05 | 3,93E-04 | 2,49E-11 | 2,49E-11 |
| Zk-Pop  | 3,84E-05 | -        | 9,60E-01 | 2,36E-11 | 2,36E-11 |
| Transf  | 3,71E-04 | 9,70E-01 | -        | 2,38E-11 | 2,38E-11 |
| DM      | 2,75E-11 | 2,62E+00 | 2,64E-11 | -        | 3,01E-11 |

Při funkci F3 podávaly opět nejlepší výsledky první 3 verze diskrétní DE následované GA a nejhůře dopadla verze s Diskrétní mutací. Verze se zaokrouhlováním při dosazení do účelové funkce při vyšším počtu dimenzí opět podává nepatrně horší výsledky než verze se zaokrouhlováním přímo v populaci a verze s transformací čísel. Wilcoxonův Rank-Sum test dopadl stejně jako u předchozí funkce.

## 9.4 F4: Schwefelův problém 1.2

Tabulka 16. Statistické vyhodnocení pro F4

| Dimenze             | 10     | Minimum: 0 |         |        |        | 2x maxFES |
|---------------------|--------|------------|---------|--------|--------|-----------|
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM     | GA     | DM        |
| Nejlepší výsledek   | 0      | 0          | 0       | 19     | 65     | 2         |
| Nejhorší výsledek   | 0      | 0          | 0       | 729    | 1298   | 134       |
| Průměrný výsledek   | 0      | 0          | 0       | 201,8  | 484,2  | 20,93     |
| Směrodatná odchylka | 0      | 0          | 0       | 141,7  | 350,6  | 24,95     |
| Medián              | 0      | 0          | 0       | 169,5  | 482,5  | 15,5      |
| Dimenze             | 30     | Minimum: 0 |         |        |        | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM     | GA     | DM        |
| Nejlepší výsledek   | 8      | 2          | 2       | 7284   | 2358   | 7112      |
| Nejhorší výsledek   | 101    | 12         | 9       | 32060  | 9551   | 25500     |
| Průměrný výsledek   | 24,2   | 5,633      | 5,833   | 20530  | 5506   | 14600     |
| Směrodatná odchylka | 19,98  | 2,553      | 2,086   | 5529   | 1689   | 3893      |
| Medián              | 18,5   | 5,5        | 6       | 20070  | 5036   | 13300     |
| Dimenze             | 100    | Minimum: 0 |         |        |        | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM     | GA     | DM        |
| Nejlepší výsledek   | 1892   | 428        | 407     | 152900 | 88670  | 119400    |
| Nejhorší výsledek   | 3383   | 871        | 908     | 275700 | 134300 | 269300    |
| Průměrný výsledek   | 2487   | 601,1      | 595,3   | 202800 | 104800 | 176600    |
| Směrodatná odchylka | 404,9  | 101,5      | 123,8   | 32850  | 10510  | 32500     |
| Medián              | 2430   | 604,5      | 580     | 196400 | 103000 | 177200    |



Obrázek 20. F4: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty

Tabulka 17. Wilcoxonův Rank-Sum test pro F4

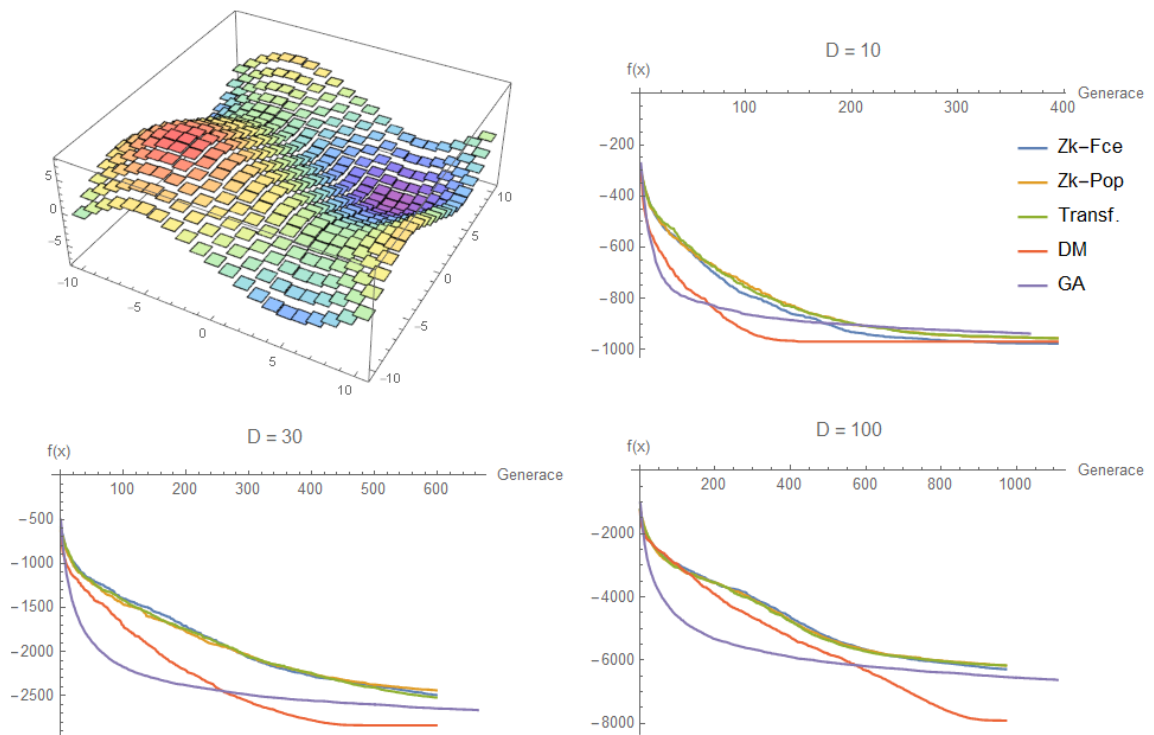
| Dimenze | 10       |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,00E+00 | 1,00E+00 | 1,07E-12 | 1,08E-12 |
| Zk-Pop  | 1,00E+00 | -        | 1,00E+00 | 1,07E-12 | 1,08E-12 |
| Transf  | 1,00E+00 | 1,00E+00 | -        | 1,07E-12 | 1,08E-12 |
| DM      | 1,20E-12 | 1,20E-12 | 1,20E-12 | -        | 2,05E-03 |
| Dimenze | 30       |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 8,37E-11 | 4,63E-11 | 2,70E-11 | 2,70E-11 |
| Zk-Pop  | 7,59E-11 | -        | 4,90E-01 | 2,58E-11 | 2,58E-11 |
| Transf  | 4,19E-11 | 5,00E-01 | -        | 2,58E-11 | 2,58E-11 |
| DM      | 2,98E-11 | 2,86E-11 | 2,85E-11 | -        | 4,07E-11 |
| Dimenze | 100      |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 3,01E-11 | 3,01E-11 | 2,73E-11 | 2,73E-11 |
| Zk-Pop  | 2,72E-11 | -        | 6,70E-01 | 2,72E-11 | 2,72E-11 |
| Transf  | 2,73E-11 | 6,60E-01 | -        | 2,73E-11 | 2,73E-11 |
| DM      | 3,01E-11 | 3,01E-11 | 3,01E-11 | -        | 3,01E-11 |

