

VĚDECKÉ SPISY VYSOKÉHO UČENÍ TECHNICKÉHO V BRNĚ

Edice Habilitační a inaugurační spisy, sv. 436

ISSN 1213-418X

Radomil Matoušek

POKROČILÉ METODY POČÍTAČOVÉ INTELIGENCE

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta elektrotechniky a komunikačních technologií

Ing. Radomil Matoušek, Ph.D.

POKROČILÉ METODY POČÍTAČOVÉ INTELIGENCE

Advanced Methods in Computational Intelligence

TEZE HABILITAČNÍ PRÁCE



Brno 2012

Klíčová slova

Počítačová inteligence, soft computing, metaheuristiky, evoluční výpočetní techniky, genetické algoritmy, HC12, GAHC, GAFIS, elitní turnajová selekce, STE, globální optimalizace, QAP, aproximace, PID regulátor, polynomiální regulátor.

Key Words

Computational intelligence, soft computing, genetic algorithms, metaheuristic, evolutionary algorithms, HC12, GAHC, GAFIS, elite tournament selection, STE, global optimization, QAP, approximation, PID controller, polynomial controller.

Místo uložení práce

Vysoké učení technické v Brně
Fakulta elektrotechniky a komunikačních technologií

OBSAH

PŘEDSTAVENÍ AUTORA.....	4
1 ÚVOD, MOTIVACE A OBSAH HABILITAČNÍ PRÁCE.....	5
2 OPTIMALIZAČNÍ METAHEURISTIKY.....	7
2.1 Základní pojmy a rozdělení metod.....	7
2.2 Reprezentace a kódování informace.....	8
2.3 Grayovo kódování a Hammingova metrika.....	10
3 HC12: DOBŘE PARALELIZOVATELNÁ METAHEURISTIKA.....	11
3.1 Princip HC12.....	12
3.2 Formalizace HC12.....	13
4 GENETICKÝ ALGORITMUS.....	19
4.1 Seleční mechanismus.....	19
4.2 GAHC: hybridní genetický algoritmus.....	21
4.3 STE: kritérium v úloze symbolické regrese.....	22
5 APLIKACE METOD POČÍTAČOVÉ INTELIGENCE.....	24
5.1 Turnaj řešičů HC12 a GAHC vs. GAMS v úloze F7.....	24
5.2 QAP: kvadratický přiřazovací problém a HC12 swap2abs.....	24
5.3 Návrh obecného lineárního regulátoru.....	27
6 ZÁVĚR.....	31
REFERENCE.....	32

PŘEDSTAVENÍ AUTORA



Radomil Matoušek se narodil v roce 1972 v Brně. Vysokoškolské vzdělání (Ing.) získal v roce 1996 v oboru Aplikovaná informatika na Fakultě strojní Vysokého učení technického v Brně. V roce 2004 pod vedením školitele prof. Ing. Pavla Ošmery, CSc. dokončil studium doktorského studijního programu (Ph.D.) v oboru Technická kybernetika.

V současnosti je vedoucím odboru Aplikované informatiky na Ústavu automatizace a informatiky, Fakulty strojního inženýrství, VUT v Brně.

Současná vědecko-výzkumná činnost autora se zaměřuje na vývoj a aplikace vybraných metod počítačové inteligence v kontextu globální optimalizace. Studován je rovněž návrh inteligentních kybernetických systémů (roboty, plánování cest aj.).

Praktické zkušenosti, získané mimo VUT v Brně, autor nabyl: v letech 1990 – 1994 jako asistent výpočetní techniky a programátor databázových aplikací v Zetor Brno, SOU, v letech 2003 – 2005 jako odborný VV pracovník AV ČR, Ústav přístrojové techniky, odbor NMR, tým inverzních a optimalizačních metodologií a v letech 2003 – 2005 jako specialista softwarového vývoje na VÚJE Česká republika, Dukovany.

V roce 2004 absolvoval stáž na Kyoto University, Department of Energy Conversion Science (prof. E. Matsumoto, Japonsko) a v roce 2009 stáž na University of Vaasa, Department of Computer Science (prof. J. Lampinen, Finsko). V průběhu doktorského studia se autor aktivně účastnil světových konferencí a sympózií, přičemž čtyřikrát obdržel prestižní cenu Best Paper Award (2000, 2007, 2009, 2011) a jednou cenu Best Presentation Award (2004). K ocenění vědeckou komunitou patří rovněž tři zvané přednášky v Německu, Japonsku a České republice.

V rámci pedagogické činnosti v letech 1997 – 2012 autor vedl cvičení v následujících předmětech: Neuronové sítě a fuzzy množiny, Teorie informace a kódování, Simulace systémů, Vyšší formy řízení, Informatika, Počítačový hardware. Pod autorovým vedením (záznam od roku 2006/2007) bylo úspěšně obhájeno celkem 21 magisterských diplomových prací, jedna zahraniční diplomová práce na University of L'Aquila (Itálie) a 15 bakalářských závěrečných prací. V současnosti je autor podmíněně školitelem čtyř doktorských prací a školitelem specialistou, přičemž dva studenti již mají úspěšně vykonanou státní zkoušku. K podpůrným aktivitám pro studenty lze zařadit realizaci a koordinaci tří programů Erasmus (UK, Finsko, Rakousko). Autor vytvořil dvojí skripta a podílel se na tvorbě vysokoškolské učebnice "Automatické řízení".

K vědecko společenským aktivitám lze zařadit členství v mnoha programových výborech světových konferencí v oboru počítačové inteligence i práci recenzenta pro periodikum SCI, Springer v roce 2010. Od roku 1995 se autor aktivně účastní jako organizátor, recenzent a editor mezinárodní konference MENDEL, v roce 2008 se autor stává ředitelem konference, od téhož roku je konference indexovaná v ISI Thomson WoS a od roku 2012 ve Scopus.

Autor vytvořil, nebo spoluvytvořil téměř sto publikací, z nichž 33 je indexováno v ISI WoS nebo Scopus. Rovněž se jako člen týmu, spoluřešitel nebo řešitel účastnil 13 úspěšně ukončených a obhájených projektů, z nichž poslední VVZ "Inteligentní systémy v automatizaci" vedený prof. Pavlem Jurou, považuje za nejvýznamnější. V současnosti je autor řešitelem projektů TAČR, IGA VUT v Brně a má podány dva mezinárodní projekty. S mezinárodním týmem připravuje knihu pro nakladatelství Springer o HPC (www.HPCbook.info). Autor spolupracuje s mnoha zahraničními autoritami, např.: prof. Lampinen – Finsko, Dr. Langdon – UK, Dr. Xin She – UK, prof. Mendel – USA, prof. Klín – Izrael, Dr. O'Neill – Irsko, Dr. Ryan – Irsko, prof. Zaharie – Rumunsko.

1 ÚVOD, MOTIVACE A OBSAH HABILITAČNÍ PRÁCE

Při řešení technických úloh se setkáváme s optimalizačními problémy, které jsou pomocí tzv. standardních metod řešitelné obtížně, částečně, případně uspokojivé řešení není známo vůbec. Takovéto závěry mohou vyplývat buď z povahy řešené úlohy, nebo z absence vhodné teorie či přímo z absence efektivních optimalizačních algoritmů.

Přesto, že se vývoj v oblasti metod matematické optimalizace nezastavil, lze konstatovat, že dnes známé postupy došly do bodu, kdy limity těchto optimalizačních metod, stojících majoritně na pevných matematických základech, jsou posouvány spíše pomocí paralelních implementací, než vlastní tvorbou nových či zdokonalováním stávajících algoritmů. Toto konstatování není degradace dané matematické disciplíny, ale naopak vyzdvižení faktu, že za dobu její existence se toho v oblasti matematické optimalizace již udělalo opravdu mnoho.

Naproti tomu metody založené na umělé inteligenci (AI, Artificial Intelligence), počítačové inteligenci (CI, Computational Intelligence), soft computingu (SC) jsou relativně nové a stále se v mnoha ohledech dynamicky vyvíjejí. Tato skutečnost je dána mezioborovostí, existencí NP úplných úloh a faktem, že k rozvoji této oblasti vědy je výpočetní technika, resp. výkonná výpočetní technika, nezbytná. Právě těmito metodám a jejich implementacím je v poslední době věnována stále větší pozornost. V oblasti optimalizace mezi tyto metody patří například: genetické algoritmy (GA, Genetic Algorithms), diferenciální evoluce (DE, Differential Evolution), gramatická evoluce (GE, Grammatical Evolution) a různé další heuristiky více či méně inspirované přírodními principy. Připomeňme, že k důležitým oblastem umělé inteligence patří rovněž umělé neuronové sítě (ANN, Artificial Neural Networks), fuzzy metody, inteligentní agenti, specifické grafové algoritmy aj.

Pochopením přírodních principů, vytvořením adekvátních počítačových modelů a následnou implementací vznikají unikátní algoritmičké postupy, které jsou schopny konkurovat klasickým optimalizačním metodám, či je selektivně překonávat. Tyto algoritmy do dnešní doby již mnohokrát prokázaly svoji opodstatněnost a poskytly unikátní řešení problémů dosud neřešitelných či neuspokojivě řešitelných. Rešerše v tomto ohledu by vydala na samostatnou a rozsáhlou publikaci.

V předložené habilitační práci jsou prezentovány autorovy původní algoritmy a navržené postupy, které zařazujeme do oblasti tzv. evolučních výpočtů, či obecněji umělé inteligence, soft computingu či počítačové inteligence. Jde o algoritmy, které nějakým způsobem využívají princip evoluce, tedy působení selekce, rekombinace a mutace.

Tyto typy algoritmů, označované jako evoluční výpočetní techniky, jsou vyvíjeny a efektivně implementovány od 80. let minulého století. Od svého vzniku dokázaly svoji opodstatněnost a spolu s dalšími třídami algoritmů z dané oblasti i svoji jedinečnost. Výsledkem jsou unikátní řešení technických problémů potvrzená i řadou světových patentů a průmyslových aplikací. Je vhodné si uvědomit, že princip této třídy algoritmů je v podstatě navržený a odzkoušený samotnou přírodou. Za dobu své existence všechny biologické entity podléhaly velmi tvrdému tlaku ze strany přírody a jen jejich vlastnosti, schopnost optimalizovat svoji funkcionalitu vůči prostředí, umožnily úspěšným přežít. Výsledkem jsou jistě v mnoha ohledech optimální řešení typu člověk, žralok, včela apod. Pochopitelně hraje v evoluci svou roli ještě náhoda, či řekněme štěstí. Avšak otázka, zda vesmír hraje kostky, nepatří k těm nejnovějším.

V předložené habilitační práci jsou shrnuty mnohé výsledky, které autor nabyt v průběhu studia výše uvedeného paradigmatu evolučních výpočetních technik. V žádném případě nejde o všechny teoretické či aplikační výsledky z autorovy tvorby. Prezentován je však hlavní proud zájmu autora, který, jak je v práci uvedeno, vedl k návrhu nových algoritmů či vylepšení existujících, včetně praktických aplikací.

Z prostorových důvodů jsou v těchto tezích stručně prezentovány pouze vybrané výsledky uvedené v habilitační práci, které autor považuje za nejzásadnější. V širším pojetí by se jednalo o následující výčet původních výsledků, kterých si autor cení:

- ❖ *Algoritmus HC12 a HC12 taboo (včetně paralelních implementací)*. Algoritmus třídy HC12 byl postupně rozpracováván a úspěšně aplikován na mnoho optimalizačních úloh. Velkou výhodou algoritmu HC12 je jeho jednoduchost a velmi dobrá paralelizovatelnost.
- ❖ *Algoritmus GAHC, elitní turnajová selekce (ETS)*. V rámci snahy o vylepšení klasického genetického algoritmu byly autorem testovány různé přístupy. Jedním z výsledků je spojení klasického genetického algoritmu a algoritmu HC12. Tímto způsobem vznikl hybridní algoritmus GAHC. Navržená selekce ETS vylepšuje standardní turnajovou selekci tak, že zachovává její vlastnosti a navíc zaručuje zachování nejlepšího jedince.
- ❖ *Algoritmus GAFIS*. Jde o klasický genetický algoritmus, jehož parametry jsou v průběhu generací adaptovány s využitím fuzzy inferenčního systému (FIS).
- ❖ *Kritérium STE*. Při praktických experimentech na úlohách tzv. symbolické regrese nedávalo minimalizační kritérium typu nejmenších čtverců uspokojivé výsledky. Bylo tedy navrženo dynamické kritérium STE, které lépe zohlednilo shodu navrženého modelu s daty a navíc umožnilo dynamickou adaptaci tohoto kritéria.
- ❖ *Metodika optimální parametrizace řešičů GA, HC12, GAHC*. Přesto, že genetický algoritmus je poměrně robustní k volbě svých parametrů, existuje jejich optimální nastavení vzhledem k řešené úloze, nebo třídě úloh. Vliv nastavení lze ověřit v tzv. generačních řezech, ze kterých vyplývá vhodná parametrizace algoritmu.
- ❖ *Řešení úlohy nelineární aproximace s vazební podmínkou (RBF model)*. V této prezentaci jde pouze o stručné shrnutí možného řešení praktické úlohy nelineární aproximace dat s podmínkou omezení funkčních hodnot aproximace. V tomto praktickém příkladu byla užita klasická RBF neuronová síť genetickým učícím algoritmem.
- ❖ *Návrh interpretace problému QAP pomocí kódování swap2abs*. Výsledkem reformulace problému QAP je originální způsob kódování řešení úlohy QAP označený jako swap2abs. Kódování swap2abs je využitelné v případě dalších metaheuristik i exaktních metod.
- ❖ *Velmi efektivní řešení úloh třídy QAP esc*. S využitím kódování swap2abs a dobře optimalizovaného řešiče postaveného na algoritmech HC12 a GAHC, bylo dosaženo velmi efektivního řešení úloh třídy QAP esc. Řešení byla srovnána ve světovém měřítku.
- ❖ *Multi-optimální nastavení PID regulátoru pomocí metod CI*. Jako příklad možného využití prezentovaných metod počítačové inteligence, resp. algoritmu HC12, byla prezentována úloha nastavení PID regulátoru.
- ❖ *Komplexní návrh (ve struktuře i parametrech) polynomiálního regulátoru pomocí CI*. Závěrečná úloha prezentuje, z hlediska metod počítačové inteligence, unikátní návrh obecného lineárního regulátoru. Výsledkem prezentované metody je návrh modelu i parametrů regulátoru. Přístup je konkurenční k metodám GP a GE.

Důležitá je cesta, neboť cíl je iluze poutníka.

2 OPTIMALIZAČNÍ METAHEURISTIKY

Relativně dlouho byla problematika optimalizace řešena klasickým matematickým aparátem, založeným na infinitezimálním počtu a variačních metodách aplikovaných ve funkcionálních prostorech či s využitím klasických numerických metod. Tyto postupy sice umožňovaly a umožňují nalezení globálního extrému pro problémy jednoduššího charakteru, ale nikoliv pro problémy složitější, kde schopnost metody nalézt globální extrém není apriorně splněna. Výpočetní náročnost řešení dané optimalizační úlohy souvisí s její povahou, dimenzí, ale i s konkrétní definicí účelové funkce a omezení. Vlastní optimalizace je determinována užitým optimalizačním algoritmem a jeho implementací.

Nástup výpočetní techniky bezesporu přinesl nové možnosti uplatnění zavedených numerických optimalizačních metod. Soudobá výkonná výpočetní technika jednoznačně posouvá hranice možností, kde a jakými prostředky může být optimalizace uplatňována. Stran nových návrhů a implementací optimalizačních algoritmů zaznamenává markantní rozšíření oblast metaheuristik, evolučních výpočtů, soft computingu, počítačové a umělé inteligence. Nové výpočetní heuristiky jsou nasazovány na problémy v rozsahu, který byl ještě před dvaceti lety nemyslitelný. Budoucnost nových typů polovodičů, nových principů počítačů založených na bioinformatice, biologických neuronech, či kvantovém principu dává tušit, že doba revoluce v počítačové, resp. umělé inteligenci teprve přijde.

Při numerické optimalizaci se uplatňuje tzv. iterativní princip optimalizace, tedy optimalizace probíhá v krocích, tzv. iteracích. V každém iteračním kroku existuje řešení, které je pomocí optimalizační metody v následném kroku zlepšováno, dokud není dosaženo podmínky ukončení. Formálně jsou optimalizované proměnné uspořádány do vektoru \mathbf{x} , a na určité množině D přípustných hodnot extremalizujeme hodnotu navržené účelové funkce f . dále můžeme předpokládat, že účelová funkce $f(\mathbf{x})$ nabývá pro určitý vektor \mathbf{x}_{opt} globálního extrému vzhledem k D . Tato skutečnost se zapisuje v kompaktní formě dle (2.1) a označuje se jako obecná formulace optimalizačního problému.

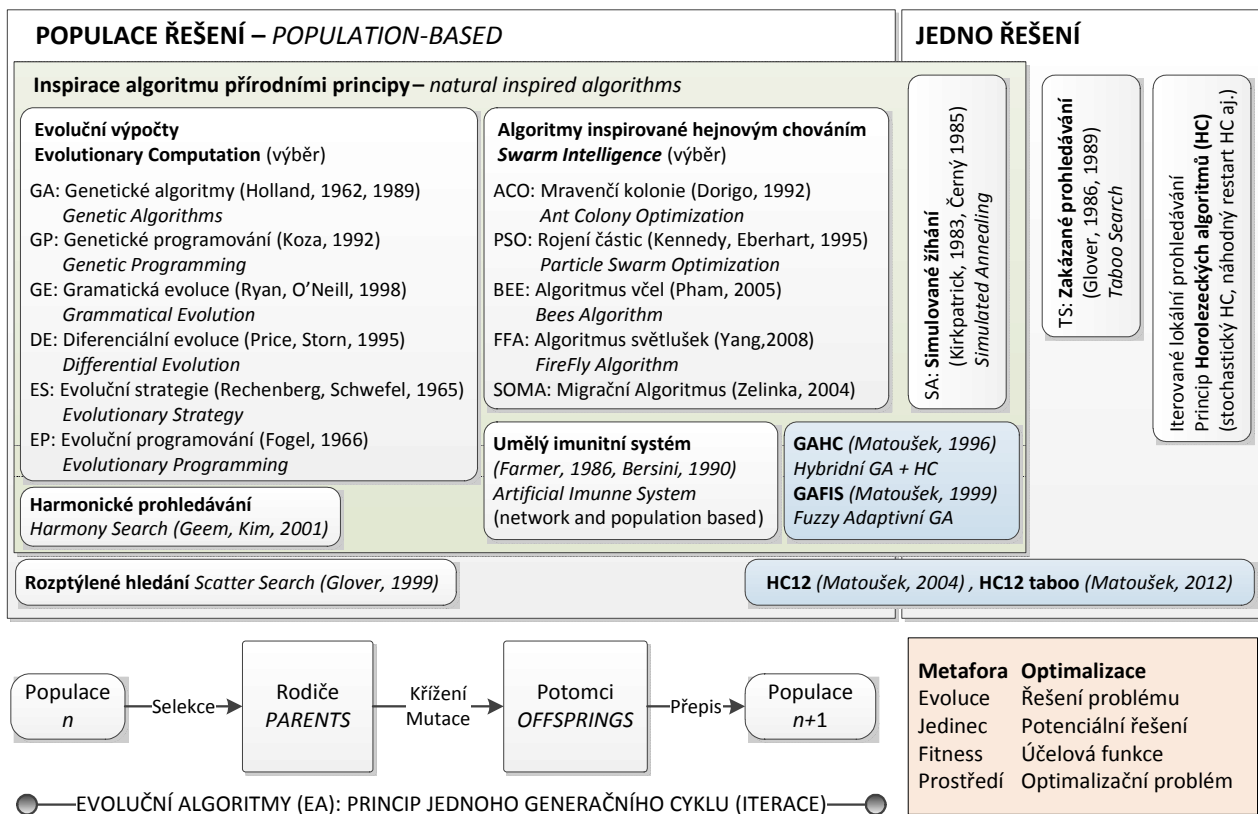
$$\mathbf{x}_{\text{opt}} = \arg \underset{\mathbf{x} \in D}{\text{opt}} f(\mathbf{x}) \quad (2.1)$$

2.1 ZÁKLADNÍ POJMY A ROZDĚLENÍ METOD

Obsah pojmů *Soft computing (SC)*, *Počítačová inteligence (Computational Intelligence, CI)* a *Umělá inteligence (Artificial Intelligence, AI)* není zcela jednotně chápán. Rovněž popis *heuristických* metod, či dále optimalizačních *metaheuristik* ve vztahu k SC, CI, a AI nabízí více pohledů.

