UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

ÚPRAVA METODY FLEXIBILNÍHO SIMPLEXU PRO ŘEŠENÍ PROBLÉMŮ GLOBÁLNÍ OPTIMALIZACE

Miroslav Provazník

Bakalářská práce
2016

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2015/2016**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Miroslav Provazník**
Osobní číslo:	**I12079**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Řízení procesů**
Název tématu:	**Úprava metody flexibilního simplexu pro řešení problémů globální optimalizace**
Zadávající katedra:	**Katedra řízení procesů**

Zásady pro vypracování:

Cílem práce je implementace metody flexibilního simplexu ve zvoleném programovacím jazyce a realizace několika modifikací tohoto algoritmu pro řešení problémů globální optimalizace. Ověření účinnosti metody na zvolené množině vhodných testovacích problémů a případné porovnání s jinou zvolenou metodou. Přehledné zpracování výsledků.
Teoretická část: Výklad obecných souvislostí a popis algoritmů implementovaných v praktické části.
Praktická část: Implementace algoritmů a přehledné zpracování experimentálních výsledků.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

NELDER, J. A.; MEAD, R. A simplex method for function minimization. Computer Journal, 1965, vol. 7, s. 308-313.
PRESS, W. H. et al. Numerical Recipes in C. Cambridge University Press, 1992.
TVRDÍK, J. Evoluční algoritmy. Ostrava: Ostravská univerzita, 2004.

Vedoucí bakalářské práce: **doc. Ing. Jan Cvejn, Ph.D.**
Katedra řízení procesů

Datum zadání bakalářské práce: **12. listopadu 2015**
Termín odevzdání bakalářské práce: **13. května 2016**

Univerzita Pardubice Česká republika

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2016

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 2. 5. 2016

Miroslav Provazník

Poděkování

Na tomto místě bych rád poděkoval mému vedoucímu doc. Ing. Janu Cvejnovi, Ph.D., za čas, rady, připomínky a trpělivost, které mi věnoval při tvorbě této bakalářské práce. Dále bych rád poděkoval všem, kteří mě podporovali během studia, především pak rodině, za morální a finanční podporu, kterou mi během studia poskytovali.

V Pardubicích dne 2. 5. 2016

Miroslav Provazník

ANOTACE

Práce se zabývá úpravou metody flexibilního simplexu. V teoretické části je obecně popsána problematika globální optimalizace, metoda flexibilního simplexu a několik jejích vytvořených modifikací. V praktické části jsou tyto jednotlivé modifikace otestovány na několika představených testovacích funkcích.

KLÍČOVÁ SLOVA

metoda flexibilního simplexu, úprava metody, globální optimalizace.

TITLE

MODIFICATION OF THE METHOD OF FLEXIBLE SIMPLEX FOR GLOBAL OPTIMALISTION PROBLEMS

ANNOTATION

The subject of this thesis is a modification of the method of flexible simplex. In the first part there is a general description of the global optimalization problems and the description of original flexible simplex method and several constructed modifications of this method. In the second part there are these modifications tested on presented test functions.

KEYWORDS

Flexible simplex method, Method modification, Global optimalization.

Obsah

Seznam zkratek

CRS	Controlled random search (řízené náhodné prohledávání)
Matlab	Matrix laboratory
SA	Simulated annealing (simulované žíhání)

Seznam značek

k	iterační krok
n	dimenze
N	prohledávaný prostor
x_c	bod kontrakce simplexu
x_e	bod expanze simplexu
x_n	druhý nejhorší bod simplexu
x_{n+1}	nejhorší bod simplexu
x_0	těžiště simplexu
x_r	bod reflexe simplexu
x_1	nejlepší bod simplexu
α	reflexní koeficient
γ	expanzní koeficient
ρ	kontrakční koeficient
σ	redukční koeficient

Seznam ilustrací

Seznam tabulek

ÚVOD

Tato práce se zabývá úpravou metody flexibilního simplexu pro řešení problémů globální optimalizace.

Cílem této práce je implementace metody flexibilního simplexu ve zvoleném programovacím jazyce a realizace několika modifikací tohoto algoritmu pro řešení problémů globální optimalizace a následné ověření účinnosti metod na zvolené množině vhodných testovacích problémů.

V teoretické části této práce jsou stručně popsány obecné problémy globální optimalizace a několik algoritmů pro hledání globálního minima. Dále je v teoretické části popsána samotná metoda flexibilního simplexu a několik vytvořených modifikací této metody.

V praktické části jsou popsány testovací funkce používané pro globální optimalizaci a použité v provedených experimentech, které jsou taktéž popsány v praktické části. V závěru praktické části jsou ve formě grafů prezentovány výsledky provedených experimentů.

Ve stručnosti je na konci této práce popsán software použitý k realizaci vytvořených modifikací a následných experimentů.

1 PROBLÉM GLOBÁLNÍ OPTIMALIZACE

Globální optimalizace, ačkoli se to nemusí na první pohled zdát, prostupuje celou řadou procesů živé i neživé přírody. Globální optimalizace se zabývá nalezením nejlepšího řešení problému tzv. optima. Poměrně známý je např. problém obchodního cestujícího, kdy obchodní cestující musí projít daný počet měst pomocí co nejkratší trasy a vrátit se do výchozího města. Hledaným optimem může být extrém (minimum nebo maximum) účelové funkce.

Problém nalezení globálního minima můžeme definovat následovně: Uvažujme účelovou funkci

$$f : N \to R, \quad N \subseteq R^n. \tag{1.1}$$

Bod x^* se nazývá globálním minimem, jestliže pro všechna $x \in N$

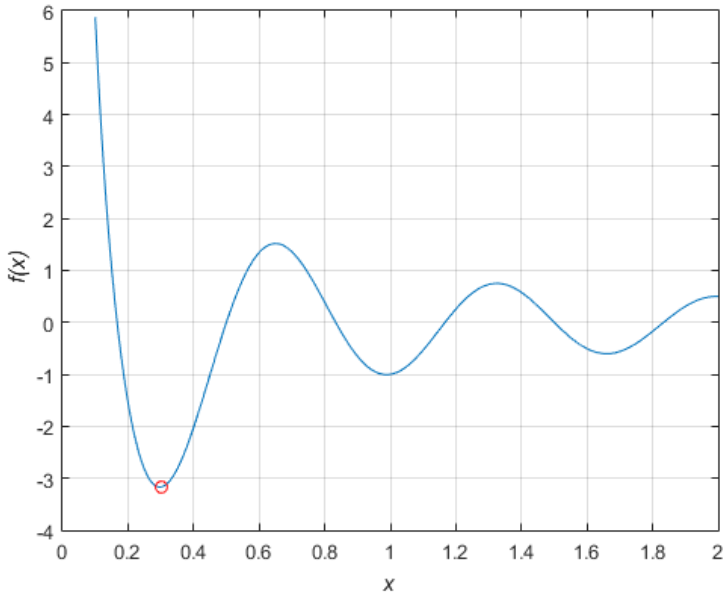
$$f(x) \geq f(x^*). \tag{1.2}$$

Globálnímu minimu tedy odpovídá jeden nebo více bodů z N, s nejmenší funkční hodnotou. Je-li potřeba nalézt globální maximum, položíme

$$g(x) = -f(x). \tag{1.3}$$

a minimalizujeme funkci $g(x)$.

Kromě globálního minima, může mít funkce více lokálních extrémů. To ilustruje obr. 1.1 na následující stránce. Na grafu je zobrazena funkce v jedné dimenzi, která má několik extrémů. Na ose x jsou body nabývající hodnot v intervalu od 0,1 do 2,0. Na ose y jsou zobrazeny funkční hodnoty v těchto bodech. Jak z grafu vyplývá, funkce má na tomto intervalu tři maxima a tři minima. Z grafu vyplývá, že funkce má pouze jedno globální minimum (označené červeným kruhem), které se nachází v bodě $x = 0{,}3$.



Obr. 1.1 – Ilustrace globálního minima funkce

Pomocí matematické analýzy je možné nalézt minimum funkcí, u kterých existuje první a druhá derivace. Tento postup, ale není možno použít vždy. Nalézt obecné řešení takto formulovaného problémů je obtížné, obzvláště pokud má účelová funkce více minim, není diferencovatelná, popř. má další nepříjemné vlastnosti. Jak uvádí Tvrdík (2004) analýza problému globální optimalizace ukazuje, že neexistuje deterministický algoritmus řešící obecnou úlohu globální optimalizace v polynomiálním čase.

Nemožnost nalezení takového deterministického algoritmu vedla k využití algoritmů stochastických algoritmů. Stochastické algoritmy nemohou spolehlivě garantovat nalezení optima v konečném počtu kroků, ale mohou nalézt řešení, které je prakticky použitelné a v přijatelném čase.

Stochastické algoritmy pro globální optimalizaci prohledávají určený prostor N s danou heuristikou. Heuristika je označení pro postup využívající pro nalezení řešení, ve kterém se využívá intuice, zkušenost, analogie a náhoda. Proto většina stochastických algoritmů obsahuje zjevný nebo skrytý proces učení (kromě algoritmu slepého náhodného prohledávání). S heuristikami se běžně setkáváme v živé i neživé přírodě, např. hledání hub, pokus o výhru sportovního utkání nebo simulované žíhání, které je modelem pomalého ochlazování tuhého tělesa.

V posledních desetiletích je čím dál častěji využíváno pro hledání globálního minima funkcí stochastických algoritmů evolučního typu. Evoluční algoritmy jsou v podstatě rozšíření Darwinovy evoluční teorie vývoje populací do oblasti globální optimalizace.

Charakteristické pro evoluční algoritmy je, že pracují s populací, kterou daným způsobem modifikují, tak aby zlepšovali její vlastnosti. Evoluční algoritmy využívají tzv. evoluční operátory, především tyto:

- selekce – nejsilnější jedinci populace mají největší pravděpodobnost přežití a předání svých vlastností,
- křížení – dva nebo více jedinců si vymění informace a dojde ke vzniku nových jedinců kombinujících vlastní rodičů,
- mutace – může dojít k náhodné modifikaci informací zakódovaných v jedinci.

1.1 VYBRANÉ METODY GLOBÁLNÍ OPTIMALIZACE

V této části bude krátce popsáno několik vybraných metod používaných pro řešení problémů globální optimalizace.

1.1.1 Náhodné prohledávání

Náhodné prohledávání, někdy nazývané jako slepé náhodné prohledávání, je velmi jednoduchý stochastický algoritmus pro hledání globálního minima. Algoritmus opakovaně generuje náhodné řešení (nový bod) v prohledávaném prostoru. Pokud je nový bod lepší než dosud nalezené řešení, je nový bod zapamatován a nahradí dosavadní nejlepší řešení. V opačném případě je bod zapomenut a generuje se bod nový.

Jak uvádí Tvrdík (2004) jde o algoritmus využívající velmi prostou heuristiku, který bychom mohli popsat jako „zkus bez rozmyslu cokoliv“.

1.1.2 Řízené náhodné prohledávání

Řízené náhodné prohledávání (anglicky controlled random search, CRS) je jednoduchý stochastický algoritmus pro hledání globálního minima v souvislé oblasti. Algoritmus CRS navrhl Price v sedmdesátých letech minulého století. Jako rozšíření metody simplexu, určené pro lokální optimalizaci. Tento přístup byl využit u rozšíření provedených v této práci. Tento algoritmus pracuje s populací M bodů v prohledávaném prostoru N, ze kterých se generuje nový bod. Pokud je funkční hodnota v novém bodu lepší, než v nejhorším bodě populace, nahradí nový bod doposud nejhorší bod v populaci. K nalezení nového bodu je u tohoto algoritmu využíváno reflexe simplexu. Při reflexi simplexu dojde k překlopení nejhoršího bodu simplexu přes těžiště simplexu. Reflexe simplexu je podrobněji popsána

v následující kapitole. K vytvoření simplexu je u tohoto algoritmu z populace náhodně vybráno *n+1* bodů.

Výměnou vždy nejhoršího bodu dochází ke koncentraci populace okolo nejlepších bodů, tj. bodů s nejnižší funkční hodnotou. Později navrhl Price úpravu, kdy se do simplexu vybere nejlepší bod populace a zbylé body simplexu se volí náhodně.