Výsledky jsou podobné jako u předchozích funkcí, kde verze 1-3 podávají nejlepší výsledky. Zhoršení 1. testované verze DE – zaokrouhlování při dosazování je zde již patrné i u třiceti dimenzionálního problému, což potvrzuje i Wilcoxonův Rank-Sum test. Genetický algoritmus zde podává při deseti dimenzionálním problému nejhorší výsledky, ovšem při vyšším počtu dimenzí jsou výsledky GA lepší než výsledky diskretní mutace.

## 9.5 F5: Schwefelův problém 2.26

Tabulka 18. Statistické vyhodnocení pro F5

| Dimenze             | 10     | Minimum: -989,21  |         |        |        | 2x maxFES |
|---------------------|--------|-------------------|---------|--------|--------|-----------|
| Verze               | Zk-Fce | Zk-Pop            | Transf. | DM     | GA     | DM        |
| Nejlepší výsledek   | -989,2 | -989,2            | -989,2  | -989,2 | -984,7 | -989,2    |
| Nejhorší výsledek   | -919,5 | -732,5            | -844,4  | -947,3 | -883,9 | -961,7    |
| Průměrný výsledek   | -976,6 | -955,5            | -955,9  | -968,9 | -937,1 | -978,5    |
| Směrodatná odchylka | 20,56  | 57,97             | 40,43   | 10,61  | 24,15  | 6,771     |
| Medián              | -989,2 | -981,6            | -974,1  | -969,5 | -936   | -980,2    |
| Dimenze             | 30     | Minimum: -2967,64 |         |        |        | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop            | Transf. | DM     | GA     | DM        |
| Nejlepší výsledek   | -2743  | -2811             | -2871   | -2901  | -2735  | -2954     |
| Nejhorší výsledek   | -2124  | -2032             | -2161   | -2769  | -2543  | -2876     |
| Průměrný výsledek   | -2492  | -2440             | -2520   | -2836  | -2665  | -2929     |
| Směrodatná odchylka | 200,3  | 227               | 197,7   | 34,67  | 37,61  | 19,26     |
| Medián              | -2545  | -2429             | -2544   | -2842  | -2663  | -2931     |
| Dimenze             | 100    | Minimum: -9892,14 |         |        |        | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop            | Transf. | DM     | GA     | DM        |
| Nejlepší výsledek   | -7200  | -7080             | -6539   | -8428  | -6925  | -9508     |
| Nejhorší výsledek   | -5285  | -5517             | -5716   | -7262  | -6305  | -8221     |
| Průměrný výsledek   | -6284  | -6167             | -6163   | -7905  | -6622  | -8912     |
| Směrodatná odchylka | 363,4  | 325               | 222,1   | 291,8  | 156,9  | 271,8     |
| Medián              | -6263  | -6091             | -6201   | -7901  | -6621  | -8968     |



