

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Ústav počítačových systémů

**Ing. Josef Schwarz, CSc.**

**BAYESOVSKÉ EVOLUČNÍ ALGORITMY S APLIKACEMI  
V ÚLOHÁCH DEKOMPOZICE A ALOKACE**

**BAYESIAN EVOLUTIONARY ALGORITHMS APPLIED  
IN DECOMPOSITION AND ALLOCATION PROBLEMS**

ZKRÁCENÁ VERZE HABILITAČNÍ PRÁCE



BRNO 2003

## **KLÍČOVÁ SLOVA**

multikriteriální optimalizační úlohy, úlohy dekompozice a alokace, tradiční optimalizační metody, genetické algoritmy, pravděpodobnostní modely, algoritmus s dvourozměrným rozložením, bayesovské sítě, Bayes-Dirichletova metrika, binární rozhodovací stromy, metriky, bayesovské evoluční algoritmy

## **KEY WORDS**

multiobjective optimization problems, decomposition and allocation problems, classical optimization methods, genetic algorithms, probabilistic models, bivariate marginal distribution algorithm, bayesian networks, Bayesian-Dirichlet metric, binary decision diagrams, scoring metrics, bayesian evolutionary algorithms

## **MÍSTO ULOŽENÍ PRÁCE**

Vědecké oddělení Fakulty informačních technologií VUT v Brně, Božetěchova 2, 612 66 Brno.

# OBSAH

|   |    |
|---|----|
| PŘEDSTAVENÍ AUTORA.....   | 4  |
| 1 PŘEDMĚT HABILITAČNÍ PRÁCE.....                                      | 5  |
| 2 ÚVOD .....  | 5  |
| 2.1 Klasické optimalizační metody vs. evoluční algoritmy .....        | 6  |
| 2.2 Heuristické metody.....   | 6  |
| 2.3 Standardní genetické algoritmy (SGA) .....                        | 6  |
| 2.3.1 Formalizace činnosti SGA .....                                  | 8  |
| 2.3.2 Teorie schémat .....  | 9  |
| 2.4 EDA algoritmy .....   | 10 |
| 3 ŘEŠENÍ ÚLOH DEKOMPOZICE A ALOKACE.....                              | 14 |
| 3.1 Úloha alokace/rozmístění obvodových uzlů .....                    | 15 |
| 3.1.1 Heuristická metoda limitovaných řezů .....                      | 16 |
| 3.1.2 Paralelní standardní genetický algoritmus (PGA) .....           | 17 |
| 3.2 Bayesovské algoritmy v úloze dekompozice hypergrafů.....          | 18 |
| 3.2.1 Srovnání evolučních algoritmů BOA, BMDA a SGA.....              | 20 |
| 3.2.2 Algoritmus KBOA využívající apriorní informaci .....            | 21 |
| 3.2.3 BOA algoritmus pro rozmíst'ování .....                          | 22 |
| 3.2.4 Multikritériální BOA algoritmus pro dekompozici hypergrafů..... | 22 |
| 3.2.5 Teorie a praxe multikritériálních BOA algoritmů .....           | 23 |
| 3.2.6 MBOA dekompoziční algoritmus s poměrovým řezem .....            | 24 |
| 3.2.7 MBOA dekompoziční algoritmus s volitelným kritériem .....       | 25 |
| 4 ZÁVĚR .....   | 26 |
| 5 LITERATURA.....   | 27 |
| 5.1 Přehled vybrané literatury .....                                  | 27 |
| 5.2 Přehled článků tvořících předloženou habilitační práci .....      | 29 |
| 5.3 Seznam vybraných prací autora .....                               | 30 |
| ABSTRACT .....  | 32 |

## PŘEDSTAVENÍ AUTORA



**Ing. Josef Schwarz, CSc.** se narodil 11. 12. 1944 v Přísnoticích. V roce 1966 absolvoval s vyznamenáním studium na FE VUT v Brně v oboru technická kybernetika. V roce 1968 nastoupil na katedru počítačů FE VUT jako odborný pracovník. Podílel se na výuce předmětů technického charakteru, což souviselo s jeho hlavním odborným zaměřením a funkcí vedoucího inženýra výpočetního centra. V roce 1984 obhájil na FE VUT kandidátskou disertační práci „Automatizace návrhu kompozice a rozmístění IO na desce s plošnými spoji“ v oboru Elektronické počítače.

V roce 1992 se stal odborným asistentem a garantem kurzu Konstrukce počítačů a Aplikované mikro počítače. Od roku 1996 je také garantem kurzu Číslicové a impulzové obvody. V roce 2001 zavedl nový předmět Aplikované evoluční algoritmy. V novém studijním programu Informační technologie je garantem předmětů Aplikované evoluční algoritmy, Softcomputing a Počítačový návrh. Je spoluautorem čtyř vysokoškolských skript. V rámci projektu Tempus 0449/92 absolvoval šestitýdenní pobyt na univerzitě v Bristolu v Anglii.

Ve své dlouholeté výzkumné činnosti se odborně zaměřuje na problematiku automatizace návrhu číslicových obvodů, návrh fuzzy vývojových systémů a fuzzy aplikací a využití pokročilých evolučních algoritmů a fuzzy logiky při řešení inženýrských úloh. Podílel se na deseti výzkumných projektech, včetně spolupráce s průmyslovými podniky a vedl dvacet diplomových prací. Výsledky své činnosti publikoval na mezinárodním fóru – je autorem nebo spoluautorem 37 článků publikovaných v časopisech a ve sbornících mezinárodních konferencí. Autor byl členem výboru pěti mezinárodních konferencí a podílel se na recenzích řady odborných článků a disertačních prací z oblasti evolučních algoritmů.

# 1 PŘEDMĚT HABILITAČNÍ PRÁCE

Habilitační práce „Bayesovské evoluční algoritmy s aplikacemi v úlohách dekompozice a alokace“ se zabývá návrhem, analýzou a využitím Bayesovských evolučních algoritmů pro řešení složitých, vesměs NP-úplných optimalizačních kombinatorických problémů zejména z oblasti dekompozice a alokace grafových struktur, které lze v kontextu fyzického návrhu číslicových obvodů interpretovat jako členění a rozmístování číslicových obvodů. Jde o pokročilé evoluční algoritmy založené na pravděpodobnostních modelech, které odstraňují nevýhody standardních evolučních algoritmů spojené s volbou genetických operátorů a nastavováním četných řídicích parametrů. Souhrnným označením pro tuto třídu algoritmů je zkratka EDA (Estimation Distribution Algorithms) – volně přeloženo algoritmy založené na odhadu pravděpodobnostního rozložení slibných řešení [Larra99], [Muel98], [Pel99c]. V roce 1999 byl poprvé publikován nový EDA algoritmus BOA (Bayesian optimization algorithm) [Pel99a], [Pel99b], který využívá pravděpodobnostní model reprezentovaný Bayesovskou sítí. Z uvedených publikací vyplývá, že tento algoritmus byl testován jen na umělých úlohách s binárním zakódováním a tzv. klamných úlohách.

Autor rozpracoval a zobecnil ideu Bayesovského evolučního optimalizačního algoritmu ve čtyřech oblastech:

- Kombinatorické optimalizační úlohy;
- Využití dodatečných znalostí o řešeném problému;
- Multikriteriální optimalizace;
- Obecnější využití binárních rozhodovacích grafů (BDD).

Výběr aplikací, převážně z oblasti fyzického návrhu číslicových obvodů, vyplývá z odborného a pedagogického zaměření autora, který se dlouhodobě zabývá návrhem číslicových obvodů, mikro počítačových podsystemů a návrhem evolučních algoritmů.

Důraz kladený na řešení úloh dekompozice a alokace vyplývá jednak z obtížnosti řešení těchto úloh pro většinu používaných heuristik (jde vesměs o NP-úplné úlohy vhodné pro testování), ale také proto, že obě úlohy jsou dostatečně obecné a pokrývají širší škálu úloh včetně syntézy a testování číslicových obvodů.

Habilitační práce je podána formou souboru devíti článků (A-1 až A-9) publikovaných v mezinárodních časopisech a mezinárodních konferencích s průvodním komentářem.

## 2 ÚVOD

V současné době jsme svědky stále širšího využívání evolučních algoritmů pro řešení teoretických úloh a rozmanitých úloh praxe. Lze dokonce hovořit o stále větší popularitě této třídy algoritmů. Je oceňována jejich robustnost – schopnost nalézt globální optimum i pro multimodální multikriteriální úlohy s ostrými zlomy povrchu účelové funkce (function/fitness landscape). Je však známo, že konkrétní verzi evolučního algoritmu lze použít efektivně jen pro jistý okruh problémů (no free lunch theorem). Není tedy účelné používat evoluční algoritmy pro jakýkoliv typ řešeného problému. Existuje řada klasických propracovaných metod, které jsou schopny uspokojivě řešit řadu optimalizačních úloh s relativně nízkou výpočetní náročností.

## 2.1 KLASICKÉ OPTIMALIZAČNÍ METODY VS. EVOLUČNÍ ALGORITMY

První významnou skupinu klasických metod tvoří algoritmy používané pro řešení klasických grafových úloh: hledání souvislých komponent, minimální cesty, minimálních stromových struktur, maximálního toku, optimálního párování hran, obarvení grafu.

Samostatnou skupinu tvoří metody operačního výzkumu [DUDO79]. Ke známým úlohám patří lineární a celočíselné programování, nelineární (kvadratické) programování a dynamické programování. Pro řešení lineárních úloh se často používá simplexová metoda, pro úlohy nelineárního programování s hladkou funkcí (smooth function) např. gradientní metody GPM (Gradient-projection Method) a GRG (Generalized Reduced Gradient). Pro NP-úplné problémy lze použít metodu větvení a hranic (Branch and Bound) a simulované žíhání.

Při rozhodování o volbě metody je tedy vhodné klasifikovat předložený problém, zda jej nelze převést na některou úlohu z teorie grafů nebo operačního výzkumu. Např. je známo, že dekompozici grafu bez omezujících podmínek na velikost vzniklých subgrafů lze převést na úlohu hledání maximálního toku v síti. Tento trend je respektován v profesionálních optimalizačních systémech, které agregují klasické metody, různé heuristické přístupy a evoluční algoritmy (např. systém Premium Solver V3.5 [Solv35]).

## 2.2 HEURISTICKÉ METODY

Heuristické metody patří k nejrozšířenějším metodám. Jsou vesměs založeny na iterativních technikách (deterministických nebo náhodných) postupného zlepšování řešení [Micha00]. Je možné nalézt globální optimum, ale v procesu hledání nelze specifikovat ani vzdálenost aktuálně dosaženého řešení od globálního optima, ani s jakou pravděpodobností může být globálního optima dosaženo. Nejznámějším přístupem např. v alokačních úlohách je uplatnění sekvencí párových nebo vícenásobných záměn objektů pro vylepšení konfigurace těchto objektů. Výběr dvojic je buď náhodný nebo systematický a akceptuje se každá rekonfigurace vedoucí ke zlepšení účelové funkce.

## 2.3 STANDARDNÍ GENETICKÉ ALGORITMY (SGA)

Standardní genetické algoritmy SGA (Standard/Simple Genetic Algorithms), jako samostatná podtřída evolučních algoritmů, patří mezi stochastické optimalizační algoritmy založené na analogii s evolučními procesy probíhajícími v biologických systémech [Holl75], [Gold86], [Back96], [Kvas00], [Foge00], [Mari01].

Biologická evoluce (podle Darwinovy evoluční teorie a Mendlovy teorie dědičnosti) je progresivní změna genetické informace jedinců populace během mnoha generací v rámci evoluční epochy. Lze ji charakterizovat třemi základními fenomény:

- Přírozeným výběrem, kdy schopnější jedinci (s větší silou, výhodností) mají více potomků než jedinci slabší, méně adaptovaní na prostředí.
- Procesem reprodukce, kdy nový jedinec vzniká rekombinací/křížením genetické informace rodičů, přičemž může docházet k poruchám vlivem mutace.
- Náhodným genetickým driftem, kterým může být exitus schopného jedince před realizací reprodukce, případně jeho náhodná mutace. Uplatňuje se u malých populací.

Tato teorie evoluce, mnohdy nazývaná syntetickou teorií evoluce nebo také neodarwinismem, je založena na genech jako nositelích dědičnosti. Individuum je charakterizováno řetězcem genů –

chromozómem (charakterizujícím jeho genetickou výbavu – genotyp). Fenotyp individua je dán interakcí jeho genotypu s prostředím. Výhodnost jedince (fitness) je dána mírou jeho schopností přežít v daném prostředí. Jednotkou vývoje je populace jedinců, která se skládá z genofondu daného genotypem všech jedinců.

V současné době existuje celá paleta evolučních algoritmů založených na různé interpretaci evoluční teorie [Back96]. K nejznámějším patří Evoluční strategie (ES), Evoluční programování (EP) a Genetické programování (GP).

Objevují se také nové expanze původního Darwinovského paradigmatu obohacující jedince o adaptabilní kognitivní orgán (Baldwinův efekt), popřípadě o sociální zkušenost reprezentovanou mémy, které zavedl R. Dawkinson ve své knize „The Selfish Gene“ [Kva00a].

Další optimalizační techniky využívají analogie se strukturou a chováním biologických pod systémů např. imunitního a reprodukčního systému. Ukazuje se také, že vhodnou agregací dílčích evolučních technik lze realizovat robustní, efektivní evoluční výpočetní prostředky pro řešení složitých multimodálních úloh s měnící se účelovou funkcí [Ošme02].

Samostatnou třídu tvoří paralelní genetické algoritmy využívající implicitní paralelnost evolučního procesu a hybridní genetické algoritmy, které využívají rychlých heuristik pro zlepšení konvergence [Ošme02].

Evoluční algoritmy (EA) jsou typické svojí robustností, schopností řešit obtížné optimalizační a rozhodovací úlohy, které lze charakterizovat vlastnostmi jako je multimodálnost, multikriteriálnost a různými typy omezujících podmínek. Jejich nasazení je efektivní v úlohách, které lze definovat následovně:

- Prostor řešení je příliš rozsáhlý a chybí expertní znalost, která by umožnila zúžit prostor slibných řešení.
- Nelze provést matematickou analýzu problému.
- Tradiční metody selhávají.
- Jde o úlohy s mnohačetnými extrémy, kritérii a omezujícími podmínkami.

EA se používají pro numerickou i kombinatorickou optimalizaci, při návrhu obvodů, plánování výroby, strojovém učení, v tvorbě ekonomických, sociálních a ekologických modelů atd.

