

Algoritmus Diferenciální Evoluce s prvky deterministického chaosu (ChaosDE) v prostředí Mathematica

Differential evolution algorithm with elements of deterministic
chaos (ChaosDE) in Mathematica software

Bc. Michal Karger

Diplomová práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2010/2011

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal KARGER**
Osobní číslo: **A09507**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Téma práce: **Algoritmus Diferenciální Evoluce s prvky deterministického chaosu (ChaosDE) v prostředí Mathematica**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Naprogramujte několik verzí Diferenciální Evoluce, které budou využívat jako generátor náhodných čísel chaotické systémy.
3. Otestujte algoritmy na sadě vybraných testovacích funkcí.
4. Provedte též srovnávací testy s originální formou Diferenciální Evoluce.
5. Výsledky testování přehledně graficky a tabulkově zobrazte.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. GILMORE, R.; LEFRANC, M. The Topology of Chaos. Wiley VCH, 2002. 518 s. ISBN 978-0-471-40816-1.
2. MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J.: Umělá inteligence, Academia, 1993, ISBN 80-200-0496-3.
3. ZELINKA, I. Umělá inteligence - hrozba nebo naděje. BEN - technická literatura, 2003, ISBN 80-7300-068-7.
4. ZELINKA I., OPLATKOVÁ Z., ŠEDA M., OŠMERA P., VČELAŘ F., Evoluční výpočetní techniky - principy a aplikace, BEN, Praha, 2008, ISBN 80-7300-218-3.
5. MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J.: Umělá inteligence 4., Academia, 2003, ISBN 80-200-1044-0.
6. ZELINKA, Ivan. Aplikovaná Informatika. Zlín. UTB, 1999. 183 s. ISBN 80-214-1423-5.
7. ZELINKA, Ivan. Umělá inteligence v problémech globální optimalizace. BEN, 2002, 190 s. ISBN 80-7300-069-5.
8. SCHUSTER, H. G. Handbook of Chaos Control. 2. Rev. Edition. Wiley VCH, 2007. 849 s. ISBN 978-3527406050.

Vedoucí diplomové práce:

Ing. Roman Šenkeřík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

24. února 2011

Termín odevzdání diplomové práce:

18. května 2011

Ve Zlíně dne 24. února 2011



prof. Ing. Vladimír Vašek, CSc.
děkan



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

ABSTRAKT

Tato diplomová práce se zabývá testováním diferenciální evoluce s prvky deterministického chaosu a porovnáním s klasickou diferenciální evolucí. Teoretická část obsahuje popis a princip diferenciální evoluce, seznámení s chaotickou diferenciální evolucí a základy deterministického chaosu. Zakończena je informacemi o testovacích funkcích, způsobu vyhodnocování a softwaru Mathematica.

Praktická část obsahuje množství simulací, jejichž účelem je porovnat klasickou diferenciální evoluci s diferenciální evolucí, která využívá chaotických systémů. V závěru práce je uvedeno shrnutí, ze kterého jednoznačně vyplývá vliv volby chaotického systému na kvalitu výsledků.

Klíčová slova: Diferenciální evoluce, deterministický chaos, optimalizace, evoluční algoritmy, testovací funkce.

ABSTRACT

This thesis deals with testing of differential evolution with elements of deterministic chaos and its comparison with classical differential evolution. The theoretical part contains a description of the principles of differential evolution, an introduction to chaotic differential evolution, and the basics of deterministic chaos. It concludes with information about test functions, evaluation methods, and Mathematica software.

The practical part contains many simulations whose purpose is to compare classical differential evolution with differential evolution which uses chaotic systems. The conclusion of this thesis contains a summary which clarifies the influence of choice chaotic system on the quality of results.

Keywords: Differential evolution, deterministic chaos, optimization, evolutionary algorithms, test functions.

Velice děkuji Ing. Romanu Šenkeříkovi, Ph.D. za odborné vedení diplomové práce, za ochotnou spolupráci při řešení problémů a za poskytnutí mnoha cenných připomínek a rad. Dále děkuji svým rodičům za podporu během celé doby mého studia.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské 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/bakalářská 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/bakalářskou 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/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské 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/bakalářské 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

.....
podpis diplomanta

OBSAH

| | |
|--|-----------|
| ÚVOD..... | 9 |
| 1 TEORETICKÁ ČÁST | 11 |
| 1 OPTIMALIZAČNÍ ALGORITMY | 12 |
| 2 EVOLUČNÍ VÝPOČETNÍ TECHNIKY | 13 |
| 2.1 POUŽITÍ EVOLUČNÍCH ALGORITMŮ | 13 |
| 2.2 VÝHODY EVOLUČNÍCH ALGORITMŮ | 13 |
| 2.3 EVOLUČNÍ CYKLUS | 14 |
| 2.4 ÚČELOVÁ FUNKCE | 14 |
| 2.5 POPULACE | 15 |
| 2.6 PENALIZACE | 15 |
| 3 DIFERENCIÁLNÍ EVOLUCE..... | 16 |
| 3.1 POPIS ČINNOSTI DIFERENCIÁLNÍ EVOLUCE | 16 |
| 3.2 PARAMETRY DIFERENCIÁLNÍ EVOLUCE | 19 |
| 3.3 MUTACE..... | 20 |
| 3.4 KŘÍŽENÍ..... | 20 |
| 3.4.1 Exponenciální křížení..... | 20 |
| 3.4.2 Binomické křížení | 21 |
| 3.5 STRATEGIE DIFERENCIÁLNÍ EVOLUCE | 22 |
| 3.6 UKONČOVACÍ KRITÉRIA..... | 23 |
| 3.7 STAGNACE..... | 23 |
| 4 DIFERENCIÁLNÍ EVOLUCE S PRVKY DETERMINISTICKÉHO CHAOSU..... | 24 |
| 5 DETERMINISTICKÝ CHAOS..... | 25 |
| 5.1 BIFURKAČNÍ DIAGRAM | 26 |
| 5.2 LORENZŮV MODEL | 27 |
| 5.2.1 Divergence trajektorií..... | 28 |
| 5.3 LOGISTICKÁ ROVNICE | 28 |
| 5.4 DISIPATIVNÍ SYSTÉMY | 30 |
| 5.5 LJAPUNOVŮV EXPONENT | 30 |
| 5.6 PŘEHLED VYBRANÝCH CHAOTICKÝCH SYSTÉMŮ | 31 |
| 6 TESTOVÁNÍ ALGORITMŮ..... | 33 |
| 6.1 TESTOVACÍ FUNKCE..... | 33 |
| 6.2 VYHODNOCOVÁNÍ..... | 35 |
| 7 SOFTWARE MATHEMATICA | 37 |

| | | |
|-----------|--|-----------|
| II | PRAKTICKÁ ČÁST | 38 |
| 8 | KLASICKÁ DIFERENCIÁLNÍ EVOLUCE..... | 39 |
| 9 | NASTAVENÍ PARAMETRŮ..... | 41 |
| 10 | TESTOVÁNÍ CHAOTICKÉ DIFERENCIÁLNÍ EVOLUCE | 44 |
| 10.1 | SCHWEFELOVA FUNKCE | 44 |
| 10.2 | RASTRIGINOVA FUNKCE | 47 |
| 10.3 | ACKLEYHO FUNKCE I..... | 49 |
| 10.4 | ACKLEYHO FUNKCE II | 51 |
| 10.5 | ROZTAŽENÁ SINUSOIDÁLNÍ V FUNKCE..... | 53 |
| 10.6 | MASTERSOVA FUNKCE | 55 |
| 11 | VLIV ROZMĚRU ÚČELOVÉ FUNKCE..... | 58 |
| 11.1 | TESTOVÁNÍ NA ACKLEYHO FUNKCI II..... | 58 |
| 11.2 | TESTOVÁNÍ NA SCHWEFELOVĚ FUNKCI..... | 61 |
| 12 | POROVNÁNÍ ALGORITMŮ | 65 |
| | ZÁVĚR | 67 |
| | CONCLUSION | 68 |
| | SEZNAM POUŽITÉ LITERATURY | 69 |
| | SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK | 71 |
| | SEZNAM OBRÁZKŮ | 72 |
| | SEZNAM TABULEK..... | 73 |
| | SEZNAM PŘÍLOH..... | 75 |

ÚVOD

Evoluční algoritmy se stejně jako mnoho jiných lidských vynálezů inspiřují principy, které nalézáme v přírodě. Je však nutno podotknout, že jde o principy značně zjednodušené. Již delší dobu se tématu evolučních algoritmů věnuje mnoho pozornosti. Jde o obor, k jehož vývoji pozitivně přispívá pokrok v oblasti výpočetní techniky, jež zvyšuje efektivnost a možnosti použití evolučních algoritmů. Obecně lze evoluční algoritmy použít na jakýkoliv problém, který lze popsat matematickým vztahem neboli účelovou funkcí. Úkolem evolučních algoritmů je optimalizovat parametry této funkce tak, abychom dostali nejlepší výsledek. Oproti klasickým matematickým postupům mají evoluční algoritmy mnoho výhod, což je činí oblíbeným nástrojem pro optimalizaci problémů z technické praxe. Spektrum použití těchto algoritmů je značně široké a týká se mnoha aspektů lidské činnosti.

Evolučních algoritmů je celá řada a každý má své specifické vlastnosti a hodí se na jiný typ problému. Neexistuje totiž jeden algoritmus, který by podával excelentní výkony pro všechny myslitelné problémy. Jedním z evolučních algoritmů je i diferenciální evoluce, která je velmi známým pojmem v oblasti optimalizace. Samotná diferenciální evoluce rovněž prochází něčím, co by se dalo nazvat evolucí. Od doby jejího vzniku se objevilo několik verzí diferenciální evoluce a jde tedy o neutuchající vývoj s cílem vytvářet stále lepší algoritmus. Aplikace deterministického chaosu na diferenciální evoluci je jednou z možností, jak ji posunout o krok dále.

Stejně jako evoluční algoritmy, jsou i některé chaotické systémy odvozeny z přírodních jevů nebo zákonitostí. Vzájemné propojení těchto dvou oblastí je hlavní myšlenkou práce.

Cílem práce je otestovat diferenciální evoluci s prvky deterministického chaosu, porovnat získané výsledky s klasickou diferenciální evolucí a poukázat na vliv výběru konkrétního chaotického systému na kvalitu evolučního procesu. Výkonnost jednotlivých algoritmů je potřeba prověřit na sadě testovacích funkcí.

V teoretické části nalezneme informace o principech evolučních výpočetních technik, podrobnější informace o diferenciální evoluci a úvod do deterministického chaosu. Teoretická část rovněž obsahuje popis způsobu testování a vyhodnocování kvality evolučních algoritmů.

Praktická část se zabývá nastavením parametrů a následným testováním algoritmů. K tomuto účelu bylo provedeno mnoho simulací, jejichž výsledky jsou zpracovány formou grafů a tabulek. V závěru praktické části je provedeno srovnání testovaných algoritmů.

I. TEORETICKÁ ČÁST

1 OPTIMALIZAČNÍ ALGORITMY

Optimalizační algoritmy slouží k řešení mnoha úloh z technické praxe. Jejich použití se hodí zejména tam, kde by analytická metoda byla méně vhodná nebo zcela nepoužitelná. Řešené úlohy lze převést na matematický zápis definováním účelové funkce. Aplikací optimalizačních algoritmů na účelovou funkci pak můžeme zjistit její minimum či maximum, kterému bude odpovídat nalezená kombinace argumentů účelové funkce. Optimalizační algoritmy můžeme rozdělit do 4 skupin:

- a) Enumerativní – algoritmus zjišťuje všechny možné kombinace řešeného problému. Enumerativní algoritmy se však hodí pouze tam, kde je malý počet různých kombinací argumentů účelové funkce, které jsou navíc pouze diskrétní.
- b) Deterministické – algoritmus využívá nástrojů klasické matematiky. Pro dosažení uspokojivých výsledků je potřeba, aby řešený problém vyhovoval určitým omezujícím podmínkám, jako je například linearita či souvislost prohledávaného prostoru.
- c) Stochastické – algoritmus zkouší náhodné kombinace argumentů účelové funkce a jako řešení označí nejlepší nalezenou kombinaci. Jejich nevýhodou je pomalost a uplatňují se spíše tam, kde potřebujeme znát pouze hrubý odhad řešení.
- d) Smíšené – do této skupiny patří algoritmy, které obsahují deterministické i stochastické prvky. Výsledkem pak bývají robustní a efektivní algoritmy, které generují kvalitní řešení bez nutnosti analytického popisu problému. Výhodou je také schopnost poskytnout více řešení.

Mezi moderní způsoby řešení optimalizačních problémů patří tzv. evoluční algoritmy, které umožňují velmi efektivní řešení problematických úloh. Pokud bychom měli evoluční algoritmy zařadit do jedné z výše uvedených skupin, byla by to skupina smíšených algoritmů. [1]

2 EVOLUČNÍ VÝPOČETNÍ TECHNIKY

Evoluční výpočetní techniky představují optimalizační numerické algoritmy, které čerpají z Darwinovy a Mendelovy teorie evoluce. S touto myšlenkou však již přišli mnohem dříve starověcí myslitelé (například Anaximandros z Milétu). Teorie evoluce uznává evoluční dogma, které říká, že jednotlivé druhy se vyvíjejí způsobem, kdy rodiče plodí nové potomky, kteří jsou během svého vzniku vystaveny mutacím. Nevhodní a slabí jedinci během jednotlivých generací vymírají a tím poskytují prostor silnějším jedincům, kteří se následně stávají novými rodiči. Aplikací těchto principů na numerické algoritmy vznikají evoluční algoritmy. [1]

2.1 Použití evolučních algoritmů

Evoluční algoritmy nacházejí uplatnění ve velmi širokém spektru aplikací. Můžeme se s nimi setkat při optimalizaci ekonomických problémů, tvarů zařízení nebo optimalizaci dopravních a energetických problémů. [2]

Mezi konkrétní příklady použití evolučních algoritmů můžeme uvést řízení plazmového reaktoru v reálném čase, aerodynamickou optimalizaci geometrie křídla letadla, optimalizaci směrovacích stanic v bezdrátových sítích, návrh elektronických obvodů nebo řízení vlakových výhybek. [1]

2.2 Výhody evolučních algoritmů

Evoluční algoritmy lze poměrně jednoduše naprogramovat a v porovnání s klasickými metodami je lze označit za rychlejší. V algoritmech lze bez potíží kombinovat čísla typu integer nebo real a není nutné jedince převádět do binární reprezentace. Evoluční algoritmy se hodí pro hledání extrémů funkcí, které obsahují šum, vysoký počet dimenzí nebo funkcí multimodálních (obsahující více lokálních extrémů). Neméně vhodnou vlastností je schopnost poskytnout vícenásobné řešení, kterého dosáhneme tím, že z finální populace vybereme určitý počet nejlepších jedinců. Jedním z důvodů oblíbenosti evolučních algoritmů je také to, že v případě správného použití umožňují nahradit člověka. [1]

2.3 Evoluční cyklus

Evoluce je opakovaně prováděna pomocí evolučního cyklu. Evoluční cyklus lze obecně definovat jako posloupnost následujících kroků:

- a) Definice parametrů včetně řídicích a ukončovacích parametrů
- b) Vygenerování prvopočáteční náhodné populace
- c) Ohodnocení kvality všech jedinců
- d) Výběr rodičů podle jejich kvality
- e) Tvorba potomků
- f) Mutace potomků
- g) Ohodnocení nových potomků
- h) Výběr nejlepších jedinců z populace
- i) Naplnění nové populace vybranými nejlepšími jedinci
- j) Stará populace vymírá a je nahrazena novou populací

Kroky d-j se opakují, dokud neuplyne předem stanovený počet generací nebo nedojde ke splnění jiné ukončovací podmínky. [1]

2.4 Účelová funkce

Účelová funkce je funkce, kterou se snažíme optimalizovat (hledáme minimum nebo maximum), čímž nalezneme optimální hodnoty argumentů této funkce. Na účelovou funkci je možné nahlížet jako na geometrický problém, ve kterém hledáme extrém na N - rozměrné ploše. Tuto plochu můžeme také nazývat hyperplochou nebo prostorem možných řešení. Má-li účelová funkce 6 argumentů, pak hledáme extrém na šestirozměrné ploše v sedmirozměrném prostoru. Sedmý rozměr zde bude představovat návratovou hodnotu účelové funkce. Definice účelové funkce je klíčovým prvkem všech evolučních algoritmů, jelikož může významně ovlivnit kvalitu dosažených výsledků. [3]

2.5 Populace

Populace tvoří množinu argumentů účelové funkce, ze které hledáme optimální kombinaci. Každý jedinec z populace s sebou nese hodnotu o své vhodnosti, což je vyčíslení hodnoty účelové funkce (někdy označováno jako fitness). Evoluční algoritmy opakovaně vytváří nové populace, které nahrazují populace předchozí. Tato náhrada je prováděna podle předem stanovených matematických pravidel, které od sebe odlišují jednotlivé evoluční algoritmy. [1]

Pro vygenerování počáteční populace je potřeba použít vzorového jedince, který se označuje jako specimen. Specimen se také využívá při korekci jedinců, kteří vybočí z prohledávaného prostoru. Vzorového jedince definujeme následovně:

$$\text{Specimen} = \{ \{ \text{Real}, \{ \text{Lo}, \text{Hi} \} \}, \{ \text{Integer}, \{ \text{Lo}, \text{Hi} \} \}, \dots, \{ \text{Real}, \{ \text{Lo}, \text{Hi} \} \} \} \quad (1)$$

Specimen definuje pro každý parametr typ (celočíslný, reálný) a dále spodní a horní hranici intervalu, ve které se může hodnota parametru nacházet. Volba intervalu má velký vliv na to, zda bude nalezené řešení fyzikálně realizovatelné (při nesprávně zvoleném intervalu, by mohl výsledek představovat například záporný rozměr nebo jiné absurdní řešení). [3]

2.6 Penalizace

Penalizace označuje omezení určitých hodnot argumentů účelové funkce (jedinců), kteří se nachází v nepřipustných oblastech. Jednou z možností penalizace je úplné zrušení jedince a náhodné nahrazení jedincem novým, který se bude vyskytovat mimo nepřipustnou oblast. Tato metoda se označuje hard-constraints. Druhá metoda penalizace nazývaná jako soft-constraints jedince znevýhodňuje úpravou hodnoty účelové funkce. Jedinec tedy není zcela zrušen, ale jeho vhodnost se snižuje. Penalizace je zaváděna z toho důvodu, abychom se vyhnuli hodnotám účelové funkce, která jsou nerealizovatelná nebo nevýhodná. [1]

3 DIFERENCIÁLNÍ EVOLUCE

Diferenciální evoluce (zkráceně DE) vychází z algoritmu genetického žhání, který byl publikován Kennethem V. Pricem v roce 1994. K. V. Price následně navázal spolupráci s Dr. Rainierem Stornem a došlo k úpravě genetického žhání pro řešení složitějších problémů. Postupně vzniklo několik verzí diferenciální evoluce, která se stala vhodným algoritmem pro numerickou optimalizaci. Název diferenciální evoluce popisuje vlastní princip algoritmu, kdy k vygenerování zkušebního výsledku dochází tak, že diferencii dvou náhodných jedinců přičítáme ke třetímu jedinci z populace. [3]

Diferenciální evoluce byla s úspěchem použita na mnohé optimalizační problémy a její výsledky byly často výrazně lepší při porovnání s jinými algoritmy. Vzhledem k těmto výsledkům lze diferenciální evoluci považovat za poměrně robustní a diverzibilní algoritmus. [3]

Diferenciální evolucí se ve svých pracích zabývá mnoho vědců z celého světa. Mezi nejznámější patří: Jouni Lampinen, Ivan Zelinka, Feng-Sheng Wang, Josef Tvrđík nebo Montaz Ali. [1]