Obrázek 21. F5: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty

Tabulka 19. Wilcoxonův Rank-Sum test pro F5

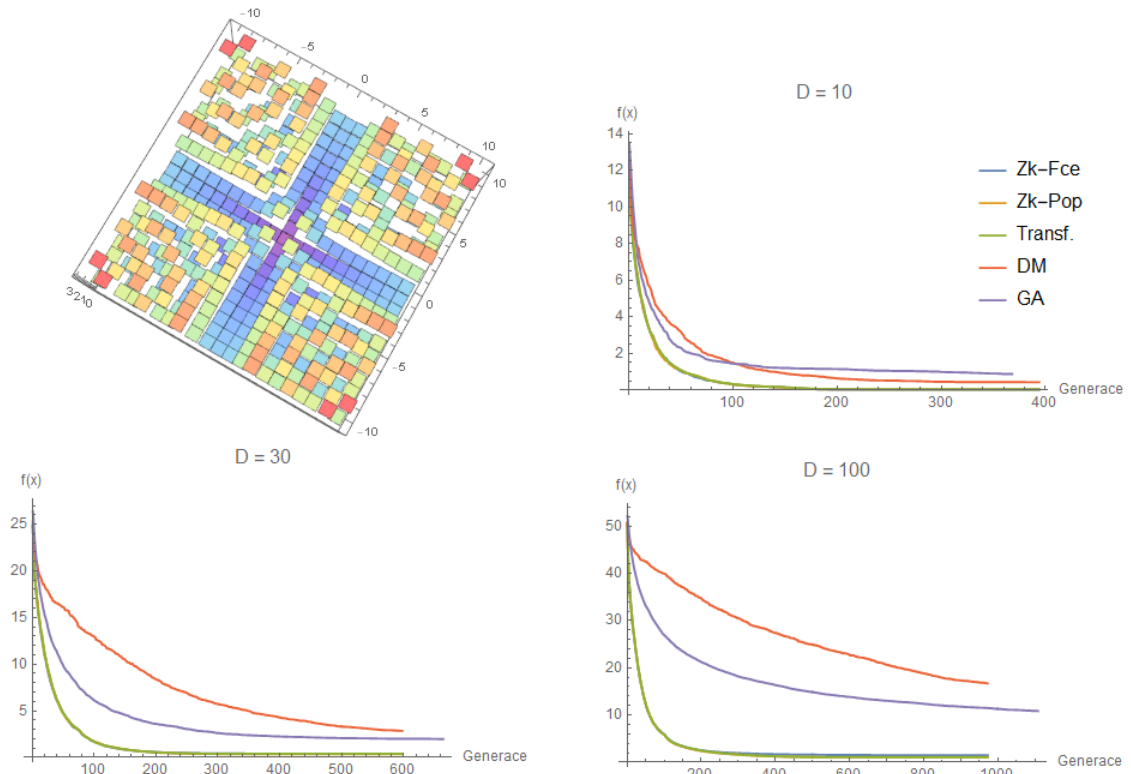
| Dimenze | 10       |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,10E-01 | 2,00E-02 | 1,22E-03 | 1,45E-07 |
| Zk-Pop  | 1,20E-01 | -        | 5,30E-01 | 1,20E-01 | 4,01E-04 |
| Transf  | 2,00E-02 | 5,40E-01 | -        | 8,70E-01 | 4,32E-03 |
| DM      | 1,29E-03 | 1,30E-01 | 8,80E-01 | -        | 1,90E-07 |
| Dimenze | 30       |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 3,40E-01 | 8,30E-01 | 3,01E-11 | 1,67E-03 |
| Zk-Pop  | 3,50E-01 | -        | 1,80E-01 | 8,99E-11 | 6,35E-05 |
| Transf  | 8,10E-01 | 1,70E-01 | -        | 4,18E-09 | 3,36E-04 |
| DM      | 2,73E-11 | 8,15E-11 | 3,82E-09 | -        | 2,73E-11 |
| Dimenze | 100      |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,00E-01 | 1,40E-01 | 3,01E-11 | 1,24E-05 |
| Zk-Pop  | 1,00E-01 | -        | 7,10E-01 | 3,01E-11 | 1,72E-07 |
| Transf  | 1,50E-01 | 7,00E-01 | -        | 3,01E-11 | 5,57E-10 |
| DM      | 2,73E-11 | 2,73E-11 | 2,73E-11 | -        | 2,73E-11 |

U této funkce se potvrzuje No Free Lunch teorém [1]. U předchozích funkcí většinou podávaly nejlepší výsledky verze 1-3 DE a poslední verze byla podstatně horší. Pro malý počet dimenzí jsou zde všechny 4 testované verze diferenciální evoluce stejně kvalitní, ale ve vyšších dimenzích je úspěšnost jednotlivých testovaných variant v opačném pořadí. Nejlépe zde dopadla verze s diskretní mutací následována genetickým algoritmem. Další zajímavostí u této funkce je, že podle Wilcoxonova testu zde nedochází při vyšším počtu dimenzí k odlišnosti první testované verze DE.

9.6 F6: Salomonova funkce

Tabulka 20. Statistické vyhodnocení pro F6

| Dimenze             | 10     | Minimum: 0 |         |       |       | 2x maxFES |
|---------------------|--------|------------|---------|-------|-------|-----------|
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0      | 0          | 0       | 0,2   | 0,4   | 0         |
| Nejhorsí výsledek   | 0,2    | 0,2        | 0,2     | 0,7   | 1,5   | 0,3       |
| Průměrný výsledek   | 0,023  | 0,04       | 0,046   | 0,433 | 0,88  | 0,186     |
| Směrodatná odchylna | 0,056  | 0,081      | 0,081   | 0,134 | 0,265 | 0,089     |
| Medián              | 0      | 0          | 0       | 0,4   | 0,84  | 0,2       |
| Dimenze             | 30     | Minimum: 0 |         |       |       | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0,2    | 0,3        | 0,3     | 2,14  | 1,238 | 0,8       |
| Nejhorsí výsledek   | 0,6    | 0,4        | 0,5     | 4,822 | 2,609 | 2,2       |
| Průměrný výsledek   | 0,383  | 0,34       | 0,38    | 2,847 | 1,963 | 1,429     |
| Směrodatná odchylna | 0,091  | 0,049      | 0,055   | 0,525 | 0,273 | 0,264     |
| Medián              | 0,4    | 0,3        | 0,4     | 2,806 | 2     | 1,411     |
| Dimenze             | 100    | Minimum: 0 |         |       |       | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 1,2    | 0,9        | 0,9     | 13,21 | 9,303 | 8,827     |
| Nejhorsí výsledek   | 1,8    | 1,2        | 1,3     | 20,52 | 12    | 13,03     |
| Průměrný výsledek   | 1,476  | 1,047      | 1,077   | 16,71 | 10,8  | 10,82     |
| Směrodatná odchylna | 0,142  | 0,107      | 0,11    | 1,683 | 0,785 | 1,15      |
| Medián              | 1,5    | 1,05       | 1,1     | 16,77 | 10,79 | 10,72     |