Evoluce, genetika, imunitní systém, chemická regulace organismu, vyšší intelektuální schopnosti atd., to vše generuje různé třídy organismů schopné různé úrovně adaptace. Inspirací nemusí být jen živá příroda, ale obecně jakýkoliv systém vykazující chování vedoucí k minimu či maximu. Pokud v případě biologické inspirace dáme do role prostředí technický problém a do role organismu jeho počítačovou implementaci (model), mohou vzniknout velmi chytré optimalizační nástroje typu genetické algoritmy, diferenciální evoluce apod. Navržené postupy jsou všeobecně označovány jako *metaheuristiky*, přičemž prefix "meta" zdůrazňuje fakt vyšší sofistikovanosti příslušné metody v porovnání s běžným, obvykle problémově specifickým, heuristickým postupem.

Přehled základních optimalizačních metaheuristik je uveden obrázkem Obr. 2.1. Autor této práce v průběhu studia a výzkumu experimentoval s mnoha přístupy SC, resp. teorií, aplikací a implementací většiny uvedených metaheuristik.



Obr. 2.1: Přehled základních metaheuristik a podstata generačního cyklu.

2.2 REPREZENTACE A KÓDOVÁNÍ INFORMACE

Jedním z ústředních pojmů dále prezentovaných algoritmů je tzv. jedinec. Jedinec je reprezentantem jednoho konkrétního potenciálního řešení dané optimalizační úlohy. Kódová reprezentace jedince je základní vlastností implementace dále diskutovaných algoritmů. Ve všech dále uvedených případech metaheuristik bude jedince reprezentovat posloupnost bitů délky n . V kontextu teorie kódování [Adámek89] můžeme hovořit o reprezentaci binárním vektorem definovaným na abecedě $\{0,1\}$. V případě genetických algoritmů, resp. základní varianty GA představuje jedinec binární vektor (2.2).

$$\mathbf{a} = (a_1, a_2, \dots, a_n) \in \{0,1\}^n \quad (2.2)$$

$$\left| \{0,1\}^n \right| = 2^n \quad (2.3)$$

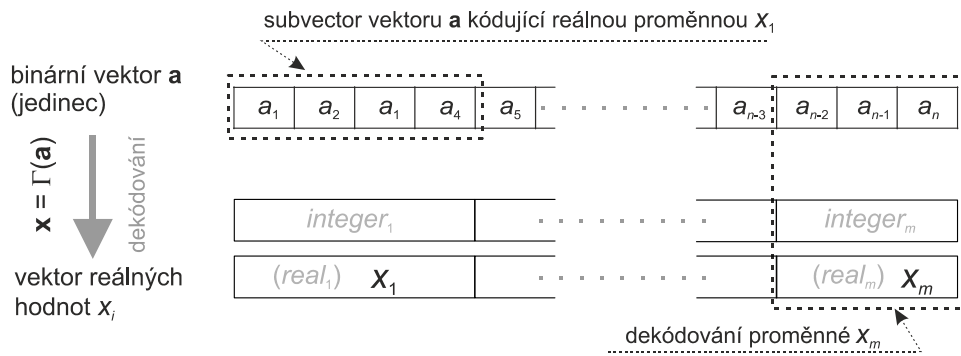
Kardinalita množiny těchto binárních vektorů je dána vztahem (2.3), z čehož vyplývá, že velikost prostoru všech možných binárních vektorů délky n roste s délkou n exponenciálně. Soubor jedinců můžeme logicky označit jako *populaci* $P = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N\}$, přičemž budeme uvažovat stejnou délku všech jedinců dle (2.4).

Z binárního řetězce \mathbf{a} , definovaného dle (2.2), je specifickou transformací Γ získán vektor optimalizovaných proměnných \mathbf{x} patřící do oblasti D . Transformace Γ je silně problémově vázaná. Z tohoto důvodu nyní pouze konstatujeme, že transformace Γ dekóduje binární řetězec na vektor reálných, resp. pseudo-reálných čísel.

$$\mathbf{P} = (a_{i,j})_{\substack{i \in \{1,2,\dots,N\} \\ j \in \{1,2,\dots,n\}}} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,n} \end{pmatrix} \quad (2.4)$$

$$\mathbf{x} = \Gamma(\mathbf{a}) \quad (2.5)$$

Standardně [Goldberg89, KvPoTi00] je původní binární řetězec \mathbf{a} dekomponován na m subvektorů stejné velikosti tak, že každý konkrétní subvektor kóduje konkrétní proměnnou x_i . Rozšířením této koncepce je implementace, kde subvektory mohou mít rozdílné velikosti, viz Obr. 2.2. Toto řešení může být pochopitelně pro určité typy problémů velmi výhodné. Například může existovat smíšeně celočíselná (*mix-integer*) úloha, která obsahuje celočíselné i reálné proměnné. S využitím rozdílné délky binárních subvektorů dosáhneme, v porovnání s případem kdy bychom volili vždy stejné délky subvektorů, mnohem výhodnějšího, příp. optimálního kódování.



Obr. 2.2: Schématická znázornění transformace binárního vektoru na vektor reálných hodnot.

Z hlediska kódování informace v jedinci a její následné reálné interpretaci je velmi přijatelné zavedení termínů *genotyp* a *fenotyp*. Tyto biologické termíny velmi dobře vystihují podstatu interpretace informace. Jedinec, jakožto potenciální řešení problému, bude v našem případě vždy kódován binárně, resp. binárním Grayovým kódem. Účelová funkce pro danou metaheuristiku však bude pracovat např. s reálnými hodnotami argumentů. Je tedy nutné provést překódování binárního vektoru na jinou formu, v tomto případě na vektor reálných hodnot, viz. Obr. 2.2. Genotyp je tedy zdrojové kódování a fenotyp je jeho reálná interpretace pro potřebu optimalizace, resp. účelovou funkci. Jak bylo naznačeno výše, řešený problém se v těchto zavedených metaforách stává prostředím, které vytváří selekční tlak na jedince a ten mu vzhledem k jeho síle k přežití, v biologii označované jako *fitness*, odolává.

Obor hodnot D dekódovaného vektoru \mathbf{x} ve vztahu (2.5) je obvykle dán (2.6) a představuje prostor možných řešení optimalizačního problému (2.1).

$$D = \prod_{i=1}^m [r_i, s_i] = [r_1, s_1] \times [r_2, s_2] \times \dots \times [r_m, s_m] \quad (2.6)$$

Účelová funkce f , definovaná na oblasti D , ohodnotí každého jedince z populace reálným číslem $f : D \rightarrow \mathbb{R}$.

Specifikem pro metaheuristiky využívající binární reprezentaci jedince, resp. binárního vektoru kódujícího řešení úlohy je, že reprezentace dané oblasti D definované na intervalech $[r_i, s_i]$ v (2.6), není vzhledem k reprezentaci (2.2) a konečné velikosti n spojitá.

$$\Gamma : \{0,1\}^n \rightarrow D \quad (2.7)$$

Oblast D je tedy prakticky aproximována ortogonální mřížkou. Po realizaci zobrazení Γ dále uvažujeme prostor D jako diskretizovaný. Pro případ kombinatorické optimalizace, či celočíselného programování, kde vystupují pouze diskrétní proměnné, neomezuje tato skutečnost kvalitu řešení a je v podstatě přínosem.

2.3 GRAYOVO KÓDOVÁNÍ A HAMMINGOVA METRIKA

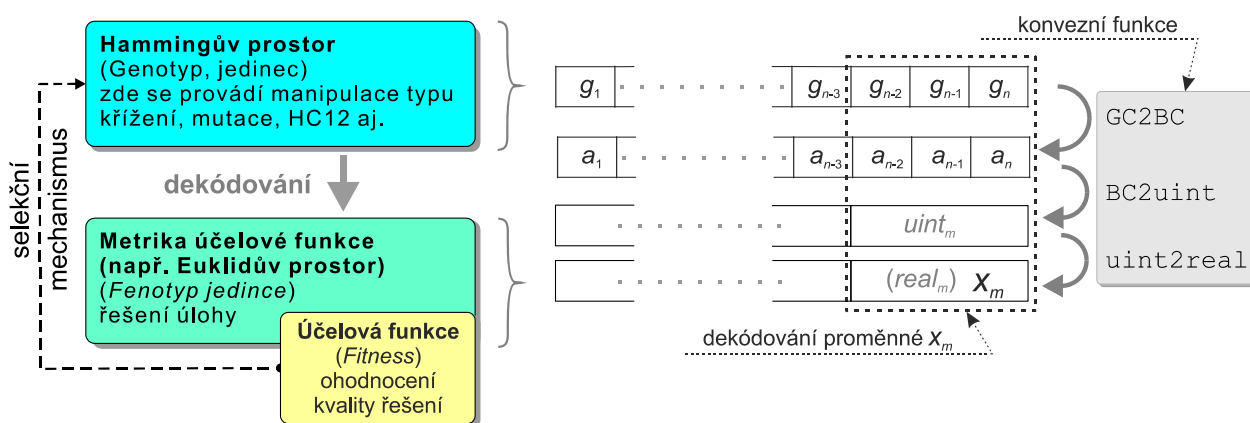
Zvolený způsob kódování každého jedince pomocí binárního vektoru implikuje otázku, jak kvantifikovat rozdíly jednotlivých jedinců v populaci na této základní úrovni.

Při splnění definice metriky je na množině $\{0,1\}^n \subset \mathbb{R}^n$ pro libovolné binární vektory délky n , např. $\mathbf{a}_1 = (a_{1,1}, \dots, a_{1,n})$ a $\mathbf{a}_2 = (a_{2,1}, \dots, a_{2,n})$ definována *Hammingova metrika*, tj. vzdálenost d_H (2.8) [Richard Wesley Hamming, 1950].

$$d_H(\mathbf{a}_1, \mathbf{a}_2) = \sum_{i=1}^n a_{1,i} \oplus a_{2,i} = \sum_{i=1}^n |a_{1,i} - a_{2,i}| \quad (2.8)$$

Z praktických důvodů je v dále prezentovaných algoritmech interpretován základní genotyp jako binární vektor v Grayově kódování. Implementace Grayova kódování je v případě binárně reprezentovaného řešení (binární vektor, jedinec, genotyp) realizována tak, že se apriorně uvažuje binární vektor \mathbf{g} v GC kódování, který se převede na vektor \mathbf{a} v BC kódování, který se následně dekóduje na číslo typu `uint`, resp. `real`

Na základě provedených experimentů lze konstatovat, že význam Grayova kódování je pro prezentované algoritmy CI zásadní. Návrhový vzor binárně reprezentovaných algoritmů třídy GA, HC12 uvádí Obr. 2.3.

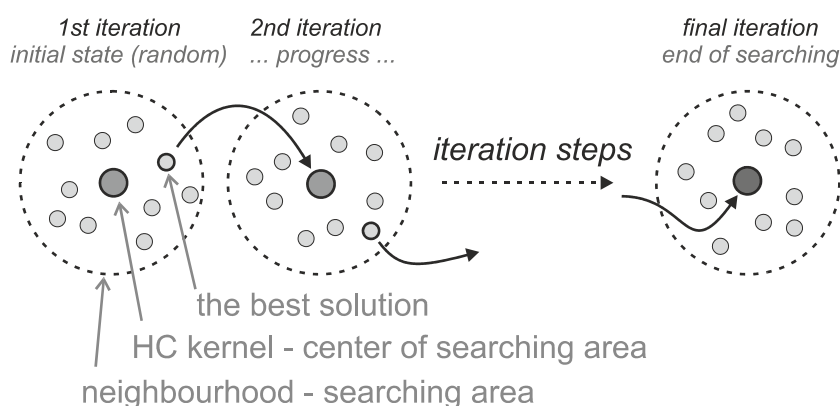


Obr. 2.3: Systém dekódování binárně reprezentovaného řešení a jeho vztah k základním operacím binárních metaheuristik (funkce GC2BC provádí konverzi z binárního Grayova kódu g do přímého binárního kódu a).

3 HC12: DOBRĚ PARALELIZOVATELNÁ METAHEURISTIKA

Metaheuristik je velmi mnoho, některé lze považovat za zcela univerzální, jiné jsou více problémově orientované. Výčet všemožných metaheuristik a jejich variant by byl obsahem samostatného knižního pojednání. Metaheuristiky pochopitelně užívají různých principů, viz například hladová heuristika, či dále uvedený genetický algoritmus. Jedním z obecných principů, nad kterým lze vystavět konkrétní metaheuristiky, je princip *Hill Climbing* (HC). Hill climbing princip se interpretuje jako princip horolezce a frekventovaně označuje jako *horolezecký algoritmus*.

Pokud bychom se pohybovali po hyperploše funkčních hodnot účelové funkce a náš horolezec by uměl prohledat "nějaké okolí" své aktuální působnosti, záviselo by nalezení extrému funkce, krom povahy funkce – tedy terénu, i na jeho první pozici (v kontextu řekněme na jeho "výsadku", jinak počátečním řešení) a na onom významu formulace "nějaké okolí". Situaci postupu znázorňuje následující Obr. 3.1.



Obr. 3.1: Schématické znázornění postupu horolezeckého algoritmu.

Z principu je zřejmé, že volba okolí může být velmi variabilní a že patrně zásadním způsobem ovlivňuje lokálnost vs. globálnost algoritmu ve vztahu k řešení úlohy. Z podstaty principu HC lze rovněž odvodit, že okolím nemusí být jen reálná reprezentace hodnot účelové funkce, ale i například kódová binární reprezentace problému.

Binární vektor původního (aktuálního) řešení budeme dále označovat jako jádro (*kernel*) a označovat indexem "ker", tedy například binární vektor \mathbf{a}_{ker} . Vygenerované okolí (*neighborhood*) tvoří soubor c nových binárních vektorů stejné délky n jako má vektor \mathbf{a}_{ker} . Tyto nové vektory bude dále reprezentovat matice $\mathbf{A}_o = (\mathbf{a}_1, \dots, \mathbf{a}_c)^T$.

V podstatě jakákoliv v literatuře prezentovaná varianta HC algoritmu je založena na odlišné metodě transformace vektoru \mathbf{a}_{ker} na c nových vektorů příslušného okolí, tedy je založena na způsobu definice relace sousednosti.

Základním krokem algoritmu je vygenerovat okolí původního řešení \mathbf{a}_{ker} na základě zvolené metodiky. Na „kvalitě“ tohoto okolí pak bude bezprostředně záviset lokálnost či globálnost uvedeného heuristického postupu. Pro další popis metody uvažujme standardní problém optimalizace funkce $f(\mathbf{x})$ na oblasti (2.6).

$$\min \{f(\mathbf{x}) \mid \mathbf{x} \in D\}, \text{ resp. } \max \{f(\mathbf{x}) \mid \mathbf{x} \in D\} \quad (3.1)$$

Tedy jde o hledání takového \mathbf{x}_{opt} , aby platilo:

$$\mathbf{x}_{opt} = \arg \min_{\mathbf{x} \in D} f(\mathbf{x}), \text{ resp. } \mathbf{x}_{opt} = \arg \max_{\mathbf{x} \in D} f(\mathbf{x}). \quad (3.2)$$

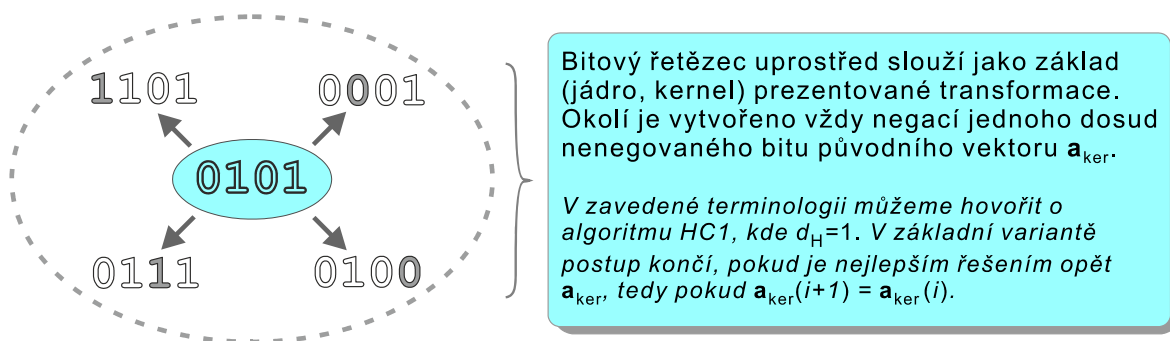
Diskretizace oblasti $D \subset \mathbb{R}$ je dána zobrazením $\Gamma: \{0,1\}^n \rightarrow D$ (2.7), tedy zavedenou binární reprezentací \mathbf{a} reálných proměnných \mathbf{x} . Protože platí $\mathbf{x} = \Gamma(\mathbf{a})$, považujeme dále za optimální řešení minimalizační úlohy (3.1) řešení dané následujícím vztahem (3.3).

$$\mathbf{a}_{opt} = \arg \min_{\mathbf{a} \in \{0,1\}^n} f(\Gamma(\mathbf{a})) \quad (3.3)$$

Nad touto binární reprezentací definujeme určitou *relaci susednosti*, která pro každé přípustné \mathbf{a}_{ker} umožňuje stanovit „okolí“ (sousedství) tvořené body $\mathbf{a} \in S(\mathbf{a}_{ker})$. Volba transformační funkce S bude determinovat chování a charakter HC algoritmu.

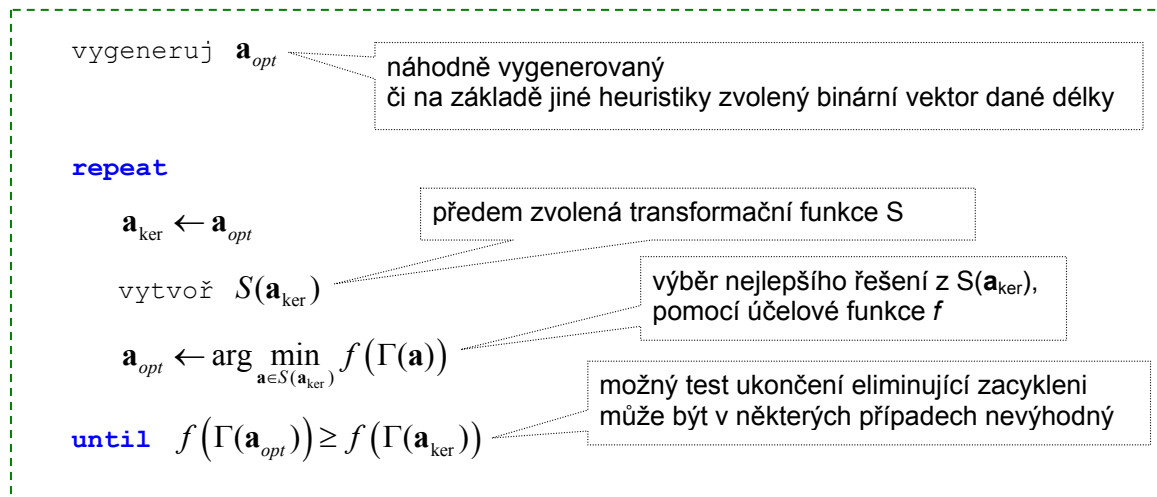
3.1 PRINCIP HC12

Nyní uvažujme následující postup, kdy pro zvolený binární vektor délky 4 bitů vygenerujeme okolí \mathbf{A}_0 způsobem dle Obr. 3.2.