3.1 Popis činnosti diferenciální evoluce

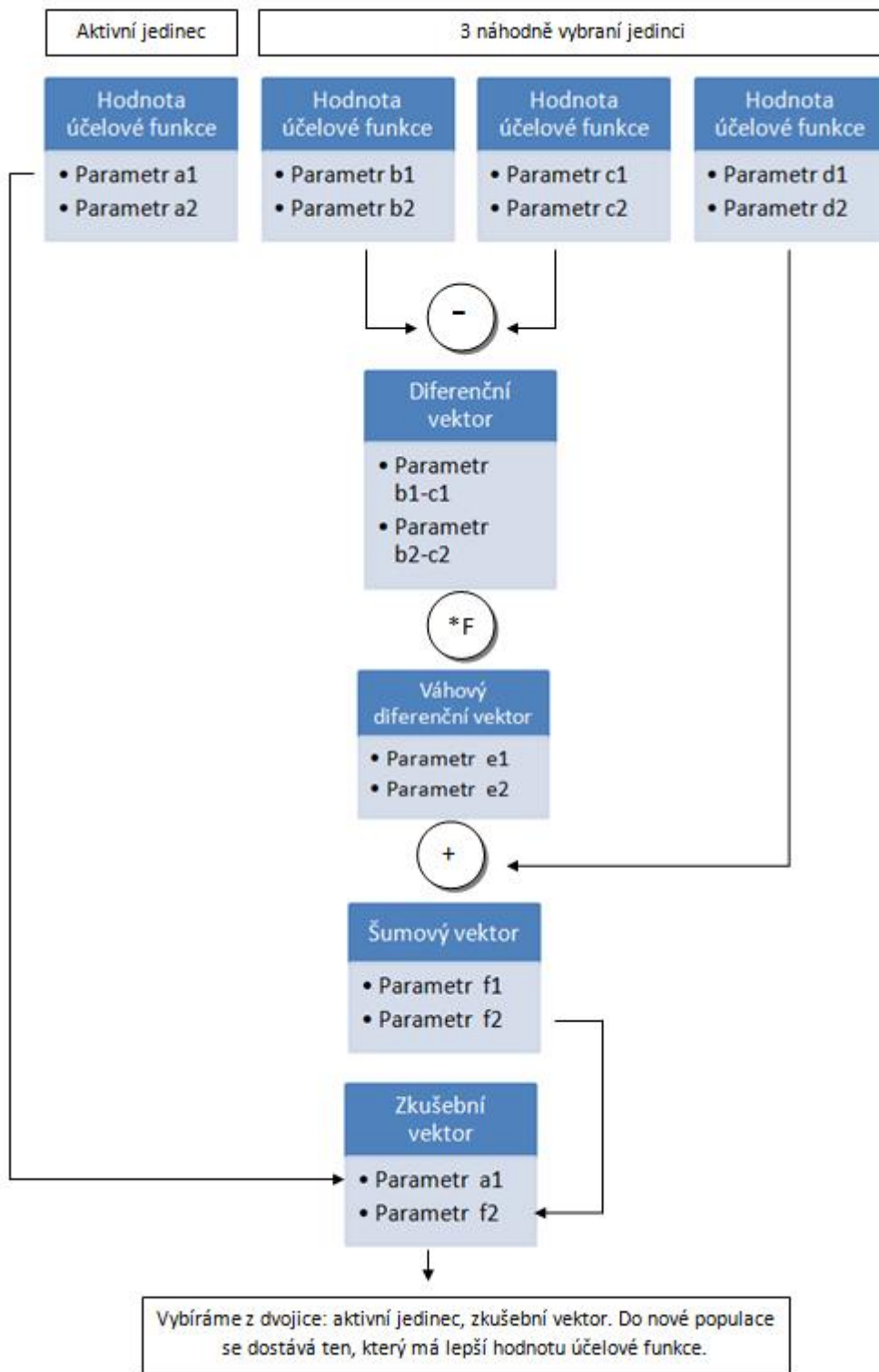
Diferenciální evoluce probíhá cyklicky v generacích s cílem nalézt nejvhodnější populaci jedinců. V této kapitole je popsán princip činnosti strategie DE/rand/1/bin. Jednotlivé generace probíhají následujícím způsobem:

- a) Nejprve je potřeba nastavit parametry DE, které ovlivňují průběh evoluce. Jedná se o mutační konstantu F , práh křížení CR , počet jedinců v populaci NP , počet argumentů účelové funkce D a vzorového jedince.
- b) Dle vzorového jedince se vygeneruje prvotní populace jedinců.
- c) Postupně vybíráme každého jedince z populace a s každým z nich provádíme následující operace. Aktuálně vybraný jedinec bývá označován jako aktivní.
- d) Náhodně vybereme tři další jedince z populace. První dva jedince od sebe odečteme, čímž vznikne diferenční vektor. Diferenční vektor vynásobíme mutační konstantou F , která jej tímto způsobem zmutuje ve váhový diferenční vektor. Tento vektor přičteme ke třetímu jedinci, čímž získáváme šumový vektor. Poté vezmeme

vždy z aktivního jedince a šumového vektoru jeden prvek a pro tento pár vygenerujeme náhodné číslo v intervalu 0-1, které se porovná s prahem křížení CR. Pokud je náhodné číslo menší než CR, pak do nového zkušebního jedince vložíme prvek z šumového vektoru. Pokud je naopak náhodné číslo větší než CR, umístíme do zkušebního jedince prvek z aktivního jedince. Tímto způsobem postupujeme, dokud nevytvoříme zkušebního jedince se všemi prvky. Nakonec provedeme výpočet hodnoty účelové funkce aktivního a zkušebního vektoru a dle této hodnoty vybereme vhodnějšího jedince do nové populace. Celý tento proces vytváří evoluční cyklus.

- e) Po skončení evolučního cyklu zjišťujeme, zda jsou splněny ukončovací podmínky. Nejčastějším ukončovacím parametrem je dosažení určitého počtu generací. Alternativním ukončovacím parametrem může být například podmínka, která zjišťuje, jestli se během několika posledních generací změnila hodnota nejlepšího jedince.
- f) V každé generaci se uchovává hodnota účelové funkce nejvhodnějšího jedince, kterou můžeme zakreslením do grafu použít k vyhodnocení evolučního procesu.

Body a) a b) se provádí pouze na samotném začátku algoritmu. Body c) až f) se pak cyklicky opakují až do konce běhu algoritmu. Na obr. 1 je graficky znázorněn princip diferenciální evoluce. [1]



Obr. 1. Zjednodušené schéma diferenciální evoluce [1]

3.2 Parametry diferenciální evoluce

- CR – práh křížení, který určuje, zda použijeme prvek z šumového vektoru nebo z aktivního jedince. Hodnoty se mohou pohybovat v rozmezí 0-1. Nastavíme-li CR = 0, evoluce se zastaví. Při hodnotě CR = 1 se bude algoritmus chovat jako náhodné hledání, které nebere ohled na kvalitu jedinců. Je tedy vhodné se těmito mezními hodnotám vyhnout.
- D – počet argumentů účelové funkce neboli dimenze problému. Parametr D je dán zadaným problémem a lze jej změnit pouze tehdy, pokud přeformulujeme řešený problém.
- NP – velikost populace jedinců. Nejmenší možnou hodnotou je 4.
- F – mutační konstanta, která se podílí na vytvoření váhového diferenčního vektoru. Hodnota se pohybuje v rozmezí 0-2.
- Generace – vyjadřuje, kolik generací (evolučních cyklů) proběhne. Nastavení tohoto parametru závisí na samotném uživateli.

Doporučené hodnoty řídicích parametrů jsou uvedeny v tab. 1. [1]

Tab. 1. Doporučené hodnoty řídicích parametrů [1]

| Parametr | Interval | Doporučená hodnota | Poznámka |
|----------|------------|--------------------|--|
| NP | [10D,100D] | 10D | Je-li funkce vysoce multimodální, volíme 100D. |
| F | [0,2] | 0,3-0,9 | - |
| CR | [0,1] | 0,8-0,9 | Pokud je funkce separabilní, volíme CR blízké 0. Je-li neseparabilní, CR nastavíme na hodnotu blížíící se 1. |

Parametry lze měnit i během evoluce na základě toho, jak kvalitně probíhá. V případě, že se k nastavení řídicích parametrů diferenciální evoluce použije samotná diferenciální evoluce, hovoříme o meta-diferenciální evoluci. Jde tedy o diferenciální evoluci na diferenciální evoluci.

Je zřejmé, že hodnoty řídicích parametrů výrazně ovlivňují kvalitu diferenciální evoluce.

Kromě řídicích parametrů ovlivňuje kvalitu evolučního procesu i počet generací, definice účelové funkce a definice omezení. [1]

Malá populace znamená horší výběr jedinců a velká zase bude potřebovat více času na vytvoření nové populace. Při nízkém počtu generací může být evoluce ukončena dříve, než nalezneme nejlepšího jedince. Nevhodně nadefinovaná účelová funkce může znamenat zpomalení procesu evoluce nebo úplnou nemožnost evoluce. Pomocí vzorového jedince je také potřeba vhodně definovat oblast působnosti evoluce, aby se jedinci nepohybovali v nepovolených oblastech nebo nekonvergovali k nekonečnu. [3]

3.3 Mutace

Mutace označuje způsob, jakým vytváříme šumový vektor. K mutaci jsou použity 3 za 4 jedinců vybraných k evolučnímu procesu. Odečtením prvních dvou dostáváme diferenční vektor, který následně vynásobíme mutační konstantou F . Matematický zápis mutace je uveden níže. [3]

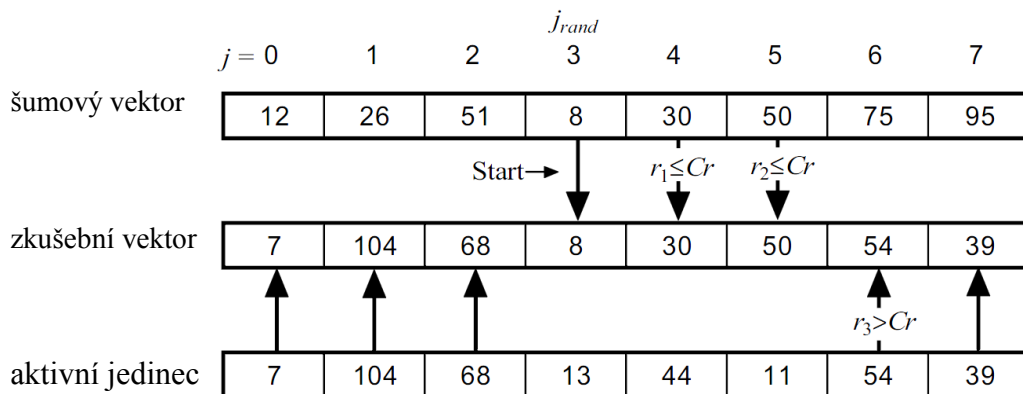
$$v_j = x_{r3,j}^G + F(x_{r1,j}^G - x_{r2,j}^G) \quad (2)$$

3.4 Křížení

Křížení je dalším krokem po provedení mutace a označuje způsob, kterým je z šumového vektoru a aktivního jedince vytvořen zkušební vektor, který se následně uchází o místo v nové populaci. [1]

3.4.1 Exponenciální křížení

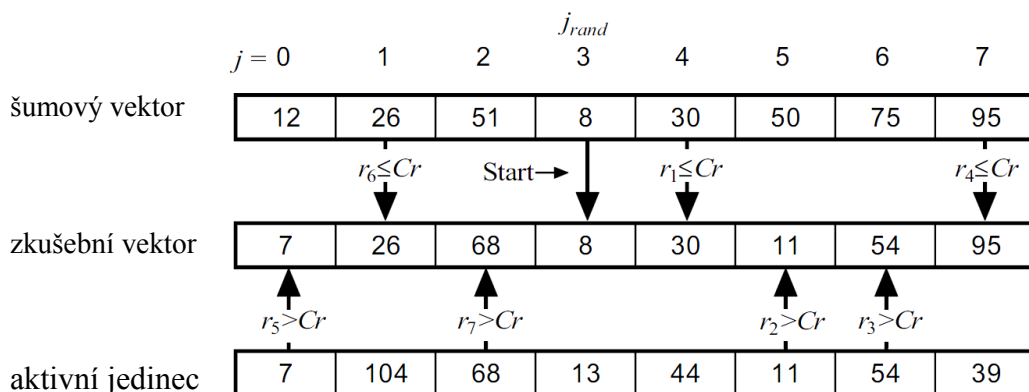
U exponenciálního křížení náhodně vybereme jeden z parametrů jedince, od kterého začne křížení. U tohoto parametru přímo zkopírujeme hodnotu z šumového vektoru do zkušebního vektoru. U každého dalšího parametru pak standardně vygenerujeme náhodné číslo, a dokud je toto číslo menší nebo rovno CR , pokračujeme v kopírování hodnot z šumového vektoru do zkušebního vektoru. Jakmile však hodnota náhodného čísla překročí CR , použijeme pro zbývající parametry automaticky hodnotu z aktivního jedince. Na obr. 2 nalezneme grafické znázornění exponenciálního křížení. [4]



Obr. 2. Ukázka exponenciálního křížení [4]

3.4.2 Binomické křížení

Při použití binomického křížení opět náhodně vybereme první křížený parametr. Hodnota tohoto parametru bude přímo zkopírována tak, jako u exponenciálního křížení. Zde však podobnost s exponenciálním křížením končí, jelikož hodnotu všech následujících parametrů bude určovat náhodně vygenerované číslo. Pokud bude toto číslo menší nebo rovno CR, použijeme parametr z šumového vektoru. V opačném případě bude použita hodnota parametru aktivního jedince. Princip binomického křížení je uveden na obr. 3. [4]



Obr. 3. Ukázka binomického křížení [4]

Při porovnání exponenciálního a binomického křížení jsou dosahovány podobné výsledky, které se výrazným způsobem neliší. [5]

3.5 Strategie diferenciální evoluce

Existuje mnoho strategií algoritmu diferenciální evoluce, které se liší způsobem výpočtu šumového vektoru a způsobem plnění zkušebního vektoru. [1]

Strategie diferenciální evoluce se zapisují dle konvence DE/x/y/z. Výraz na pozici x značí způsob, jakým bude vybírán aktivní jedinec, y značí počet diferenčních vektorů a výraz z popisuje použitý typ křížení (exponenciální nebo binomické). [5]

Mezi jednotlivé strategie diferenciální evoluce patří:

$$\text{DE/best/1/exp} \quad v = x_{best,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G) \quad (3)$$

$$\text{DE/rand/1/exp} \quad v = x_{r1,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G) \quad (4)$$

$$\text{DE/rand-to-best/1/exp} \quad v = x_{i,j}^G + \lambda \cdot (x_{best,j}^G - x_{i,j}^G) + F \cdot (x_{r1,j}^G - x_{r2,j}^G) \quad (5)$$

$$\text{DE/best/2/exp} \quad v = x_{best,j}^G + F \cdot (x_{r1,j}^G + x_{r2,j}^G - x_{r3,j}^G - x_{r4,j}^G) \quad (6)$$

$$\text{DE/rand/2/exp} \quad v = x_{5,j}^G + F \cdot (x_{r1,j}^G + x_{r2,j}^G - x_{r3,j}^G - x_{r4,j}^G) \quad (7)$$

$$\text{DE/best/1/bin} \quad v = x_{best,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G) \quad (8)$$

$$\text{DE/rand/1/bin} \quad v = x_{r1,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G) \quad (9)$$

$$\text{DE/rand-to-best/1/bin} \quad v = x_{i,j}^G + \lambda \cdot (x_{best,j}^G - x_{i,j}^G) + F \cdot (x_{r1,j}^G - x_{r2,j}^G) \quad (10)$$

$$\text{DE/best/2/bin} \quad v = x_{best,j}^G + F \cdot (x_{r1,j}^G + x_{r2,j}^G - x_{r3,j}^G - x_{r4,j}^G) \quad (11)$$

$$\text{DE/rand/2/bin} \quad v = x_{5,j}^G + F \cdot (x_{r1,j}^G + x_{r2,j}^G - x_{r3,j}^G - x_{r4,j}^G) \quad (12)$$

3.6 Ukončovací kritéria

Pokud předem známe globální extrém optimalizované funkce, můžeme algoritmus ukončit v okamžiku, kdy je nejlepší nalezená hodnota v určité toleranci od globálního extrému.

Neznáme-li dopředu globální extrém, je možné algoritmus ukončit po uplynutí předem stanoveného počtu generací. Počet generací by měl být dostatečně velký na to, aby algoritmus dokázal najít hledaný extrém. [4]

Další možností je ukončení algoritmu v momentu, kdy statistika populace dosáhne určité hodnoty. Může to být například rozdíl mezi nejlepším a nejhorším jedincem v populaci. Při nesprávném použití však může dojít k předčasnému ukončení evolučního procesu. Jako ukončovací kritérium může být použit i čas. Algoritmus bude mít pouze omezenou dobu na provedení výpočtu a po uplynutí této doby se ukončí. Tato možnost může být využita například v online optimalizaci nebo tam, kde je potřeba dodržet stanovené časové termíny. [4]

3.7 Stagnace

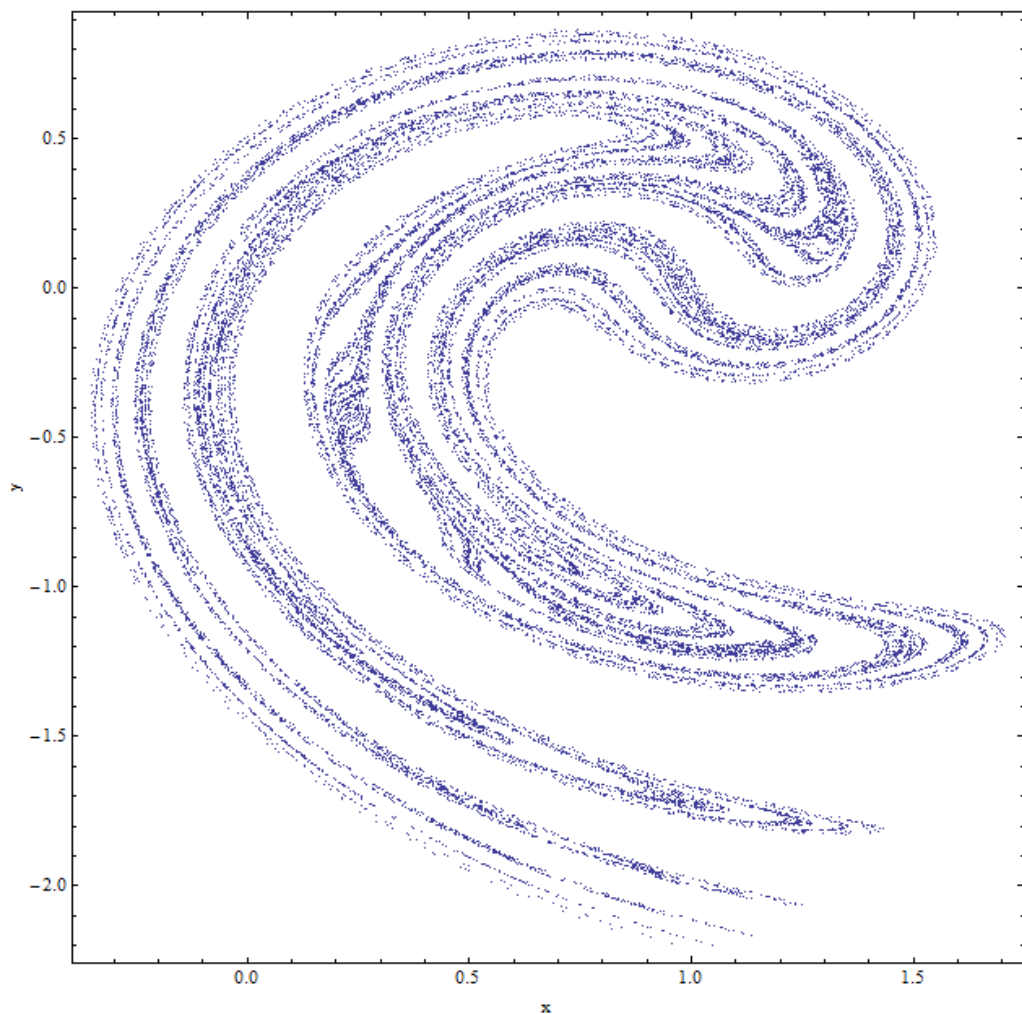
Stagnace je situace, kdy dochází k zastavení vývoje hodnoty účelové funkce před dosažením hledaného globálního extrému. U evolučních algoritmů se stává, že hodnota účelové funkce předčasně konverguje k suboptimálnímu řešení. To je způsobeno zavedením populace do lokálního extrému, ztrátou diverzibility populace nebo zpomalením či úplným zastavením evolučního procesu. [1]

Stagnace znamená, že došlo k zastavení optimalizačního procesu i přesto, že populace nezkonvergovala do lokálního extrému, populace neztratila diverzibilitu a dochází k vytváření nových jedinců. [1]

Riziko stagnace je možné snížit použitím větší populace. Nevýhodou tohoto řešení je pomalejší nalezení optimálního jedince. [3]

4 DIFERENCIÁLNÍ EVOLUCE S PRVKY DETERMINISTICKÉHO CHAOSU

Jak již bylo uvedeno, jednou z příčin stagnace je i ztráta diverzibility populace. Z hlediska diverzibility je možné diferenciální evoluci upravit přidáním chaotického generátoru. Takto upravený algoritmus můžeme nazvat chaotickou diferenciální evolucí. Spojení diferenciální evoluce a chaotických sekvencí může být pro kvalitu algoritmu poměrně výhodné. U klasické diferenciální evoluce se při vytváření zkušebního vektoru porovnává číslo získané generátorem náhodných čísel s prahem křížení. Chaotická diferenciální evoluce místo náhodného generátoru využívá chaotický systém, což má pozitivní vliv na diverzibilitu populace a snižuje riziko stagnace evolučního procesu. [6] Jako chaotický systém můžeme využít například tyto systémy: Lozi, Ikeda, Sinai, Duffing, Tinkerbell, Arnold, Baker, Nose-Hoover, Henon-Heiles. Atraktor Ikeda je zobrazen na obr. 4. [1]