1.1.3 Simulované žíhání

Metoda simulovaného žíhání (anglicky Simulated annealing, SA) nebo také simulované ochlazování je stochastický optimalizační algoritmus inspirovaný procesem žíhání oceli. Jde tedy o další inspiraci pro tvorbu algoritmů z oblasti přírody, v tomto případě neživé. Při žíhání oceli se postupně kov ochlazuje a zahřáté atomy mohou s danou pravděpodobností přejít do stavu s vyšší energií. Postupným ochlazování kovu ztrácejí atomy energii a čím dál méně se nacházejí ve stavech s vyšší energií. Tohoto jevu je využito v metodě simulovaného žíhání, kdy je algoritmus schopen dostat se z lokálního minima a předejít uváznutí dočasným přijetím horší hodnoty.

V případě, že algoritmu hrozí při hledání globálního minima uváznutí v lokálním minimu, dojde k přijetí bodu s horší funkční hodnotou na základě dané pravděpodobnosti. Tato pravděpodobnost je dána rychlostí konvergence k hledanému minimu, tj. pokud jsou prováděné změny velké (teplota kovu rychle klesá) je pravděpodobnost přijetí bodu s horší funkční hodnotou nižší než za situace, kdy jsou prováděné změny malé (teplota kovu klesá pomalu).

2 METODA FLEXIBILNÍHO SIMPLEXU

Metodu flexibilního simplexu známé podle příjmení autorů jako Nelder-Mead (někdy označována anglicky jako „downhill simplex method“, česky „metoda svažujícího se simplexu“) je jedna z klasických metod lokální optimalizace, nevyužívající pro nalezení minima derivací. Tuto metodu poprvé popsali John Nelder a Roger Mead v roce 1965 ve své práci „A simplex method for function optimalization“, kterou publikovali v britském vědeckém časopise Computer Journal.

Nelder a Mead ovšem nebyli první, kdo navrhl pro optimalizaci metodu užívající tzv. simplex. Metodu simplexů navrhli v roce 1962 Spendley, Hext a Himsworth ve článku „Sequential Application of Simplex Desing in Optimalization and Evolutionary Operation“ (Taufer, 2010).

Základním prvkem metody je tzv. simplex. Simplex je množina N+1 bodů v N-rozměrném prostoru, resp. počet neznámých dané funkce. Dle rozměru, ve kterém se simplex nachází, se označuje jako 0-simplex, 1-simplex atd. Simplex v jednorozměrném prostoru je úsečka, ve dvojrozměrném prostoru jde o trojúhelník, pro třírozměrný prostor jde o tzv. tetraedr (jehlan s trojúhelníkovou základnou). Počet vrcholů simplexu je tudíž vždy o jeden vrchol vyšší než je rozměr prostoru, ve kterém se nachází, resp. o jeden více, než je počet nezávisle proměnných (Press, 1992).

2.1 ALGORITMUS METODY FLEXIBILNÍHO SIMPLEXU

Základním principem metody Nelder-Mead pro nalezení minima je výměna bodů simplexu za lepší body tj. body s nižší funkční hodnotou.

Před započetím celého výpočtu je nutné zvolit hodnoty koeficientu reflexe α, expanze γ, kontrakce ρ a redukce σ. Značení koeficientů koresponduje s názvy proměnných, které je v dané metodě reprezentují. Zvolené hodnoty těchto koeficientů jsou uvedeny v tab. 2.1. Tento výběr koeficientů byl zvolen, jelikož vychází jako nejlepší (Nelder, 1965).

Tab. 2.1 – Koeficienty pro metodu Nelder-Mead

Zn. koeficientu	α	γ	ρ	σ
Hodnota	1	2	0,5	0,5

Před započetím je vytvoření počátečního simplexu. Při vytváření počátečního simplexu je důležitá jeho velikost. Příliš velký počáteční simplex může vést k pomalému prohledávání. Naproti tomu příliš malý počáteční simplex může vést k uváznutí.

Původní algoritmus, který navrhli Nelder a Mead obsahuje pravidelný počáteční simplex, který je ovšem při expanzi deformován. Proto již ve stejném roce, kdy Nelder a Mead svojí metodu publikovali, navrhl M. J. Box již v počátku nevycházet z pravidelného uspořádání, ale vrcholy volit náhodně (např. ve tvaru obecného trojúhelníku), jak uvádí Taufer (2010).

Po vytvoření počátečního simplexu dojde k výpočtu těžiště simplexu. Těžiště simplexu počítáme jako průměr všech vrcholů simplexu, kromě vrcholu s nejhorší funkční hodnotou. Vztah pro výpočet těžiště je dán následujícím vztahem

$$x_0 = \frac{1}{n}\sum_{i=1}^{n} x_i \,, \tag{2.1}$$

kde x_i – je bod simplexu na *i*-té pozici.

Dále metoda opakovaně provádí kroky reflexe, expanze, kontrakce, popř. redukce, jak je popsáno v následujících podkapitolách.

2.1.1 Reflexe

Po vypočtení těžiště následuje hledání nového bodu simplexu. Pro vypočtení tohoto bodu se užívá tzv. reflexe. Reflexe je překlopení (zrcadlení) nejhoršího bodu přes těžiště. Bod reflexe získáme podle následujícího vztahu

$$x_r = x_0 + \alpha(x_0 - x_{n+1}), \tag{2.2}$$

kde x_0 – těžiště simplexu,

x_{n+1} – nejhorší bod simplexu,

α – reflexní koeficient.

Následně se testuje podmínka

$$f(x_1) \leq f(x_r) < f(x_{n+1}), \tag{2.3}$$

kde x_1 – nejlepší bod simplexu,

x_r – bod reflexe,

x_{n+1} – nejhorší bod simplexu.

Pokud je tato podmínka splněna, je původně nejhorší bod simplexu nahrazen nový bodem a celý cyklus se opakuje.

2.1.2 Expanze

Pokud platí následující podmínka

$$f(x_r) < f(x_1), \tag{2.4}$$

kde x_1 – nejlepší bod simplexu,

x_r – bod reflexe.

Je určen bod expanze podle následujícího vztahu

$$x_e = x_0 + \gamma(x_0 - x_{n+1}), \tag{2.5}$$

kde x_0 – těžiště simplexu,

x_{n+1} – nejhorší bod simplexu,

γ – expanzní koeficient.

Pokud je bod expanze lepší jak bod reflexe, tj. platí

$$f(x_e) < f(x_r), \tag{2.6}$$

kde x_e – bod expanze,

x_r – bod reflexe,

dojde k nahrazení nejhoršího bodu simplexu bodem expanze. Pokud není tato podmínka splněna, je nejhorší bod nahrazen bodem reflexe a cyklus se opakuje.

2.1.3 Kontrakce

Pokud nedošlo ke splnění podmínek reflexe ani expanze, je podle následujícího vztahu vypočítán bod kontrakce

$$x_c = x_0 + \rho(x_0 - x_{n+1}), \tag{2.7}$$

kde x_c – bod kontrakce,

x_0 – těžiště simplexu,

x_{n+1} – nejhorší bod simplexu,

ρ – kontrakční koeficient.

Poté se testuje následující podmínka

$$f(x_c) < f(x_{n+1}), \tag{2.8}$$

kde x_c – bod kontrakce,

x_{n+1} – nejhorší bod simplexu.

Pokud podmínka platí, dojde k nahrazení nejhoršího bodu simplexu bodem kontrakce. Pokud není tato podmínka splněna, přistoupí se k tzv. redukci.

2.1.4 Redukce

Při redukci dojde k nahrazení všech bodů simplexu, kromě bodu nejlepšího podle následujícího vztahu

$$x_i = x_1 + \sigma(x_i - x_1) \quad i \in \{2,\ldots,n+1\}, \tag{2.9}$$

kde x_i – bod simplexu na *i*-té pozici,

x_1 – nejlepší bod simplexu,

σ – redukční koeficient.

Nahrazením všech bodů, kromě bodu nejlepšího dojde ke smrštění celého simplexu směrem k nejlepšímu bodu.

2.2 UKONČOVACÍ PODMÍNKY

Optimalizace prováděná pomocí algoritmu uvedeného v předchozí části může být ukončena jednou z několika podmínek. První z nich je podmínka, která ukončí optimalizaci v případě, pokud je rozdíl funkčních hodnot nejlepšího a nejhoršího vrcholu simplexu menší než předem stanovená odchylka. Tato podmínka je vyjádřena následujícím vztahem

$$f(x_{n+1}) - f(x_1) \leq \xi, \tag{2.10}$$

kde x_{n+1}– nejhorší vrchol simplexu,

x_1– nejlepší vrchol simplexu,

ξ – konvergenční odchylka.

Druhou podmínkou je vyhodnocení překročení maximálního počtu iteračních kroků. Tuto podmínku můžeme zapsat tímto vztahem

$$k < k_{\max}, \tag{2.11}$$

kde k – je aktuální počet iteračních kroků,

$k_{\max}$ – je maximální zadaný počet iteračních kroků.

3 NAVRŽENÁ ROZŠÍŘENÍ METODY FLEXIBILNÍHO SIMPLEXU

V této kapitole jsou popsány tři metody, které byly vytvořeny úpravou původní metody flexibilního simplexu, která je popsána v předchozí kapitole. Všechny tři zde popsané upravené metody pracují s populací bodů, podobně jako původní metoda CRS, zmíněná v oddíle 1.1. Jednotlivé varianty nemají speciální název a jsou dále uvedeny jako varianta 1, 2 a 3. Všechny varianty využívají k vypočtení nového bodu metodu flexibilního simplexu, ovšem bez smrštění (redukce).

Pro vytvoření náhodné počáteční populace je u všech variant užito rovnoměrného rozdělení pravděpodobnosti, v intervalu daném testovacím prostorem pro danou funkci.

Ukončovací podmínky mohou být stejné jako u simplexové metody, které jsou popsány v předchozí kapitole. Oproti simplexové metodě, kde byl porovnáván rozdíl funkčních hodnot nejlepšího a nejhoršího bodu simplexu, u těchto modifikací byl porovnáván rozdíl funkčních hodnot nejlepšího a nejhoršího bodu populace. U zde popsaných modifikací simplexové metody je jako ukončovací podmínka stanoven pouze maximální počet kroků. Jako krok je zde označeno vypočtení jednoho bodu populace.

3.1 VARIANTA 1

První variantou je taková modifikace simplexové metody, kdy v případě nalezení nového bodu pomocí simplexové metody, který je lepší než nejhorší bod simplexu, dojde k nahrazení nejhoršího bodu populace tímto novým bodem. Tato metoda se nejvíce podobá metodě CRS.

Po spuštění dojde k vytvoření náhodné počáteční populace P o zvolené velikosti. V každém iteračním kroku je z této populace náhodně vytvořen simplex z bodů populace a je provedena operace vedoucí k nalezení nového bodu (reflexe, expanze nebo kontrakce). Nový bod je zařazen do populace na pozici bodu, který má nejhoršího funkční hodnotu z celé populace.

3.2 VARIANTA 2

I u této varianty je po spuštění prvním krokem vytvoření náhodné počáteční populace P o zvolené velikosti. Tato varianta podobně jako varianta předchozí ovšem na rozdíl od

předchozí varianty nezaujme nový bod pozici patřící původně nejhoršímu bodu populace, ale dojde k obsazení pozice v populaci, která odpovídá nejhoršímu bodu vybraného simplexu.

Poté dojde k vytvoření náhodného simplexu z bodů populace. Následně je pomocí simplexové metody nalezen nový bod, který je zařazen do populace na pozici odpovídající pozici nejhoršího bodu simplexu.

3.3 VARIANTA 3

Třetí varianta se od předchozích variant liší více, jelikož na rozdíl od nich tato varianta pracuje s dvojící populací. Stejně jako v předchozích variantách je na začátku vytvořena náhodná počáteční populace P. Následně jsou pomocí simplexové metody, na základě bodů populace P, nalezeny nové body, ze kterých je vytvořena nová populace Y, která obsahuje stejný počet jedinců (bodů), jako populace P. Po naplnění populace Y na stejnou velikost jako má populace P je z těchto dvou populací vytvořena nová populace P, která je stejně velká jako obě původní populace. Nová populace P obsahuje ty nejlepší jedince (body) z obou původních populací.

Výběr těchto nejlepších jedinců je realizován jako spojení obou původních populací do populace o dvojnásobné velikosti oproti původním populacím. Následně jsou jedinci populace setříděni dle jejich funkční hodnoty a nová populace je zkrácena na poloviční velikost tj. velikost odpovídající původní populaci P resp. populaci Y.

Tento cyklus se opakuje, dokud není plněna některá ze zvolených ukončovacích podmínek. V tomto případě dojde k ukončení cyklu, pokud je překročen maximální počet kroků.