Obr. 3.2: Generování okolí 4 bitového řetězce pomocí jednobitové inverze.

Jak je patrné z Obr. 3.2, je při daném způsobu tvorby okolí každý vektor \mathbf{a}_i vzdálen od \mathbf{a}_{ker} vždy o $d_H = 1$. Při délce n binárního řetězce \mathbf{a}_{ker} tedy obdržíme n vektorů okolí \mathbf{a}_i . Takovýto algoritmus tvorby okolí byl označen jako HC1. Obecně může být tvorba okolí realizována nějakým exaktním nebo heuristickým způsobem jak ukazuje obrázek Obr. 3.2, nebo může být okolí \mathbf{a}_{ker} generováno zcela náhodně. Jak ukázaly praktické experimenty, je při stejné kardinalitě množiny okolí podstatně výhodnější využít systém tvorby okolí dále prezentovaný třídou algoritmů HC12, než volit náhodně generované okolí. Poznamenejme, že prezentovaná implementace algoritmů třídy HC12 nepřipouští zhoršení hodnot účelové funkce a iterační cyklus algoritmu je zastaven, pokud není danou transformací nalezeno lepší řešení než je \mathbf{a}_{ker} . Tento bod není v kontradikci s možným použitím principu Taboo search do dalších implementací. Pseudokód obecného vzoru binárního HC algoritmu, který je plně slučitelný s algoritmy třídy HC12, je uveden dle na Obr. 3.3.



Obr. 3.3: Pseudo-kód: návrhový vzor binárního HC algoritmu pro úlohu minimalizace. Ukončovací podmínka zde nepřipouští více stejných hodnot účelové funkce. Transformace S může generovat okolí, tj. populaci řešení například dle schématu třídy algoritmů HC12, ale také zcela náhodně či dle jiné heuristiky. Pokud by na toto schéma byl aplikován *taboo search* princip, nemusela by ukončovací podmínka nutně znamenat konec výpočtu. Rovněž by byla ovlivňována množina přípustných řešení generovaná funkcí S .

3.2 FORMALIZACE HC12

Dále popsané algoritmy třídy HC12 jsou založeny na libovolné, ale pevně dané množině transformací příslušných binárních vektorů (3.4).

$$\mathbf{a} \in \{0,1\}^n, \quad \text{pro } n \in \mathbb{N} \quad (n \dots \text{délka binárního vektoru}) \quad (3.4)$$

Množinu navržených transformací označme H a jednotlivé transformace jako s .

$$H = \{s_0, s_1, \dots, s_n\} \quad (3.5)$$

Užitím transformace $s \in H$ realizujeme zobrazení binárního vektoru \mathbf{a}_{ker} na množinu A_o binárních vektorů \mathbf{a} , dále též označovanou jako matici \mathbf{A}_o (index "o" jako okolí).

$$A_o = \{\mathbf{a}_1, \dots, \mathbf{a}_c\}, \quad \mathbf{A}_o = \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_c \end{pmatrix}, \quad \text{pro } c \in \mathbb{N} \quad (3.6)$$

$$s : \mathbf{a}_{ker} \rightarrow A_o, \quad \text{tedy } s : \{0,1\}^n \rightarrow \left(\{0,1\}^n\right)^c \quad (3.7)$$

Kardinalita c množiny A_o je dána zvolenou transformací s_k a délkou n binárního vektoru \mathbf{a}_{ker} .

$$c_k(s_k, n) = |A_o| = \binom{n}{k}, \quad \text{pro } k \in \{0,1, \dots, n\}, \quad (3.8)$$

kde index k označuje příslušnost ke konkrétnímu prvku z množiny H dle (3.5).

K realizaci množiny transformací H je zaveden systém matic \mathbf{M} . O matici \mathbf{M} příslušející dané transformaci s_k budeme hovořit jako o \mathbf{M} matici k -tého řádu a označíme ji \mathbf{M}_k (tento řád není totožný s řádem čtvercové matice).

$$\{s_k \mid k \in \{0, 1, \dots, n\}\} \Rightarrow \mathbf{M}_k \quad (3.9)$$

Definice: \mathbf{M} -matice řádu k , zkráceně \mathbf{M}_k , je taková matice, jejíž řádky reprezentují všechny body Hammingova metrického prostoru \mathbf{H}^n se vzdálenostmi k od počátku (tj. nulového vektoru délky n) ve smyslu Hammingovy metriky d_H .

□

Schéma možné konstrukce \mathbf{M} matic je následující:

$$\begin{aligned} \mathbf{M}_0 &= (0_{1,1} \quad 0_{1,2} \quad \dots \quad 0_{1,n}) \\ \mathbf{M}_1 &= \begin{pmatrix} 1_{1,1} & 0_{1,2} & \dots & 0_{1,n} \\ 0_{2,1} & 1_{2,2} & & \\ \vdots & & \ddots & \\ 0_{c_1,1} & & & 1_{c_1,n} \end{pmatrix} \\ \mathbf{M}_2 &= \begin{pmatrix} 1_{1,1} & 1_{1,2} & 0_{1,3} & \dots & 0_{1,n} \\ 1_{2,1} & 0_{2,2} & 1_{2,3} & & 0_{2,n} \\ \vdots & & & \ddots & \\ 0_{c_2,1} & & 1_{c_2,n-1} & 1_{c_2,n} \end{pmatrix} \\ \mathbf{M}_3 &= \begin{pmatrix} 1_{1,1} & 1_{1,2} & 1_{1,3} & 0_{1,4} & \dots & 0_{1,n} \\ 1_{2,1} & 1_{2,2} & 0_{2,3} & 1_{2,4} & & 0_{2,n} \\ \vdots & & & & \ddots & \\ 0_{c_3,1} & & 1_{c_3,n-2} & 1_{c_3,n-1} & 1_{c_3,n} \end{pmatrix} \\ &\vdots \\ \mathbf{M}_{n-1} &= \begin{pmatrix} 0_{1,1} & 1_{1,2} & \dots & 1_{1,n} \\ 1_{2,1} & 0_{2,2} & & \\ \vdots & & \ddots & \\ 1_{c_{n-1},1} & & & 0_{c_{n-1},n} \end{pmatrix} \\ \mathbf{M}_n &= (1_{1,1} \quad 1_{1,2} \quad \dots \quad 1_{1,n}) \end{aligned} \quad (3.10)$$

K vytvoření matice \mathbf{A}_0 je pro dále prezentovaný systém výpočtů třeba zavést operaci s významem tzv. *replikace vektoru* \mathbf{a}_{ker} . Tato operace vytvoří matici \mathbf{A}_{ker} obsahující po řádcích identické kopie binárního vektoru \mathbf{a}_{ker} (3.11). Počet řádků této matice odpovídá počtu řádků příslušné \mathbf{M} matice a tedy kardinalitě c cílové množiny A_0 dle (3.8).

$$\mathbf{A}_{\text{ker}} = (\mathbf{a}_{1,\text{ker}}, \dots, \mathbf{a}_{c,\text{ker}})^T \quad (3.11)$$

Nyní můžeme pomocí (3.9), respektive (3.10) realizovat příslušnou transformaci s (3.7), dále označovanou jako *generující transformaci* s .

$$s_k : \mathbf{A}_0 = \mathbf{A}_{\text{ker}} \oplus \mathbf{M}_k, \quad \text{pro } s_k \in H \text{ a } k = 0, 1, \dots, n \quad (3.12)$$

O transformaci s_k lze říci, že generuje úplnou množinu vektorů, které jsou ve smyslu metriky d_H (2.8) vzdáleny od počátku o hodnotu k .

$$s_k \in H \Rightarrow d_H(\mathbf{a}_{\text{ker}}, \mathbf{a}) = k, \quad \text{pro } \forall \mathbf{a} \in A_0 \quad (3.13)$$

Zobecnění vztahu (3.12) pro libovolný, ale pevně daný výběr prvků z množiny H , je zřejmé. Množinu vybraných, a dále pro prezentovaný HC algoritmus (Obr. 3.3) pevně daných transformací označíme H_{select} .

$$H_{\text{select}} \subseteq H \quad (3.14)$$

Množina H_{select} vzájemně jednoznačně určuje transformaci S , která je sjednocením vybraných transformací z množiny H .

$$S : \mathbf{A}_S = \begin{pmatrix} \mathbf{A}_{\text{ker},k_1} \oplus \mathbf{M}_{k_1} \\ \vdots \\ \mathbf{A}_{\text{ker},k_{\text{select}}} \oplus \mathbf{M}_{k_{\text{select}}} \end{pmatrix}, \quad (3.15)$$

kde $k_i \in I$ a I je indexová množina vybraných prvků z množiny H definovaná výběrem H_{select} .

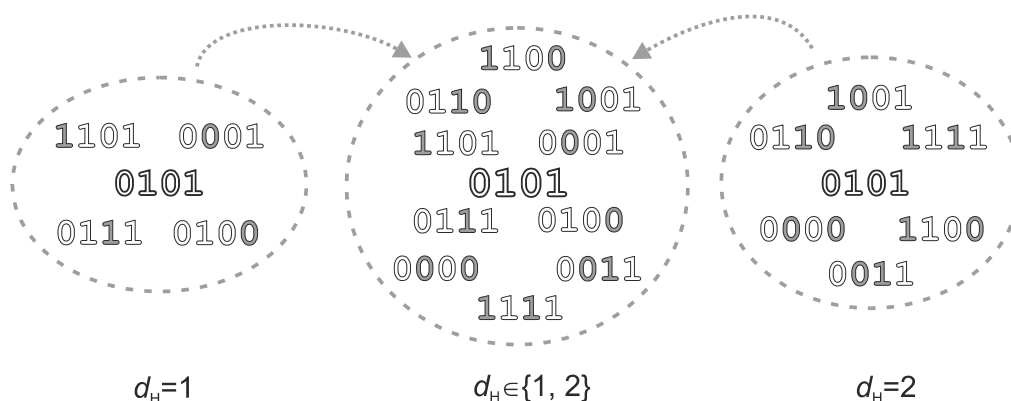
Praktické testy [A02, A03] i jednoduché kombinatorické úvahy vzhledem k (3.15) vedly k realizaci algoritmu HC12 jako nejefektivnější varianty. HC12 je kompromisem mezi únosnou rychlostí algoritmu a efektivitou prohledávání prostoru řešení. Základní charakteristikou a odlišností původního algoritmu HC12 od jiných metaheuristik je systém kódování a transformací.

Implementace HC12 jsou založeny na restartech, což významně posiluje schopnost algoritmu nalézat globální řešení. Pro aplikaci HC12 algoritmů musí být prakticky zvážen typ úlohy a požadovaná přesnost výpočtu, dále výkon výpočetní platformy a typ implementace.

Jedinečnost algoritmu HC12 spočívá v kombinaci specifického kódování problému a způsobu generování populace řešení, tedy tzv. okolí. Na těchto principech byl prakticky zrealizován perspektivní optimalizační algoritmus, který přes svoji jednoduchost prokazuje až překvapivě dobré vlastnosti. Algoritmus HC12 umí efektivně prohledávat prostor a nalézat řešení spojitých i kombinatorických problémů, navíc, a to je velmi poplatné dnešní době, je velmi dobře škálovatelný.

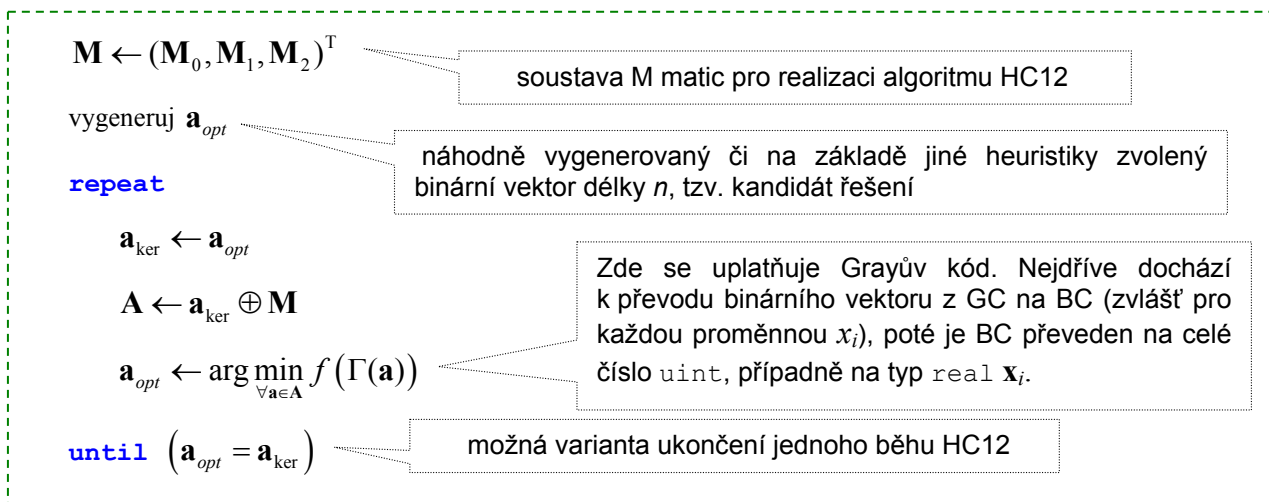
Implementované varianty HC12 disponují možnostmi:

- Volby účelové funkce s případnou parametrizací (například čas) a definici počtu optimalizovaných proměnných omezenou pouze dostupnými zdroji, viz dále.
- Volby rozsahu binární reprezentace pro každou reálnou proměnnou, což u spojitých úloh determinuje přesnost výpočtu. Vzhledem k nejefektivnější implementaci je tento parametr určen maximálním rozsahem datového typu `uint32`, tedy délkou 32 bit/proměnnou.
- Volby definičních intervalů $[x_{\min,i}, x_{\max,i}]$ všech optimalizovaných proměnných x_i úlohy, tedy v podstatě volby omezujících podmínek typu $x_{\min,i} \leq x_i \leq x_{\max,i}$. Tím obdržíme omezení (2.6) geometricky interpretovatelné jako hyperkvádr. V případě zadání pouze jednoho intervalu je tento interval automaticky použit na všechny optimalizované proměnné.



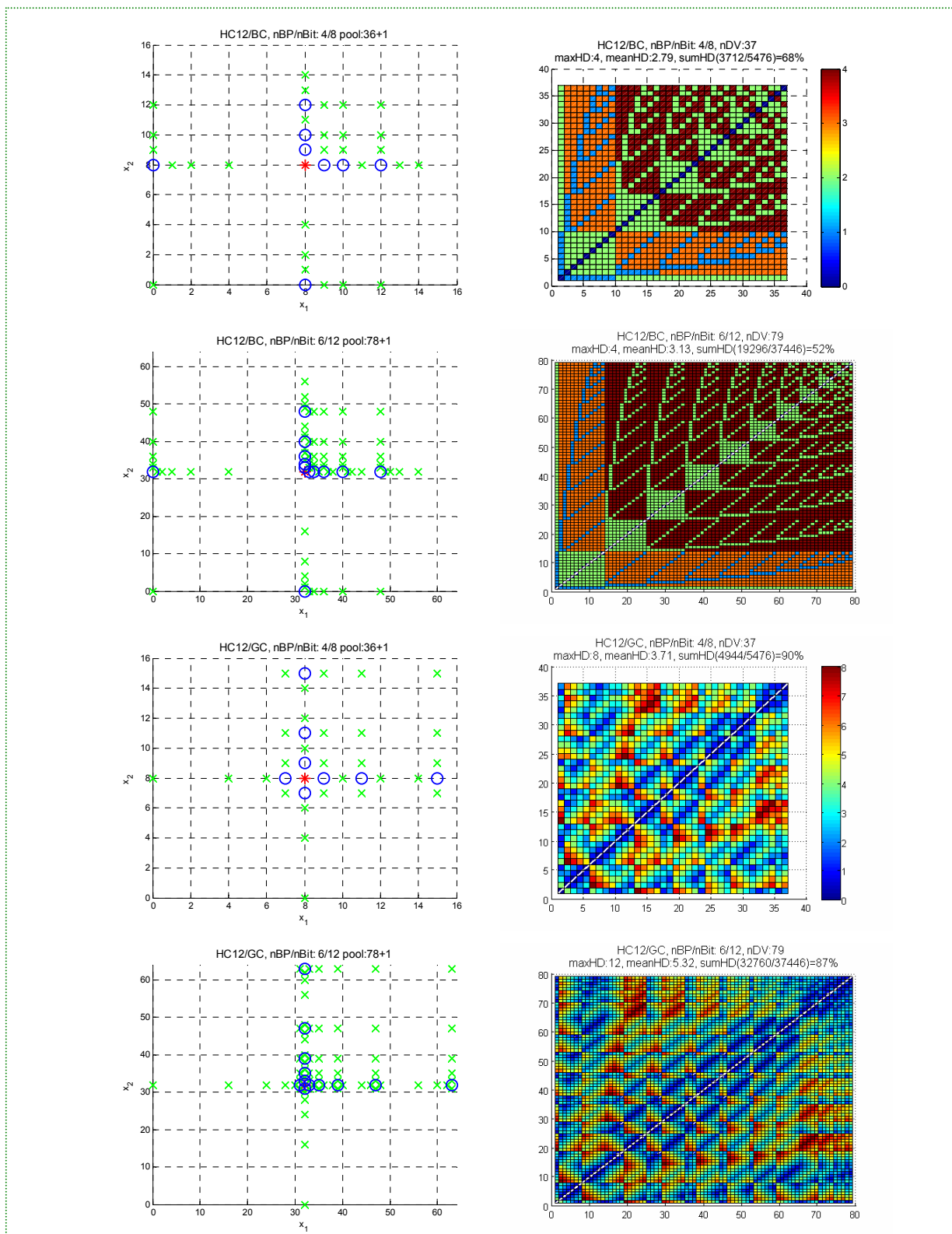
Obr. 3.4: Příklad generování okolí na 4b řetězci dle algoritmu HC12. Binární řetězec uprostřed slouží jako základ transformací (jádro) \mathbf{a}_{ker} , ke kterému se generuje okolí se vzdáleností $d_H = 1$ a $d_H = 2$.

Soubor transformací, resp. algoritmus HC12 dokumentovaný dle Obr. 3.4 a popsaný soustavou transformací S dle (3.15) můžeme popsat dle Obr. 3.5.