Je však třeba poukázat na jisté nevýhody evolučních algoritmů:

- Kvalitu řešení lze ohodnotit pouze relativně. Nelze otestovat, zda se jedná o globální optimum.
- Pro mnohé úlohy je typická velká časová náročnost.
- Pro příliš rozsáhlé úlohy poskytuje řešení příliš vzdálená od optima.
- Ukončení optimalizace je explicitní na základě časového limitu nebo stagnace kritériální funkce.

Pro návrh evolučních algoritmů existuje řada komerčních produktů, které umožňují rychlý návrh aplikací. Klíčovým krokem je volba zakódování problému, která by měla respektovat použité genetické operátory a náročnost výpočtu účelové funkce.

### 2.3.1 Formalizace činnosti SGA

Návrh genetického algoritmu pro řešení optimalizačního problému je určen vždy konkrétní specifikací evolučních principů a jejich algoritmizací. Všechny realizace genetických algoritmů však využívají reprodukčního paradigmatu standardního genetického algoritmu SGA. Z hlediska cíle této práce je tedy postačující vymezovat EDA algoritmy jen vůči struktuře SGA algoritmů.

Nejdříve definujme základní pojmy, které budeme dále používat. Chromozóm (jedinec/individuum) kódující řešení je reprezentován binárním vektorem (řetězcem) konstantní délky  $n$ :

$$\begin{aligned} X &= (X_0, X_1, \dots, X_{n-1}), \text{ kde } X_i \text{ je } i\text{-tou proměnnou řetězce,} \\ x &= (x_0, x_1, \dots, x_{n-1}) \text{ je řetězec konkrétních instancí proměnných } X_i = x_i, x_i \in \{0, 1\}, \\ D &= (X^1, X^2, \dots, X^N), X^j \in D \text{ je množina } N \text{ řetězců, která specifikuje populaci } D, \\ D &\subseteq \{0, 1\}^n. \end{aligned}$$

Nechť  $f$  je účelová/cenová funkce definovaná nad množinou binárních vektorů délky  $n$

$$f: \{0, 1\}^n \rightarrow \mathbb{R}, \quad (1)$$

kteřá ohodnotí každý binární vektor  $x$  reálným číslem. Cílem je nalézt globální extrém funkce  $f$ . V případě minimalizační úlohy jde o nalezení vektoru

$$x_{opt} = \arg \min_{x \in \{0, 1\}^n} f(x) \quad (2)$$

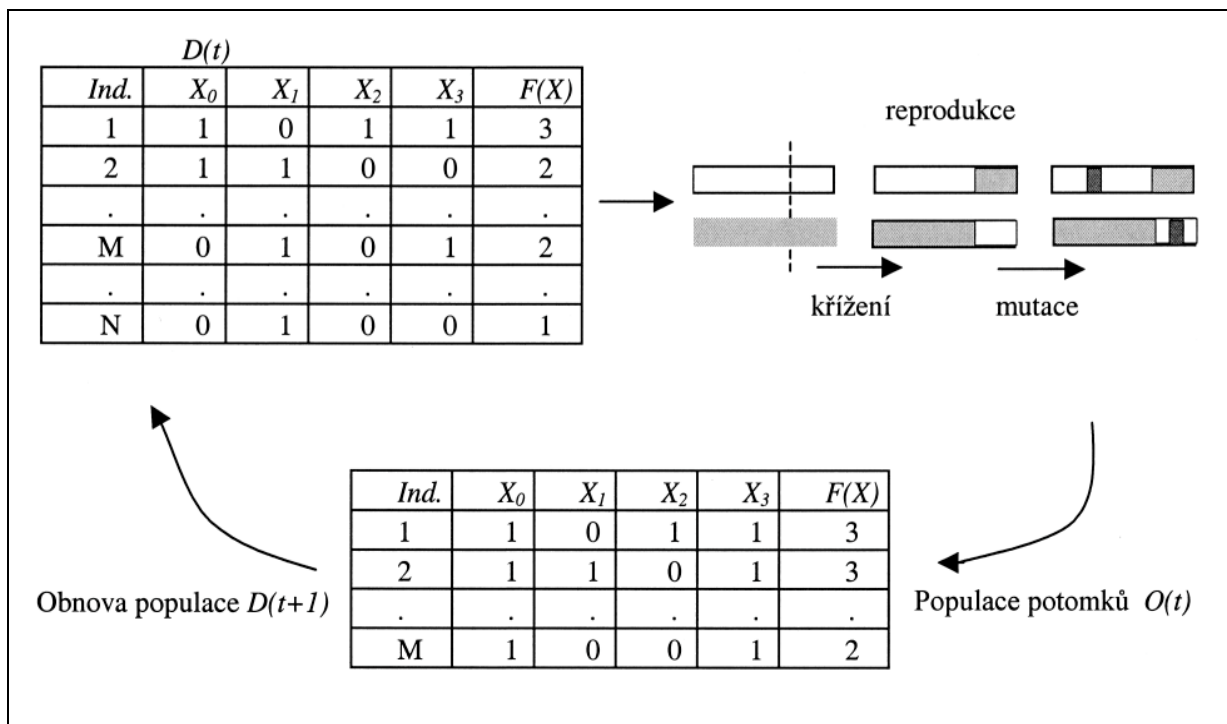
Funkce  $f$  se zpravidla transformuje na funkci výhodnosti  $F$  (fitness funkci) tak, aby původní optimalizační úloha byla převedena vždy na maximalizační úlohu a bylo dosaženo vhodného měřítka fitness funkce. Užití této transformace usnadňuje také změnu selekčního tlaku, který výrazně ovlivňuje konvergenci evolučního procesu.

Činnost standardního GA algoritmu lze popsat následujícím pseudokódem:

- (1) Nastav  $t=0$ , náhodně generuj počáteční populaci  $D(0)$  s mohutností  $N$ ,
- (2) Proveď ohodnocení jedinců populace  $D(t)$  funkcí výhodnosti  $F(X)$ ,
- (3) Generuj populaci potomků  $O(t)$  s mohutností  $M \leq N$  použitím operátorů křížení a mutace,
- (4) Vytvoř novou populaci  $D(t+1)$  nahrazením části populace  $D(t)$  jedinci z  $O(t)$ ,
- (5) Nastav  $t \leftarrow t+1$ ,
- (6) Pokud není splněna podmínka pro ukončení algoritmu, jdi na (2).

Schematické znázornění genetického algoritmu je na obr.1. V prvním kroku je generována počáteční populace vektorů řešení/jedinců. Každý jedinec je ohodnocen pomocí fitness funkce  $F(X)$ . V procesu křížení (uplatňovaném s četností křížení  $p_c$ ) vznikají rekombinací vybraných dvojic individuí/rodičů dva potomci/řetězce, které jsou s četností  $p_m$  náhodně mutovány a zařazovány do populace potomků. V procesu tvorby nové rodičovské populace část jedinců z populace potomků vytěsňuje jedince předchozí rodičovské populace. Celý proces se opakuje v rámci evoluční epochy, která je specifikována maximálním počtem generací nebo na základě stupně saturace aktuální populace identickými jedinci. Před aktivací GA je nutné specifikovat typy genetických operátorů a hodnoty parametrů – četnost selekce, křížení a mutace.





Obr. 1 Ilustrace standardního genetického algoritmu

### 2.3.2 Teorie schémat

Evoluční proces reprodukce populace řešení, specifikovaný operátory selekce, křížení a mutace vedoucí k nalezení optima nebo suboptima, byl formalizován J. Hollandem [Holl75] s použitím teorie schémat, která byla později rozpracována dalšími autory [Gold86]. Z teorie i praxe genetických algoritmů je znám stále aktuální problém vhodného zakódování řešení a volby vhodného sortimentu genetických operátorů, které výrazně ovlivňují konvergenci řešení. Většina modelů GA se opírá o zmíněnou teorii schémat – schéma je podobnostní šablona definovaná nad abecedou  $\{0, 1, *\}$ . Volný symbol „\*“ znamená, že se na příslušné pozici binárního řetězce odpovídající danému schématu může vyskytovat libovolná hodnota 0 nebo 1. Říkáme, že řetězec  $r \in \{0, 1\}^n$  je příkladem/vzorem schématu  $t \in \{0, 1, *\}^n$ , pokud v každé pozici schématu s jiným znakem než „\*“ je hodnota znaku v řetězci stejná jako ve schématu. Počet pevně stanovených pozic ve schématu  $H$  se nazývá řádem schématu  $o(H)$ , vzdálenost mezi první a poslední pevnou pozicí se nazývá délkou schématu  $\delta(H)$ . Schéma  $(0^*1)$  má tedy řád 2 a délku 3. Místo sledování šíření jednotlivých jedinců je účelnější sledovat šíření schémat, která reprezentují příslušnou podmnožinu jedinců.

Nechť v čase  $t$  obsahuje populace  $D(t)$  počet  $m$  chromozomů vyhovujících podobnostnímu schématu  $H$ , tedy

$$m = m(H, t).$$

Výskyt schématu v populaci po realizaci reprodukčních operátorů je dán vztahem [Gold86]

$$m(H, t+1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \left[ 1 - p_c \cdot \frac{\delta(H)}{n-1} - o(H) \cdot p_m \right]. \quad (3)$$

Vztah (3) je důležitou formulí pro vysvětlení činnosti SGA. Slovně se vyjadřuje jako teorém o schématech: *Počet krátkých nadprůměrných schémat nízkého řádu v jednotlivých generacích exponenciálně roste. Z teorému bezprostředně vyplývá hypotéza o stavebních blocích: Genetický algoritmus upřednostňuje a přeskupuje nadprůměrná schémata nízkého řádu nazývaná stavební*

*bloky (částečná řešení daného problému)*. Delší schémata vyšších řádů jsou operátory poškozována, a to úměrně své definující délce a svému řádu. Tento fenomén rozbíjení stavebních bloků je hlavní příčinou pomalé konvergence standardních genetických algoritmů hlavně v případě nelineárních multimodálních účelových funkcí a klamných problémů (deceptive problems). Při řešení klamných úloh vykazují SGA anomální chování. Stavební bloky obsahují komponenty, které mohou směřovat řešení do lokálního optima. Existují totiž suboptimální schémata, která mají vyšší ohodnocení než optimální schémata stejného řádu a délky.

Dodejme, že vztah (3) je často citován, ale jde jen o jednu z možných verzí popisu šíření schémat v populaci, která předpokládá proporcionální výběr rodičovských jedinců pro rekombinaci a dostatečně velkou populaci jedinců. V uvedeném vztahu dominuje fenomén rozbíjení schémat, obecně však platí, že křížením rodičů schémata nejen zanikají, ale také vznikají. Jednou z možností, jak snížit pravděpodobnost rozbíjení částečných řešení, je použití genetického operátoru inverze, který zrcadlově přeskupuje proměnné řetězce v náhodně vybraném segmentu řetězce, což řeší také problém Hammingovy bariéry. V současné době je klasická teorie schémat podrobována kritické analýze a hledají se nové obecnější modely činnosti SGA.

Účinnost evolučního procesu závisí na detekci, spojování, filtraci a expanzi stavebních bloků, které jsou částí optimálního řešení. Předpokládá to identifikovat závislosti a nezávislosti mezi proměnnými (linkage learning). Standardní genetické algoritmy zachycují tyto vztahy implicitně selekcí slibných rodičovských jedinců a jejich rekombinací. V genetickém algoritmu není uchována žádná explicitní informace, která skupina proměnných a jak významně přispívá k účelové funkci slibných řešení. Proces rekombinace realizovaný křížením je náhodný a může porušit mnohé z vazeb mezi proměnnými. Hlavní problém je v tom, že rekombinační proces je odstíněn od znalosti řešeného problému a jeho potenciální dekompozice.

Dílcí překonání těchto problémů umožnilo zavedení messy-genetických algoritmů (messy-GA) [Gold99] s proměnnou délkou chromozómů a explicitní indexací genů. Pro standardní GA a messy-GA však zůstává nevyřešen problém nastavení četností aktivace jednotlivých genetických operátorů – předpokládá znalost experta pro daný problém. Tato expertní znalost je často využívána při návrhu fuzzy regulátoru pro nastavování výše uvedených parametrů na základě statistik odvozených z aktuální populace. Výsledky získané genetickými algoritmy s tímto typem řízení parametrů však nejsou jednoznačné.

Jak už bylo zmíněno v první kapitole, principiálním řešením uvedených problémů vyskytujících se při návrhu a ladění SGA je zavedení nové pokročilé koncepce evolučních algoritmů označovaných jako EDA algoritmy (Estimation Distribution Algorithm) nebo jako PMBGA algoritmy (Probabilistic Model Building Genetic Algorithms) [Pel99c], [Larra02], které využívají grafové pravděpodobnostní modely.

## 2.4 EDA ALGORITMY

Jejich činnost je založená na teorii pravděpodobnosti – používají statistickou informaci obsaženou v populaci nadějných řešení/jedinců k detekci (ne)závislostí mezi proměnnými řešení. Novým znakem tohoto přístupu je globální pohled na celou populaci při tvorbě pravděpodobnostního modelu pro daný problém. Noví jedinci jsou generováni podle odhadnutého pravděpodobnostního rozdělení. Na rozdíl od standardních genetických algoritmů jsou EDA algoritmy schopny explicitně vyjádřit jednoduché i vícenásobné závislosti mezi proměnnými a tedy detekovat stavební bloky optimálního řešení, což umožňuje efektivně optimalizovat složité nelineární a klamné úlohy.

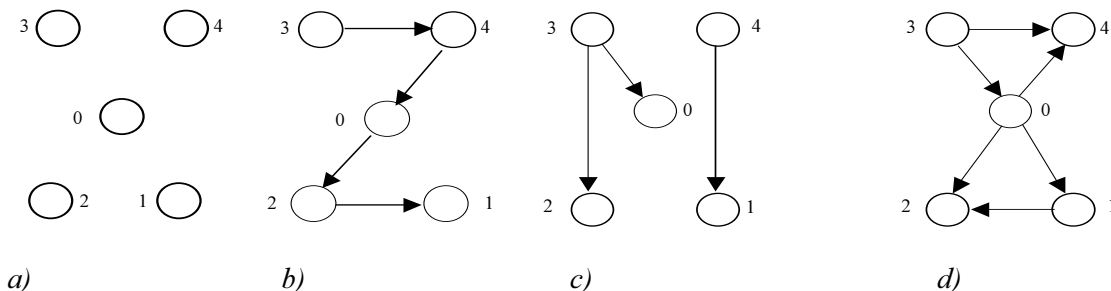
Činnost EDA algoritmu lze popsat následujícím pseudokódem:

- (1) Nastav  $t=0$ , náhodně generuj počáteční populaci  $D(0)$  s mohutností  $N$ ,
- (2) Vyber množinu  $D^s(t)$  slibných jedinců z populace  $D(t)$  s mohutností  $M \leq N$ ,
- (3) Proveď odhad pravděpodobnostního rozložení vybraných jedinců  $p(X | D^s(t))$ ,
- (4) Generuj množinu potomků  $O(t)$  s mohutností  $M$  (vzorkováním získaného rozložení),
- (5) Vytvoř novou populaci  $D(t+1)$  nahrazením části populace  $D(t)$  jedinci z  $O(t)$ ,
- (6) Nastav  $t \leftarrow t+1$ ,
- (7) Pokud není splněna podmínka pro ukončení algoritmu, jdi na (2).