4 PRAKTICKÁ ČÁST

V této kapitole je popsán způsob testování vytvořených modifikací metody flexibilního simplexu uvedených v předchozích kapitolách, na vybraných testovacích problémech. Dále jsou v této kapitole prezentovány výsledky experimentů.

4.1 TESTOVACÍ FUNKCE

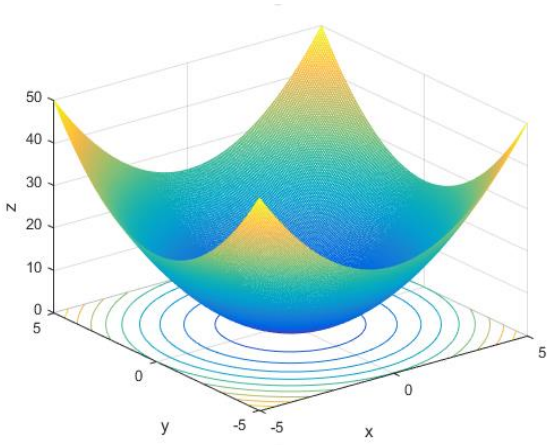
Pro testování algoritmů pro globální optimalizaci se užívá celá řada testovacích funkcí (problémů). U těchto funkcí známe jejich správné řešení resp. globální minimum. V tomto oddíle je uvedeno pět testovacích funkcí, z nichž poslední tři uvedené byly vybrány pro testování algoritmů popsaných v předchozí kapitole. Grafy všech uvedených testovacích funkcí byly vytvořeny v prostředí Matlab a to pro $N = 2$. Informace o funkcích jsou uvedeny tak, jak je uvádí Tvrdík (2004).

4.1.1 První De Jongova funkce

První De Jongova funkce je ze zde popsaných funkcí nejjednodušší. Jde o jednoduchý rotační paraboloid. Tato funkce má pouze jediné minimum a to v bodě $x = (0, 0,\ldots, 0)$, ve kterém je funkční hodnota $f(x) = 0$. Testovací prostor pro tuto funkci je obvykle vymezen podmínkou $x_i \in [-5{,}12; 5{,}12]$. Funkce je definována následovně

$$f(x) = \sum_{i=1}^{n} x_i^2 \,. \tag{4.1}$$

Graf této funkce je uveden na obr. 4.1.



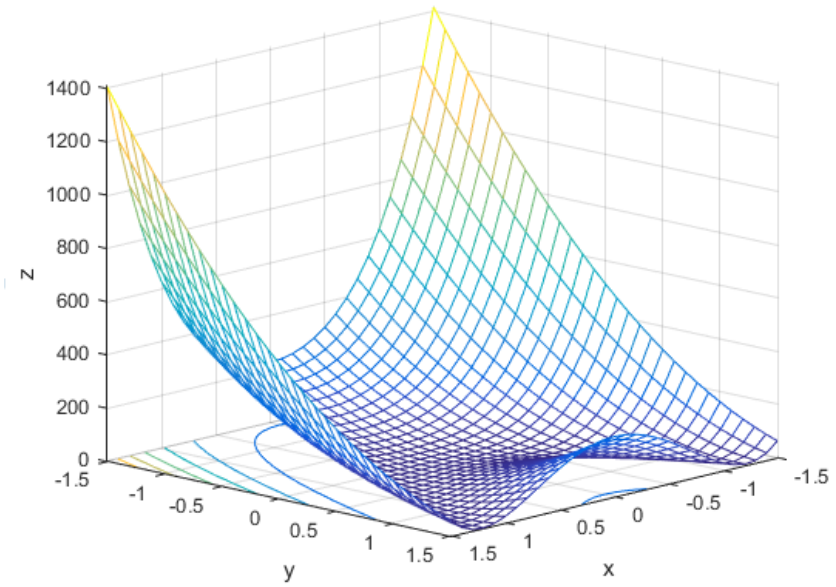
Obr. 4.1 – První De Jongova

4.1.2 Druhá De Jongova funkce

Druhá De Jongova funkce, také známá pod názvem Rosenbrockovo sedlo a banánové údolí, je již o něco složitější než předešlá funkce. Funkce má jedno minimum nacházejí se v bodě $x = (1, 1,\ldots, 1)$, ve kterém je funkční hodnota $f(x) = 0$. Náročnost této funkce spočívá v umístění minima, které se nachází v zahnutém údolí o malém spádu. Testovací prostor pro funkci je obvykle vymezen podmínkou $x_i \in [-2{,}048; 2{,}048]$. Funkce je definována následovně

$$f(x) = \sum_{i=1}^{n-1}\left[100\left(x_i^2 - x_{i-1}\right)^2 + \left(1 + x_i\right)^2\right]. \tag{4.2}$$

Graf pro tuto funkci je uveden na obr. 4.2.



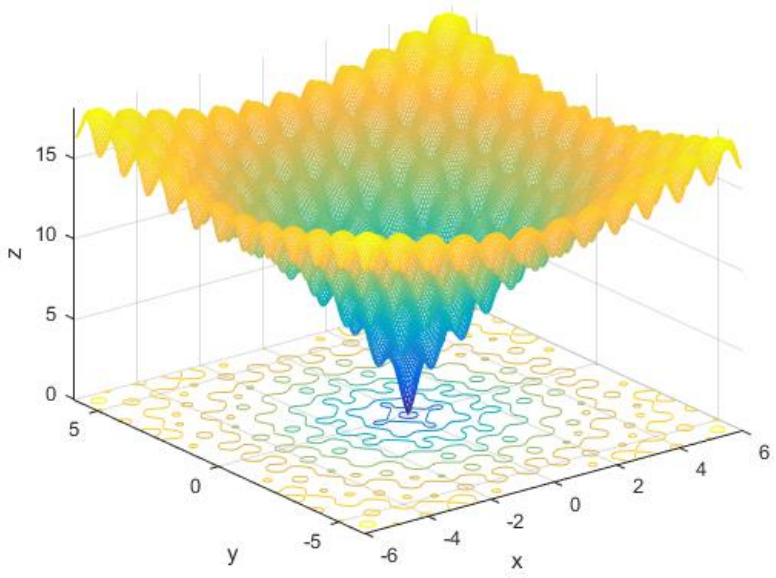
Obr. 4.2 – Druhá De Jongova funkce

4.1.3 Ackleyho funkce

Ackleyho funkce je na rozdíl od předchozích funkcí již funkcí s více extrémy, i když tvar funkce podobný nálevce tomu nemusí napovídat. Globální minimum funkce se nachází v bodě $x = (0, 0,\ldots, 0)$, ve kterém je funkční hodnota $f(x) = 0$. Funkce je kvalifikována jako středně obtížná, s mnoha lokálními minimy. Testovací prostor pro funkci je obvykle vymezen podmínkou $x_i \in [-30; 30]$. Funkce je definována následovně

$$f(x) = -20\exp\left(-0{,}02\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + \exp(1). \tag{4.3}$$

Graf pro tuto funkce je uveden na obr. 4.3.



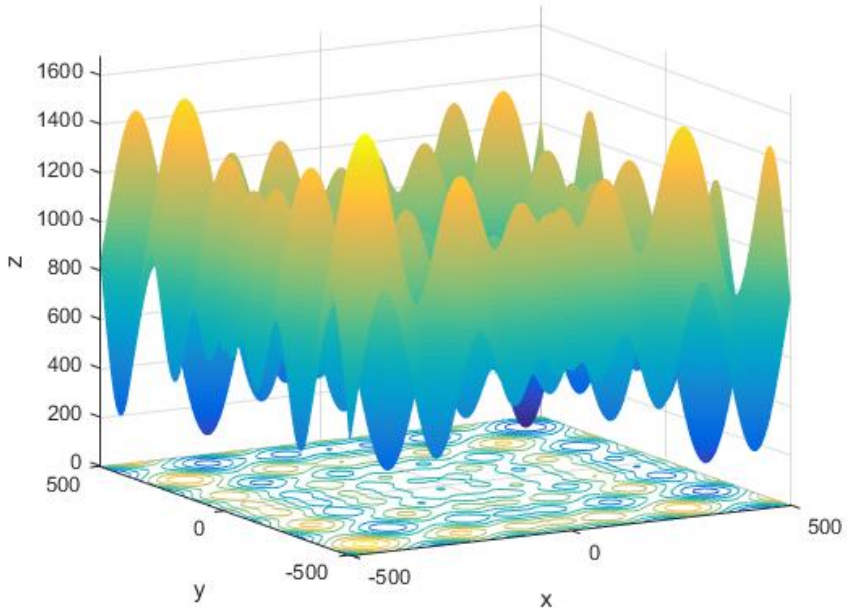
Obr. 4.3 – Ackleyho funkce

4.1.4 Schewefelova funkce

Schwefelova funkce je středně obtížnou funkcí (Tvrdík, 2004). Globální minimum funkce se nachází v bodě x = (420,9687,…, 420,9687), ve kterém je funkční hodnota $f(x)$ = 0. Testovací prostor pro funkci je obvykle vymezen podmínkou $x_i \in$ [-500; 500]. Funkce je definována následovně

$$f(x) = 418{,}9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|}) . \tag{4.4}$$

Graf pro tuto funkci se nachází na obr. 4.4.



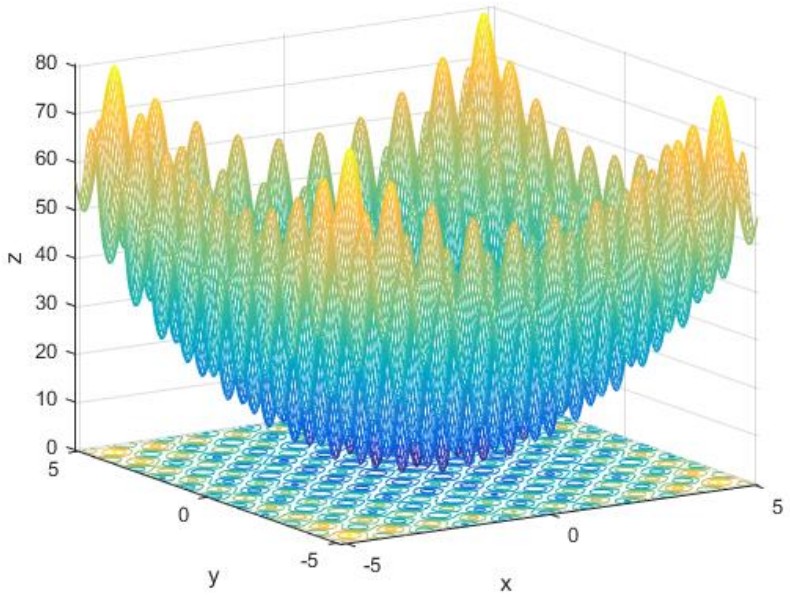
Obr. 4.4 – Schwefelova funkce

4.1.5 Rastriginova funkce

Rastriginova funkce je považována za obtížnou funkci s několika minimy a nalezení jejího minima je považováno za obtížnou úlohu globální optimalizace (Tvrdík, 2004). Globální minimum této funkce se nachází v bodě $x = (0, 0,\ldots, 0)$, ve kterém je funkční hodnota $f(x) = 0$. Testovací prostor pro funkci je obvykle vymezen podmínkou $x_i \in [-5{,}12; 5{,}12]$. Funkce je definována následovně

$$f(x) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right]. \tag{4.5}$$

Graf pro tuto funkci je uveden na obr. 4.5.



Obr. 4.5 – Rastriginova funkce

4.2 TESTOVÁNÍ VLASTNOSTÍ NAVRŽENÝCH METOD

V následujícím oddíle je popsán způsob ověření vlastností navržených metod. První část popisuje způsob provedení. V druhé části jsou pak prezentovány výsledky tohoto testování.

4.2.1 Zvolené parametry experimentů

Pro každou z testovaných metod bylo nutné zvolit několik parametrů, které mají vliv na testování daných metod, popsaných v kapitole 3. Každá z testovaných metod byla otestována na jedné ze čtyř testovacích funkcí (Ackleyho, Rastriginova, Rosenbrockova a Schwefelova), které jsou popsány v kapitole 4.1. Rosenbrockova funkce byla do testování zařazena, i přesto že jde o unimodální funkci, k ověření funkčnosti vytvořených metod.

