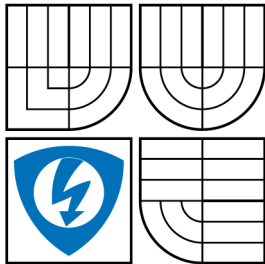


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ VUT V BRNĚ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION BRNO UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF CONTROL AND INSTRUMENTATION

Ing. JAN POHL

ALGORITMUS S PRAVDĚPODOBNOSTNÍM SMĚROVÝM
VEKTOREM

OPTIMIZATION ALGORITHM WITH PROBABILITY DIRECTION
VECTOR

Zkrácené znění doktorské práce

Školitel:

doc. Ing. VÁCLAV JIRSÍK, CSc.

KLÍČOVÁ SLOVA

optimalizační algoritmus, PDV, MPDV, MPDVC, MPDVCa, stochastická kooperace, statisticky ovlivněná perturbace, stochastický optimalizační algoritmus, rojový algoritmus

KEYWORDS

optimization algorithm, PDV, MPDV, MPDVC, MPDVCa, stochastic cooperation, statistically effected perturbation, stochastic optimization algorithm, swarm algorithm

Disertační práce je k dispozici na Vysokém učení technickém v Brně, Fakultě elektrotechniky a komunikačních technologií, Vědecké a zahraniční oddělení, Technická 3058/10, Brno 616 00, Česká republika

OBSAH

Úvod	4
1 Cíle dizertace	5
2 Navržený algoritmus	6
2.1 Algoritmus PDV	7
2.1.1 Definice množiny P	8
2.1.2 Definice pohybových rovnic	8
2.1.3 Definice koeficientu λ	9
2.1.4 Modifikace množiny P	9
2.1.5 Definice koeficientů α a β	10
2.2 Algoritmus MPDV	10
2.2.1 Definice populace jedinců matice X a množiny P	12
2.2.2 Pohybová rovnice k -tého jedince v algoritmu MPDV	12
2.3 Algoritmus MPDVC	13
2.3.1 První fáze algoritmu MPDVC - pohyb jedinců	13
2.3.2 Druhá fáze algoritmu MPDVC - kooperace jedinců	13
2.3.3 Algoritmus MPDVCa	15
3 Testování navržených algoritmů	18
3.1 Příklady z testů	19
4 Závěr	21
Literatura	23

ÚVOD

Náhoda je nepopíratelným fenoménem reálného života mající zásadní vliv na každého z nás. Náhody je využíváno v reálných podmínkách, stejně tak ve virtuálním prostředí. Výjimkou pro využití náhody nejsou ani optimalizační algoritmy. Algoritmy s čistě deterministickým přístupem jsou minoritní a nehodí se k řešení složitých optimalizačních úloh.

Moderní optimalizační metody využívají náhodu v různých fázích běhu algoritmu a pro různé účely. Kupříkladu výběr jedinců pro křížení, výběr místa v kódu pro dělení, náhodný směr posuvu jedince nebo přijetí potenciálně horšího výsledku. Algoritmy, využívající k prohledávání náhody bez dalších pomocných metod, se nazývají algoritmy stochastické. Moderní optimalizační algoritmy obvykle kombinují stochastický přístup s heuristickými nebo metaheuristickými metodami.

V dnešní době se realizují optimalizační algoritmy obvykle ve dvou směrech. Jsou to formy rojové a kolektivní inteligence a evoluční algoritmy. Rojové algoritmy využívají kooperující populaci, která lokálně prohledává stavový prostor. Při prohledávání využívají metod odvozených od přírodních dějů jako je kooperace hmyzu, lovcích zvířat a podobně. Naproti tomu evoluční algoritmy využívají jiných přírodních heuristických metod, které pomocí evolučních principů upravují či šlechtí každou následující generaci jedinců. K těmto postupným úpravám populace se využívá selekce, křížení a mutace.

Dizertační práce vychází z kategorie algoritmů stochastických. Jedinec či populace jedinců využívají statisticky ovlivněné perturbace stavovým prostorem.

1 CÍLE DIZERTACE

Cílem této práce je návrh optimalizačního algoritmu, který kombinuje náhodnou složku algoritmu spolu s informací, získanou heuristickými a metaheuristickými metodami. Výsledkem by měl být rychlý, robustní a v komplexnosti co nejjednodušší optimalizační algoritmus. Tento algoritmus by měl být schopen řešit širokou škálu úloh. Úlohy se spojitým stavovým prostorem či nespojitým, s parametry v oboru diskrétních i spojitých čísel. Toto vše bez nutnosti modifikace, či jen s jednoduchou obměnou základního algoritmu. V souhrnu definujeme cíle práce následovně.

1. Návrh heuristického algoritmu, který k lokálnímu prohledávání využívá statisticky ovlivněné perturbace.
2. Rozšíření algoritmu do populační podoby.
3. Zavedení kooperace mezi jedinci v populaci.
4. Statistické srovnání navržených algoritmů s běžně používanými optimalizačními metodami Simulované žíhání a Samo Organizující se Migrující Algoritmus.

2 NAVRŽENÝ ALGORITMUS

Na fraktály obvykle nahlížíme jen z pohledu generování obrazců v n rozměrném prostoru. Nově jsou fraktály používány ke kompresi obrazů, k popisu eukleidovskou geometrií složitě popsatečných struktur, k návrhu plošných spojů nebo realizaci antén.

Na iterativní generování fraktálního obrazce je možné nahlížet jako na trajektorii ve stavovém prostoru. Po aplikaci transformace či sady transformací jsme schopni z náhodného bodu ve stavovém prostoru docílit konkrétní trajektorie.

Pokud nahlížíme na generovanou trajektorii ve stavovém prostoru z pohledu optimalizace, daná trajektorie může být vývojem našeho optimalizovaného systému.

Algoritmus PDV přebírá ze stochastického systému iterovaných funkcí jednak translační matice pro pohyb bodu v n rozměrném prostoru a také množinu pravděpodobností P příslušnou k těmto translačním maticím.

V dizertační práci je navržen optimalizační algoritmus s pravděpodobnostním směrovým vektorem ve zkratce PDV. Tento algoritmus je svojí činností blízký stochastickým optimalizačním metodám. Jeho modifikace pak rojovým optimalizačním metodám. Optimalizační algoritmus s pravděpodobnostním směrovým vektorem aktuálně existuje ve čtyřech variantách.

1. Algoritmus PDV [17], [18] využívá k prohledávání stavového prostoru jednoho jedince, popsán je v podkapitola 2.1.
2. Algoritmus MPDV [19], [20] tato modifikace využívá populaci nekooperujících jedinců a ve své podstatě je paralelní formou algoritmu PDV a je popsán v podkapitole 2.2.
3. Algoritmus MPDVC [21] tato modifikace využívá populaci kooperujících jedinců a je popsán v podkapitole 2.3.
4. Algoritmus MPDVCa [21] rozšiřuje předchozí algoritmus MPDVC a ke stochastické kooperaci jedinců přidává adaptivní koeficient kroku 2.3.3.