Obrázek 22. F6: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty

Tabulka 21. Wilcoxonův Rank-Sum test pro F6

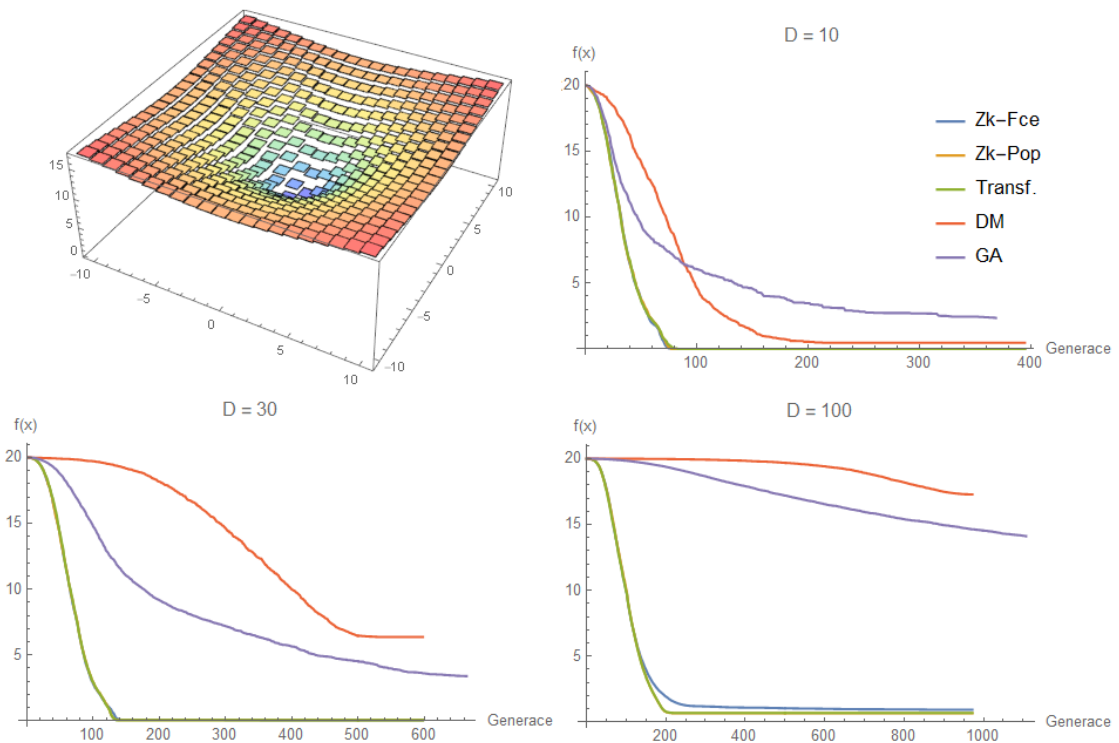
| Dimenze | 10       |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 5,90E-01 | 2,80E-01 | 4,94E-12 | 4,33E-12 |
| Zk-Pop  | 6,00E-01 | -        | 6,20E-01 | 8,98E-12 | 5,27E-12 |
| Transf  | 2,80E-01 | 6,30E-01 | -        | 1,30E-11 | 7,68E-12 |
| DM      | 5,51E-12 | 1,00E-11 | 1,44E-11 | -        | 2,48E-09 |
| Dimenze | 30       |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 2,00E-02 | 8,80E-01 | 1,89E-11 | 1,88E-11 |
| Zk-Pop  | 2,00E-02 | -        | 6,00E-03 | 1,19E-11 | 1,18E-11 |
| Transf  | 8,70E-01 | 6,00E-03 | -        | 1,07E-11 | 1,06E-11 |
| DM      | 2,10E-11 | 1,32E-11 | 1,19E-11 | -        | 1,74E-10 |
| Dimenze | 100      |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 2,99E-11 | 6,74E-11 | 2,44E-11 | 2,44E-11 |
| Zk-Pop  | 2,70E-11 | -        | 3,80E-01 | 2,26E-11 | 2,26E-11 |
| Transf  | 6,10E-11 | 3,90E-01 | -        | 2,12E-11 | 2,11E-11 |
| DM      | 2,70E-11 | 2,50E-11 | 2,34E-11 | -        | 3,01E-11 |

Výsledky Funkce F6 jsou podobné jako u většiny testovaných funkcí. Opět jsou zde nejúspěšnější verze se zaokrouhlováním v populaci a verze s transformací čísel.

## 9.7 F7: Ackleyho funkce

Tabulka 22. Statistické vyhodnocení pro F7

| Dimenze             | 10     | Minimum: 0 |         |       |       | 2x maxFES |
|---------------------|--------|------------|---------|-------|-------|-----------|
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0      | 0          | 0       | 0     | 0     | 0         |
| Nejhorsí výsledek   | 0      | 0          | 0       | 2,075 | 5,936 | 0         |
| Průměrný výsledek   | 0      | 0          | 0       | 0,469 | 2,348 | 0         |
| Směrodatná odchylna | 0      | 0          | 0       | 0,696 | 1,106 | 0         |
| Medián              | 0      | 0          | 0       | 0     | 2,376 | 0         |
| Dimenze             | 30     | Minimum: 0 |         |       |       | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0      | 0          | 0       | 3,082 | 2,554 | 0,717     |
| Nejhorsí výsledek   | 0,717  | 1,007      | 0,717   | 10,08 | 5,297 | 3,013     |
| Průměrný výsledek   | 0,047  | 0,0057     | 0,047   | 6,362 | 3,385 | 1,847     |
| Směrodatná odchylna | 0,181  | 0,221      | 0,181   | 1,35  | 0,665 | 0,522     |
| Medián              | 0      | 0          | 0       | 6,375 | 3,148 | 1,776     |
| Dimenze             | 100    | Minimum: 0 |         |       |       | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0,396  | 0          | 0       | 13,19 | 13,38 | 4,064     |
| Nejhorsí výsledek   | 1,491  | 1,061      | 1,1     | 18,61 | 15,42 | 15,88     |
| Průměrný výsledek   | 0,924  | 0,664      | 0,681   | 17,28 | 14,11 | 11,8      |
| Směrodatná odchylna | 0,218  | 0,216      | 0,232   | 1,006 | 0,42  | 2,382     |
| Medián              | 0,956  | 0,681      | 0,732   | 17,36 | 14,03 | 11,91     |