Fundamentálním problémem je odhad pravděpodobnostního rozložení  $p(X|D^s(t))$ . Z teorie je znám vztah pro sdružené rozložení pravděpodobnosti náhodného vektoru pomocí lokálních pravděpodobnostních rozložení s využitím podmíněných pravděpodobností

$$p(X) = \prod_{i=0}^{n-1} p(X_i | X_0, X_1, \dots, X_{i-1}) . \quad (4)$$

Složitost pravděpodobnostního modelu závisí na předpokládané interakci mezi proměnnými. Tyto závislosti/nezávislosti jsou modelovány grafovými strukturami. Uzly grafu odpovídají proměnným vektoru  $X$ , orientované hrany modelují závislost/kausalitu mezi proměnnou a rodičovskými proměnnými. Na obr. 2 jsou grafové modely používané pro EDA algoritmy různé složitosti s příslušným typem faktorizace.



- a)  $p(X) = p(X_3) p(X_0) p(X_2) p(X_4) p(X_1)$ ,  
 b)  $p(X) = p(X_3) p(X_4 | X_3) p(X_0 | X_4) p(X_2 | X_0) p(X_1 | X_2)$   
 c)  $p(X) = p(X_3) p(X_0 | X_3) p(X_2 | X_3) p(X_4) p(X_1 | X_4)$   
 d)  $p(X) = p(X_3) p(X_0 | X_3) p(X_4 | X_3, X_0) p(X_1 | X_0) p(X_2 | X_1, X_0)$

Obr. 2 Grafické modely a sdružené rozložení pravděpodobnosti pro algoritmy:  
 a) UMDA, b) MIMIC, c) BMDA, d) BOA

První evoluční algoritmy založené na jednoduchých pravděpodobnostních modelech znázorněných na obr. 2a, 2b, 2c vznikly asi před pěti lety.

Nejjednodušší model předpokládá, že všechny proměnné jsou vzájemně nezávislé a sdružené pravděpodobnostní rozložení je dáno součinem marginálních pravděpodobností. S tímto modelem pracuje algoritmus PBIL (Population Based Incremental Learning) [Balu94], UMDA (Univariate Marginal Distribution Algorithm) [Mueh96]. Algoritmus MIMIC (Mutual Information Maximizing for Input Clustering) [Etxe99], [Larr99], [Balu97] a BMDA (Bivariate Marginal Distribution Algorithm) [Pel98a] pracují s grafem závislosti (dependency graphs) a umožňují modelovat

podvojně závislosti. FDA algoritmus (Factorized Distribution Algorithm) využívá grafový pravděpodobnostní model specifikovaný expertem [Mueh98] – je hlavně určen pro aditivně rozložitelné problémy.

Nejobecnějším pravděpodobnostním modelem je Bayesovská síť  $(B, B_p)$ , kde  $B$  je struktura sítě (obr. 2d) a  $B_p$  parametry sítě specifikující sdružené pravděpodobnostní rozložení. Bayesovská síť je obecně používaná pro modelování multinomických dat s diskrétními i spojitými proměnnými [Brad00], [Gelm98], [Gott99]. Umožňuje reprezentovat vícenásobné pravděpodobnostní interakce mezi proměnnými. Necht' pro každou proměnnou  $X_i$  je  $\Pi_{X_i} \subseteq \{X_0, X_1, \dots, X_{i-1}\}$  podmnožina (rodičovských) proměnných, které jí ovlivňují a přitom proměnné  $X_0, X_1, \dots, X_{i-1}$  jsou podmíněně nezávislé, pak sdružené pravděpodobnostní rozložení náhodného vektoru  $X$  lze vyjádřit vztahem

$$p(X) = \prod_{i=0}^{n-1} p(X_i | \Pi_{X_i}). \quad (5)$$

Vztah (5) lze interpretovat jako zakódování struktury Bayesovské sítě  $B$  reprezentující podmíněné nezávislosti mezi proměnnými. Bayesovská síť na obr. 2d reprezentuje např. nezávislost proměnné  $X_4$  na proměnných  $X_1, X_2$  (6) nebo nezávislost proměnné  $X_1$  na proměnných  $X_2, X_3, X_4$  (7).

$$p(X_4 | X_0, X_1, X_2, X_3) = p(X_4 | X_3, X_0), \quad (6)$$

$$p(X_1 | X_0, X_2, X_3, X_4) = p(X_1 | X_0). \quad (7)$$

Ilustrace činnosti Bayesovského evolučního optimalizačního algoritmu je na obr. 3. Klíčovou operací je konstrukce/indukce grafu Bayesovské sítě z populace slibných jedinců/řešení  $D^s(t)$  a výpočet parametrů  $B_p$ . Jde o proces učení Bayesovské sítě, který se opakuje každou generací evolučního procesu. V první fázi učení jde o určení struktury Bayesovské sítě, která reflektuje (ne)závislosti uzlů/proměnných.

Ve druhé fázi navazuje proces učení parametrů – hodnot podmíněných pravděpodobností s ohledem na strukturu sítě  $B$ . Parametry jsou reprezentovány množinou tabulek  $CPT_0$ - $CPT_{n-1}$ , specifikujících podmíněné pravděpodobnosti pro každou proměnnou  $X_i=x_i$  (uzel sítě) pro všechny instance  $\pi_{X_i}$  jejich rodičovských proměnných.

Pro nalezení struktury  $B$  lze použít stochastické metody, genetické algoritmy a rychlé heuristiky. K ohodnocení kvality  $B$  lze použít metriky s různou složitostí. K nejpoužívanějším patří *MDL* metrika (*Minimal Length Description*) a *BD* metrika (*Bayes-Dirichletova metrika*) [Larr99], [Heck95], [Frid02]. V EBNA algoritmu (*Estimation of Bayes Network Algorithm*) [Etxe99] je použita *BIC* metrika (*Bayesian Information Criterion*).

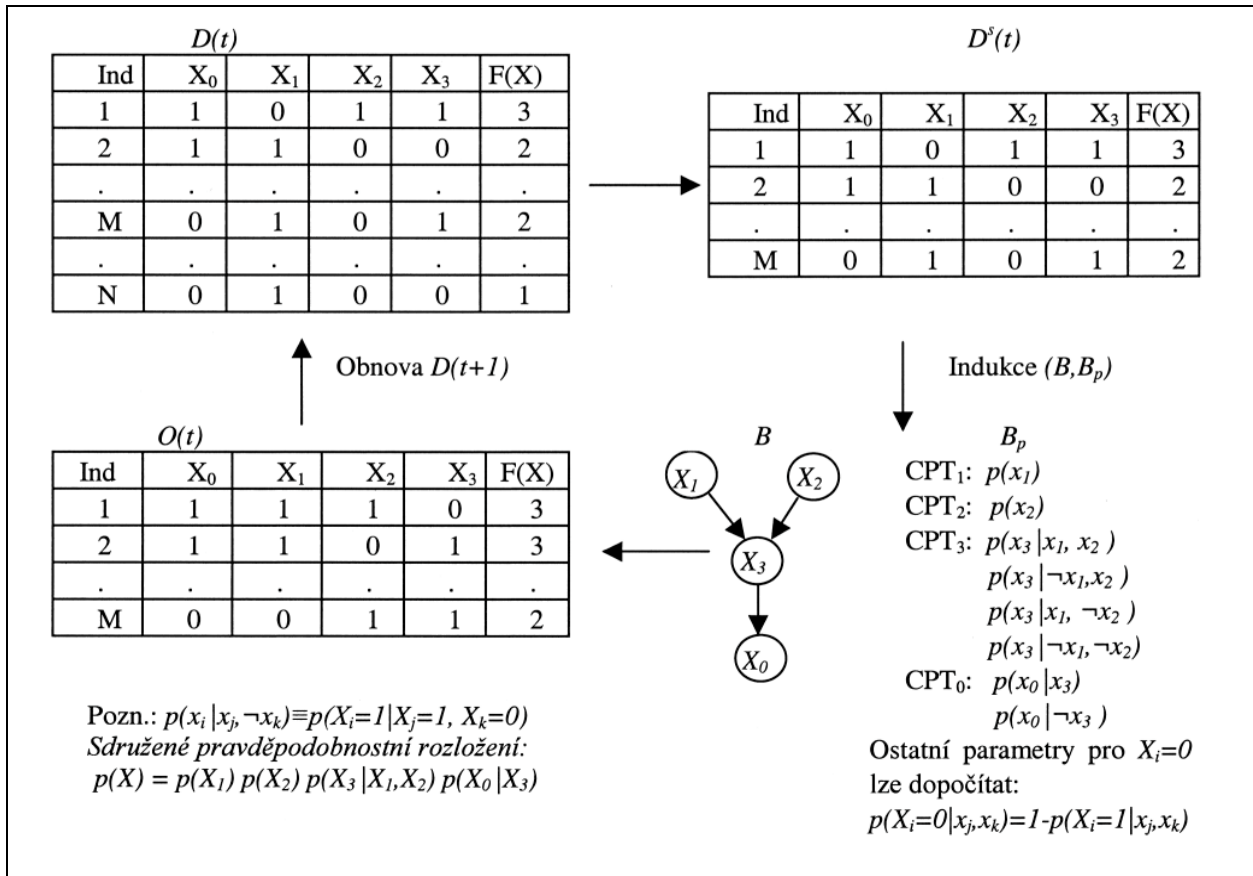
Použití *MDL* metriky vychází z filozofického přístupu nazývaného Occamova břitva/pravidlo (Occam razor), které preferuje ze soupeřících teorií nejjednodušší z nich. V kontextu teorie kódování jde o nalezení optimálního modelu (kodéru) pro kompresi dat  $D$  ( $D \equiv D^s$ ). Optimální model minimalizuje celkovou délku kódu složeného z kódu dat a kódu modelu. Celkovou délku kódu (v bitech) lze vyjádřit celkovou sumou

$$L(B, B_p, D) = L(B) + L(B_p) + L(D/B), \quad (8)$$

přičemž  $L(B)$  je délka zakódování struktury grafu,  $L(B_p)$  je délka zakódování CPT tabulek a  $L(D/B)$  je zakódování dat pro danou strukturu grafu  $B$ . Výpočet délky kódu prvních dvou položek je jednoduché. Zakódování dat  $D$  pro známou síť  $B$  vychází z Huffmanova kódování.

$$L(D/B) = M \sum_{i=0}^{n-1} H(X_i | \Pi_{X_i}), \quad (9)$$

kde  $H(X_i | \Pi_{X_i})$  je podmíněná entropie,  $M$  je počet jedinců populace. Výhodou této metriky je, že není třeba explicitně stanovit horní mez složitosti modelu. Nevýhodou je, že nelze použít apriorní informace o řešeném problému.



Obr. 3 Ilustrace Bayesovského evolučního algoritmu

Při použití Bayes-Dirichletovy (BD) metriky (odvozené z metod Bayesovské statistiky) lze zahrnout do procesu konstrukce  $B$  sítě apriorní informace o pravděpodobnostním rozložení struktury a parametrů sítě. Aposteriorní pravděpodobnost Bayesovské sítě  $B$  pro data  $D$  (zde interpretujeme  $D \equiv D^s$ ) lze stanovit použitím Bayesova teorému

$$p(B|D) = \frac{p(D, B)}{p(D)} = \frac{p(D|B)p(B)}{p(D)}, \quad (10)$$

kde  $p(B|D)$  je aposteriorní pravděpodobnost sítě  $B$  podmíněná daty  $D$ ,  
 $p(B)$  je apriorní pravděpodobnost sítě  $B$ ,  
 $p(D|B)$  je pravděpodobnost dat  $D$  pro síť  $B$  (marginální věrohodnostní funkce),  
 $p(D)$  představuje normativní koeficient.

Čím větší je hodnota  $p(B|D)$ , tím je síť  $B$  věrohodnějším modelem dat  $D$ . Při srovnávání různých sítí pro stejnou datovou strukturu (populaci) lze vypustit jmenovatele  $p(D)$  v rovnici (10)

$$p(B|D) \propto p(D|B) p(B). \quad (11)$$

V případě, že všechny sítě jsou stejně pravděpodobné (není k dispozici znalost experta), platí

$$p(B|D) \propto p(D|B). \quad (12)$$

Nalezení sítě  $B$  s maximální aposteriorní pravděpodobností (věrohodností) je pak ekvivalentní nalezení sítě  $B$ , která maximalizuje pravděpodobnost dat  $D$ . Obecně výpočet pravděpodobnosti  $p(D, B)$  je velmi nesnadný. David Heckerman [Heck95] odvodil metriku, která vychází kromě jiného z předpokladu, že rozložení lokálních podmíněných pravděpodobností má charakter Dirichletova rozdělení.  $BD$  metrika je pak dána vztahem

$$p(D, B) = p(B) \prod_{i=0}^{n-1} \prod_{\pi_{X_i}} ML(p(X_i | \pi_{X_i})), \quad (13)$$

kde  $ML$  je marginální věrohodnostní funkce ( $\pi_{X_i}$  je množina instancí  $\Pi_{X_i}$ ). Podrobnější vztah je uveden v článku A\_4 a A\_3. Nevýhodou klasického přístupu konstrukce Bayesovské sítě je nutnost specifikace maximální složitosti sítě – většinou dané maximálním počtem rodičovských proměnných  $k$  (jinak řečeno  $k$  určuje maximální počet vstupních hran každého uzlu sítě  $B$ ). Možným řešením tohoto problému je využít  $p(B)$  pro penalizaci složitosti sítě.

Ve všech navržených verzích BOA algoritmů je pro nalezení sítě  $B$  použit jednoduchý konstruktivní algoritmus, který využívá elementární grafovou operaci – opakované přidávání jedné hrany v aktuální síti  $B$  s maximální hodnotou  $BD$  metriky. Hrany, které vytvářejí cyklus, nejsou akceptovány.

Po nalezení Bayesovské sítě  $B$  lze z populace  $D^s$  stanovit množinu parametrů  $B_p$ . Podmíněné pravděpodobnosti pro jednotlivé proměnné lze stanovit podle definice podmíněné pravděpodobnosti

$$p(X_i | \Pi_{X_i}) = \frac{p(X_i, \Pi_{X_i})}{p(\Pi_{X_i})}. \quad (14)$$

Příklad stanovení položek CPT je uveden v článku A-7.