Obr. 4. Systém Ikeda v prostředí Mathematica

5 DETERMINISTICKÝ CHAOS

Existuje mnoho definicí chaotického chování. Ve většině z nich jsou za chaotický proces považována řešení diferenciálních nebo diferenčních rovnic, která jsou charakterizována lokální nestabilitou a globálním omezením. Znamená to, že řešení, která mají nepatrně odlišné počáteční podmínky, po určité době divergují k jisté konečné vzdálenosti. [7]

Chaotické chování při běžném pohledu vypadá téměř úplně jako náhodný systém, který je ovlivňován vnějším zcela náhodným šumem nebo složitým chováním systému s mnoha stupni volnosti. Ukázalo se však, že chaotické chování je generováno velmi jednoduchými systémy, které neobsahují téměř žádné prvky šumu. Ve skutečnosti jsou chaotické systémy v zásadě deterministické, takže podrobná znalost jejich struktury nám umožňuje předpovídat budoucí stavy systému. Důležitou roli zde hraje nelinearita systému. Pokud do systému provedeme určitý zásah, pozorujeme odpovídající odezvu systému. Pokud bude tento zásah dvakrát větší a i odezva bude dvakrát větší, jde o lineární systém. Bude-li naopak odezva systému jiná, můžeme takový systém nazývat nelineárním. Zkoumáním nelineárního chování se zabývá nelineární dynamika. [8]

Některé náhlé a dramatické změny v nelineárních systémech mohou způsobit vznik chaotického chování. Chaotický systém obecně definují:

- a) Časově závislé rovnice
- b) Hodnoty parametrů popisující systém
- c) Počáteční podmínky

Systém nazýváme deterministický, pokud jsme schopni při znalosti všech 3 výše uvedených bodů určit následující stavy systému. [8]

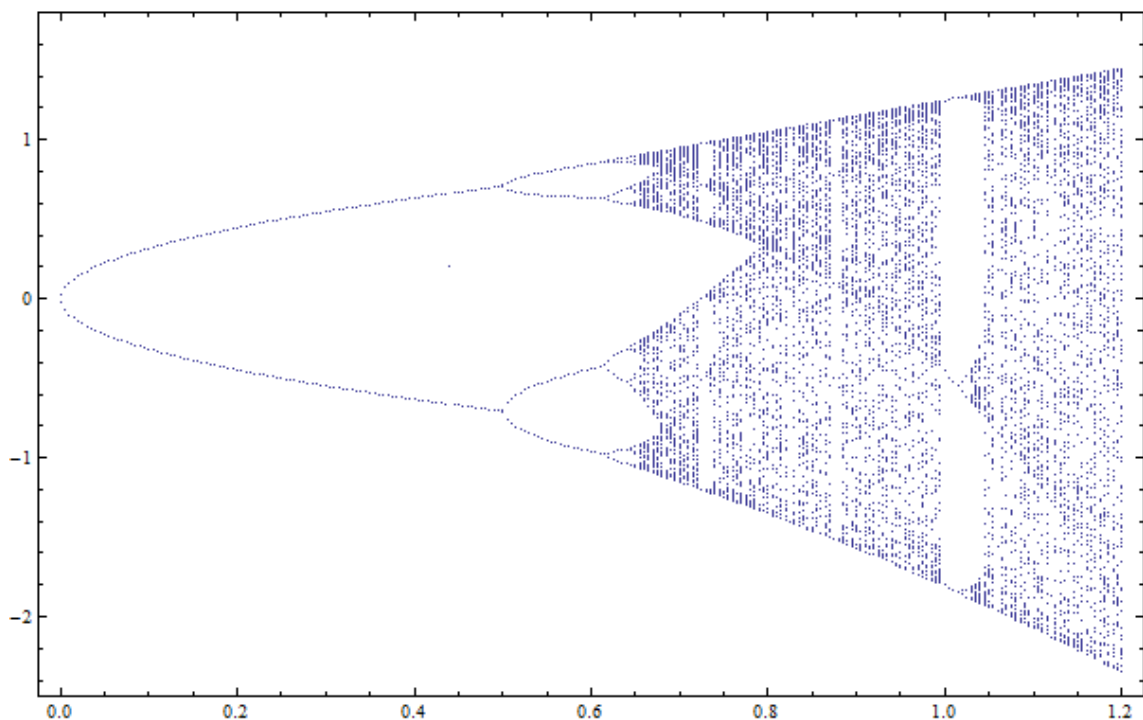
Deterministický chaos objevil v roce 1963 Edward Lorenz a jeho základním principem je soustava diferenciálních rovnic, která se vyznačuje chaotickým chováním. Základní dělení chaotických systému je na spojitě a diskrétní. Výzkum chaosu je v dnešní době slibný vědecký směr s možností praktického uplatnění (využití v kryptografii při přenosu informací či při řízení chaotického chování). [1]

Jedním z důvodů, proč se tolik vědců, inženýrů a matematiků zajímá o chaos je to, že chaotické chování se ukazuje jako velmi univerzální, jelikož ho můžeme najít v elektrických obvodech, chemických reakcích, nervových buňkách nebo laserech. [8]

5.1 Bifurkační diagram

Pro zobrazování chaotického chování se používá bifurkačních diagramů. Jde o graf reprezentující závislost chování systému na řídicím parametru. Ukázku bifurkačního diagramu nalezneme na obr. 5. Při pohledu na bifurkační diagram vidíme v částech grafu plné křivky, které znázorňují ustálený stav systému. [1]

Naopak široké skvrny plné rozptýlených bodů označují, že systém v této části vykazuje chaotické chování. Samotný pojem bifurkace znamená dělení na dvě části. V oblasti nelineární dynamiky se používá k označení jakékoliv náhlé změny chování systému na základě změny parametru. [8]



Obr. 5. Ukázka bifurkačního diagramu

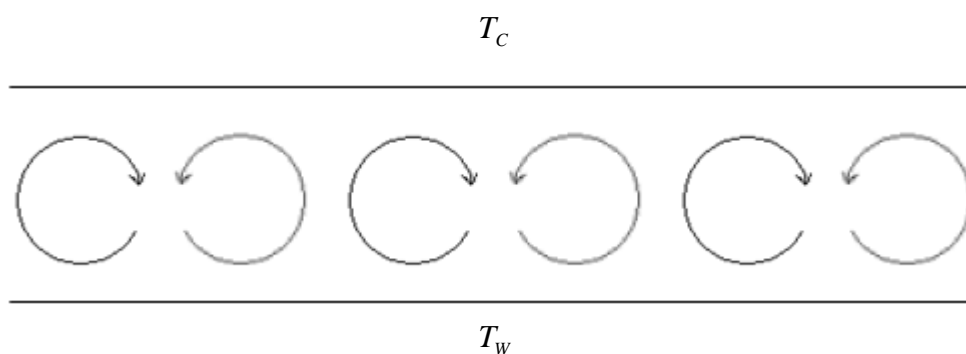
5.2 Lorenzův model

Model byl představen roku 1963 meteorologem Edwardem Lorenzem a jde o zjednodušený model cirkulace kapaliny. Lorenzův model popisuje atmosféru jako vrstvu, která je zespondu ohřívána a z vrchní části ochlazována. Spodní část atmosféry má teplotu T_w , která je vyšší než teplota vrchního okraje označovaná jako T_c . U modelu se předpokládá, že rozdíl teplot $\delta T \equiv T_w - T_c$ je udržován na konstantní hodnotě. Zjednodušené schéma Lorenzova modelu je uvedeno na obr. 6.

V případě, že rozdíl teplot není příliš velký, teplo je vedeno směrem nahoru a díky předávání tepla okolnímu médiu lineárně klesá teplota z hodnoty T_w na hodnotu T_c .

Pokud však bude rozdíl teplot dostatečně velký, dostane se teplá kapalina až na vrchol, kde se ochladí. Následkem toho klesne dolů, kde se opět začíná ohřívát a kapalina bude takto opakovaně cirkulovat.

Při ještě vyšším rozdílu teplot se začnou cirkulační proudy a výsledné teplotní rozdíly $T_w - T_c$ měnit v závislosti na čase. To je typickým příkladem nelineárního chování. I přesto, že je prostředí v čase stabilní, systém spontánně vykazuje časově závislé chování. [8]



Obr. 6. Cirkulace kapaliny v Lorenzově modelu [8]

5.2.1 Divergence trajektorií

Lorenz se snažil ukázat, že i velmi jednoduchá soustava rovnic může generovat chování, které je v podstatě nepředvídatelné. Lorenzův model tvoří soustava tří diferenciálních rovnic, která obsahuje proměnné X, Y a Z.

Mějme dvě trajektorie Lorenzova modelu. První s hodnotami $X=20$, $Y=0$, $Z=163$ a druhou $X=20$, $Y=0$, $Z=164$. Již po několika oscilacích se trajektorie výrazně liší. Jestliže systém pro určitý rozsah svých parametrů vykazuje divergenci blízkých trajektorií, chování takového systému se stává nepředvídatelné. Systém je tedy sice stále deterministický (vzhledem k tomu, že při znalosti přesných počátečních podmínek jsme schopni předpovídat budoucí stavy systému), ale při velmi malé změně počátečních parametrů se trajektorie vydává zcela jiným směrem. Jakákoliv malá změna počátečních parametrů tedy vede k podstatně jinému dlouhodobému chování, které nemůžeme přesně předvídat. Takovéto chování se začalo metaforicky nazývat efektem motýlích křídel. [8]

5.3 Logistická rovnice

Logistická rovnice je nejjednodušším modelem deterministického chaosu a vznikla pro účely simulace biologických procesů. Příkladem takového procesu může být soužití 2 druhů v uzavřeném prostředí, kdy jeden druh slouží jako potrava druhého. Následkem bude to, že konzumovaný druh bude decimován, ale jakmile bude jedinců tohoto duhu nedostatek, bude naopak vymírat hlady opačný druh, čímž se opět zvýší počet jedinců prvního druhu. Z popisu je zřejmé, že populace obou druhů budou periodicky oscilovat nebo se ustálí na konstantní hodnotě. Na základě simulací bylo dokázáno, že takový systém může vykazovat velmi složité chaotické chování. [1],[9]

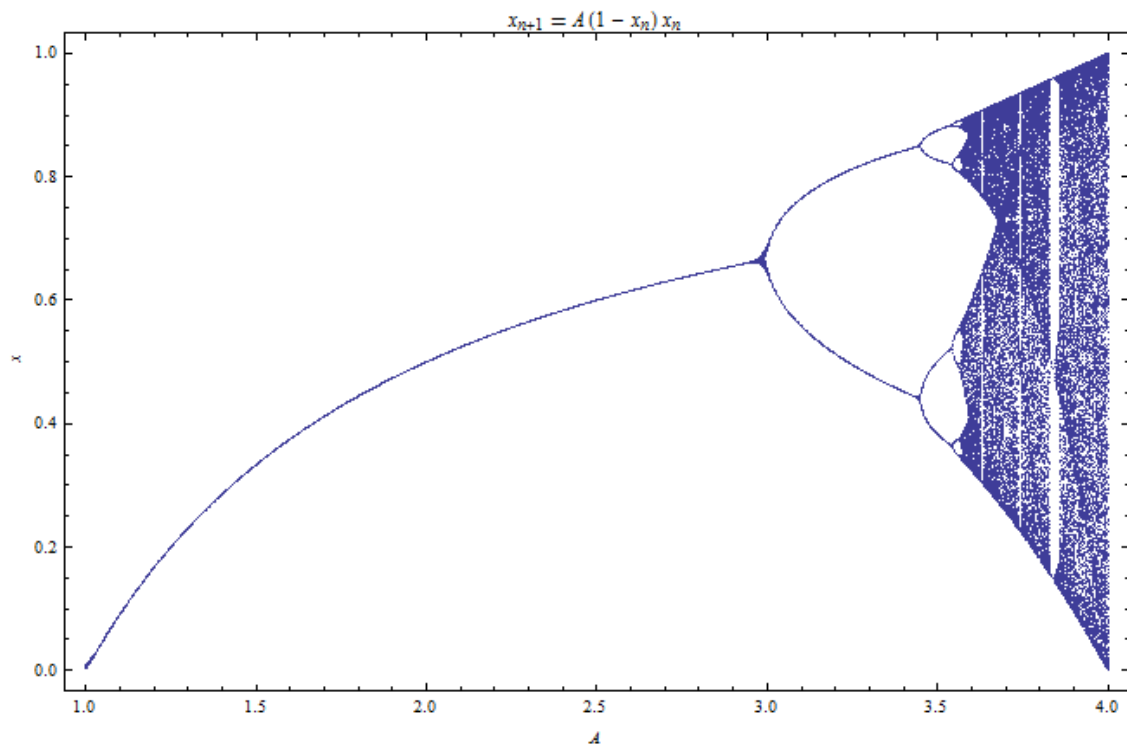
$$x_{n+1} = Ax_n(1 - x_n) \equiv f_A(x_n) \quad (13)$$

Ve výše uvedeném matematickém zápisu logistické rovnice vyjadřuje hodnota x_n populaci v n-tém roce. Tuto funkci můžeme nazvat jako iterační, protože k dosažení výsledku je potřeba provést určitý počet iterací. Pokud pomocí iterace vygenerujeme sekvenci hodnot x , dostáváme trajektorii (orbit). Počátky trajektorií závisí na počátečné hodnotě x . Další

průběhy trajektorií jsou však stejné pro téměř všechny počáteční hodnoty x v intervalu $[0,1]$. Některé počáteční hodnoty x se však od ostatních odlišují. Příkladem může být, zvolíme-li $x_0 = 0$. Hodnota funkce $f_A(x_0)$ pak bude také 0 a na této hodnotě zůstává i v následujících iteracích. Hodnoty x , které po dosazení do iterační funkce generují stejný výsledek jako je jejich hodnota, se nazývají fixními body. Fixní bod lze definovat následujícím vztahem. [8]

$$x_A = f_A(x_A) \quad (14)$$

Pokud se všechny trajektorie pro určité počáteční hodnoty s rostoucím počtem iterací přibližují danému bodu, pak se tento bod nazývá atraktorem. Obecně je atraktor množina bodů, ke které jsou trajektorie přitahovány v souvislosti s počtem iterací, který se blíží nekonečnu. Domény přitažlivosti pak obsahují počáteční body, které dávají vznik trajektoriím, jež se přibližují atraktoru. Na obr. 7 je vykreslen bifurkační diagram logistické rovnice. Vidíme, že pro hodnoty $A < 1$ je atraktorem $x = 0$ a pro $1 < A < 3$ je atraktorem fixní bod $x = 1 - \frac{1}{A}$. Pro A blíží se hodnotě 4 pozorujeme chaotické chování systému. [8]



Obr. 7. Bifurkační diagram logistické rovnice v prostředí Mathematica

5.4 Disipativní systémy

Disipativní systémy mají tu vlastnost, že dlouhodobé chování systému je převážně nezávislé na tom, jakým způsobem byl systém započat. Tím, jak se disipativní systém vyvíjí v čase, blíží se k určité skupině finálních bodů (může však jít i o jiné geometrické objekty). Soubor těchto bodů tvoří atraktor systému, protože jednotlivé trajektorie jsou k těmto bodům přitahovány. U disipativních systémů je dlouhodobé chování definováno vlastnostmi těchto atraktorů. [8]

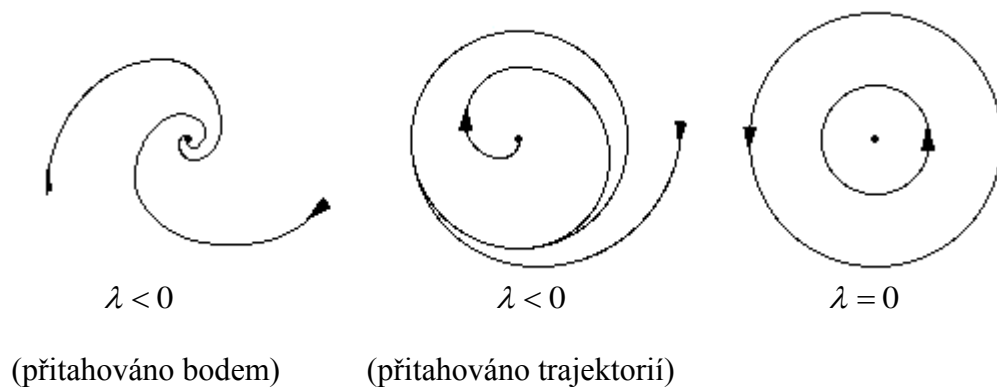
5.5 Ljapunovův exponent

Jestliže trajektorie atraktoru vykazují exponenciální divergenci blízkých trajektorií, mluvíme o tzv. chaotickém atraktoru, který je někdy také nazýván jako podivný atraktor. Ljapunovův exponent je měřítkem divergence blízkých trajektorií a označujeme jej symbolem λ . Jde tedy o kvantitativní měřítko chaotičnosti, které nám umožňuje odlišit chaotické chování od šumu a zároveň sledovat, jak systém reaguje na změnu jeho parametrů.

Ljapunovův exponent je definován jako průměr hodnoty přirozeného logaritmu absolutní hodnoty derivace funkce v jednotlivých bodech trajektorie. [8]

$$\lambda = \frac{1}{n} (\ln|f'(x_0)| + \ln|f'(x_1)| + \dots + \ln|f'(x_n)|) \quad (15)$$

Je-li λ záporné číslo, trajektorie jsou přitahovány fixním bodem nebo stabilní periodickou trajektorií. Záporný Ljapunovův exponent je typický pro disipativní systémy. Systémy se záporným λ vykazují asymptotickou stabilitu. Čím vyšší je záporná hodnota λ , tím stabilnější je systém. Pokud je Ljapunovův exponent roven nule, systém je v ustáleném režimu a vyazuje Ljapunovskou stabilitu. Kladný Ljapunovův exponent označuje systémy, jejichž trajektorie jsou nestabilní a chaotické. Obr. 8 ukazuje, jak Ljapunovův exponent ovlivňuje trajektorie. [10]



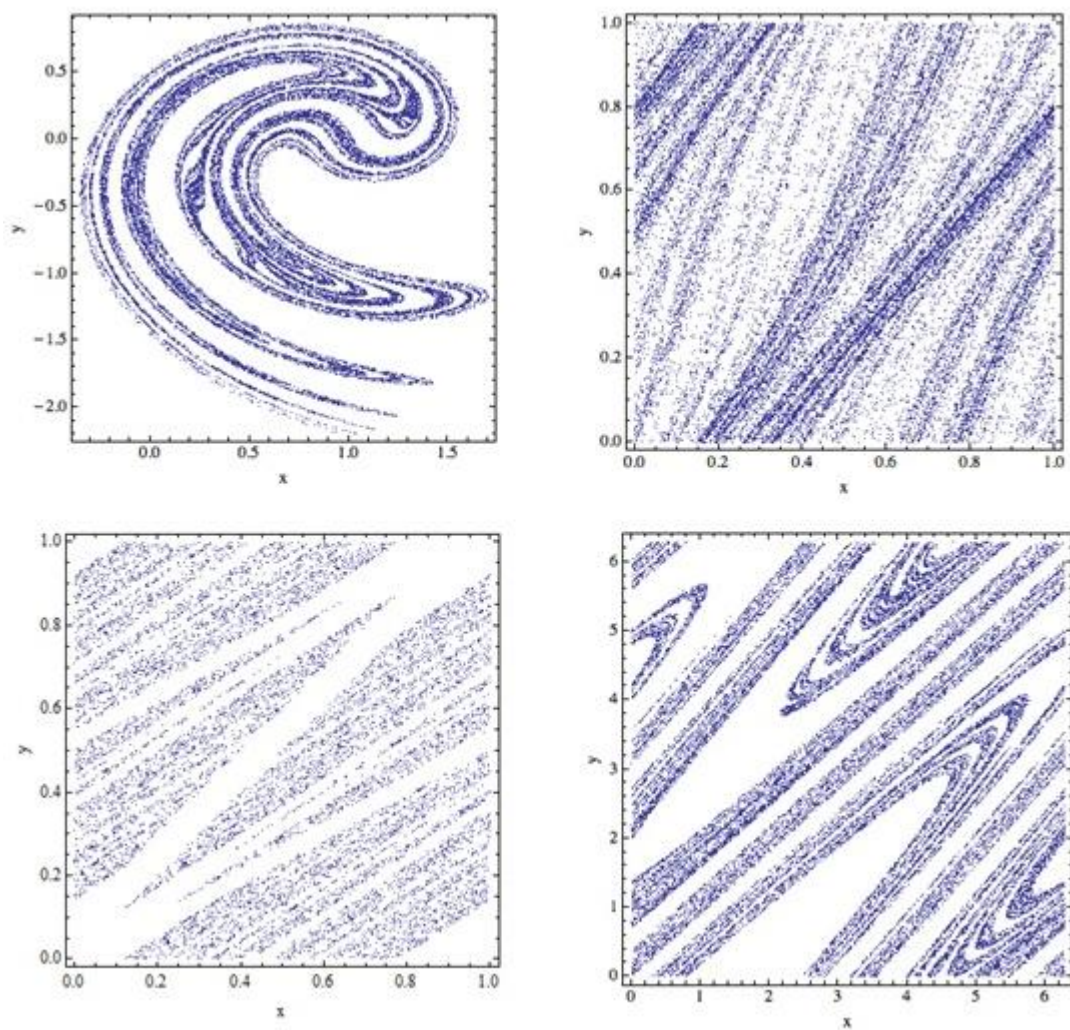
Obr. 8. Trajektorie s různými hodnotami Ljapunovova exponentu [10]

5.6 Přehled vybraných chaotických systémů

V tab. 2 jsou uvedeny definice chaotických systémů Ikeda, Sinai, Arnold Cat map a disipativního systému. V tomto případě se jedná o diskrétní systémy. Grafické znázornění těchto systémů nalezneme na obr. č. 9.