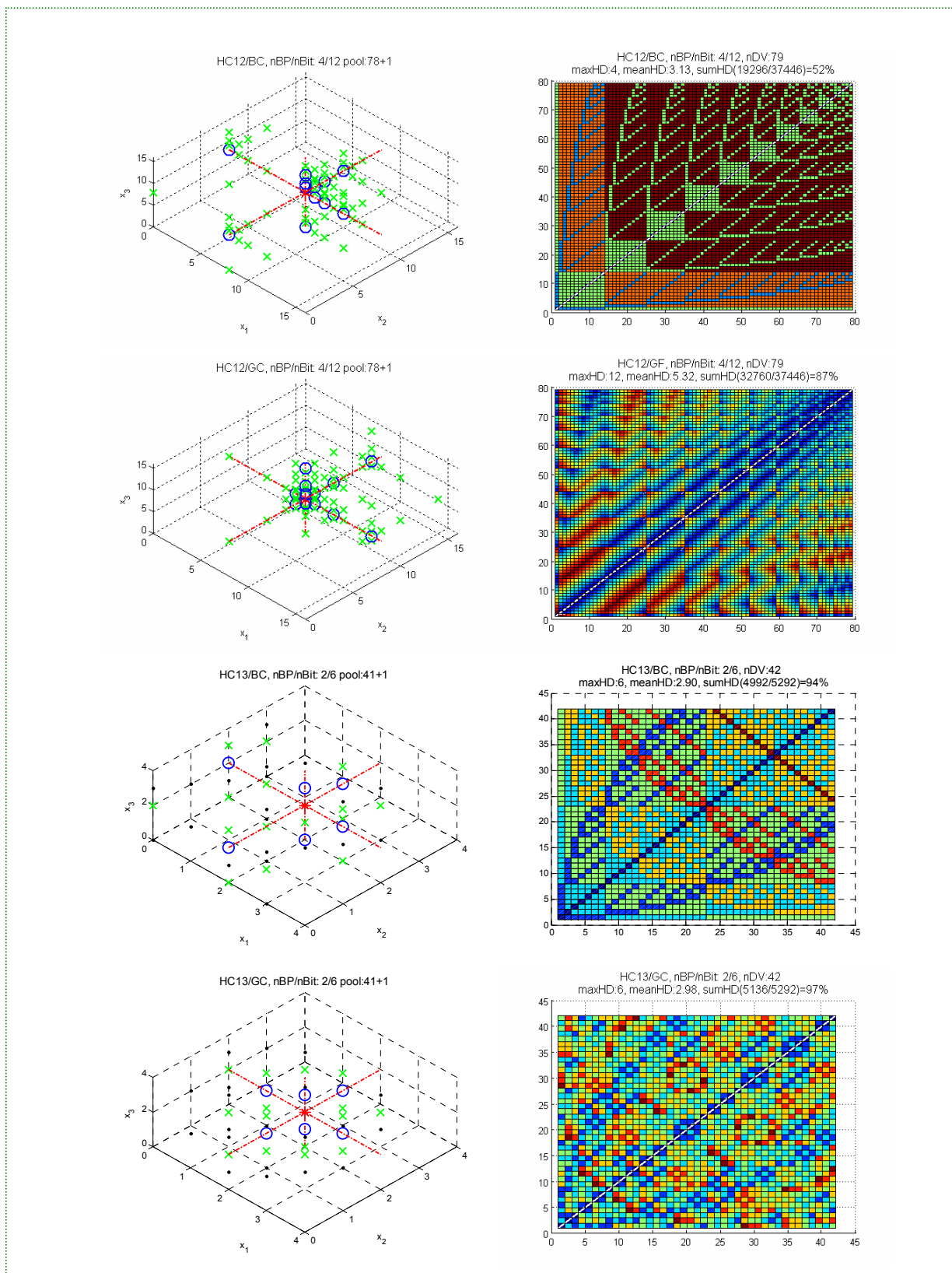


Obr. 3.5: Pseudo-kód: návrhový vzor binárního HC12 algoritmu pro úlohu minimalizace. Ukončovací podmínka zde zahrnuje test stejných hodnot binárních vektorů, tedy nepřipouští "pohyb" algoritmu v rovině s ekvivalentní hodnotou účelové funkce. Rovněž by mohl být kontrolován např. počet iterací nebo čas.

Pseudo-kód (vzor) dle Obr. 3.5 prezentuje základní variantu algoritmu HC12, která je prakticky implementovaná v cyklu `o nRun` restartech. Vektor \mathbf{a}_{ker} je pro každý restart generován náhodně, či podle jiné heuristiky.



Obr. 3.6: Generovaná okolí k bodu „*“ o souřadnicích $[x_1, x_2]$ vzniklá užitím algoritmu HC12 (vlevo). Okolí odpovídá bodům „○“ o vzdálenosti $d_H=1$ a bodům „X“ o vzdálenosti $d_H=2$. Zajímavá je vizuální interpretace odpovídajících matic vzdáleností (vpravo) binárních vektorů tvořících okolí, která tvoří strukturu typu IFS fraktálu. Generátorem této struktury je systém transformací uplatňovaný u algoritmu HC12. První dva řádky odpovídají BC kódování, zbývající pak odpovídají GC kódování. Liché, resp. sudé řádky odpovídají bitové reprezentaci 4bitů, resp. 6bitů na jednu proměnnou typu `uint`.

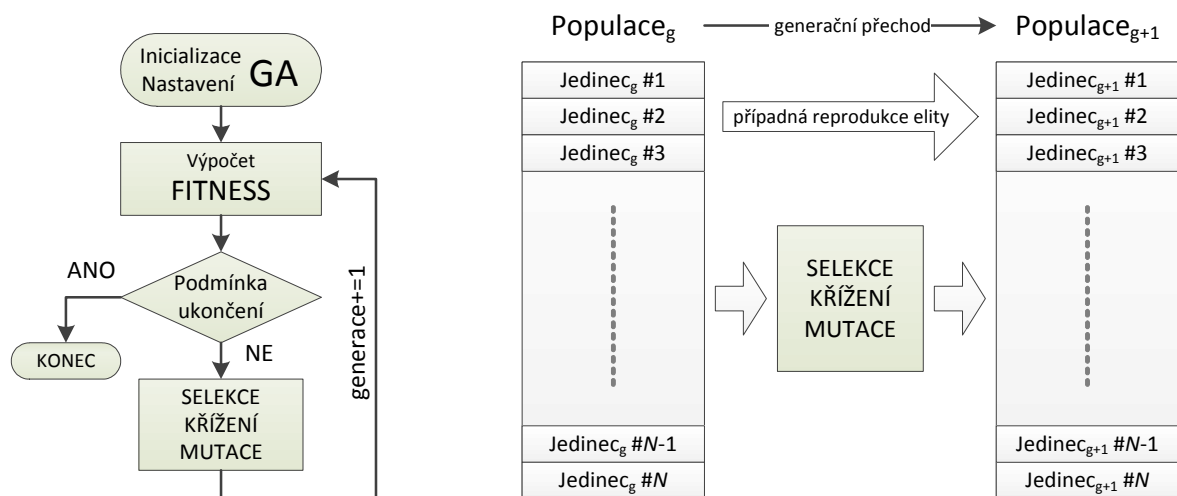


Obr. 3.7: Generovaná 3D okolí k bodu „*“ o souřadnicích $[x_1, x_2, x_3]$ vzniklá užitím algoritmu HC12 (první dva řádky) a algoritmu HC13 (následující dva řádky). Okolí odpovídá bodům „○“ o vzdálenosti $d_H=1$, bodům „X“ o vzdálenosti $d_H=2$ a bodům „•“ o vzdálenosti $d_H=3$. Zajímavá je vizuální interpretace odpovídajících matic vzdáleností (vpravo) binárních vektorů tvořících okolí, která tvoří strukturu typu IFS fraktálu. Liché, resp. sudé řádky odpovídají BC, resp. GC kódování.

4 GENETICKÝ ALGORITMUS

Genetické algoritmy jsou charakteristické počtem N možných řešení optimalizačního problému. Tato jednotlivá řešení jsou reprezentována *jedinci*, soubor N jedinců je označován jako *populace* \mathbf{P} . Z matematického pohledu můžeme GA chápat jako soubor transformací, které pracují nad populací \mathbf{P} . Tyto transformace v cyklu generačních kroků (generací) g , převedou populaci \mathbf{P}_g do nového stavu \mathbf{P}_{g+1} . Standardní vlastností GA je, že při změně z generace g na $g+1$ je zachován počet jedinců N i jejich binární délka n .

Některé významné modifikace GA jako je třeba Messy GA [Goldberg89] používají proměnlivou délku jedince. Rovněž samostatný evoluční algoritmus inspirovaný GA, kterým je genetické programování (*genetic programming*, GP) [Koza92] reprezentuje jedince pomocí stromové struktury o proměnné velikosti. Základní schéma genetického algoritmu je patrné z Obr. 4.1. Genetické algoritmy využívají základní *populaci*, která je pomocí operátorů *selektce*, *křížení* a *mutace* modifikována.



Obr. 4.1: Vývojový diagram GA. Výpočet fitness je prakticky interpretován jako výpočet hodnot účelové funkce. Operátory **selektce** upřednostňují jedince s lepší hodnotou fitness, kteří pak mohou podstoupit další dvě operace a to **křížení** a **mutaci** (někdy označeno jako rekombinace). Po provedení operací (transformací): selektce, křížení a mutace, se provede vyhodnocení nové fitness čímž vstupuje populace do další generace. Tento klasický mechanismus připouští zhoršení hodnot účelové funkce (tedy fitness) a je třeba si nejlepší nalezená řešení v každé generaci ukládat. Druhou možností je zavedení tzv. **elitismu**, kdy je zvolený soubor nejlepších jedinců v každé iteraci chráněn před důsledky operátorů selektce, křížení mutace a až po jejich vykonání jsou tito "elitní" jedinci zpět zařazeni do populace na úkor jiných (zpravidla nejhorších) jedinců. Tímto způsobem je zabráněno zhoršení hodnot účelové funkce vzhledem ke generacím.

4.1 SELEKČNÍ MECHANISMUS

Selektce představuje výběr jedinců ze stávající populace \mathbf{P}_{BS} a jejich přepis do časově následující populace \mathbf{P}_{AS} . Tento výběr jedinců je klíčovou částí GA.

Z více hledisek je nejvhodnějším selekčním mechanismem tzv. turnajový výběr, či **turnajová selektce (TS) – tournament selection**. Tento typ selektce pracuje následujícím způsobem: vybereme náhodně t jedinců z populace \mathbf{P}_{BS} čítající N jedinců, přičemž obecně platí $2 \leq t \ll N$. Nejlepšího z takto vybrané skupiny t jedinců zkopírujeme do populace následující \mathbf{P}_{AS} . Proces opakujeme N krát. Častou variantou tohoto postupu je tzv. *binární výběr*, tedy výběr pro $t=2$. Hodnota t udávající velikost „bojující“ skupiny je označována jako *velikost turnaje (tournament size)*, přičemž je rovněž možné mluvit přímo o *síle selektce*.

Z biologického hlediska tato metoda velmi věrně imituje přírodní výběr, který se řídí pravidly, která lze ve svých důsledcích shrnout jako „boj o přežití“. Z hlediska implementace, resp. rychlosti algoritmu, je podstatnou výhodou turnajové selekce to, že algoritmus ke své činnosti nepotřebuje provádět setřídění populace ani počítat výběrové pravděpodobnosti. Výpočetní složitost tohoto algoritmu je $\mathcal{O}(N)$.

% tournament selection (TS)	% elite tournament selection (ETS)
Input: \mathbf{P}_{BS} populace před selekcí, t velikost turnaje	Input: \mathbf{P}_{BS} populace před selekcí, t velikost turnaje
Output: \mathbf{P}_{AS} populace po selekci	Output: \mathbf{P}_{AS} populace po selekci
$\mathbf{P}_{AS} \leftarrow \text{tournament}(\mathbf{P}_{BS}, t)$	$\mathbf{P}_{AS} \leftarrow \text{tournament}(\mathbf{P}_{BS}, t)$
for $i=1:N$	for $i=1:N$
$\mathbf{I}'_i \leftarrow$ nejlepší jedinec z množiny t náhodně vybraných jedinců populace \mathbf{P}_{BS}	$\mathbf{I}'_i \leftarrow$ nejlepší jedinec z množiny tvořené jedincem \mathbf{I}_i a dále $t-1$ náhodně vybranými jedinci populace \mathbf{P}_{BS}
end	end

Obr. 4.2: Algoritmus TS a originální varianta tzv. elitního turnajového výběru (elite tournament selection, ETS).

Pravděpodobně jedinou nevýhodou turnajové selekce tak zůstává fakt, že tento algoritmus nemůže ze své podstaty zaručit zachování nejlepšího jedince z populace \mathbf{P}_{BS} . Z tohoto důvodu byla navržena modifikace, která byla v kontextu zachování elitního jedince nazvána jako **elitní turnajová selekce (ETS)** – *elite tournament selection*, viz algoritmus dle Obr. 4.2. ETS je realizována stejným mechanismem jako původní varianta TS s rozdílem ve výběru jedinců. Výběr prvního jedince do skupiny t jedinců je realizován deterministicky tak, aby postupně „oslovil“ všech N jedinců v populaci. Ke každému takto pevně zvolenému jedinci je dle velikosti turnaje t přiřazována další, již náhodně vybraná skupina $t-1$ jedinců populace \mathbf{P}_{BS} a je učiněn standardní turnajový výběr, viz pravá část Obr. 4.2. Tímto způsobem je zajištěno, že v rámci selekčního mechanismu nemůže dojít ke ztrátě nejlepšího jedince, přičemž efektivita algoritmu zůstala zachována (opět není nutné setřídění populace, tedy složitost je $\mathcal{O}(N)$).

Statistická analýza ETS je založena na srovnání teoretických i empirických distribucí elitní a standardní turnajové selekce. Předpokládejme, že při volbě jedinců pro turnaj užíváme náhodný generátor s uniformní distribucí. Tedy i pravděpodobnost náhodného výběru t jedinců je stejná jak pro seřazenou (dle fitness) tak neseřazenou populaci (což je pro turnajovou selekci běžné). Pro lepší představu a vyjádření následujících vztahů můžeme bez ztráty adekvátnosti popisu přepokládat, že výběr t proběhne v seřazené populaci, kde budou pro všechny jedince rozdílné hodnoty fitness. Dále budeme sledovat skupinu řekněme k jedinců a popíšeme jejich distribuci v kontextu jejich pravděpodobnosti dostat se z populace \mathbf{P}_{BS} do populace \mathbf{P}_{AS} za předpokladu, že existuje l lepších jedinců a j horších jedinců z celkového počtu N jedinců.

S využitím binomického rozdělení obdržíme pro TS pravděpodobnostní funkci (4.1).

$$p(z) = \binom{N}{z} \left[\left(\frac{j+k}{N} \right)^t - \left(\frac{j}{N} \right)^t \right]^z \left[1 - \left(\frac{j+k}{N} \right)^t - \left(\frac{j}{N} \right)^t \right]^{t-z}, \quad z = 0, 1, \dots, N \quad (4.1)$$

Pravděpodobnostní funkce $p(Z)$ vyjádřená jako (4.2) pak popisuje distribuci jedinců pro elitní turnajovou selekci, kde Z je daný počet kopií sledovaných jedinců, kteří skončí v \mathbf{P}_{AS} .

$$p(z) = \sum_{\substack{x=0, \dots, j \\ y=0, \dots, k \\ z=x+y}} \binom{j}{x} \left[\left(\frac{j+k}{N} \right)^{t-1} - \left(\frac{j}{N} \right)^{t-1} \right]^x \left[1 - \left(\frac{j+k}{N} \right)^{t-1} \left(\frac{j}{N} \right)^{t-1} \right]^{j-x} \cdot \binom{k}{y} \left[\left(\frac{j+k}{N} \right)^{t-1} \right]^y \left[1 - \left(\frac{j+k}{N} \right)^{t-1} \right]^{k-y} \quad (4.2)$$

4.2 GAHC: HYBRIDNÍ GENETICKÝ ALGORITMUS

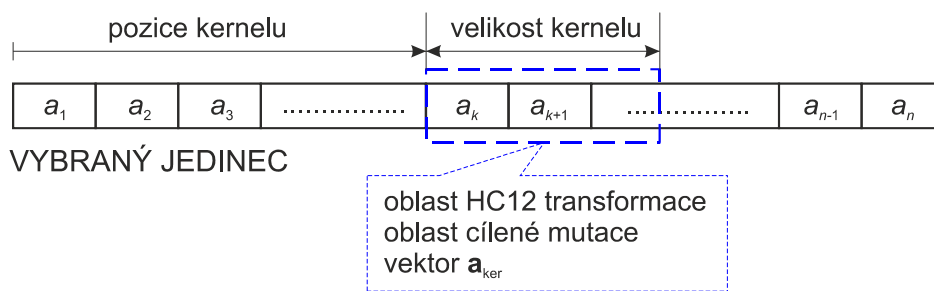
V literatuře lze nalézt různé hybridní způsoby koexistence dvou optimalizačních algoritmů, např. GA a dalších optimalizačních technik. V případě GA algoritmu můžeme např. na dané úloze využít jeho globálně optimalizační potenciál a získané řešení použít jako první iteraci nějaké jiné optimalizační metody. Dobrým příkladem může být GA a Nelder-Meadův algoritmus, které jsou velmi efektivně implementovány v prostředí MATLAB, kde je tak tato koexistence, resp. implementace bezproblémová. V tomto případě ani jeden z algoritmů neomezuje druhý a každý maximálně těží potenciál ze své domény. Uvedený způsob se diskutuje jako víceúrovňová optimalizace.

Druhou možností hybridizace dvou algoritmů, z nichž jeden je GA, je založena na faktu, že GA manipuluje s množinou potenciálních řešení, tedy s jedinci. Častou hybridizací je v tomto případě nasazení další optimalizační techniky přímo na jedince v populaci. Princip této suboptimalizace je většinou takový, že se po určitém počtu generací vybere určitý počet jedinců, kteří se jako celek jinou metodou zoptimalizují a poté dál existují v populaci ve své nové "kondici". Tento způsob může vést k nežádoucí předčasné konvergenci. Důvod je prostý, je to velmi rychlé vyčerpání potenciálně vhodného genetického materiálu, či jinými slovy rychlý zánik schémat.

Při hybridizaci označené **GAHC** jsou zachovány všechny principy standardního GA. Hybridizace je realizována prostřednictvím dalšího operátoru mutace, který je aplikován s danou pravděpodobností obvyklou u mutačních operátorů. Z tohoto pohledu se tedy jedná o **cílenou mutaci**.

Jedinečnost algoritmu GAHC spočívá v kombinaci silné globální metaheuristiky, kterou bezesporu genetické algoritmy jsou, spolu s lokálněji chápanou optimalizací ve formě mutace realizované pomocí HC12. Toto spojení je inspirující i pro jiné varianty synergismu optimalizačních metod. U této realizace je důležité, že se zachovává binární báze kódování problému, globálnost prohledávání GA a síla konvergence algoritmu HC12.

GAHC představuje velmi efektivní spojení genetického algoritmu a algoritmu HC12. Spojení je realizováno prostřednictvím, za tím účelem vytvořeného, mutačního operátoru. Algoritmus HC12 provádí cílenou mutaci malého rozsahu v prostoru populace. Počet vybraných jedinců určených pro HC12 optimalizaci odpovídá zvoleným pravděpodobnostním parametrům. Další parametry a princip GAHC je nejlépe patrný z Obr. 4.3. V pojetí moderních "trendů" bychom tuto cílenou mutaci v rámci populace mohli označit termínem nano optimalizace. Vzniklá optimalizační metoda GAHC dle testů schopnosti efektivně nalézt extrém převyšuje původní doménu GA i HC12. GAHC byla diskutována například v práci [A44].