Populace potomků  $O(t)$  se získává PLS vzorkováním (Probabilistic Logic Sampling). PLS produkuje hodnotu proměnné kvazínáhodně:

$X_i=1$  pokud  $r \leq p(X_i=1 | \Pi_{X_i} = \pi_{X_i})$  jinak  $X_i=0$  ( $r$  je náhodné číslo s rovnoměrným rozložením v intervalu  $[0,1]$ ). Proces vzorkování začíná uzly s marginálním rozdělením (v obr. 3 jsou to uzly  $X_1, X_2$ ) a pokračuje uzly, pro které jsou již definovány rodičovské proměnné. Z takto získané populace potomků  $O(t)$  a předchozí rodičovské populace  $P(t)$  je vytvořena populace  $P(t+1)$  s přihlédnutím k fitness funkci jedinců z obou populací. Celková časová složitost konstrukce sítě  $B$  a nové populace jedinců je dána vztahem  $O(n^2N+n^3)$ .

Konvergence BOA algoritmu a obecně EDA algoritmů závisí na globálním selekčním tlaku a vyváženosti pravděpodobnostního modelu. Model by měl být dostatečně přesný a přitom dostatečně obecný, aby v procesu inference byla také generována nová slibná řešení mimo rámec aktuální populace. Ověřování účinnosti jednotlivých variant EDA algoritmů je tedy stále aktuální. Je málo experimentálních studií, které porovnávají EDA algoritmy se standardními SGA algoritmy a tradičními metodami. Letos se k nim přiřadila kniha: Pedro Larrañaga a Jose Lozano: „Estimation of Distribution Algorithms“ [Larra02].

### 3 ŘEŠENÍ ÚLOH DEKOMPOZICE A ALOKACE

S úlohou dekompozice se lze setkat při řešení rozmanitých úloh inženýrské praxe. Jde o známou koncepci „divide et impera“, kdy složitá rozsáhlá úloha je rozdělena na několik podúloh s menší velikostí a složitostí. Tyto úlohy nejsou nezávislé a je snahou nalézt dekompozici s minimem vzájemných interakcí. Jako příklad lze uvést dekompozici rozsáhlých telekomunikač-

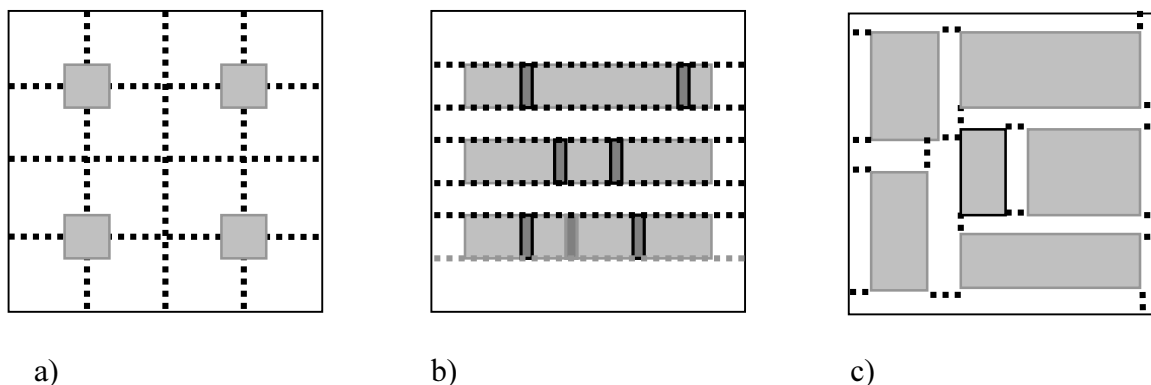
ních sítí, programových produktů, databázových systémů a rozsáhlých číslicových integrovaných obvodů.

Úloha alokace je zmiňována často v kontextu alokace zdrojů (finančních), plánování výroby apod., které jsou řešeny často metodami operačního výzkumu. V této práci je analyzována alokační úloha, která je charakterizována hledáním vhodných pozic pro objekty s definovanou interakcí ve dvou nebo trojrozměrném prostoru v kontextu fyzického návrhu číslicových obvodů.

Návrh číslicových integrovaných obvodů lze charakterizovat pěti základními etapami: specifikací, logickým návrhem (zahrnuje návrh architektury, funkčních bloků a obvodovou realizaci), fyzickým návrhem, výrobou a testováním. Fyzický návrh se standardně člení na tři základní etapy: členění/dekompozici, rozmístování/alokaci a propojování. Charakter těchto etap závisí na použitém návrhovém stylu: desky s plošnými spoji (PCB), VLSI čipy, multičipové moduly (MCM) a hybridní integrované obvody (v dnešní době se již příliš nepoužívají).

### 3.1 ÚLOHA ALOKACE/ROZMÍSTĚNÍ OBVODOVÝCH UZLŮ

Úloha rozmístování je determinována použitým návrhovým stylem; základní typy jsou schematicky zobrazeny na obr. 4. V případě platformy standardních buněk, hradlových polí a PCB jde o pravidelné pozice (šedě označené pravoúhelníky), do kterých jsou obvodové komponenty (buňky) umísťovány. Kritériem optimality v případě PCB a hradlových polí je realizace propojení komponent zabezpečující plnou funkčnost obvodu.



Obr. 4 Návrhové styly pro alokaci a) hradlové pole a PCB, b) standardní buňky, c) funkční bloky

V případě standardních buněk jde o minimalizaci plochy čipu, které se dosahuje minimalizací sumární délky propojení. Rozmístování funkčních bloků je složitější, protože bloky jsou různých rozměrů a tvarů [Drec98]. Pokud navíc není tvar a rozměr fixní (floorplanning problem), jde o velmi obtížně řešitelný problém. Kritérium optimality zahrnuje plochu čipu, poměrový koeficient tvaru (aspect ratio) a kritické zpoždění signálů na spojích.

Existuje řada metod, jak dosáhnout stanovených cílů. Univerzální grafově orientovanou metodou je metoda minimálních řezů (min-cut algorithm) [Breu77]. Použitím sekvence horizontálních a vertikálních řezů jsou obvodové komponenty členěny tak, aby počet spojů v řezu byl minimální. V současné době se používají sofistikovanější varianty s lineární složitostí [Fidu82], [Alpe96]. V případě hradlových polí, PCB desek a standardních buněk s pravidelnými pozicemi obvodových komponent je možné použít kanonickou množinu řádkových a sloupcových řezů (v případě standardních buněk a sloupcových řezů je nutné ošetřit případné překrytí buněk). V případě funkčních bloků není poloha řezu předem specifikována – je nutné splnit omezující podmínku přípustnosti řezu tak, aby podmnožiny komponent vymezené řezem bylo možné umístit do vyme-

zené oblasti čipu. Obtížnost tohoto rozmístování shora dolů vedla ke vzniku duální metody rozmístování zdola nahoru, která umožňuje uplatnit rotace agregovaných funkčních bloků a vzájemné propojení, musí však navíc ošetřit možné překročení poměrového koeficientu tvaru (aspect ratio) [Esbe92], [Sch96].

Velmi populární alokační metodou (používanou např. v návrhovém systému FPGA Xilinx) je simulované žihání, které je založeno na analogii s procesem postupného ochlazováním taveniny. Jde o stochastickou optimalizační metodu, kdy náhodně generovaná rekonfigurace obvodových komponent je akceptována kvazi-náhodně na základě kvality nové konfigurace. Tento přístup zabraňuje uvíznutí v lokálním optimu účelové funkce. Další oblíbenou a názornou metodou je metoda založená na analogii s mechanickým systémem těles vzájemně propojených pružinami (forced-directed placement). Tato metoda je dvoufázová – v první fázi se nalezne relativní pozice komponent na volné ploše a v druhé fázi je nutné ošetřit jejich vzájemné překrytí. V případě regulární struktury pozic je nutná normalizace jejich relativních pozic. Tato metoda byla používána hlavně pro rozmístování diskretních součástek na PCB deskách nebo pro nalezení počátečního rozmístění funkčních bloků VLSI čipu.

Moderní metodiky rozmístování jsou vesměs založeny na sofistikovaných iterativních heuristických metodách někdy v kombinaci s genetickými algoritmy [Drec98], [Mazu99], [Alpe96]. Pro rozmístování standardních buněk jsou používány algoritmy s permutačním zakódováním, podobně jako při řešení úloh obchodního cestujícího (TSP). V případě alokace funkčních bloků je používáno sofistikované zakódování podle složitosti zvolených cílů. V návrhovém systému SAGA [Mazu99] je použito binárního grafu určujícího absolutní pozici pevných bloků.

Kritériem je minimalizace čipu s garantovaným propojením. Systém SAGA je kombinací genetického algoritmu a simulovaného žihání. SA algoritmus řídí velikost populace GA a četnost mutace během evoluční epochy. Komplexnější návrhový systém EXPLORER umožňuje fyzický návrh čipu s flexibilním tvarem funkčních bloků (floorplaning) [Drec98] se čtyřmi optimalizačními kritérii zahrnujícími plochu čipu, jeho poměrový koeficient tvaru, maximální zpoždění propojovacích cest a nahuštění spojů. Řešení je zakódováno pomocí stromové struktury řezů (slicing tree) specifikujících rozvržení funkčních bloků, jejich implementaci, orientaci a kritické zpoždění sítí (spojů).

### 3.1.1 Heuristická metoda limitovaných řezů

V článku [A-1](#) autor této práce prezentoval originální heuristickou metodu rozmístování obvodových uzlů do pravidelných pozic, která je zaměřená zejména na platformu desek s plošnými spoji (PCB) a hradlová pole dle obr. 4a. Pro tento styl lze rozmístovací problém formalizovat následovně: Je zadán hypergraf  $H=(V,E)$  reprezentující obvod s  $n=|V|$  uzly a  $m=|E|$  hranami odpovídající signálovým sítím a poziční regulární graf  $G=(P,D)$  reprezentující  $p=|P|$  pozic a  $d=|D|$  hran s jednotkovou délkou spojující sousední pozice. Cílem je nalezení jednoznačného zobrazení  $f_p: V \rightarrow P$  které minimalizuje účelovou funkci, jež je koncipována tak, že umožňuje nejen minimalizaci celkové délky spojů, ale současně rovnoměrné rozložení spojů. Metoda využívá koncept dynamicky generované posloupnosti kanonických řezů, přičemž hodnota kanonických řezů (počet spojů v řezu) je limitována zdola. Minimalizační proces se zastaví, pokud hodnota horizontálních řezů dosáhne hranice  $L_h$  a hodnota vertikálních řezů hranice  $L_v$ . Nastavení těchto parametrů výrazně ovlivňuje optimalizační proces. Při příliš nízké hodnotě se algoritmus mění na standardní algoritmus minimálních řezů, který minimalizuje pouze sumární délku spojů. Při nastavení obou parametrů na hodnoty vyšší, než je střední hodnota horizontálních, respektive vertikálních řezů, algoritmus produkuje rozmístění sice s rovnoměrným rozložením spojů, ale sumární délka spojů narůstá.

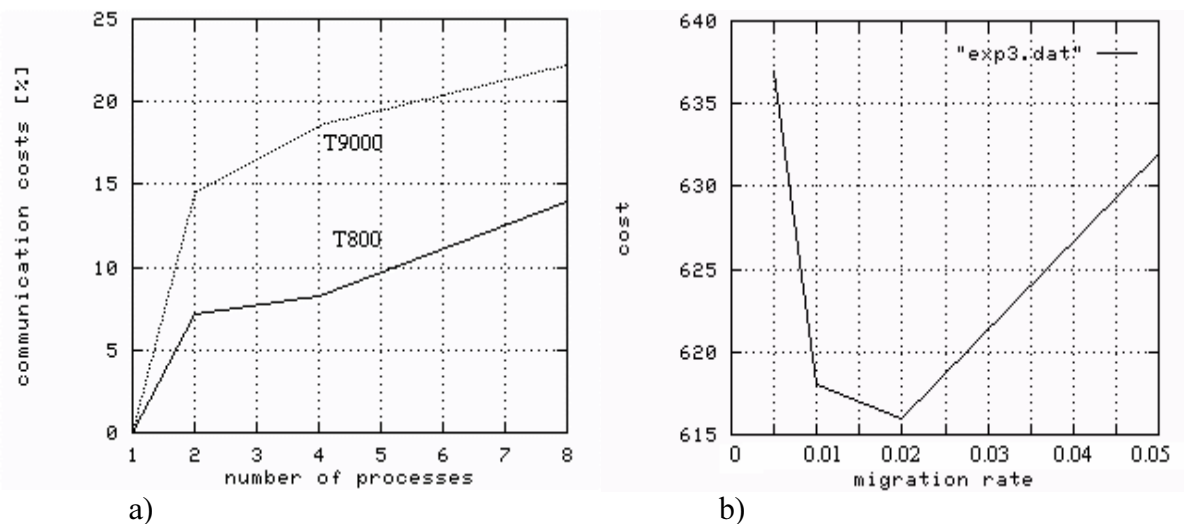


### 3.1.2 Paralelní standardní genetický algoritmus (PGA)

Článek A-2 navazující na dřívější práci autora B-5 a B-6 je experimentální studií využití standardního genetického algoritmu pro řešení rozmíst'ovacího problému pro platformu PCB a hradlových polí. Pro zlepšení konvergence evolučního procesu byl algoritmus obohacen o dva fenomény – fenomén hybridizace a paralelizace.

Účelová funkce je standardní – minimální sumární délky spojů. Délka spojů se aproximuje minimálním poloobvodem *MSP* (Minimal Semi-Perimeter – polovina obvodu minimálního obdélníka obepínajícího obvodové uzly spoje). Tato aproximace se hojně používá, protože je dostatečně přesná a výpočetní složitost je podstatně nižší, než v případě použití minimální kostry *MST* (Minimal Spanning Tree).

Paralelizace byla realizována na platformě transputeru T800 a T9000 s kruhovou topologií procesorů (ring topology) umožňující současný běh až osmi procesů – izolovaných genetických algoritmů se vzájemnou kooperací pouze na bázi migračního operátoru. Hybridizace genetického algoritmu byla realizována přidáním rychlé heuristiky založené na párových záměnách pozic obvodových uzlů, která je aktivována s nastavitelnou četností a intenzitou.