Tab. 2. Přehled vybraných chaotických systémů [11]

| Systém | Matematický zápis | Nastavení parametrů | Počáteční podmínky |
|--------------------|--|--|-------------------------------------|
| Ikeda | $X_{n+1} = \gamma + \mu(X_n \cos \phi - Y_n \sin \phi)$ $Y_{n+1} = \mu(X_n \sin \phi + Y_n \cos \phi)$ | $\alpha = 6; \beta = 0,4$ $\gamma = 1; \mu = 0,9$ | $X_0 = 0; Y_0 = 0$ |
| Sinai | $X_{n+1} = X_n + Y_n + \delta \cos 2\pi Y_n \pmod{1}$ $Y_{n+1} = X_n + 2Y_n \pmod{1}$ | $\delta = 0,1$ | $X_0 = 0,5; Y_0 = 0,5$ |
| Arnold Cat Map | $X_{n+1} = X_n + Y_n \pmod{1}$ $Y_{n+1} = X_n + kY_n \pmod{1}$ | $k = 2$ | $X_0 = 0; Y_0 = \frac{1}{\sqrt{2}}$ |
| Disipativní systém | $X_{n+1} = X_n + Y_{n+1} \pmod{2\pi}$ $Y_{n+1} = bY_n + k \sin X_n \pmod{2\pi}$ | $b = 0,1; k = 8,8$ | $X_0 = 0,1; Y_0 = 0,1$ |



Obr. 9. Chaotické systémy. Zleva doprava a shora dolů: Ikeda, Sinai, Arnold
Cat map, Disipativní systém.

6 TESTOVÁNÍ ALGORITMŮ

Testování je důležité z mnoha rozličných důvodů. Dokáže nám sdělit, jakým způsobem závisí výkon algoritmu na parametrech účelové funkce (dimenze, počet lokálních minim, šum). Testování také ukazuje, jaké kombinace řídicích parametrů jsou nejvhodnější, což je velmi užitečná informace. Na základě simulačních pokusů lze porovnat různé typy algoritmů a ve výsledku nám testování může přinést nové poznatky, které můžeme využít pro zdokonalení optimalizačního procesu. [4]

Samotné testování algoritmu se provádí tak, že mnohonásobně opakujeme hledání extrému na stejné testovací funkci a výsledky pak zobrazuje pomocí grafu nebo tabulky. [1]

6.1 Testovací funkce

Pomocí testovacích funkcí můžeme zjistit, jak je daný algoritmus úspěšný na konkrétním problému. Pro otestování algoritmu používáme soubor testovacích funkcí, které záměrně trpí různými patologiemi nebo jsou často nelineární. Například Schwefelova funkce má dvě zajímavé vlastnosti. Její minimum není situováno ve středu prohledávaného intervalu, ale u jeho okrajů. Druhou vlastností je, že poloha globálního extrému se střídavě objevuje na obou stranách. Graf dvourozměrné Schwefelovy funkce je vykreslen na obr. 10.

Zajímavou funkcí je i Rosenbrokovo sedlo, které má kromě globálního extrému ještě jeden podobný, který se svou hodnotou blíží globálnímu extrému. Podobné funkce jsou schopny úspěšně otestovat robustnost použitého evolučního algoritmu.

U mnoha testovacích funkcí známe hodnoty globálních extrémů. Porovnáním těchto hodnot s globálními extrémy nalezenými konkrétním algoritmem získáváme informaci o úspěšnosti algoritmu. Testovací funkce mohou být unimodální (s jedním extrémem) nebo multimodální (s více extrémy). Samostatnou skupinou jsou fraktální testovací funkce, u kterých předem neznáme hodnoty globálního extrému, ale lze je rovněž použít pro porovnání úspěšnosti jednotlivých algoritmů. [1]

Matematické předpisy vybraných testovacích funkcí:

$$\text{První de Jongova funkce} \quad \sum_{i=1}^D x_i^2 \quad (16)$$

$$\text{Druhá de Jongova funkce} \quad \sum_{i=1}^{D-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \quad (17)$$

$$\text{Třetí de Jongova funkce} \quad \sum_{i=1}^D |x_i| \quad (18)$$

$$\text{Čtvrtá de Jongova funkce} \quad \sum_{i=1}^D ix_i^4 \quad (19)$$

$$\text{Rastriginova funkce} \quad 10D + \sum_{i=1}^D x_i^2 - 10 \cos(2\pi x_i) \quad (20)$$

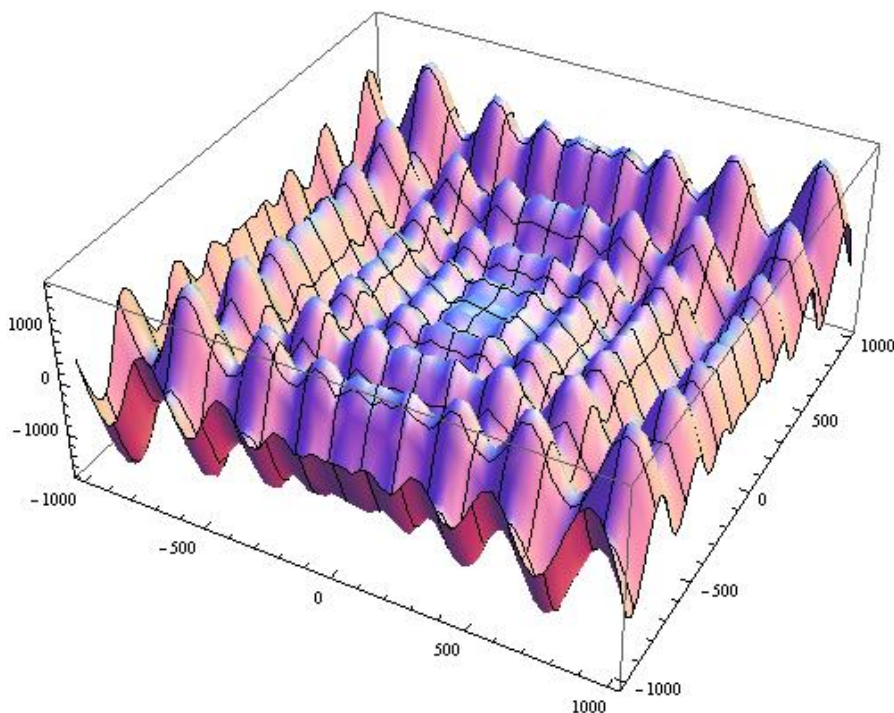
$$\text{Schwefelova funkce} \quad \sum_{i=1}^D -x_i \sin(\sqrt{|x_i|}) \quad (21)$$

$$\text{Roztažená sinusoidální funkce} \quad \sum_{i=1}^{D-1} \left(4\sqrt{(x_i^2 + x_{i+1}^2)} \sin\left(50^{0.9}\sqrt{(x_i^2 + x_{i+1}^2)}\right)^2 + 1 \right) \quad (22)$$

$$\text{Ackleyho funkce I} \quad \sum_{i=1}^{D-1} \left(\frac{1}{e^5} \sqrt{(x_i^2 + x_{i+1}^2)} + 3(\cos(2x_i) + \sin(2x_{i+1})) \right) \quad (23)$$

$$\text{Ackleyho funkce II} \quad \sum_{i=1}^{D-1} \left(20 + e - \frac{20}{e^{0.2\sqrt{\frac{(x_i^2 + x_{i+1}^2)}{2}}}} - e^{0.5(\cos(2\pi x_i) + \cos(2\pi x_{i+1}))} \right) \quad (24)$$

$$\text{Mastersova funkce} \quad \sum_{i=1}^{D-1} \left(e^{\frac{-(x_i^2 + x_{i+1}^2) + 0.5x_i x_{i+1}}{8}} \cos(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}) \right) \quad (25)$$



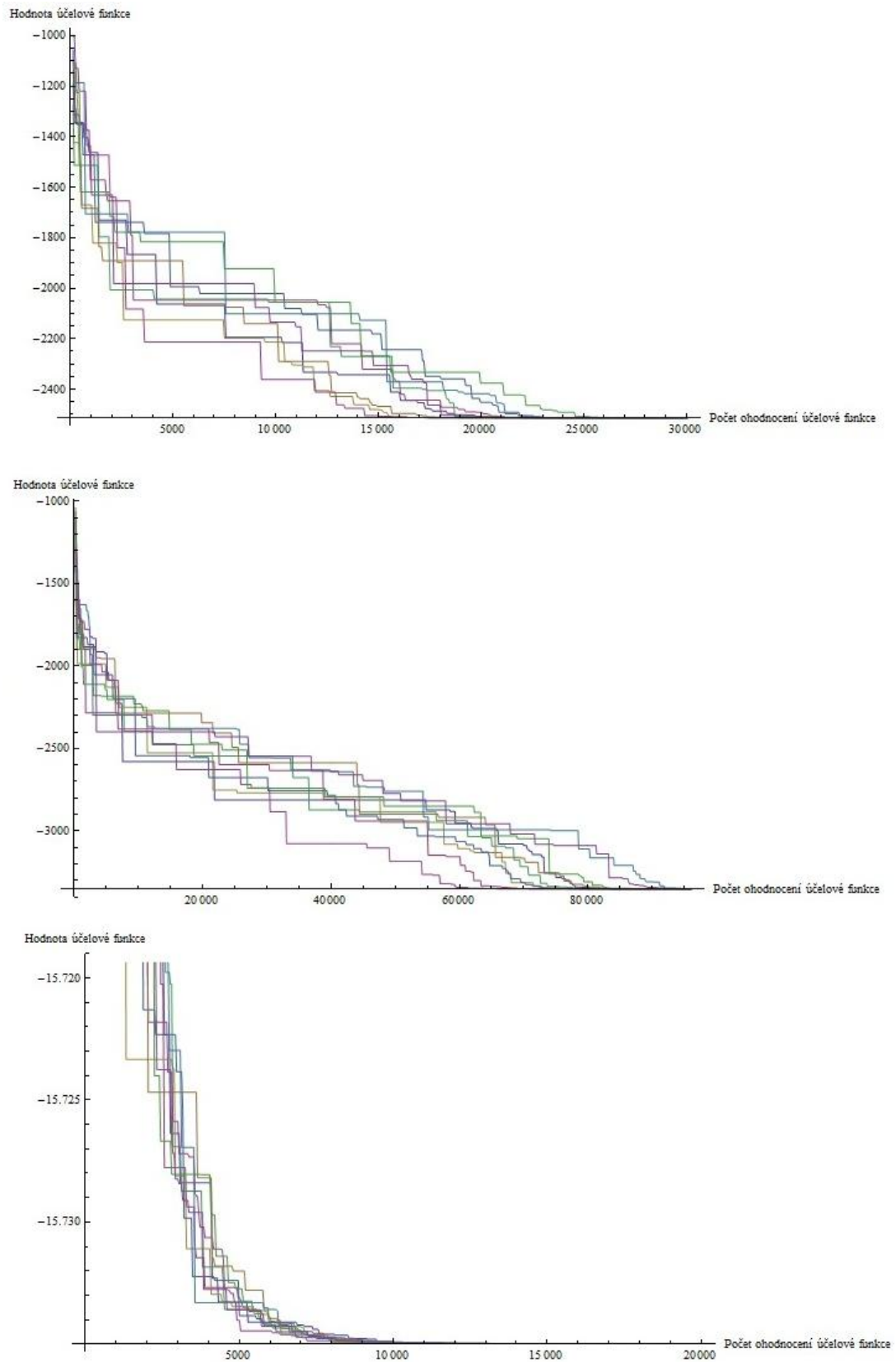
Obr. 10. Schwefelova funkce

6.2 Vyhodnocování

Kvalita průběhu evolučního procesu se zobrazuje pomocí historie vývoje hodnoty účelové funkce zanesené do grafu. Graf může být vytvořen různými způsoby. Jednou z možností je zobrazení nejhorších a nejlepších jedinců ze všech populací, což nám dává informaci o celkové konvergenci populace. Jako lepší řešení se však ukázalo zobrazení závislosti hodnoty účelové funkce na počtu jejího ohodnocení. Je to dáno tím, že každý algoritmus provádí během jednoho cyklu různý počet ohodnocení účelové funkce. Nedochází tak ke zkreslení informace o skutečné kvalitě evolučního procesu a můžeme si dovolit porovnávat algoritmy lišící se vnitřní strukturou. Příklad takovýchto grafů je uveden na obr. 11.

Další možností vyhodnocení je zobrazení pouze nejlepších jedinců ze všech realizovaných simulací formou histogramu. Při použití této formy prezentace výsledků je dobře viditelné, jaký nejlepší výsledek našly jednotlivé simulace.

Pro hodnocení kvality algoritmů je také velmi důležitý čas výpočtu, který se odvíjí od počtu provedených elementárních operací. Tento čas bývá ovlivňován složitostí účelové funkce a konkrétním nastavením jednotlivých parametrů algoritmu. S výhodou se zde používá paralelizace algoritmu, která snižuje celkový čas výpočtu. [1]



Obr. 11. Průběhy evolučního procesu při mnohonásobném opakování

7 SOFTWARE MATHEMATICA

Mathematica je považována za nejvýkonnější systém, který obecně slouží k vykonávání výpočtů na počítači. Od svého vzniku v roce 1988 měla velký vliv na to, jakým způsobem byly počítače zapojeny do technických oblastí. Již od šedesátých let existovaly nejrůznější výpočetní systémy, které se však zaměřovaly pouze na určitou problematiku. Mathematica přišla s konceptem, který měl za cíl integrovat všechny potřebné funkce do jednoho systému. Původně sloužila Mathematica primárně pro matematické výpočty. S postupem času se však její možnosti rozšířily natolik, že ji dnes používá mnoho vědců z nejrůznějších odvětví a v mnoha případech hrála roli při důležitých objevech. V inženýrství se stala Mathematica standardem a její bohaté rozšíření nalezneme i ve školství. [13]

Mathematica je interpretovaný a velmi bohatý jazyk, který umožňuje soustředit se přímo na řešený problém díky vysokému počtu vestavěných funkcí. Kód se obecně vyznačuje menším rozsahem, ale vyšší mírou abstrakce. Mathematica také obsahuje široké možnosti vizualizace výsledků. Výhodou je také kompatibilita skrze různé platformy a zpětná kompatibilita mezi jednotlivými verzemi. [14]

II. PRAKTICKÁ ČÁST

8 KLASICKÁ DIFERENCIÁLNÍ EVOLUCE

Algoritmus klasické diferenciální evoluce vytvořený v prostředí Mathematica verze 8.0 jsem nejprve aplikoval na základní testovací funkce, abych ověřil jeho správnost pro další výpočty. Zvolil jsem strategii DE/Best/2/bin a jako testovací funkce jsem vybral všechny 4 de Jongovy funkce, Rastriginovu funkci a Schwefelovu funkci. Algoritmus byl spuštěn s parametry uvedenými v tab. 3. Získané hodnoty se nachází v tab. 4 a 5.

Tab. 3. Parametry simulací

| Parametr | Hodnota |
|-------------------|----------------------|
| Počet generací | 100 |
| Počet opakování | 50 |
| CR | 0,8 |
| F | 0,6 |
| Specimen | {{Real, {-500,500}}} |
| Velikost populace | 10 |
| Strategie | DE/best/2/bin |
| Počet dimenzí | 2 |

Tab. 4. Nejlepší a nejhorší výsledky

| Testovací funkce | Známa hodnota globálního extrému | Nejlepší nalezená hodnota globálního extrému | Nejhorší nalezená hodnota globálního extrému |
|--------------------------|----------------------------------|--|--|
| První de Jongova funkce | 0 | $7,9328 \cdot 10^{-17}$ | $1,4426 \cdot 10^{-11}$ |
| Druhá de Jongova funkce | 0 | $3,0741 \cdot 10^{-15}$ | $4,2919 \cdot 10^{-8}$ |
| Třetí de Jongova funkce | 0 | $2,6298 \cdot 10^{-16}$ | $1,8333 \cdot 10^{-11}$ |
| Čtvrtá de Jongova funkce | 0 | $3,0083 \cdot 10^{-33}$ | $4,3834 \cdot 10^{-24}$ |
| Rastriginova funkce | 0 | $1,8474 \cdot 10^{-12}$ | 1,98992 |
| Schwefelova funkce | -837,966 | -837,966 | -719,527 |

Tab. 5. Průměrná odchylka od známé hodnoty globálního extrému

| Testovací funkce | Známa hodnota globálního extrému | Průměrná odchylka od známé hodnoty globálního extrému |
|--------------------------|----------------------------------|---|
| První de Jongova funkce | 0 | $8,394 \cdot 10^{-13}$ |
| Druhá de Jongova funkce | 0 | $1,1755 \cdot 10^{-9}$ |
| Třetí de Jongova funkce | 0 | $8,9067 \cdot 10^{-13}$ |
| Čtvrtá de Jongova funkce | 0 | $1,9527 \cdot 10^{-25}$ |
| Rastriginova funkce | 0 | 0,398018 |
| Schwefelova funkce | -837,966 | 18,9504 |

U funkcí, které mají globální minimum rovno nule, se algoritmus dostával do velmi nízkých hodnot, ale nikdy nebyl schopen nalézt zcela přesnou hodnotu. Jelikož se však průměrná odchylka od nuly pohybovala v řádu 10^{-13} , lze nalezená řešení považovat za uspokojivá.

U Schwefelovy funkce se podařilo nalézt přesnou hodnotu globálního minima ve 42 případech z celkových 50 realizací. Průměrná odchylka od známé hodnoty globálního minima zde však byla mnohonásobně vyšší než u předchozích funkcí. I při dalších pokusech se ukázalo, že algoritmus v naprosté většině případů nalezne hodnotu globálního minima Schwefelovy funkce a odchylku pak tvoří pouze několik méně úspěšných opakování. Počet případů, kdy se nalezená hodnota odchyluje od globálního minima, lze úspěšně snižovat navyšováním počtu realizovaných generací během evolučního procesu. Na základě těchto poznatků bylo možné vytvořený algoritmus využít pro další výpočty.

9 NASTAVENÍ PARAMETRŮ

Pro klasickou diferenciální evoluci bylo provedeno mnoho pokusů s různými hodnotami mutační konstanty F a prahu křížení CR . V potaz byly brány všechny kombinace obou parametrů v rozsahu 0,1 - 0,9 s krokem 0,1. Ostatní parametry odpovídají tab. 3. Měřena byla průměrná odchylka od globálního minima Schwefelovy funkce o dvou dimenzích, které je rovno -837,966. Získané výsledky jsou uvedeny v tab. 6.