Obrázek 23. F7: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty



Tabulka 23. Wilcoxonův Rank-Sum test pro F7

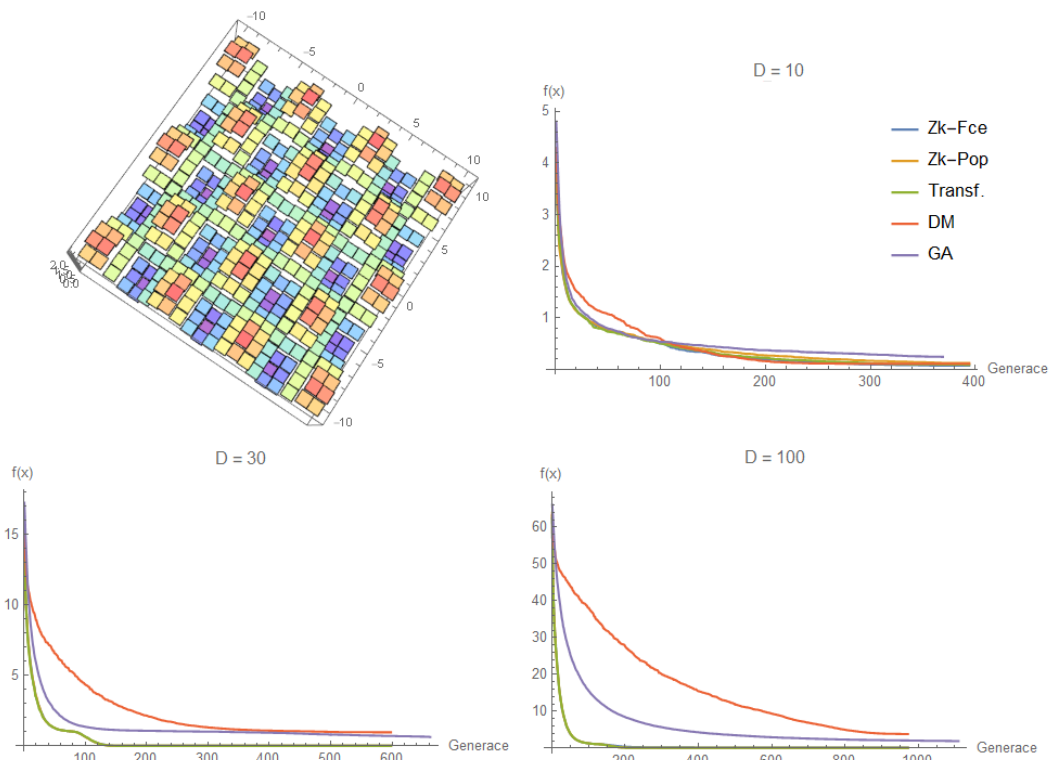
| Dimenze | 10       |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 1,00E+00 | 1,00E+00 | 5,94E-04 | 3,54E-12 |
| Zk-Pop  | 1,00E+00 | -        | 1,00E+00 | 5,94E-04 | 3,54E-12 |
| Transf  | 1,00E+00 | 1,00E+00 | -        | 5,94E-04 | 3,54E-12 |
| DM      | 6,46E-04 | 6,46E-04 | 6,46E-04 | -        | 1,10E-09 |
| Dimenze | 30       |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 9,50E-01 | 9,80E-01 | 2,11E-12 | 2,08E-12 |
| Zk-Pop  | 9,80E-01 | -        | 9,80E-01 | 2,11E-12 | 2,08E-12 |
| Transf  | 9,80E-01 | 9,50E-01 | -        | 2,11E-12 | 2,08E-12 |
| DM      | 2,36E-12 | 2,36E-12 | 2,36E-12 | -        | 3,60E+10 |
| Dimenze | 100      |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 3,27E-05 | 1,71E-04 | 2,61E-11 | 2,61E-11 |
| Zk-Pop  | 3,07E-05 | -        | 7,00E-01 | 2,47E-11 | 2,47E-11 |
| Transf  | 1,61E-04 | 7,10E-01 | -        | 2,49E-11 | 2,49E-11 |
| DM      | 2,89E-11 | 2,74E-11 | 2,76E-11 | -        | 5,56E-10 |

U Ackelyho funkce jsou výsledky prvních 3 testovaných verzí podobné jako u ostatních funkcí, ovšem rozdíl je zde u poslední verze DE s diskrétní mutací a GA. Ty si touto funkcí při vyšším počtu dimenzí vůbec nedokázaly poradit.

## 9.8 F8: Greiwankova funkce

Tabulka 24. Statistické vyhodnocení pro F8

| Dimenze             | 10     | Minimum: 0 |         |       |       | 2x maxFES |
|---------------------|--------|------------|---------|-------|-------|-----------|
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0      | 0          | 0       | 0,035 | 0,123 | 0         |
| Nejhorší výsledek   | 0,182  | 0,296      | 0,245   | 0,208 | 0,397 | 0,094     |
| Průměrný výsledek   | 0,069  | 0,118      | 0,085   | 0,106 | 0,24  | 0,026     |
| Směrodatná odchylna | 0,045  | 0,072      | 0,068   | 0,044 | 0,071 | 0,024     |
| Medián              | 0,062  | 0,125      | 0,066   | 0,106 | 0,23  | 0,031     |
| Dimenze             | 30     | Minimum: 0 |         |       |       | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0      | 0          | 0       | 0,573 | 0,375 | 0         |
| Nejhorší výsledek   | 0,08   | 0,062      | 0,064   | 1,112 | 0,85  | 0,621     |
| Průměrný výsledek   | 0,007  | 0,003      | 0,004   | 0,969 | 0,636 | 0,27      |
| Směrodatná odchylna | 0,021  | 0,012      | 0,013   | 0,103 | 0,112 | 0,16      |
| Medián              | 0      | 0          | 0       | 1,015 | 0,643 | 0,239     |
| Dimenze             | 100    | Minimum: 0 |         |       |       | 2x maxFES |
| Verze               | Zk-Fce | Zk-Pop     | Transf. | DM    | GA    | DM        |
| Nejlepší výsledek   | 0,045  | 0,018      | 0,03    | 2,357 | 1,655 | 1,152     |
| Nejhorší výsledek   | 0,235  | 0,108      | 0,013   | 5,269 | 2,284 | 3,62      |
| Průměrný výsledek   | 0,121  | 0,059      | 0,063   | 3,822 | 1,897 | 1,628     |
| Směrodatná odchylna | 0,045  | 0,022      | 0,029   | 0,772 | 0,157 | 0,501     |
| Medián              | 0,122  | 0,06       | 0,057   | 3,809 | 1,866 | 1,487     |