Každá z metod byla na dané funkci otestována na testovacím prostoru, který je pro každou z vybraných funkcí uveden v kapitole 4.1 (pro všechny metody testované na jedné funkci je tento prostor stejný).

Testování bylo provedeno pro dvě rozdílné dimenze a to $n = 3$ a $n = 10$. Zvolená dimenze ovlivní velikost vytvořené populace bodů, jelikož počet bodů dané populace je nastaven na desetinásobek zvolené dimenze. Každé testování dané metody na testované

funkci pro vybranou dimenzi bylo opakováno desetkrát, přičemž při každém opakování byla vygenerována náhodná počáteční populace bodů.

Ukončovací podmínka byla vždy stejná a to dosažení maximálního počtu iteračních kroků *k*. Jeden iterační krok představuje vypočtení nového minima. Jako maximální počet iteračních kroků byla zvolena hodnota $k = 10\,000$ pro $n = 3$ a $k = 30\,000$ pro $n = 10$.

Hodnoty parametrů, které se během testování měnily v závislosti na zvolené testovací funkci, metodě a dimenzi jsou shrnuty v tab. 4.1.

Tab. 4.1 – Parametry experimentu

Funkce	Testovací prostor	Dimenze	Velikost populace	Metoda
Rosenbrockova	[-2,048; 2,048]	3	30	V1
			30	V2
			30	V3
		10	100	V1
			100	V2
			100	V3
Ackleyho	[-30; 30]	3	30	V1
			30	V2
			30	V3
		10	100	V1
			100	V2
			100	V3
Rastriginova	[-5,12; 5,12]	3	30	V1
			30	V2
			30	V3
		10	100	V1
			100	V2
			100	V3

Tab. 4.1 – Parametry experimentu – pokračování

Funkce	Testovací prostor	Dimenze	Velikost populace	Metoda
Schwefelova	[-500; 500]	3	30	V1
			30	V2
			30	V3
		10	100	V1
			100	V2
			100	V3

4.2.2 Výsledky experimentů

V tomto pododdíle jsou uvedeny výsledky provedených experimentů. Výsledky jsou prezentovány ve formě grafů. Pro každou testovací funkci je zobrazeno několik grafů. Pro každou metodu u dané funkce je zobrazen graf, na kterém je zobrazen pokles funkční hodnoty v minimu pro všechna provedená opakování v dané dimenzi. Dále je pro všechny metody u dané funkce a dimenze je uveden graf obsahující průměrné hodnoty poklesu funkční hodnoty v minimu pro všechny varianty (s výjimkou Rosenbrockovy funkce pro $n = 3$).

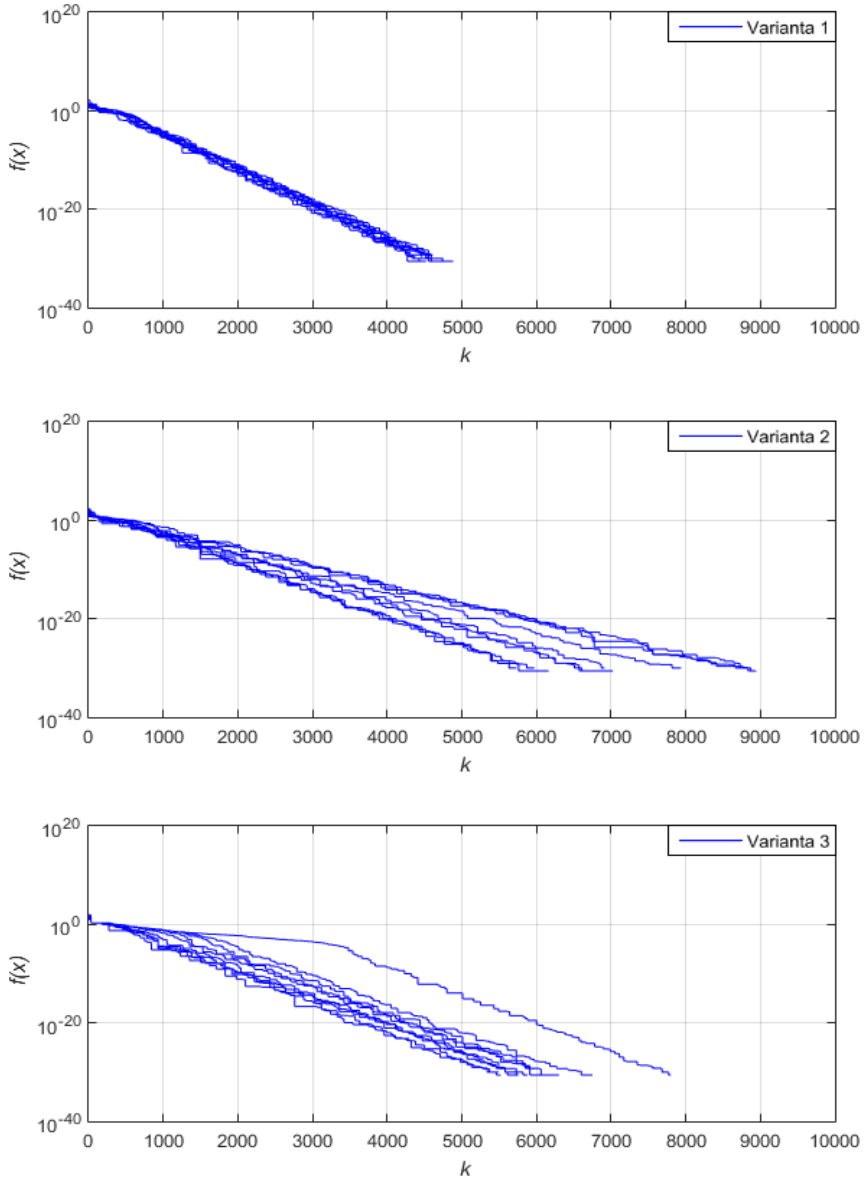
Pro přehlednější zobrazení je u většiny grafů namísto poklesu funkční hodnoty zobrazen pokles logaritmu funkční hodnoty.

a) Minimalizace Rosenbrockovy funkce, $n = 3$

Na obr. 4.6 jsou zobrazeny grafy poklesu logaritmů funkčních hodnot všech variant pro Rosenbrockovu funkci v dimenzi $n = 3$.

Jak je vidět z grafů, tak minimum této funkce v daném počtu kroků bez problémů nalezla každá z variant. Jak je vidět z prvního grafu na obr. 4.6 nejlépe si hledání minima vedla varianta č. 1. Této variantě trvalo nejdéle nalezení minima při druhém opakování ($k = 4885$). Z grafu je také vidět, že všechna opakování u této varianty tvoří kompaktní oblast.

Naopak nejhůře si při hledání minima vedla varianta č. 2, které trvalo nalezení minima nejdéle při osmém opakování ($k = 8944$). Z grafu je také patrné, že její opakování tvoří méně kompaktní oblast, než u předešlé varianty.

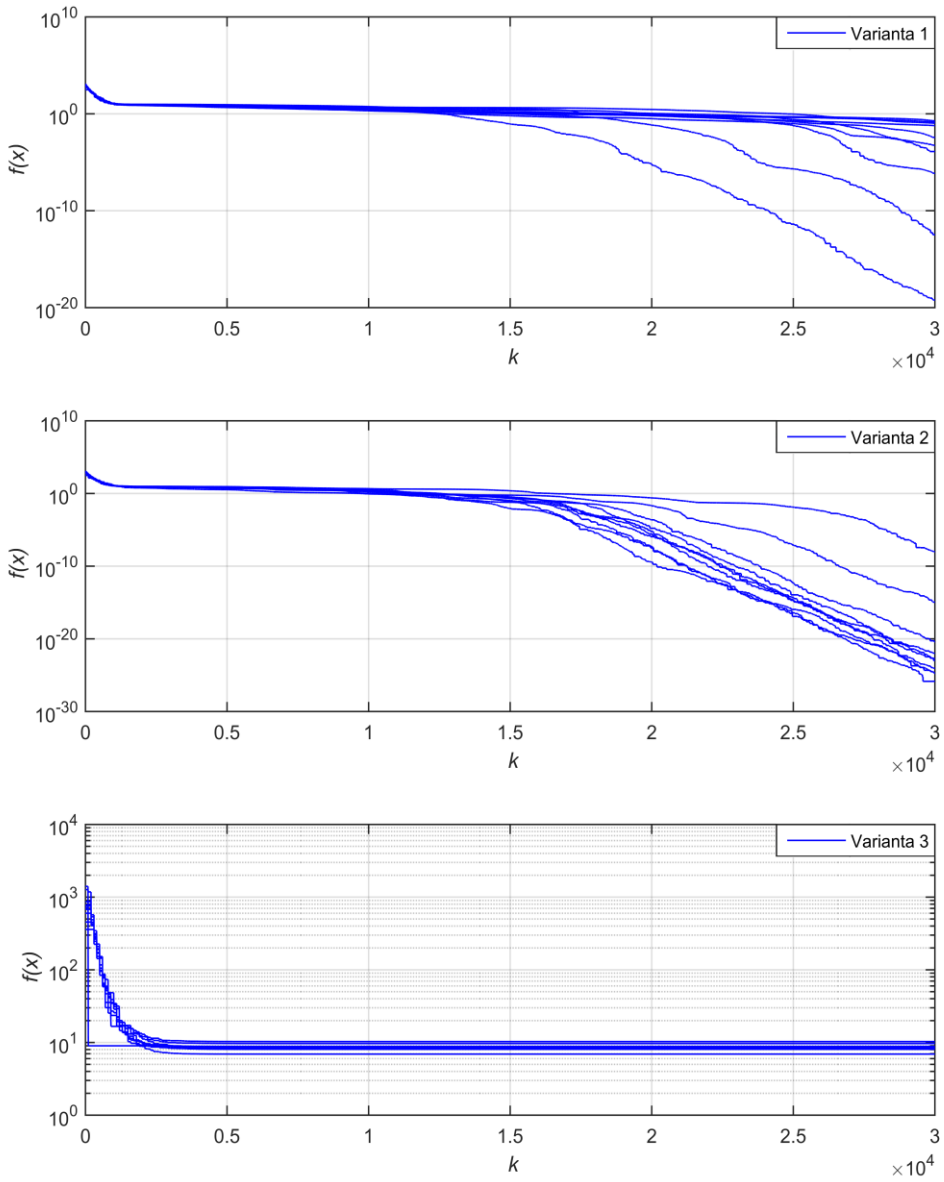


Obr. 4.6 – Pokles logaritmů funkčních hodnot, Rosenbrock. funkce, $n = 3$

b) Minimalizace Rosenbrockovy funkce, $n = 10$

Na obr. 4.7 jsou zobrazeny grafy poklesu logaritmů funkčních hodnot všech variant pro Rosenbrokovu funkci v dimenzi $n = 10$.

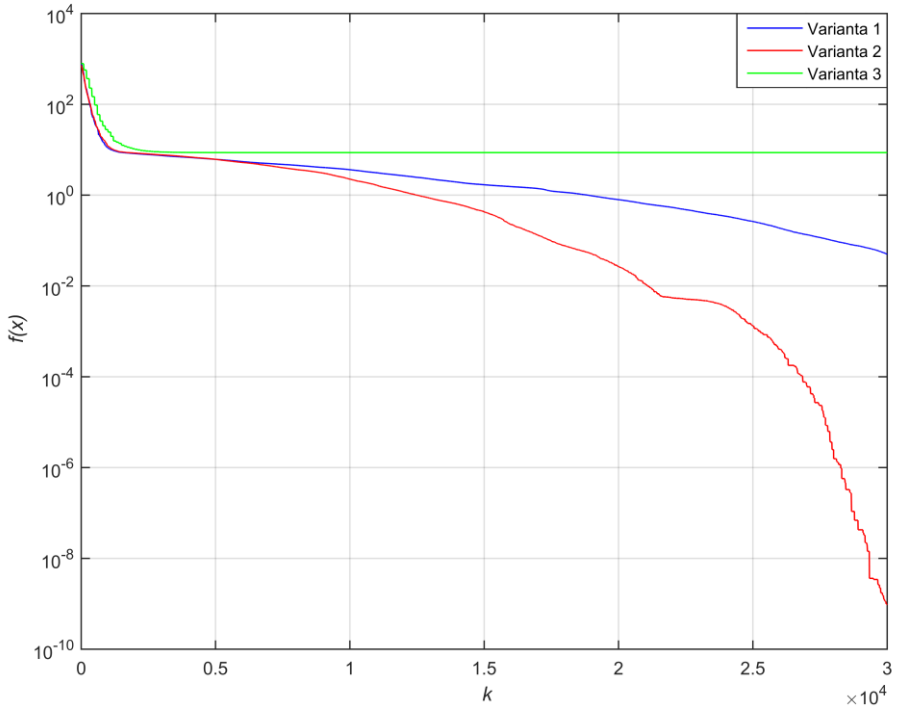
Najít jediné minimum této funkce v desáté dimenzi nedokázala spolehlivě před dosažením maximálního počtu kroků žádná z variant. Důvodem pro nenalezení minima před dosažením maximálního počtu kroků bude tvar oblasti, ve které leží jediné minimum funkce (zahnuté údolí s malým spádem), která komplikuje nalezení minima ve vyšších dimenzích.