Tab. 6. Matice výsledků klasické diferenciální evoluce pro parametry F a CR

| CR \ F | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 |
|--------|--------|--------|--------|--------|--------|--------|---------|----------|----------|
| 0,1 | 117,92 | 92,046 | 60,449 | 57,283 | 26,091 | 26,893 | 32,446 | 29,890 | 41,4752 |
| 0,2 | 131,57 | 97,136 | 60,798 | 45,539 | 29,507 | 21,218 | 7,4543 | 3,3243 | 3,87099 |
| 0,3 | 129,87 | 119,62 | 58,824 | 43,427 | 18,950 | 9,4916 | 4,7404 | 7,1374 | 0,02582 |
| 0,4 | 137,49 | 107,38 | 80,539 | 28,030 | 35,531 | 16,581 | 0,00026 | 2,3692 | 0,00273 |
| 0,5 | 159,62 | 96,331 | 64,747 | 32,768 | 21,319 | 23,687 | 18,555 | 2,369 | 0,00046 |
| 0,6 | 178,09 | 151,16 | 101,46 | 36,716 | 16,581 | 14,212 | 0,00022 | 2,3689 | 2,369 |
| 0,7 | 211,30 | 181,49 | 119,25 | 58,035 | 21,319 | 4,7377 | 7,1065 | 0,000226 | 0,000249 |
| 0,8 | 187,88 | 155,43 | 128,31 | 56,455 | 30,399 | 16,186 | 7,1065 | 0,000226 | 0,000242 |
| 0,9 | 228,45 | 200,93 | 105,10 | 62,782 | 18,555 | 14,212 | 2,3689 | 0,000227 | 7,10657 |

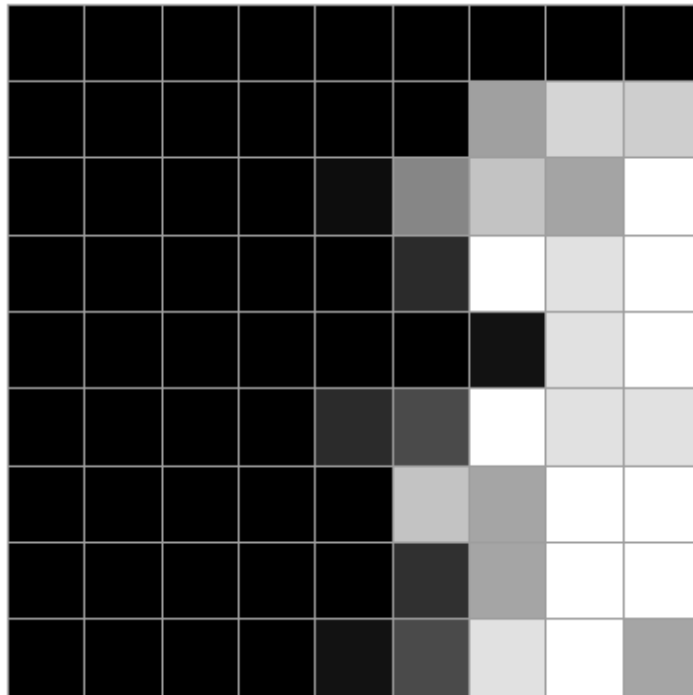
Pro chaotickou diferenciální evoluci se systémem Sinai byla také vytvořena matice hodnot odchylky od globálního minima Schwefelovy funkce pro různé kombinace parametrů F a CR . Průměrné hodnoty odchylek jsou zapsány v tab. 7.

Tab. 7. Matice výsledků chaotické diferenciální evoluce pro parametry F a CR

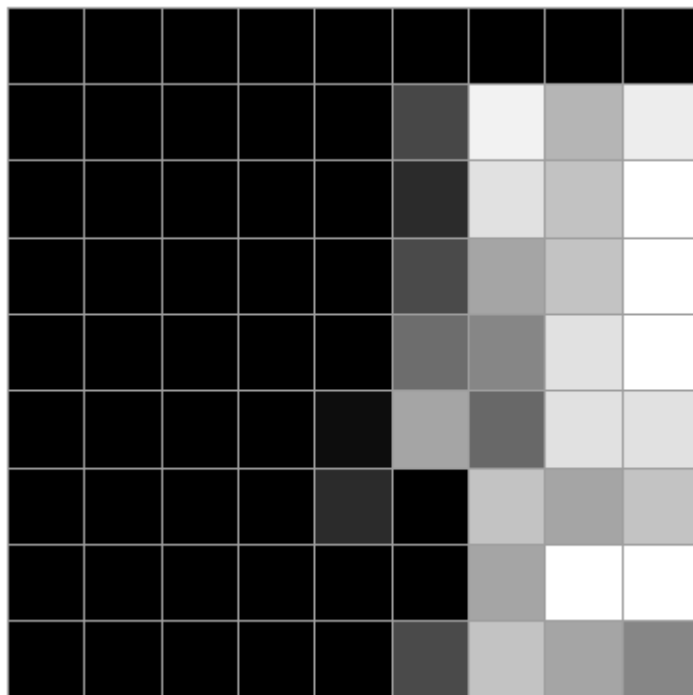
| CR \ F | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 |
|--------|--------|--------|--------|--------|--------|--------|---------|----------|----------|
| 0,1 | 129,97 | 92,565 | 58,129 | 54,613 | 52,792 | 40,526 | 42,269 | 36,826 | 29,3830 |
| 0,2 | 131,58 | 110,93 | 53,302 | 40,274 | 26,260 | 14,402 | 1,0330 | 5,8357 | 1,47300 |
| 0,3 | 162,32 | 128,70 | 41,848 | 40,269 | 30,794 | 16,584 | 2,37923 | 4,77176 | 0,04368 |
| 0,4 | 156,94 | 146,83 | 71,063 | 42,243 | 21,319 | 14,212 | 7,10654 | 4,74100 | 0,00064 |
| 0,5 | 162,17 | 127,57 | 75,011 | 39,874 | 21,319 | 11,410 | 9,47529 | 2,36901 | 0,000367 |
| 0,6 | 163,34 | 138,23 | 81,723 | 32,768 | 18,950 | 7,1065 | 11,8441 | 2,36899 | 2,36900 |
| 0,7 | 162,73 | 108,34 | 94,772 | 58,429 | 16,581 | 23,687 | 4,73776 | 7,10653 | 4,73776 |
| 0,8 | 202,91 | 186,90 | 89,229 | 75,406 | 26,056 | 23,687 | 7,10653 | 0,000226 | 0,000229 |
| 0,9 | 226,55 | 172,84 | 118,54 | 50,958 | 32,768 | 14,212 | 4,73776 | 7,10653 | 9,47548 |

Klasická i chaotická diferenciální evoluce byla použita se strategií DE/Best2/Bin. Převedením obou výše uvedených tabulek do grafické reprezentace dostáváme obr. 12 a 13.

Čím světlejší je buňka, tím menší je její průměrná odchylka. Buňky s průměrnou odchylkou > 20 jsou zakresleny černou barvou. Oba výstupy jsou si velice podobné, a proto byly pro veškeré následující pokusy zvoleny hodnoty parametrů $CR = 0,8$ a $F = 0,8$. Tyto hodnoty také korespondují s doporučeným nastavením parametrů dle literatury.



Obr. 12. Grafická reprezentace odchylek pro klasickou diferenciální evoluci



Obr. 13. Grafická reprezentace odchylek pro chaotickou diferenciální evoluci

10 TESTOVÁNÍ CHAOTICKÉ DIFERENCIÁLNÍ EVOLUCE

Pro chaotickou diferenciální evoluci byly při generování zkušebního vektoru použity různé chaotické systémy, které nahradily náhodný generátor, který je využíván v klasické diferenciální evoluci. Existuje několik strategií diferenciální evoluce a u každé strategie se můžeme rozhodnout, jaký chaotický systém v algoritmu použijeme. Celkové množství různých variant je tedy obrovské. Na následujících stránkách bude několik variant chaotické diferenciální evoluce porovnáváno s klasickou diferenciální evolucí na vybraných testovacích funkcích. Algoritmy byly aplikovány na testovací funkce, u nichž známe přesnou hodnotu globálního extrému.

10.1 Schwefelova funkce

Jako první byla vybrána Schwefelova funkce. Testována byla klasická diferenciální evoluce (KDE) a 4 chaotické diferenciální evoluce (CDE). Konkrétně šlo o chaotickou diferenciální evoluci se systémy Ikeda, Arnold Cat map, Sinai a disipativním systémem. Celkem byly provedeny 3 simulace, jejichž parametry jsou uvedeny v tab. 8.

Tab. 8. Parametry simulací pro Schwefelovu funkci

| Parametr | Hodnoty pro 1. simulaci | Hodnoty pro 2. simulaci | Hodnoty pro 3. simulaci |
|-------------------|-------------------------|-------------------------|-------------------------|
| Počet generací | 200 | 200 | neomezeně |
| Počet opakování | 20 | 20 | 1 |
| CR | 0,8 | 0,8 | 0,8 |
| F | 0,8 | 0,8 | 0,8 |
| Specimen | {{Real, {-500 ; 500}}} | {{Real, {-500 ; 500}}} | {{Real, {-500 ; 500}}} |
| Velikost populace | 10D | 10D | 10D |
| Strategie | DE/best/2/bin | DE/rand/2/bin | DE/best/2/bin |

U prvních dvou simulací byla účelová funkce optimalizována v několika variantách, které se od sebe lišily dimenzí účelové funkce (značeno jako D), čili počtem optimalizovaných argumentů. Výsledky těchto simulací jsou uvedeny v tab. 9 a 10.

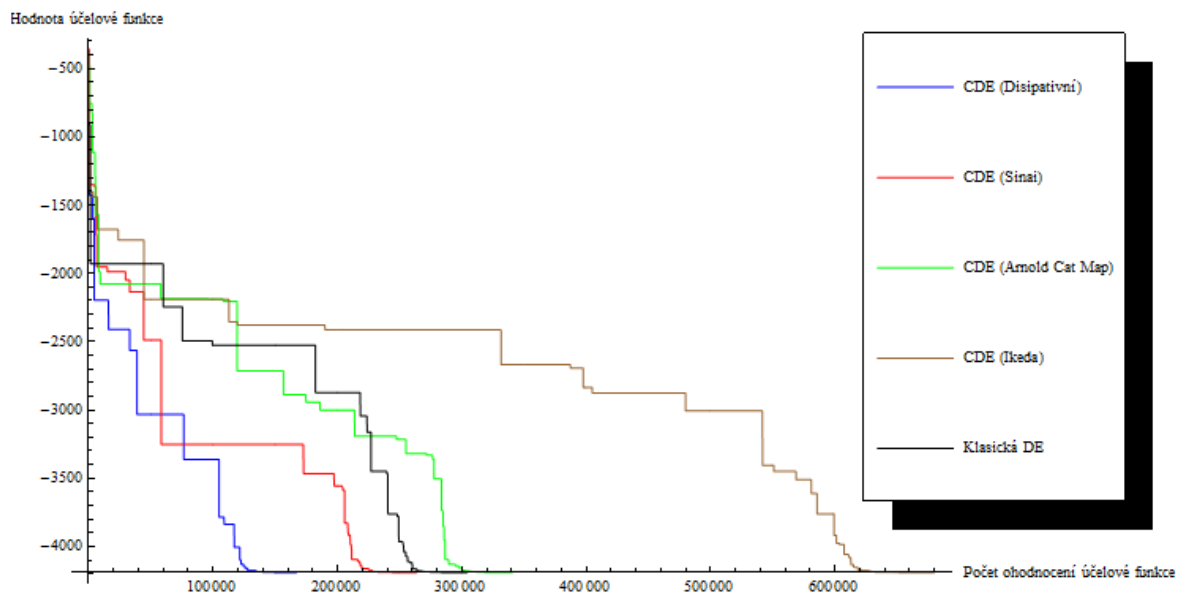
Tab. 9. Simulace 1: Odchylky od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| 2 | -837,966 | $2,2545 \cdot 10^{-4}$ | $2,2545 \cdot 10^{-4}$ | $2,2545 \cdot 10^{-4}$ | 5,92214 | $2,2545 \cdot 10^{-4}$ |
| 4 | -1675,932 | $7,1499 \cdot 10^{-4}$ | 6,05681 | $3,4091 \cdot 10^{-3}$ | $2,0311 \cdot 10^{-3}$ | $5,0084 \cdot 10^{-4}$ |
| 6 | -2513,898 | 228,818 | 491,997 | 315,082 | 306,371 | 158,723 |
| 8 | -3351,864 | 800,731 | 1024,35 | 872,714 | 822,857 | 805,248 |

Tab. 10. Simulace 2: Odchylky od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| 2 | -837,966 | $2,2545 \cdot 10^{-4}$ | $2,2545 \cdot 10^{-4}$ | $2,2545 \cdot 10^{-4}$ | $2,2545 \cdot 10^{-4}$ | $2,2545 \cdot 10^{-4}$ |
| 4 | -1675,932 | 9,83358 | 117,67 | 10,58 | 7,92193 | 0,97831 |
| 6 | -2513,898 | 370,338 | 591,326 | 399,529 | 466,854 | 338,056 |
| 8 | -3351,864 | 928,976 | 1128,390 | 917,182 | 963,050 | 880,255 |

V obou případech dosahoval nejlepších výsledků algoritmus chaotické diferenciální evoluce s disipativním systémem. Naopak nejhorší výsledky podával algoritmus chaotické diferenciální evoluce se systémem Ikeda. Při rostoucím počtu rozměrů účelové funkce se enormně zvětšuje měřená odchylka, takže při více jak 4 dimenzích dostáváme velmi nepřesné výsledky. Z tohoto důvodu byla provedena třetí simulace, kde byl evoluční proces spuštěn tak, aby byl ukončen až po dosažení hodnoty globálního maxima pro 10-ti rozměrnou Schwefelovu funkci. Proces nebyl vícenásobně opakován a ostatní parametry zůstaly zachovány. Výsledky této simulace jsou zobrazeny na obr. č. 14, kde je viditelná závislost hodnoty účelové funkce na počtu ohodnocení účelové funkce.



Obr. 14. Průběh evolučního procesu pro Schwefelovu funkci ($D=10$)

Z průběhů je viditelné, že disipativní systém a systém Sinai jsou schopny nalézt globální minimum rychleji než klasická diferenciální evoluce. Naopak Arnold Cat map a zejména systém Ikeda potřebovaly k nalezení extrému větší počet ohodnocení účelové funkce. Zhodnocení této simulace je provedeno v tab. 11, kde je uveden i celkový čas potřebný k nalezení globálního minima.

Tab. 11. Výsledky 3. simulace pro Schwefelovu funkci (10 D)

| Algoritmus | Počet ohodnocení účelové funkce | Doba trvání výpočtu [s] |
|----------------------|---------------------------------|-------------------------|
| KDE | 304400 | 162,63 |
| CDE (Ikeda) | 680000 | 501,22 |
| CDE (Arnold Cat map) | 340800 | 249,58 |
| CDE (Sinai) | 263700 | 193,77 |
| CDE (disipativní) | 167100 | 124,62 |

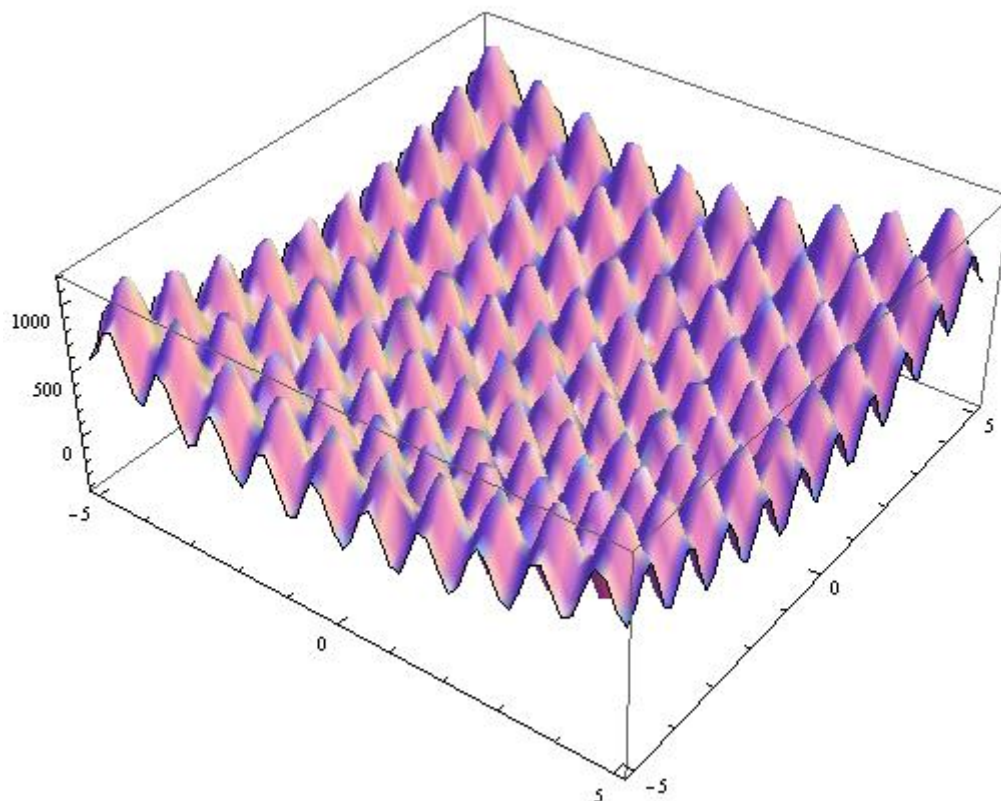
Doba výpočtu se pohybovala řádově v minutách. Od ostatních algoritmů se výrazně odklání systém Ikeda, který pro nalezení extrému potřeboval 501,22 sekund, což je v porovnání s ostatními výrazné navýšení doby výpočtu.

10.2 Rastriginova funkce

Testování na Rastriginově funkci bylo prováděno s parametry uvedenými v tab. 12. Vizualizace dvourozměrné Rastriginovy funkce je na obr. 15.

Tab. 12. Parametry simulací pro Rastriginovu funkci

| Parametr | Hodnoty pro 1. simulaci | Hodnoty pro 2. simulaci | Hodnoty pro 3. simulaci |
|-------------------|--------------------------|--------------------------|--------------------------|
| Počet generací | 200 | 200 | neomezeně |
| Počet opakování | 20 | 20 | 1 |
| CR | 0,8 | 0,8 | 0,8 |
| F | 0,8 | 0,8 | 0,8 |
| Specimen | {{Real, {-5,12 ; 5,12}}} | {{Real, {-5,12 ; 5,12}}} | {{Real, {-5,12 ; 5,12}}} |
| Velikost populace | 10D | 10D | 10D |
| Strategie | DE/best/2/bin | DE/rand/2/bin | DE/best/2/bin |



Obr. 15. Rastriginova funkce

Výsledky simulací měřící průměrné odchylky od globálního minima jsou zapsány v tab. 13 a 14.

Tab. 13. Simulace 1: Odchylky od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|---------|-------------|------------------|-------------|-------------------|
| 2 | 0 | 0 | 0,24874 | 0 | 0 | 0 |
| 4 | 0 | 1,31435 | 3,8895 | 1,81004 | 1,83477 | 0,66387 |
| 6 | 0 | 10,8232 | 16,0798 | 11,9057 | 11,7926 | 10,4494 |
| 8 | 0 | 24,4106 | 32,7473 | 23,6621 | 24,578 | 21,6622 |

Tab. 14. Simulace 2: Odchylky od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota a minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|------------------------|-------------------------|------------------------|-------------------------|-------------------------|-------------------------|
| 2 | 0 | $3,5953 \cdot 10^{-12}$ | $1,8017 \cdot 10^{-4}$ | $4,1566 \cdot 10^{-14}$ | $2,2382 \cdot 10^{-14}$ | $1,3725 \cdot 10^{-12}$ |
| 4 | 0 | 3,0503 | 4,5527 | 3,2291 | 2,8915 | 1,8577 |
| 6 | 0 | 12,0527 | 15,0474 | 14,203 | 13,8104 | 10,7813 |
| 8 | 0 | 26,1863 | 34,1293 | 27,7919 | 28,4417 | 22,7742 |

Všechny algoritmy dosahovaly poměrně vysokých odchylek. Výjimku tvoří pouze simulace pro $D = 2$. Nejnižší průměrnou odchylku dosahovala chaotická diferenciální evoluce s disipativním systémem následovaná klasickou diferenciální evolucí. Výsledky 3. simulace jsou zahrnuty v tab. 15.

Tab. 15. Výsledky 3. simulace pro Rastriginovu funkci (10 D)

| Algoritmus | Počet ohodnocení účelové funkce | Doba trvání výpočtu [s] |
|----------------------|---------------------------------|-------------------------|
| KDE | 3097800 | 1686,921 |
| CDE (Ikeda) | - | - |
| CDE (Arnold Cat map) | 5789600 | 4360,638 |
| CDE (Sinai) | 6414600 | 4739,920 |
| CDE (disipativní) | 1922100 | 1523,790 |

Doba potřebná k nalezení hodnoty globálního minima se u nejúspěšnějších algoritmů pohybovala v desítkách minut. Chaotická diferenciální evoluce se systémy Sinai a Arnold Cat map potřebovala k dosažení výsledku více jak hodinu. Chaotická diferenciální evoluce se systémem Ikeda nenalezla globální extrém ani po uplynutí 5 hodin. Z tohoto důvodu nejsou hodnoty pro tento algoritmus v tabulce zahrnuty.

10.3 Ackleyho funkce I

Stejný postup byl aplikován i v případě Ackleyho funkce I. Pro všech 5 algoritmů byly provedeny 3 simulace. Jejich parametry jsou uvedeny v tab. 16.

Tab. 16. Parametry simulací pro Ackleyho funkci I

| Parametr | Hodnoty pro 1. simulaci | Hodnoty pro 2. simulaci | Hodnoty pro 3. simulaci |
|-------------------|-------------------------|-------------------------|-------------------------|
| Počet generací | 200 | 200 | neomezeně |
| Počet opakování | 20 | 20 | 1 |
| CR | 0,8 | 0,8 | 0,8 |
| F | 0,8 | 0,8 | 0,8 |
| Specimen | {{Real, {-30 ; 30}}} | {{Real, {-30 ; 30}}} | {{Real, {-30 ; 30}}} |
| Velikost populace | 10D | 10D | 10D |
| Strategie | DE/best/2/bin | DE/rand/2/bin | DE/best/2/bin |

Jednotlivé výsledky pro 1. a 2. simulaci jsou uvedeny v tab. 17 a 18.

