

Nástroj pro vizualizaci vnitřní dynamiky vybraných evolučních výpočetních technik

Pavel Šťasta

Bakalářská práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Pavel ŠŤASTA**

Osobní číslo: **A10804**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Forma studia: **prezenční**

Téma práce: **Nástroj pro vizualizaci vnitřní dynamiky vybraných evolučních výpočetních technik**

Téma anglicky: **A Tool for the Visualization of the Internal Dynamics of Selected Evolutionary Computation Techniques**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Prozkoumejte možnosti exportování vícerozměrných polí v jazyce C a jejich následného importování v prostředí Wolfram Mathematica.
3. Naprogramujte interaktivní nástroje pro vizualizaci chování populace vybraných evolučních algoritmů.
4. Prozkoumejte možnosti vizualizace u vícerozměrných problémů.
5. Vytvořte demonstrativní aplikaci pro představení možností jednotlivých nástrojů.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ZELINKA, Ivan. Umělá inteligence v problémech globální optimalizace. BEN, 2002, 190 s. ISBN 80-7300-069-5.
2. ZELINKA, Ivan. Evoluční výpočetní techniky: principy a aplikace. 1. vyd. Praha: BEN – technická literatura, 2009, 534 s. ISBN 978-80-7300-218-3.
3. HEROUT, Pavel. Učebnice jazyka C. Praha: Kopp, 2004, 271 s. ISBN 8072322206. MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J.: Umělá inteligence, Academia, 1993, ISBN 80-200-0496-3.
4. TROTT, Michael. The Mathematica guidebook for programming. New York: Springer, c2004, xxxvii, 1028 s. ISBN 0-387-94282-3.
5. TROTT, Michael. The Mathematica guidebook for graphics. New York: Springer, c2004, xxxv, 1340 s. ISBN 0-387-95010-9.
6. TROTT, Michael. The Mathematica guidebook for symbolics. New York: Springer, c2006, xxxviii, 1453 s. ISBN 0-387-95020-6.
7. TROTT, Michael. The Mathematica guidebook for numerics. New York: Springer, 2006, xxxv, 1208 s. ISBN 0-387-95011-7.
8. PLUHACEK, Michal. PSO Algoritmus v prostředí Mathematica. Zlín, 2011. Diplomová práce. Univerzita Tomáše Bati ve Zlíně.

Vedoucí bakalářské práce:

Ing. Michal Pluháček

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

6. března 2015

Termín odevzdání bakalářské práce:

22. května 2015

Ve Zlíně dne 6. března 2015



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

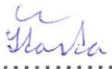
Prohlašuji, že

- 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 připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/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/bakalářské 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

ABSTRAKT

Hlavním cílem bakalářské práce je vytvoření nástroje pro vizualizaci vnitřní dynamiky evolučních výpočetních technik v prostředí Wolfram Mathematica. Teoretická část obsahuje úvod do tématu evolučních výpočetních technik, princip evolučních algoritmů, popis testovacích funkcí, popis možností vizualizace a v neposlední řadě také popis prostředí Wolfram Mathematica a jazyku C. Praktická část se pak věnuje exportu dat v jazyku C a importu dat do prostředí Wolfram Mathematica, analýze těchto dat a popisu samotného vizualizačního nástroje.

Klíčová slova: evoluční algoritmy, vizualizace, Wolfram Mathematica

ABSTRACT

The main goal of this bachelor thesis is to create a tool for the visualization of the internal dynamics of evolutionary computation techniques in Wolfram Mathematica. The theoretical part contains an introduction to the topic of evolutionary computation techniques, principle of evolutionary algorithms, description of test functions, description of visualization possibilities and last but not least, the description of the Wolfram Mathematica environment and the C programming language. The practical part deals with export data in C and importing the data into the Wolfram Mathematica environment, analysis of these data and the description of visualization tool.

Keywords: evolutionary algorithms, visualization, Wolfram Mathematica

Rád bych poděkoval vedoucímu mé práce panu Ing. Michalu Pluháčkovi za odborné vedení, za pomoc a rady při zpracování této práce.

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

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 EVOLUČNÍ VÝPOČETNÍ TECHNIKY.....	11
1.1 PRINCIP EVOLUČNÍCH ALGORITMŮ	11
1.2 OBECNÝ CYKLUS EVOLUČNÍHO ALGORITMU	12
1.3 ZÁKLADNÍ POJMY	12
1.3.1 Účelová funkce.....	12
1.3.2 Populace	13
1.3.3 Jedinec a jeho reprezentace	14
2 TESTOVACÍ FUNKCE.....	16
2.1 UNIMODÁLNÍ.....	16
2.1.1 První De Jongova funkce	16
2.1.2 Třetí De Jongova funkce	17
2.1.3 Čtvrtá De Jongova funkce	19
2.2 MULTIMODÁLNÍ FUNKCE.....	20
2.2.1 Schwefelova funkce	20
2.2.2 Rastriginova funkce	22
3 VIZUALIZACE.....	24
4 WOLFRAM MATHEMATICA	25
4.1 ZÁKLADNÍ VLASTNOSTI.....	25
4.1.1 Programovací jazyk.....	25
4.1.2 Struktura softwaru Wolfram Mathematica.....	25
4.1.3 Notebookový systém	25
4.1.4 Grafika.....	25
4.1.5 Náповěda.....	26
5 C JAZYK.....	27
5.1 ZÁKLADNÍ VLASTNOSTI.....	27
II PRAKTICKÁ ČÁST	28
6 EXPORT DAT V JAZYCE C A IMPORT DAT DO PROSTŘEDÍ WOLFRAM MATHEMATICA	29
6.1 EXPORT DAT DO TEXTOVÉHO SOUBORU V JAZYCE C	29
6.2 IMPORT DAT Z TEXTOVÉHO SOUBORU DO PROSTŘEDÍ WOLFRAM MATHEMATICA	30
7 ANALÝZA DAT.....	31
8 VIZUALIZACE.....	32
8.1 GRAFY TESTOVACÍCH FUNKCÍ	32
8.1.1 Graf 2D a vrstevnicový graf.....	32
8.1.2 Graf 3D.....	33
8.2 INTERPRETACE JEDINCE.....	34
8.2.1 Interpretace jednodimenzionálních jedinců	35
8.2.2 Interpretace dvojdimenzionálních jedinců	35
8.2.3 Interpretace vícedimenzionálních jedinců.....	37

8.3	ANIMACE VÝVOJE POPULACE	38
9	POPIS APLIKACE A UŽIVATELSKÉHO PROSTŘEDÍ.....	40
9.1	SPUŠTĚNÍ APLIKACE	40
9.2	ROZLOŽENÍ APLIKACE	41
9.2.1	Panel tlačítek	41
9.2.1.1	Vizualizace.....	41
9.2.1.2	Lupa	41
9.2.1.3	Export.....	42
9.2.1.4	Nová data	43
9.2.2	Panel pro výběr jedinců a jejich souřadnic	43
9.2.2.1	Část pro výběr jedince	43
9.2.2.2	Část pro výběr souřadnice.....	44
9.2.3	Ovládací panel.....	46
9.2.3.1	Jméno načteného souboru	47
9.2.3.2	Ovládací prvek animace.....	47
9.2.3.3	Posuvník pro nastavení historie	47
9.2.3.4	Přepínač druhu grafu.....	47
9.2.3.5	Rozevírací seznam pro výběr funkce.....	47
9.2.3.6	Spojnice	48
9.2.3.7	Gbest	48
9.2.3.8	Informační tabulka.....	48
	ZÁVĚR	49
	SEZNAM POUŽITÉ LITERATURY.....	50
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	51
	SEZNAM OBRÁZKŮ	52
	SEZNAM TABULEK.....	54
	SEZNAM PŘÍLOH.....	55

ÚVOD

V posledních dvou desetiletích se k řešení náročných a rozsáhlých úloh stále více využívá tzv. evolučních algoritmů, které jsou inspirované evoluční teorií. K rozvoji těchto algoritmů také přispívají stále větší pokroky v oblasti výpočetní techniky. Tyto algoritmy jsou velice výkonné a umožňují řešit velmi složité problémy efektivním způsobem. Jsou tedy velmi oblíbené v mnoha inženýrských oborech. Cílem této práce je vytvoření nástroje v prostředí Wolfram Mathematica, který umožní vizualizaci dat pocházejících z vybraných evolučních algoritmů, napsaných v jazyku C. Právě pomocí vizualizace je možné vidět, jak je daný algoritmus úspěšný při řešení problémů. První kapitola teoretické části je zaměřena na úvod do problematiky evolučních výpočetních technik a obecný popis principu evolučních algoritmů. V dalších kapitolách jsou charakterizovány vybrané testovací funkce a možnosti vizualizace. V závěru teoretické části je pak popsáno vývojové prostředí Wolfram Mathematica a jazyk C. V praktické části je v úvodu vysvětlen způsob exportování vícerozměrných polí v jazyce C a následného importování do prostředí Wolfram Mathematica. Je zde také provedena analýza importovaných dat a vysvětlen jejich význam. V neposlední řadě je tu objasněn princip vizualizace a v závěru je pak popsáno uživatelské rozhraní aplikace, která představuje vizualizační nástroj.

I. TEORETICKÁ ČÁST

1 EVOLUČNÍ VÝPOČETNÍ TECHNIKY

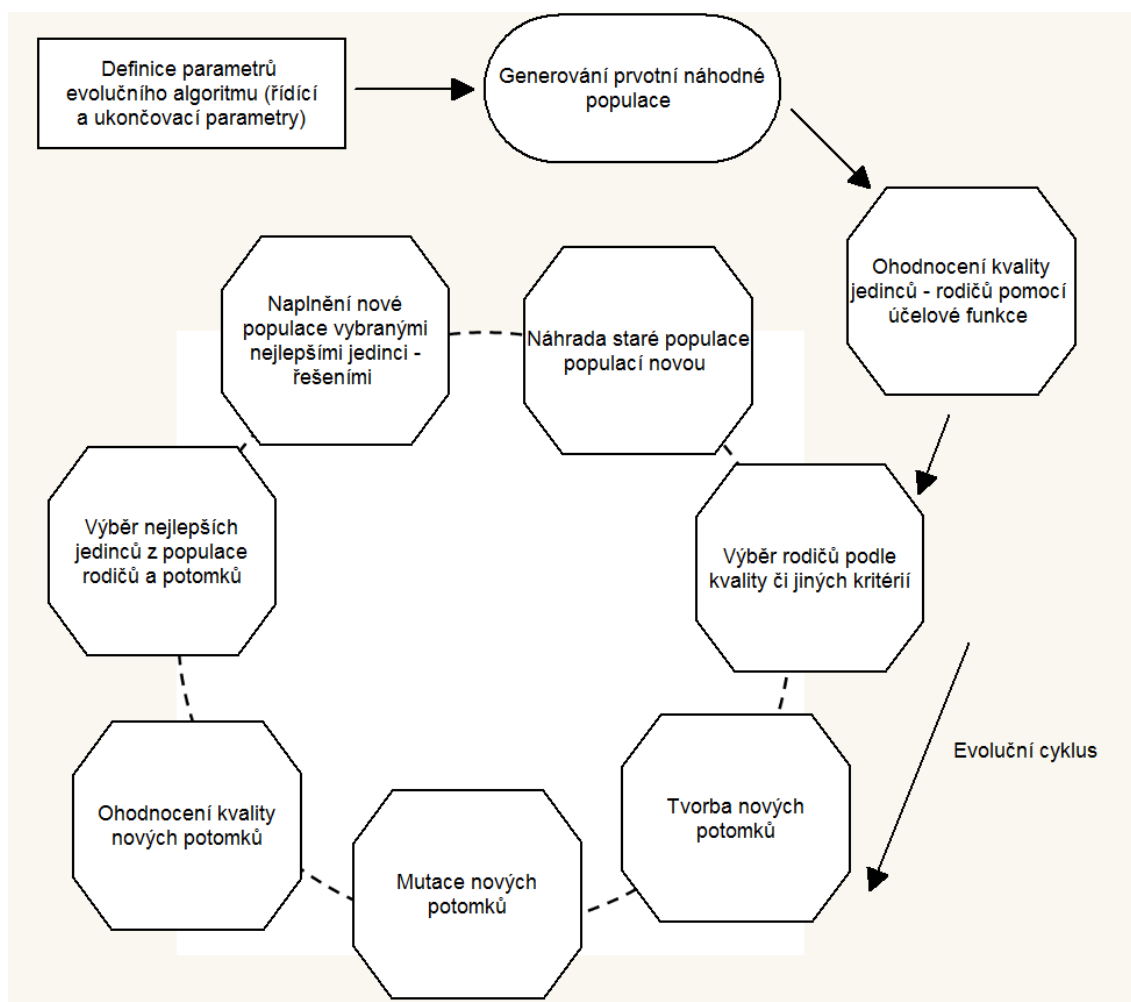
Evoluční výpočetní techniky jsou numerické algoritmy, založené na základních principech Darwinovi a Mendelovy teorie evoluce. Hlavní myšlenkou této teorie je předávání rodičovského genomu novým potomkům a následné uvolnění životního prostoru potomkům. Většinu evolučních výpočetních technik tvoří tzv. evoluční algoritmy. [1]

1.1 Princip evolučních algoritmů

Pro evoluční algoritmy je charakteristické, že pracují s populací tvořenou jedinci (kandidáty možných řešení). Tato populace se postupně vyvíjí podle evolučních principů tak, že rodiče plodí potomky, kteří podléhají při svém vzniku mutacím. Nevhodní jedinci vymírají cyklicky po generacích a uvolňují místo novým a lepším potomkům, kteří se stávají novými rodiči. To vše se děje podle přesně definovaných matematických pravidel. To, jaká pravidla a reprezentace jedinců v populaci jsou použity, pak určuje, o jaký evoluční algoritmus se jedná. V dnešní době existuje velké množství těchto algoritmů (Ant Colony Optimization, Immunology System Optimalization, Particle Swarm), přičemž žádný není univerzální, ale z principu své činnosti je každý vhodný jen pro řešení specifické třídy problémů. Touto třídou se rozumí třída problémů, na níž dává daný algoritmus alespoň uspokojivé výsledky. Pokud je však dobře známý typ problému, který se má řešit, je možné také pro jeho řešení vyvinout speciální algoritmy založené na vnitřní znalosti daného problému. Tyto algoritmy jsou téměř vždy efektivnější, než naslepo zvolené algoritmy. Použití evolučních algoritmů má smysl hlavně pro řešení velkých a komplexních problémů. Řešení evolučními algoritmy nezávisí na gradientních informacích, a tak jsou především vhodné pro problémy, kde jsou takové informace nedostupné, drahé nebo odhadnuté. [1, 3]

1.2 Obecný cyklus evolučního algoritmu

Na obr. 1 je zobrazeno schéma obecného cyklu evolučního algoritmu.



Obr. 1. Cyklus evolučního algoritmu [1]

1.3 Základní pojmy

Zde jsou zmíněny některé ze základních pojmů evolučních algoritmů.