Obr. 4.7 – Pokles logaritmů funkčních hodnot, Rosenbrock. funkce, $n = 10$

Z grafu je vidět, že varianta č. 1 a č. 2 mají tendenci přibližně od $k = 10\,000$ nacházet lepší minimu. Dá se očekávat, že při zvětšení maximálního počtu kroků by alespoň varianta č. 2 měla nalézt známé minimum, ovšem další zvýšení maximálního počtu kroků se nejeví jako efektivní.

Jak je vidět z grafu na obr. 4.8, kde je zobrazen pokles logaritmů průměrů funkčních hodnot na Rosenbrockově funkci v desáté dimenzi, nejlépe se při hledání minima dařilo variantě č. 2. Průměrná funkční hodnota v minimu nalezeném touto variantou při ukončení výpočtu je $f(x) = 9{,}54 \cdot 10^{-10}$.



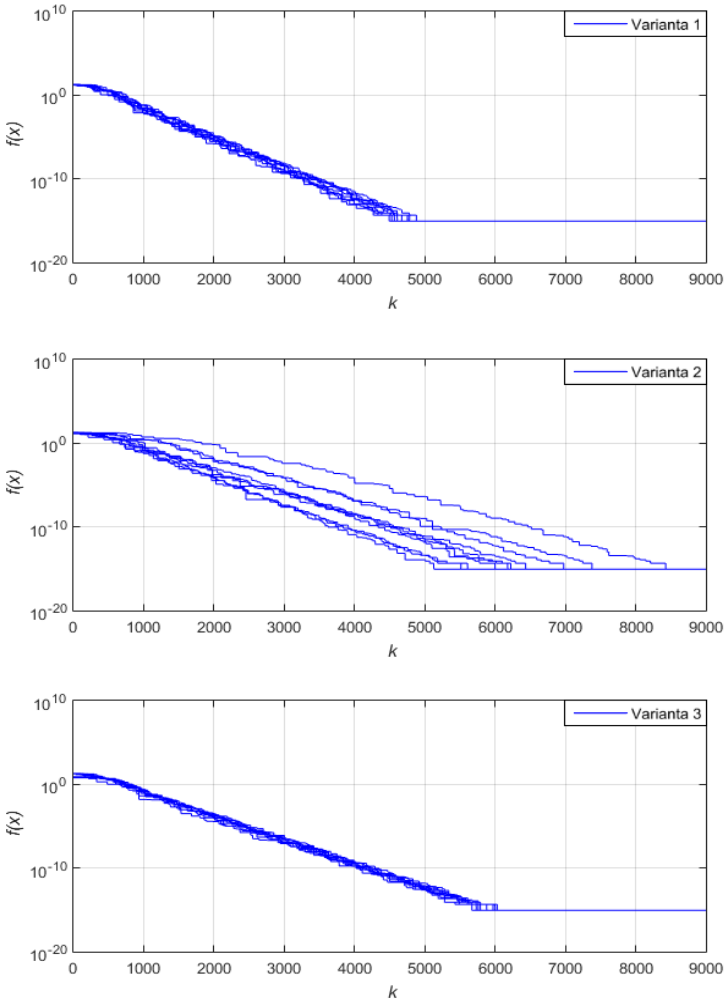
Obr. 4.8 – Pokles průměru logaritmů funkčních hodnot, Rosenbrock. funkce, $n = 10$

c) Minimalizace Ackleyho funkce, $n = 3$

Na obr. 4.9 jsou zobrazeny grafy poklesu logaritmů funkčních hodnot všech variant Ackleyho funkci v dimenzi $n = 3$.

Z grafů na obr. 4.9 je patrné, že minimum funkce dokázali najít všechny varianty. Jako nejlepší opět vychází varianta č. 1., které trvalo nejdéle nalezení minima při pátém opakování ($k = 4884$). Nejkratší dobu trvalo nalezení minim při třetí opakování ($k = 4494$). Rozdíl mezi nejlepším a nejhorším opakováním je jen 390 iteračních kroků a jak je vidět z grafu, všechna opakování tvoří poměrně kompaktní oblast.

Z grafů je vidět, že nejhůře si zde vedla varianta č. 2. Oproti předchozí variantě jí nalezení minima trvalo delší dobu, přičemž nejdéle trvalo nalezení minima při osmém opakování ($k = 8429$), které bylo ze všech opakování výrazně nejhorší. Nejkratší dobu trvalo nalezení minima při druhém opakování ($k = 5128$), což je déle, než u nejhoršího opakování u předchozí varianty. Rozdíl mezi nejlepším a nejhorším opakováním je tak 3301 iteračních kroků a i z grafu je patrné, že opakování již netvoří tak kompaktní oblast jako v předchozím případě.



Obr. 4.9 – Pokles logaritmů funkčních hodnot, Ackleyho. funkce, $n = 3$

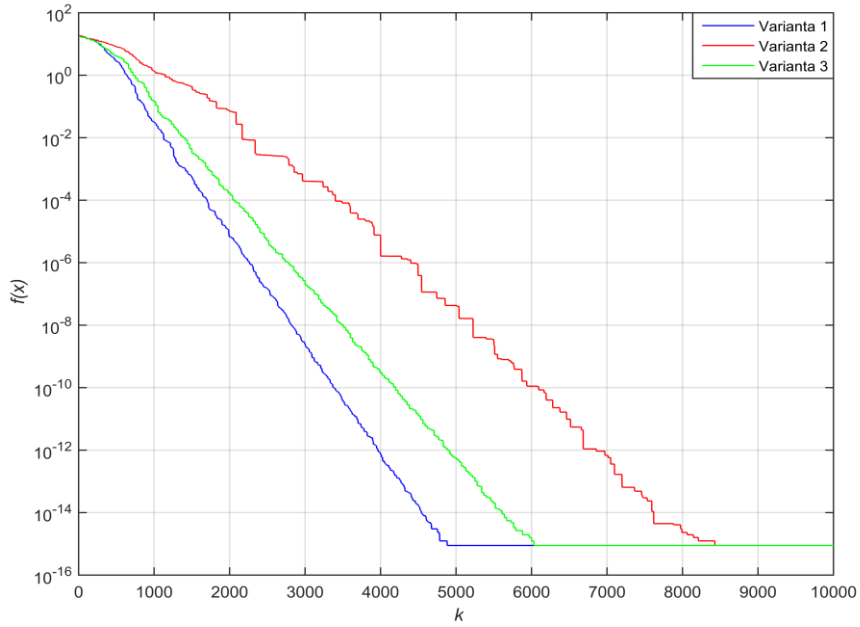
Pro Ackleyho funkci je nutné uvést, že ačkoliv skutečná funkční hodnota v globální minimu této funkce je $f(x) = 0$, tak během testování je vždy pro tuto funkci v globálním minimu funkční hodnota $f(x) = 8{,}88 \cdot 10^{-16}$.

I varianty č. 3 trvalo nejdéle nalezení minima při druhém opakování ($k = 6031$). Nejdříve bylo nalezeno minimum shodně při třetím a desátém opakování ($k = 5671$), což je déle, než u nejhoršího opakování u předchozí varianty.

Ačkoliv nalezení globálního minima trvalo déle než u první varianty, tak zobrazená oblast je nejvíce kompaktní ze všech variant v této dimenzi. Rozdíl mezi nejlepším a nejhorším opakováním je 360 iteračních kroků.

Na obr. 4.10 se nachází graf zobrazující pokles logaritmu průměru funkčních hodnot pro každou ze tří variant, testovaných na Ackleyho funkci pro dimenzi $n = 3$. Všechny varianty nalezly globální minimum před dosažením maximálního počtu kroků. Z grafu je

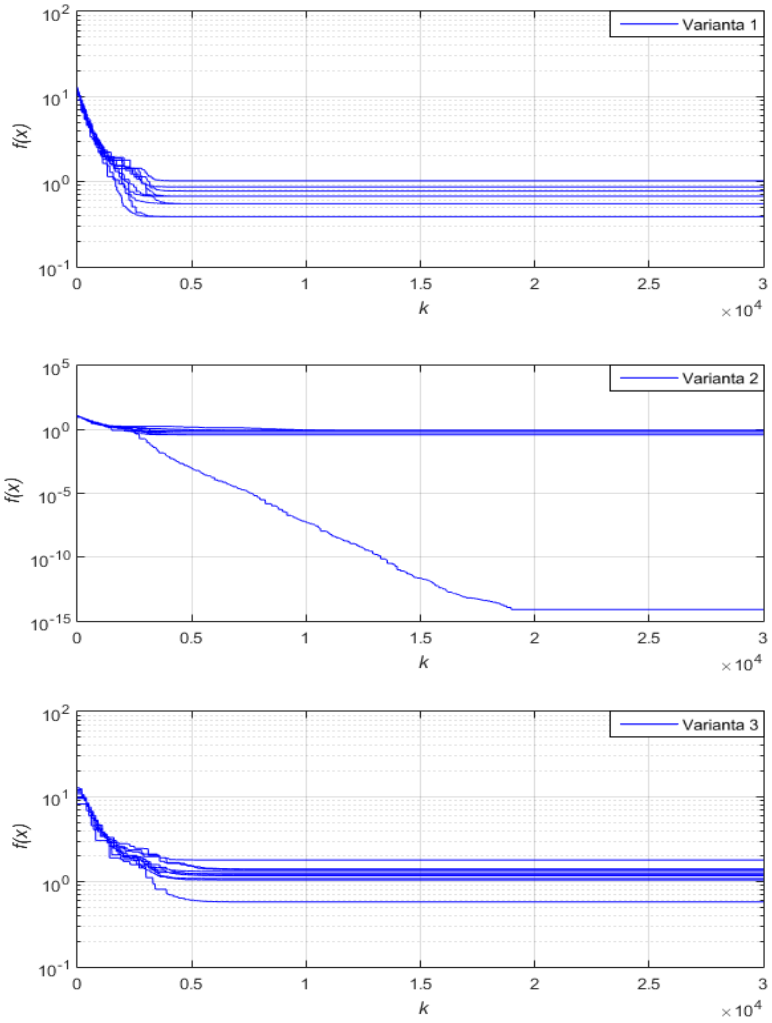
patrné, že výrazně nejhůře si z testovaných variant vedla varianta č. 2., nahrazující nejhorší bod simplexu na jeho pozici v populaci. Naopak nejlépe si vedla varianta č. 1., následovaná variantou č. 3.



Obr. 4.10 – Poklesy průměru poklesu logaritmu funkčních hodnot, Ackley. funkce, $n = 3$

d) Minimalizace Ackleyho funkce, $n = 10$

Na obr. 4.11 jsou zobrazeny grafy poklesu logaritmů funkčních hodnot všech variant pro Ackleyho funkci v dimenzi $n = 10$.