Základním principem činnosti PDV a jeho modifikací je statisticky ovlivněná perturbace jedince ve stavovém prostoru. Tento princip je základem všech navržených modifikací základního algoritmu. Zvolený způsob pohybu po stavovém prostoru napodobuje tvorbu soběpříbuzných fraktálů generovaných pomocí algoritmu SIFS. Algoritmus SIFS je upraven tak, že využívá translační transformace spojené s pravděpodobnostmi. Perturbace čili porucha nebo nahodilost, je v tomto případě myšlena náhodným pohybem jedince po stavovém prostoru.

Jedinec se na počátku pohybuje z náhodné pozice. Výchozí pravděpodobnost pro každý směr je volena náhodně. Počáteční pravděpodobnost je možné také volit rovnoměrně rozloženou pro každý směr. Pravděpodobnost pohybu jedince určitým směrem se po každé iteraci upravuje v závislosti na změně hodnoty účelové funkce.

Pokud byl krok daným směrem úspěšný, je pro příští iteraci krok tímto směrem pravděpodobnější. V opačném případě pokud pohyb jedince daným směrem nebyl úspěšný, je chod stejným směrem v další iteraci méně pravděpodobný. Pokud je směr několikrát vybrán a je neúspěšný, dojde u jedince k opačnému pohybu podél dané osy. V pravděpodobnostním vektoru je tedy podstatný rozdíl algoritmu PDV oproti algoritmu SHC stochastický horolezecký algoritmus. Algoritmus SHC řeší pravděpodobnost stávajícího výpočetního kroku vzhledem k aktuálnímu rozložení hodnot účelové funkce v okolí. Oproti tomu u PDV je tento pohyb dán trendem vývoje účelové funkce od počátečního rozestavení jedinců po stavovém prostoru.

Druhá varianta algoritmu PDV označená jako MPDV. Tato modifikace využívá prohledávání stavového prostoru nekooperujícími jedinci. U algoritmu MPDV byl předpoklad, že paralelní prohledávání stavového prostoru má větší úspěšnost nalezení extrému funkce za stejné množství iteračních cyklů, než při využití jednoho jedince použitého v algoritmu PDV. Každý jedinec se řídí svými pravidly prohledávání jako by byl v prostoru sám.

Třetí variantou je algoritmus MPDVC. Jedná se o algoritmus MPDV rozšířený o fázi kooperace mezi jedinci. Koeficienty směrů a pravděpodobností jedince, jsou ovlivněny dvěma faktory.

1. Trendem vývoje účelové funkce samotného jedince.
2. Směrem k nejlepšímu dosud nalezenému jedinci ve stavovém prostoru.

Nejlepší nalezená pozice ovlivňuje pravděpodobnostní vektory celé populace jedinců. Pohyb každého jedince je touto polohou ovlivněn na pravděpodobnostní úrovni. Nedochozí k přímému ovlivnění polohového vektoru jedince vůči vítěznému jedinci. Ovlivněn je pouze pravděpodobnostní a směrový vektor. Tím dochází k efektu stochastická kooperace.

Čtvrtá varianta algoritmu je označena MPDVCa. Jedná se o předchozí variantu algoritmu MPDVC s adaptivním koeficientem kroku S . Tato modifikace sestavuje žebříček jedinců v závislosti na jejich hodnotě účelové funkce. Pořadí v žebříčku je následně použito jako adaptivní koeficient kroku.

2.1 Algoritmus PDV

Algoritmus Probability Direction Vector PDV používá tolik translačních matic, kolik je stupňů volnosti optimalizovaného systému. To znamená pro n proměnných máme n rozměrný stavový prostor a n translačních matic. Z toho vyplývá že množina pravděpodobností P zahrnuje vektor n pravděpodobností.

2.1.1 Definice množiny P

V předchozím odstavci byla k sadě translačních transformací přiřazena množina pravděpodobností a směrů P . Tato množina má tvar podle vzorce 2.1.

$$P = \{p_1, p_2, \dots, p_n\}. \quad (2.1)$$

Označovat tuto množinu za množinu čistě pravděpodobností by bylo krajně odvážné a pro další práci se tohoto vyvaruji. Množina P oproti množině pravděpodobností ze SIFS obsahuje v první řadě informace o pravděpodobnosti příslušné translační transformace vyjádřené modulem koeficientu $|p_i|$ tak jak je uveden v rovnici

$$probability_{(\mathbf{A}_i)} = |p_i|. \quad (2.2)$$

Vlastnosti členů množiny P jsou dány následujícími předpoklady

$$0 < |p_i| < 1, \quad (2.3)$$

$$-1 < p_i < 1, \quad (2.4)$$

$$\sum_{i=1}^n |p_i| = 1. \quad (2.5)$$

Druhou informací, kterou obsahují koeficienty množiny P jsou směry příslušných modifikací. Tedy zda se jedinec \mathbf{X} bude vůči zvolené ose pohybovat ve směru $+$ či ve směru $-$. Směr, kterým daná translační transformace pohybuje jedincem, \mathbf{X} vůči vybrané ose, je dán znaménkem příslušného koeficientu p_i z množiny P . Postup je uveden v rovnici

$$direction = \text{sgn}(p_i). \quad (2.6)$$

2.1.2 Definice pohybových rovnic

Pohyb jedince X po stavovém prostoru je tedy dán transformacemi 2.7 ve směru osy x_1 , a transformací 2.8 podél osy x_2 .

$$\mathbf{A}_1 \mathbf{X} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} p_1 \lambda \\ 0 \end{pmatrix}, \quad (2.7)$$

$$\mathbf{A}_2 \mathbf{X} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ p_2 \lambda \end{pmatrix}. \quad (2.8)$$

Jiný možný zápis je dán rovnicemi

$$\mathbf{A}_1 \mathbf{X} = \begin{pmatrix} 1 + \frac{p_1 \lambda}{x} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad (2.9)$$

$$\mathbf{A}_2 \mathbf{X} = \begin{pmatrix} 1 & 0 \\ 0 & 1 + \frac{p_2 \lambda}{y} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \quad (2.10)$$

2.1.3 Definice koeficientu λ

V rovnicích 2.7, 2.8, 2.9 a 2.10 byl použit koeficient λ . Tento koeficient vyjadřuje krok jedince vybraným směrem. Ve spolupráci s příslušnou hodnotou pravděpodobnosti p_i vyjadřuje velikost pohybu jedince ve stavovém prostoru.

Jelikož je optimalizace systému iterativní proces, můžeme jedince \mathbf{X} po aplikaci transformace $\hat{\mathbf{X}}$ chápat jako jedince v iteračním kroku $t + 1$. Rovnice 2.9 a 2.10 upravíme do podoby změny polohy jedince ve stavovém prostoru pro iterační krok $t + 1$

$$\hat{x}_1 = x_1^{t+1} = x_1^t + \left(1 + \frac{p_1 \lambda}{x_1^t}\right) = x_1^t + p_1 \lambda, \quad (2.11)$$

$$\hat{x}_2 = x_2^{t+1} = x_2^t + \left(1 + \frac{p_2 \lambda}{x_2^t}\right) = x_2^t + p_2 \lambda. \quad (2.12)$$

Z rovnic 2.11 a 2.12 je vidět, že pohyb jedince je řízen pouze na základě parametru p_i a koeficientu kroku λ .