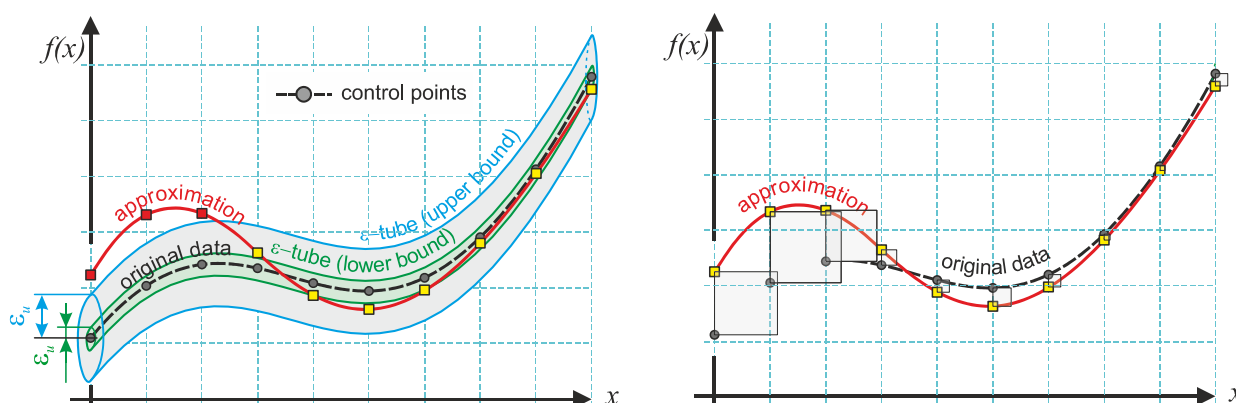


Obr. 4.3: Princip GAHC algoritmu: HC12 optimalizace je provedena pouze na části binární reprezentace jedince. Účelová funkce vyhodnocující efektivitu řešení je realizována nad celým jedincem. Volba jedince je stochastická ve volitelném rozsahu, např. dle zvyklostí pro mutační operátory. Jak je zřejmé, pozice kernelu i jeho velikost mohou být voleny deterministicky i stochasticky. Velikost kernelu, tedy délka vektoru \mathbf{a}_{ker} , je běžně 4 – 10 bitů. Možnost volby pozice a velikosti kernelu je výhodou. Tímto způsobem lze v případě potřeby cíleně ovlivňovat danou reálnou proměnnou (fenotyp). Princip GAHC lze pochopitelně aplikovat i s jiným binárně orientovaným optimalizačním algoritmem.

4.3 STE: KRITÉRIUM V ÚLOZE SYMBOLICKÉ REGRESE

Jednou z nejnovějších metod spadajících do evolučních výpočetních technik je **gramatická evoluce** (GE, *grammatical evolution*) (Ryan a kol., 1998) [RyaONei98], (O’Neill a Ryan, 2001) [ONeiRya03, ONeill08]. Jde v podstatě o nástroj pro automatické generování počítačových programů v jazycích popsaných *Backus-Naurovou formou* (BNF). Evoluční mechanismus metody může být založen na standardním genetickém algoritmu (GA, GAHC, GAFIS), což je podstatnou výhodou. Jednou z častých aplikací GE je řešení problému tzv. **symbolické regrese**, což je úloha, při které hledáme nejen vlastní parametry modelu (jak je běžné u mnohých jiných optimalizačních metod), ale i samotnou strukturu modelu. Jako běžné hodnotící kritérium je v těchto úlohách používáno kritérium SSE (*Sum Square Error*, součet kvadratických chyb). V této podkapitole však představíme originální kritérium označené jako **STE** (*Sum ε -Tube Error*).

Pro další popis budeme uvažovat úlohu numerické aproximace dat, tedy nebudeme uvažovat vliv náhodné veličiny, jak by činila statistika v úloze hledání regresního modelu. V procesu ohodnocení kvality řešení, resp. návrhu aproximačního modelu, je v absolutní většině případů využito jako hodnotící kritérium metoda nejmenších čtverců.



Obr. 4.4: Princip STE a jeho dynamické parametrizace (vlevo) a klasické statické SSE kritérium (vpravo). Hodnota STE se pro daný aproximační model liší podle velikosti ε . Hodnota parametru ε se na začátku procesu optimalizace nastavuje na svoji horní mez a postupně se snižuje s rostoucí kvalitou (mírou shody) dle adaptačního kritéria. Tento dynamický způsob je pro GE symbolickou regresi vhodnou volbou.

V případě GE a úlohy symbolické regrese se však ukazuje, že SSE minimalizační kritérium nemusí být pro konvergenci algoritmu vhodné. Z tohoto důvodu bylo navrženo nové kritérium ohodnocení označené jako STE. Jde o kritérium, které stanovuje míru podobnosti dat s navrženou aproximací. Toto kritérium si lze pak představit jako trubici o poloměru ε , která se táhne přes všechny kontrolní body, které tvoří její osu. Pokud navržená aproximace "skončí" uvnitř této trubice, je označena jako podobná a tedy v rámci daného ε jako vhodná. Hovoříme o míře shody a "náležitost" aproximace trubici v daném kontrolním bodě hodnotíme jako 1 a naopak 0, viz (4.3).

$$STE_{\varepsilon} = \sum_{i=1}^c e_{\varepsilon}(\hat{y}_i), \quad e_{\varepsilon}(\hat{y}_i) = \begin{cases} 0 & \hat{y}_i \notin [y - \varepsilon, y + \varepsilon] \\ 1 & \hat{y}_i \in [y - \varepsilon, y + \varepsilon] \end{cases} \quad (4.3)$$

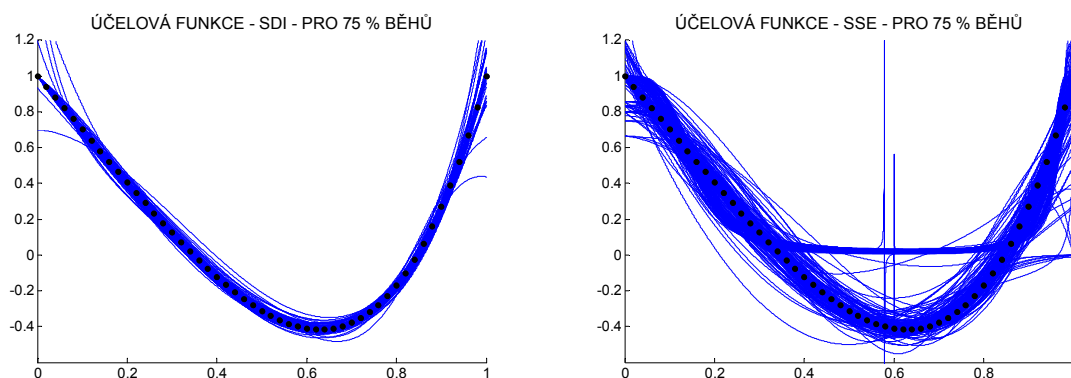
Výsledek STE_{ε} kritéria (4.3) reprezentuje počet shod s epsilonovou trubicí při daném ε . Další vlastností STE kritéria je to, že se jeho hodnota v průběhu optimalizačního procesu jednosměrně mění dle splnění podmínek adaptace hodnoty ε . Metod zavedení adaptační změny může být navržena celá řada. Poměrně jednoduché je například následující řešení.

Nechť číslo n_{adapt} udává procento kontrolních bodů, při kterém prohlásíme, že byla dosažena požadovaná shoda (nemusí být nutně 100%), dále nechť je ve významu indexů stanovena horní a spodní hranice hodnot ε_{max} a ε_{min} tak, aby odpovídala řešenému problému. Pak můžeme zavést podmínku s lineární adaptací parametru ε pro zvolený parametr kroku ε_{Δ} jako podmínku (4.4), nebo zavést podmínku s úrokovou mírou adaptace ε pro zvolený procentuální parametr $\varepsilon_{\%}$ jako podmínku (4.5). Pochopitelně je nutné dodržet logická hlediska a omezení typu $\varepsilon_{\Delta} \ll \varepsilon_{max} - \varepsilon_{min}$, $\varepsilon_{\%} < 100\%$ apod.

$$\varepsilon_{\Delta} : \quad \frac{n_{KB}}{STE_{\varepsilon}} \cdot 100 \geq n_{adapt} \Rightarrow \varepsilon = \begin{cases} \varepsilon - \varepsilon_{\Delta}, & \varepsilon - \varepsilon_{\Delta} \geq \varepsilon_{min} \\ \varepsilon & jinak \end{cases} \quad (4.4)$$

$$\varepsilon_{\%} : \quad \frac{n_{KB}}{STE_{\varepsilon}} \cdot 100 \geq n_{adapt} \Rightarrow \varepsilon = \begin{cases} \frac{1}{100} \varepsilon \cdot \varepsilon_{\%}, & \frac{1}{100} \varepsilon \cdot \varepsilon_{\%} \geq \varepsilon_{min} \\ \varepsilon & jinak \end{cases} \quad (4.5)$$

Podmínka ukončení adaptace STE může být volena vzhledem k dosažení limitní hodnoty ε , obecně také času či počtu generací, ale může být též dána počtem povolených adaptačních změn.



Obr. 4.5: Výsledek symbolické regrese realizované pomocí gramatické evoluce (GE) na totožné množině kontrolních bodů. Nalezená řešení při použití STE kritéria s úrokovým modelem adaptace ε (vlevo) a řešení dosažená pomocí SSE kritéria (vpravo). Funkce generující kontrolní body ($n_{KN} = 50$) $y = 3x^4 - 3x + 1$. Výsledky odpovídají 75 nalezeným nejlepším průběhům ze 100 realizací.

5 APLIKACE METOD POČÍTAČOVÉ INTELIGENCE

Objektivní zhodnocení, zda jsou prezentované metody počítačové inteligence více či méně efektivním optimalizačním nástrojem, není obecně možné. Subjektivní názor autora by zde nebyl korektní, protože bez jeho nadšení by nevznikl tento text ani prezentované metody. V rámci jisté objektivizace je možné konstatovat, že na prezentované třídě úloh prokázaly algoritmy HC12 a GAHC dobré vlastnosti, tj. našly efektivně (čas a přesnost) řešení problému a vedly si dobře až výborně i v porovnání s tradičními exaktními postupy.

Při návrhu nových, ale i testování výkonu stávajících optimalizačních algoritmů, je pochopitelně vhodné mít na reprezentativní množině úloh možnost srovnání s jinými algoritmy. V oblasti implementace optimalizačních metaheuristik je tato potřeba ještě intenzivnější z důvodu často silné parametrizovatelnosti daných metod (GA, DE, GE aj.).

5.1 TURNAJ ŘEŠIČŮ HC12 A GAHC VS. GAMS V ÚLOZE F7

Je velmi nepravděpodobné, že by mohl existovat "král" mezi optimalizačními algoritmy. Vždy je třeba pohlížet komplexně na optimalizační problém a pro něj zvolit vhodnou metodu řešení. Testování výkonu a vlastností algoritmů HC12 a GA/GAHC na různých úlohách věnoval autor více prostoru, jak mimo publikačního, tak s publikačními výstupy, např. [A46, A56, A69, A86]. Příklad výsledků pro úlohu globální optimalizace je uveden v následující tabulce, která je převzata z autorova článku [A78], viz Příloha B. V této tabulce jsou srovnány výsledky dosažené pomocí standardních řešičů systému GAMS. Vzhledem k multimodalitě úloh je pro všechny typy řešičů prováděno vždy 1000 spuštění, přičemž první iterace je volena náhodně.

Tab. 5.1: Výsledky různých metod optimalizace (řešičů) pro úlohu F7 vzhledem k počtu proměnných

Dimenze úlohy F7 (Schwefel)	2	3	4	5
HC12 (16bit na parametr) , % úspěšnosti	100%	100%	100%	100%
počet nalezených optim / počet pokusů	1000/1000	1000/1000	1000/1000	1000/1000
optimální hodnota účelové funkce	-837,9658	-1256,9487	-1675,9316	-2094,9145
medián hodnot účelových funkcí	-837,9658	-1256,9487	-1675,9316	-2094,9145
průměrný čas řešení (1CPU)	0,04 s	0,13 s	0,048 s	1,09 s
CONOPT^a , % úspěšnosti	15%	15%	15%	15%
počet optim / počet pokusů	150/1000	150/1000	150/1000	150/1000
CONOPT^b , % úspěšnosti	3,5%	0,7%	0,1%	0,1%
počet optim / počet pokusů	35/1000	7/1000	1/1000	1/1000
COINOPT^a , % úspěšnosti	19%	19%	19%	19%
počet optim / počet pokusů	190/1000	190/1000	190/1000	190/1000
medián hodnot účelových funkcí	-403,6864	-605,5297	-807,3729	-1009,2161
COINOPT^b , % úspěšnosti	3%	0,5%	0,3%	0%
počet optim / počet pokusů	30/1000	5/1000	3/1000	0/1000

^a První iterace je volena náhodně, ale shodně (generováno jedno číslo), pro všechny komponenty hledaného vektoru x .

^b První iterace je volena náhodně pro všechny komponenty hledaného vektoru x .

5.2 QAP: KVADRATICKÝ PŘÍRAZOVACÍ PROBLÉM A HC12 SWAP2ABS

Kvadratický přiřazovací problém, dále **QAP** (Quadratic Assignment Problem), patří v současnosti k jednomu z nejzajímavějších a z pohledu autora nejvyzývajících problémů kombinatorické optimalizace. Zajímavost QAP je dána více hledisky, z nichž to praktické představují mnohé reálné problémy, které QAP může reprezentovat.

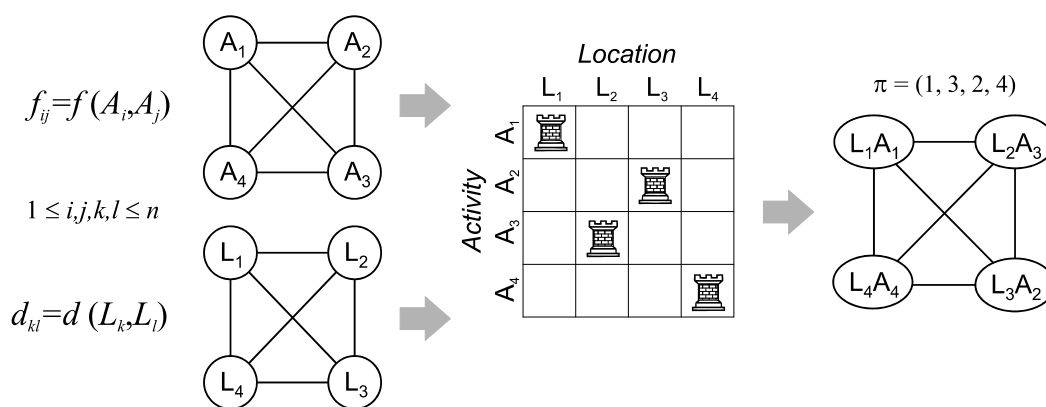
Problém QAP byl poprvé formulován v roce 1957 Koopmansem a Beckmanem [KoopBeck57]. Od první formulace problému po současnost jde o velmi intenzivně studovaný problém, který je právě dnes s možnostmi HPC výpočtů o to aktuálnější.

Formálně tedy předpokládáme, že máme množinu n lokací L a množinu n aktivit A . Zavedeme dvě matice $n \times n$: matici $\mathbf{D}_L = (d_{ij})$ (*distance matrix*) udávající vzdálenosti lokací i a j a matici $\mathbf{F}_A = (f_{kl})$ (*flow matrix*), která charakterizuje ceny toků mezi aktivitami k a l přes jednotku vzdálenosti. Bijekce množiny aktivit na množinu lokací implicitně definuje součiny $d_{ij}f_{\pi(i)\pi(j)}$. Podstatou úlohy QAP je najít optimální permutaci π_{opt} , která minimalizuje účelovou funkci (5.1) přes všechny permutace $\pi \in S_n$, kde S_n je množina permutací množiny $\{1, 2, \dots, n\}$ a $k = \pi(i)$, $l = \pi(j)$, tedy řešení (5.2).

$$C(\pi) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{\pi(i)\pi(j)} \quad (5.1)$$

$$\pi_{opt} = \arg \min_{\pi \in S_n} C(\pi) \quad (5.2)$$

Tento problém velmi elegantně reprezentuje matice \mathbf{X} , kterou si můžeme představit jako šachovnici, kde jedničky v matici \mathbf{X} interpretujeme jako šachové věže. Přípustným řešením bude libovolné rozmístění věží takové, aby se tyto vzájemně neohrožovaly. Pokud například sloupce matice \mathbf{X} interpretujeme jako vrcholy grafu lokací L a řádky jako vrcholy grafu aktivit A , je zřejmé, že princip věží generuje permutace, čímž splňuje implicitní podmínky v (5.1). Stručně, z této interpretace matice \mathbf{X} logicky plyne přípustné řešení problému QAP.



Obr. 5.1: Interpretace problému QAP, princip věží a operace swap2. V permutační matici \mathbf{X} (rovněž incidenční matice pro lokace-aktivity) si lze jedničky představit jako šachové figury věží. Princip věží pak znamená takové rozmístění (konfiguraci) věží na "šachovnici" \mathbf{X} , aby se tyto vzájemně neohrožovaly. Bijekce vrcholů grafu lokací L a vrcholů grafu aktivit A je takto zajištěna. Tímto způsobem jsou rovněž implicitně určeny bijekce množin $\{d_{ij}\}$ a $\{f_{kl}\}$, tedy součiny d_{ij} a f_{kl} . Pokud na této šachovnici přehodíme libovolné dva sloupce, resp. řádky, obdržíme novou konfiguraci matice \mathbf{X} , tedy novou permutaci a princip věží zůstane zachován. Toto přehození označíme jako operaci swap2.

Princip optimalizace (5.2) je tedy interpretován jako hledání optimální konfigurace "matice věží", tedy konfigurace permutační matice \mathbf{X} . Z podstaty problému i grafové interpretace QAP je dále zřejmé, že přehazováním libovolných sloupců matice obdržíme vždy přípustné řešení. Stejný efekt bude mít přehazování řádků. Obě operace jsou ekvivalentní a vždy povedou k přípustnému řešení. Dále zvolíme libovolně, ale pevně, jednu z uvedených variant přehazování, např. budeme přehazovat řádky (dle Obr. 5.1 umístíme A na L). Protože přehazujeme vždy dvojice, volíme pro tuto operaci označení **swap2**.

Přesto, že přípustných permutací věží a tedy řešení problému QAP je $n!$, je zřejmé, že se teoreticky dostaneme z libovolné konfigurace \mathbf{X}_1 matice \mathbf{X} do optimální konfigurace \mathbf{X}_{opt} po maximálně $(n - 1)$ krocích, tj. po $(n - 1)$ aplikacích operace swap2 . To vše za předpokladu, že tyto kroky, tedy postupné aplikace operace swap2 , budou optimální.