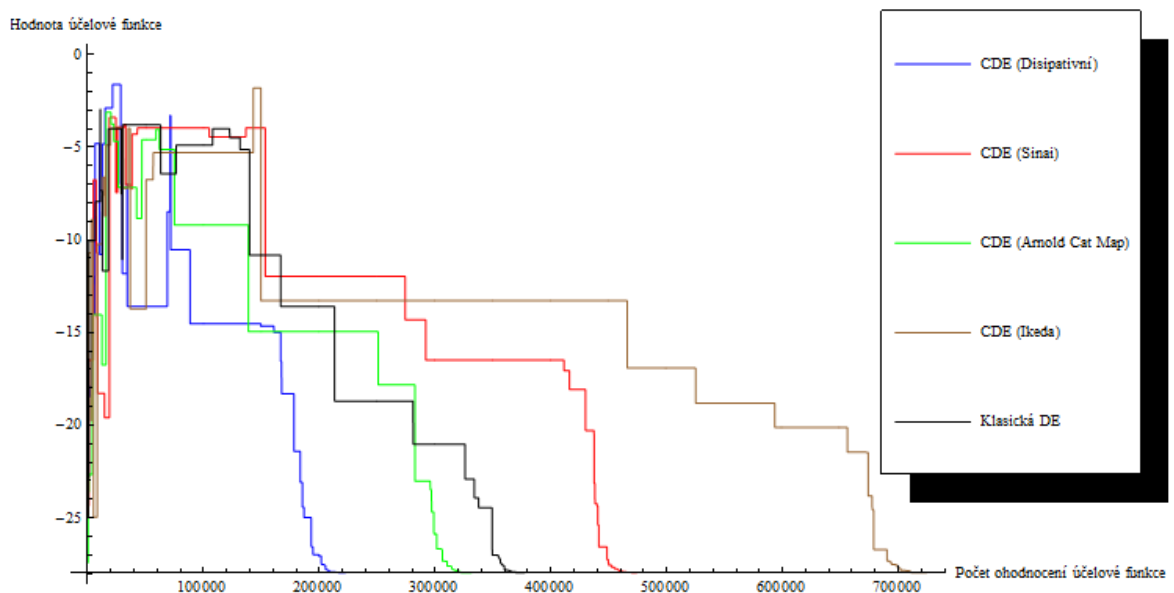
Tab. 17. Simulace 1: Odchyly od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| 2 | -4,5901 | $1,6341 \cdot 10^{-6}$ | $1,6341 \cdot 10^{-6}$ | $1,6341 \cdot 10^{-6}$ | $1,6341 \cdot 10^{-6}$ | $1,6341 \cdot 10^{-6}$ |
| 4 | -10,46137 | $2,39 \cdot 10^{-3}$ | 0,22039 | 0,01275 | 0,07054 | $2,32 \cdot 10^{-3}$ |
| 6 | -16,29877 | 4,3451 | 7,86864 | 4,68185 | 4,5361 | 3,31813 |
| 8 | -22,13611 | 14,1343 | 20,6386 | 14,5216 | 14,6120 | 11,7813 |

Tab. 18. Simulace 2: Odchyly od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|------------------------|-----------------------|-----------------------|-----------------------|------------------------|
| 2 | -4,5901 | $1,6341 \cdot 10^{-6}$ | $1,633 \cdot 10^{-6}$ | $1,634 \cdot 10^{-6}$ | $1,634 \cdot 10^{-6}$ | $1,6339 \cdot 10^{-6}$ |
| 4 | -10,46137 | 0,28525 | 1,38969 | 0,50405 | 0,31834 | 0,07042 |
| 6 | -16,29877 | 6,38524 | 10,7668 | 6,9545 | 6,47749 | 5,22594 |
| 8 | -22,13611 | 19,5879 | 25,7695 | 19,6978 | 18,9648 | 16,808 |

Nejllepších výsledků opět dosahoval disipativní systém následovaný systémem Arnold Cat map. Získané výsledky však byly uspokojivé jen pro $D = 2$ a $D = 4$ a zejména u strategie DE/best/2/bin, jelikož DE/rand/2/bin podávala viditelně horší výsledky pro klasickou i chaotickou diferenciální evoluci. Při 3. simulaci byl evoluční proces zastaven až v momentě, kdy byla dosažena hodnota globálního minima. Počet dimenzí účelové funkce byl navýšen na 10. Průběh této simulace byl zakreslen do obr. č. 16.

Obr. 16. Průběh evolučního procesu pro Ackleyho funkci I ($D=10$)

Nejúspěšnějším algoritmem byla chaotická diferenciální evoluce s disipativním systémem, která se v porovnání s ostatními dokázala velmi rychle dopracovat k hodnotě globálního extrému. Zhodnocení simulace je uvedeno v tab. 19.

Tab. 19. Výsledky 3. simulace pro Ackleyho funkci I (10 D)

| Algoritmus | Počet ohodnocení účelové funkce | Doba trvání výpočtu [s] |
|----------------------|------------------------------------|-------------------------|
| KDE | 377400 | 210,188 |
| CDE (Ikeda) | 724400 | 527,990 |
| CDE (Arnold Cat map) | 331300 | 237,598 |
| CDE (Sinai) | 474100 | 371,515 |
| CDE (disipativní) | 222500 | 172,520 |

Kromě disipativního systému byl poměrně úspěšný i systém Arnold Cat map, který k nalezení globálního minima potřeboval o 40 000 ohodnocení účelové funkce méně než klasická diferenciální evoluce. Doba výpočtu mu však trvala o více jak 27 sekund déle.

10.4 Ackleyho funkce II

Pro další simulaci byla použita Ackleyho funkce II. Parametry jednotlivých simulací jsou uvedeny v tab. 20.

Tab. 20. Parametry simulací pro Ackleyho funkci II

| Parametr | Hodnoty pro 1. simulaci | Hodnoty pro 2. simulaci | Hodnoty pro 3. simulaci |
|-------------------|----------------------------|----------------------------|----------------------------|
| Počet generací | 200 | 200 | neomezeně |
| Počet opakování | 20 | 20 | 1 |
| CR | 0,8 | 0,8 | 0,8 |
| F | 0,8 | 0,8 | 0,8 |
| Specimen | {{Real, {-2 ; 2}}} | {{Real, {-2 ; 2}}} | {{Real, {-2 ; 2}}} |
| Velikost populace | 10D | 10D | 10D |
| Strategie | DE/best/2/bin | DE/rand/2/bin | DE/best/2/bin |

Průměrné odchylky od globálního minima pro varianty DE/best/2/bin a DE/rand/2/bin jsou uvedeny v tab. 21 a 22.

Tab. 21. Simulace 1: Odchylky od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| 2 | 0 | $1,8527 \cdot 10^{-13}$ | $1,0356 \cdot 10^{-13}$ | $1,1954 \cdot 10^{-13}$ | $3,7836 \cdot 10^{-14}$ | $2,5419 \cdot 10^{-13}$ |
| 4 | 0 | $1,6623 \cdot 10^{-5}$ | $7,3157 \cdot 10^{-5}$ | $1,6262 \cdot 10^{-5}$ | $1,7792 \cdot 10^{-5}$ | $1,219 \cdot 10^{-5}$ |
| 6 | 0 | 0,01606 | 0,0715 | 0,02064 | 0,01578 | 0,01248 |
| 8 | 0 | 0,7009 | 3,15815 | 0,82246 | 0,74808 | 0,37949 |

Tab. 22. Simulace 2: Odchylky od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| 2 | 0 | $2,4916 \cdot 10^{-11}$ | $2,1114 \cdot 10^{-11}$ | $1,9086 \cdot 10^{-11}$ | $1,2926 \cdot 10^{-11}$ | $1,8121 \cdot 10^{-11}$ |
| 4 | 0 | $4,5210 \cdot 10^{-4}$ | $2,4536 \cdot 10^{-3}$ | $5,5712 \cdot 10^{-4}$ | $4,4026 \cdot 10^{-4}$ | $2,6139 \cdot 10^{-4}$ |
| 6 | 0 | 0,14067 | 0,91120 | 0,18789 | 0,18753 | 0,09366 |
| 8 | 0 | 3,56406 | 7,36 | 4,06312 | 3,88325 | 2,05312 |

Zde se ukázala strategie DE/rand/2/bin jako mnohonásobně horší než DE/best/2/bin. Nejlepší průměrnou odchylku při použití dvourozměrné funkce dosáhl systém Sinai. S rostoucím počtem dimenzí však byl překonán disipativním systémem. Nejhorších výsledků opět dosáhla chaotická diferenciální evoluce se systémem Ikeda. Na základě zvoleného rozsahu vzorového jedince byla tato simulace kratší než předchozí. Výsledky 3. simulace jsou k dispozici v tab. 23.

Tab. 23. Výsledky 3. simulace pro Ackleyho funkci II (10 D)

| Algoritmus | Počet ohodnocení účelové funkce | Doba trvání výpočtu [s] |
|----------------------|---------------------------------|-------------------------|
| KDE | 226000 | 145,927 |
| CDE (Ikeda) | 376700 | 308,640 |
| CDE (Arnold Cat map) | 244300 | 200,446 |
| CDE (Sinai) | 242100 | 200,357 |
| CDE (disipativní) | 205700 | 174,808 |

Z hlediska doby výpočtu byla nejúspěšnější klasická diferenciální evoluce, ale nejmenší počet ohodnocení účelové funkce potřebovala k dosažení extrému chaotická diferenciální evoluce s disipativním systémem.

10.5 Roztažená sinusoidální V funkce

Roztažená sinusoidální V funkce je další ze skupiny funkcí, jež mají hodnotu globálního minima rovnu 0. Parametry všech simulací jsou zahrnuty v tab. 24.

Tab. 24. Parametry simulací pro roztaženou sinusoidální V funkci

| Parametr | Hodnoty pro 1. simulaci | Hodnoty pro 2. simulaci | Hodnoty pro 3. simulaci |
|-------------------|-------------------------|-------------------------|-------------------------|
| Počet generací | 200 | 200 | neomezeně |
| Počet opakování | 20 | 20 | 1 |
| CR | 0,8 | 0,8 | 0,8 |
| F | 0,8 | 0,8 | 0,8 |
| Specimen | {{Real, {0 ; 10}}} | {{Real, {0 ; 10}}} | {{Real, {0 ; 10}}} |
| Velikost populace | 10D | 10D | 10D |
| Strategie | DE/best/2/bin | DE/rand/2/bin | DE/best/2/bin |

Hodnoty průměrných odchylek od globálního extrému z prvních dvou simulací jsou uvedeny v tab. 25 a 26.

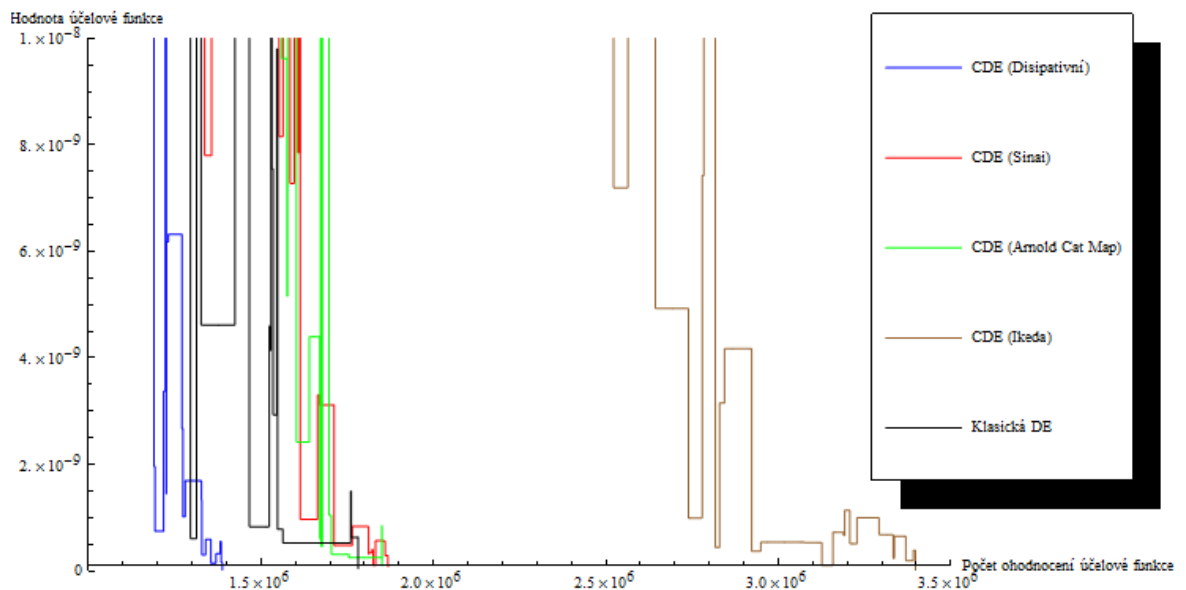
Tab. 25. Simulace 1: Odchyly od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|------------------------|-----------------------|-----------------------|----------------------|-----------------------|
| 2 | 0 | $6,3633 \cdot 10^{-4}$ | $6,207 \cdot 10^{-3}$ | $2,225 \cdot 10^{-3}$ | $9,09 \cdot 10^{-4}$ | $1,538 \cdot 10^{-3}$ |
| 4 | 0 | 0,43779 | 0,65539 | 0,45610 | 0,50088 | 0,39181 |
| 6 | 0 | 3,00447 | 3,65472 | 2,97886 | 2,81253 | 2,7542 |
| 8 | 0 | 6,83747 | 8,28608 | 7,40881 | 7,28129 | 6,51784 |

Tab. 26. Simulace 2: Odchyly od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|-----------------------|-------------|-----------------------|-----------------------|-----------------------|
| 2 | 0 | $6,964 \cdot 10^{-3}$ | 0,01003 | $4,679 \cdot 10^{-3}$ | $4,823 \cdot 10^{-3}$ | $6,025 \cdot 10^{-3}$ |
| 4 | 0 | 1,01222 | 1,33987 | 0,96123 | 1,01908 | 0,87666 |
| 6 | 0 | 4,18569 | 5,38517 | 4,42186 | 4,5112 | 3,90695 |
| 8 | 0 | 9,41851 | 10,67 | 9,21779 | 9,47737 | 8,84027 |

Při vzrůstajícím počtu dimenzí účelové funkce byla průměrná odchylka od globálního minima vysoká pro všechny testované algoritmy. Nejlépe si vedla chaotická diferenciální evoluce s disipativním systémem a klasická diferenciální evoluce. Průběh 3. Simulace je zobrazen na obr. 17.

Obr. 17. Průběh evolučního procesu pro roztaženou sinusoidální V funkci ($D=10$)

Systém Ikeda je svou úspěšností daleko za ostatními algoritmy. Disipativní systém se stejně jako v předchozích simulacích ukázal jako nejefektivnější. Zhodnocení této simulace je uvedeno v tab. 27.

Tab. 27. Výsledky 3. simulace pro roztaženou sinusoidální V funkci
(10 D)

| Algoritmus | Počet ohodnocení účelové funkce | Doba trvání výpočtu [s] |
|----------------------|------------------------------------|-------------------------|
| KDE | 1781900 | 1294,516 |
| CDE (Ikeda) | 3398000 | 2614,049 |
| CDE (Arnold Cat map) | 1850600 | 1403,320 |
| CDE (Sinai) | 1868300 | 1432,738 |
| CDE (disipativní) | 1387600 | 1026,029 |

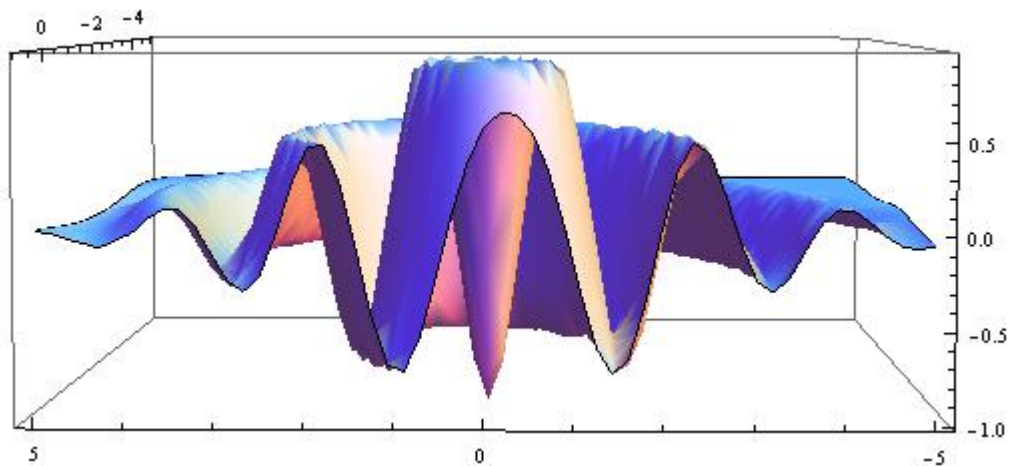
Doba výpočtu globálního minima roztažené sinusoidální V funkce se pohybovala v řádech desítek minut. Nejmenší počet ohodnocení účelové funkce potřeboval disipativní systém a klasická diferenciální evoluce. Systémy Arnold Cat map a Sinai dosáhly velmi podobných výsledků.

10.6 Mastersova funkce

Poslední testovanou funkcí byla Mastersova funkce. Parametry simulací jsou zapsány v tab. 28. Obr. 18 zobrazuje průřez Mastersovou funkcí, na kterém je zřetelně viditelná pozice globálního extrému.

Tab. 28. Parametry simulací pro Mastersovu funkci

| Parametr | Hodnoty pro 1. simulaci | Hodnoty pro 2. simulaci | Hodnoty pro 3. simulaci |
|-------------------|----------------------------|----------------------------|----------------------------|
| Počet generací | 200 | 200 | neomezeně |
| Počet opakování | 20 | 20 | 1 |
| CR | 0,8 | 0,8 | 0,8 |
| F | 0,8 | 0,8 | 0,8 |
| Specimen | {{Real, {-5 ; 5}}} | {{Real, {-5 ; 5}}} | {{Real, {-5 ; 5}}} |
| Velikost populace | 10D | 10D | 8D |
| Strategie | DE/best/2/bin | DE/rand/2/bin | DE/best/2/bin |



Obr. 18. Průřez Mastersovou funkcí

Naměřené hodnoty z 1. a 2. simulace jsou uvedeny v tab. 29 a 30.

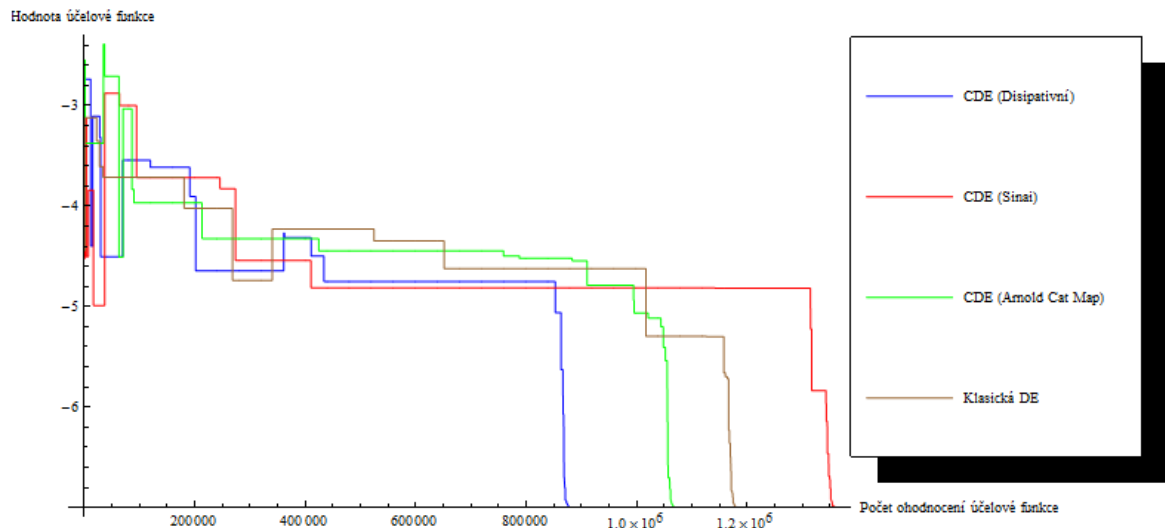
Tab. 29. Simulace 1: Odchyly od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|------------------------|-------------------------|------------------------|------------------------|------------------------|
| 2 | -1 | $2,9928 \cdot 10^{-7}$ | $2,7751 \cdot 10^{-12}$ | $2,2909 \cdot 10^{-9}$ | $6,7912 \cdot 10^{-5}$ | $6,8078 \cdot 10^{-8}$ |
| 4 | -3 | 0,45768 | 0,61467 | 0,45570 | 0,51063 | 0,38445 |
| 6 | -5 | 1,46114 | 1,58719 | 1,55562 | 1,44605 | 1,37862 |
| 8 | -7 | 2,80832 | 3,11487 | 2,84802 | 2,93738 | 2,61412 |

Tab. 30. Simulace 2: Odchyly od známé hodnoty globálního minima

| Počet dimenzí | Známa hodnota minima | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| 2 | -1 | $1,2549 \cdot 10^{-3}$ | $5,3177 \cdot 10^{-3}$ | $3,6101 \cdot 10^{-3}$ | $1,2927 \cdot 10^{-3}$ | $1,8452 \cdot 10^{-3}$ |
| 4 | -3 | 0,49683 | 0,63337 | 0,54748 | 0,54465 | 0,49067 |
| 6 | -5 | 1,56629 | 1,61386 | 1,50073 | 1,53809 | 1,47462 |
| 8 | -7 | 2,92283 | 2,95697 | 2,81450 | 2,92799 | 2,68766 |

U obou simulací se stal nejúspěšnějším algoritmus s disipativním systémem, který následovala klasická diferenciální evoluce a systém Arnold Cat map. U dvouřozměrné funkce v první simulaci bylo dosahováno poměrně dobrých výsledků, zejména pak u systému Ikeda, který v ostatních případech výrazně ztrácí. Průběh 3. simulace nalezneme na obr. 19.