1.3.1 Účelová funkce

Účelovou funkcí se rozumí funkce, jejíž optimalizace (nalezení extrému), povede k nalezení optimálních hodnot jejích argumentů. Na každou účelovou funkci se dá nahlížet jako na geometrický problém, u něhož je cílem najít nejnížší (minimum) či nejvyšší (ma-

ximum) bod na N rozměrné ploše. Počet dimenzí N je dán počtem argumentů účelové funkce. [2]

1.3.2 Populace

Populace může být znázorněna jako matice $N \times M$, kde sloupce představují jednotlivé jedince (aktuální řešení daného problému). Každý takový jedinec je složen z množiny argumentů účelové funkce a z hodnoty účelové funkce (fitness, vhodnost). Počet argumentů je shodný s číslem dimenze řešeného problému. Hodnota účelové funkce se samotného evolučního procesu neúčastní a udává pouze kvalitu jedince. Celá počáteční populace se generuje pomocí tzv. vzoru (specimen). [1]

$$Specimen = \{ \{Real, \{Lo, Hi\}\}, \{Integer, \{Lo, Hi\}\}, \dots, \{Real, \{Lo, Hi\}\} \}$$

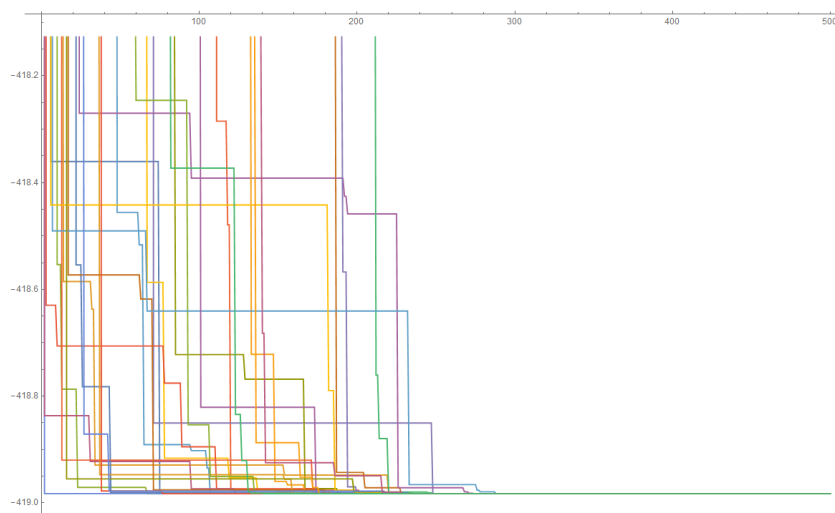
Tento vzorový jedinec je složen ze tří parametrů a to typu proměnné (reálný, celočíselný, diskretní, atd.), dolní hranice intervalu (Lo) a horní hranice intervalu (Hi). Hranice udávají interval, v němž se může hodnota parametru každého jedince pohybovat a na jejich správné volbě závisí, jestli budou nalezená řešení ještě fyzicky realizovatelná. [2]

Tab. 1. Populace jako matice $N \times M$ [1]

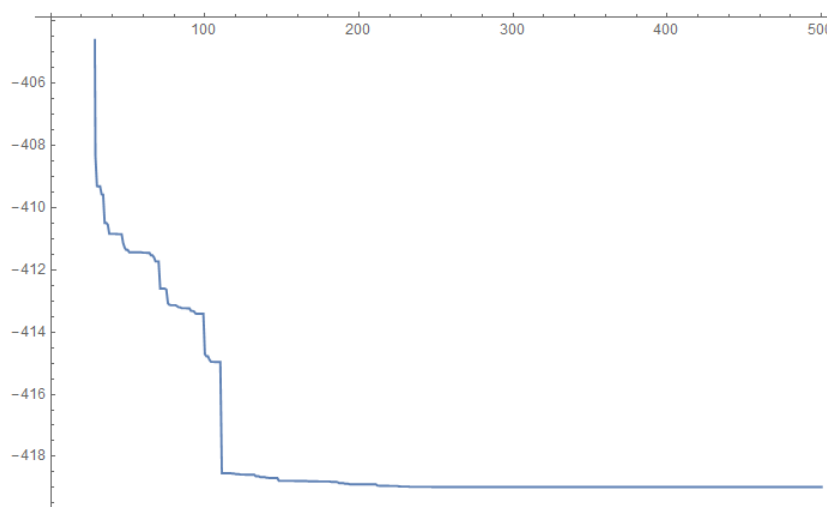
	J_1	J_2	J_3	J_4	J_M
Vhodnost	55,2	68,3	5,36	9,5	0,89
P_1	2,55	549,3	-55,36	896,5	1,89
P_2	0,25	66,2	2	-10	-2,2
P_3	-66,3	56	4	15,001	-83,66
...
P_N	259,3	-10	22,22	536,22	-42,22

Při běhu daného evolučního algoritmu dochází k tomu, že jedinci se shromažďují okolo jednoho či více extrémů. Vývoj populace musí být vždy konvergentní k lepším hodnotám, nesmí tedy vykazovat divergenci. Hledá-li se tedy minimum (maximum), vývoj musí klesat k nižším (vyšším) hodnotám. V případě, že tomu tak není, je v daném algoritmu porušen tzv. „elitismus“, který slouží jako jakýsi jednosměrný filtr, jenž propouští do nové populace pouze ta řešení, která jsou lepší či stejně dobrá jako ta ze staré populace. Zobrazení informace o tom, jak kvalitně proběhla evoluce u dané populace, se provádí pomocí tzv. historie vývoje hodnoty účelové funkce ve formě 2D grafu. Jde o sekvenci nejlepších řeše-

ní v závislosti na evolučním cyklu (generaci). Tato historie vývoje může být v rámci jedince, v takovém případě ji nazýváme pBest, nebo v rámci celé populace gBest. [1]



Obr. 2. Graf pBest populace o 30 jedincích



Obr. 3. Graf gBest populace o 30 jedincích

1.3.3 Jedinec a jeho reprezentace

Existuje několik způsobů reprezentace jedinců. Mezi nejstarší patří binární, kde je jedinec tvořen sekvencí jedniček a nul, které se říká chromozom. Tato reprezentace se používá zejména u genetických algoritmů. U binárního zápisu však mohou mutace způsobit skoko-

vou změnu čísla, což nemusí mít dobrý dopad na průběh evoluce. Tyto nerovnoměrnosti je sice možné odstranit Grayovým kódováním, avšak práce s reálnými čísly je stále výhodnější. Mezi další způsoby reprezentace patří, jak již bylo zmíněno čísla reálná, nebo celá, případně jejich kombinace. Existují ještě speciální reprezentace jako jedinec obsahující nenumerické hodnoty či jedinec reprezentovaný stromovou strukturou. [1]

2 TESTOVACÍ FUNKCE

Pro testování evolučních algoritmů se používá množina speciálních testovacích funkcí. Tato množina zahrnuje funkce různých vlastností, jako je nelinearita, různé patologie typu rovina okolo extrému atd. Vzhledem k tomu, že pro vybrané funkce jsou známy analytické vztahy, je pro většinu z nich jednoduché vypočítat pozici a hodnotu extrému pro libovolnou dimenzi. Testovací funkce se dělí na unimodální a multimodální. [2, 4]

2.1 Unimodální

Jsou takové funkce, které mají jeden extrém na daném intervalu.