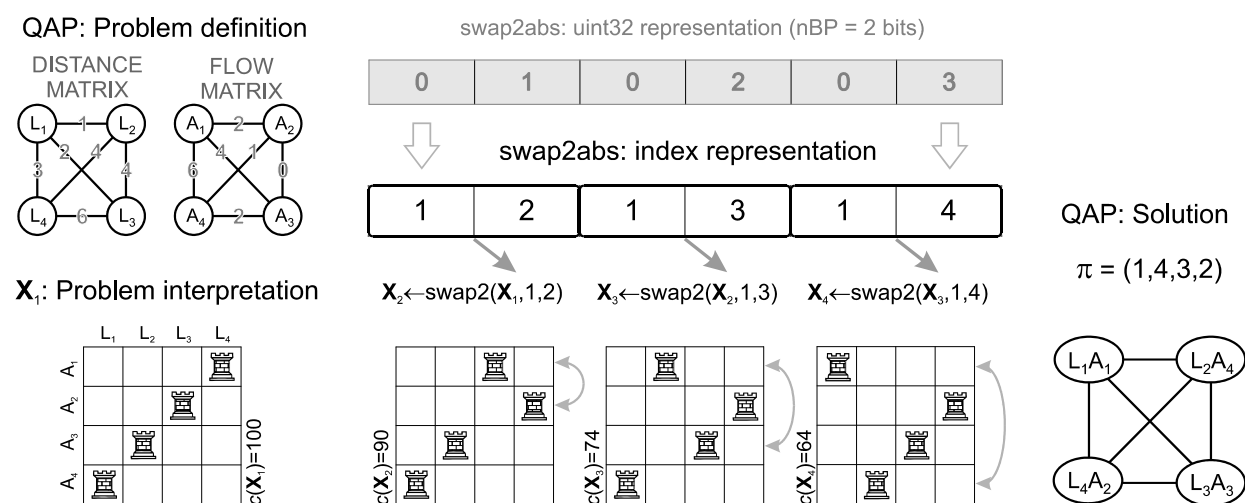
Protože jde o realizaci posloupnosti kroků z prvotní konfigurace \mathbf{X}_1 , která je pevně zadaná a vůči konečnému stavu \mathbf{X}_{opt} nebo \mathbf{X}_{opt} neměnná, je pro tuto variantu posloupnosti kroků zvoleno označení **swap2abs** (5.3).

$$\mathbf{X}_{\text{opt}} = \text{swap2}_k \circ \dots \circ \text{swap2}_1(\mathbf{X}_1) = \text{swap2}_k(\dots(\text{swap2}_1(\mathbf{X}_1))\dots) \quad (5.3)$$

Binární řetězec kódující řešení problému tedy není explicitně permutací, ale je "návodem", jak výslednou permutaci získat.

Řešení kombinatorických úloh vyžaduje při implementaci evolučních algoritmů (EA) speciální přístupy plynoucí z permutační reprezentace problému. Systém reprezentace problému QAP pomocí **principu věží** spolu se specifickou reprezentací řešení **swap2abs**, tj. posloupnosti skládání operací **swap2**, vytváří prostor pro snadnou implementaci všech EA a obecně i dalších metod matematického programování.

Parametr udávající hodnotu počtu kroků k jako jediný parametrizuje kódování swap2abs a jak je zřejmé, má významný vliv na schopnost algoritmu dosáhnout optimální řešení dané úlohy.



Obr. 5.2: Princip kódování swap2abs a příklad řešení modelové úlohy QAP o rozsahu $n = 4$. V tomto případě je využita teoreticky maximální hodnota počtu kroků $k = 3$. Jak je z obrázku patrné, této hodnotě odpovídají tři dvojice indexů pro operace swap2 . Počáteční konfigurace matice \mathbf{X}_1 byla volena náhodně.

Jak je zřejmé, uvedeným způsobem kódování swap2abs lze pro daný problém bezproblémově implementovat standardní varianty EA. Standardní operátory křížení i mutace ovlivňují pouze volbu sloupců matice \mathbf{X} a nemohou tak způsobit ztrátu (rozbití) permutace. Toto je podstatný rozdíl od všech autorovi známých přístupů permutačního kódování, typicky kódování problémů typu TSP v GA.

Pro prezentaci efektivity algoritmů HC12 a GAHC spolu s kódováním swap2abs byla zvolena třída problémů označená jako **esc**. Úlohy třídy **esc** formulovali v roce 1990 Eschermann a Wunderlich [EscWun90]. Dle dostupných zdrojů lze konstatovat, že kompletní řešení úloh třídy **esc** odolávalo exaktním postupům i metaheuristikám do roku 2011 včetně.

Výsledky pro implementace řešičů HC12 (vč. verze taboo) s optimální parametrizací k získané z poměrně časově rozsáhlých testů udává Tab. 5.2, ve které jsou navíc prezentována dosud nejlepší řešení. Optimální parametrizace GAHC nebyla vzhledem k robustnosti algoritmu nutná.

Tab. 5.2: Ilustrativní porovnání dosažených časů řešení instancí QAP třídy *esc* pomocí nejlepších exaktních řešičů (užita vhodná parametrizace řešiče i formulace úlohy) a vybraných neúspěšnějších metaheuristik.

instance	OPT	exaktní řešení (2011)♦ IBM ILOG	exaktní řešení (2011)♠ MIPG Gurobi	optTime♣ (2012) HC12/GAHC	IFLSa (2008)	NIFLSb (2009)	GAc (2006)	ITSd (2012)
esc16a	68	0,35 s	11 s	0,006 s	3,17 s	2,172 s	N/A	N/A
esc32a	130	51 s	1 618 580 s	245 s	137 s	≠OPT 89,531 s	37 s	4 s
esc32b	168	N/A	N/A	142 s	110 s	73 s	24 s	1,9 s
esc32c	642	1 156 s (9 644s)*	24 364 s	0,05 s	55 s	35 s	20 s	1,8 s
esc32d	200	472 s (2 973s)*	36 256 s	2,29 s	74 s	49s	15 s	2,2 s
esc32e	2	0,04 s	N/A	0,001 s	46 s	29 s	23 s	1,3 s
esc32g	6	0,06 s	N/A	0,001 s	28 s	19 s	33 s	1,1 s
esc32h	438	7 800 s	N/A	3,50 s	86 s	55 s	26 s	2,0 s
esc64a	116	84 s (510s)*	16 370 s	1,43 s	1 522 s	1 060 s	253 s	11,5 s
esc128	64	9,63 s	N/A	1 342 s	N/A	23 370 s	N/A	70 s

♦ čas je poplatný implementaci řešiče a užitě výpočetní platformě: quadcore Intel Xeon 3.2GHZ, 16GB (Mac OS),

♠ čas je poplatný implementaci řešiče a užitě výpočetní platformě: quadcore Intel i7 2.8GHz, 6GB,

♣ čas je poplatný implementaci řešiče a užitě výpočetní platformě: quadcore Intel Xeon 2.66GHx, 8GB, nVidia Tesla C1060 (nvmex),

* výsledek se standardně parametrizovaným nastavením řešiče,

a IFLS: Iterated Fast Local Search with OXSM recombination operator, Ramkumar at al. (2008),

b NIFLS: Iterated Fast Local Search with new recombination operator, Ramkumar at al. (2009), [RaPoJa09],

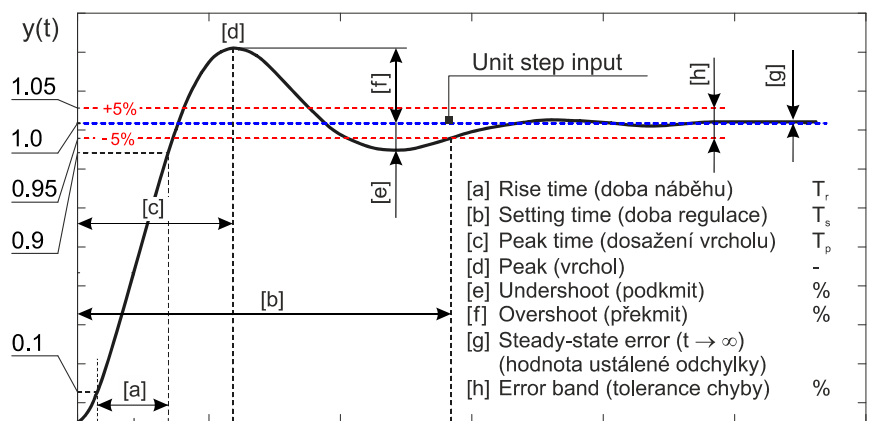
c GA with intra-kindred selection, Ji et al. (2006), [JiWuLiu06],

d ITS: Iterated Taboo Search, , optimalizované parametry řešiče ITS, Misevicius (2012), [Misevicius12].

5.3 NÁVRH OBECNÉHO LINEÁRNÍHO REGULÁTORU

Při návrhu regulátorů může být uplatněno více přístupů založených na metodách algebraických i frekvenčních, které jsou poté interpretovány exaktně, numericky nebo empiricky. Prakticky by se při návrhu regulátoru mělo vždy vycházet z provozních hledisek, tedy podmínek, ve kterých bude regulační obvod pracovat. K těmto hlediskům patří vlastnosti regulované soustavy, průběhy řídicí veličiny, předpokládané poruchové veličiny, limity akčních veličin a specifické požadavky kladené na kvalitu regulace.

V práci byla pro potřebu prezentace návrhu regulátoru použita kritéria kvality regulace vyplývající z přechodové charakteristiky, viz Obr. 5.3.



Obr. 5.3: Běžně užívaná hlediska kvality regulace [a] – [g] získaná z přechodové charakteristiky (odezva systému na jednotkový skok – unit step response). Toleranční pásmo [h] je běžně 0÷5% žádané hodnoty.

Současný specifický výzkum v oblasti strukturálního návrhu regulátorů zje aměřen na využití dnes již standardních přístupů genetického programování (GP) a gramatické evoluce (GE). V kontextu GA není reprezentace jedince v GP bloková (všichni jedinci mají stejnou délku).

V práci [A81] byl představen původní a originální návrh regulátoru využívající přístup optimálního rozmístování kořenů. Cílem nebyl návrh zcela obecné struktury regulátoru, ale návrh "omezenější" struktury regulátoru ve formě racionální lomené funkce, tedy návrh **obecného lineárního regulátoru**. Tento regulátor je vzhledem ke své definici dále "pracovně" označovaný jako **polynomiální regulátor**.

Pro návrh regulátorů byla vyvinuta řada metod. Uvažujeme-li lineární regulátory, kam řadíme například PID regulátor, pak se dá navržený regulátor popsat operátorovým přenosem.

Nejčastější reprezentací je vyjádření přenosu ve formě podílu dvou polynomů (5.4), odtud zvolený alternativní název polynomiální regulátor. Další forma zápisu (5.4) může být uvedena s vytknutím koeficientů b_m a a_n u nejvyšších mocnin (5.5), kde $K = b_m/a_n$.

$$G_R(s) = \frac{b_m s^m + \dots + b_2 s^2 + b_1 s + b_0}{a_n s^n + \dots + a_2 s^2 + a_1 s + a_0}, \quad m \leq n \quad (5.4)$$

$$G_R(s) = K \frac{s^m + \dots + \frac{b_2}{b_m} s^2 + \frac{b_1}{b_m} s + \frac{b_0}{b_m}}{s^n + \dots + \frac{a_2}{a_n} s^2 + \frac{a_1}{a_n} s + \frac{a_0}{a_n}}, \quad m \leq n \quad (5.5)$$

Pokud bychom se zvolenou optimalizační metodou uskutečnili pro zvolené stupně m, n polynomů čitatele a jmenovatele hledání optimálních parametrů regulátoru, tedy hledání koeficientů polynomů a_i a b_j , zjistili bychom, že zvolená metoda je málo efektivní, neboť při generování (koeficientů) polynomů vzniká velké množství nestabilních řešení.

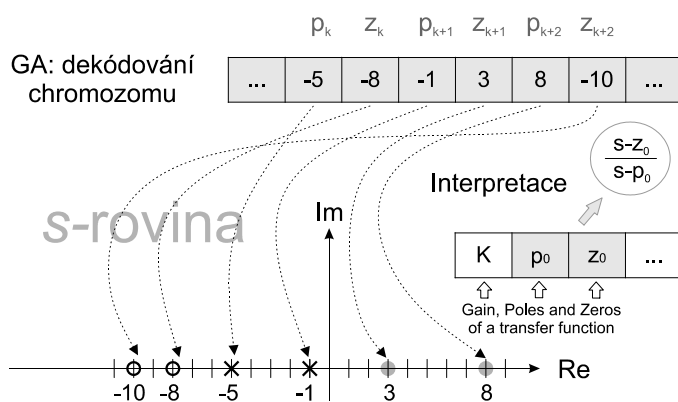
Z uvedeného důvodu budeme optimalizovat nikoli koeficienty polynomů, nýbrž kořeny z_i a p_i polynomů (tvořících přenos) dle (5.6). Zde již snadno zajistíme stabilitu například tím, že budeme hledat pouze kořeny se zápornou reálnou částí.

$$G_R(s) = K \frac{(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)}, \quad m \leq n \quad (5.6)$$

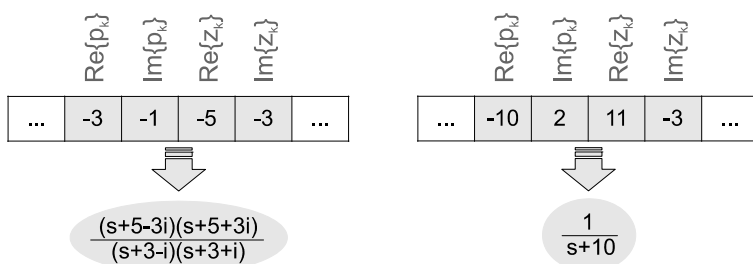
Podstatou diskutovaného návrhu polynomiálního regulátoru, tedy návrhu struktury (modelu) a současně parametrů regulátoru, je **manipulace s kořeny polynomů** tak, aby výsledný návrh apriorně splňoval požadavek stability a realizovatelnosti a současně byla minimalizována hodnota zvoleného kritéria kvality regulace. Pojem manipulace s kořeny znamená ovlivnění jejich existence a lokace v komplexní rovině.

Veškeré naše představy se proto mohou odvolávat na grafickou reprezentaci regulátoru pomocí pólů a nul. Pro uvedený návrh polynomiálního regulátoru je nejdříve nutné stanovit jeho maximální stupeň, který je dán hodnotou n (stupeň polynomu jmenovatele). Dále je v rámci reprezentace problému kódováno zesílení regulátoru a v alternující posloupnosti kořeny čitatele (nuly) a jmenovatele (póly).

Princip kódování/dekódování parametrů polynomiálního regulátoru a následná **pravidla pro tvorbu jeho struktury** jsou základem *know-how* uvedené koncepce návrhu. Následující Obr. 5.4 a Obr. 5.5 demonstrují systém interpretace/dekódování optimalizovaných parametrů regulátoru.



Obr. 5.4: Příklad interpretace a dekódování jedince GA pro "pole-placement" návrh polynomiálního regulátoru typu GAR. Na začátku řetězce se nachází gen reprezentující zesílení regulátoru, následují geny v nichž je zakódována poloha pólů a nul. Vzhledem k interpretaci kořenů pro výsledný přenos regulátoru není pořadí kořenů v rámci kódování důležité. Parametr zesílení K má ovšem zásadní význam pro celý přenos regulátoru. Kořeny spadající do kladné poloroviny nejsou interpretovány a dle definovaných pravidel snižují řád regulátoru. V obrázku je uveden příklad zániku kořenů z_{k+1} a p_{k+2} .



Obr. 5.5: Příklad interpretace části jedinců pro variantu GARnC. Interpretovány jsou komplexně sdružené kořeny čitatele a jmenovatele (vlevo), resp. pouze reálný kořen, imaginární složka je eliminována (vpravo).

Princip úpravy zesílení je druhým základním atributem prezentovaného strukturálního návrhu regulátoru. Tento princip eliminuje vliv dílčích zesílení generovaných kořeny systému tak, aby optimalizované zesílení nebylo nijak ovlivňováno. Jinými slovy, pokud chceme docílit, aby výsledné zesílení regulátoru nebylo závislé na změně počtu svých kořenů, musíme zařídit, aby případný vznik nového kořenového činitele polynomu, respektive zánik kořenového činitele, neovlivnil podíl absolutních členů polynomů b_0/a_0

$$\begin{aligned}
G_R(s) &= K \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)} \dots \\
&= K \frac{\prod_{j=1}^m z_j \left(\frac{1}{z_1}s-1\right)\left(\frac{1}{z_2}s-1\right)\dots\left(\frac{1}{z_m}s-1\right)}{\prod_{i=1}^n p_i \left(\frac{1}{p_1}s-1\right)\left(\frac{1}{p_2}s-1\right)\dots\left(\frac{1}{p_n}s-1\right)} \dots \\
&= k \frac{(\tau_1s-1)(\tau_2s-1)\dots(\tau_ms-1)}{(T_1s-1)(T_2s-1)\dots(T_ns-1)}
\end{aligned} \tag{5.7}$$

$$k = K \cdot K_{zp} = K \frac{\prod_{j=1}^m z_j}{\prod_{i=1}^n p_i} \tag{5.8}$$

Část zesílení regulátoru označená jako K_{zp} je v případě přenosu vyjádřeného ve formě pólů a nul vázána na kořenové činitele. Pokud tuto část zesílení eliminujeme, bude výsledné zesílení přenosu regulátoru sice jiné, ale invariantní vůči změnám kořenových činitelů, což je žádoucí. Navíc je parametr zesílení optimalizovaným parametrem při návrhu regulátoru a tedy je uvedená úprava bezkonfliktní. Po této úpravě je optimalizačním algoritmem možno hledat nezávislé optimální zesílení polynomiálního regulátoru K a optimální polohy kořenů z_i a p_i pro definovaný maximální stupeň polynomu čitatele.

6 ZÁVĚR

Předložená habilitační práce shrnula a ukázala vybranou část autorovy dosavadní vědecko-výzkumné činnosti, převážně zaměřené na tvorbu a aplikaci vybraných metod počítačové inteligence. Z pohledu autora je nejcennější návrh optimalizační metaheuristiky HC12 a hybridního algoritmu GAHC. Nesporným kladem algoritmu HC12 je jeho velmi dobrá škálovatelnost, přičemž tato vlastnost může být v budoucnu v kontextu nových HPC ještě výraznější. Algoritmy HC12 a GAHC (vznikly efektivním spojením HC12 a GA) prokázaly schopnost na dané třídě úloh více než úspěšně konkurovat exaktním řešičům v systému GAMS, ale i jiným v kontextu světové tvorby. Úspěchy algoritmů při řešení úloh DeJongova typu lze hodnotit pozitivně, nikoliv však excelentně (mnoho dalších evolučních algoritmů je schopno dosahovat obdobných výsledků, byť v horších časech vzhledem ke GPU implementaci). Poslední výsledky v rámci řešených úloh QAP jsou však srovnatelné (a v některých instancích časem řešení i lepší) s tím nejlepším, co lze v současnosti dohledat. Rovněž systém návrhu polynomiálního regulátoru lze považovat ve více ohledech za jedinečný a hodný další pozornosti jak s teoretickou, tak s praktickou perspektivou do budoucna. Tento projekt spolu s návrhem optimalizačních metaheuristik a jejich implementací byl zahrnut jako součást již úspěšně oponovaného VVZ Inteligentní systémy v automatizaci (prof. Jura, FEKT VUT v Brně).

Pro propagaci algoritmů HC12, GAHC a jejich paralelních implementací slouží vytvořená webová stránka www.HPCbook.info, kde se autor bude snažit uvedené algoritmy šířit a podporovat mimo svoji alma mater. Rovněž mezinárodní konference MENDEL (www.mendel-conference.org), zabývající se oblastí Soft Computing, může dobře sloužit tomuto účelu. Autor této práce aktivně spolupracuje s mnoha vědci světových parametrů (např.: prof. Lampinen – Finsko, Dr. Langdon – UK, Dr. Xin She – UK, prof. Mendel – USA, prof. Klín – Izrael, Dr. O'Neill – Irsko, Dr. Ryan – Irsko, prof. Zaharie - Rumunsko), kteří o uvedené metody a přístupy jeví zájem, či je využívají. V oblasti kombinatorické optimalizace autor pro roky 2013-2014 vypsal společný mezinárodní projekt s izraelskou stranou, dále je autor zapojen jako zástupce VUT do mezinárodního projektového týmu pro rok 2013-2015 vedeného prof. Lampinenem se zaměřením na aplikace MILP.