2.1.4 Modifikace množiny P

Klíčovou funkcí algoritmu PDV je práce s koeficienty množiny P . S každou iterací je v první řadě použita vážená ruleta a vybrán směr modifikace jedince. Tomuto směru odpovídá koeficient z množiny P označíme jej p_i . Vítězný směr je použit pro pohyb jedince stavovým prostorem.

Hodnotu účelové funkce si označíme $fitness_t$ a hodnotu účelové funkce posunutého jedince $fitness_{t+1}$. Po použití vybrané transformace \mathbf{A}_i dojde k vyhodnocení rozdílu hodnot účelových funkcí $fitness_t$ a $fitness_{t+1}$. V tuto chvíli mohou nastat následující dva případy.

První případ je $fitness_{t+1} < fitness_t$. Pokud je hodnota účelové funkce posunutého jedince $fitness_{t+1}$ lepší než stávající hodnota jedince $fitness_t$, je tento směr pohybu jedince vyhodnocen jako přínosný. S daným směrem spojená hodnota pravděpodobnosti a směru p_i je zvýšena podle rovnice

$$p_i^{t+1} = p_i + \text{sgn}(p_i)\alpha. \quad (2.13)$$

Aby byly zaručeny vlastnosti množiny P podle zápisů 2.3, 2.4 a 2.5 provedeme normalizaci množiny P podle vzorce

$$P_{norm} = \frac{P}{\sum_{i=1}^n |p_i|}. \quad (2.14)$$

Výsledkem těchto úprav je jedinec s větší pravděpodobností pohybu ve směru i -té osy v iteraci $t + 1$. Dále tento jedinec splňuje požadavky kladené na množinu P dané body 2.3, 2.4 a 2.5.

Druhý případ je $\text{fitness}_{t+1} \geq \text{fitness}_t$. Ve tomto případě je vybraný směr nežádoucí a příslušná pravděpodobnost p_i z množiny P je snížena podle vzorce

$$p_i^{t+1} = p_i + \text{sgn}(p_i)(-\beta). \quad (2.15)$$

Obdobně jako v prvním případě, musí dojít po zmenšení koeficientu p_i k normalizaci vektoru P podle vzorce 2.14. Tímto je jedinec připraven pro další iterační krok algoritmu. V průběhu chodu algoritmu je ukládána vždy nejlepší dosažená pozice, tedy jedinec s nejmenší hodnotou účelové funkce fitness . Optimalizační proces se iterativně opakuje krok za krokem. Algoritmus končí po dosažení požadované úrovně účelové funkce, případně po dosažení povoleného počtu iteračních kroků.

2.1.5 Definice koeficientů α a β

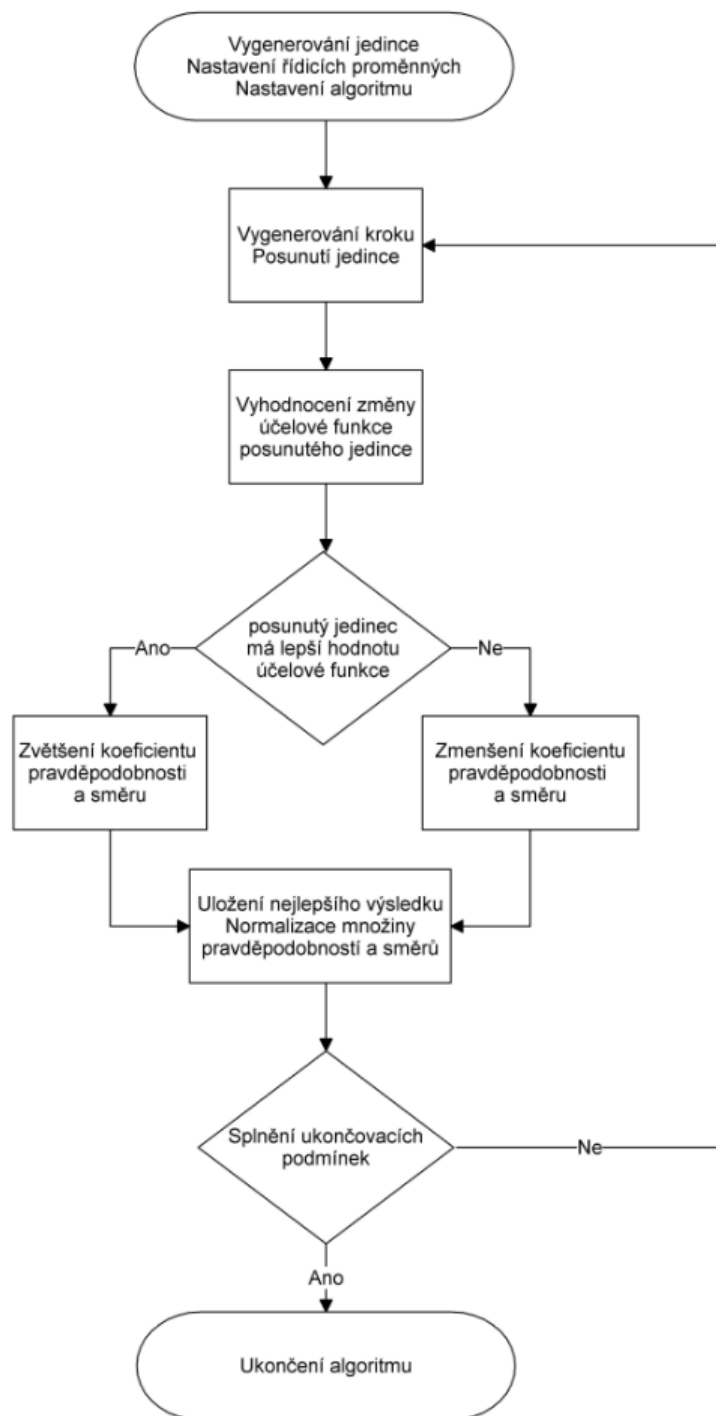
Ve vzorcích 2.13 a 2.15 byly použity nové koeficienty α , β . Jedná se o koeficienty setrvačnosti, které ovlivňují reakce jedince \mathbf{X} při změně hodnoty jeho účelové funkce fitness . Pro oba koeficienty platí následující vztah.

$$\alpha \in (0, 1) \cup \beta \in (0, 1) \cup \alpha \leq \beta. \quad (2.16)$$

Volbou vhodné velikosti koeficientu kroku λ a setrvačných koeficientů α a β se zabývá kapitola v dizertační práci.

2.2 Algoritmus MPDV

Multi Probability Direction Vector MPDV je první modifikací základního algoritmu PDV. Algoritmus využívá k prohledávání populaci nekooperujících jedinců.