Obr. 5 Experimentální výsledky: a) komunikační režie pro kruhovou topologii, b) účinnost migrace

Byla testována citlivost řešení na řídicích parametrech hybridního algoritmu zahrnujících četnost migrace, četnost mutace, četnost a intenzitu přídavné heuristiky a vliv počtu procesů. Se zvyšujícím se počtem procesů (number of processes) roste rychlost konvergence algoritmu, ale také režijní čas komunikace (communication cost) mezi procesy (viz obr. 5a). Byl potvrzen předpoklad o nutnosti udržování dostatečné úrovně izolace vývoje jednotlivých subpopulací-migrace byla aktivována v optimálním případě jedenkrát za epochu trvající 50 generací (migration rate equals to 0.02), viz obr. 5b. Prokázalo se také, že použití rychlé heuristiky výrazně přispělo k nalezení kvalitnějšího řešení.

***Nutnost nastavování velkého počtu řídicích parametrů a výběru vhodných typů genetických operátorů je velkou nevýhodou všech variant standardních genetických algoritmů. Jak bylo řečeno v první kapitole, principiálním řešením těchto problémů je použití EDA algoritmů. V následujících kapitolách jsou prezentovány Bayesovské evoluční algoritmy, které představují sofistikovaný typ EDA algoritmů navržených pro úlohy dekompozice a alokace.***

### 3.2 BAYESOVSKÉ ALGORITMY V ÚLOZE DEKOMPOZICE HYPERGRAFŮ

Dekompozice systémů, např. telekomunikačních systémů, databázových systémů a VLSI obvodů, patří k častým úlohám členění rozsáhlých systémů na menší celky. Jde o systémy charakterizované množinou entit a vícečlennými relacemi mezi nimi. Takové systémy lze reprezentovat hypergrafy a problém členění lze převést na členění hypergrafů.

Základní úlohu členění hypergrafu lze formalizovat následovně: Je dán hypergraf  $H=(V,E)$  s  $n=|V|$  uzly a  $m=|E|$  hyperhranami, dále jen hranami. Cílem je nalezení členění množiny uzlů  $V$  do disjunktních podmnožin (modulů)  $V_1, V_2$  ( $V=V_1 \cup V_2$ ), které minimalizuje počet externích hran (15) při splnění omezující podmínky na mohutnost obou podmnožin (16). Množina externích hran je označena  $E_{cut}(V_1, V_2)$  a jejich počet  $C_1$ .

Účelová/cenová funkce (metrika)  $C_1$  je definována vztahem:

$$C_1(V_1, V_2) = |E_{cut}(V_1, V_2)| = |\{e \in E / e \cap V_1 \neq \emptyset, e \cap V_2 \neq \emptyset\}| \quad (15)$$

s omezující podmínkou

$$(1-\alpha)n/2 \leq |V_1|, \quad |V_2| \leq (1+\alpha)n/2, \quad \alpha \in (0, 1). \quad (16)$$

Balanční koeficient  $\alpha$  určuje charakter dekompozice. V případě, že koeficient má nulovou hodnotu, jde o bisekci/půlení (bisection) produkující subgrafy/moduly se stejným počtem uzlů, viz obr. 6a, nebo členění s nenulovým koeficientem, viz obr. 6b. Je zřejmé, že nenulový balanční koeficient poskytuje členění s menší hodnotou řezu.

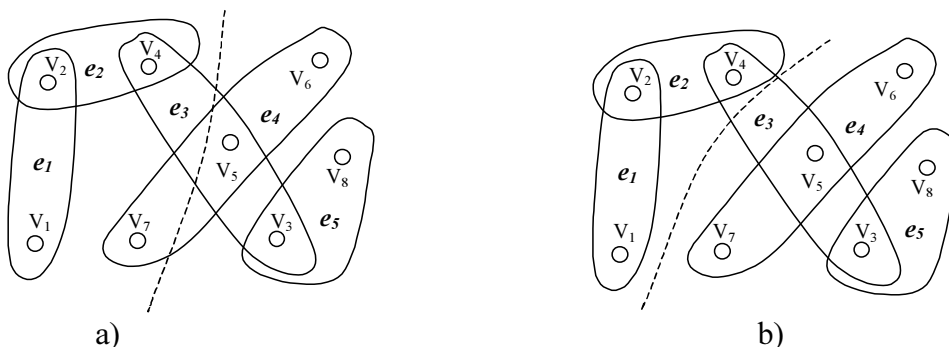
Jinou formou vyjádření rozdílné velikosti podgrafů je prostý rozdíl jejich velikostí  $C_2 = ||V_1| - |V_2||$ . Pro realizaci dekompozice hypergrafu na  $k$  částí je možné použít dva přístupy. Rekurzivní přístup je založen na opakované dekompozici vznikajících modulů/subgrafů na dvě části. Nerekurzivní přístup předpokládá konstrukci počátečního rozdělení hypergrafu na  $k$  částí, které je iterativně je vylepšováno.

Rekurzivní algoritmy se používají častěji. Účelová funkce je definována vztahem

$$C_1(V_1, V_2, \dots, V_k) = |E_{cut}(V_1, V_2, \dots, V_k)| = |\{e \in E / \exists i, j, i \neq j, e \cap V_i \neq \emptyset, e \cap V_j \neq \emptyset\}| \quad (17)$$

s omezující podmínkou

$$(1-\alpha)n/k \leq |V_i| \leq (1+\alpha)n/k, \quad \alpha \in (0, 1), \quad i, j = 1, \dots, k. \quad (18)$$



Obr. 6 Příklad dekompozice hypergrafu a) půlení (s nulovým balančním koeficientem)  $\alpha=0$ ,  $C_1=2$ , b) členění na dvě nestejně části,  $\alpha=1/4$ ,  $C_1=1$

Z hlediska rychlosti dekompozice je nejvýhodnější použít bisekci. Tento přístup však neumožňuje detekovat přirozené shluky v hypergrafu. Řešením je zavedení složené účelové funkce

$$R_c = C_l / F_R(V_1, V_2, \dots, V_k), \quad (19)$$

kde  $C_l$  je počet externích hran, jmenovatel je funkcí velikostí modulů  $V_1$  až  $V_k$ . Jednou z možností je použít součinnou funkci  $F_R(V_1, V_2, \dots, V_k) = |V_1| * |V_2| * \dots * |V_k|$ . Další varianty  $R_c$  jsou uvedeny v článku [Alpe95].

Úlohu dekompozice hypergrafu lze také ilustrovat v kontextu dekompozice obvodu např. takto: je dán obvod  $O = (B, S)$ , kde  $B = \{b_1, b_2, \dots, b_n\}$  je množina obvodových uzlů,  $S = \{s_1, s_2, \dots, s_m\}$  je množina signálových sítí. Cílem je rozdělit obvod  $O$  do  $k$  modulů  $\{M_1, M_2, \dots, M_k\}$  podle zvoleného kritéria. Kromě kritéria  $C_l$  reprezentující počet externích signálových sítí lze použít další dvě kritéria – celkový počet vývodů modulů  $P_c$  a celkový počet spojek  $W_e$ , které bez redundance propojí obvodové uzly jednotlivých sítí. Počet vývodů  $P_c$  lze vyjádřit vztahem:

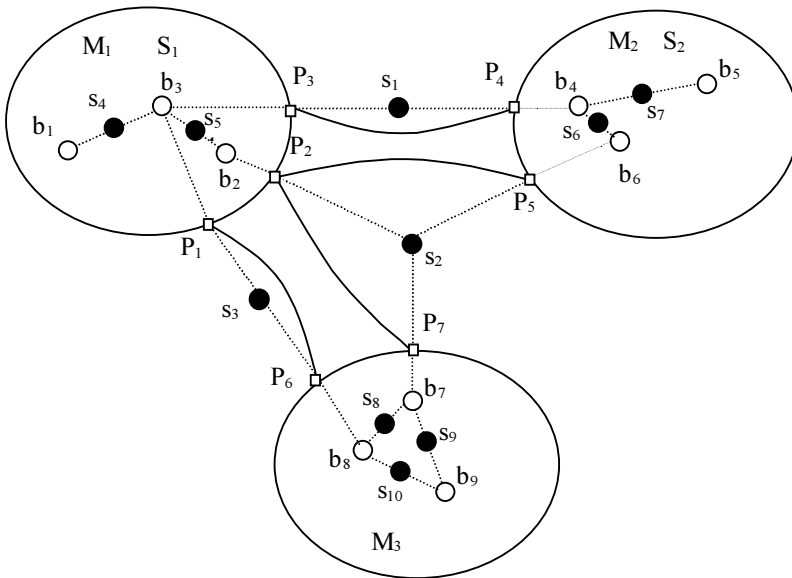
$$P_c = \sum_{s \in S} |\{i \mid 0 < |s \cap M_i| < |s|\}|, \quad i=1, \dots, k, \quad (20)$$

$$W_e = \sum_{s \in S} (|\{i \mid 0 < |s \cap M_i| < |s|\}| - 1), \quad i=1, \dots, k. \quad (21)$$

Mezi uvedenými účelovými funkcemi platí vztah:

$$P_c = W_e + C_l. \quad (22)$$

Na obr. 7 je příklad dekompozice obvodu s devíti obvodovými uzly a deseti sítěmi na tři části. K externím sítím patří podmnožina sítí  $\{s_1, s_2, s_3\}$ , počet externích spojek  $C_l = 3$ , počet vývodů  $P_c = 7$ , počet spojek  $W_e = 4$ .



Obr. 7 Dekompozice obvodu do tří modulů se zobrazením vývodů a spojek externích sítí

Dekompozice hypergrafu (obvodů) se specifikovaným koeficientem balance je NP-úplným problémem. Často používané heuristiky založené na F-M algoritmu (Fiduccia-Matheyses) [Fidu82] a mnohdy i standardní genetické algoritmy neprodukují globálně optimální řešení.

### 3.2.1 Srovnání evolučních algoritmů BOA, BMDA a SGA

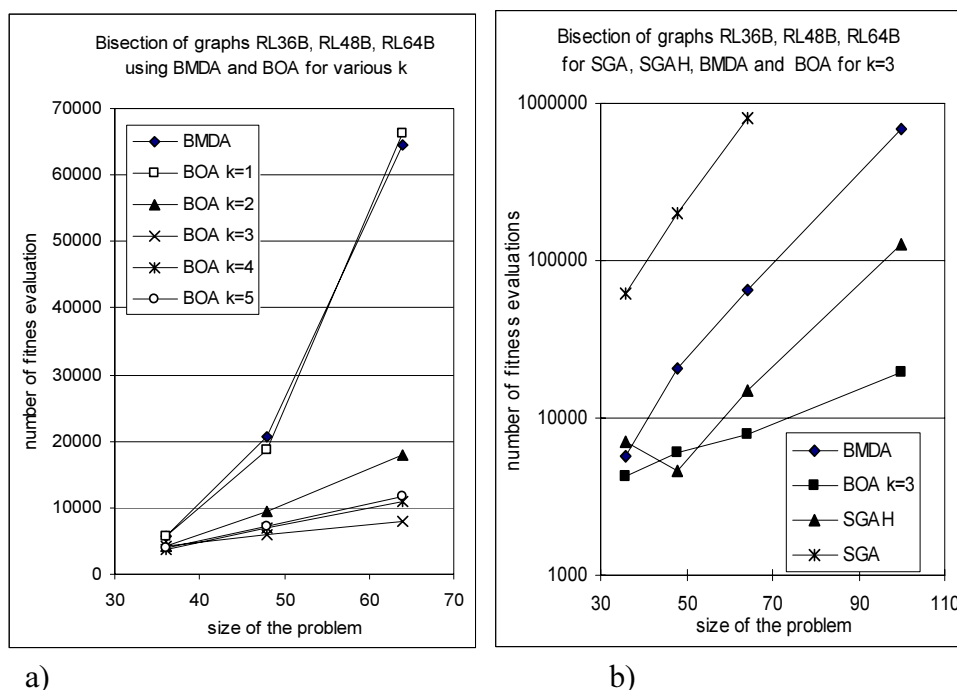
V článku [A-3](#) jsou prezentovány výsledky získané dvěma verzemi EDA algoritmů a třemi variantami standardního genetického algoritmu na úloze členění hypergrafů na dvě a čtyři části (bisekce a kvadrisekce) a šest částí.

Byly navrženy a implementovány dva algoritmy EDA: a) BMDA (Bivariate Marginal Distribution Algorithm) pracující s celočíselnými proměnnými, což umožnilo realizovat nerekurzivní členění hypergrafu na více částí ( $k$ -way partitioning), b) BOA algoritmus na bázi původního algoritmu BOA [Pel99a], který umožňoval řešit jen jednoduché umělé problémy nad binární abecedou pro omezený počet proměnných. Nově navržená koncepce BOA algoritmu má stavebnicovou strukturu a umožňuje explicitní specifikaci řešeného problému na úrovni fenotypu a globálních dat. Pro ohodnocení kvality konstruované Bayesovské sítě byla použita metrika K2.

Byly realizovány tři typy experimentů pro zjištění celkového počtu evaluací (a tedy rychlosti konvergence) pro nalezení předem známého globálního optima:

1. Srovnání BMDA a BOA algoritmu s různým stupněm složitosti pravděpodobnostního modelu BOA (obr. 8a).
2. Srovnání algoritmů BMDA, BOA ( $k=3$ ), SGA a SGAH (hybridní SGA) (obr. 8b).
3. Srovnání BMDA, SGAN (SGA s normalizací genotypu) a SGAH pro nerekurzivní dekompozice na dva, čtyři a šest modulů.

Z experimentů (viz obr. 8) je zřejmé, že nejmenší počet evaluací vyžaduje algoritmus BOA s omezeným stupněm složitosti Bayesovské sítě ( $k=3$ ).