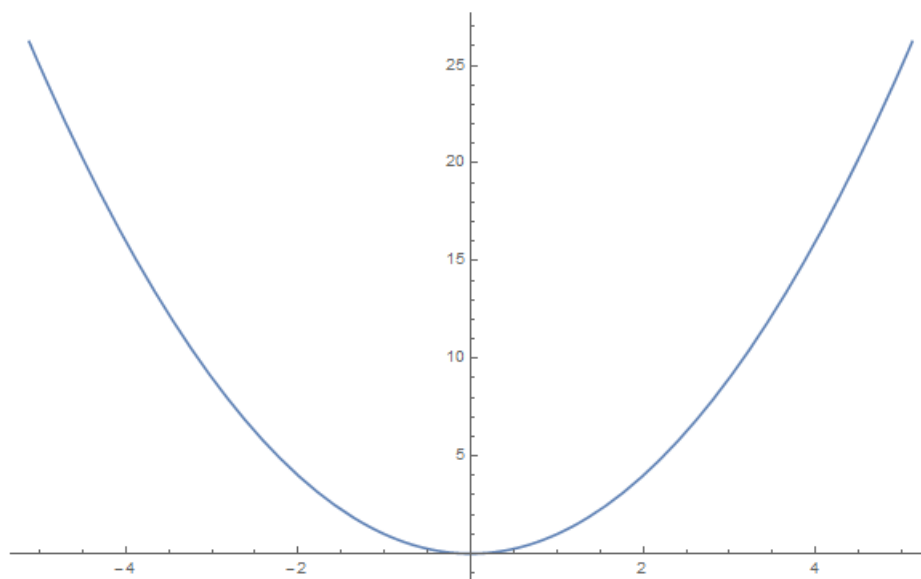
2.1.1 První De Jongova funkce

Analytický zápis

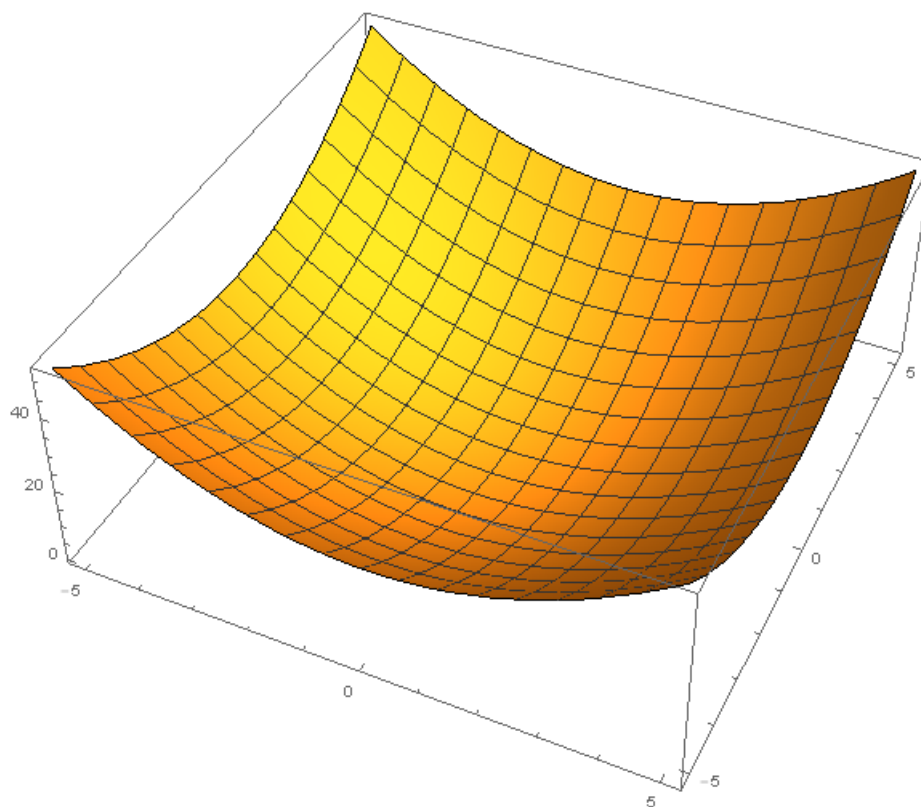
$$f(x) = \sum_{i=1}^D x_i^2$$

Globální minimum

V E_n na pozici $(x_1, x_2, \dots, x_n) = (0, 0, \dots, 0)$ o hodnotě $y = 0$



Obr. 4. První De Jongova funkce ve 2D



Obr. 5. První De Jongova funkce ve 3D

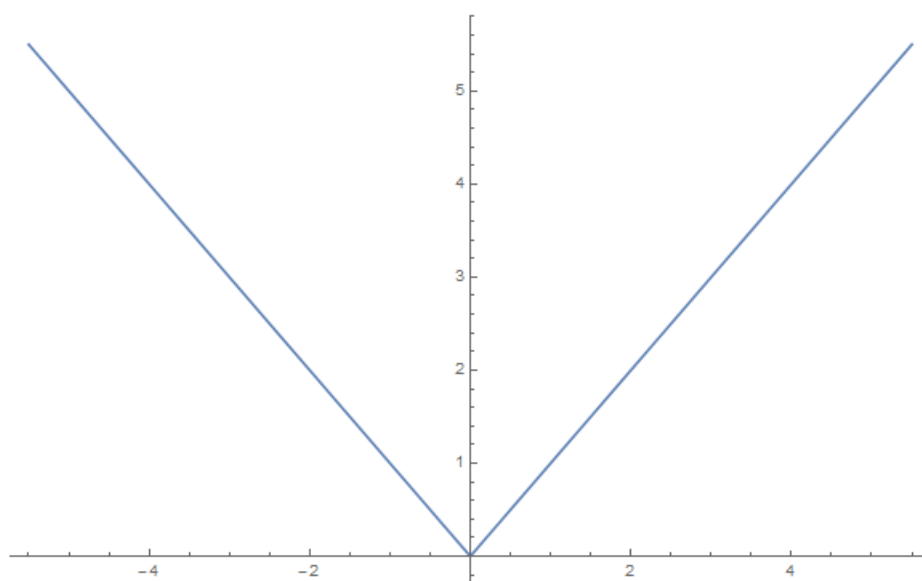
2.1.2 Třetí De Jongova funkce

Analytický zápis

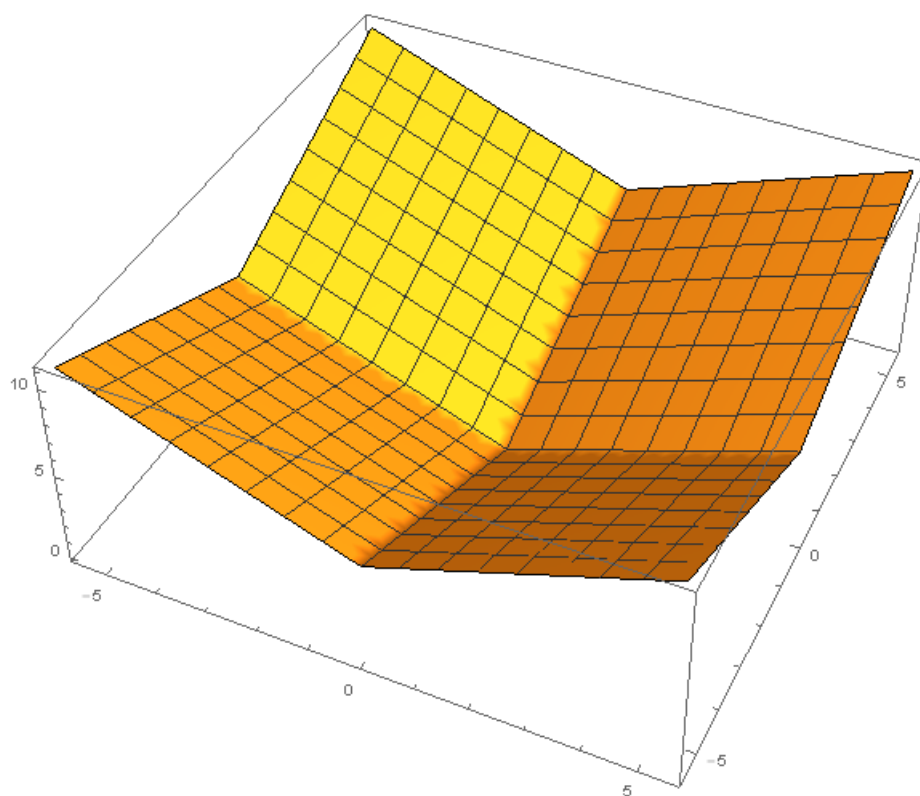
$$f(x) = \sum_{i=1}^D |x_i|$$

Globální minimum

V E_n na pozici $(x_1, x_2, \dots, x_n) = (0, 0, \dots, 0)$ o hodnotě $y = 0$



Obr. 6. Třetí De Jongova funkce ve 2D



Obr. 7. Třetí De Jongova funkce ve 3D

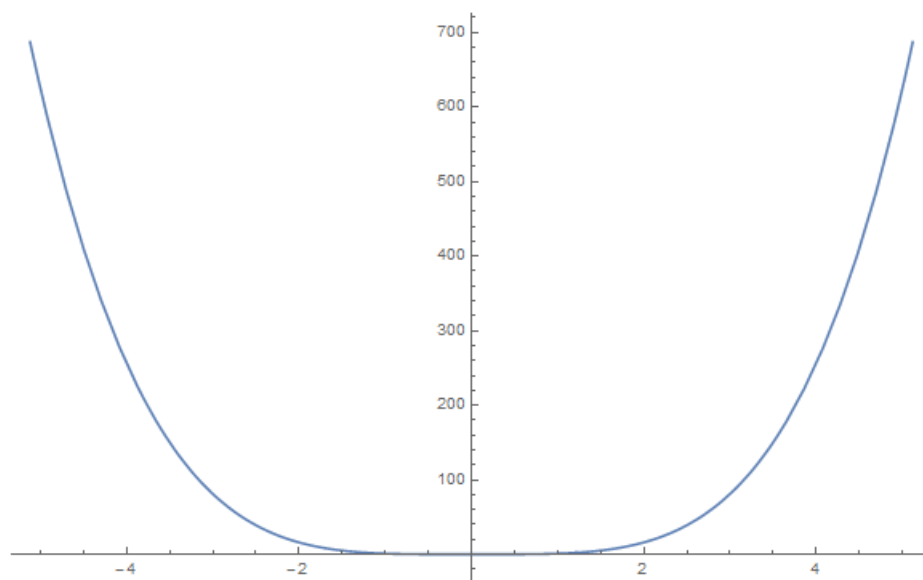
2.1.3 Čtvrtá De Jongova funkce

Analytický zápis

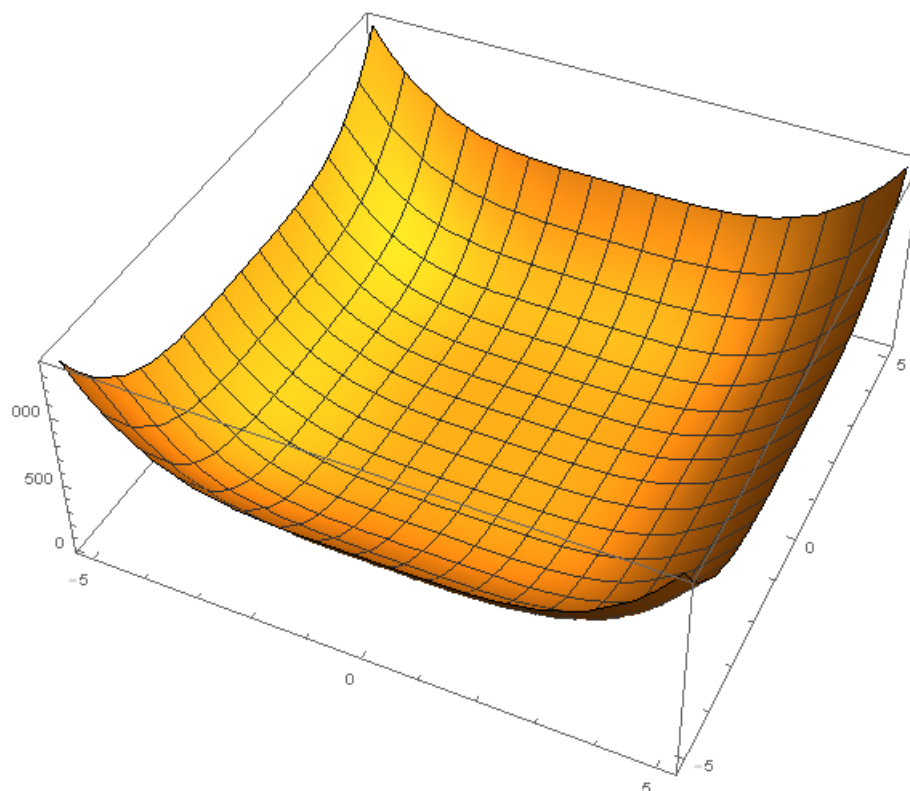
$$f(x) = \sum_{i=1}^D ix_i^4$$

Globální minimum

V E_n na pozici $(x_1, x_2, \dots, x_n) = (0, 0, \dots, 0)$ o hodnotě $y = 0$



Obr. 8. Čtvrtá De Jongova funkce ve 2D



Obr. 9. Čtvrtá De Jongova funkce ve 3D

2.2 Multimodální funkce

Jsou funkce s více, než jedním extrémem na daném intervalu.

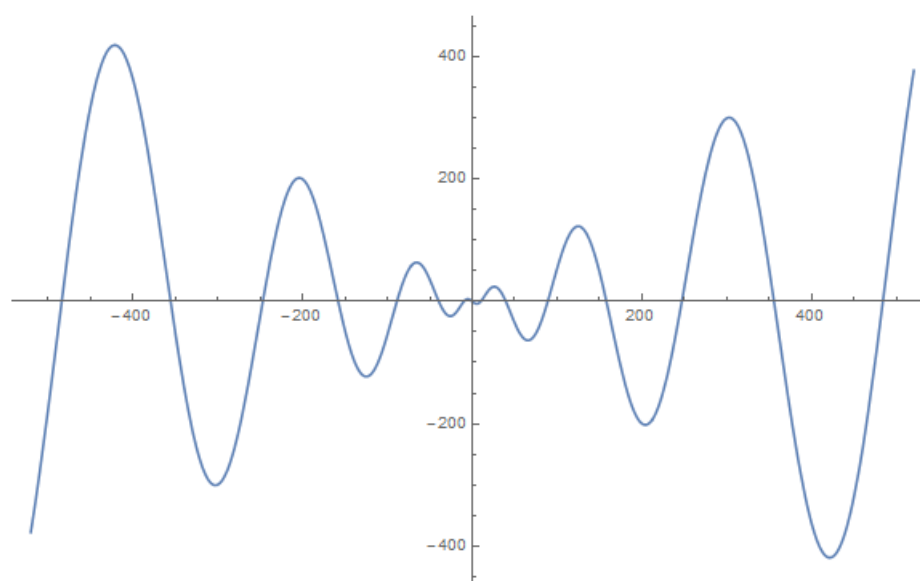
2.2.1 Schwefelova funkce

Analytický zápis

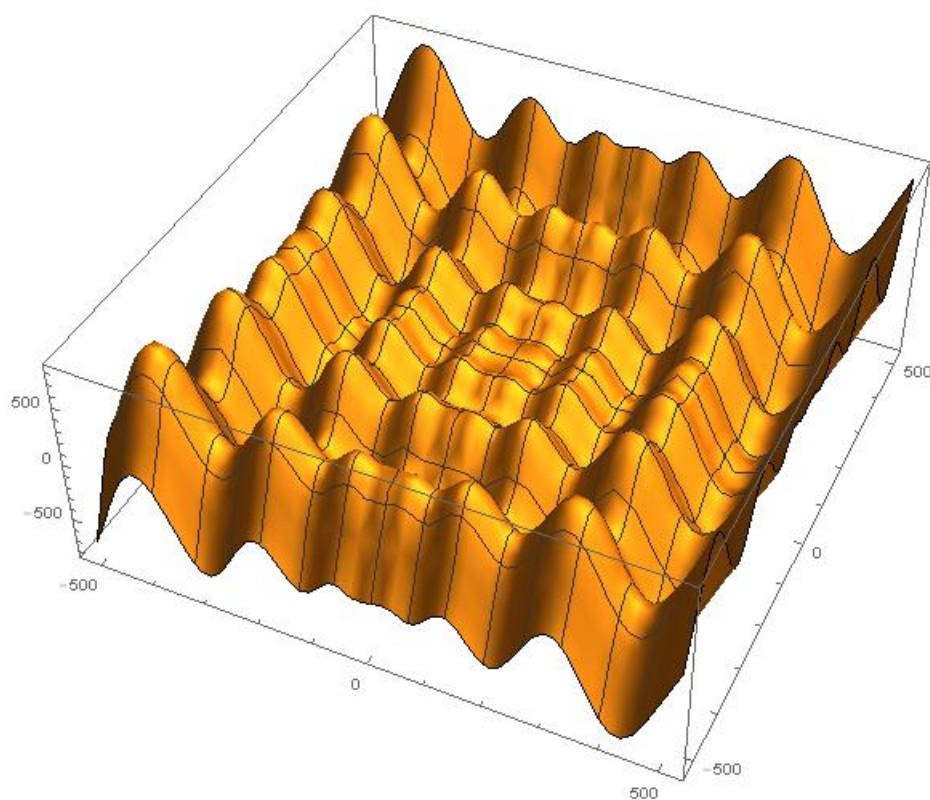
$$\sum_{i=1}^D -x_i \sin(\sqrt{|x_i|})$$

Globální minimum

V E_n na pozici $(x_1, x_2, \dots, x_n) = (420,969; \dots; 420,969)$ o hodnotě $y = -418,983 \times n$



Obr. 10. Schwefelova funkce ve 2D



Obr. 11. Schwefelova funkce ve 3D

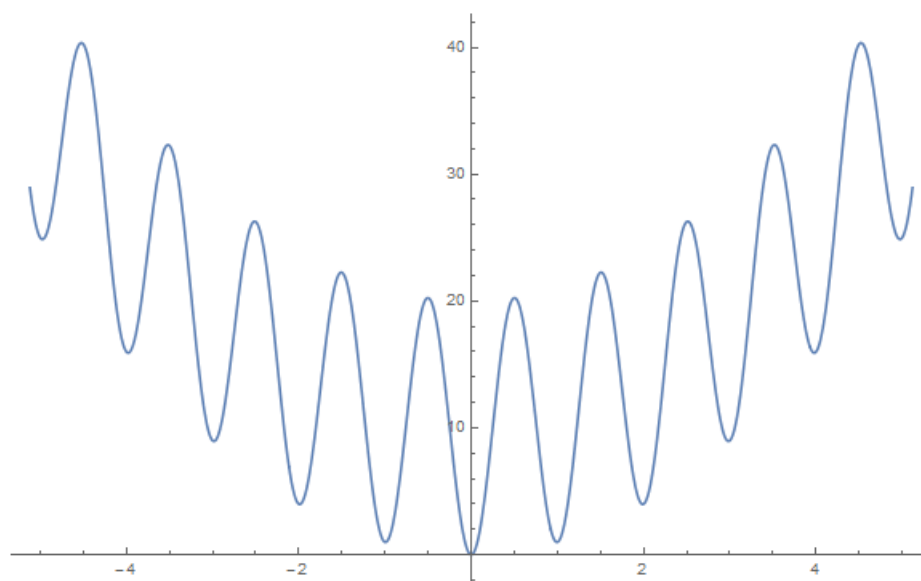
2.2.2 Rastriginova funkce

Analytický zápis

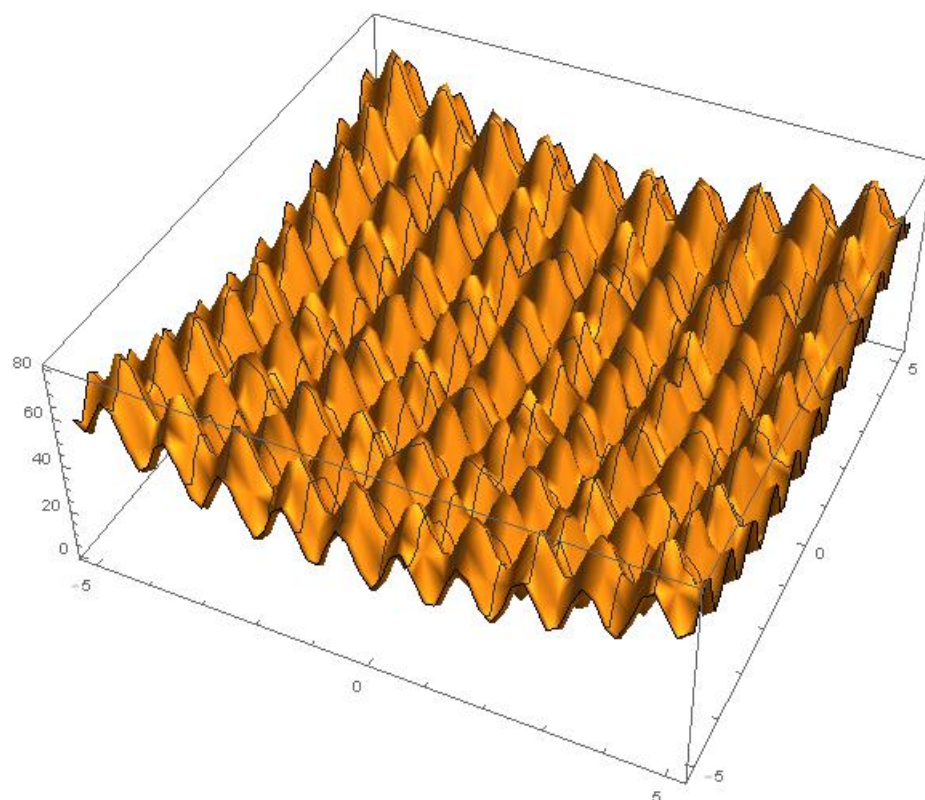
$$2D \sum_{i=1}^D x_i^2 - 10 \cos(2\pi x_i)$$

Globální minimum

V E_n na pozici $(x_1, x_2, \dots, x_n) = (0, 0, \dots, 0)$ o hodnotě $y = -200 \times n$