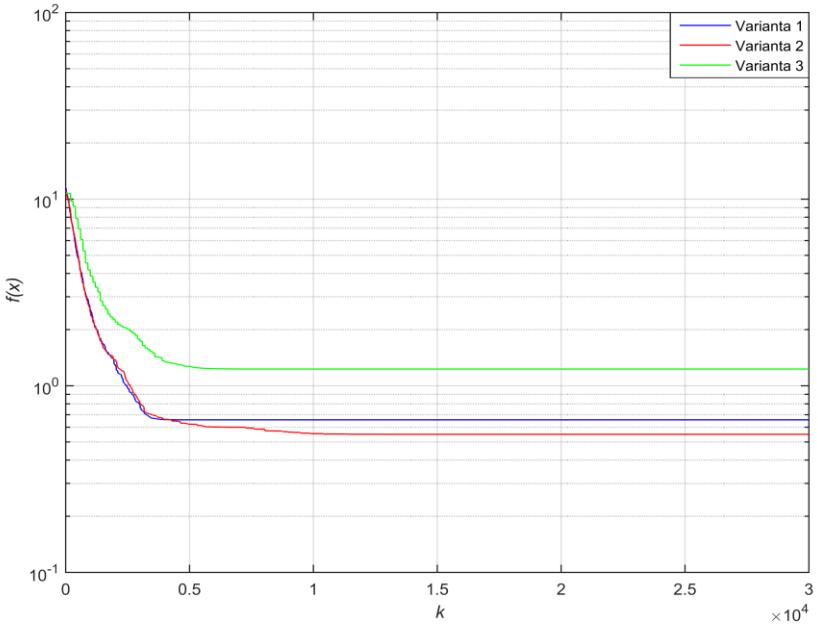


Obr. 4.11 – Pokles logaritmů funkčních hodnot, Ackleyho funkce, $n = 10$

Z grafu je patrné, že před dosažením maximálního počtu kroků nebylo dosaženo minima ani u jedné z variant. Nejlepší výsledek před dosažení maximálního počtu kroků byl nalezen variantou č. 2 při jejím desátém opakování, kdy byla dosažena funkční hodnota v minimu $f(x) = 7{,}99 \cdot 10^{-15}$.

Jak je z grafu vidět, tak všechny varianty, kromě jednoho opakování u varianty č. 2 ukončily hledání minima již před krokem $k = 10\,000$ a funkční hodnoty nalezených minim se až do ukončení výpočtu nemění.

Na obr. 4.12 je graf zobrazující pokles logaritmu průměru funkčních hodnot pro každou ze tří variant užitých pro Ackleyho funkci v dimenzi $n = 10$.



Obr. 4.12 – Pokles logaritmu průměru funkčního hodnot, Ackleyho funkce, $n = 10$

Z tohoto grafu je patrné, že pro desátou dimenzi všechny varianty nedokáží nalézt globální minimum s dostatečnou rychlostí. Z grafu se jeví jako nejlepší varianta č. 2, která sice nenalezla minimum ani v jednom případě, ale dokázala se minimu během jednoho z opakování výrazněji přiblížit na rozdíl od varianty č. 1 a č. 3. Jako nejhorší vychází varianta č. 3.

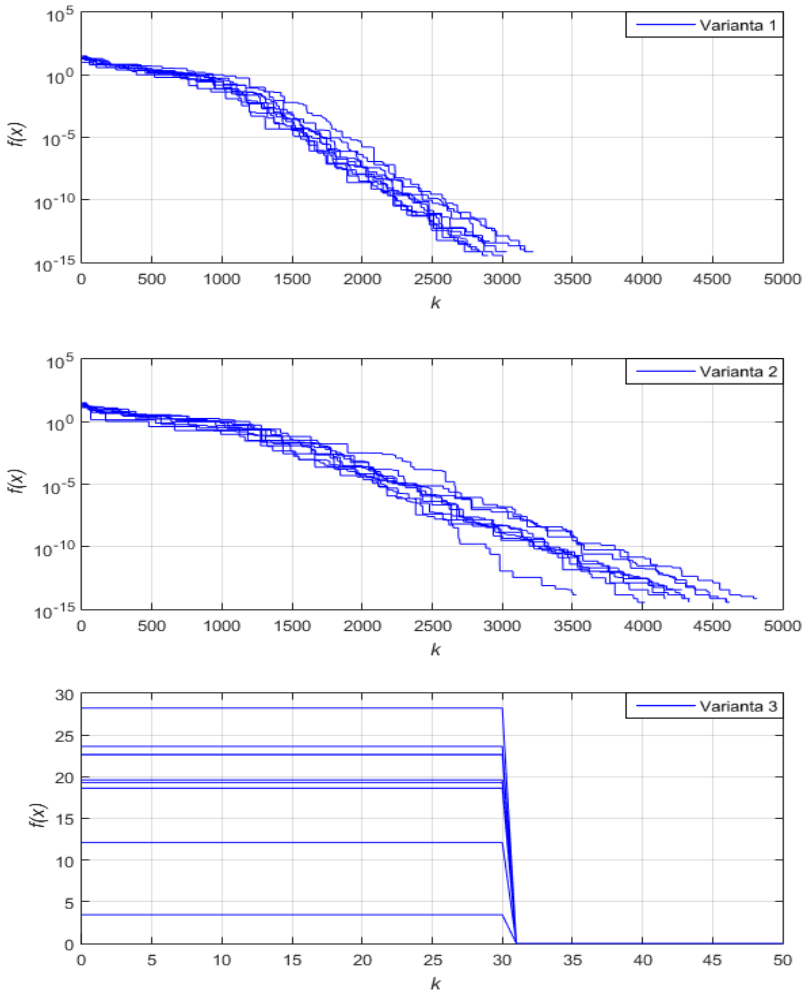
e) Minimalizace Rastriginovy funkce, $n = 3$

Na obr. 4.13 jsou zobrazeny grafy poklesu logaritmů funkčních hodnot všech variant na Rastriginově funkci v dimenzi $n = 3$.

Z grafů na obr. 4.13 je vidět, že problém s nalezením minima neměla žádná z variant. Jako nejlepší se v tomto případě jeví varianta č. 3, u které došlo k nalezení minima při každém opakování již při třicátém kroku. Z toho důvodu je pro přehlednost na posledním grafu na obr. 4.13 (zobrazující pokles logaritmu funkčních hodnot u varianty č. 3) zobrazeno na ose x pouze 50 kroků.

Je nutné podotknout, že zaznamenané chování u této varianty nebylo omezeno pouze na tento jediný případ, ale podobně se tato varianta na Rastriginově funkci pro $n = 3$, chovala i při ostatních experimentech. Z provedených experimentů vyplývá, že tato metoda dokáže najít minimum Rastriginovy funkce ve třetí dimenzi, již při třicátém kroku s pravděpodobností 92 %.

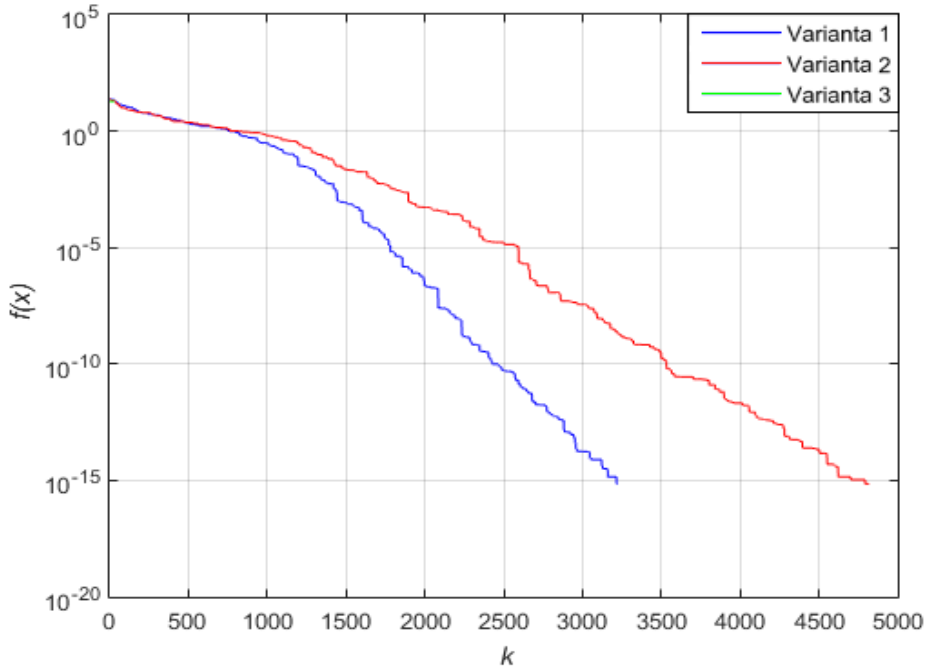
Jako druhá nejlepší vychází varianta č. 1, která neměla problém s nalezením minima této funkce a pokles logaritmu funkční hodnoty v minimu je poměrně strmý. Nejdéle trvalo nalezení minima při druhém opakování (k = 3221). Naopak nejdříve bylo dosaženo minima během třetího opakování (k = 2780).



Obr. 4.13 – Pokles logaritmů funkčních hodnot, Rastriginova funkce, n = 3

Jak je patrné z grafu, jako nejhorší zde vychází varianta č. 2., která stejně jako zbylé varianty neměla problém dosáhnout globálního minima v každém opakování. Ovšem pokles logaritmu funkční hodnoty v minimu již není tak strmý jako u varianty č. 1 a nalezení minima touto variantou trvalo déle. Nejdéle trvalo nalezení minima při prvním opakování (k = 4819). Naopak nejdříve bylo dosaženo minima během třetího opakování (k = 3530).

Na následujícím obr. 4.14 je graf zobrazující pokles logaritmu průměru funkčních hodnot pro každou ze tří variant, užitých pro Rastriginovu funkci ve třetí dimenzi.



Obr. 4.14 – Pokles logaritmu průměru funkčního hodnot, Rastriginova funkce, $n = 3$

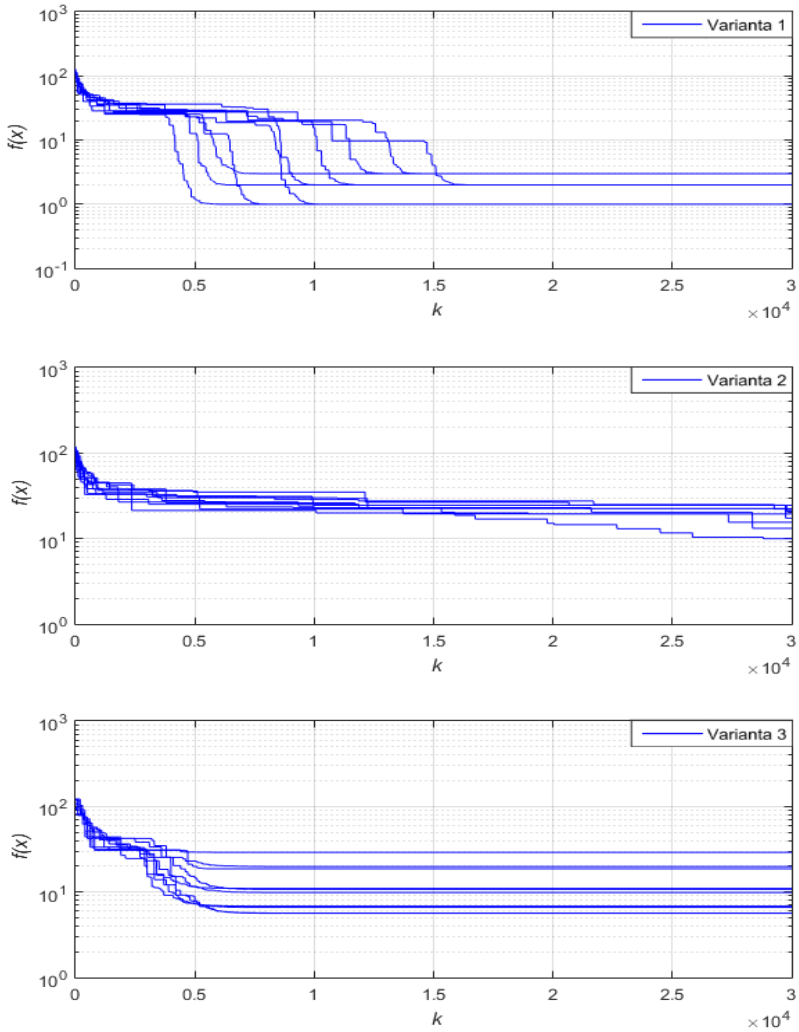
Z grafu je vidět, že žádná z variant neměla problém s nalezením globálního minima. Ze všech variant jako nejlepší pro Rastriginovu funkci vychází varianta č. 3 (pracující s dvojící populací), která byla schopna nalézt minimu s 92% pravděpodobností již v kroku $k = 30$. Varianta č. 3 ovšem v tomto grafu není zobrazena, jelikož její křivka zobrazující pokles logaritmu průměru funkčních hodnot, končí s krokem $k = 30$.

f) Minimalizace Rastriginovy funkce, $n = 10$

Na obr. 4.15 jsou zobrazeny grafy poklesu logaritmů funkčních hodnot všech variant pro Rastriginovu funkci v dimenzi $n = 10$.