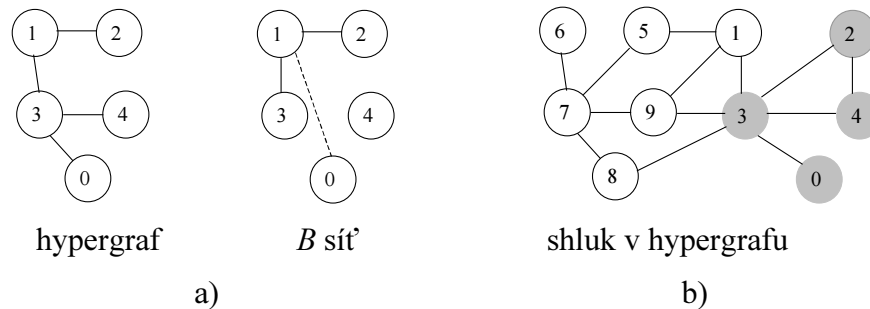
Obr. 8 a) Počet evaluací pro bisekci hypergrafů pro dosažení globálního optima pro BMDA algoritmus a BOA algoritmus s různým stupněm složitosti, b) Počet evaluací pro bisekci grafů pro dosažení globálního optima, při použití algoritmů SGA, SGAH, BMDA a BOA

### 3.2.2 Algoritmus KBOA využívající apriorní informaci

V článku [A-4](#) je popsána originální aplikace využívající při konstrukci sítě  $B$  dodatečných znalostí o řešeném problému, tzv. apriorní informaci, viz vztah (10). Větší apriorní pravděpodobnost má síť, která reflektuje znalost experta. Podle Heckermana [Heck95] lze apriorní pravděpodobnost sítě vyjádřit vztahem:

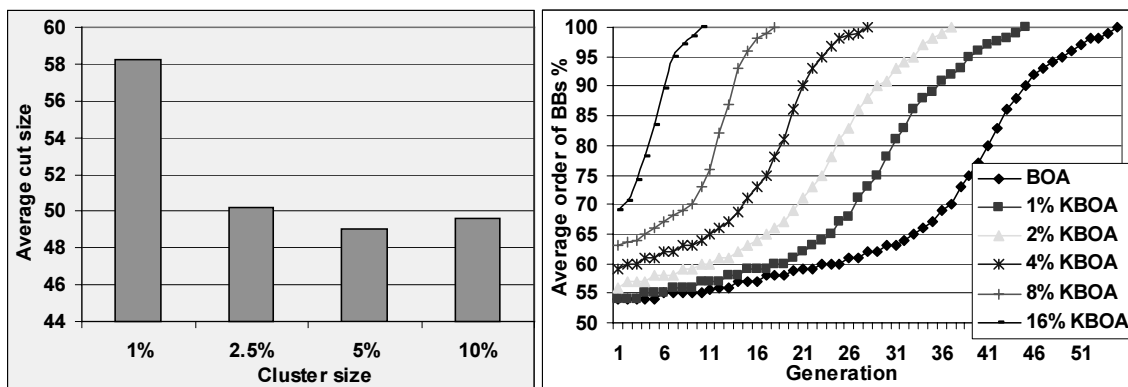
$$p(B) = c\kappa^\delta, \quad (23)$$

kde  $c$  je normalizační konstanta,  $\kappa \in (0, 1]$  je penalizační koeficient a  $\delta$  je symetrická diference mezi sítí  $B$  a apriorní sítí specifikované expertem. V našem případě se vychází z požadavku podobnosti Bayesovské sítě  $B$  a struktury hypergrafu  $(V, E)$ , jehož dekompozice je cílem optimalizačního procesu. Jestliže existuje hrana mezi uzly hypergrafu a tedy existuje závislost mezi oběma uzly, lze předpokládat, že by měla existovat také hrana mezi příslušnými uzly Bayesovské sítě. Při porovnávání dvou variant Bayesovské sítě  $B$  lze tedy využít informaci o počtu nepárových hran vůči členěnému hypergrafu. Preferuje se síť  $B$  s minimálním počtem nepárových hran. Při konstrukci sítě  $B$  postupným přidáváním jedné hrany platí volba  $\delta = 0$ , pokud se přidaná hrana (plná čára) vyskytuje i v hypergrafu. V opačném případě (čárkovaná hrana) je použito nastavení  $\delta = 1$  (viz obr. 9a).



Obr. 9 Uplatnění apriorní informace a) pro síť  $B$ , b) pro počáteční populaci  $D(0)$

Byl realizován také další experiment s využitím apriorní informace o struktuře hypergrafu. V hypergrafu byly vyhledány shluky uzlů (vyznačené šedým odstínem v obr. 9b) zvolené velikosti, které byly injektovány do počáteční populace řešení. Tímto postupem se významně ovlivnila počáteční struktura sítě  $B$ . Optimální velikost shluků činila 5–10 % z celkového počtu uzlů hypergrafu (viz obr. 10a). Použití obou typů apriorních informací vedlo ke zrychlení konvergence řešení (viz obr. 10b).



b)

Obr. 10 Vliv apriorní informace: a) střední hodnota řezů (average cut size) vs. velikost shluků (cluster size), b) evoluce stavebních bloků (average order of BBs)

### 3.2.3 BOA algoritmus pro rozmíst'ování

Techniku dekompozice hypergrafu lze také použít pro řešení rozmíst'ovacího problému. V článku [A-5](#) je popsán rozmíst'ovací algoritmus pro PCB platformu (popřípadě pro standardní buňky) založený na hierarchickém rekurzivním algoritmu bisekce hypergrafu. Ve srovnání se standardním GA pracujícím s permutačním zakódováním řešení poskytuje rozmíst'ovací algoritmus BOA kvalitnější řešení.

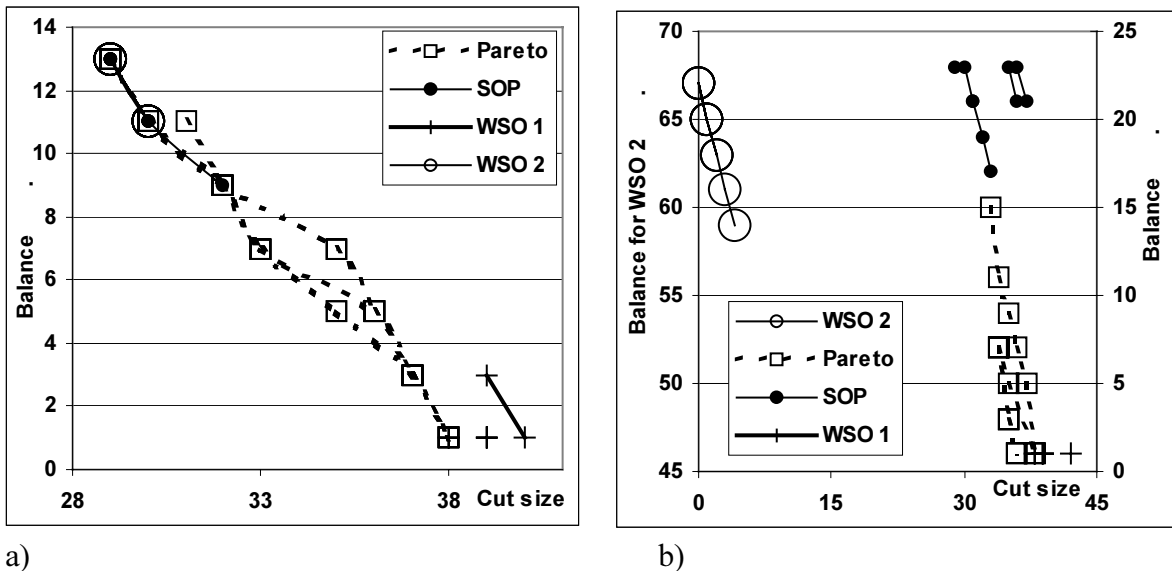
### 3.2.4 Multikriteriální BOA algoritmus pro dekompozici hypergrafů

V předchozích uvedených člancích bylo vždy použito jediné kritérium optimality dekompozice hypergrafů (minimalizace spojů v řezu – hodnoty řezu). V článku [B-12](#) byl poprvé publikován BOA algoritmus pro multikriteriální optimalizaci, který byl testován na známé úloze batohu (knapsack problem).

Výsledky získané v článku [B-12](#) byly pro autora inspirací pro návrh bi-kriteriálního BOA algoritmu pro členění hypergrafů, který je popsán v článku [A-6](#). Jedním kritériem je minimální hodnota řezu, druhým je rozdíl velikostí (balance) obou částí hypergrafu. Pro detekci a ohodnocení nedominovaných a dominovaných řešení byla použita metoda publikovaná v práci [\[Zitz99\]](#).

Byly navrženy a porovnávány čtyři navržené varianty BOA algoritmu:

- Algoritmus Pareto- BOA s vektorovou účelovou funkcí,
- Algoritmus WSO1 BOA se skalarizovanou účelovou funkcí 1. typu,
- Algoritmus WSO2 BOA se skalarizovanou účelovou funkcí 2. typu,
- Algoritmus SOP BOA s jednoduchou účelovou funkcí reprezentující první kritérium. Druhé kritérium je transformováno do omezující podmínky.



Obr. 11 Bisekce hypergrafu: balance podgrafů ( $C_2$ ) versus hodnota řezu (cut size)

a) *IC116/329*,  $N=4000$ , b) *IC67/134*,  $N=2500$

V obr. 11 jsou prezentovány výsledky bisekce dvou hypergrafů reprezentujících reálné obvody (prvé číslo udává počet uzlů, druhé počet hran a  $N$  udává velikost použité populace). Z výsledků řešení je zřejmé, že WSO BOA (weighted sum optimization BOA) algoritmy mají nízkou výpočetní složitost, ale jsou velmi citlivé na použité hodnoty váhových koeficientů a typ členěných grafů. Algoritmus SOP (simple optimization process) produkuje řešení na jednom okraji Pareto-

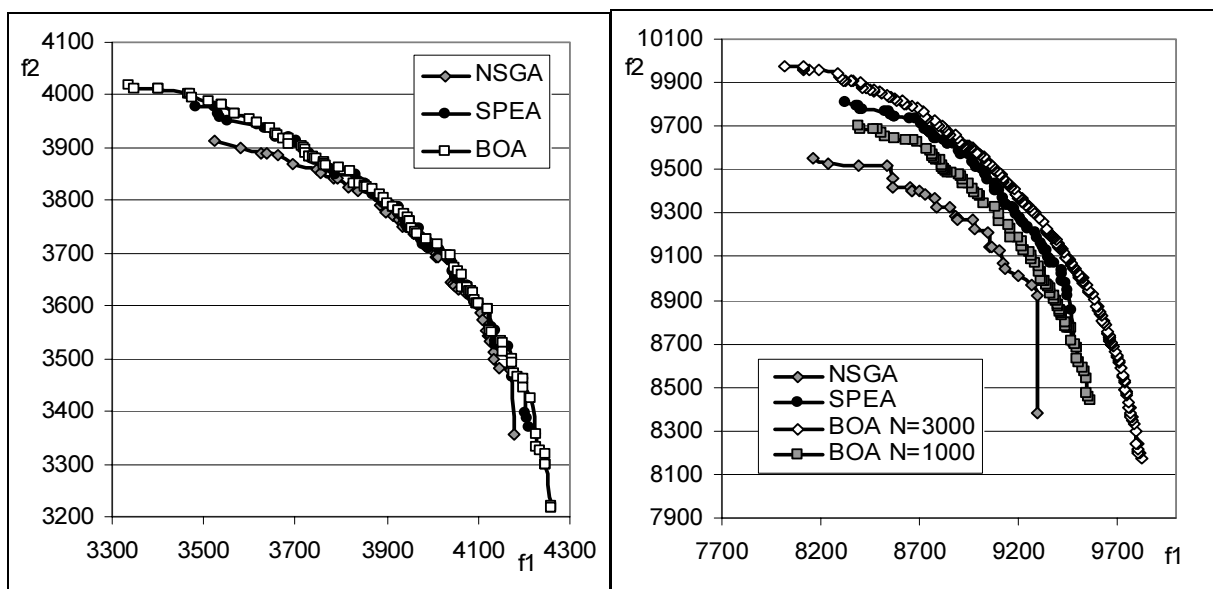
ské hranice, což odráží vliv omezující podmínky. Testy prokázaly, že navržený multikriteriální algoritmus Pareto-BOA produkuje kvalitní řešení rovnoměrně rozložená v Pare-tovské hranici.

### 3.2.5 Teorie a praxe multikriteriálních BOA algoritmů

Systematické zpracování problematiky multikriteriální optimalizace kombinatorických problémů je předmětem článku [A-7](#). Jsou zde popsány často používané metody multikriteriální optimalizace, které nejsou založeny na koncepci Pareto-optimality, a je specifikována propracovanější verze algoritmu Pareto-BOA. Jsou uvedeny základní pravděpodobnostní modely používané v algoritmech EDA a prezentována Bayesovská statistika pracující nad fragmentem populace řešení a fragmentem Bayesovské sítě pro stanovení hodnot podmíněné pravděpodobnosti (položky CPT tabulky) pro jeden uzel Bayesovské sítě.

Algoritmus Pareto-BOA je testován na třech bi-kriteriálních kombinatorických úlohách:

1. Úloze batohu, která navazuje na výsledky publikované v B-12. Byl analyzován vliv velikosti populace a nalezena velikost populace, při které Pareto-BOA algoritmus produkuje Pareto-ovskou hranici plně pokrývající Pareto-ovskou hranici produkovanou algoritmy [Horn94] a [Zitz99] a navíc poskytuje další řešení na jejích okrajích (viz obr. 12).



Obr. 12 Srovnání Pareto-ovských front pro úlohy batohu a) Kn100, N=2000, b) Kn250, N=1000 a N=3000, (N specifikuje velikost populace Pareto-BOA algoritmu)

2. Teoretické úloze Onemax/Xor se známým globálním extrémem, kdy jedním kritériem je počet jedniček v binárním řetězci a druhým počet dvojic sousedních znaků/bitů s různou hodnotou. Pareto-BOA algoritmus našel všechna existující řešení Pareto-ovské hranice (fronty), komparační algoritmus [Horn94] nenašel řešení na jednom jejím okraji.
3. Úloze členění reálných číslicových obvodů. Jde o pozměněnou variantou úlohy publikované v článku [A-6](#). Výsledné Pareto hranice byly konstruovány agregací lokálních hranic získaných v opakovaných nezávislých optimalizačních bězích. Z výsledků je evidentní, že Pareto-BOA algoritmus umožňuje generovat bohatší sortiment řešení než WSO algoritmy se skalarizovanou účelovou funkcí.

### 3.2.6 MBOA dekompoziční algoritmus s poměrovým řezem

Jistou nevýhodou optimalizačních Paretovských metod je jejich algoritmická a výpočetní složitost. Je vždy na zvážení experta, zda zvolí tuto metodu nebo zjednodušené metodiky (skalarizaci účelové funkce, prioritní sekvence účelových funkcí apod.). Ve vybraných aplikacích je možné a účelné agregovat dvě, popřípadě více kritérií do jedné účelové funkce. To je i případ stále aktuální úlohy hierarchické dekompozice rozsáhlých obvodových struktur respektující přirozené shluky obvodových uzlů. Lze vyzvednout dva stejně významné požadavky – nalezení subgrafů/modulů s minimální interakcí (hodnotou řezů) a minimálním balančním koeficientem reflektující rozdílnou velikost modulů. Řešením je zavedení složené účelové funkce  $R_c$  (ratio-cut partitioning) definované podílem hodnoty minimálních řezů a součinu velikostí modulů, viz kap. 3.2.