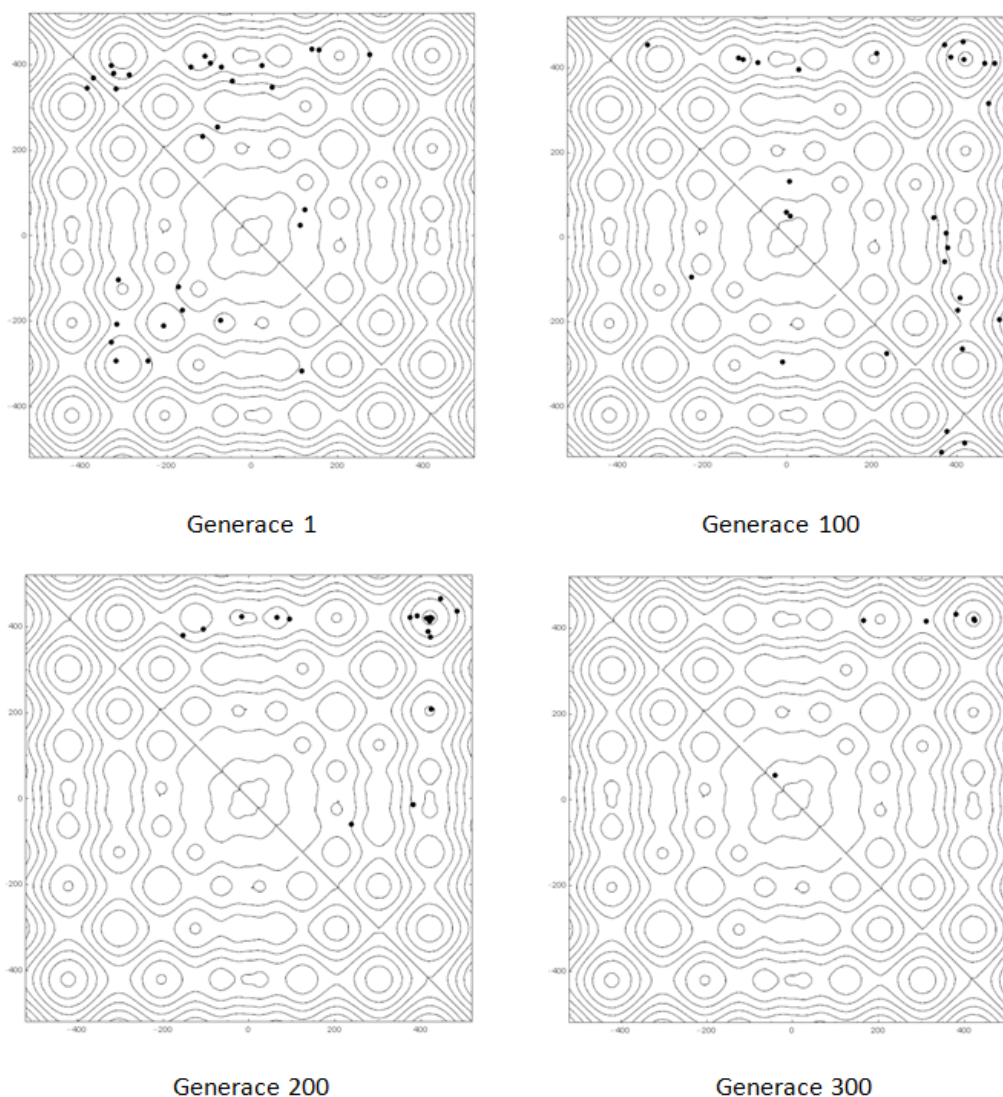
Obr. 12. Rastriginova funkce ve 2D



Obr. 13. Rastriginova funkce ve 3D

3 VIZUALIZACE

Vizualizace vývoje populace jednotlivých evolučních algoritmů se provádí pomocí množiny 2D, vrstevnicových a 3D grafů vybraných testovacích funkcí, kde jednotliví jedinci z populace jsou symbolizováni tečkami. Vizualizace pomocí 2D grafu se používá v případě, že je populace složena z jednodimenzionálních jedinců. Populace s dvojdimenzionálními jedinci může být vizualizována pomocí 3D či vrstevnicových grafů. Daleko obtížnější je to ovšem s populací, která je složena ze tří a vícedimenzionálních jedinců a to z důvodu, že člověk si nedovede představit více, jak třírozměrné geometrické konstrukce. V dnešní době se to často řeší pomocí různých technik pro zobrazení objektů z N – rozměrného prostoru do K – rozměrného prostoru, kde K je rovno dvěma nebo třem.



Obr. 14. Konvergence populace o třiceti dvojdimenzionálních jedincích ke globálnímu extrému v jednotlivých evolučních cyklech (generacích)

4 WOLFRAM MATHEMATICA

Software Mathematica je prostředí, ve kterém jsou integrovány nástroje pro numerickou a symbolickou matematiku, grafický a dokumentační systém. Zajišťuje také pokročilé propojení s dalšími aplikacemi. Vznik softwaru je spojen se Stephenem Wolframem, který založil firmu Wolfram Research, ve které v roce 1988 vznikla první verze produktu. Vývoj stále probíhá a v současnosti je k dispozici již verze 10. 1. [5, 8]

4.1 Základní vlastnosti

Software Wolfram Mathematica má několik specifických vlastností.

4.1.1 Programovací jazyk

V Mathematice se programuje v symbolickém jazyku, který byl navržen tak, aby byl co nejobecnější. Při programování není potřeba předeklarovat typ proměnné, dimenzi matice, překládat program ani přímé řízení paměti. Symbolické programování umožňuje reprezentovat data, funkce, grafy, programy i celý dokument jako symbolický výraz. Např. výraz $f[x]$ může být matematickou funkcí či grafikou, programem i celým dokumentem. [5, 8]

4.1.2 Struktura softwaru Wolfram Mathematica

Mathematica je složena ze dvou částí a těmi jsou jádro (kernel) a font-end. Jádro plní funkci tzv. interpretu což je u interpretovaných jazyků speciální program, který vykonává zdrojový kód. Font-end je prostředí do kterého uživatel zadává veškerý kód. Tento kód je po potvrzení vyslán do kernelu, kde je vykonán a výsledek se zobrazí opět uživateli ve font-endu. Font-end neboli prostředí je tvořen tzv. Notebooky. [8]

4.1.3 Notebookový systém

Notebook kombinuje textový procesor se zřetelně definovanou představou "buněk", které jsou seřazeny svisle v rozbalovacím okně jako odstavce textu. Buňky vizuálně a funkčně oddělují text na vstupy, výstupy, grafiku atd. [5]

4.1.4 Grafika

Součástí programu je velké množství vestavěných grafických vzorů pro vizualizaci výsledků, 2D, 3D, vrstevnicové grafy, grafy hustoty atd. Uživatel má také možnost si vestavěné

vzory přizpůsobit či vytvořit vlastní. Velmi jednoduše se zde dají také vytvořit animace. [5]

4.1.5 Nápořěda

Nápořěda zahrnuje kompletní dokumentaci pro všechny funkce v prostředí Mathematica. Jsou zde popsány vlastnosti všech funkcí, jejich možné vstupy, parametry a samozřějmě také příklady použití. Jednotlivé funkce jsou navíc sdružovány ve skupinách podle účelu použití. V případě, že uživatel hledá například funkci pro sčítání čísel, najde ji ve skupině Aritmetické funkce. [8]

5 C JAZYK

C jazyk je jeden z nejrozšířenějších programovacích jazyků. Jazyk byl navržen a implementován Kenem Thompsonem a Dennisem Ritchiem pod operačním systémem UNIX v počátku sedmdesátých let. C se ovšem na UNIX ani na jiný operační systém či konkrétní počítač nijak neváže. [6, 9]

5.1 Základní vlastnosti

Jedná se o univerzální programovací jazyk nízké úrovně, což znamená, že pracuje pouze se standardními datovými typy, jako jsou znaky, celá a reálná čísla atd. Má velmi úsporné vyjadřování, je strukturovaný a má velký soubor operátorů. Není navíc specializovaný na jednu oblast použití. Patří mezi kompilované jazyky, což znamená, že programy jsou kompilátorem a sestavovacím programem převedeny do strojového kódu a strojový kód je přímo vykonáván procesorem. Kompilovaný program je z podstaty věci rychlejší, než interpretovaný, který je vykonáván pomocí interpretu. Samotný jazyk je poměrně chudý a složitější věci se řeší voláním funkcí z knihoven. Součástí normy c je i standardní knihovna obsahující nejčastěji používané funkce jako jsou terminálový vstup a výstup, základní operace se soubory, práce s pamětí, řetězci a matematické funkce. [6, 7, 9]

II. PRAKTICKÁ ČÁST

6 EXPORT DAT V JAZYCE C A IMPORT DAT DO PROSTŘEDÍ WOLFRAM MATHEMATICA

Vzhledem k tomu, že evoluční algoritmy jsou poměrně náročné na výpočetní výkon počítače, je dobré, aby byly naprogramovány co nejefektivněji ve výkonném programovacím jazyku. V tomto případě jde o jazyk C, který jakožto kompilovaný jazyk je velmi rychlý. Vzhledem k potřebě vizualizace dat je nutné exportovat data z programu napsaného v C jazyce a následně je importovat do prostředí Wolfram Mathematica. Oba jazyky umožňují jak export dat do textového souboru tak import dat z textového souboru, takže textový soubor poslouží jako „prostředník“ mezi oběma jazyky.

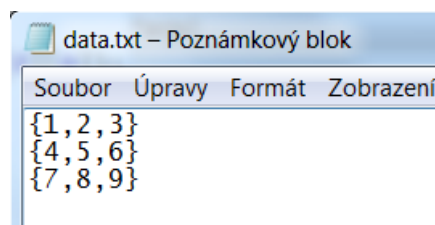
6.1 Export dat do textového souboru v jazyce C

V jazyce C existuje pro zápis dat do textového souboru funkce *fprintf*. Jako příklad použití bylo definováno dvojrozměrné pole (matice) velikosti 3x3.

```
int main(int argc, char **argv)
{
    int pole[3][3]= {{1,2,3},{4,5,6},{7,8,9}};
    int i,j;
    FILE *soubor;
    int pocet_radku=sizeof(pole)/ sizeof(pole[0]);
    int pocet_sloupcu=sizeof(pole[0])/ sizeof(pole[0][0]);
    soubor = fopen("data.txt","w");
    for (j = 0; j < pocet_radku; j++) {
        fprintf(soubor, "{");
        for (i = 0; i < pocet_sloupcu; i++) {
            fprintf(soubor, "%d",pole[j][i] );
            if(i<(pocet_sloupcu-1))
                fprintf(soubor, ",");
        }
        fprintf(soubor, "}\n");
    }
    fclose(soubor);
    return 0;
}
```

Obr. 15. Export dat v jazyce C

Zápis do souboru probíhá po jednotlivých číslech tak, že řádky matice jsou uzavřeny ve složených závorkách a odděleny znakem pro nový řádek. Obsah souboru je pak vidět na obr. 16.



Obr. 16. Formát dat v textovém souboru

6.2 Import dat z textového souboru do prostředí Wolfram Mathematica

Pro importování dat do prostředí Wolfram Mathematica se používá funkce *Import*. Tato funkce je univerzální a je možné ji například použít i pro import obrázků. Funkce *Import* načte jednotlivé řádky matice jako textové řetězce oddělené čárkami a uložené ve složených závorkách. Celé pole pak stačí jen převést pomocí funkce *ToExpression* na číselné výrazy.

```
In[19]:= pole = Import["data.txt", "List"]  
  
In[20]:= {"{1,2,3}", "{4,5,6}", "{7,8,9}" }  
  
In[21]:= ToExpression[pole]  
Out[21]= {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}  
  
In[22]:= ToExpression[pole] // MatrixForm  
Out[22]/MatrixForm=  

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

```

Obr. 17. Import dat v prostředí Wolfram Mathematica

7 ANALÝZA DAT

Data z evolučních algoritmů jsou do Mathematici importována v následujícím formátu na obr. 18.

```
{ { {25, {0.1, 0.6, 0.8}} , {45, {0.4, 4.2, 0.1}} } ,  
  { {18, {0.6, 0.2, 0.4}} , {50, {0.1, 4.4, 0.4}} } ,  
  { {30, {0.3, 0.1, 0.5}} , {40, {0.2, 4.6, 0.3}} } }
```

Obr. 18. Analýza dat

Pro lepší představu jsou jednotlivé části na obr. 18 barevně odlišeny. Černé závorky představují množinu jednotlivých generací vybrané populace seřazených od nejstarší po nejmladší. V tomto konkrétním příkladu se jedná o tři generace, které symbolizují zelené závorky. Každá jedna generace je dále složena z jednotlivých jedinců. Jedinci jsou v tomto příkladu dva v závorkách označených červenou barvou. Každý jedinec je pak složen z hodnoty účelové funkce (zvýrazněna hnědou barvou) a množiny argumentů účelové funkce (zvýrazněna modrými závorkami). Počet argumentů účelové funkce je závislý na dimenzi problému, pro tento příklad je dimenze rovna třem.

8 VIZUALIZACE

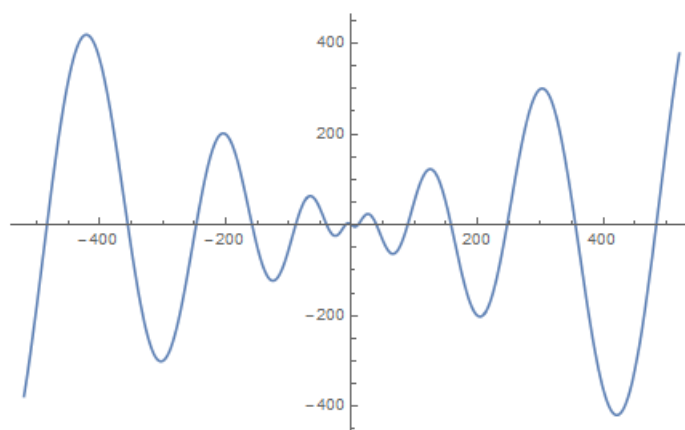
Jak již bylo zmíněno v teoretické části, vizualizace se provádí pomocí grafů testovacích funkcí, na kterých se pohybují tečky reprezentující jedince.

8.1 Grafy testovacích funkcí

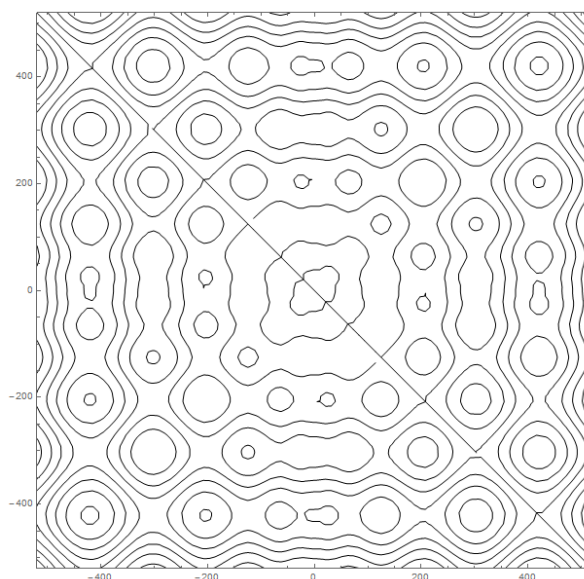
Vizualizace probíhá prostřednictvím tří typů grafů, jimiž jsou 2D graf, 3D graf a vrstevnicový graf.