Obrázek 24. F8: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty

Tabulka 25. Statistické vyhodnocení pro F8

| Dimenze | 10       |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 3,66E-03 | 4,30E-01 | 1,80E-03 | 1,20E-10 |
| Zk-Pop  | 3,83E-03 | -        | 4,00E-02 | 3,90E-01 | 1,15E-07 |
| Transf  | 4,40E-01 | 4,00E-02 | -        | 3,00E-02 | 4,94E-09 |
| DM      | 1,90E-03 | 3,80E-01 | 3,00E-02 | -        | 1,17E-09 |
| Dimenze | 30       |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 6,20E-01 | 9,40E-01 | 3,68E-12 | 3,68E-12 |
| Zk-Pop  | 6,00E-01 | -        | 6,60E-01 | 2,83E-12 | 2,83E-12 |
| Transf  | 9,20E-01 | 6,80E-01 | -        | 3,68E-12 | 3,68E-12 |
| DM      | 4,10E-12 | 3,16E-12 | 4,11E-12 | -        | 3,47E-10 |
| Dimenze | 100      |          |          |          |          |
|         | Zk-Fce   | Zk-Pop   | Transf   | DM       | GA       |
| Zk-Fce  | -        | 2,02E-07 | 1,10E+06 | 2,72E-11 | 2,73E-11 |
| Zk-Pop  | 1,87E-07 | -        | 8,80E-01 | 2,72E-11 | 2,73E-11 |
| Transf  | 1,02E-06 | 8,90E-01 | -        | 2,72E-11 | 2,73E-11 |
| DM      | 3,01E-11 | 3,01E-11 | 3,01E-11 | -        | 3,01E-11 |

Poslední testovaná funkce dopadla podobně jako většina z testovaných funkcí. Opět se ukázaly jako nejlepší verze diferenciální evoluce s transformací čísel a zaokrouhlováním v populaci. Podle Wilcoxonova Rank-Sum testu zde při nízkém počtu dimenzí nebyla odlišná ani poslední verze diferenciální evoluce, ale při vyšším počtu dimenzí už je opět horší.

## 9.9 Shrnutí výsledků

Celkově dopadly nejlépe verze 2 a 3 diferenciální evoluce. Tedy verze se zaokrouhlováním přímo v populaci a verze s transformací čísel. Mezi těmito dvěma variantami byly jen nepatrné rozdíly a v nejlepší nalezené hodnotě se při různých funkcích střídaly. Průběhy nejlepší nalezené hodnoty u těchto dvou funkcí se prakticky všude překrývají, takže rychlost, jak se dostaly k minimální hodnotě, je stejná. To, že jsou tyto varianty velmi podobné, potvrzuje i Wilcoxonův Rank-Sum test, který tyto 2 verze téměř ve všech případech označil jako stejně kvalitní.

Další v pořadí skončila první verze diferenciální evoluce – zaokrouhlení při dosazení do účelové funkce. Ta při nízkém počtu dimenzí vracela stejné výsledky jako verze 2 a 3, ale s rostoucím počtem dimenzí se její výsledky zhoršují. Její průběh nejlepší nalezené hodnoty je podobný jako u verze 2 a 3, ovšem její výsledky jsou nepatrně horší. Při testování pomocí Wilcoxonova Rank-Sum testu jsou její výsledky pro deseti dimenzionální problémy stejně kvalitní jako u verze s transformací čísel a zaokrouhlováním v populaci.

Celkově nejhůř dopadla verze 4, tedy verze s diskrétní mutací. Její výsledky jsou téměř u všech funkcí podstatně horší než u ostatních variant. Z průběhu nejlepší nalezené hodnoty lze říci, že její konvergence k minimu je nejpomalejší a potřebovala by daleko více ohodnocení pomocí účelové funkce, aby dokázala podávat stejně kvalitní řešení jako ostatní varianty. To potvrzuje testování této varianty při dvojnásobném počtu ohodnocení. Výsledky jsou sice pořád horší než u ostatních variant, ale dochází k výraznému zlepšení nalezených hodnot. Pokud se ovšem podíváme na funkci F5 a konkrétně na výsledky diskrétní mutace, dostaneme potvrzení platnosti No Free Lunch teorému. U této konkrétní funkce se ukázala jako nejlepší funkce s diskrétní mutací. Naopak u funkce F7 si verze s diskrétní mutací při sto dimenzionálním problému nedokázala s touto funkcí poradit vůbec. Takže, i když se tato verze celkově jeví jako nejhorší, vždy záleží na konkrétním problému a v některých případech může být užitečná [1].

Z porovnání testovaných verzí s genetickým algoritmem je vidět, že u všech funkcí byla lepší některá z variant diferenciální evoluce. Při deseti dimenzionálních problémech byly jeho výsledky nejhorší. Ve vyšších dimenzích se genetický algoritmus umístil většinou na čtvrtém místě a jako poslední skončila verze s diskrétní mutací.

## ZÁVĚR

Účelem této diplomové práce bylo navrhnout a otestovat diskrétní variantu diferenciální evoluce ve variantě DISH. Tato varianta prozatím byla navržena pouze ve spojitě oblasti a bylo potřeba vyřešit, jak ji lze převést do diskrétní oblasti.

V teoretické části práce jsou rozebrány základní vlastnosti evolučních algoritmů a základní verze diferenciální evoluce, ze které je tato varianta odvozena. Hlavní část je následně věnována popisu funkce implementované varianty DISH a možnostem, jak ji převést do diskrétní oblasti.

Praktická část je zaměřena na testování zvolených metod diskretizace. Byly zde testovány čtyři diskrétní verze algoritmu – dvě verze se zaokrouhlováním, verze s transformací čísel a verze s diskrétní mutací. Všechny tyto verze byly otestovány na osmi testovacích funkcích pro trojí nastavení dimenzí – 10, 30 a 100. Pro lepší představu o efektivnosti algoritmu byly výsledky porovnány s výsledkem genetického algoritmu, který je již v základní verzi určen pro diskrétní úlohy.