Obr. 2.1: Vývojový diagram algoritmu PDV

2.2.1 Definice populace jedinců matice X a množiny P

Matice \mathbf{X} obsahuje populaci jedinců podle rovnice

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{k1} & x_{k2} & \dots & x_{kn} \end{pmatrix}. \quad (2.17)$$

Každý řádek odpovídá jednomu z k jedinců v n rozměrném stavovém prostoru. Obdobným způsobem je rozšířena i množina P do podoby matice

$$P = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \dots & \dots & \dots & \dots \\ p_{k1} & p_{k2} & \dots & p_{kn} \end{pmatrix}. \quad (2.18)$$

Pro pohyb jedinců ve stavovém prostoru je použito n translačních transformací obdobně jak jsou uvedeny pro dvourozměrný problém v dizertaci.

2.2.2 Pohybová rovnice k -tého jedince v algoritmu MPDV

Zápis pohybové rovnice pro pohyb k -tého jedince ve směru osy i je podle vzorce

$$\dot{x}_{ki} = x_{ki}^{t+1} = x_{ki}^t + \left(1 + \frac{p_{ki}\lambda}{x_{ki}^t}\right) = x_{ki}^t + p_{ki}\lambda. \quad (2.19)$$

Obdobný je i postup pro úpravu množiny pravděpodobností a směrů P . Pro každého k -tého jedince je známá hodnota jeho účelové funkce pro stávající pozici ve stavovém prostoru $fitness_k^t$ a hodnota účelové funkce posunutého k -tého jedince $fitness_k^{t+1}$.

Průběh algoritmu je obdobný jako u základní verze PDV. Počáteční polohy jedinců \mathbf{X} jsou náhodně vygenerovány po stavovém prostoru. Pro celou populaci je vygenerována množina koeficientů pravděpodobností a směrů P . Množina P obsahuje pro každého jedince n koeficientů pravděpodobností a směrů jak je uvedeno v rovnici 2.18. Tímto je ukončen krok inicializace.

Obdobně jak byla popsáno chování jedince v PDV je přístupováno stejným způsobem k celé populaci. Pro každého jedince z populace \mathbf{X} je vygenerován jedinec posunutý. Na základě změny hodnoty účelové funkce každého jedince $fitness_i^t$ a $fitness_i^{t+1}$ je upraven příslušný koeficient pravděpodobnosti a směru z množiny P . Po úpravě posledního jedince a změně posledního vybraného koeficientu pravděpodobnosti a směru je ukončena iterace a algoritmus se opakuje. Celou populaci reprezentuje jedinec s nejlepší dosaženou hodnotou účelové funkce $fitness_{best}$.

Předpokládaným výsledkem paralelního prohledávání stavového prostoru je vyšší pravděpodobnost nalezení extrému v menším počtu iteračních cyklů.

2.3 Algoritmus MPDVC

Druhou modifikací základního algoritmu je Multi Probability Direction Vector with Cooperation MPDVC. Tedy populační algoritmus s pravděpodobnostním směrovým vektorem a kooperací. Fáze inicializace je totožná pro MPDV, MPDVC a MPDVCa. Samotný iterační krok je u algoritmu MPDVC rozdělen do dvou fází.

2.3.1 První fáze algoritmu MPDVC - pohyb jedinců

V první fázi probíhá stejný postup jako u algoritmu MPDV popsáném v podkapitole 2.2.

U každého jedince populace je porovnána hodnota účelové funkce stávající pozice a posunuté pozice. Dle rozdílu hodnot $fitness_{ki}^t$ a $fitness_{ki}^{t+1}$ je upraven pro každého jedince vybraný koeficient pravděpodobností a směrů p_{ki} . Po obslužení posledního jedince je ukončena první fáze iteračního kroku.

2.3.2 Druhá fáze algoritmu MPDVC - kooperace jedinců

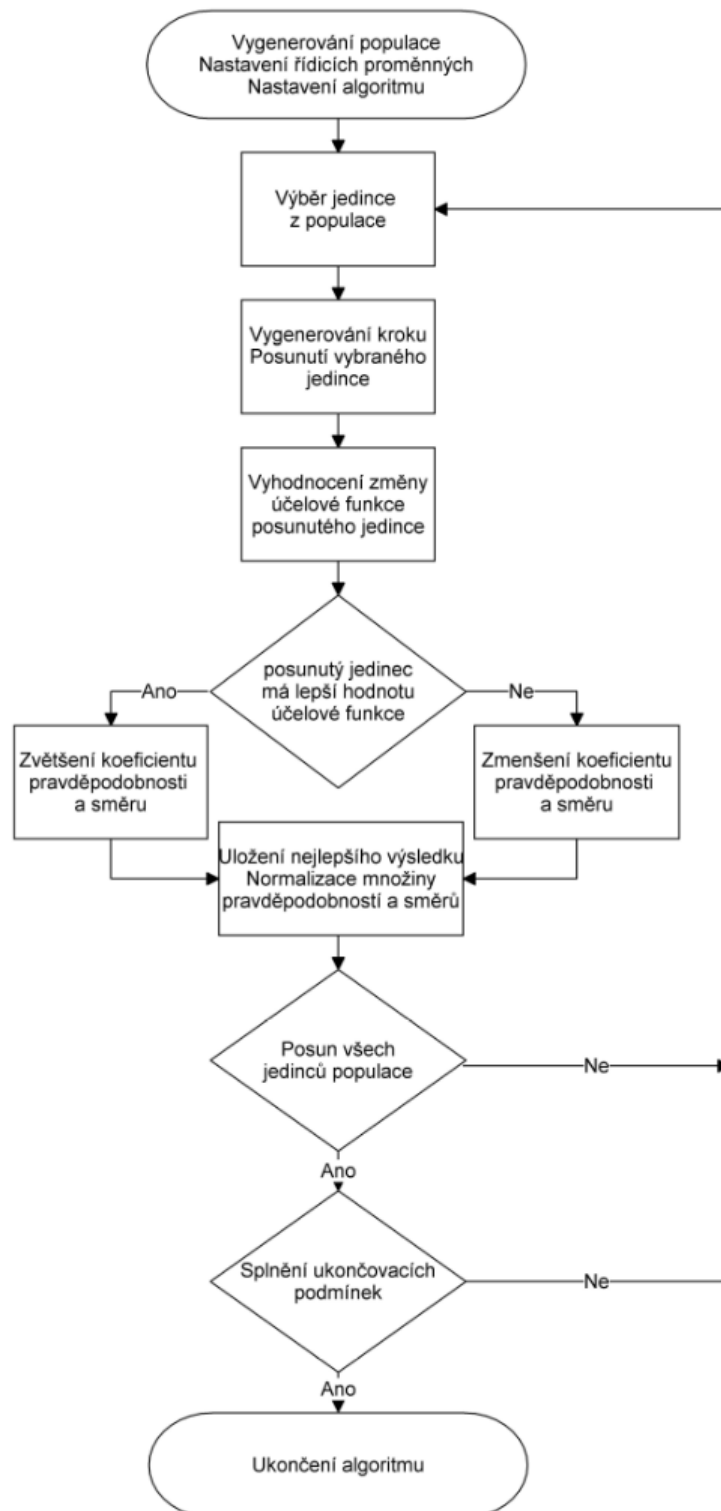
Ve druhé fázi je použita stochastická kooperace mezi jedinci populace \mathbf{X} s nejlepší dosud nalezenou hodnotou účelové funkce $fitness_{best}$. Hodnota $fitness_{best}$ nemusí vycházet jen z nejlepší aktuální pozice jedinců, ale je nejlepším výsledkem nalezeným celou populací za všechny předchozí iterační kroky. Pozici s nejlepší hodnotou účelové funkce si označíme \mathbf{X}^{best} je popsán vzorcem

$$\mathbf{X}^{best} = \left(x_1^{best} \quad x_2^{best} \quad \dots \quad x_n^{best} \right). \quad (2.20)$$