8.1.1 Graf 2D a vrstevnicový graf

Vytvoření 2D a vrstevnicového grafu se v Mathematicce provádí pomocí funkcí *Plot* a *ContourPlot*. Do argumentu obou funkcí stačí zadat funkční předpis a meze, v jakých bude funkce vykreslena.



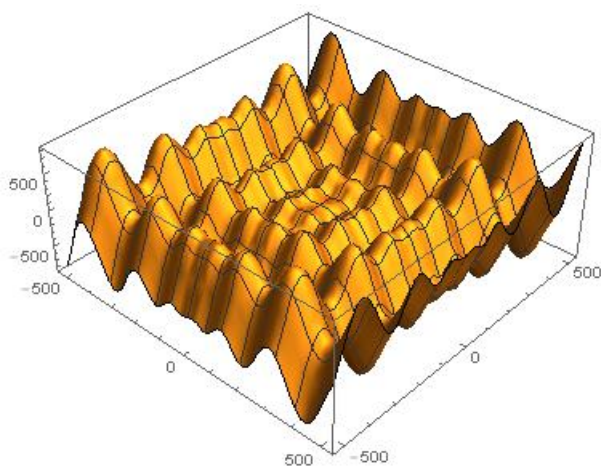
Obr. 19. 2D graf testovací funkce



Obr. 20. vrstevnicový graf testovací funkce

8.1.2 Graf 3D

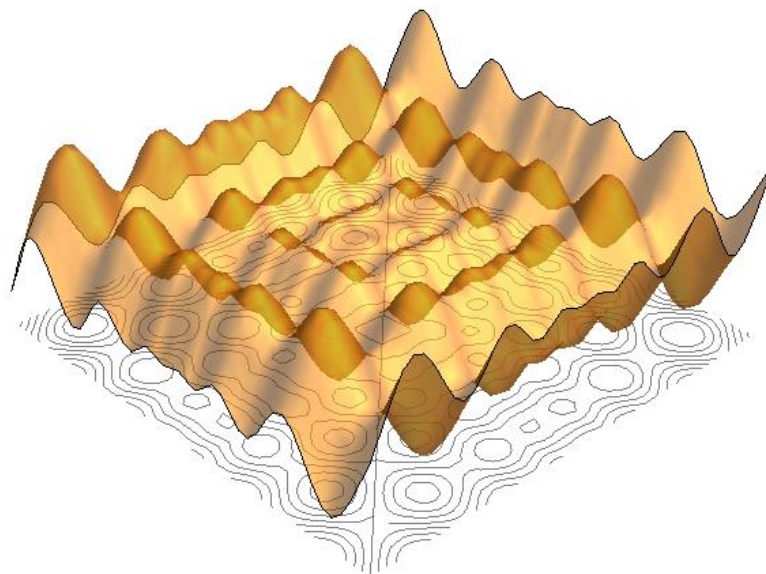
Graf 3D se vytváří pomocí funkce *Plot3D*, která funguje na stejném principu jako předchozí dvě funkce pro 2D zobrazení.



Obr. 21. 3D graf testovací funkce

Vzhledem k tomu, že by vizualizace pomocí grafu v této podobě byla značně nepřehledná, bylo nutné toto zobrazení upravit. Proto byla zvolena vizualizace pomocí průhledného 3D grafu, pod kterým se nachází taktéž průhledný vrstevnicový graf. Protože Mathematica na takový typ grafu žádnou funkci typu *Plot* nenabízí, byl graf vytvořen pomocí funkce *Show*, která dokáže zkombinovat několik grafických 2D nebo 3D objektů dohromady. Nejdříve

bylo ovšem nutné převést vrstevnicový graf do 3D prostoru. To bylo provedeno tak, že byly zjištěny x -ové a y -ové souřadnice vrstevnic na 2D vrstevnicovém grafu a k těmto souřadnicím byla přidána z -ová souřadnice symbolizující vzdálenost vrstevnicového grafu od 3D grafu. Pomocí těchto souřadnic pak bylo možné vykreslit konturu v 3D prostoru pomocí funkce *GraphicsComplex*, a následně ji zkombinovat pomocí zmíněné funkce *Show* s 3D grafem.



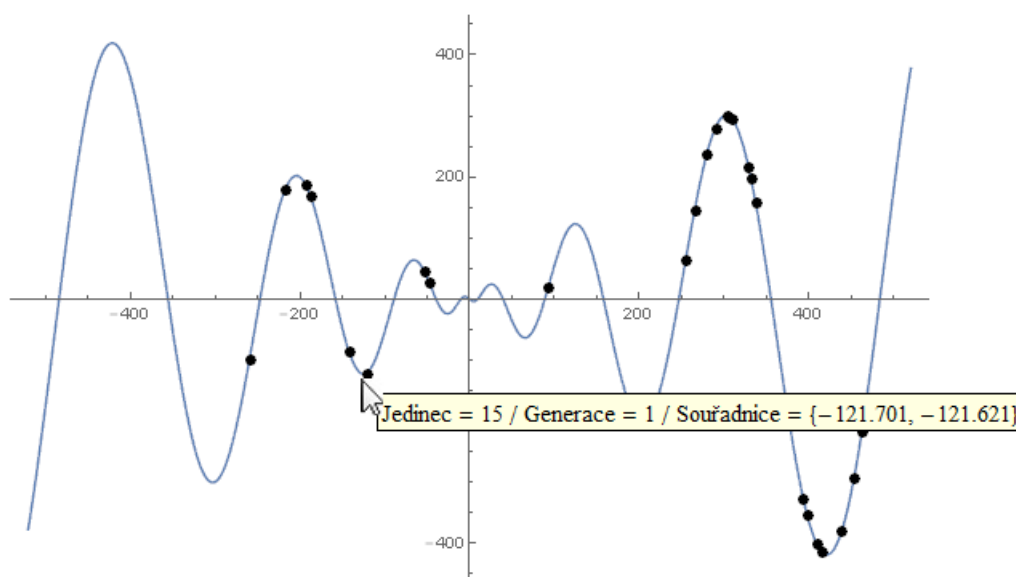
Obr. 22. Spojení 3D a vrstevnicového grafu

8.2 Interpretace jedince

V programu Mathematica existuje funkce *Point*, která vytvoří bod, nebo množinu bodů o souřadnicích $\{x, y\}$ ve 2D zobrazení a $\{x, y, z\}$ v 3D. Jako souřadnice se dosazují v závislosti na dimenzi problému, buď jen samotné argumenty účelové funkce jedince, nebo argumenty i s hodnotou účelové funkce. Tyto body vytvořené funkcí *Point* jsou pak následně vykresleny do 2D či 3D pomocí funkcí *Graphics* (*Graphics3D*) a zkombinovány pomocí funkce *Show* s příslušným grafem testovací funkce.

8.2.1 Interpretace jednodimenzionálních jedinců

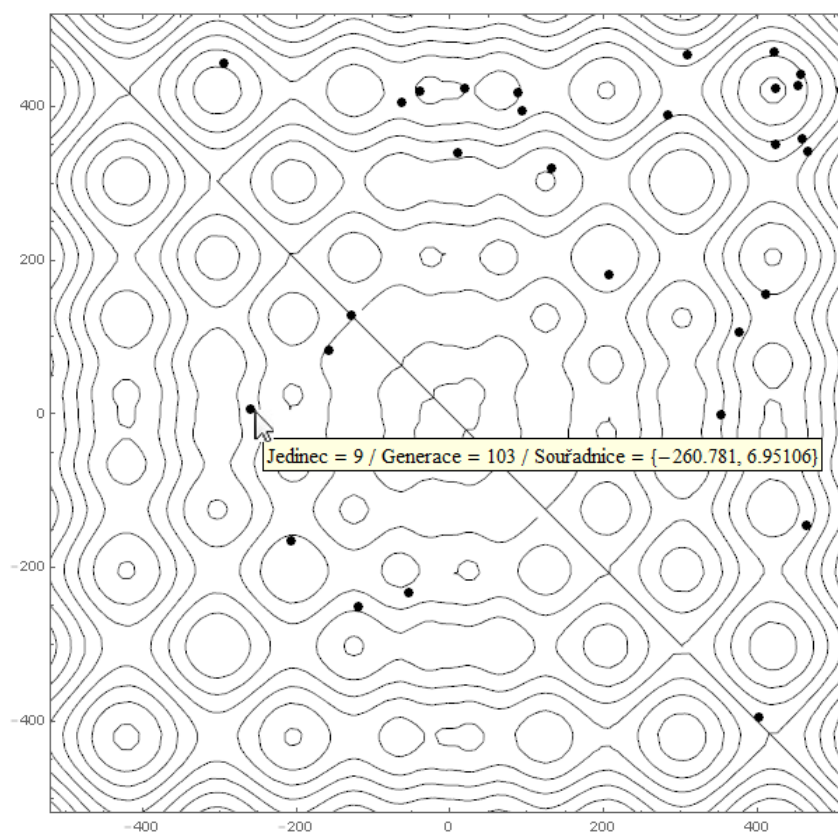
V případě že se jedná o jednodimenzionální problém a jedinec je např. ve tvaru $\{25, \{0.1\}\}$, je jako x -ová souřadnice funkce *Point* použita hodnota argumentu účelové funkce 0.1 a jako y -ová souřadnice hodnota účelové funkce 25. Na obr. 23 se nachází vizualizace první generace populace jednodimenzionálních jedinců pomocí 2D grafu Schwefelovy funkce. Jak je také z obrázku vidět, u každého jedince je možné zobrazit jeho popisek po najetí kurzorem myši na příslušného jedince. Na tento popisek existuje v Mathematice funkce *Tooltip*. Z popisku na obrázku je pak vidět, že se jedná o patnáctého jedince z první generace populace se souřadnicemi jeho polohy $\{-121.701, -121.621\}$.



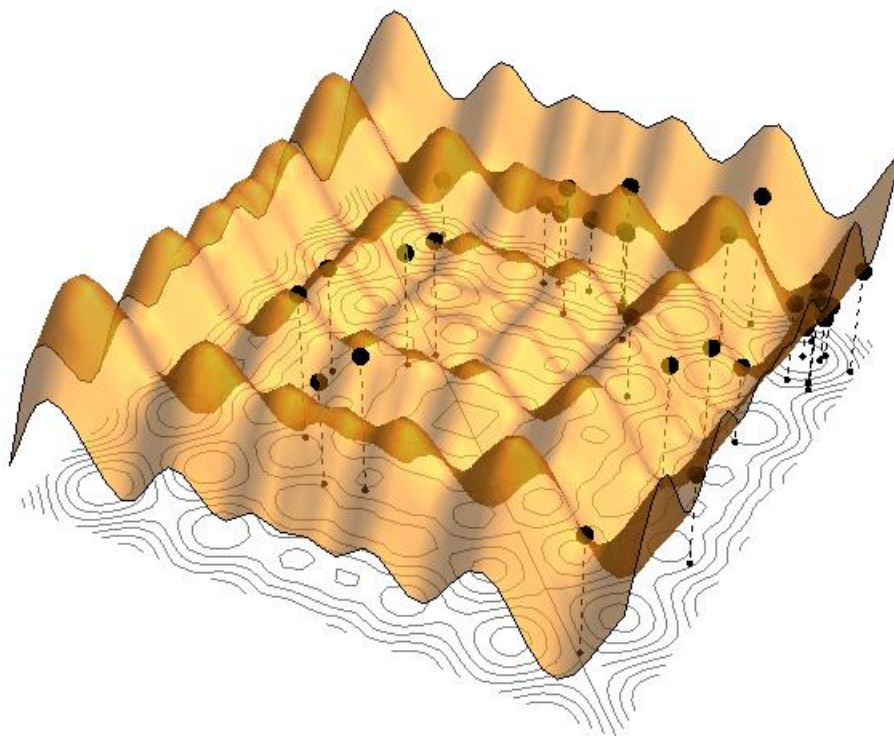
Obr. 23. Zobrazení jednodimenzionálních jedinců na 2D grafu

8.2.2 Interpretace dvojdimenzionálních jedinců

Jedná-li se o problém dvojdimenzionální a jedinec je např. ve tvaru $\{25, \{0.1, 0.2\}\}$, jsou dvě možnosti vizualizace a to pomocí 2D vrstevnicového grafu, nebo 3D grafu. U první možnosti se za obě souřadnice funkce *Point* dosadí argumenty účelové funkce. U 3D grafu se pak za y -ovou a x -ovou souřadnici dosadí argumenty funkce a za z -tovou souřadnici hodnota účelové funkce. Jak je vidět na obr. 24 u vrstevnicového grafu je opět možné zobrazit popisek u každého jedince, avšak u 3D grafu se z důvodu překrývání jedinců s grafem tato možnost již nenachází.



Obr. 24. Zobrazení dvojdimenzionálních jedinců na vrstevnicovém grafu

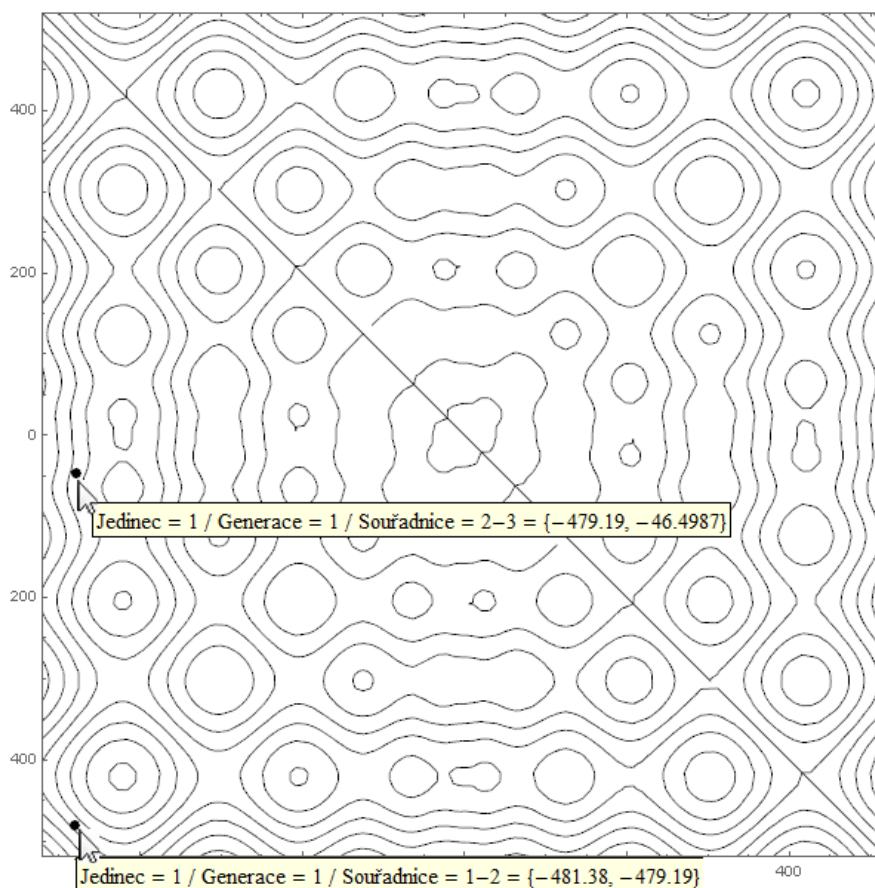


Obr. 25. Zobrazení dvojdimenzionálních jedinců na 3D grafu

V rámci zvýšení přehlednosti zobrazení jedinců na 3D grafu je každý jedinec symbolizován dvěma tečkami, z nichž první (větší) se pohybuje po 3D grafu a druhá kopíruje pohyb první tečky na vrstevnicovém grafu. Obě jsou pak spojeny přerušovanou čarou, aby šlo jasně vidět, že jde o jednoho jedince.