Při testování se ukázalo, že všechny navržené varianty jsou funkční. Nejlepší výsledky podávají verze s transformací čísel a verze se zaokrouhlováním přímo v populaci. Tyto dvě verze podávaly stejně dobré výsledky na většině testovaných funkcí pro všechny testované dimenze. O něco hůře dopadla verze se zaokrouhlováním až při dosazení do účelové funkce. Ta při deseti dimenzionálním problému podávala stejně kvalitní výsledky jako předchozí verze, ale s rostoucím počtem dimenzí již byly výsledky horší.

Nejhůře dopadla verze s diskrétní transformací. Ta podávala podstatně horší výsledky než zbývající verze téměř u všech funkcí. Z průběhu nejlepšího nalezeného řešení je patrné, že její konvergence je nejpomalejší a pravděpodobně by potřebovala daleko větší počet ohodnocení pro nalezení stejně kvalitního řešení jako ostatní verze. Ovšem ani zde nelze říct, že by byla tato verze špatná, jelikož u jedné z testovaných funkcí podávala lepší výsledky než zbylé verze, což jen potvrzuje No Free Lunch teorém – tedy, že žádný algoritmus není univerzální a každý je vhodný pro jiný typ problému.

Při porovnání s genetickým algoritmem si vždy vedly lépe všechny testované verze diferenciální evoluce. Pro deseti dimenzionální problém byly výsledky genetického algoritmu nejhorší a při vyšším počtu dimenzí se většinou umístil na čtvrtém místě před verzí s diskrétní mutací.

Diskrétní varianta DISH algoritmu se jeví jako úspěšné rozšíření evolučních algoritmů. Z porovnání s genetickým algoritmem je zřejmé, že tento algoritmus je úspěšný a určitě najde své uplatnění.

V práci byly vyzkoušeny pouze čtyři varianty, jak převést spojité řešení do diskrétní oblasti. Těchto způsobů ovšem jistě existuje velké množství, a tudíž se nabízí prostor pro další výzkumnou činnost. Dalším potenciálním rozšířením by mohla být úprava algoritmu na permutační problémy. Tato práce tedy může sloužit jako podklad pro další rozvoj diskrétní optimalizace a evolučních algoritmů.

## SEZNAM POUŽITÉ LITERATURY

- [1] ZELINKA, Ivan. *Evoluční výpočetní techniky: principy a aplikace*. Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-218-3.
- [2] VIKTORIN, Adam, Roman SENKERIK, Michal PLUHACEK, Tomas KADAVY a Ales ZAMUDA. DISH Algorithm Solving the CEC 2019 100-Digit Challenge. *2019 IEEE Congress on Evolutionary Computation (CEC)* [online]. IEEE, 2019, 2019, , 1-6 [cit. 2020-03-09]. DOI: 10.1109/CEC.2019.8789936. ISBN 978-1-7281-2153-6. Dostupné z: <https://ieeexplore.ieee.org/document/8789936/>
- [3] VIKTORIN, Adam, Roman SENKERIK, Michal PLUHACEK, Tomas KADAVY a Ales ZAMUDA. DISH Algorithm Solving the CEC 2019 100-Digit Challenge. *2019 IEEE Congress on Evolutionary Computation (CEC)* [online]. IEEE, 2019, 2019, , 1-6 [cit. 2020-03-29]. DOI: 10.1109/CEC.2019.8789936. ISBN 978-1-7281-2153-6. Dostupné z: <https://ieeexplore.ieee.org/document/8789936/>
- [4] CUEVAS, Erik, Daniel ZALDIVAR, Marco PÉREZ-CISNEROS a Marte RAMÍREZ-ORTEGÓN. Circle detection using discrete differential evolution optimization. *Pattern Analysis and Applications* [online]. 2011, **14**(1), 93-107 [cit. 2020-03-29]. DOI: 10.1007/s10044-010-0183-9. ISSN 1433-7541. Dostupné z: <http://link.springer.com/10.1007/s10044-010-0183-9>
- [5] WANG, Ling, Quan-Ke PAN, P.N. SUGANTHAN, Wen-Hong WANG a Ya-Min WANG. A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems [online]. 2010, **37**(3), 509-520 [cit. 2020-03-29]. DOI: 10.1016/j.cor.2008.12.004. ISSN 03050548. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0305054808002608>
- [6] DOERR, Carola, Furong YE, Naama HORESH, Hao WANG, Ofer M. SHIR a Thomas BÄCK. Benchmarking discrete optimization heuristics with IOHprofiler. *Applied Soft Computing* [online]. 2020, **88** [cit. 2020-03-29]. DOI: 10.1016/j.asoc.2019.106027. ISSN 15684946. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S1568494619308099>
- [7] *Benchmark functions* [online]. [cit. 2020-04-05]. Dostupné z: <http://benchmarkfcns.xyz/fcns>
- [8] *EVOLUČNÍ VÝPOČETNÍ TECHNIKY - ZÁKLADNÍ POJMY, KRIZOVÉ STAVY, BENCHMARKING* [online]. In: . [cit. 2020-05-27]. Dostupné z: <https://moodle.utb.cz/mod/folder/view.php?id=365098>
- [9] *Wolfram Mathematica* [online]. [cit. 2020-05-27]. Dostupné z: <https://www.wolfram.com/mathematica/>
- [10] BEHESHTI, Zahra, Siti Mariyam SHAMSUDDIN a Shafaatunnur HASAN. Memetic binary particle swarm optimization for discrete optimization problems. *Information Sciences* [online]. 2015, **299**, 58-84 [cit. 2020-05-30]. DOI: 10.1016/j.ins.2014.12.016. ISSN 00200255. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0020025514011530>
- [11] WOLPERT, D.H. a W.G. MACREADY. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* [online]. **1**(1), 67-82

[cit. 2020-06-25]. DOI: 10.1109/4235.585893. ISSN 1089778X. Dostupné z: <http://ieeexplore.ieee.org/document/585893/>

[12] Storn, Rainer & Price, Kenneth. (1995). Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. *Journal of Global Optimization*. 23.