Pro každého jedince z populace \mathbf{X} je vypočítán směrový vektor k pozici s nejlepší hodnotou účelové funkce \mathbf{X}^{best} podle vzorce 2.21. Matici směrů si označíme **DIRECTION**.

$$\mathbf{DIRECTION} = \begin{pmatrix} x_1^{best} - x_{11} & x_2^{best} - x_{12} & \dots & x_n^{best} - x_{1n} \\ x_1^{best} - x_{21} & x_2^{best} - x_{22} & \dots & x_n^{best} - x_{2n} \\ \dots & \dots & \dots & \dots \\ x_1^{best} - x_{k1} & x_2^{best} - x_{k2} & \dots & x_n^{best} - x_{kn} \end{pmatrix}. \quad (2.21)$$

Aby bylo možné zpracovat matici **DIRECTION** a množinu P je nutné matici **DIRECTION** v každém řádku normalizovat. Výsledkem po normalizaci jednotlivých směrových vektorů je matice **DIRECTION**_{norm}.



Obr. 2.2: Vývojový diagram algoritmu MPDV

$$\mathbf{DIRECTION}_{\text{norm}} = \begin{pmatrix} \frac{x_1^{\text{best}} - x_{11}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{1i}|} & \frac{x_2^{\text{best}} - x_{12}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{1i}|} & \cdots & \frac{x_n^{\text{best}} - x_{1n}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{1i}|} \\ \frac{x_1^{\text{best}} - x_{21}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{2i}|} & \frac{x_2^{\text{best}} - x_{22}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{2i}|} & \cdots & \frac{x_n^{\text{best}} - x_{2n}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{2i}|} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{x_1^{\text{best}} - x_{k1}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{ki}|} & \frac{x_2^{\text{best}} - x_{k2}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{ki}|} & \cdots & \frac{x_n^{\text{best}} - x_{kn}}{\sum_{i=1}^n |x_i^{\text{best}} - x_{ki}|} \end{pmatrix}. \quad (2.22)$$

Získanou matici $\mathbf{DIRECTION}_{\text{norm}}$ vynásobíme kooperační konstantou γ . Výsledek sečteme s maticí pravděpodobností a směrů P .

Výsledkem této operace vznikne matice P_{COOP} . Tedy matice pravděpodobností a směrů upravená o kooperaci jedinců v populaci s nejlepším doposud nalezeným jedincem. Výsledný součet matic P_{COOP} normalizujeme obdobně jako předchozí matici $\mathbf{DIRECTION}_{\text{norm}}$ v 2.22.

$$P_{\text{COOP}} = P + \gamma \cdot \mathbf{DIRECTION}_{\text{norm}}. \quad (2.23)$$

Výsledkem je množina pravděpodobností a směrů upravená o kooperaci jedinců s nejlepším řešením P_{COOP} . Tímto krokem končí druhá fáze algoritmu MPDVC.

Pro koeficienty kooperace γ platí následující zápis

$$\gamma \in \langle 0, 1 \rangle. \quad (2.24)$$

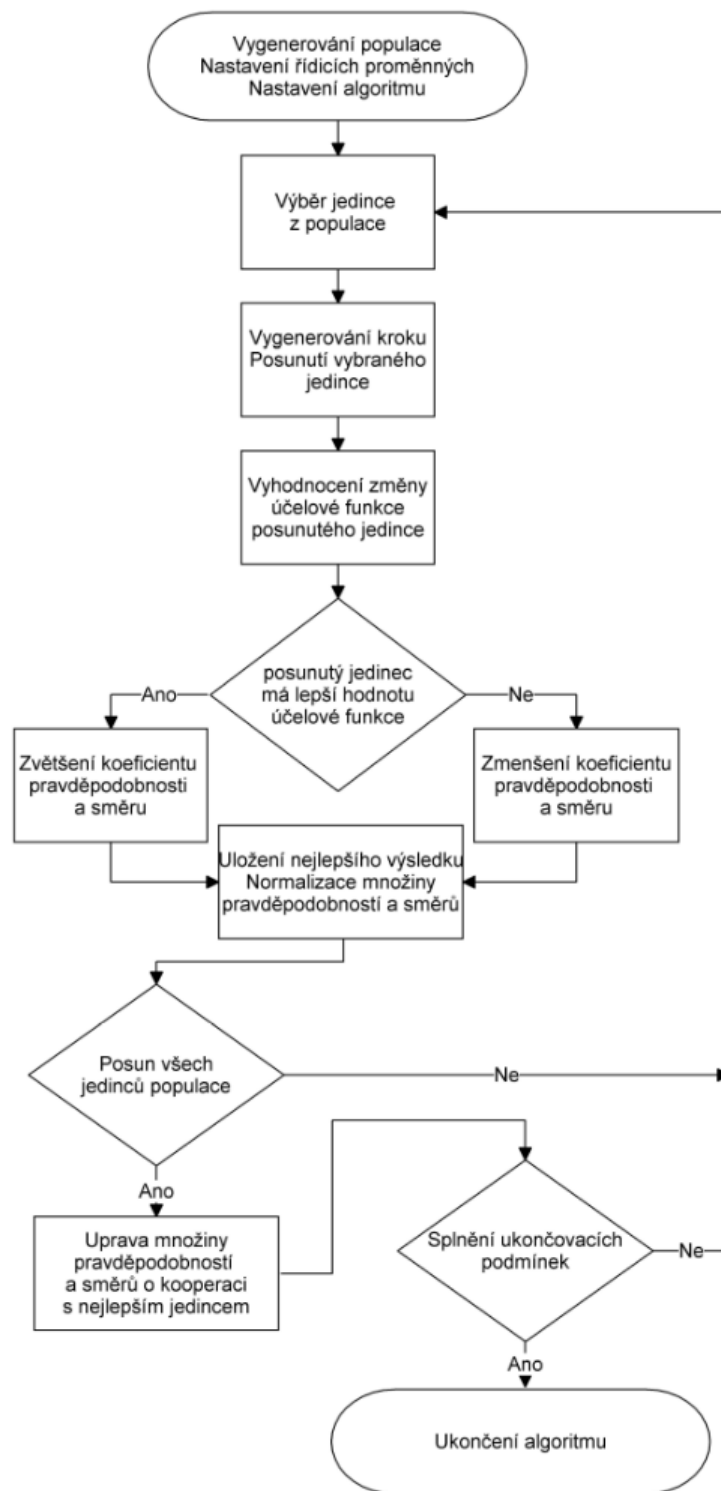
Pokud je koeficient γ rovný 0 nedochází k žádné spolupráci a algoritmus je ekvivalentní k algoritmu MPDV. Naopak pokud je koeficient rovný 1 dochází k plnému využití informace o poloze nejlepšího doposud nalezeného řešení.