8.2.3 Interpretace vícedimenzionálních jedinců

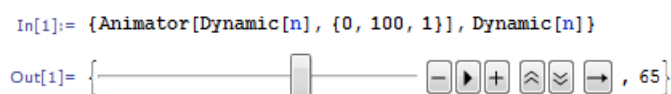
V případě že jedinec má více jak dva argumenty účelové funkce např. $\{96.35, \{-481.38, -479.18, -46.49\}\}$, dojde k rozdělení těchto argumentů do skupin po dvou hodnotách pomocí funkce *Partition*. Jedinec se třemi argumenty má pak následující tvar $\{96.35, \{\{-481.38, -479.18\}, \{-479.18, -46.49\}\}\}$. Toto řešení umožní zobrazení vícedimenzionálního jedince ve 2D vrstevnicovém grafu. Výsledný jedinec v tomto příkladu je pak symbolizován dvěma body (tečkami), z nichž první bod má souřadnice $\{x, y\} = \{-481.38, -479.18\}$ a druhý $\{x, y\} = \{-479.18, -46.49\}$.



Obr. 26. Zobrazení třídímenzionálního jedince na vrstevnicovém grafu

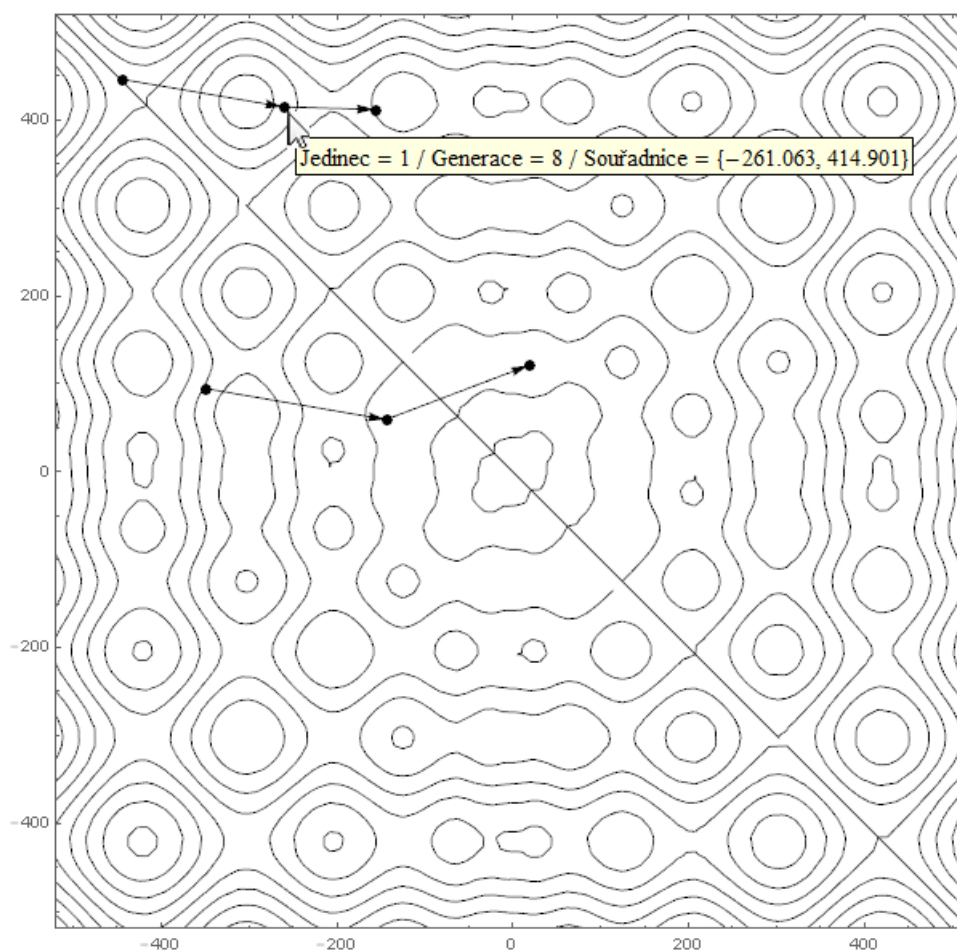
8.3 Animace vývoje populace

Celá animace vývoje populace je založena na práci s polem hodnot a na tzv. animátoru, což je ovládací prvek, který projíždí nastaveným intervalem po nadefinovaných krocích. Na obr. 27 je zobrazen animátor, který se pohybuje na intervalu velikosti 0 až 100 po krocích velikosti 1. Aktuální hodnota animátoru je ukládána do proměnné, v tomto případě jde o proměnnou n , ve které je uložena hodnota 65.



Obr. 27. Animátor

K jednotlivým prvkům pole se v Mathematice přistupuje podobným způsobem jako v ostatních jazycích a to pomocí indexu v dvojitéch hranatých závorkách. Není tedy problém dosadit jako index proměnnou animátoru tak, že se budou postupně zobrazovat jednotlivé generace populace za sebou. Na tomto principu je pak založena i možnost výběru jedinců, kteří mají být zobrazeni na jednotlivých grafech a historie vývoje jedinců. V tomto případě byly samozřejmě použity i další funkce na práci s polem jako např. funkce *Table*, která slouží k vytváření polí, nebo *Take* sloužící k výběru pouze některých prvků pole.



Obr. 28. Historie vývoje vybraných jedinců

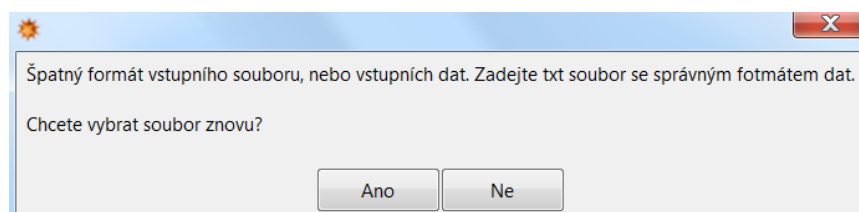
Na obr. 28 jsou vidět dva vybraní jedinci z populace s nastavenou historií na hodnotu dva, to znamená, že je zobrazen pohyb (vývoj) jedinců v posledních třech generacích. Šipky nebo také spojnice byly vytvořeny pro větší přehlednost pomocí funkce *Arrow*, která je podobná funkci *Point* s tím rozdílem, že vyžaduje souřadnice počátečního a konečného bodu. Šipky znázorňují směr pohybu jedince od nejstarší generace k nejmladší. Z obrázku je tedy patrné, že oba jedinci se vyvíjí v lepší jedince, kteří jsou blíže ke globálnímu extrému funkce, který se nachází v pravém horním rohu.

9 POPIS APLIKACE A UŽIVATELSKÉHO PROSTŘEDÍ

Celá aplikace je vytvořena pomocí funkce *Manipulate*, která umožňuje snadné vytvoření uživatelského prostředí s různými ovládacími prvky.

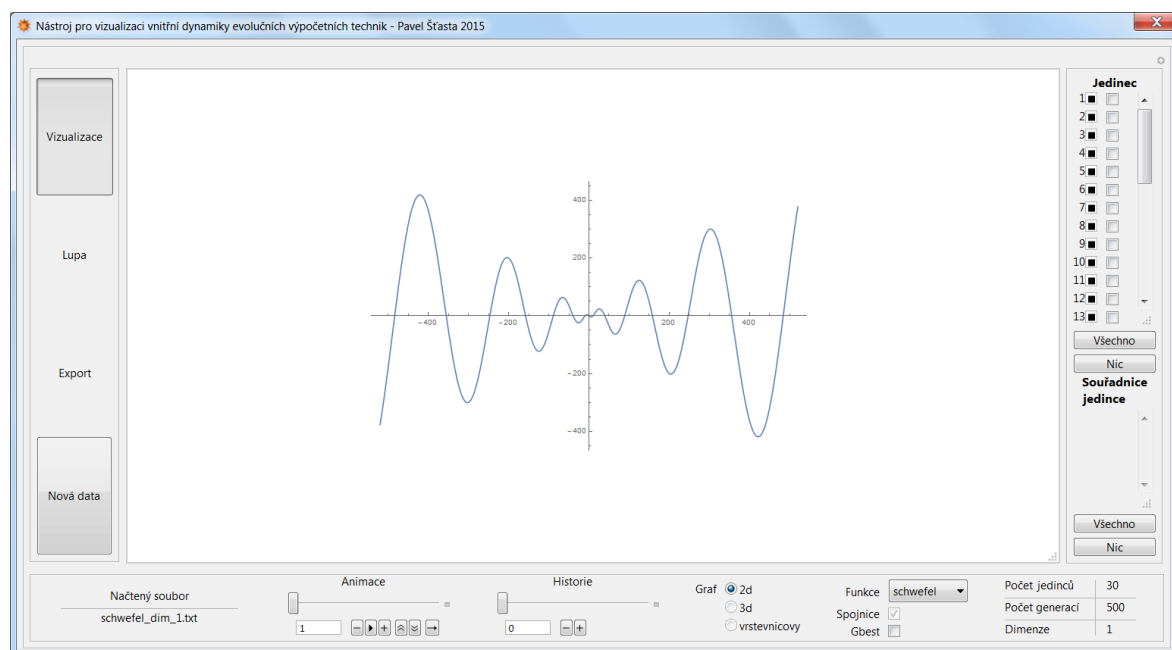
9.1 Spuštění aplikace

Po vyhodnocení zdrojového kódu v prostředí Wolfram Mathematica je uživatel vyzván ke zvolení textového souboru se vstupními daty. V případě, že uživatel zvolí špatný formát souboru, soubor se špatným formátem dat, nebo nezvolí žádný soubor bude upozorněn dialogovým oknem na obr. 29.



Obr. 29. Dialogové okno při špatném formátu souboru

Po zvolení správného datového souboru se uživateli po načtení, které může trvat řádově desítky sekund v závislosti na počtu dat vstupního souboru, verzi softwaru Mathematica a hardwaru počítače, zobrazí v dialogovém okně samotná aplikace (obr. 30).



Obr. 30. Dialogové okno s aplikací

9.2 Rozložení aplikace

Aplikace je složena ze tří panelů a vizualizace uprostřed. Panel tlačítek umístěný vlevo se používá pro přepínání mezi jednotlivými nástroji a k načítání dat. Panel vpravo slouží pro výběr jedinců a jejich souřadnic. Dole se pak nachází ovládací panel sloužící k ovládání a nastavení vizualizace.

9.2.1 Panel tlačítek

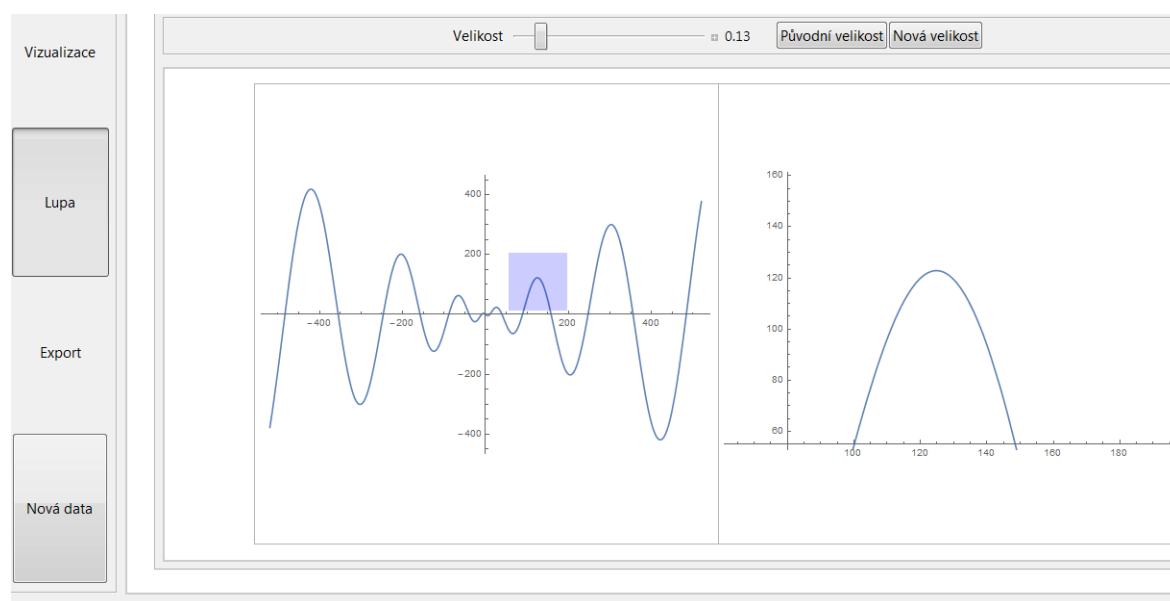
Obsahuje tři tlačítka pro přepínání mezi vizualizací, nástrojem lupa, možností exportu a tlačítko pro načtení nových dat.

9.2.1.1 Vizualizace

Vizualizace byla již popsána v kapitole 8.