Na tomto grafu je vidět, že stejně jako u Ackleyho funkce nebyla žádná z variant schopna nalézt minimum před dosažením maximálního počtu kroků. Podle nejnižší dosažené průměrné hodnoty při ukončení výpočtu vychází jako nejlepší varianta č. 1. Minimum s nejlepší funkční hodnotou nalezené touto variantou při ukončení výpočtu, bylo dosaženo shodně při osmém, devátém a desátém opakování. Funční hodnota v tomto minimu je $f(x) = 0{,}99$. Průměrná funkční hodnota dosažená touto variantou při ukončení výpočtu je $f(x) = 1{,}98$.

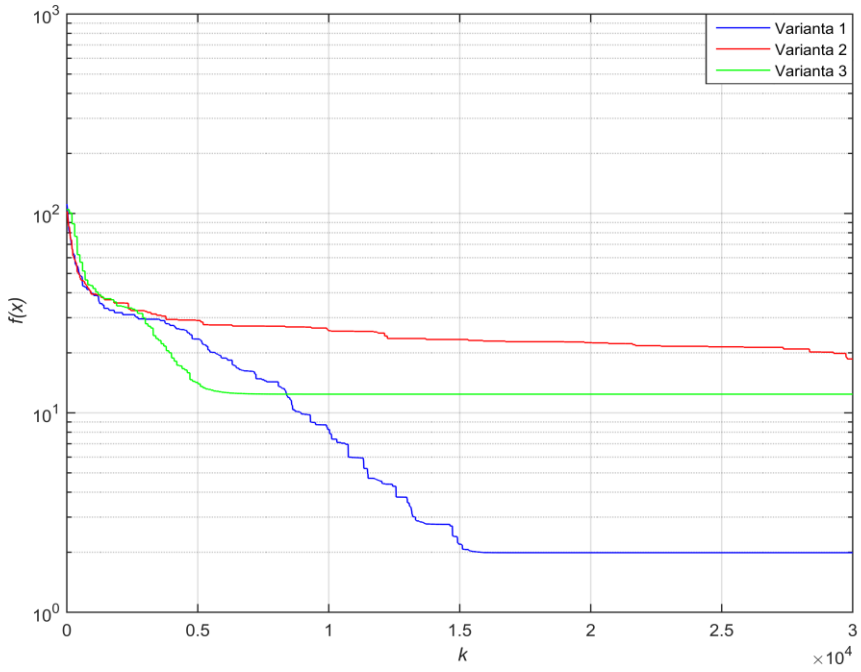
Jako nejhorší z variant se zde jeví varianta č. 2, jejíž průměrná funkční hodnota dosažená při ukončení výpočtu je ze všech variant nejvyšší a to $f(x) = 18{,}62$.



Obr. 4.15 – Pokles logaritmů funkčních hodnot, Rastriginova funkce, $n = 10$

Jak je patrné při pohledu na graf na obr. 4.15, tak i přes to, že varianta č. 2 vychází podle hodnot nalezených minim jako nejhorší, tak jako jediná z variant pokračovala v hledání minima po celou dobu výpočtu.

Na obr. 4.16 se nachází graf zobrazující pokles logaritmu průměru funkčních hodnot pro každou ze tří variant užitých pro Rastriginovu funkci v desáté dimenzi.



Obr. 4.16 – Pokles logaritmu průměru funkčního hodnot, Rastriginova funkce, $n = 10$

Z grafu na obr. 4.16 je patrné, že pro desátou dimenzi žádná z variant nedokázala nalézt globální minimum před dosažením maximálního počtu kroků. I přesto, že nedokázala ve stanoveném počtu kroků nalézt globální minimu, jeví se z grafu jako nejlepší varianta č. 1., která má ze všech variant největší spád.

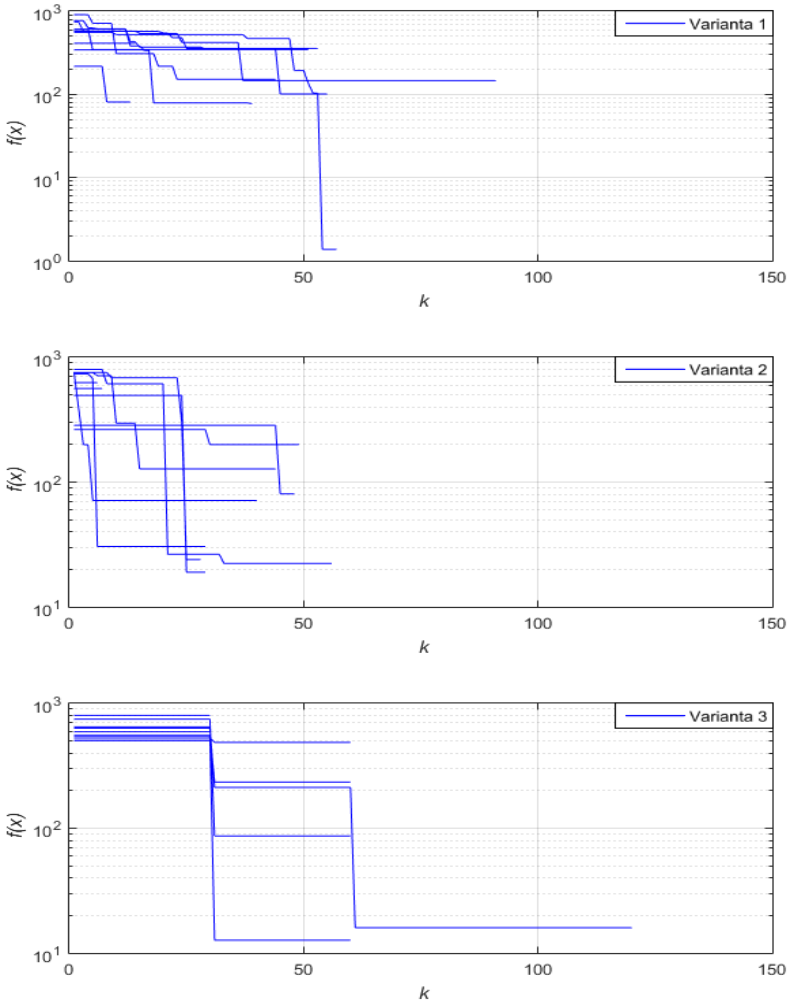
g) Minimalizace Schwefelovy funkce, $n = 3$

Na obr. 4.17 jsou zobrazeny grafy poklesu logaritmů funkčních hodnot všech variant, pro Schwefelovu funkci v dimenzi $n = 3$.

Pro Schwefelovu funkci došlo jak ve třetí, tak v desáté dimenzi k selhání všech testovaných variant, které nebyly schopny nalézt globální minimum funkce nebo se k němu výrazněji přiblížit.

Graf zobrazující pokles logaritmů průměrů funkčních hodnot všech variant, pro Schwefelovu funkci ve třetí dimenzi je na obr. 4.18. Průměrná funkční hodnota v minimu pro Schwefelovu funkci v této dimenzi po dosažení maximálního počtu kroků je $f(x) = -1{,}90 \cdot 10^{69}$ pro variantu č. 1, $f(x) = -2{,}00 \cdot 10^{105}$ pro variantu č. 2 a $f(x) = -2{,}51 \cdot 10^{5}$ pro variantu č. 3.

Všechny varianty jsou sice schopné nalézt bod s velmi nízkou funkční hodnotu, ale žádný z těchto bodů není známé globální minimum této funkce (nebo bod ležící v jeho blízkosti), které se pro $n = 3$ nachází v bodě $x = [420{,}9687; 420{,}9687; 420{,}9687]$ a má funkční hodnotu $f(x) = 0$.



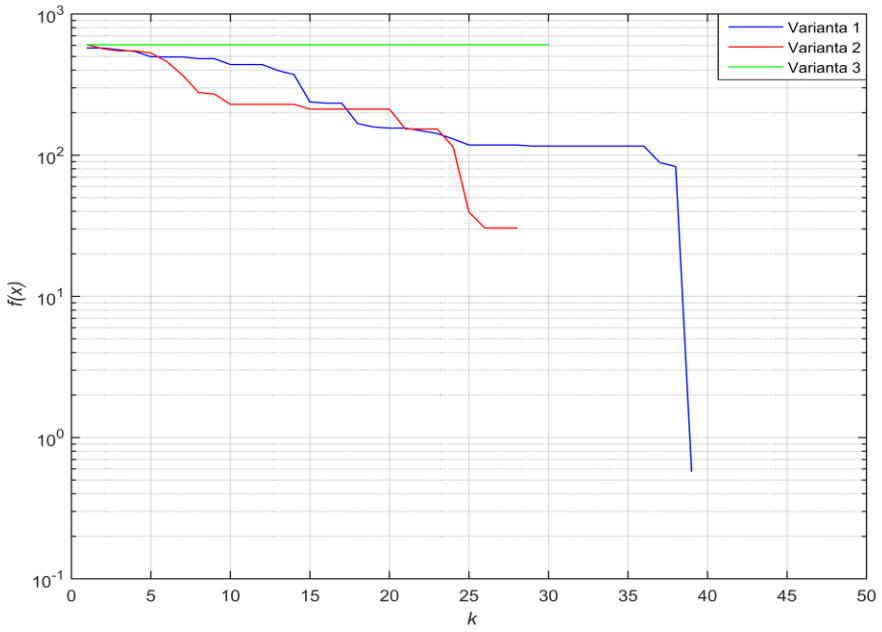
Obr. 4.17 – Pokles logaritmů funkčních hodnot, Schwefelova funkce, $n = 3$

h) Minimalizace Schwefelovy funkce, $n = 10$

Na předposledním grafu na obr. 4.19 jsou zobrazeny grafy poklesu logaritmů funkčních hodnot všech variant, pro Schwefelovu funkci v dimenzi $n = 10$.

Na posledním obr. 4.20 je zobrazen pokles průměrů funkčních hodnot pro deset opakování všech variant pro Schwefelovu funkci v desáté dimenzi. Stejně jako v předchozí situaci i zde je vidět, že všechny varianty selhávají v nalezení známého globálního minima funkce.

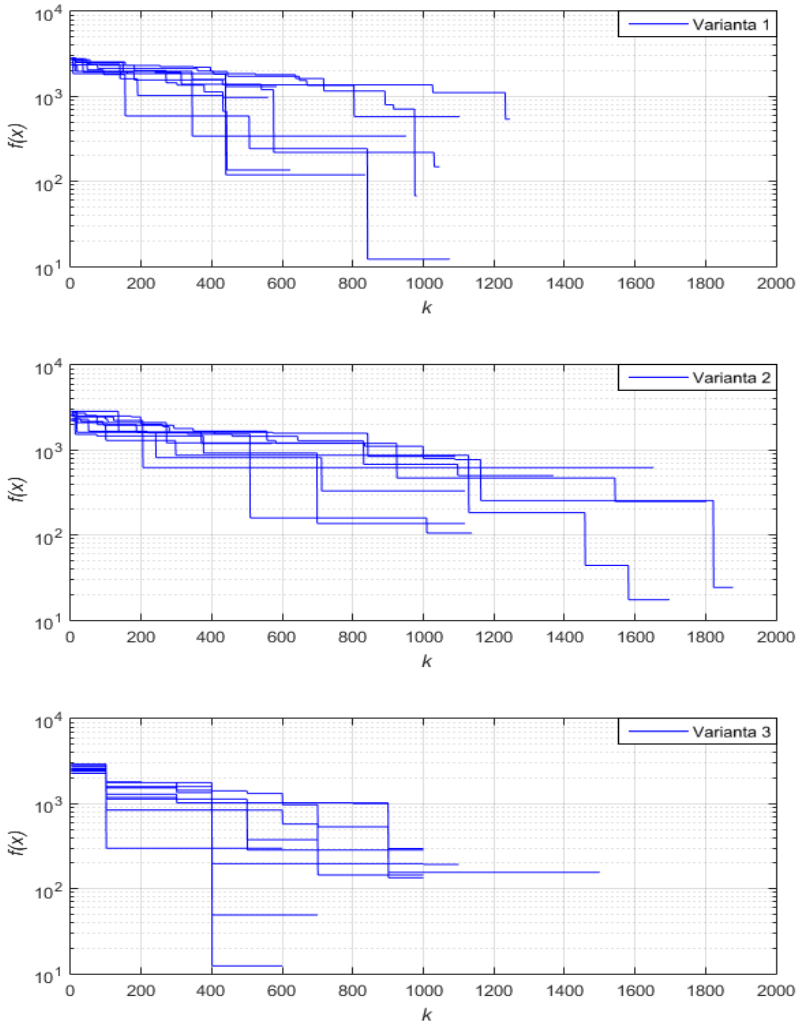
Nalezené průměrné funkční hodnoty v minimu pro Schwefelovu funkci v desáté dimenzi po dosažení maximálního počtu kroků jsou: $f(x) = -7{,}50 \cdot 10^4$ pro variantu č. 1, $f(x) = -2{,}41 \cdot 10^4$ pro variantu č. 2 a $f(x) = -2{,}50 \cdot 10^4$ pro variantu č. 3.