[13] *GENETICKÉ ALGORITMY* [online]. [cit. 2020-07-12]. Dostupné z: <https://moodle.utb.cz/mod/folder/view.php?id=365098>

[14] Evoluce. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-07-20]. Dostupné z: <https://cs.wikipedia.org/wiki/Evoluce>

[15] Mendelovy zákony dědičnosti. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-07-20]. Dostupné z: Mendelovy zákony dědičnosti



**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

|               |   |
|---------------|---|
| A             | archiv  |
| b             | hranice prohledávaného prostoru   |
| CR            | práh křížení  |
| D             | dimenze   |
| DE            | diferenciální evoluce   |
| DISH          | <b>D</b> istance Based Parameter Adaptation for <b>S</b> uccess- <b>H</b> istory based Differential Evolution |
| DM            | diskrétní mutace  |
| EPSDE         | DE se sadou řídicích parametrů a strategií  |
| F             | mutační konstanta   |
| FES           | počet ohodnocení účelovou funkcí  |
| G             | generace  |
| GA            | genetický algoritmus  |
| JADE          | Adaptive Differential Evolution With Optional External Archive  |
| jDE           | DE s propagací řídicích parametrů   |
| $M_F, M_{CR}$ | paměť starších hodnot $F$ a $CR$  |
| NP            | velikost populace   |
| P             | populace  |
| SaDE          | Self-adaptive differential evolution  |
| $S_F, S_{CR}$ | pomocná paměť v rámci jedné generace pro hodnoty $F$ a $CR$   |
| SHADE         | <b>S</b> uccess- <b>H</b> istory based <b>A</b> daptive <b>D</b> ifferential <b>E</b> volution                |
| Transf.       | transformace čísel  |
| u             | nový jedinec  |
| v             | zmutovaný jedinec   |
| x             | jedinec   |

Zk-Fce      zaokrouhlování při dosazení do účelové funkce

Zk-Pop      zaokrouhlování v populaci

**SEZNAM OBRÁZKŮ**

|   |    |
|---|----|
| Obrázek 1. Evoluční výpočetní techniky – tvorba generace [1].....   | 11 |
| Obrázek 2. No Free Lunch teorém [1].....  | 13 |
| Obrázek 3. Příklad populace v evolučním algoritmu .....   | 14 |
| Obrázek 4. Populace ve dvou dimenzionálním problému a výběr jedinců pro strategii DE/rand/1 .....   | 17 |
| Obrázek 5. Na obrázku <b>a)</b> jsou znázorněni vybraní jedinci a na obrázku <b>b)</b> je znázorněno, jak je tvořen „zmutovaný“ potomek $v_i$ ..... | 17 |
| Obrázek 6. Binomiální křížení.....  | 19 |
| Obrázek 7. Inicializace parametrů [2,3].....  | 26 |
| Obrázek 8. Načtení paměti [2,3].....  | 27 |
| Obrázek 9. Generování CR [2,3] .....  | 27 |
| Obrázek 10. Generování F [2,3] .....  | 27 |
| Obrázek 11. Úprava CR [2,3] .....   | 28 |
| Obrázek 12. Aktualizace archivu [2,3] .....   | 29 |
| Obrázek 13. Změna velikosti populace [2,3].....   | 30 |
| Obrázek 14. Aktualizace paměti [2,3] .....  | 30 |
| Obrázek 15. Diskrétní mutace, $F = 0,5$ [5] .....   | 34 |
| Obrázek 16. Genetický algoritmus – princip [13] .....   | 36 |
| Obrázek 17. F1: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty.....   | 44 |
| Obrázek 18. F2: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty.....   | 46 |
| Obrázek 19. F3: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty.....   | 48 |
| Obrázek 20. F4: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty.....   | 50 |
| Obrázek 21. F5: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty.....   | 52 |
| Obrázek 22. F6: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty.....   | 54 |
| Obrázek 23. F7: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty.....   | 56 |
| Obrázek 24. F8: zobrazení ve 2D prostoru a průběhy nejlepší nalezené hodnoty.....   | 58 |

**SEZNAM TABULEK**

|   |    |
|---|----|
| Tabulka 1. Mutační strategie [1].....             | 18 |
| Tabulka 2. Hodnota a poloha minima funkce F1..... | 40 |
| Tabulka 3. Hodnota a poloha minima funkce F2..... | 40 |
| Tabulka 4. Hodnota a poloha minima funkce F3..... | 40 |
| Tabulka 5. Hodnota a poloha minima funkce F4..... | 41 |
| Tabulka 6. Hodnota a poloha minima funkce F5..... | 41 |
| Tabulka 7. Hodnota a poloha minima funkce F6..... | 41 |
| Tabulka 8. Hodnota a poloha minima funkce F7..... | 41 |
| Tabulka 9. Hodnota a poloha minima funkce F8..... | 42 |
| Tabulka 10. Statistické vyhodnocení pro F1 .....  | 43 |
| Tabulka 11. Wilcoxonův Rank-Sum test pro F1 ..... | 45 |
| Tabulka 12. Statistické vyhodnocení pro F2 .....  | 46 |
| Tabulka 13. Wilcoxonův Rank-Sum test pro F2 ..... | 47 |
| Tabulka 14. Statistické vyhodnocení pro F3 .....  | 48 |
| Tabulka 15. Wilcoxonův Rank-Sum test pro F3 ..... | 49 |
| Tabulka 16. Statistické vyhodnocení pro F4 .....  | 50 |
| Tabulka 17. Wilcoxonův Rank-Sum test pro F4 ..... | 51 |
| Tabulka 18. Statistické vyhodnocení pro F5 .....  | 52 |
| Tabulka 19. Wilcoxonův Rank-Sum test pro F5 ..... | 53 |
| Tabulka 20. Statistické vyhodnocení pro F6 .....  | 54 |
| Tabulka 21. Wilcoxonův Rank-Sum test pro F6 ..... | 55 |
| Tabulka 22. Statistické vyhodnocení pro F7 .....  | 56 |
| Tabulka 23. Wilcoxonův Rank-Sum test pro F7 ..... | 57 |
| Tabulka 24. Statistické vyhodnocení pro F8 .....  | 58 |
| Tabulka 25. Statistické vyhodnocení pro F8 .....  | 59 |

## SEZNAM PŘÍLOH

Příloha P I: Implementace diskrétní varianty DISH v prostředí Mathematica