Obr. 19. Průběh evolučního procesu pro Mastersovu funkci ($D=8$)

Při třetí simulaci si chaotická diferenciální evoluce v porovnání s klasickou diferenciální evolucí vedla také velmi dobře. Algoritmy s disipativním systémem a Arnold Cat map dosáhly lepších výsledků než klasická diferenciální evoluce, která překonala pouze diferenciální evoluci se systémem Sinai. U této simulace byl vynechán algoritmus se systémem Ikeda, jelikož se tomuto algoritmu při stanovaném počtu dimenzí účelové funkce nedařilo nalézt hodnotu globálního minima. V grafu také můžeme pozorovat zajímavý jev, kdy pro všechny algoritmy bylo nejobtížnější dosáhnout hodnoty blízké -5, ale po dosažení této hodnoty již bylo nalezení globálního minima velmi rychlé. Shrnutí 3. simulace je uvedeno v tab. 31.

Tab. 31. Výsledky 3. simulace pro Mastersovu funkci

| Algoritmus | Počet ohodnocení účelové funkce | Doba trvání výpočtu [s] |
|----------------------|---------------------------------|-------------------------|
| KDE | 1179040 | 588,22 |
| CDE (Arnold Cat map) | 1067360 | 678,386 |
| CDE (Sinai) | 1357840 | 886,304 |
| CDE (disipativní) | 877200 | 575,531 |

11 VLIV ROZMĚRU ÚČELOVÉ FUNKCE

Účelem této kapitoly je zjistit, jak se algoritmy chovají při vzrůstajícím počtu rozměrů účelové funkce. Jako testovací funkce byla vybrána Ackleyho funkce II a Schwefelova funkce.

11.1 Testování na Ackleyho funkci II

Parametry diferenciální evoluce byly nastaveny dle tab. 32.

Tab. 32. Parametry simulací pro Ackleyho funkci II

| Parametr | Hodnota |
|-------------------|--------------------|
| Počet generací | neomezeně |
| Počet opakování | 1 |
| CR | 0,8 |
| F | 0,8 |
| Specimen | {{Real, {-2 ; 2}}} |
| Velikost populace | 10D |
| Strategie | DE/best/2/bin |

Prvotní simulace byla provedena na účelové funkci o dvou dimenzích a s každou další simulací byl počet dimenzí navýšen. Po nalezení hodnoty globálního minima byl vždy zapsán počet ohodnocení účelové funkce a celkový čas výpočtu. Získané hodnoty jsou uvedeny v tab. 33 a 34.

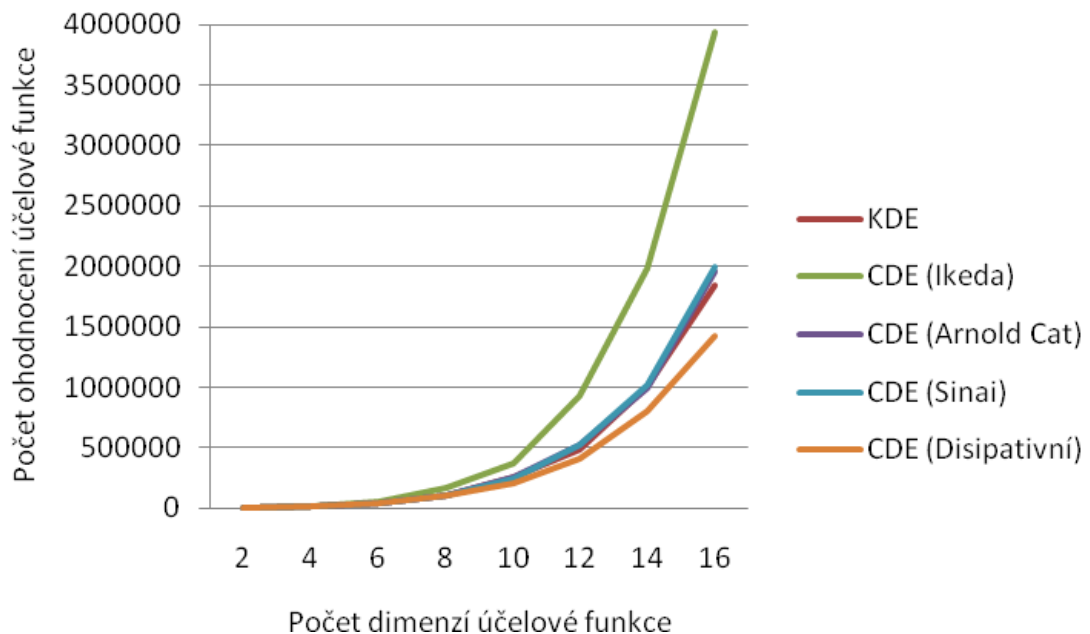
Tab. 33. Počet ohodnocení účelové funkce při zvyšujícím se počtu dimenzí

| Počet dimenzí | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|---------|-------------|------------------|-------------|-------------------|
| 2 | 3080 | 2940 | 2940 | 2960 | 2920 |
| 4 | 13640 | 15520 | 15320 | 13200 | 14760 |
| 6 | 44280 | 54300 | 37080 | 45540 | 39480 |
| 8 | 101200 | 165760 | 106000 | 108960 | 100320 |
| 10 | 252700 | 373500 | 252900 | 245400 | 209600 |
| 12 | 490440 | 934800 | 521280 | 528000 | 411840 |
| 14 | 1001000 | 1978620 | 996800 | 1023960 | 802620 |
| 16 | 1845280 | 3932960 | 1952640 | 1989760 | 1419200 |

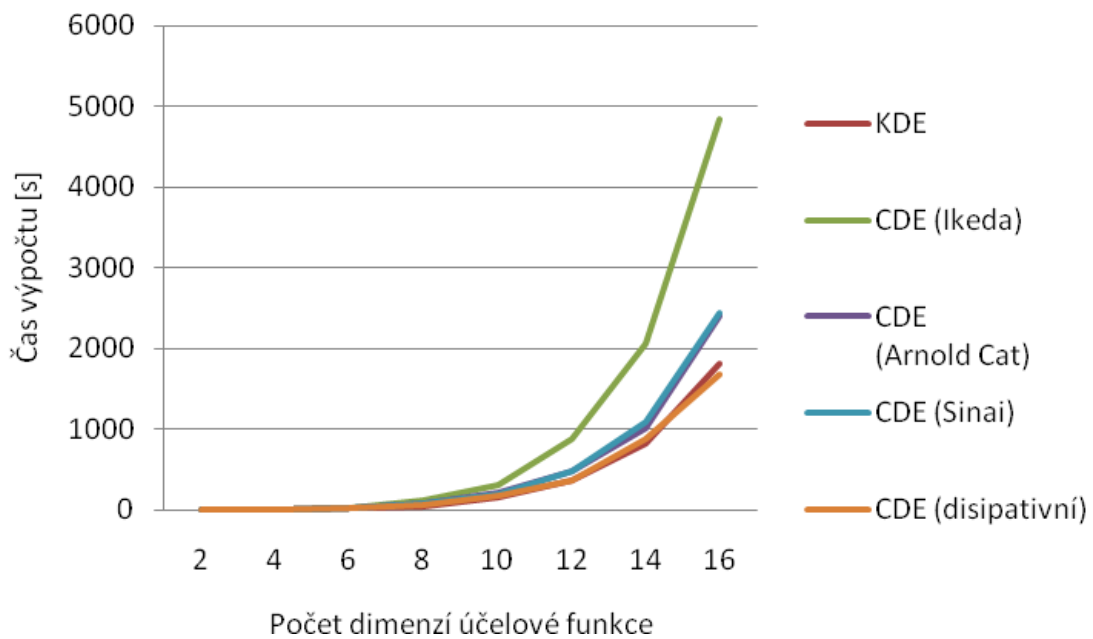
Tab. 34. Čas výpočtu v sekundách při zvyšujícím se počtu dimenzí

| Počet dimenzí | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------|-------------|------------------|-------------|-------------------|
| 2 | 0,639 | 1,092 | 1,07 | 1,062 | 1,016 |
| 4 | 3,834 | 6,426 | 6,238 | 5,352 | 6,220 |
| 6 | 16,498 | 28,290 | 18,884 | 23,772 | 20,258 |
| 8 | 50,054 | 111,228 | 70,261 | 71,977 | 68,787 |
| 10 | 163,999 | 307,628 | 206,589 | 199,440 | 172,275 |
| 12 | 366,205 | 875,705 | 471,084 | 483,090 | 373,286 |
| 14 | 829,155 | 2050,600 | 1014,212 | 1091,662 | 875,757 |
| 16 | 1808,487 | 4839,797 | 2408,706 | 2437,908 | 1687,085 |

Se zvyšujícím se rozměrem účelové funkce velmi rapidně narůstá jak počet ohodnocení účelové funkce, tak i celková doba výpočtu. Zatímco u dvourozměrné funkce stačila pro nalezení extrému pouhá jedna vteřina, u deseti-rozměrné funkce se doba pohybovala v rámci minut. U šestnácti-rozměrné funkce již doba výpočtu překročila hranici jedné hodiny. Pro lepší porovnání efektivity jednotlivých algoritmů byla do obr. 20 a 21 zakreslena závislost počtu ohodnocení účelové funkce a doby výpočtu na vzrůstajícím počtu dimenzí účelové funkce.



Obr. 20. Počet ohodnocení účelové funkce (Ackley II)



Obr. 21. Čas výpočtu (Ackley II)

U obou grafů pozorujeme od desetirozměrné účelové funkce prudký nárůst počtu ohodnocení účelové funkce a výpočetního času. Lze předpokládat, že podobný trend bude pokračovat i nadále, tudíž bychom se mohli při dalším navyšování počtu rozměrů účelové funkce snadno dostat do situace, kdy se bude výpočetní doba pohybovat v rámci dnů, týdnů či měsíců.

Chaotická diferenciální evoluce s disipativním systémem byla nejvíce efektivní z hlediska počtu ohodnocení účelové funkce a v tomto směru úspěšně překonávala i klasickou diferenciální evoluci. Chaotické diferenciální evoluce se systémy Arnold Cat map a Sinai podávaly velmi podobné výsledky, které byly srovnatelné s klasickou diferenciální evolucí. Nejméně efektivní se ukázala diferenciální evoluce se systémem Ikeda, která oproti ostatním algoritmům často potřebovala přibližně dvojnásobný počet ohodnocení účelové funkce. Průběh časové náročnosti kopíruje trend z předchozího grafu. Jediný podstatný rozdíl je v tom, že rozdíly mezi klasickou diferenciální evolucí a chaotickou diferenciální evolucí s disipativním systémem se stírají a tudíž nelze s určitostí říci, který algoritmus je z hlediska časové náročnosti lepší. Je však potřeba zdůraznit, že klasická diferenciální evoluce využívá nativní funkci pro generování náhodných čísel, která je podstatně rychlejší

než speciální funkce, která na základě vygenerované simulace chování chaotického systému vrací hodnotu podobně jako generátor náhodných čísel.

11.2 Testování na Schwefelově funkci

U Schwefelovy funkce došlo pouze k úpravě rozsahu vzorového jedince. Všechny parametry evolučního procesu jsou uvedeny v tab. 35.

Tab. 35. Parametry simulací pro Schwefelovu funkci

| Parametr | Hodnota |
|-------------------|------------------------|
| Počet generací | neomezeně |
| Počet opakování | 1 |
| CR | 0,8 |
| F | 0,8 |
| Specimen | {{Real, {-500 ; 500}}} |
| Velikost populace | 10D |
| Strategie | DE/best/2/bin |

Jednotlivé simulace byly opakovány od 2 do 16 rozměrů účelové funkce. Měřen byl opět počet ohodnocení účelové funkce a celková doba výpočtu. Získaná data jsou uvedena v tab. 36 a 37.

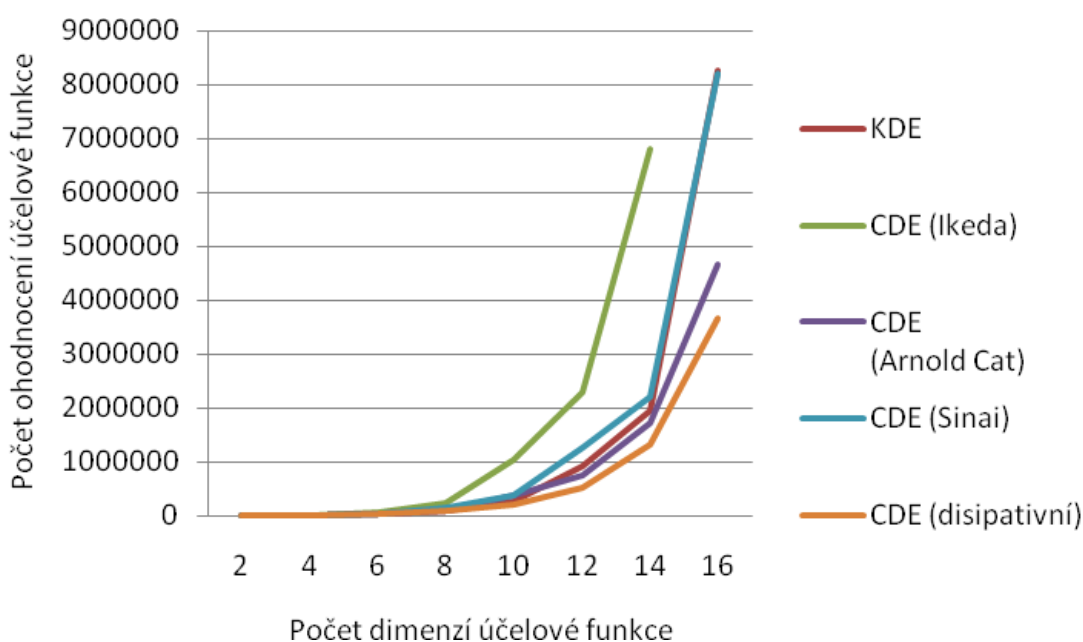
Tab. 36. Počet ohodnocení účelové funkce při zvyšujícím se počtu dimenzí

| Počet dimenzí | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|---------|-------------|------------------|-------------|-------------------|
| 2 | 1140 | 1580 | 1040 | 1340 | 1240 |
| 4 | 8000 | 9800 | 7640 | 8360 | 7960 |
| 6 | 33900 | 49440 | 27780 | 28920 | 24660 |
| 8 | 99760 | 219120 | 96160 | 138080 | 82720 |
| 10 | 271500 | 1031600 | 366000 | 389600 | 210900 |
| 12 | 914400 | 2303640 | 759600 | 1259160 | 515040 |
| 14 | 1947820 | 6816880 | 1725780 | 2206960 | 1306060 |
| 16 | 8247200 | - | 4666720 | 8200960 | 3671840 |

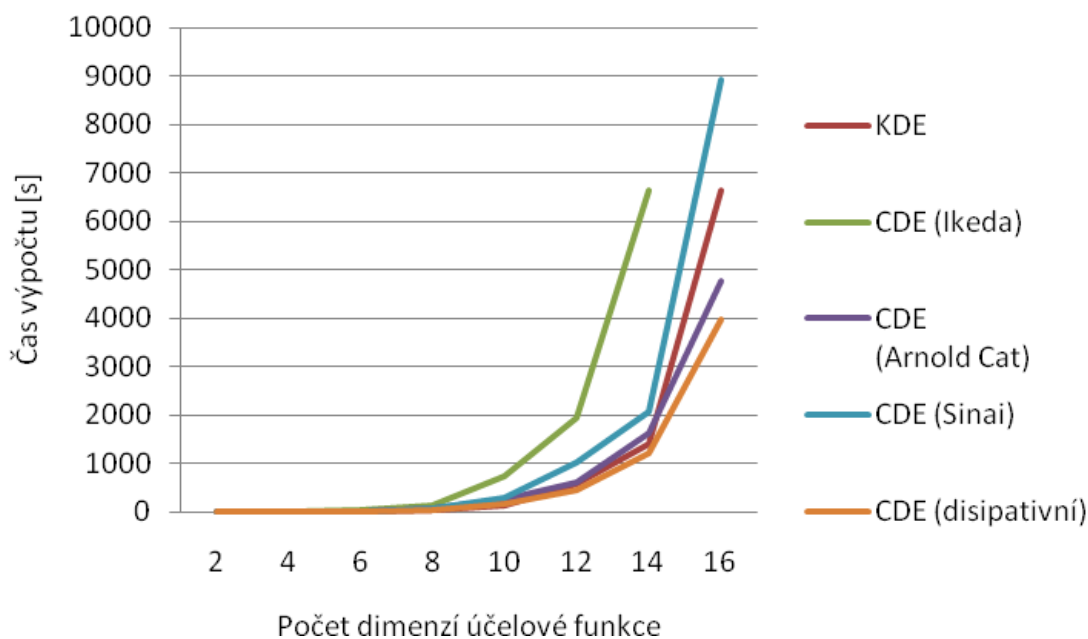
Tab. 37. Čas výpočtu v sekundách při zvyšujícím se počtu dimenzí

| Počet dimenzí | KDE | CDE (Ikeda) | CDE (Arnold Cat) | CDE (Sinai) | CDE (disipativní) |
|---------------|----------|-------------|------------------|-------------|-------------------|
| 2 | 0,212 | 0,580 | 0,370 | 0,480 | 0,470 |
| 4 | 2,010 | 3,820 | 3,000 | 3,260 | 3,230 |
| 6 | 11,342 | 24,518 | 13,222 | 13,624 | 11,458 |
| 8 | 41,969 | 126,627 | 54,426 | 77,965 | 46,249 |
| 10 | 146,369 | 741,357 | 257,157 | 282,753 | 158,216 |
| 12 | 585,266 | 1957,110 | 594,795 | 1029,744 | 442,962 |
| 14 | 1397,695 | 6638,115 | 1614,096 | 2067,527 | 1217,391 |
| 16 | 6631,818 | - | 4762,386 | 8915,719 | 3958,490 |

Nalezení hodnoty globálního minima Schwefelovy funkce bylo pro $D > 10$ mnohem více časově náročné než u předchozí Ackleyho funkce II. Chaotická diferenciální evoluce se systémem Ikeda dokonce se nebyla schopna ani po 10 hodinách alespoň přiblížit k hodnotě globálního minima pro $D = 16$. Z tohoto důvodu údaje v obou výše uvedených tabulkách chybí. Grafické znázornění výsledků je uvedeno na obr. 22 a 23.