Byly navrženy tři varianty algoritmu (článek A-8):

1. Rekurzivní algoritmus RRC, který realizuje hierarchickou dekompozici grafů s využitím opakovaného členění na dvě části minimalizující účelovou funkci  $R_c$ .
2. Nerekurzivní algoritmus MRC, který realizuje (paralelní) dekompozici grafu na zvolený počet modulů minimalizující hodnotu  $R_c$  v jenom optimalizačním běhu.
3. Rekurzivní algoritmus RB, který se od RRC algoritmu liší tím, že používá pro dekompozici bisekci, tedy půlení grafů (s nulovým balančním koeficientem).

Byly použity čtyři skupiny testovacích grafů. Regulární grafy různé velikosti a složitosti: graf typu housenka, náhodně generovaný graf s předem definovaným stupněm uzlů grafu a dva reálné grafy reprezentující číslicové obvody (náhodná logika). V případě regulárních grafů a grafu typu housenka, často používaných jako obtížné testovací úlohy, bylo globální optimum známo. Experimentálně bylo ověřeno, že oba algoritmy RRC a MRC produkují řešení srovnatelné kvality a oba našly globální optimum pro regulární grafy a graf typu housenka.

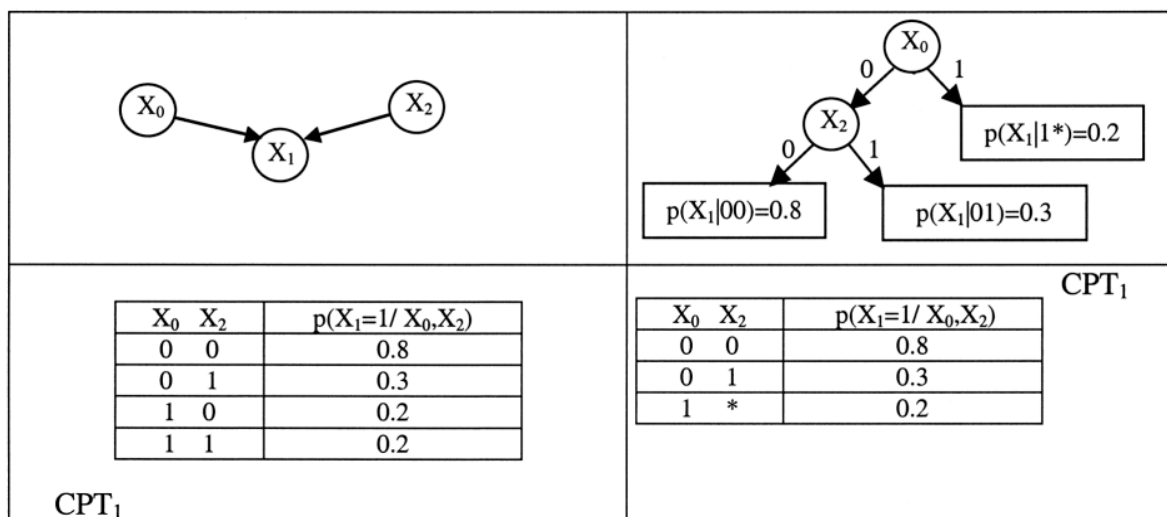
Nevýhodou MRC algoritmu je jeho velká časová složitost, vůči RRC algoritmu až desetinásobná. Algoritmus RB produkoval podstatně horší výsledky vůči RRC i MRC, protože použití bisekce omezuje detekci přirozených shluků grafových struktur.

RRC a RB algoritmy pracují s binárním zakódováním řešení, MRC algoritmus pracuje s řetězci nad celými čísly. To vyžadovalo výraznou modifikaci původního BOA algoritmu pracujícího nad binárními řetězci na pokročilejší verzi MBOA, která také akceptuje řetězce s celočíselnou abecedou.

Pro konstrukci grafové struktury pravděpodobnostního modelu byl použit koncept binárních rozhodovacích grafů (BDD), které reprezentují (ne)závislosti proměnných a podmíněné pravděpodobnosti úsporněji, viz obr. 13. CPT tabulka pro  $i$ -tou proměnnou Bayesovské sítě obsahuje v případě  $k$  rodičovských proměnných  $2^k$  parametrů pro stanovení podmíněné pravděpodobnosti  $p(X_i | \Pi_{X_i})$  přičemž  $k = |\Pi_{X_i}|$ . BDD kóduje pravděpodobnostní model úsporněji a přitom detailněji – umožňuje reprezentovat nezávislost proměnných pro konkrétní kontext/instance. Na obr. 13 je hvězdičkou vyznačena nezávislost proměnné  $X_1$  na proměnné  $X_2$  pro konkrétní instanci proměnné  $X_0=1$ .

Pro každou proměnnou je konstruován jeden BDD graf, jehož listy specifikují podmíněné pravděpodobnosti této proměnné.





a)

b)

Obr. 13 Zakódování pravděpodobnostního rozložení  $p(X) = p(X_1 | X_0, X_2) p(X_0) p(X_2)$  a příslušné CPT tabulky: a) Bayesovská síť, b) BDD graf

V MBOA algoritmu je použit zobecněný BDD model umožňující pracovat s hybridním genotypem řetězců zahrnující binární, celočíselné i spojité parametry – bližší popis je v publikaci [B-16](#) a [B-14](#).

### 3.2.7 MBOA dekompoziční algoritmus s volitelným kritériem

Aktuálnost použití kritérií  $C_e$  (zde  $C_e \equiv C_l$ ),  $W_e$ ,  $P_c$  (viz rovnice 20 až 22) závisí na kontextu řešeného problému – zda jde o dekompozici obvodu pro vícevrstvé propojování, minimalizaci průměrného zpoždění sítě nebo minimalizaci počtu sítí mezi subgrafy/moduly pro účely snadné testovatelnosti.

Na základě realizovaných experimentů (publikovaných v článku [A-9](#)) lze konstatovat, že minimalizační kritérium  $W_e$  je univerzálnější a je vhodné jej použít při minimalizaci průměrného zpoždění v síti a při dekompozici obvodu do subobvodů realizovaných vícevrstvou technologií. Hodnotu  $W_e$  lze tak interpretovat jako počet průchozích otvorů. Pro realizaci experimentů byl použit MBOA algoritmus popsáný v článku [A-8](#) s BDD pravděpodobnostním modelem a byla navržena koncepce paralelizace konstruování BDD stromů.

V článku [A-9](#) je také prezentována idea využití BDD grafu s  $BD$  metrikou nejen jako pravděpodobnostního modelu MBOA algoritmu, ale také pro konstrukci BDD grafů pro reprezentaci jedno a vícevýstupových booleovských funkcí.

## 4 ZÁVĚR

Cílem předkládané habilitační práce bylo prezentovat novou třídu evolučních algoritmů založených na pravděpodobnostních modelech, které by postrádaly typické problémy standardních genetických algoritmů s konvergencí, laděním řídicích parametrů a volbou reprodukčních operátorů. Byly uvedeny teoretické základy nového evolučního paradigmatu a navrženy nové varianty Bayesovského evolučního algoritmu umožňující řešit složité kombinatorické optimalizační úlohy s jedním a více kritérii s využitím apriorních informací o řešeném problému. Účinnost těchto algoritmů byla testována na vybrané skupině kombinatorických úloh dekompozice a alokace systémů, které je možné reprezentovat hypergrafy, popřípadě ohodnocenými hypergrafy. Obě úlohy byly řešeny také v kontextu fyzického návrhu číslicových obvodů, kam nesporně patří úloha členění a rozmístování. Experimentální výsledky potvrzují schopnost algoritmů BOA, KBOA, Pareto-BOA a MBOA řešit složité nelineární úlohy a nalézt globální optimum známých testovaných úloh [Benc98], [Merz00], [Buim96].

Je nesporným faktem, že význam evolučních algoritmů stále roste, rovněž v souvislosti s prudce se rozvíjející oblastí inteligentních výpočtů (Softcomputing). Evoluční algoritmy (EA), fuzzy systémy (FS) a neuronové sítě (NS) samostatně, ale stále více ve vzájemné kooperaci, se stávají efektivním nástrojem při řešení složitých problémů z oblasti informačních technologií (IT). Klasické výpočetní metody včetně distribuovaných systémů často nestačí v reálném čase produkovat přijatelná řešení pro tento typ úloh. Tato skutečnost vedla ke vzniku klasických kooperačních dvojic algoritmů: EA/FS pro nastavování parametrů fuzzy regulátorů a EA/NS pro nastavování parametrů a struktury neuronových sítí. Obrácená kooperace FS/EA využívající fuzzy regulátoru pro nastavení parametrů genetických operátorů je stále více používána, i když výsledky do značné míry závisejí na znalostech experta. Totální kooperace EA/FS/NS principiálně poskytuje velkou výpočetní sílu, vyvážení všech komponent však není snadné. Jestliže použijeme EDA variantu evolučního algoritmu EA v některé z výše zmíněných kooperací, získáme robustnější výpočetní prostředek a celkový počet nastavovaných parametrů bude významně redukován.

Pro rychlé prototypování aplikací EA na bázi Bayesovského optimalizačního algoritmu byl implementován vývojový systém DEBOA B-17, který byl navržen na základě zkušeností prezentovaných v článcích A-3 a A-4. Jeho aplikační možnosti budou dále rozvíjeny, zejména v oblasti multikriteriální optimalizace a paralelního zpracování, viz B-14 až B-16.

Otevřeným problémem Bayesovských evolučních algoritmů zůstává řešení velmi rozsáhlých optimalizačních úloh. Kromě paralelizace jednotlivých etap BOA algoritmu je možná kooperace BOA a SGA algoritmů, což by umožnilo využít předností obou algoritmů. Další možností je návrh BOA algoritmu, který primárně optimalizuje systém produkčních pravidel následně použitých pro vlastní optimalizační proces. Dalším záměrem autora je návrh kooperujícího algoritmu na bázi BOA a PBIL algoritmu (Population Based Incremental Learning Algorithm). Uvedená témata budou předmětem dalšího výzkumu v rámci grantových projektů a diplomových prací.

## 5 LITERATURA

### 5.1 PŘEHLED VYBRANÉ LITERATURY

- [Alpe95] ALPERT, C. J., KAHNG, A. B.: Recent Directions in Netlist Partitioning: a Survey. *Integration: The VLSI Journal*, 1995, vol. 19, pp. 1–81.
- [Alpe96] ALPERT, C. J., HAGEN, W., KAHNG, A. B.: A Hybrid Multilevel/Genetic Approach for Circuit Partitioning. In *Proceedings of ACM SIGDA Physical Design Workshop*. Washington DC, 1996, pp. 100–105.
- [Balu94] BALUJA, S.: *Population-Based Incremental Learning: a Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. Carnegie-Mellon Report, CMU-CS-94-163, 1994.
- [Balu97] BALUJA, S., DAVIES, S.: Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proceedings of the 14th International Conference on Machine Learning*. Morgan Kaufmann, 1997, pp. 30–38.
- [Back96] BACK, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996. ISBN 0-19-509971-0.
- [Benc98] *A benchmark set*. University of California, Los Angeles, VLSI CAD Laboratory. Dostupný na: <http://vlsicad.cs.ucla.edu/~cheese/benchmarks.html>.
- [Brad00] BRADLEZ, P. LOUIS, T. A.: *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall/CRC, 2000. 419 p. ISBN 1-58488488-170-4.
- [Breu77] BREUER, M. A.: Min-cut Placement. *Journal of Design Automation and Fault Tolerant Computing*, 1977, Vol. 1, No. 4, pp. 343–362.
- [Breu82] BREUER, M. A.: *Forced-oriented Placement Algorithm: Experimental Study with IC Benchmarks, Chapter 5 and 6*, pp. 89–162, delivered under personal request.
- [Buim96] BUI, T. N., MOON, B. R.: Genetic Algorithm and Graph Partitioning. *IEEE Transactions on Computers*, July 1996, Vol. 45, No.7, pp. 841–855.
- [Dasg98] DASGUPTA, D., MICHALEWICZ, Z.: *Evolutionary Algorithms in Engineering Application*. Berlin: Springer Verlag, 1998. 554 p. ISBN 3-540-62021-4.
- [Drec98] DRECHSLER, R.: *Evolutionary algorithms for VLSI CAD*. London: Kluwer Academic Publishers, 1998. ISBN 0-7923-8168-8.
- [Dudo79] DUDORKIN, J.: *Operační analýza*. Praha, ČVUT, 1979. 342 s.
- [Esbe92] ESBENSEN, H.: A genetic Algorithm for Macro Cell Placement. In *Proceedings of the European Design Automation Conference*, September 1992, pp. 52–57.
- [Etxe99] ETXEBERRIA, R., LARRAÑAGA, P.: *Global optimization using Bayesian networks. II. Symposium on Artificial Intelligence, CIMA99*, 1999.
- [Fide92] The FIDE (Fuzzy Inference Development Environment). User and Reference Manual. Apronix, 1992.
- [Fidu82] FIDUCCIA, C. M., MATTHEYSES, R.: A Linear Time Heuristic for Improving Network Partitions. In *Proc. ACM/IEEE DAC*, 1982, pp.175–181.
- [Foge00] FOGEL, D.: *Evolutionary computation*. New York: IEEE Press, 2000. 270 p. ISBN 0-7803-5379-X.
- [Frie98] FRIEDMAN, N., GOLDSZMIDT, M.: Learning Bayesian Networks with Local Structure. In JORDAN (ed.). *Learning and Inference in Graphical Models*, 1998.