2.3.3 Algoritmus MPDVCa

Poslední navrženou modifikací základního algoritmu je Multi Probability Direction Vector with Cooperation adaptive MPDVCa. Úprava algoritmu spočívá v předpokladu, že čím je daná hodnota účelové funkce horší tím je více žádoucí dané místo jedincem opustit. Jedinci jsou v populaci umístěni do žebříčku úspěšností. Jedinec s nejlepší hodnotou účelové funkce je na první příčce. Naopak s nejhorší hodnotou účelové funkce na poslední příčce žebříčku. Číselné pořadí v žebříčku označíme písmenem S . Pořadí musí splňovat následující požadavek

$$S \in \langle 1, N \rangle. \quad (2.25)$$

Kde N je počet jedinců v populaci.



Obr. 2.3: Vývojový diagram algoritmu MPDVC

Rozšířením rovnice 2.11 o adaptivní koeficient pohybu S dostaneme rovnici

$$\dot{x}_1 = x_1^{t+1} = x_1^t + p_1 \cdot S \cdot \lambda. \quad (2.26)$$

Z rovnice 2.26 je zřejmé že modifikace kroku se uplatňuje vždy se zpožděním jednoho iteračního kroku a její trvání je omezeno pouze na tento jediný iterační krok. V následujícím kroku pokud se změní žebříček jedinců dojde u každého jedince z populace k odlišné adaptaci.

3 TESTOVÁNÍ NAVRŽENÝCH ALGORITMŮ

Cílem této kapitoly je vybrat a navrhnout způsob testování co nejméně zatížený programátorskými schopnostmi. Proto byla zvolena strategie srovnávacích testů inspirovaných článkem [7]. Srovnávacím kritériem, pro všechny použité algoritmy, bylo dosažení co nejnižší hodnoty minima účelové funkce. Ze stejného důvodu byl zvolen pevně stanovený limit počtu volání účelové funkce. Oproti článku [7] byl tento limit stejný pro všechny minimalizační úlohy. Povolený počet volání účelové funkce byl stanoven na $n = 10000$ pro každý jednotlivý běh testovaného algoritmu.

Vybrané testovací funkce jsou:

- 1st De Jong function,
- 3rd De Jong function,
- 4th De Jong function,
- Ackley's function,
- Egg holder function,
- Griewangk's function,
- Masters's cosine,
- Michalewicz's function,
- Pathological function,
- Rana's function,
- Rastrigin's function,
- Rosenbrock's saddle,
- Schwefel's function,
- Stretched V sine wave function,
- Test function (Ackley),
- Sine envelope sine wave function.

Vybrány byly dimenze 2, 10 a 100. Každé nastavení řídicích proměnných, pro každý optimalizační algoritmus, bylo 100x opakováno. Ze 100 získaných hodnot minima byla vypočítána průměrná hodnota dosaženého minima. Tato průměrná hodnota dále reprezentovala dané testované nastavení.

Z celé sady nastavení řídicích parametrů testovaných na každé funkci a pro každou dimenzi testovací funkce, byl vybrán vždy ten nejlepší průměrný výsledek. Nejlepší výsledky byly porovnány s obdobně získanou hodnotou ze srovnávacího algoritmu SA [1]-[6] a SOMA [9]-[16]. Sada 100 výsledků s nejlepší průměrnou hodnotou z navrženého algoritmu a ze srovnávacího algoritmu byla vždy podrobena Kruskal Wallisovu neparametrickému testu. Výsledkem je zjištění zda jsou hodnoty získané oběma algoritmy statisticky srovnatelné či nikoliv.

Souhrn výsledků navržených algoritmů pro všechny testy je sjednocen v tabulce 3.1. Uvedeny jsou ty hodnoty kde byl navržený algoritmus srovnatelný nebo lepší

než testovací. Tabulka vyjadřuje jednak výsledky pro každý testovaný algoritmus přes všechny dimenze testů. Ty jsou uvedeny v řádcích tabulky. Dále pak úspěšnost algoritmů na jednotlivých dimenzích testů. Ty jsou uvedeny ve sloupcích tabulky. Navržené algoritmy jsou dle výsledků, nejvýhodnější v řešení problémů jednotek až desítek neznámých. Na základě těchto výsledků můžeme prohlásit nulovou hypotézu za neplatnou a tedy alternativní hypotézu za platnou pro algoritmy PDV, MPDVC a MPDVCa.

Tab. 3.1: Souhrn výsledků

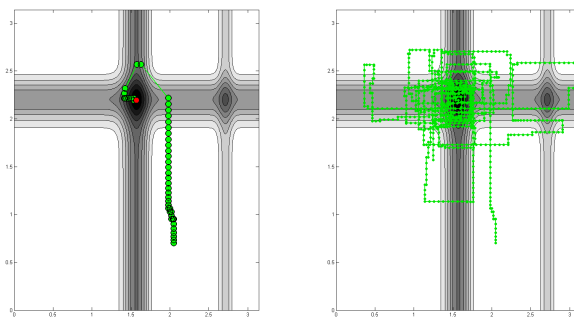
	d2	d10	d100	RES./počet f.
PDV	16	16	9	41/48
MPDVC	6	12	4	22/48
MPDVCa	6	13	5	24/48
RES./počet f.	28/48	41/48	18/48	

3.1 Příklady z testů

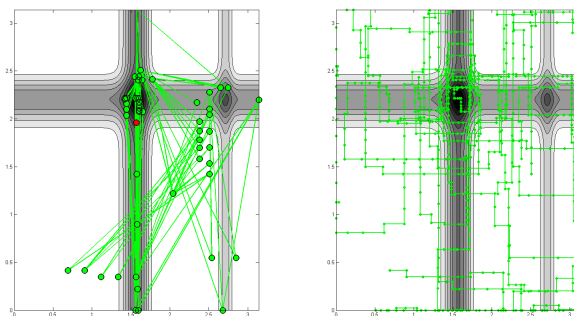
Obrázek 3.1 ukazuje samotného jedince z algoritmu PDV při prohledávání 2D stavového prostoru. Na levém obrázku je vývoj nejlepší nalezené hodnoty. Na pravém obrázku je pohyb jedince.

Obrázek 3.2 zachycuje pohyb nekooperující populace. Levý obrázek zachycuje vývoj nejlepší nalezené hodnoty. Vpravo je vidět pohyb nekooperující populace. Každý jedinec se chová v prostoru samostatně a není ostatními jakkoliv ovlivněn.