Mnoho myšlenek, na kterých v současnosti autor pracuje, se z časově prostorových důvodů do práce nevešly. V kontextu práce naznačená aplikační témata typu: globální optimalizace, smíšené celočíselné programování, aproximace s vazební podmínkou nezápornosti, optimalizace regulátoru a další aplikace, kde lze rozvíjet zajímavé metody počítačové inteligence, lze pochopitelně dále rozvíjet.

V současnosti autor pracuje na aplikacích v oblasti evoluční optimalizace regulátorů v podmínkách nelinearit. Druhým směrem současného zájmu je aplikace metod počítačové inteligence na kombinatorické problémy třídy QAP. V tomto ohledu autor doufá v přijetí vypsaných projektů. Autor si je rovněž vědom, že obzvláště v prostředí technické university by měly být myšlenky transformovatelné do využitelné praxe. Stran evolučně navrhovaných regulátorů budou praktické experimenty na platformě B&R automation následovat. Původní optimalizační algoritmy HC12 a GAHC jsou již nyní využitelné jako optimalizační nástroje v prostředí MATLAB a na jejich dalším vývoji se pracuje.

Otevřená učebnice přírody nabízí nespočet inspirací pro všechny, kdo mají dar v ní číst, naslouchat a pochopit. Metody počítačové inteligence (CI) v této knize jistě stále nechávají mnoho nedotčených stránek. V současné době není snadné učinit průlomový vědecký objev patrně v žádné oblasti vědy. Srdce odvážného se však tímto nesmí nechat odradit, je třeba hledat nové cesty a nacházet nové světy, je to skvělé dobrodružství, které nekončí...

REFERENCE

- [Adámek89] ADÁMEK, J. Kódování. SNTL. Praha, 1989.
- [Anděl93] ANDĚL, J. Statistické metody. MatFyzPress, Praha, 1993.
- [Appleg06] APPLGATE, L. D.; BIXBY E. R.; CHVÁTAL V.; COOK J. W. The Travelling Salesman Problem: A Computational Study. Princeton University Press, Princeton, New Jersey, 2006, ISBN 978-0-691-129938.
- [AstromHagg06] ASTROM, K.J.; HAGGLUND, T. Advanced PID Control, ISA (The International Society of Automation), 2006, ISBN 978-1556179426.
- [Baker95] BAKER, J. E. Selection Methods for Evolutionary Algorithms. In Chambers, L. (ed.): Practical Handbook of Genetic Algorithms: New Frontiers. CRC Press, New York, 1995, pp. 67-92.
- [Bartsekas99] BERTSEKAS, P. D. Nonlinear Programming. Athena Scientific, Belmont, Massachusetts, 1999, ISBN 1-886529-00-0.
- [Bäck96] BÄCK, T. Evolutionary Algorithms in Theory and Practice. Oxford University Press, Oxford, 1996.
- [Blickle95] BLICKLE, T., THIELE, L. A Comparison of Selection Schemes used in Genetic Algorithms. TIK-Report ver.2, Zurich, 1995.
- [Bovet94] BOVET, D.P.; CRESCENZI, P. Introduction to the Theory of Complexity. Englewood Cliffs, N.J., Prentice-Hall, 1994.
- [Boyd04] BOYD, S.; VANDENBERGHE, L. Convex Optimization. Cambridge University Press, 2004, ISBN 978-0-521-83378-3. Dostupné z: <http://www.stanford.edu/~boyd/cvxbook/>
- [BroLow88] BROOMHEAD, D. S.; LOWE, D. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 1988, no.2, pp.321-355.
- [BrClMaPe96] BRUENGER, A.; CLAUSEN, J.; MARZETTA, A.; PERREGAARD, M. Joining forces in solving large-scale quadratic assignment problems in parallel. DIKU Technical Report, University of Copenhagen, 1996.
- [BucTur92] BUCHANAN, J. L., TURNER, P. R. Numerical Methods and Analysis. McGraw-Hill, New York, 1992, ISBN 0-07-008717-2.
- [BuKaRe91] BURKARD, R.E., KARISCH, S., RENDL, F. QAPLIB – A quadratic assignment problem library. *European Journal of Operational Research*, Volume 55, Issue 1, 6 November 1991, Pages 115–119.
- [Bui08] BUI, L. T.; ALAM, S. Multi-Objective Optimization in Computational Intelligence: Theory and Practice. International Science Reference, USA, 2008, ISBN 13-978-1-59904-498-9.
- [Čela10] ČELA, E. The quadratic Assignment Problem – Theory and Algorithms. Volume 1 of *Combinatoric Optimization*, Kluwer Academic Publishers, Netherlands, 2010. ISBN 978-1-4419-4786-4.
- [ClaPer94] CLAUSEN, J.; PERREGAARD, M. Solving large quadratic assignment problems in parallel. *Computational Optimization and Applications*, 8(2):11-127, 1997.
- [Cook71] COOK, S.A. The Complexity of Theorem-Proving Procedures, In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, 1971, pp.151-158.
- [Coello07] COELLO, C. A. C.; LAMONT, G. B.; VAN VELDHUIZEN, A. A. Evolutionary Algorithms for Solving Multi-Objective Problems. In *Genetic and Evolutionary Computation*. 2nd edition, Springer, USA, 2007, ISBN 978-0-387-33254-3.
- [Coello04] COELLO, C. A. C. List of references on evolutionary multiobjective optimization. [online] [cit. 2012-3-1]. Dostupné z: <http://www.lania.mx/~ccoello/EMOO/EMOObib.html>
- [Davis89] DAVIS, L. Adapting operator probabilities in genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications* (Grefenstette, J. J., ed.). Lawrence Erlbaum, 1989.

- [Deb02] DEB, K.; AGRAWAL, S. A Fast and Ellitist Multi Objective Genetic Algorithm: NSGAI. *IEEE Transactions on Evolutionary Computation*, 2002, vol. 6, no. 2, pp. 182-198.
- [DeJong75] DEJONG, K. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. University of Michigan, 1975.
- [Došlá00] DOŠLÁ, Z.; DOŠLÝ, O. Metrické prostory. 2. vydání. Vydavatelství Masarykovy university v Brně, Brno, 2000, ISBN 80-210-1328-1.
- [Drezner08] DREZNER, Z. Tabu Search and Hybrid Genetic Algorithms for Quadratic Assignment Problems. In W. Jaziri (ed.) *Local Search Techniques: Focus on Tabu Search*, IN-TECH, V5, 89-108. Dostupné z: http://sciyo.com/articles/show/title/tabu_search_and_hybrid_genetic_algorithms_for_quadratic_assignment_problems.
- [Eliza66] WEIZENBAUM, J. ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the Association for Computing Machinery*, 1966, vol. 9, pp. 36-45.
- [Eliza76] WEIZENBAUM, J. Computer power and human reason. San Francisco, CA: W.H. Freeman, 1976.
- [Elshafei77] ELSHAFEI, A.N. Hospital Layout as a Quadratic Assignment Problem. *Operational research Quarterly*, 1977, vol. 28, No.1, part2, pp. 167-179.
- [EscWun90] ESCHERMANN, B.; WUNDERLICH, H.J. , LSHAFEI, A.N. Optimized synthesis of self-testable finite state machines. In *20th International Symposium on Fault-Tolerant Computing (FTCS 20)*. Newcastle upon Tyne, 26-28th June, 1990.
- [FabLag08] FABIJANSKY, P.; LAGODA, R. On-line PID controller tuning using genetic algorithm and DSP PC board. *Power Electronics and Motion Control Conference*, 2008. EPE-PEMC 2008. 13th , vol., no., pp.2087-2090, 1-3 Sept. 2008.
- [Fogel95] FOGEL, D. B. Evolutionary Computation, Toward a New Philosophy of Machine Inteligence. IEEE Press, 1995.
- [GaoTong06] GAO, F.; TONG, H. Q. Differential Evolution: An Efficient Method in Optimal PID Tuning and on-line Tuning. In *Proceedings of the First International Conference on Complex Systems and Applications*, 2006, Wuxi, China (SCI).
- [Garey79] GAREY, M. R.; JOHNSON, D. S. Computer and Intractability: Guide to the Theory of NP-Completeness. W.H.Freeman and Co., 1979.
- [Glover89] GLOVER, F. Tabu Search – Part I. *ORSA Journal of Operations Research*, 1998, no.1.
- [Glover90] GLOVER, F. Tabu Search – Part I. *ORSA Journal of Operations Research*, 1990, no. 1.
- [GloLag97] GLOVER, F.; LAGUNA, M. Tabu Search. Kluwer Academic Publishers, Boston, 1997.
- [Goldberg89] GOLDBERG, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing, 1989.
- [Grefenstette86] GREFENSTETTE, J. J. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1986, vol. 16, no. 1, pp. 122-128.
- [Grefenstette90] GREFENSTETTE, J. J. Genetic algorithms and their applications. In Kent, A.; Williams, J. G. (Eds.): *The Encyclopedia of Computer Science and Technology*, 21 (Supplement 6). New York: Marcel Dekker, 1990.
- [Grefenstette97a] GREFENSTETTE, J. J. Rank-based selection. In Baeck, T.; Fogel, D.; Michalewicz, Z. (Eds.): *The Handbook of Evolutionary Computation*. IOP Publishing and Oxford University Press, 1997.
- [Grefenstette97b] GREFENSTETTE, J. J. Proportional selection and sampling algorithms. In Baeck, T.; Fogel, D.; Michalewicz, Z. (Eds.): *The Handbook of Evolutionary Computation*. IOP Publishing and Oxford University Press, 1997.
- [Haupt98] HAUPT, R. L.; HAUPT, S. E. Practical Genetic Algorithms. John Wiley & Sons, New York, 1998, ISBN 0-471-18873-5.
- [Herrera97] HERRERA, F.; LOZANO, M.; VERDEGAY, L. Fuzzy Connectives Based Crossover Operators to Model Genetic Algorithms Population Diversity. *Fuzzy Sets and Systems*, 1997.

- [Himmelbl69] HIMMELBLAU, D. M. Process Analysis by Statistical Methods. John Wiley & Sons., New York, 1969.
- [Himmelbl72] HIMMELBLAU, D. M. Applied Nonlinear Programming. McGraw-Hill, New York, 1972.
- [Holland75] HOLLAND, J. H. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975.
- [Hromkovič01] HROMKOVIČ, J. Algorithmics for Hard Problems, Introduction to Combinatorial Optimization, Randomization, Approximation and Heuristics. Springer, Berlin, Germany, 2001.
- [Hugosson10] HUGOSSON, J.; HEMBERG, E.; BRABAZON, A.; O'NEILL, M. Genotype Representations in Grammatical Evolution. *Applied Soft Computing*, 2010, vol.10, pp.36-43.
- [JiWuLiu06] JI, P.; WU, Y.; LIU, H. A Solution Method for the Quadratic Assignment Problem (QAP). In *the Sixth International Symposium on Operations Research and Its Applications (ISORA'06)*, Xinjiang, China, 2006, pp. 106-117.
- [Jura03] JURA, P. Základy fuzzy logiky pro řízení a modelování. VUTIUM, Brno, 2003, 132s., ISBN 80-214-2261-0.
- [JaSuMi97] JANG, J. S. R.; SUN, C. T.; MIZUATNI, E. Neuro-Fuzzy and Soft Computing. Prentice Hall, 1997.
- [KaPoBe02] KARPÍŠEK, Z.; POPELA, P.; BEDNÁŘ, J. Statistika a pravděpodobnost. Akademické nakladatelství CERM, Brno, 2002, ISBN 80-7204-261-0.
- [Karpíšek03] KARPÍŠEK, Z.: Matematika IV, statistika a pravděpodobnost. Akademické nakladatelství CERM Brno (2. doplněné vydání), Brno, 2003, ISBN 80-214-2522-93.
- [KalWal94] KALL, P.; WALLACE, S. W. Stochastic Programming. John Wiley & Sons, Chichester, 1994.
- [Kazarlis01] KAZARLIS, S. A. Microgenetic algorithms as generalized hill climbing operators for GA optimization. *IEEE Transaction on Evol. Comp.*, 2011, no. 5, pp. 204-217.
- [Kennedy01] KENNEDY, J.; EBERHART, R. C. Swarm Intelligence. Morgan Kaufmann, San Francisco, CA, 2001.
- [Kirk10] KIRK, B. D.; HWU, W. W. Programming Massively Parallel Processors. Morgan Kaufman, USA, 2010 (NVIDIA series), ISBN 978-0-13-138768-3.
- [KIDvPo01] KLAPKA, J.; DVOŘÁK, J.; POPELA, P. Metody operačního výzkumu. Nakladatelství VUTIUM, Brno, 2001, ISBN 80-214-1839-7.
- [Klir95] KLIR, G. J.; YUAN, B. Fuzzy Sets and Fuzzy Logic. Prentice Hall, 1995.
- [Konar05] KONAR, A. Computational Intelligence (Principles, Techniques and Applications). Springer, Netherlands, 2005, ISBN 3-540-20898-4.
- [KoopBeck57] KOOPMANS, T.C; BECKMAN, M. Assignment problems and location of economic activities. *Econometric* 25, pp. 53-76, 1957.
- [Koza92] KOZA, J. R. Genetic Programming: On the programming of computers by mean of natural selection. MIT Press, New York, 1992.
- [Koza94] KOZA, J. R. Genetic Programming II, Automatic Discovery of Reusable Programs. MIT Press, New York, 1994.
- [KvBePo97] KVASNIČKA, V.; BEŇUŠKOVÁ, Ľ.; POSPÍCHAL, J.; FARKAŠ, I.; TIŇO, P.; KRÁL, A. Úvod do teórie neuronových sietí. Vydavateľstvo IRIS, Bratislava, 1997, ISBN 80-88778-30-1.
- [KvPoPe95] KVASNIČKA, V.; POSPÍCHAL, J.; PELIKÁN, M. Hill Climbing with Learning. In *Proceedings of MENDEL '95, 1st International Conference on Genetic Algorithms*. Brno, 1995, pp. 65-70.
- [KvaPos95] KVASNIČKA, J.; POSPÍCHAL, J. Evoluční algoritmy, Tabu search. Computer word 95, Praha, 1995.
- [KvaPos08] KVASNIČKA, J.; POSPÍCHAL, J. Algebra a diskrétna matematika. STU Bratislava, 2008, ISBN 978-80-227-2934-5.

- [KvPoPe97] KVASNIČKA, V.; POSPÍCHAL, J.; PELIKÁN, M. Read's Linear Codes and Evolutionary Computation Over Populations of Rooted Trees. In *Proceedings Košice 97*. Košice, 1997.
- [KvPoTi00] KVASNIČKA, V.; POSPÍCHAL, J.; Tiňo, P. Evolučné algoritmy. Vydavateľstvo STU, Bratislava, 2000, ISBN 80-227-1377-5, pp. 1-179.
- [KukLam04] KUKKONEN, S.; LAMPINEN, J. Comparison of generalized differential evolution to other multi-objective evolutionary algorithms. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*. Jyväskylä, July 2004.
- [LamZel99] LAMPINEN, J.; ZELINKA, I. New Ideas in Optimisation – Mechanical Engineering Design Optimisation by Differential Evolution. Volume 1. McGraw-Hill, London, 1999, ISBN 007-709506-5.
- [Light92] LIGHT, W. Advances in Numerical Analysis, volume II. – Wavelets, Subdivision Algorithms and Radial Basis Functions. Clarendon Press, Oxford, 1992, ISBN 0-19-853439-6.
- [LiPardalos94] LI, Y., PARDALOS, P.M. RESENDE, MGC. A greedy randomized adaptive search procedure for the quadratic assignment problem. Quadratic assignment and related problems. In: Pardalos PM, Wolkowicz H, editors. *DIMACS series in discrete mathematics and theoretical computer science*. American Mathematical Society; 1994. p. 237–61.
- [Looney97] LOONEY, C. G. Pattern Recognition Using Neural Networks. Oxford University Press, New York, Oxford, 1997.
- [Luger02] LUGER, G. F. Artificial Intelligence – Structures and Strategies for Complex Problem Solving. Addison Wesley, Fourth Edition, Harlow, 2002, ISBN 0-201-64866-0.
- [ManCha02] MANDIC, D. P.; CHAMBERS, J. A. Recurrent Neural Networks for Prediction. John Wiley, Chichester, 2002, ISBN 0-471-49517-4.
- [Maroš99] MAROŠ, B. Empirické modely. VUT FSI Brno, Brno, 1999.
- [MJBM97] MAROŠ, B.; JANIČEK, L.; BENEŠ, M.; MAROŠOVÁ, M. Katalog přetvárných odporů. VUT-FS Brno, 1997.
- [MaŠtLa01] MAŘÍK V.; ŠTĚPÁNKOVÁ, O.; LAŽANSKÝ, J. a kol. Umělá inteligence (3). Academia, Praha, 2001, ISBN 80-200-0472-6.
- [MaŠtLa03] MAŘÍK V.; ŠTĚPÁNKOVÁ, O.; LAŽANSKÝ, J. a kol. Umělá inteligence (4). Academia, Praha, 2001, ISBN 80-200-0472-6.
- [McCPit43] MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943, no. 5, pp. 487-492.
- [MeKaMi92] MEZNÍK, I.; KARÁSEK, J.; MIKLÍČEK, J. Matematika I pro strojní fakulty. SNTL, Praha, 1992, ISBN 80-03-00313-X.
- [Michalewicz92] MICHALEWICZ, Z. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, 2nd edition, 1992.
- [Micchelli86] MICCHELLI, C. A. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Construct. Approx.*, 1986, no. 2, pp. 11-22.
- [Mitchell02] MITCHELL, M. An Introduction to Genetic Algorithms. The MIT Press, Cambridge, 2002, ISBN 0-262-13316-4.
- [Misevicius12] MISEVICIUS, V. An implementation of the iterated tabu search algorithm for the quadratic assignment problem. *OR SPECTRUM*, Springer (Business and Economic), 2012. Vol. 34: 3, pp. 665-690.
- [MuKoKhNi08] MUANGSONG, R.; KOOLPIRUCK, D.; KHANTACHAWANA, A. A.; NIRANATLUMPONG, P. A particle swarm optimization approach for optimal design of PID controller for position control using Shape Memory Alloys. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2008. ECTI-CON 2008. vol.2, no., pp.677-680, 14-17 May 2008.
- [NagVij10] NAGARAJ, B. a P. VIJAYAKUMAR. A comparative study of PID controller tuning using GA, EP, PSO and ACO. In *Communication Control and Computing Technologies (ICCCCT)*. 2010, c. 1, 305 - 313.