9.2.1.2 Lupa

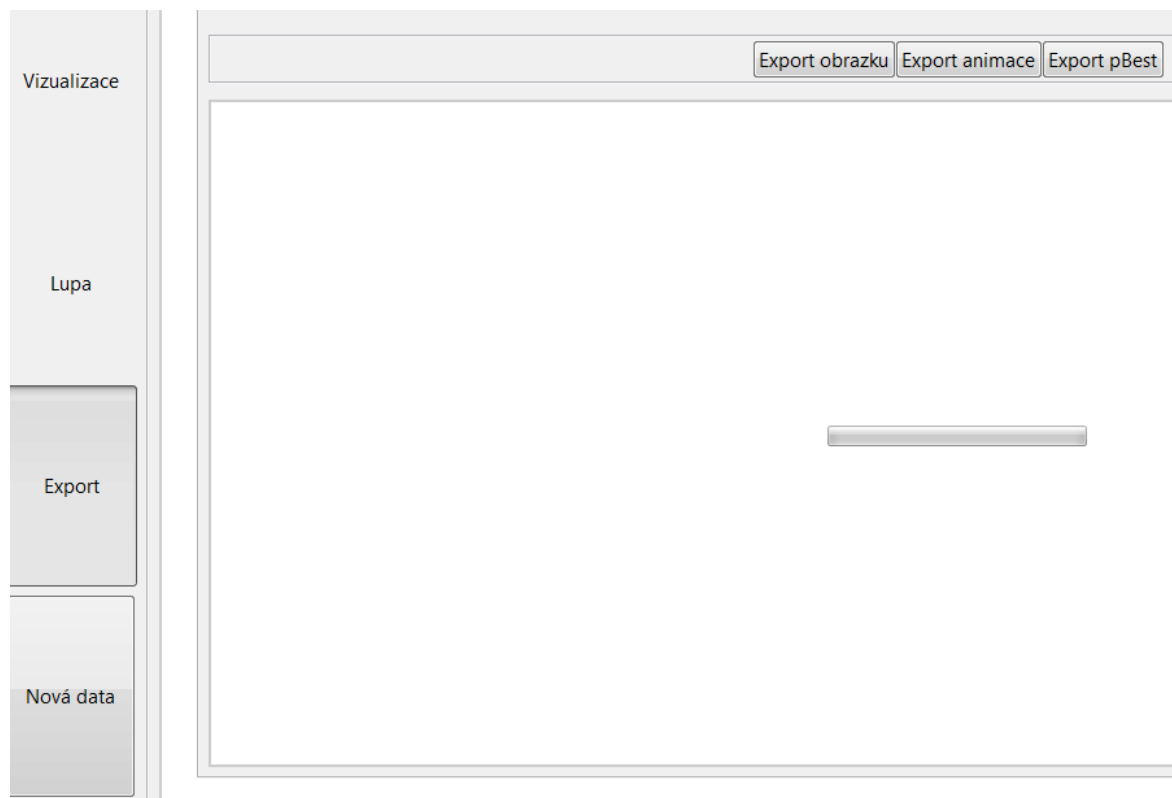
Pomocí nástroje lupa je možné přiblížení určité části grafu sloužícího k vizualizaci. Nástroj se skládá ze dvou grafů a panelu s posuvníkem a tlačítky jak je vidět na obr. 31. Na prvním grafu je umístěn lokátor, kterým je možné pomocí myši vybrat jakoukoliv část. Velikost lokátoru se nastavuje pomocí posuvníku *Velikost*. Část vybraná lokátorem se potom zobrazuje na druhém grafu. Stisknutím tlačítka *Nová velikost* dojde k uložení a při návratu zpět do vizualizace se již bude zobrazovat pouze ta část grafu, která byla vybrána. Původní velikost grafu se pak nastaví tlačítkem *Původní velikost*.



Obr. 31. Nástroj lupa

9.2.1.3 Export

Export slouží k exportu obrázků, animací a dat pBest. Skládá se z panelu, na kterém se nacházejí jednotlivá tlačítka a *progres indikátoru* znázorňujícím, zda byl export již dokončen, nebo stále probíhá.



Obr. 32. Export

Export obrázku

Zvolením tlačítka *Export obrázku* je uživatel vyzván k výběru umístění, názvu souboru a formátu. Uživatel má možnost uložit obrázek ve třech formátech. První je rastrový formát TIFF a další dva vektorové SVG a EPS. Export obrázku proběhne téměř okamžitě, takže na *progres indikátoru* není viditelné žádné načítání. Exportuje se vždy aktuální obrázek vizualizace.

Export animace

Tlačítko *Export animace* funguje na stejném principu jako export obrázku. V tomto případě si však uživatel volí mezi formáty souboru AVI a FLV. Export animace již není tak rychlý jako export obrázku, takže po spuštění exportování se začne pohybovat *progres*

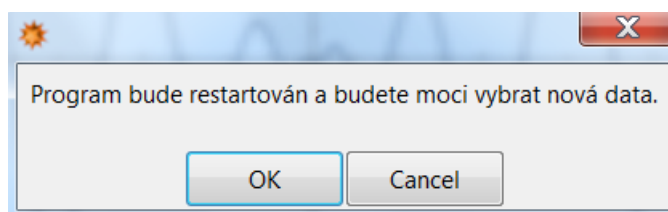
indikátor značící, že export stále probíhá. Celý proces může trvat až několik minut v závislosti na délce animace a formátu souboru. Délku animace uživatel volí posuvníkem *Animace* na ovládacím panelu.

Export pBest

Tlačítko *pBest* funguje opět jako dvě předešlé, jen uživatel má na výběr exportování pouze do souboru TXT. Do textového souboru se exportují data historie pBest.

9.2.1.4 Nová data

Po stisknutí tlačítka nová data je uživatel upozorněn, že dojde k novému spuštění aplikace.



Obr. 33. Dialogové okno pro načtení nových dat

V případě, že se uživatel rozhodne pro načtení nových dat, dialogové okno s aplikací bude ukončeno a dojde k opětovnému přeložení kódu a spuštění aplikace jako v kapitole 9.1.

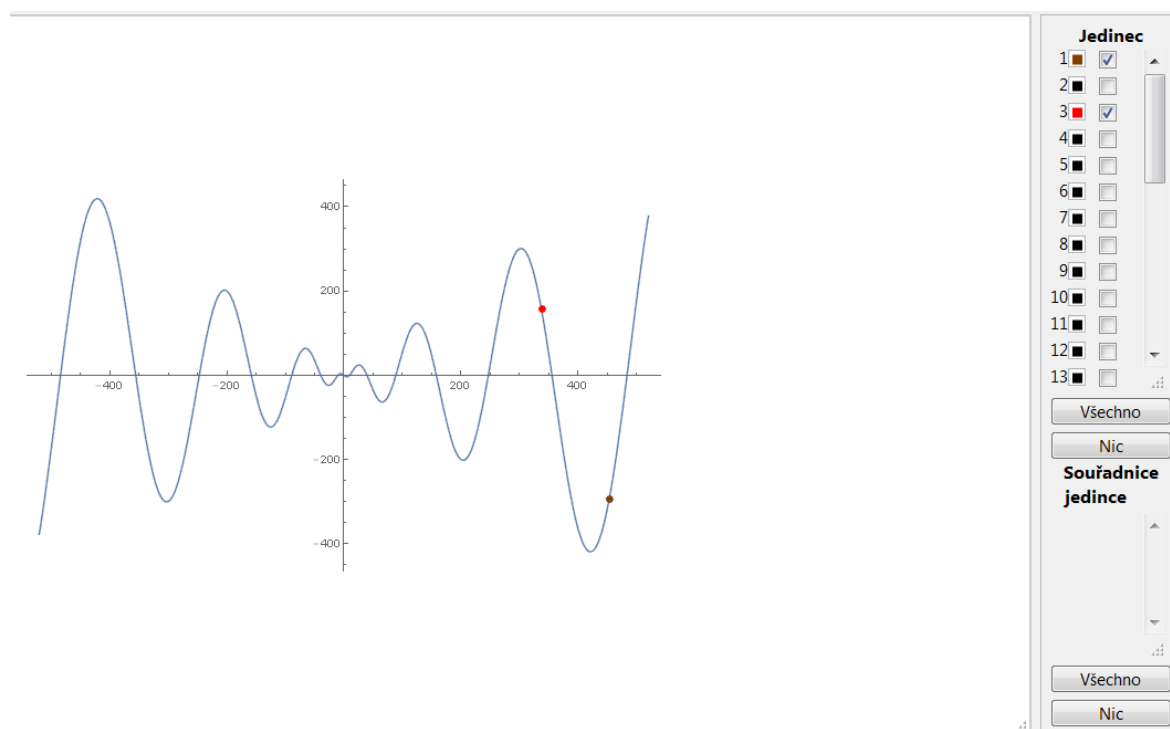
9.2.2 Panel pro výběr jedinců a jejich souřadnic

Panel je rozdělen na dvě části. Jedna část slouží pro výběr jedinců a druhá pro výběr souřadnic jedince. Obě části jsou tvořeny zaškrťovacími políčky a políčky pro změnu barvy. Část pro výběr souřadnic se používá pouze v případě, že je načten soubor s tří a vícedimenzionálními daty. V opačném případě tato část neobsahuje žádná políčka.

9.2.2.1 Část pro výběr jedince

V případě načtení vstupního souboru s populací např. o třiceti jedincích, bude v této části vygenerováno třicet zaškrťovacích políček a třicet políček pro změnu barvy. Každá dvojice těchto políček odpovídá jednomu jedinci v populaci. Pomocí zaškrťovacího políčka je pak

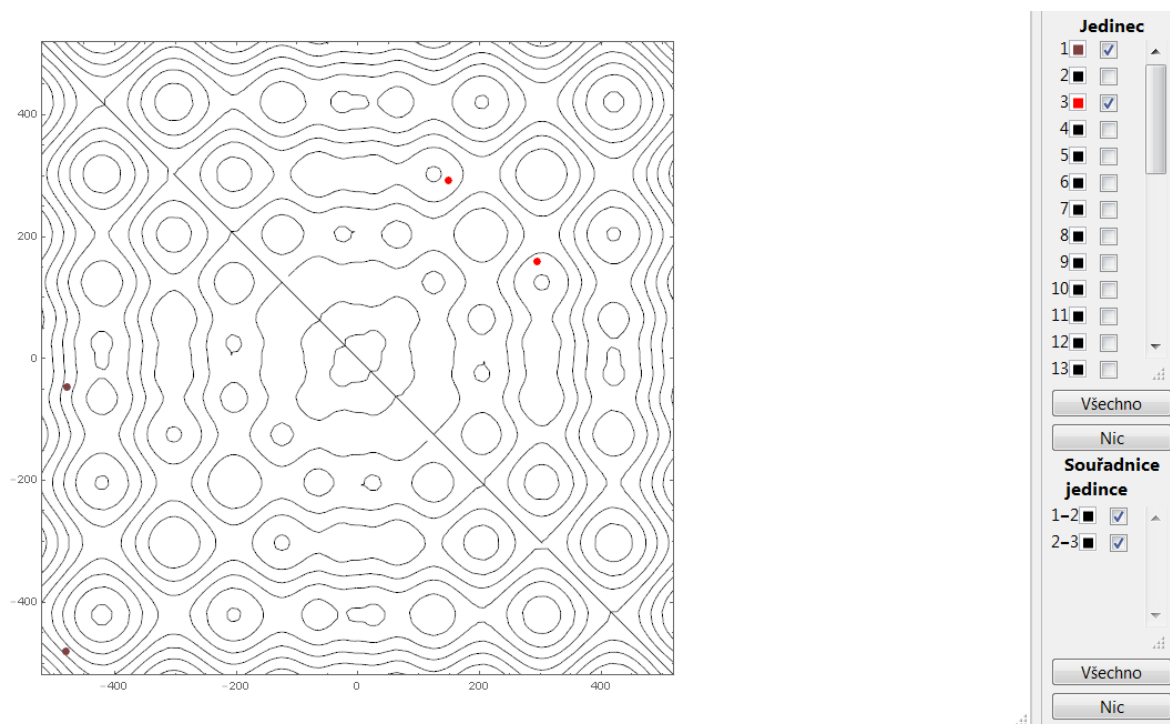
možné nastavovat, zda bude jedinec zobrazen při vizualizaci či ne a pomocí políčka pro změnu barvy je možné nastavovat barvu tečky (bodu), která daného jedince symbolizuje.



Obr. 34 Výběr jedince a nastavení jeho barvy

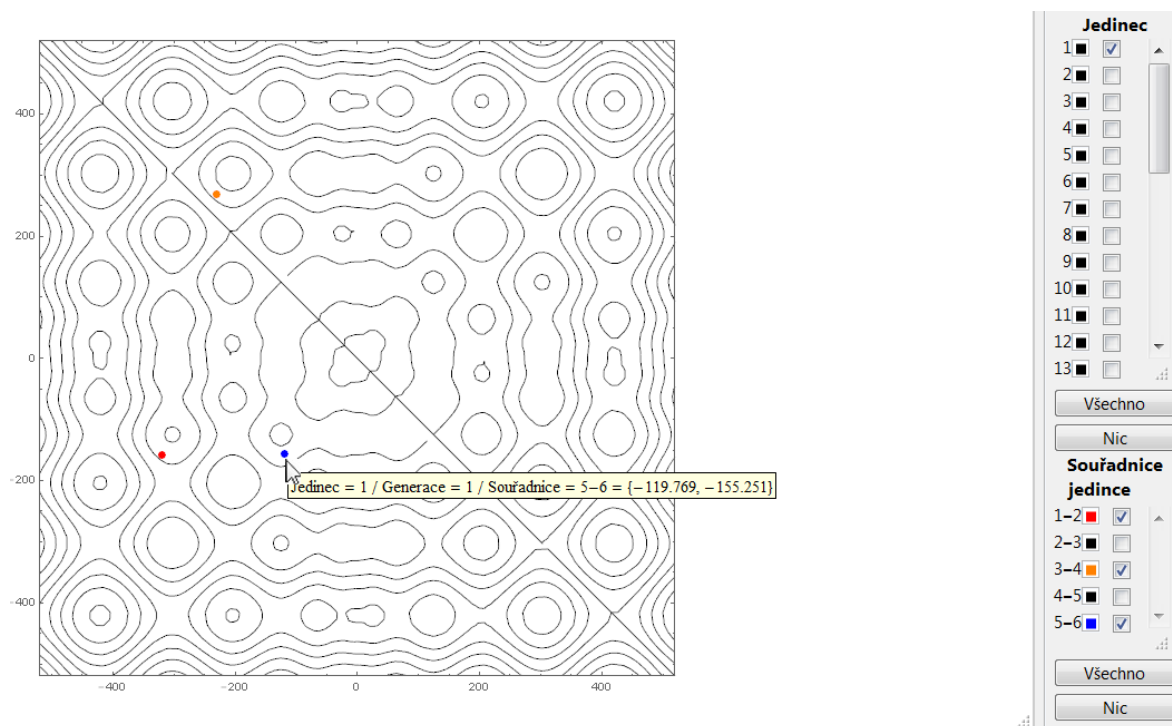
9.2.2.2 Část pro výběr souřadnice

Jak bylo již zmíněno, v této části se generují políčka jen v případě, že jde o třídímní a vícedímní data. Bude-li tedy načten např. soubor obsahující jedince, jejichž počet argumentů bude roven třem, dojde k rozdělení na jednotlivé souřadnice tak, jak to bylo popsáno v kapitole 8.2.3 a budou vygenerovány dvě zaškrťovací políčka a dvě políčka pro změnu barvy jak je vidět na obr. 35.