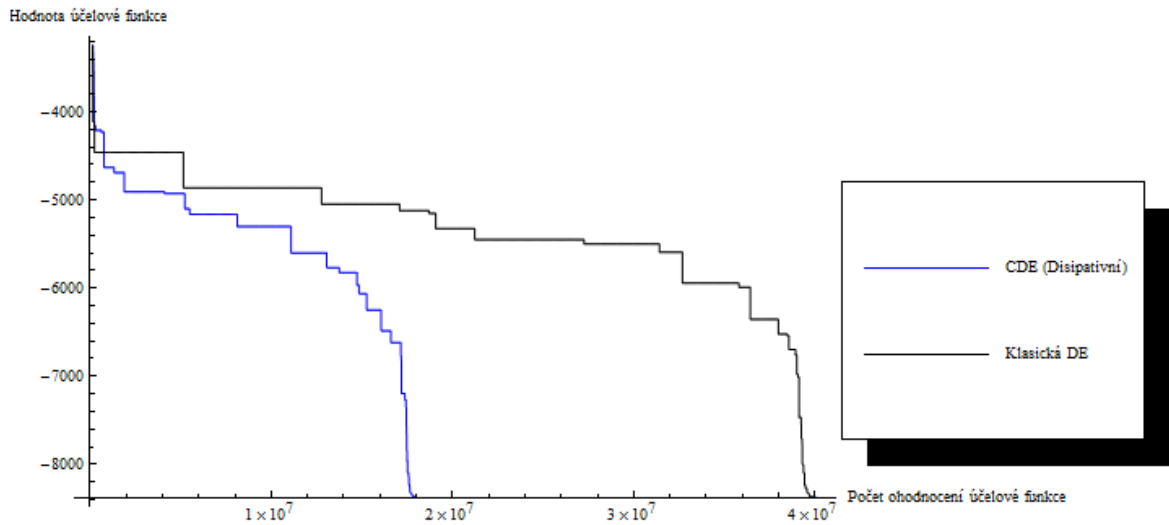
Obr. 22. Počet ohodnocení účelové funkce (Schwefel)



Obr. 23. Čas výpočtu (Schwefel)

Chaotická diferenciální evoluce s disipativním systémem potřebovala nejméně ohodnocení účelové funkce pro nalezení globálního minima pro všechny simulace s $D > 2$. Velmi dobrých výsledků dosahoval i systém Arnold Cat map. Obě chaotické diferenciální evoluce byly výrazně lepší než klasická diferenciální evoluce. Systém Sinai dosahoval horších výsledků než klasická diferenciální evoluce. Od ostatních algoritmů se značně odlišuje chaotická diferenciální evoluce se systémem Ikeda, která je velmi náročná na počet ohodnocení účelové funkce. Graf závislosti času výpočtu na počtu dimenzí účelové funkce je velmi podobný grafu předchozímu. U účelové funkce se 16 dimenzemi se již doba výpočtu pohybovala v řádu hodin.

Následně byl spuštěn evoluční proces na Schwefelově funkci o 20 dimenzích. Simulace byla provedena pro nejúspěšnější chaotickou diferenciální evoluci z předchozích simulací a klasickou diferenciální evoluci. Průběhy obou simulací jsou viditelné na obr. 24.



Obr. 24. Evoluční proces pro $D=20$

Chaotická diferenciální evoluce s disipativním systémem zde zcela jednoznačně překonává klasickou diferenciální evoluci. Klasické diferenciální evoluci trval výpočet přibližně 13 hodin, kdežto chaotické diferenciální evoluci pouhých 6 hodin a 38 minut. Časová úspora je tedy enormní. Podobný rozdíl nacházíme i v počtu ohodnocení účelové funkce. Přesné hodnoty jsou uvedeny v tab. 38.

Tab. 38. Výsledky po nalezení globálního minima při $D=20$

| Algoritmus | Počet ohodnocení účelové funkce | Doba trvání výpočtu [s] |
|-------------------|---------------------------------|-------------------------|
| KDE | 40320400 | 46821,043 |
| CDE (disipativní) | 18230800 | 23923,702 |

12 POROVNÁNÍ ALGORITMŮ

Pro každou testovací funkci byly provedeny 3 simulace. V prvních dvou byla měřena průměrná odchylka od hodnoty globálního minima pro strategii DE/best/2/bin a DE/rand/2/bin. Do 3. Simulace byla vždy vybrána strategie, která byla v prvních dvou simulacích nejlépeší. Ve všech případech byla pro třetí simulaci použita strategie DE/best/2/bin, protože stabilně podávala lepší výsledky než DE/rand/2/bin. Účelem 3. simulace bylo nechat běžet evoluční proces tak dlouho, dokud nebude nalezena hodnota globálního minima. Měřenou veličinou zde byl počet ohodnocení účelové funkce a celkový čas výpočtu.

U Schwefelovy funkce podávala nejlepší výkony chaotická diferenciální evoluce s disipativním systémem následovaná klasickou diferenciální evolucí. Při třetí simulaci překonala klasickou diferenciální evoluci i chaotická diferenciální evoluce se systémem Sinai. Při všech simulacích na Rastriginově funkci dosahovala nejlepších výsledků chaotická diferenciální evoluce s disipativním systémem. Ve třetí simulaci se také ukázalo, že algoritmy se Systémem Sinai a Arnold Cat map mají s tímto typem funkce problém, jelikož potřebovaly výrazně vyšší počet ohodnocení účelové funkce při porovnání se zbývajícími algoritmy. Na Ackleyho funkci I byla rovněž nejlépeší chaotická diferenciální evoluce s disipativním systémem, která jako jediná v prvních dvou simulacích překonávala klasickou diferenciální evoluci. U třetí simulace podala dobré výsledky i chaotická diferenciální evoluce se systémem Arnold Cat map. Při aplikaci na Ackleyho funkci II se podařilo překonat klasickou diferenciální evoluci jen chaotickou diferenciální evolucí s disipativním systémem. Ve třetí simulaci byl disipativní systém rovněž úspěšnější, ale pouze vzhledem k počtu ohodnocení účelové funkce. Disipativní systém překonal klasickou diferenciální evoluci i na roztažené sinusoidální V funkci. Algoritmy se systémy Arnold Cat map a Sinai podávaly velmi podobné výsledky, které však byly horší než u klasické diferenciální evoluce. Na Mastersově funkci se při třetí simulaci osvědčily dva algoritmy chaotické diferenciální evoluce – algoritmus s disipativním systémem a se systémem Arnold Cat map. Oba byly z hlediska počtu ohodnocení účelové funkce efektivnější než klasická diferenciální evoluce.

Nejhorší výsledky podávala chaotická diferenciální evoluce se systémem Ikeda, která byla svou úspěšností daleko za ostatními algoritmy. V některých případech byly její výsledky i několikanásobně horší v porovnání s ostatními algoritmy.

Na nárůst počtu dimenzí Ackleyho funkce II nejlépe reagovala chaotická diferenciální evoluce s disipativním systémem, jelikož hodnotu globálního minima nacházela s nejmenším počtem ohodnocení účelové funkce. Co se týče doby výpočtu, tak zde byly neúspěšnější algoritmy klasické diferenciální evoluce a chaotické diferenciální evoluce s disipativním systémem. U Schwefelovy funkce se s nárůstem dimenzí opět nejlépe vyrovnala chaotická diferenciální evoluce s disipativním systémem a také chaotická diferenciální evoluce se systémem Arnold Cat map. Oba algoritmy podávaly podstatně lepší výsledky než klasická diferenciální evoluce a to i z hlediska časové náročnosti výpočtu.

ZÁVĚR

Účelem diplomové práce bylo porovnat klasickou diferenciální evoluci s chaotickou diferenciální evolucí. Během simulací bylo provedeno 222 437 440 ohodnocení účelové funkce. Na základě těchto simulací se ukázalo, že chaotický systém může být v algoritmu diferenciální evoluce velice prospěšný, anebo může kvalitu evolučního procesu naopak zhoršovat. Volba správného chaotického systému je tedy rozhodující, jelikož mezi nimi mohou být v tomto směru propastné rozdíly. Chaotický systém Ikeda se v diferenciální evoluci neosvědčil, protože stabilně podával nejhorší výsledky. Systémy Sinai a Arnold Cat map podávaly výkony srovnatelné s klasickou diferenciální evolucí a mohou tak být možnou alternativou.

Nejlepší výsledky v naprosté většině simulací podávala chaotická diferenciální evoluce s disipativním systémem se strategií DE/best/2/bin. Tento algoritmus poskytoval nejmenší odchylky od globálního minima. Pokud byl evoluční proces zastaven až po nalezení globálního minima, disipativní systém provedl nejmenší počet ohodnocení účelové funkce. Tato vlastnost se projevovala zejména při navyšování dimenzí účelové funkce. Disipativní systém byl také velice úspěšný z hlediska časové náročnosti. Ve většině případů byl výpočet s disipativním systémem nejrychlejší. Tuto výhodu je možné dále využít a pomocí navýšení výpočetního výkonu nebo paralelizací výpočtu dosáhnout ještě lepších výsledků.

Chaotickou diferenciální evoluci s disipativním systémem mohu doporučit jako náhradu za klasickou diferenciální evoluci. Svou sílu předvádí hlavně v případech, kdy má účelová funkce vyšší počet dimenzí, tudíž se hodí na řešení složitých optimalizačních problémů. Tento algoritmus se tedy může stát dalším stupněm v evoluci samotné diferenciální evoluce.

Do práce nebylo možné zahrnout všechny možné varianty chaotické diferenciální evoluce, jelikož bychom mohli nalézt až stovky typů algoritmů, které kombinují určitou strategii klasické diferenciální evoluce s velkým počtem chaotických systémů. Chaotické systémy mají stejně jako diferenciální evoluce své parametry, které nebyly v průběhu práce měněny, tudíž se zde nabízí prostor pro další výzkumnou činnost. Tato práce tedy může sloužit jako podklad pro další studium chaotické diferenciální evoluce, jelikož se domnívám, že tento směr bude pro algoritmus diferenciální evoluce přínosný.

CONCLUSION

The purpose of this thesis has been to compare classical differential evolution with chaotic differential evolution. 222 437 440 cost function evaluations has been made during simulations. On the basis of these simulations, it has been show that presence of chaotic system in a differential evolution algorithm may either be very beneficial or may reduce the quality of evolutionary process. The choice of correct chaotic system is critical because of the vast differences among them. Chaotic system Ikeda was unsuccessful in the differential evolution as it consistently produced the worst results. The Sinai and Arnold Cat map systems produced results comparable with classical version of differential evolution, and those systems may provide a possible alternative.

Best results in the vast majority of simulations were produced by chaotic differential evolution with a dissipative system with strategy DE/best/2/bin. This algorithm provided the smallest divergences from a global minimum. If the evolutionary process was stopped after the retrieval of global minimum, dissipative system carried out the lowest number of cost function evaluations. This feature was especially visible during the increase of cost function dimensions. The dissipative system was also very successful in the matter of time complexity. Calculations with dissipative system were fastest one in most cases. It is possible to use this advantage and achieve even better results with increased computing performance or by parallelization of calculations. I recommend chaotic differential evolution with dissipative system as a replacement of classical differential evolution. It especially shows its power when the cost function has higher number of dimesions and thus it fits for solving complicated optimization problems. This algorithm may become the next level in the evolution of differential evolution itself.

It was not possible to include all variations of differential evolution into this thesis, because we may find even hundreds of algorithm types, which combine certain strategy of differential evolution with a large number of chaotic systems. Chaotic systems have, similarly to differential evolution their own parameters which had not been changed during the work on this thesis, that means that there is a possibility for further research. Therefore, this thesis may serve as a basis for additional study of chaotic differential evolution, because I expect that this tendency will be beneficial for differential evolution algorithm.

SEZNAM POUŽITÉ LITERATURY

- [1] ZELINKA, Ivan, et al. *Evoluční výpočetní techniky : Principy a aplikace*. Praha : BEN, 2008. 536 s. ISBN 978-80-7300-218-3.
- [2] OŠMERA, Pavel. *Evoluční algoritmy a jejich aplikace*. Praha : České vysoké učení technické v Praze, 2008. 30 s.
- [3] ZELINKA, Ivan. *Umělá inteligence v problémech globální optimalizace* . Praha : BEN, 2002. 192 s. ISBN 80-7300-069-5.
- [4] PRICE, Kenneth V.; STORN, Rainer M.; LAMPINEN, Jouni A. *Differential Evolution : A Practical Approach to Global Optimization*. Berlin : Springer, 2005. 538 s. ISBN 978-3-540-20950-8.
- [5] ONWUBOLU, Godfrey C.; DAVENDRA, Donald. *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*. Berlin : Springer, 2009. 214 s. ISBN 978-3-540-92151-6.
- [6] COELHO, Leandro dos Santos. Reliability–redundancy optimization by means of a chaotic differential evolution approach. *Chaos, Solitons and Fractals* [online]. 2009, 41, [cit. 2011-02-27]. Dostupný z WWW: <<http://www.produstronica.pucpr.br/sip/conteudo/ProdBibliog/Leandro/Realibility-artigo7.pdf>>. ISSN 0960-0779.
- [7] SCHOLL, Ekehard; SCHUSTER, Heinz Georg. *Handbook of Chaos Control*. 2nd Ed. Weinheim : WILEY-VCH Verlag GmbH & Co. KGaA, 2008. 849 s. ISBN 978-3-527-40605-0.
- [8] HILBORN, Robert C. *Chaos and nonlinear dynamics : an introduction for scientists and engineers*. New York : Oxford University Press, Inc., 1994. 672 s. ISBN 0-19-508816-8.
- [9] MAY, Robert M. *Stability and Complexity in Model Ecosystems*. Princeton : Princeton University Press, 2001. ISBN 0-691-08861-6.
- [10] ELERT, Glenn. *The Chaos Hypertextbook* [online]. 1995-2007 [cit. 2011-02-27]. 4.3 Lyapunov Exponent. Dostupné z WWW: <<http://hypertextbook.com/chaos/43.shtml>>.

- [11] SPROTT, Julien C. *Chaos and time-series analysis*. Oxford : Oxford University Press, 2003. 528 s. ISBN 0198508395.
- [12] BOTEV, Zdravko, et al. *School of Mathematics and Physics* [online]. The University of Queensland, 2004-12-17 [cit. 2011-04-30]. Non-linear Continuous Multi-Extremal Optimization. Dostupné z WWW: <<http://www.maths.uq.edu.au/CEToolBox/node3.html>>.
- [13] Wolfram Research. *WolframTones* [online]. 2011 [cit. 2011-03-20]. More about Mathematica. Dostupné z WWW: <<http://tones.wolfram.com/about/mathematica-more.html>>.
- [14] SHIFRIN, Leonid. *Mathematica programming: an advanced introduction* [online]. 2008 [cit. 2011-03-20]. Dostupné z WWW: <<http://www.mathprogramming-intro.org/download/MathProgrammingIntro.pdf>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

| | |
|-----|---|
| CDE | Chaotická diferenciální evoluce |
| CR | Práh křížení diferenciální evoluce |
| D | Dimenze účelové funkce |
| DE | Diferenciální evoluce |
| F | Mutační konstanta diferenciální evoluce |
| KDE | Klasická diferenciální evoluce |
| NP | Velikost populace diferenciální evoluce |

SEZNAM OBRÁZKŮ

| | |
|---|----|
| <i>Obr. 1. Zjednodušené schéma diferenciální evoluce [1]</i> | 18 |
| <i>Obr. 2. Ukázka exponenciálního křížení [4]</i> | 21 |
| <i>Obr. 3. Ukázka binomického křížení [4]</i> | 21 |
| <i>Obr. 4. Atraktor Ikeda v prostředí Mathematica</i> | 24 |
| <i>Obr. 5. Ukázka bifurkačního diagramu</i> | 26 |
| <i>Obr. 6. Cirkulace kapaliny v Lorenzově modelu [8]</i> | 27 |
| <i>Obr. 7. Bifurkační diagram logistické rovnice v prostředí Mathematica</i> | 29 |
| <i>Obr. 8. Trajektorie s různými hodnotami Ljapunovova exponentu [10]</i> | 31 |
| <i>Obr. 9. Atraktory chaotických systémů. Zleva doprava a shora dolů: Ikeda, Sinai, Arnold Cat map, Disipativní systém.</i> | 32 |
| <i>Obr. 10. Schwefelova funkce</i> | 35 |
| <i>Obr. 11. Průběhy evolučního procesu při mnohonásobném opakování</i> | 36 |
| <i>Obr. 12. Grafická reprezentace odchylek pro klasickou diferenciální evoluci</i> | 43 |
| <i>Obr. 13. Grafická reprezentace odchylek pro chaotickou diferenciální evoluci</i> | 43 |
| <i>Obr. 14. Průběh evolučního procesu pro Schwefelovu funkci ($D=10$)</i> | 46 |
| <i>Obr. 15. Rastriginova funkce</i> | 47 |
| <i>Obr. 16. Průběh evolučního procesu pro Ackleyho funkci I ($D=10$)</i> | 50 |
| <i>Obr. 17. Průběh evolučního procesu pro roztaženou sinusoidální V funkci ($D=10$)</i> | 54 |
| <i>Obr. 18. Průřez Mastersovou funkcí</i> | 56 |
| <i>Obr. 19. Průběh evolučního procesu pro Mastersovu funkci ($D=8$)</i> | 57 |
| <i>Obr. 20. Počet ohodnocení účelové funkce (Ackley II)</i> | 59 |
| <i>Obr. 21. Čas výpočtu (Ackley II)</i> | 60 |
| <i>Obr. 22. Počet ohodnocení účelové funkce (Schwefel)</i> | 62 |
| <i>Obr. 23. Čas výpočtu (Schwefel)</i> | 63 |
| <i>Obr. 24. Evoluční proces pro $D=20$</i> | 64 |

SEZNAM TABULEK

| | |
|--|----|
| <i>Tab. 1. Doporučené hodnoty řídicích parametrů [1]</i> | 19 |
| <i>Tab. 2. Přehled vybraných chaotických systémů [11]</i> | 31 |
| <i>Tab. 3. Parametry simulací.....</i> | 39 |
| <i>Tab. 4. Nejlepší a nejhorší výsledky</i> | 39 |
| <i>Tab. 5. Průměrná odchylka od známé hodnoty globálního extrému</i> | 40 |
| <i>Tab. 6. Matice výsledků klasické diferenciální evoluce pro parametry F a CR</i> | 41 |
| <i>Tab. 7. Matice výsledků chaotické diferenciální evoluce pro parametry F a CR.....</i> | 42 |
| <i>Tab. 8. Parametry simulací pro Schwefelovu funkci</i> | 44 |
| <i>Tab. 9. Simulace 1: Odchylky od známé hodnoty globálního minima.....</i> | 45 |
| <i>Tab. 10. Simulace 2: Odchylky od známé hodnoty globálního minima.....</i> | 45 |
| <i>Tab. 11. Výsledky 3. simulace pro Schwefelovu funkci (10 D)</i> | 46 |
| <i>Tab. 12. Parametry simulací pro Rastriginovu funkci.....</i> | 47 |
| <i>Tab. 13. Simulace 1: Odchylky od známé hodnoty globálního minima.....</i> | 48 |
| <i>Tab. 14. Simulace 2: Odchylky od známé hodnoty globálního minima.....</i> | 48 |
| <i>Tab. 15. Výsledky 3. simulace pro Rastriginovu funkci (10 D)</i> | 48 |
| <i>Tab. 16. Parametry simulací pro Ackleyho funkci I</i> | 49 |
| <i>Tab. 17. Simulace 1: Odchylky od známé hodnoty globálního minima.....</i> | 49 |
| <i>Tab. 18. Simulace 2: Odchylky od známé hodnoty globálního minima.....</i> | 50 |
| <i>Tab. 19. Výsledky 3. simulace pro Ackleyho funkci I (10 D)</i> | 51 |
| <i>Tab. 20. Parametry simulací pro Ackleyho funkci II</i> | 51 |
| <i>Tab. 21. Simulace 1: Odchylky od známé hodnoty globálního minima.....</i> | 52 |
| <i>Tab. 22. Simulace 2: Odchylky od známé hodnoty globálního minima.....</i> | 52 |
| <i>Tab. 23. Výsledky 3. simulace pro Ackleyho funkci II (10 D).....</i> | 52 |
| <i>Tab. 24. Parametry simulací pro roztaženou sinusoidální V funkci.....</i> | 53 |
| <i>Tab. 25. Simulace 1: Odchylky od známé hodnoty globálního minima.....</i> | 53 |
| <i>Tab. 26. Simulace 2: Odchylky od známé hodnoty globálního minima.....</i> | 54 |
| <i>Tab. 27. Výsledky 3. simulace pro roztaženou sinusoidální V funkci (10 D)</i> | 55 |
| <i>Tab. 28. Parametry simulací pro Mastersovu funkci.....</i> | 55 |
| <i>Tab. 29. Simulace 1: Odchylky od známé hodnoty globálního minima.....</i> | 56 |
| <i>Tab. 30. Simulace 2: Odchylky od známé hodnoty globálního minima.....</i> | 56 |
| <i>Tab. 31. Výsledky 3. simulace pro Mastersovu funkci</i> | 57 |

| | |
|---|-----------|
| <i>Tab. 32. Parametry simulací pro Ackleyho funkci II</i> | <i>58</i> |
| <i>Tab. 33. Počet ohodnocení účelové funkce při zvyšujícím se počtu dimenzí</i> | <i>58</i> |
| <i>Tab. 34. Čas výpočtu v sekundách při zvyšujícím se počtu dimenzí</i> | <i>59</i> |
| <i>Tab. 35. Parametry simulací pro Schwefelovu funkci</i> | <i>61</i> |
| <i>Tab. 36. Počet ohodnocení účelové funkce při zvyšujícím se počtu dimenzí</i> | <i>61</i> |
| <i>Tab. 37. Čas výpočtu v sekundách při zvyšujícím se počtu dimenzí</i> | <i>62</i> |
| <i>Tab. 38. Výsledky po nalezení globálního minima při $D=20$</i> | <i>64</i> |

SEZNAM PŘÍLOH

Příloha PI: CD-ROM obsahující vlastní práci a zdrojové kódy