- [Needl70] NEEDLEMAN, S. B., WUNSCH, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 1970, vol. 48, no. 3, pp. 443-453. DOI: 10.1016/0022-2836(70)90057-4.
- [Novák90] NOVÁK, V. Fuzzy množiny a jejich aplikace. Matematický seminář SNTL, druhé upravené vydání, Praha, 1990, ISBN 80-03-00325-3.
- [ODwyer09] O'Dwyer, A. Handbook of PI and PID Controller Tuning Rules, 3rd edition. Imperial College Press, Third edition, 2009.
- [Ogata08] OGATA, K. Modern Control Engineering: International Version. 5 edition, Pearson, 2008, ISBN 978-0137133376.
- [ONeiBrab06] O'NEILL, M.; BRABAZON, A. Grammatical Differential Evolution. In *Proc. International Conference on Artificial Intelligence (ICAI'06)*. CSEA Press, 2006, pp. 231-236.
- [ONeiRya03] O'NEILL, M.; RYAN, C. Grammatical Evolution (Evolutionary Programming in an Arbitrary Language). Springer, 2003, ISBN 978-1-4020-7444-8.
- [ONeill08] O'NEILL, M.; HEMBERG, E.; GILLIGAN, C.; BARTLEY, E.; MCDERMOTT, J.; BRABAZON, A. GEVA: Grammatical Evolution in Java. *SIGEVolution*, 2008, vol.3, no.2, pp.17-23.
- [OšKvPo97] OŠMERA, P.; KVASNIČKA, V.; POSPÍCHAL, J. Genetic Algorithms with Diploid Chromosomes. In *Proceedings of the 3rd International Conference MENDEL '97*. Brno, 1997.
- [Ošmera98a] OŠMERA, P. Design of fuzzy logic controllers by means of genetic algorithms with diploid chromosomes. In *Proceeding of the 6th Fuzzy-Coloquium*. Zittau, Germany, 1998, pp. 153-158.
- [Ošmera98b] OŠMERA, P. An application of genetic algorithms with diploid chromosomes. In *3rd Japan-Central Europe Joint Workshop on Modelling and Simulation of Non-linear Engineering Systems and Related Phenomena*. STU, Bratislava, Slovensko, 1998, pp. 79.
- [Ošmera03] OŠMERA, P. Paralelní evoluce s hierarchickým uspořádáním. In *Sborník konference Kognicie a umělý život III*. Stará Lesná, Slovensko, 2003, pp. 115-124.
- [PapSte98] PAPADIMITRIOU, C.H.; STEIGLITZ, K. Combinatorial Optimization: Algorithms and Complexity. Dover Publications, New York, 1998.
- [Poggio90] POGGIO, T.; GIROSI, F. Networks for approximation and learning. In *Proceedings of the IEEE*, vol. 78, 1990.
- [Popela04] POPELA, P. Integer and Dynamic Programming. Scriptum, University of Malta, Malta, 2004.
- [Pokorný96] POKORNÝ, M. Umělá inteligence v modelování a řízení. BEN, Praha, 1996, ISBN 80-901984-4-9.
- [Pospíchal03a] POSPÍCHAL, J. Vznik druhov, adaptivny fitness a model odpadlíka. In *Vedecký zborník z III. Slovensko-Českého seminára: Kognícia, umělý život a počítačová inteligencia*. Stará Lesná, 2003, ISBN 80-89066-64.
- [Pospíchal03b] POSPÍCHAL, J. Adaptive fitness and the seceder model of group formation. In *Proceedings of the 9th International Conference MENDEL 2003*. Brno, Czech Republic, 2003, pp. 76-81, ISBN 80-214-2411-7.
- [Powel92] POWELL, M. J. D. The theory of radial basis function approximation. In Light, W. (ed.): *Advances in Numerical Analysis*. Oxford Science Publications, 1992, pp. 105-210.
- [Price06] PRICE, K. V.; STORN, R. M.; LAMPINEN, J. A. Differential evolution: A Practical Approach to Global Optimization. Springer, 2006.
- [RaPoJa09] RAMKUMAR, A.S; PONNAMBALAM, S.G.; JAWAHAR, N. A new iterated fast local search heuristic for solving QAP formulation in facility layout design. *Robotics and Computer-Integrated Manufacturing*. Elsevier (Science direct), vol. 25, issue 3, pp. 620-629. 2009.
- [Rechenberg65] RECHENBERG, I. Cybernetic Solution Path of an Experimental Problem. Farnborough Hants: Ministry of Aviation, Royal Aircraft Establishment, 1965.
- [Rektorys95a] REKTORYS, K. a spol. Přehled užití matematiky I. Šesté přepracované vydání. Prométheus, Praha, 1995, ISBN 80-85849-92-5.

- [Rektorys95b] REKTORYS, K. a spol. Přehled užité matematiky II. Šesté přepracované vydání. Prométheus, Praha, 1995, ISBN 80-85849-62-3.
- [Rosenblatt58] ROSENBLATT, F. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958, no. 65, pp. 386-408.
- [Rosypal98] ROSYPAL, S. Přehled biologie. Scientia, Praha, 1998.
- [Roupec01] ROUPEC, J. Vývoj genetického algoritmu pro optimalizaci parametrů fuzzy regulátorů. Diplomová práce, FSI, VUT Brno. 2001.
- [RyaONei98] RYAN C.; O'NEILL, M.; COLLINS, J. J. Grammatical Evolution: Solving Trigonometric Identities. In *Proceedings of Mendel 1998: 4th International Mendel Conference on Genetic Algorithms, Optimisation Problems, Fuzzy Logic, Neural Networks, Rough Sets*. 1998, Brno, Czech Republic, pp. 111-119.
- [Sandgren90] SANDGREN, E. Nonlinear integer and discrete programming in mechanical design optimisation. *Transactions of the ASME, Journal of Mechanical Design*, 1990, vol. 112, no. 2, pp.223-229, ISSN 0738-0666.
- [Sanders11] SANDERS, J.; KANDROT, E. CUDA by Example. Wesley, USA, 2011 (NVIDIA series), ISBN 978-0-13-138768-3.
- [SchBel92] SCHRANDOLPH, N. N.; BELEW, R. K. Dynamic Parameter Encoding for GAs. Machine Learning, 1992.
- [Šeda01] ŠEDA, M. Steinerovy problémy a přibližné metody jejich řešení. Habilitační práce, FSI VUT Brno, Brno, 2001.
- [Sekaj12] Sekaj, I; Kadlic, B.; Gasparik, T; Cartesian Genetic Programming Based Controller Design. In Radomil Matousek editor, *18th International Conference on Soft Computing, MENDEL 2012*, pages 19-23, Brno, Czech Republic, 2012.
- [Šíma96] ŠÍMA, J.; NERUDA, R. Teoretické otázky neuronových sítí. MatFyzPress, UK Praha, 1996.
- [Šlechta99] ŠLECHTA J. Optimální rozhodování a řízení. Skriptum ČVUT – FE, Praha, 1999.
- [Steinberg61] STEINBERG, L. The backboardwiring problem: A placement algorithm. *SIAM Review*, USA, 3:37-50, 1961.
- [Storn95] STORN, R.; PRICE, K. Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. TR-95-012, Int CS Institute, University of California, Mar 1995.
- [Stützle99] STÜTZLE, T. Iterated local search for the quadratic assignment problem. Technical Report AIDA-99-03, FG Intellektik, FB Informatik, TU Darmstadt, Darmstadt, Germany, March 1999.
- [Sundaram09] SUNDARAM, R. K. A First Course in Optimization Theory. 15th print. Cambridge University Press, 2009, ISBN 978-0-521-49770-1
- [Taillard91] TAILLARD E.D. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 1991, vol. 17, pp. 443-455.
- [Taillard95] TAILLARD E.D. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 1995, vol. 3, pp. 87-105.
- [Vávra84] VÁVRA, P. Strojnické tabulky. Druhé vydání, SNTL, Praha, 1984.
- [Vose99] VOSE ,M. D. The Simple Genetic Algorithm – Foundations and Theory. The MIT Press, Cambridge, 1999, ISBN 0-262-22058-X.
- [Voudouris97] VOUDOURIS, C. Guided Local Search for Combinatorial Optimization Problems. PhD thesis, Department of Computer Science, University of Essex, Colchester, UK, 1997.
- [Wang06] WANG, F. Y., DERONG, L. Advances in Computational Intelligence (Theory and Applications). World Scientific, New Jersey, 2006, ISBN 981-256-734-8.
- [WihCho95] WHITEHEAD, B. A.; CHOATE, T. D. Cooperative – Competitive Genetic Evolution of Radial Basis Function for Time Series Prediction. *IEEE Transactions on Neural Networks*, 1995.
- [Wolpert95] WOLPERT, D. H.; MACREADY, W. G. No Free Lunch Theorems for Search. Technical Report SFI-TR-95-02-010 (Santa Fe Institute), 1995.

- [Wolsey98] WOLSEY, L. A. Integer Programming. John Wiley & Sons, Canada, 1998. ISBN 0471-28366-5.
- [Zadeh65] ZADEH, L. A. Fuzzy Sets. In *Information and Control*, 1965, pp. 338-353.
- [Zadeh94] ZADEH, L. A. Soft Computing and Fuzzy Logic. *IEEE Software*, 1994, vol. 11, no. 6, pp.48-56.
- [Zadeh01] ZADEH, L. A. Applied Soft Computing. *Applied Soft Computing*, 2001, vol. 1, no. 1-2.
- [Zamp10] ŽAMPACHOVÁ, E. Approximations in stochastic optimization and their applications. *Disertační práce*, VUT v Brně, 2010.
- [Zelinka02] ZELINKA, I. Umělá Inteligence – v problémech globální optimalizace. Nakladatelství BEN, Praha, 2002, ISBN 80-7300-069-5.
- [Zelinka09] ZELINKA, I.; OPLATKOVÁ, Z.; ŠEDA, M., OŠMERA, P.; VČELAŘ, F. Evoluční výpočetní techniky. BEN, 2009. ISBN: 978-80-7300-218-3.

VYBRANÉ REFERENCE AUTORA

- [A01] MATOUŠEK, R. Genetické algoritmy – knihovna metod. Diplomová práce, ÚAI VUT-FS Brno, 1996.
- [A02] OŠMERA, P., MATOUŠEK, R. Automatic Optimal Design of Fuzzy Controllers Based on GA Algorithms. In Proceedings of 2nd International Conference GALEZIA '97. Glasgow, UK, 1997, pp. 439-443.
- [A03] OŠMERA, P., MATOUŠEK, R. Fuzzy Controllers Based on Genetic Algorithms. 3NWGA, Helsinki, Finland, 1997.
- [A05] MATOUŠEK, R., OŠMERA, P. Possibilities of Fuzzy Control of GA Parameters. In Proceedings of the 5th International Conference MENDEL '99. Brno, Czech Republic, 1999, pp.266-270, ISBN 80-214-1131-7.
- [A08] MATOUŠEK, R., OŠMERA, P. A Fuzzy Setting of GA Parameters. In Digest of the Joint Seminar '99 Sapporo. Japan Society of Applied Electromagnetics and Mechanics, Japan, 1999, pp.52-56.
- [A10] MATOUŠEK, R., OŠMERA, P. GA-FIS for Dynamic Environment. In Sinčák, P., Vaščák, J., Kvasnička, V., Mesiar, R. (eds.): The State of the Art in Computational Intelligence. Physica –Verlag, Springer - Verlag Company, 2000, pp. 191-196, ISBN 3-7908-1322-2.
- [A13] MATOUŠEK, R., PFLÉGROVÁ, J., PRAX, P., MIČÍN, J. Application of Genetic Algorithms in Urban Drainage. In Proceedings of the 6th International Conference MENDEL 2000. Brno, Czech Republic, 1999, pp. 90-96, ISBN 80-214-1609-2.
- [A14] OŠMERA, P., ROUPEC, J., MATOUŠEK, R.: Genetic Algorithms with Sexual Reproduction. In Proceeding of World Multiconference on Systemics, Cybernetics and Informatics SCI 2000. Orlando, Florida, USA, July 23-26, 2000, pp. 306-311.
- [A39] MATOUŠEK, R. Vybrané metody umělé inteligence - Implementace a aplikace, Disertační práce v oboru Technická kybernetika, VUT v Brně, 2004. s. 200 (s.)
- [A40] MATOUŠEK, R. GAFIS: Genetic Algorithm with Fuzzy Inference System. In SCIS & ISIS 2004. Yokohoma, Japan: Keio University, 2004. s. WP-2- 3 (s.)
- [A44] MATOUŠEK, R. GAHC: Improved GA with HC mutation. In WCECS 2007. San Francisco, USA: Newswood Limited, 2007. s. 915-920. ISBN: 978-988-98671-6- 4.
- [A46] MATOUŠEK, R. GAHC: Hybrid Genetic Algorithm. In Mendel 2007. 2007. Brno: BUT, 2007. s. 7-11. ISBN: 978-80-214-3473- 8.
- [A49] MATOUŠEK, R.; NĚMEC, Z.; MARADA, T. Intelligent Weighing of Goods for Packing. *Cybernetic Letters*, 2008, roč. 6, č. II, s. 1-5. ISSN: 1802- 3525.
- [A55] MATOUŠEK, R.; BEDNÁŘ, J. Grammatical Evolution: Epsilon Tube in Symbolic Regression Task. In MENDEL 2009. Mendel Journal series. MENDEL. Brno: BUT, 2009. s. 9-15. ISBN: 978-80-214-3884- 2. ISSN: 1803- 3814.
- [A56] ŠEDA, M.; MATOUŠEK, R.; OŠMERA, P.; PIVOŇKA, P.; ŠANDERA, Č. A Flexible Heuristic Algorithm for Resource- Constrained Project Scheduling. In World Congress on Engineering and Computer Science 2009. San Francisco, USA: IAENG, 2009. s. 730-734. ISBN: 978-988-18210-2- 7.
- [A57] MATOUŠEK, R. Genetic Algorithm and Advanced Tournament Selection Concept. 2009, roč. 2009, č. 236, s. 189-196. ISSN: 1860- 949X.

- [A61] MATOUŠEK, R. Grammatical Evolution: STE criterion in Symbolic Regression Task. In World Congress on Engineering and Computer Science 2009. San Francisco, USA: IAENG, 2009. s. 730-734. ISBN: 978-988-18210-2- 7.
- [A63] MATOUŠEK, R.; LANG, S. Optimalizace parametrů PID regulátoru pomocí GA toolboxu. In Bratislava: RT Systems, 2010. s. 69-73. ISBN: 978-80-970519-0- 7.
- [A64] WEISSER, R.; OŠMERA, P.; ROUPEC, J.; MATOUŠEK, R. Two- Level Transplant Evolution for Optimization of General Controllers. In New Trends in Technologies: Control, Management, Computational Intelligence and Network Systems. 2010. Croatia, <http://www.intechopen.com/articles/show/title/transplant-evolution-for-optimization-of-general-controllers>: Sciy, 2010. s. 50-68. ISBN: 978-953-307-213- 5.
- [A69] MATOUŠEK, R. HC12: Highly Scalable Optimization Algorithm. Studies in Computational Intelligence, Springer Berlin / Heidelberg, 2010, roč. 2010, č. 284, s. 177-183. ISSN: 1860- 949x.
- [A70] MATOUŠEK, R. HC12 Implemented Using the CUDA Platform. Applied Computer Science, 2010, roč. 2010, č. 2010, s. 649-652. ISSN: 1792- 4863.
- [A72] MATOUŠEK, R.; BEDNÁŘ, J. Grammatical Evolution and STE Criterion (Statistical Properties of STE Objective Function). Lecture notes in Electrical Engineering, 2010, roč. 2010, č. 68, s. 131-142. ISSN: 1876- 1100.
- [A73] MATOUŠEK, R. HC12: The Principle of CUDA Implementation. In MENDEL 2010. Mendel Journal series. 2010. Brno: VUT, 2010. s. 303-308. ISBN: 978-80-214-4120- 0. ISSN: 1803- 3814.
- [A74] WEISSER, R.; OŠMERA, P.; MATOUŠEK, R. Transplant Evolution with Modified Schema of Differential Evolution: Optimization Structure of Controllers. In MENDEL 2010. Mendel Journal series. 2010. Brno: VUT, 2010. s. 113-120. ISBN: 978-80-214-4120- 0. ISSN: 1803- 3814.
- [A76] KLAPKA, J.; MATOUŠEK, R.; ŠEVČÍK, V. Improvement of Time- Periodical Production Schedule of the Group of Products in the Group of Workplaces Through the Lot Sizes Alteration. In Proceedings of MENDEL 2011 -- 17th International Conference on Soft Computing. Mendel Journal series. Brno: Brno University of Technology, 2011. s. 334-340. ISBN: 978-80-214-4302- 0. ISSN: 1803- 3814.
- [A77] MATOUŠEK, R. Using AI Methods to Find a Non- Linear Regression Model with a Coupling Condition. Engineering Mechanics, 2011, roč. 17, č. 5/ 6, s. 419-431. ISSN: 1802- 1484.
- [A78] MATOUŠEK, R.; ŽAMPACHOVÁ, E. Promising GAHC and HC12 algorithms in global optimization tasks. OPTIMIZATION METHODS & SOFTWARE, 2011, roč. 26, č. 3, s. 405-419. ISSN: 1055- 6788.
- [A80] MATOUŠEK, R.; NĚMEC, Z. Optimal Criteria for Non- Minimal Phase Plants. An international Journal of Science, Engineering and Technology, World Academy of Science Engineering and Technology, 2011, roč. 2011, č. 59, s. 824-828. ISSN: 2010- 376X.
- [A81] MATOUŠEK, R.; LANG, S.; MINÁŘ, P.; PIVOŇKA, P. Evolutionary Design of Polynomial Controller. An international Journal of Science, Engineering and Technology, World Academy of Science Engineering and Technology, 2011, roč. 2011, č. 59, s. 639-644. ISSN: 2010- 376X.
- [A83] MATOUŠEK, R.; KARPÍŠEK, Z. Exotic Metrics for Function Approximation. In 17th International Conference of Soft Computing, MENDEL 2011 (id 19255). Mendel Journal series. 2011. Brno: VUT, 2011. s. 560-566. ISBN: 978-80-214-4302- 0. ISSN: 1803- 3814.
- [A86] MATOUŠEK, R.; MINÁŘ, P.; LANG, S.; ŠEDA, M. HC12: Efficient PID Controller Design. Engineering Letters, 2012, roč. 20, č. 1, s. 42-48. ISSN: 1816- 093X.
- [A88] MATOUŠEK, R.; NĚMEC, Z.; MARADA, T. Weighting of Goods by means of DC Motor. Advanced Materials Research, 2012, roč. 2012, č. 463- 464, s. 1427-1431. ISSN: 1022- 6680.
- [A89] ABBADI, A.; MATOUŠEK, R. RRTs Review and Statistical Analysis. International Journal of Mathematics and Computers in Simulation, 2012, vol. 6, No. 1, pp. 1-8. ISSN: 1998- 0159.

Abstrakt

Předložená habilitační práce prezentuje vybranou část autorova výzkumu v oblasti metod počítačové inteligence s důrazem na řešení úloh globální optimalizace. Jsou popsány originální optimalizační algoritmy HC12, GAHC a GAFIS. Je prezentována nová metoda selekce a nové kritérium optimality označené jako STE. Práce rovněž ukazuje jednu ze zásadních výhod algoritmu HC12, a to dobrou škálovatelnost. V praktické části je na standardních testovacích funkcích prezentován výkon algoritmů a ukázána metodika optimální parametrizace GA. Praktická řešení úloh aproximace s funkčním omezením, QAP a návrh obecného lineárního regulátoru prezentují ve více ohledech unikátní řešení daných problémů. V případě řešení úloh QAP třídy esc jde o světově porovnatelné výsledky. Kódování swap2abs i algoritmy HC12 a GAHC jsou potenciálně využitelné v řadě dalších aplikací globální optimalizace.

Abstract

The thesis presents a selected part of the author's research in methods of computer intelligence focusing on solutions to global optimization problems. Original HC12, GAHC, and GAFIS algorithms are described. A new selection method is presented and a new optimality criterion denoted by STE is proposed. The thesis also shows a major advantage of the HC12 algorithm, which is a good scalability. In the practical part, standard testing functions are used to show the performance of each algorithm presented and a methodology is described for finding optimum GA parameters. In multiple aspects, the practical solutions to approximation problems, QAP combinatorial problems, and a general linear controller design present unique solutions to the above problems. The solutions to the class esc QAP problems bear comparison on an international scale. Both the swap2abs coding and the HC12 and GAHC algorithms have the potential of being used in a number of other global optimisation applications.