Obr. 35 výběr dvou třídimenzionálních jedinců

Uživatel si může pomocí zaškrťovacích políček zvolit, zda bude jedinec tvořen v tomto případě na obr. 35 dvěma body nebo jen jedním z těchto dvou bodů. První bod je reprezentován zaškrťovacím políčkem 1-2, je tedy tvořen první dvojicí souřadnic, což znamená prvními dvěma argumenty účelové funkce jedince. Druhý bod je pak reprezentován zaškrťovacím políčkem 2-3 a je tedy tvořen druhou dvojicí souřadnic, což znamená druhým a třetím argumentem účelové funkce jedince. Políčka pro změnu barvy v části *Souřadnice jedince* jsou aktivní jen v případě, že je vybrán pouze jeden jedinec, pak je možné u tohoto jedince měnit barvu bodů, kterými je tvořen tak, jak je to vidět na obr. 36.

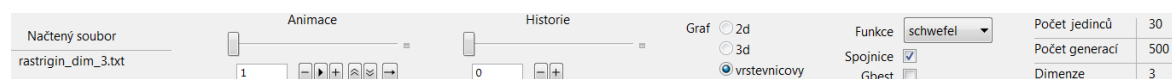


Obr. 36. Výběr šestidimenzionálního jedince

Na obr. 36. je vidět jeden vybraný šestidimenzionální jedinec, který je reprezentovaný třemi body o vybraných souřadnicích.

9.2.3 Ovládací panel

Na ovládacím panelu se nachází jméno načteného souboru, ovládací prvek animace, posuvník pro nastavení historie, přepínač druhu grafu, rozevírací seznam pro výběr funkce, zaškrtnutí políčko pro zobrazení spojnic, zaškrtnutí políčko pro zobrazení grafu gBest a informační tabulka o vlastnostech načtených dat.



Obr. 37. Ovládací panel

9.2.3.1 *Jméno načteného souboru*

Zde je zobrazený přesný název načteného souboru se vstupními daty v tomto případě (obr. 37) *schwefel_dim_3.txt*, což napovídá, že jde o třídímenzionální data určená k vizualizaci pomocí grafu Schwefelovi funkce.

9.2.3.2 *Ovládací prvek animace*

Jde o takzvaný animátor, což je vlastně posuvník umožňující automatický posun, který se spustí pomocí tlačítka *Play*. Obsahuje i další tlačítka jako *Step Forward* a *Step Backward* značené pomocí znaků plus a mínus, které slouží k posuvu o jeden krok zpět, nebo dopředu. Dále pak tlačítka *Faster* a *Slower*, značené dvojími šipkami, sloužící ke zrychlení či zpomalení animace a poslední tlačítko značené šipkou, které slouží k nastavení, zda poběží animace od začátku, od konce, anebo nejprve od začátku a pak od konce.

9.2.3.3 *Posuvník pro nastavení historie*

Pomocí tohoto posuvníku se nastavuje historie vývoje populace, v případě nastavení na hodnotu nula se budou zobrazovat jedinci pouze z aktuální generace. Nastavením např. na hodnotu 2 se budou zobrazovat jedinci z aktuální generace, ale také jedinci z předchozích dvou generací. Historii lze nastavovat posuvníkem, nebo také jako v případě animátoru pomocí tlačítek *Step Forward* a *Step Backward* po jedné hodnotě.

9.2.3.4 *Přepínač druhu grafu*

Tento přepínač slouží k přepínání pouze v případě, že je načtený soubor s dvojdimenzionálními daty, v takovém případě je možné přepínat mezi 3D grafem a vrstevnicovým grafem.

9.2.3.5 *Rozevírací seznam pro výběr funkce*

Rozevírací seznam slouží pro výběr některé z testovacích funkcí. Uživatel má na výběr Schwefelovu funkci, První, Třetí, Čtvrtou Dejongovu funkci, Rastriginovu funkci a Uživatelskou funkci. Uživatelskou funkci si může uživatel sám nadefinovat přímo ve zdrojovém kódu aplikace. Hned na začátku kódu jsou definovány testovací funkce, stačí tedy najít funkci se jménem *uzivatelska* a změnit funkční předpis této funkce. Poté jsou potřeba ještě nastavit meze, v jakých se má vybraná funkce vykreslovat. To se provádí ve funkci *meze-Funkce*, která se nachází hned za definicemi testovacích funkcí. Ke každé funkci jsou zde

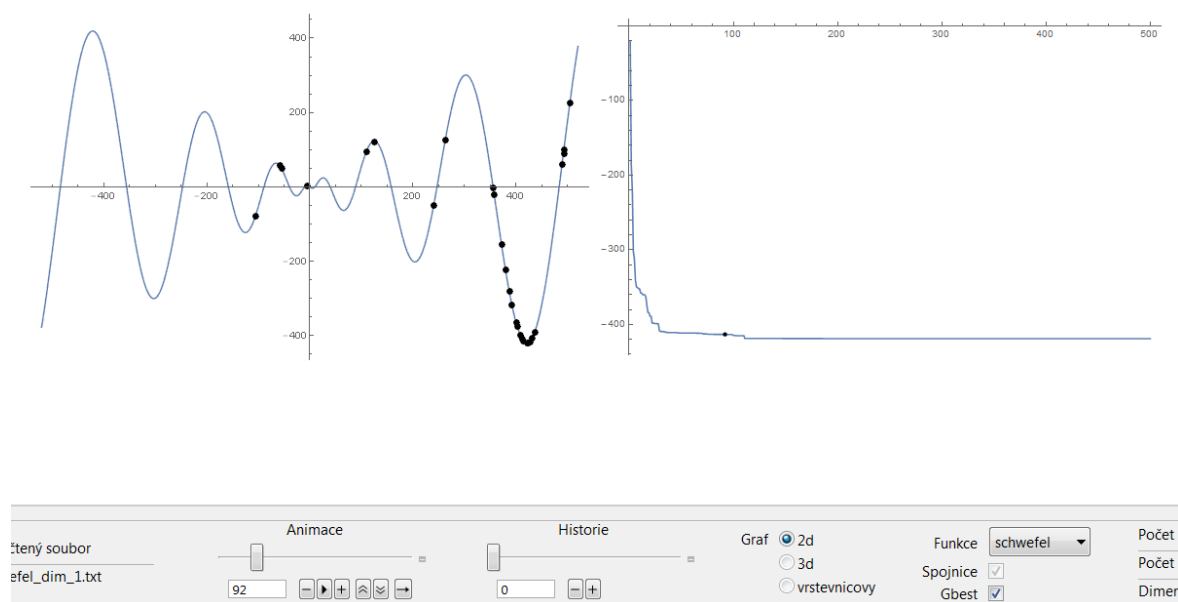
nadefinované její meze formou dvojrozměrného pole např. $\{\{-520,520\}, \{-520,520\}\}$, kde první dvojice čísel představuje meze osy x a druhá meze osy y .

9.2.3.6 Spojnice

Zaškrtnutí políčko slouží k zobrazení spojníc (šipek) v případě, že vizualizace probíhá pomocí vrstevnicového grafu a historie je nastavena na hodnotu větší než nula.

9.2.3.7 Gbest

Pomocí zaškrtnutí políčka se nastavuje, zda má být zobrazen také graf gBest. Na grafu gBest se nachází černá tečka, která značí, jaké hodnoty na grafu nabývá aktuální vizualizovaná generace. V případě obr. 38 jde o 92. generaci.



Obr. 38. Zobrazení grafu gBest

9.2.3.8 Informační tabulka

Zobrazuje informace o načtených datech.

ZÁVĚR

Hlavním cílem této bakalářské práce bylo vytvoření vizualizačního nástroje v prostředí Wolfram Mathematica, který by umožňoval sledování vnitřní dynamiky vybraných evolučních algoritmů, a který by bylo možné využít ve výzkumu a výuce na FAI UTB.

V teoretické části práce byla zpracována literární rešerše, kde byly v první kapitole popsány principy a základní pojmy evolučních algoritmů. V dalších kapitolách byly představeny vybrané testovací funkce a možnosti vizualizace. V poslední kapitole byl pak čtenář seznámen s prostředím Wolfram Mathematica a programovacím jazykem C.

V praktické části byl v první kapitole řešen export dat v jazyce C a jejich následný import do prostředí Mathematica, v následující kapitole pak proběhla analýza těchto dat. Poslední dvě kapitoly byly věnovány samotnému vizualizačnímu nástroji, který představuje vytvořená aplikace. Tato aplikace v sobě sdružuje nástroj pro vizualizaci, nástroj pro přiblížení určité části grafu a nástroj pro export animace, obrázků či dat pBest. Uživatelské prostředí aplikace bylo naprogramováno co nejpřívětivěji s ohledem na uživatele. Velká pozornost byla také kladena na optimalizaci kódu a celkovou plynulost běhu aplikace při spuštění animace. Plynulost běhu a délka spouštění aplikace při načtení souboru s daty ovšem závisí i na jednotlivých verzích softwaru Wolfram Mathematica. Chod aplikace byl testován ve verzi softwaru 10 a 9 s tím, že ve verzi 10 byla délka spouštění aplikace až třikrát větší, než ve verzi 9. Aplikace byla také naprogramována tak, aby jí bylo možné rozšířit o další testovací funkce a uživatel tak nebyl svázán použitím pouze předdefinovaných funkcí. Celý popis ovládání a jednotlivé funkce aplikace byly popsány v praktické části bakalářské práce. Spustitelný zdrojový kód zmíněné aplikace pro software Mathematica se nachází na přiloženém CD.

SEZNAM POUŽITÉ LITERATURY

- [1] ZELINKA, Ivan, Zuzana OPLATKOVÁ, Miloš ŠEDA, Pavel OŠMERA a František VČELAŘ. *Evoluční výpočetní techniky: Principy a aplikace*. Praha: BEN, 2009. ISBN 978-80-7300-218-3.
- [2] ZELINKA, Ivan. *Umělá inteligence v problémech globální optimalizace*. 1. vyd. Praha: BEN - technická literatura, 2002, 189 s. ISBN 80-730-0069-5.
- [3] VOLNÁ, Eva. *EVOLUČNÍ ALGORITMY A NEURONOVÉ SÍTĚ* [online]. 1. vyd. Ostrava, 2012 [cit. 2015-04-29]. Dostupné z: http://www1.osu.cz/~volna/Evolucni_algoritmy_a_neuronove_site.pdf
- [4] PLUHACEK, Michal. *PSO Algoritmus v prostředí Mathematica*. Zlín, 2011. Diplomová práce. Univerzita Tomáše Bati ve Zlíně.
- [5] CHRAMCOV, Bronislav. *Základy práce v prostředí Mathematica*. Zlín, 2005. Skripta. Univerzita Tomáše Bati ve Zlíně.
- [6] HEROUT, Pavel. *Učebnice jazyka C*. Vyd. 3. České Budějovice: Kopp, 1994, 269 s. ISBN 80-85828-21-9.
- [7] MIČKA, Pavel. Konstrukce překladače. In: *Algoritmy.net* [online]. 2011 [cit. 2015-04-29]. Dostupné z: <http://www.algoritmy.net/article/100/Konstrukce-prekladace>
- [8] Mathematica. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2015 [cit. 2015-04-29]. Dostupné z: <http://en.wikipedia.org/wiki/Mathematica>
- [9] NĚMEC, Jan. Úvodní díl seriálu o programování v C a C++. In: *Linuxsoft* [online]. 2004 [cit. 2015-04-29]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=370

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

E_n	N - rozměrný prostor.
TIFF	Tagged Image File Format.
SVG	Scalable Vector Graphics.
EPS	Encapsulated PostScript.
AVI	Audio Video Interleaved.
FLV	Flash Video.
TXT	Text File.

SEZNAM OBRÁZKŮ

<i>Obr. 1. Cyklus evolučního algoritmu [1]</i>	12
<i>Obr. 2. Graf pBest populace o 30 jedincích</i>	14
<i>Obr. 3. Graf gBest populace o 30 jedincích</i>	14
<i>Obr. 4. První De Jongova funkce ve 2D</i>	16
<i>Obr. 5. První De Jongova funkce ve 3D</i>	17
<i>Obr. 6. Třetí De Jongova funkce ve 2D</i>	18
<i>Obr. 7. Třetí De Jongova funkce ve 3D</i>	18
<i>Obr. 8. Čtvrtá De Jongova funkce ve 2D</i>	19
<i>Obr. 9. Čtvrtá De Jongova funkce ve 3D</i>	20
<i>Obr. 10. Schwefelova funkce ve 2D</i>	21
<i>Obr. 11. Schwefelova funkce ve 3D</i>	21
<i>Obr. 12. Rastriginova funkce ve 2D</i>	22
<i>Obr. 13. Rastriginova funkce ve 3D</i>	23
<i>Obr. 14. Konvergence populace o třiceti dvojdimenzionálních jedincích ke globálnímu extrému v jednotlivých evolučních cyklech (generacích)</i>	24
<i>Obr. 15. Export dat v jazyce C</i>	29
<i>Obr. 16. Formát dat v textovém souboru</i>	30
<i>Obr. 17. Import dat v prostředí Wolfram Mathematica</i>	30
<i>Obr. 18. Analýza dat</i>	31
<i>Obr. 19. 2D graf testovací funkce</i>	32
<i>Obr. 20. vrstevnicový graf testovací funkce</i>	33
<i>Obr. 21. 3D graf testovací funkce</i>	33
<i>Obr. 22. Spojení 3D a vrstevnicového grafu</i>	34
<i>Obr. 23. Zobrazení jednodimenzionálních jedinců na 2D grafu</i>	35
<i>Obr. 24. Zobrazení dvojdimenzionálních jedinců na vrstevnicovém grafu</i>	36
<i>Obr. 25. Zobrazení dvojdimenzionálních jedinců na 3D grafu</i>	36
<i>Obr. 26. Zobrazení třídimenzionálního jedince na vrstevnicovém grafu</i>	37
<i>Obr. 27. Animátor</i>	38
<i>Obr. 28. Historie vývoje vybraných jedinců</i>	39
<i>Obr. 29. Dialogové okno při špatném formátu souboru</i>	40
<i>Obr. 30. Dialogové okno s aplikací</i>	40
<i>Obr. 31. Nástroj lupa</i>	41

<i>Obr. 32. Export</i>	<i>42</i>
<i>Obr. 33. Dialogové okno pro načtení nových dat</i>	<i>43</i>
<i>Obr. 34 Výběr jedince a nastavení jeho barvy.....</i>	<i>44</i>
<i>Obr. 35 výběr dvou třidimenzionálních jedinců</i>	<i>45</i>
<i>Obr. 36. Výběr šestidimenzionálního jedince</i>	<i>46</i>
<i>Obr. 37. Ovládací panel</i>	<i>46</i>
<i>Obr. 38. Zobrazení grafu gBest</i>	<i>48</i>

SEZNAM TABULEK

<i>Tab. 1. Populace jako matice $N \times M$ [1]</i>	13
---	----

SEZNAM PŘÍLOH

Příloha P I: CD se zdrojovým kódem pro software Wolfram Mathematica.