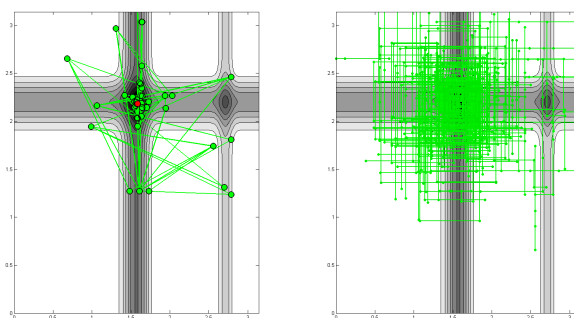
Obrázek 3.3 reprezentuje chování kooperující populace ve stavovém prostoru. Na levém obrázku je vývoj nejlepší nalezené hodnoty populací. Pravý obrázek ukazuje pohyb celé populace stavovým prostorem. Z obrázku je zjevná kooperace mezi jedinci. Ač je kooperace opět dána jen modifikací pravděpodobnostního a směrového vektoru, je zjevný pohyb celé populace do zajímavějších částí funkce v prostoru globálního extrému. I v případě této kooperace se jedná o modifikaci základní myšlenky statisticky ovlivněné perturbace jedinců.



Obr. 3.1: PDV Michalewicz's function a) pohyb vítěze b) pohyb jedince



Obr. 3.2: MPDV Michalewicz's function a) pohyb vítěze b) pohyb populace



Obr. 3.3: MPDVCa Michalewicz's function a) pohyb vítěze b) pohyb populace

4 ZÁVĚR

Cílem této práce bylo navrhnout optimalizační algoritmus využívající stochastického přístupu spolu s heuristicky získanou informací k prohledávání stavového prostoru. Navržený algoritmus PDV spojuje tyto dva požadavky. Stochastický přístup je dán použitím vážené rulety pro výběr směru pohybu jedince stavovým prostorem. Heuristika se uplatňuje úpravou jednotlivých položek pravděpodobnostního a směrového vektoru v závislosti na vývoji hodnoty účelové funkce jedince. Další modifikace základního algoritmu posiluje vliv heuristicky získané informace.

Přínosem nového řešení je zavedení stochastické kooperace jedinců v populaci. Součástí jedince je pravděpodobnostní a směrový vektor, který určuje možnosti pohybu jedince stavovým prostorem. Ten je u algoritmu MPDVC jednak modifikován v závislosti na vývoji účelové funkce jedince, ale také na základě vývoje hodnoty účelové funkce nejlepšího jedince v populaci. U konkurenčních algoritmů je pohyb jedinců po stavovém prostoru kooperací ovlivněn přímo. Jedinci v populaci nejsou nuceni k pohybu směrem k nejlepšímu jedinci přímo, ale směr k nejlepšímu jedinci je pro ně pravděpodobnější než jiné směry. Z výsledků vyplývá, že tento způsob kooperace zásadně ovlivňuje kvalitu nalezených řešení.

Vlastní funkcí byl algoritmu PDV vybrán jako nejbližší algoritmus SA. Při komparaci algoritmů na testovacích funkcích dimenze D2, navržený algoritmus PDV, podává statisticky lepší výsledky na 9 vybraných testovacích funkcích. Na zbylých 7 testovacích funkcích jsou jeho výsledky statisticky srovnatelné. Srovnání na testovacích funkcích dimenze D10 dopadlo výhrou pro algoritmus PDV v 10 případech. Na zbylých 6 testovacích funkcích jsou výsledky PDV statisticky srovnatelné. Testování na funkcích dimenze D100 byly výsledky v 9 případech statisticky srovnatelné s algoritmem SA a ve zbylých 7 případech byl algoritmus PDV statisticky horší.

Dalším navrženým algoritmem byl MPDV. Ten byl rozšířením základní myšlenky na populaci nekooperujících jedinců. Od prvních testů se výsledky ukázaly jako neperspektivní a nekonkurenceschopné srovnávacímu algoritmu SOMA. V práci mu krom popisu, nebylo věnováno víc pozornosti.

Přidáním statistické kooperace do populace jedinců vznikl algoritmus MPDVC. Tento srovnávací soubor dat byl především testem navržené statistické kooperace mezi jedinci. První soubor testů porovnává výsledky algoritmu MPDVC se srovnávacím algoritmem SOMA. Na sadě testovacích funkcí dimenze D2 byl algoritmus MPDVC statisticky srovnatelný v 6 případech. Ve zbylých 10 případech byly jeho výsledky statisticky horší. Následný soubor testů probíhal na funkcích dimenze D10. Zde byl algoritmus MPDVC statisticky lepší v 1 případě. V 11 případech byly výsledky statisticky srovnatelné a ve 4 případech byly jeho výsledky oproti srovnávacímu algoritmu statisticky horší. V posledním souboru testů funkcí dimenze D100

byly výsledky algoritmu lepší ve 3 případech. V 1 případě byly jeho výsledky statisticky srovnatelné a v 12 případech byly výsledky statisticky horší.

V pořadí čtvrtým navrženým algoritmem je MPDVCa. Ten do výpočtu zavádí žebříček jedinců, který v závislosti na pořadí jedince, zajišťuje adaptivní koeficient kroku. První sada testů obsahuje data porovnání výkonů na sadě testovacích funkcí dimenze D2. Algoritmus MPDVCa byl v 6 případech statisticky srovnatelný. Ve zbylých 10 případech byly jeho výsledky na testovacích funkcích statisticky horší. Následuje sada testů funkcí dimenze D10. V tomto případě byl algoritmus MPDVCa statisticky úspěšnější v 1 případě. Ve 12 případech byl algoritmus statisticky srovnatelný se svým konkurentem a ve 3 případech byly jeho výsledky statisticky horší. Poslední byla sada testů funkcí dimenze D100. V posledním případě byl testovaný algoritmus statisticky úspěšnější ve 3 případech. Ve 2 případech byly jeho výsledky srovnatelné a ve zbylých 11 případech byly jeho výsledky horší.

Algoritmus PDV, můžeme na základě daných výsledků, hodnotit jako lepší než algoritmus SA na daném souboru testů. U dalších navržených algoritmů MPDVC a MPDVCa již nemůžeme být v hodnocení takto optimističtí. Obě další modifikace základního algoritmu můžeme prohlásit nanejvýše za srovnatelné s algoritmem SOMA. V obou případech však můžeme považovat, jednak základní princip statisticky ovlivněné perturbace populace jedinců prostorem a statistickou kooperaci mezi jedinci, za funkční. Zároveň můžeme prohlásit nulovou hypotézu definovanou v dizertaci, za vyvrácenou.

Další možnosti výzkumu algoritmu PDV. V této práci byly testovány základní myšlenky statisticky ovlivněné perturbace jedince stavovým prostorem a forma statistické kooperace mezi jedinci. Dalším možným směrem práce je možné zaměřit na plnou adaptivitu řídicích parametrů algoritmu. Cílem by mělo být omezení řídicích parametrů pouze na prohledávaný rozsah. Všechny ostatní jmenované parametry by měl být plně v režiji samotného optimalizačního algoritmu. Tímto by se značně zlepšil uživatelský komfort.