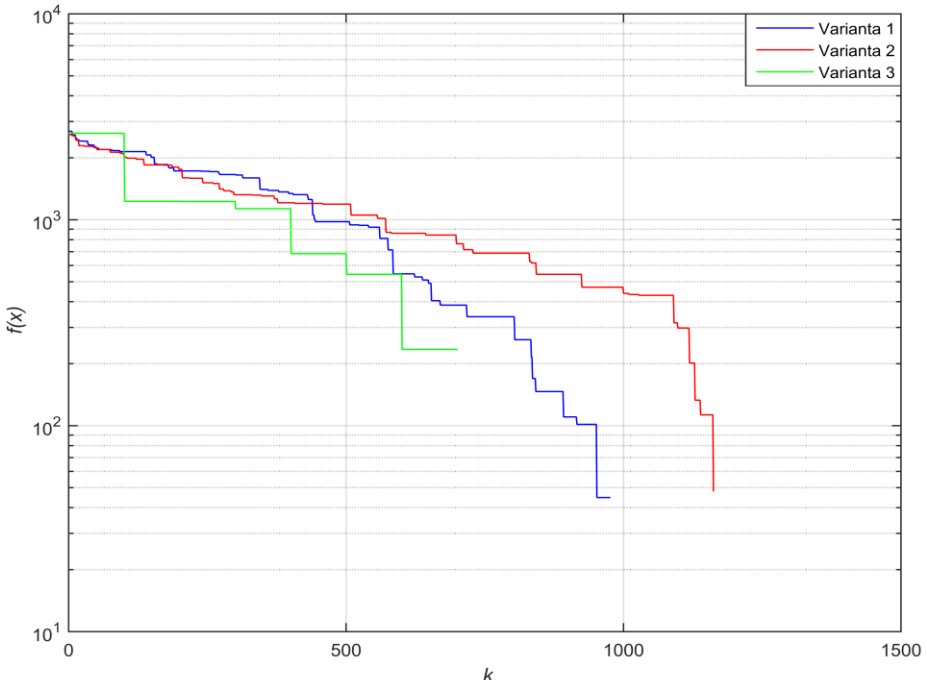
Obr. 4.19 – Pokles logaritmu průměru funkčního hodnot, Schwefelov funkce, $n = 3$

Pravděpodobnou příčinou selhání všech testovaných variant na Schwefelově funkci bez ohledu na zvolenou dimenzi, bude velmi komplikovaný tvar funkce s velkým počtem lokálních minim a umístění globálního minima, na okraji prohledávaného prostoru.

Naproti tomu u Ackleyho a Rastriginovy funkce dokázaly všechny varianty globální minimum nalézt spolehlivě, alespoň ve třetí dimenzi. Vliv na nalezení globálního minima těchto funkcí, které mají stejně jako Schwefelova funkce několik lokálních minim, je umístění globálního minima ve středu prohledávaného prostoru a tvar obou funkcí. Obě funkce ačkoliv obsahují více lokálních minim, se díky svému tvaru ze všech stran svažují směrem ke globálnímu minimu, což zvyšuje šanci na nalezení globálního minima.



Obr. 4.20 – Pokles logaritmů funkčních hodnot, Schwefelova funkce, $n = 10$



Obr. 4.20 – Pokles logaritmu průměru funkčního hodnot, Schwefelova funkce, n = 10

5 PROGRAMOVÉ ŘEŠENÍ

V této kapitole je stručně popsán software použitý k realizaci popsaných variant metody flexibilního simplexu popsaných v předchozí kapitole. První část této kapitoly popisuje přímo použitý software a možnost vytvoření programů (a funkcí) v jeho prostředí. Druhá část této kapitoly pak popisuje realizaci těchto programů, jejich seznam a stručný popis. Dále je uveden příklad nastavení vybraných parametrů a chování spuštěného programu.

5.1 POUŽITÝ SOFTWARE

Pro vytvoření a následné testování algoritmů pro hledání minima funkcí byl použit software Matlab společnosti MathWorks. Název Matlab je akronymem z anglických slov Matrix laboratory (v češtině maticová laboratoř), který odkazuje na skutečnost, že klíčovým prvkem pro výpočty v Matlabu jsou matice. Matlab je výpočetním a programovacím prostředím a zároveň vyšším programovacím jazykem, vycházejícím z jazyka Fortran. Prostřední Matlab podporuje objektově orientované programování.

Matlab umožňuje, kromě interpretaci příkazů z příkazové řádky a operací se zabudovanými funkcemi, také vytváření grafů a vlastních programů ve formě skriptů (dávek) nebo funkcí. Programy pro prostředí Matlab se ukládají do souboru s příponou *m* (např. generuj_cislo.m), jehož jméno musí být v případě, že jde o funkci stejné jako název dané funkce, aby byla zaručena správná funkčnost. Program neobsahující žádné vstupní parametry se označuje jako skript. Naopak program, který obsahuje vstupní a výstupní parametry se označuje jako funkce.

Takto definovanou funkci může volat jak skript, tak funkce. Volat může funkce i sama sebe. Díky možnosti volání funkcí jinými funkcemi je možné několika úrovňové volání funkcí. Proměnné uvnitř funkce jsou lokální a stejně pojmenované proměnné ve volající funkci či skriptu jimi nejsou přepsány. V následující části bude popsán způsob, jakým je řešena implementace jednotlivých variant popsaných v kapitole č. 3. právě pomocí skriptu a funkcí.

5.2 PROGRAMOVÁ REALIZACE

V této části práce bude popsána samotná realizace v prostředí Matlab a stručně popsáno, k jakému účelu daná funkce slouží. Každá vytvořená funkce obsahuje na druhém řádku (popř. i dalším) komentář, kde je popsán účel dané funkce, vstupní a výstupní parametry.

Pro potřeby této práce byl vytvořen jeden skript a 14 funkcí. Funkce jsou pojmenovány podle činnosti, která je od nich vyžadována.

- **start** – pomocný skript k nastavení parametrů experimentů a zaznamenávání výsledků,
- **spusteniSimplexuPopulace** – funkce sloužící ke spuštění vybrané metody a následnému vypsání sledovaných hodnot,
- **simplex_nahrazeniNejhorsihoPopulaceV1** – upravená simplexová metoda varianta č. 1,
- **simplex_nahrazeniBoduSimplexuV2** – upravená simplexová metoda varianta č. 2,
- **simplex_2populaceV3** – upravená simplexová metoda varianta č. 3,
- **F** – funkce určená pro volbu účelové funkce,
    - **Ackley** – funkce pro výpočet hodnot Ackleyho funkce,
    - **Rastrig** – funkce pro výpočet hodnot Rastriginovy funkce,
    - **Rosenbrock** – funkce pro výpočet hodnot Rosenbrockovy funkce,
    - **Schwefel** – funkce pro výpočet hodnot Schwefelovy funkce,
- **histMatrixS** – pomocná funkce sloužící k zaznamenání změny např. minima,
- **vytvorPopulaci** – funkce určená k vytvoření náhodné populace v daném rozsahu,
- **setrideniPopulace** – funkce určená k setřídění dané populace, bod s nejnižší hodnotou účelové funkce je po setřídění na prvním řádku,
- **porovnaniPopulaci** – funkce určená k porovnání dvou populací, výstupem je jediná populace stejné velikosti jako obě původní populace obsahující nejlepší body z obou populací,
- **vytvorSimplexzPopulace** – funkce určená k vytvoření náhodného simplexu z dané populace, kromě vytvořeného simplexu vrací i záznam o vybraných bodech, který je nutný pro funkci simplex_nahrazeniBoduSimplexuV2.

Jelikož celé programové vybavení bylo konstruováno pouze pro tento experiment a nikoliv jako uživatelský program, je změna parametrů poměrně složitá a vyžaduje zvýšenou pozornost.

Nyní proto bude popsán příklad nastavení parametrů pro tento experiment: Ackleyho funkce, varianta č. 2, prohledávaný prostor [-30; 30], dimenze $n = 10$, s populací o velikosti desetinásobku dimenze, 10 opakování.

Pro testování metody na ackleyho funkci je nutné upravit soubor **F.m** a odstranit komentář na tomto řádku: `hodnota=ackley(x);` poté je nutno tento soubor uložit. Další je úpravu je nutno provést v souboru **vytvorPopulaci.m** a provést změnu hodnoty proměnné `a` na řádku č. 4, podle velikosti prohledávaného prostoru. Poté je opět nutné soubor uložit. Nyní již stačí spustit soubor **start.m**. Po spuštění skript vypíše v příkazovém okně výzvu: „Vyber variantu: 1, 2 nebo 3.“ Po zadání příslušné číslice a potvrzení klávesou ENTER dojde k zobrazení dalších výzev a to k zadání dimenze resp. velikosti populace a počtu opakování. V tomto případě se použije stejný postup jako v předchozí situaci. Po zadání posledního z dotazovaných parametrů, dojde k zobrazení zprávy „Všechny parametry zadány.“ Po zobrazení této zprávy započne program hledání minima zvolené funkce.

Po nalezení minima dané funkce v každém opakování zobrazí program v příkazovém okně funkční hodnotu v nalezeném minimu, počáteční populaci, konečnou populaci, počet zbývajících opakování a potřebný čas nutný pro nalezení minima. Poté co je dokončeno poslední opakování dojde k vykreslení grafu znázorňující pokles funkční hodnoty v minimu a to pro všechny provedená opakování.

6 ZÁVĚR

V této bakalářské práci byly vytvořeny tři modifikace metody flexibilního simplexu pro řešení problémů globální optimalizace. Všechny vytvořené varianty pracují s populací bodů. Pro testování těchto variant simplexové metody byly zvoleny tři z běžně používaných multimodálních testovacích funkcí a jedna funkce unimodální. Každá z vytvořených variant byla otestována dvakrát na každé funkci. První test byl proveden pro nižší dimenzi ($n = 3$), druhý test potom pro dimenzi vyšší ($n = 10$).

Nalezení globálního minima Ackleyho, Rastriginovy a Rosenbrockovy funkce pomocí v této práci vytvořených modifikací metody flexibilního simplexu se v nižší dimenzi nejeví jako problém a všechny modifikace vždy nalezly globální minimum těchto funkcí. Pro vyšší dimenzi těchto funkcí již měly všechny modifikace problém nalézt v daném počtu kroků globální minimum.

Pro poslední z testovaných funkcí, Schwefelovu funkci dochází k selhání všech modifikací a to jak při nižší, tak při vyšší dimenzi.

Z vytvořených metod, jako nejlepší vychází metoda č. 1 a to především pro nižší dimenzi, kde funguje spolehlivě pro všechny testovací funkce, kromě funkce Schwefelovy. Pro vyšší dimenze, kde nedošlo ke spolehlivému nalezení minima žádnou variantou si nejlépe se její jako nejlepší střídavě varianta č. 1 a variantou č. 2 v závislosti na testované funkci.

LITERATURA

NELDER, J. A.; MEAD, R. 1965. A simplex method for function minimization. *Computer Journal*, vol 7., s. 308 – 313.

PRESS, W. H. 1992. *Numerical Recipes in C*, Cambridge University Press, s. 502 – 509. ISBN 978-0521880688.

TAUFER, I; DRÁBEK, O; JAVŮREK, M. Metoda simplexů – efektivní nástroj pro řešení optimalizačních úloh, *CHEMagazín*, ročník XX (2010), č. 6, Pardubice: CHEMagazín s.r.o., Pardubice, s. 31 – 34. ISSN 1210-7409.

TVRDÍK, J. 2004. *Evoluční algoritmy*, Ostrava: Ostravská univerzita, 73 s.

SEZNAM PŘÍLOH

A – CD

Příloha A

Příloha k bakalářské práci

Úprava metody flexibilního simplexu pro řešení problémů globální optimalizace

Miroslav Provazník

CD

Obsah

1 Text bakalářské práce ve formátu PDF
2 Zdrojové kódy vytvořených programů