Dalším možným rozšířením navržených algoritmů by mělo být zavedení dynamického počtu jedinců v populaci. Kdy velikost populace bude pružně reagovat na vývoj účelové funkce ať zvyšováním či snižováním počtu jedinců v populaci.

LITERATURA

- [1] KIRPATRICK S., GELATT C. D., VECCHI M. P. *Optimization by Simulated annealing* Science. New Series 220 (4598): 671–680. , 1983, ISSN 00368075
- [2] ČERNÝ, V. *A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm* Journal of Optimization Theory and Applications, 1985
- [3] DUECK G., SCHEUER T., *Treshhold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing* J. Comp. Phys. 90, 161175, 1990
- [4] HENDERSON D., JACOBSON S. H., JOHNSON A. W., *THE THEORY AND PRACTICE OF SIMULATED ANNEALING* Handbook of Metaheuristics International Series in Operations Resasrch & Management Science Volume 57,2003,pp287-319, ISBN 978-1-4020-7263-5
- [5] RUTENBAR R. A., *Simulated Annealing Algorithms: An Overview* IEEE Circuits and devices Magazine January 1989,p19-26, 8755-3996/89/0100-0019
- [6] BERTSIMAS D., TSITSIKLIS J., *Simulated Annealing* Statistical Science 1993, Vol. 8, No.1,p10-15 Dostupné z URL: <http://projecteuclid.org/download/pdf_1/euclid.ss/1177011077>.
- [7] LIANG, J.J., QIN A.K., PONNUTHURAI NAGARATNAM SUGANTHAN, BASKAR S. *Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal functions* IEEE Transaction on evolutionary computation, VOL. 10, NO. 3, JUNE 2006
- [8] VALLE Y., VENAYAGAMOORTHY G.K., MOHAGHEGHI S., HERNANDEZ G.K., HARLEY R.G. *Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems*IEEE Transaction on evolutionary computation, VOL. 12, NO. 2, Aprile 2008
- [9] ZELINKA, I. LAMPINEN, J. *SOMA-Self-Organizing Migrating Algorithm* Proceedings of MENDEL 2000, 6-th Int. Conference on Soft Computing, 177–187. Brno: Technical University, 2000
- [10] COELHO L. dos S., *Self-Organizing Migrating Strategies Applied to Reliability-Redundancy Optimization of Systems* New Optimization Techniques in Engineering, Studies in Fuzziness and Soft Computing Volume 141,2004, p 167-212, ISBN 978-3-540-39930

- [11] ZELINKA I., *SOMA-Self-Organizing Migrating Algorithm* New Optimization Techniques in Engineering, Studies in Fuzziness and Soft Computing Volume 141,2004, p 167-212, ISBN 978-3-540-39930
- [12] ZELINKA I., KOLOMAZNÍK K., LAMPIEN J., NOLLE L., *SOMA - Self-organizing Migrating Algorithms and its Applications in Mechanical Engineering. Part. 1* Algorithms Theory, Engineering Mechanics, (Czech ed.), 21 p. ISSN 1210-2717
- [13] ZELINKA I., KOLOMAZNÍK K., LAMPIEN J., NOLLE L., *SOMA - Self-organizing Migrating Algorithms and its Applications in Mechanical Engineering. Part. 2* Testing and Applications, Engineering Mechanics, (Czech ed.), 23p. ISSN 1210-2717
- [14] NOLLE L., ZELINKA I., *SOMA - Applied to Optimum Work Roll Profile Selection in the Hot Rolling of Wide Steel* In ESM 2003, European Simulation Multiconference, The Nottigham Trent University, Nottigham, England, UK, p. 53-58, ISBN:3-936150-25-7
- [15] ZELINKA I., LAMPINEN J., *SOMA - Self-Organizing Migrating Algorithm* In Mendel 2000, 6th International Conference on Soft Computing, Brno, Czech Republic, ISBN 80-214-1609-2
- [16] ZELINKA I., *SOMA - Self-Organizing Migrating Algorithm* Nostradamus 2000, 3rd International Conference on Prediction and Nonlinear Dynamic, Zlin, Czech Republic, ISBN 80-214-1668-8
- [17] POHL J., *STOCHASTICKÝ ALGORITMUS UČENÍ PRO UMĚLÉ NEURONOVÉ SÍŤE S PRAVDĚPODOBNOSTNÍM SMĚROVÝM VEKTOREM* Elektrovue - Internetový časopis 2009, roč. 2009, č 52,s :1-4, ISSN:1213-1539
- [18] POHL J., POLÁCH P., *Stochastic learning algorithm* In Mendel Journal series 2009, s 166-170, ISBN:978-80-214-3884-2
- [19] POHL J., POLÁCH P., *Filtration of samples for secondary surveillance radar* In Mendel Journal series 2010, s 260-265, ISBN:978-80-214-4120-0
- [20] POHL J., JIRSÍK V., HONZÍK P., *Stochastic optimization algorithm with probability vector* WSEAS Transaction on Information Science and Applications, 2010, roč. 7,č. 7,p 975-984, ISSN:1790-0832
- [21] POHL J., JIRSÍK V., HONZÍK P., *Optimalizační algoritmy s pravděpodobnostním směrovým vektorem a jejich srovnání s principiálně podobnými metodami*

Elektrorevue - Internetový časopis 2014, roč. 2014, č. 1, svazek 16, ISSN 1213-1539

ABSTRAKT

Disertační práce prezentuje navržený algoritmus s pravděpodobnostním směrovým vektorem. Tento algoritmus ve své základní formě spadá do kategorie algoritmů stochastických. Pro svoji práci využívá statisticky ovlivněné perturbace jedince stavovým prostorem. Dále práce pojednává o rozšíření základní myšlenky do podoby rojové formy algoritmu. Přínosem této modifikace je zavedení stochastické kooperace. Jedinci kooperují s nejlepším jedincem pouze na úrovni pravděpodobnosti a nejsou jí řízeni přímo. V práci jsou dále použity statistické testy pro srovnání výsledků navrženého algoritmu s podobně pracujícími algoritmy Simulované žíhání a SOMA. Kromě vybraných testovacích funkcí jsou zde uvedeny další experimentální použití navržených algoritmů. Práce je uzavřena kapitolou, která hledá obecně použitelné nastavení řídicích parametrů jednotlivých algoritmů.

ABSTRACT

This dissertation presents optimization algorithm with probability direction vector. This algorithm, in its basic form, belongs to category of stochastic optimization algorithms. It uses statistically effected perturbation of individual through state space. This work also represents modification of basic idea to the form of swarm optimization algorithm. This approach contains form of stochastic cooperation. This is one of the new ideas of this algorithm. Population of individuals cooperates only through modification of probability direction vector and not directly. Statistical tests are used to compare resultes of designed algorithms with commonly used algorithms Simulated Annealing and SOMA. This part of dissertation also presents experimental data from other optimization problems. Dissertation ends with chapter which seeks optimal set of control variables for each designed algorithm.