- [Gelm98] GELMAN, A. et al.: *Bayesian Data Analysis*. London: Chapman & Hall, 1998. ISBN 0-412-03991-5.
- [Gold89] GOLDBERG, E., D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [Gold99] GOLDBERG, E. D. et al.: Messy Genetic Algorithms Revisited. Nonuniform Size and Scale. *Complex system*, 1990, Vol. 4, pp. 415–444.
- [Gott99] GOTTVALD, A.: Bayesian Evolutionary optimization. *5th International Mendel Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets, June 24-26, 1999, Brno, Czech Republic*, pp. 30–35. ISBN 80-214-1131-7.
- [Heck95] HECKERMAN, D., GEIGER, D., CHICKERING, M.: *Learning Bayesian Networks: The combination of Knowledge and Statistical Data (Technical Report MSR-TR-94-09)*. Redmont, WA: Microsoft Research, 1995, pp. 1–53.
- [Holl75] HOLLAND, J., H.: *Adaptation in Natural and Artificial System*. Ann Arbor, MI: The University of Michigan Press, 1975.
- [Horn94] HORN, J., NAFPLIOTIS, N., GOLDBERG, D.E.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Vol. 1*. Piscataway, New Jersey: IEEE Service Center, June 1994, pp. 82–87.
- [Kvas00] KVASNIČKA, V., POSPÍCHAL, J., TIŇO, P.: *Evolučné algoritmy*. Bratislava: STU, 2000. 215 s. ISBN 80-227-1377-5.
- [Kva00a] KVASNIČKA, V.: Umelá evolúcia. In *Seminár kognitívne vedy: CogSci2000*. Bratislava: katedra matematiky CHTF STU, 2000, pp. 1–17.
- [Larr99] LARRAÑAGA, P. et al.: A review of the cooperation between evolutionary computation and probabilistic graphical models. *Artificial Intel Review*, 1999. Preprint.
- [Larra02] LARRAÑAGA, P., LOZANO, J. A. (ed.): *Estimation of Distribution Algorithms*. London: Kluwer Academic Publishers, 2002. ISBN 0-7923-7466-5.
- [Leng90] LENGAUER, T.: *Combinatorial Algorithms for Integrated Circuit Layout*. John Wiley & Sons, 1990. 697 p. ISBN 0-471-92838-0.
- [Mari01] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol.: *Umělá inteligence (3)*. Praha: Academia, 2001. ISBN 80-200-0472-6.
- [Mazu99] MAZUMDER, P.: *Genetic Algorithms for VLSI Design, Layout & Test Automation*. Prentice Hall, 1999. 338 p. ISBN 0-13-011566-5.
- [Merz00] MERZ P., FREISLEBEN, B.: Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning. *Evolutionary Computation*, 2000, Vol. 8, No. 1, pp. 61–91.
- [Mueh98] MUEHLENBEIN, H., RODRIGUEZ A. O.: Schemata Distributions and Graphical Models in Evolutionary Optimization. In *GMD Forschungs Zentrum Informationstechnik, 53754-St. Augustin*, 1998, pp.1–21.
- [Mueh96] MUEHLENBEIN, H., PAAS, G.: From recombination of genes to the estimation of distributions, I. Binary parameters. In *Parallel Problem Solving from Nature, PPSN IV*, pp. 178–187.
- [Micha00] MICHALEWICZ, Z., FOGEL, B., D.: *How to Solve It: Modern Heuristics*. Berlin: Springer-Verlag, 2000. 467 p.
- [Ošme02] OŠMERA P.: *Genetické algoritmy a jejich aplikace*. Brno: FSI VUT, 2002. 86 s. Habilitační práce.

- [Pel98a] PELIKAN M., MUEHLENBEIN, H.: Marginal Distribution in Evolutionary Algorithms. In *4<sup>th</sup> International Mendel Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets, June 24–26, 1998, Brno, Czech Republic*, pp. 91–95. ISBN 80-214-1199-6.
- [Pel99b] PELIKAN, M., GOLDBERG, D. E., CANTÚ-PAZ, E.: *Linkage Problem, Distribution Estimation, and Bayesian Networks*. Illigal Report No. 98013. February 1999, pp. 1–25.
- [Pel99a] PELIKAN, M.: *A Simple Implementation of Bayesian Optimization Algorithm in C++ (Version 1.0)*. Illigal Report 99011. February 1999, pp. 1–16.
- [Pel99c] PELIKAN, M., GOLDBERG, D. E., LOBO, F.: A Survey of Optimization by Building and Using Probabilistic Model. Illigal Report 99018. September 1999, pp. 1–12.
- [Pel00a] PELIKAN, M., GOLDBERG, E., SASTRY, K.: Bayesian Optimization Algorithm, Decision Graphs, and Occams Razor. Illigal Report No. 2000020. May 2000, pp.1–24.
- [Pel00b] PELIKAN, M., GOLDBERG, D. E., & CANTÚ-PAZ, E.: *Bayesian Optimization Algorithm, Population Sizing, and Time to Convergence*. Illigal Report 2000001. January 2000, pp. 1–13.
- [Peli02] PELIKAN, M.: *Bayesian optimization algorithm: From single level to hierarchy*. Urbana, IL: University of Illinois at Urbana-Champaign, 2002. Ph.D. thesis. Také jako IlliGAL Report No. 2002023, 2002.
- [Sadi96] SADIC, M., SAIT, HABIB, YOUSSEF: *Iterative Computer Algorithms with Application in Engineering*. IEEE Computer Society. ISBN 0-7695-0100-1.
- [Sasa96] SASAO T., FUJITA, M. (ed.): *Representations of Discrete Functions*. London: Kluwer Academic Publisher, 1996.
- [Shah91] SHAHOOKAR, K., MAZUMDER, P.: VLSI Cell Placement Techniques. *ACM Computing Surveys*, June 1991, Vol. 23, No. 2, pp. 195–220.
- [Schn96] SCHNECKE, V.: *Hybrid Genetic Algorithm for Solving Constrained Packing and Placement Problems*. University of Osnabrueck, 1996, pp. 1–186. PhD thesis.
- [Solv35] *Premuim Solver V35*. Dostupné na: <http://evonet.dcs.napier.ac.uk/evoweb/resources/software/software174.html>.
- [Stro99] STROOBANDT, D.: Pin Count Prediction in Ratio Cut Partitioning for VLSI and ULSI. In BAYOUMI, M. A. (ed.). *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 1999, pp. VI/262–VI/265.
- [Weic91] WEI, Y.-C., CHENG, C.-K.: Ratio Cut Partitioning for Hierarchical design. *IEEE Trans. Computer Aided Design Integrated Circuit & System*, July 1991, Vol.10, No.7, pp. 911–921.
- [Zitz99] ZITZLER, E.: *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Zurich: Swiss Federal Institute of Technology (ETH), November 1999. PhD thesis.

## 5.2 PŘEHLED ČLÁNKŮ TVOŘÍCÍCH PŘEDLOŽENOU HABILITAČNÍ PRÁCI

- A-1 SCHWARZ, J.: The optimum placement based on a method of limited cut. *Computer and Artificial Intelligence*, 1986, Vol. 5, No. 4, pp. 375–384.
- A-2 SCHWARZ, J.: Experimental study on parallel genetic algorithm for placement optimalization. In *Proceedings of the 3rd International Mendel Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets, Mendel '97*. Technical University of Brno, 1997, pp. 148–153. ISBN 80-214-0884-7.

- A-3 SCHWARZ, J., OČENÁŠEK, J.: Experimental study: Hypergraph partitioning based on the simple and advanced genetic algorithm BMDA and BOA. In *Proceedings of the 5th International Conference on Soft Computing, Mendel '99*. Technical University of Brno, 1999, pp. 124–130. ISBN 80-214-1131-7 (70/30 %).
- A-4 SCHWARZ, J., OČENÁŠEK, J.: A problem knowledge-based evolutionary algorithm KBOA for hypergraph bisectioning. In *Proceedings of the Fourth Joint Conference on Knowledge-Based Software Engineering*. Brno, 2000, pp. 51–58. ISBN 1-58603-060-4 (70/30 %).
- A-5 SCHWARZ, J., OČENÁŠEK, J.: Partitioning-oriented placement using advanced genetic algorithm BOA. In *Proceedings of the 6th International Conference on Soft Computing, Mendel 2000*. Technical University of Brno, 2000, pp. 145–150. ISBN 80-214-1609-2 (70/30 %).
- A-6 SCHWARZ, J., OČENÁŠEK, J.: Evolutionary Multiobjective Bayesian Optimization Algorithm: Experimental Study. In *Proceedings of the 35th Spring International Conference MOSIS '01, Vol. 1*. Hradec nad Moravicí: MARQ, 2001, pp. 101–108. ISBN 80-85988-57-7 (70/30 %).
- A-7 SCHWARZ, J., OČENÁŠEK, J.: Multiobjective Bayesian Optimization Algorithm for Combinatorial Problems: Theory and Practice. *Neural Network World*, roč. 11, č. 5, pp. 423–441. ISSN 1210-0552 (80/20 %).
- A-8 SCHWARZ, J., OČENÁŠEK, J.: Ratio cut hypergraph partitioning using BDD based MBOA optimization algorithm. In *Proceedings of the Fifth International Workshop IEEE Design and Diagnostics of Electronic Circuits and Systems*. Brno: University of Technology, Faculty of Information Technology; IEEE Computer Society, 2002, pp. 87–96. ISBN 80-214-2094-4 (70/30 %).
- A-9 SCHWARZ, J., OČENÁŠEK, J.: Bayes-Dirichlet BDD as a probabilistic model for logic function and evolutionary circuit decomposer. In *Proceedings of the 8th International Mendel Conference on Soft Computing, Brno, Mendel 2002*. Technical University of Brno, 2002, pp. 117–124. ISBN 80-214-2135-5 (60/40 %).

### 5.3 SEZNAM VYBRANÝCH PRACÍ AUTORA

- B-1 SCHWARZ, J.: Algoritmy stochastické a simulované evoluce. In *Sborník kolokvia Vybrané problémy simulačních modelů*. Ostrava, 1993, pp. 89–91. ISBN 80-901229-6-5.
- B-2 SCHWARZ, J.: Simulation oriented placement methods for VLSI chips. In *Proceedings of MOSIS'93 conference*. Olomouc: MARQ, 1993, pp. 295–300.
- B-3 SCHWARZ, J.: Placement optimization using the genetic algorithm. In *Proceedings of MOSIS'94*. Zábřeh na Moravě: MARQ, 1994, pp. 171–176. ISBN 80-901229-8-1.
- B-4 SCHWARZ, J.: Simulation based placement algorithm. In *Proceedings of EDS '95 conference*. Brno, 1995, pp. 217–228.
- B-5 SCHWARZ, J.: PGPLACE: Genetic algorithm for placement optimization. In *Proceedings of the 1st International Mendel Conference on Genetic Algorithms, Mendel '95*. Brno: Technical University of Brno, 1995, pp. 139–144. ISBN 80-214-0672-0.
- B-6 SCHWARZ, J., NĚMEC, V.: Parallel genetic algorithms implemented on transputers. In *Proceedings of the 2nd International Mendel Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets, Mendel '96*. Brno: Technical University of Brno, 1996, pp. 85–90. ISBN 80-214-0769-7 (70/30 %).

- B-7** SCHWARZ, J.: The FIDE system flexibility in the process of the fuzzy system design. In *Proceedings of MOSIS'96*. Krnov, 1996, pp. 82–87. ISBN 80-85988-02-X.
- B-8** SCHWARZ, J.: Educational fuzzy development system. In *Proceedings of Conference MOSIS '97*. Hradec nad Moravici: MARQ, 1997, pp. 233–238. ISBN 80-85988-16-X.
- B-9** SCHWARZ, J.: Case study: Genetic object designer. In *ASIS 1998*. Krnov: MARQ, 1998, pp. 109–114. ISBN 80-85988-26-7.
- B-10** SCHWARZ, J.: Genetic algorithm for partitioning circuits. In *Proceedings of the 4th International Mendel Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets. Mendel '98*. Brno: Technical University of Brno, 1998, pp. 126–131. ISBN 80-214-1199-6.
- B-11** SCHWARZ, J.: Utilizing Genetic Algorithms for VLSI Physical Design—a Brief Survey. In *Electronic Devices and Systems 1999. Proceedings*. Brno, 1999, pp. 88–91. ISBN 80-214-1466-9.
- B-12** SCHWARZ, J., OČENÁŠEK, J.: Pareto Bayesian Optimization Algorithm for the Multiobjective 0/1 Knapsack Problem. In *Proceedings of the 7th International Mendel Conference on Soft Computing. Mendel 2001*. Brno: Technical University of Brno, pp. 131–136. ISBN 80-214-1894-X (50/50 %).
- B-13** SCHWARZ, J.: The probability models for combinatorial optimization problems. In *Proceedings of the 4th Japan-Central Europe Joint Workshop on Energy and Information in Non-Linear Systems. Brno, Czech Republic, November 10–12, 2000*. Brno, 2000, pp. 72–75.
- B-14** OČENÁŠEK, J., SCHWARZ, J.: The Parallel Bayesian Optimization Algorithm. In *Proceedings of the European Symposium on Computational Intelligence, Košice, SK*. Springer, 2000, pp. 61–67. ISBN 3-7908-1322-2, ISSN 1615-3871 (80/20 %).
- B-15** OČENÁŠEK, J., SCHWARZ, J.: The Distributed Bayesian Optimization Algorithm for Combinatorial Optimization. In *EUROGEN 2001: Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems, Athens, GR, 2001*, pp.115–120. ISBN 84-89925-97-6 (80/20 %).
- B-16** OČENÁŠEK, J., SCHWARZ, J.: Estimation Distribution Algorithm for mixed continuous-discrete optimization problems. In *Proceedings of 2nd Euro-International Symposium on Computational Intelligence, Kosice, Slovakia*. IOS Press, 2002, pp. 227–232. ISBN 1-58603-256-9, ISSN 0922-6389 (80/20 %).
- B-17** OČENÁŠEK, J., SCHWARZ, J.: Development system DEBOA for rapid prototyping of evolutionary applications. In *Proceedings of International Conference MOSIS '02, Ostrava, CZ*. MARQ, 2002, pp. 169–176. ISBN 80-85988-71-2 (80/20 %).

## ABSTRACT

The habilitation thesis „Bayesian evolutionary algorithms applied in decomposition and allocation problems“ deals with the design, analysis and applications of Bayesian evolutionary algorithms for the solution of complex almost NP-complete combinatorial optimization problems mainly from the area of decomposition and allocation of graph structures. This task can be often interpreted in the context of physical design of digital circuits as partitioning and placement of digital circuits.

Bayesian evolutionary algorithms are advanced evolutionary algorithms based on the probabilistic graph models. These algorithms lack the well known problem of the standard genetic algorithms with the convergence and the drawback arising from the requirement on the specification of the control parameters and genetic operators.

The original Bayesian Optimization Algorithm (BOA), firstly published in [Pel99a], [Pel99b], was only capable for the solution of theoretical binary single-criterion problems.

The main goal of the habilitation thesis was to develop advanced Bayesian evolutionary optimization algorithms capable for the solving of real optimization tasks including decomposition and allocation of hypergraphs. Four important capabilities of these algorithms were demanded:

- Optimization of combinatorial problems
- Usage of problem knowledge
- Multiobjective optimization
- General usage of binary decision graphs (BDD)

As a result of long-term research four basic issues of advanced Bayesian optimization algorithms (BOA) were developed that fulfil the stated demands. The main attention was focused on the various type of decomposition of hypergraphs using well known benchmarks and objective functions. These decomposition algorithms can be used in various context e.g. for decomposition of complex systems, telecommunications networks, complex circuit structures etc. In addition developed version of advanced BOA algorithms can be simply modified for the solution of other combinatorial or numerical optimization problems. The probability model was based on the Bayesian network and the binary decision diagram.

In the Knowledge-based BOA algorithm (KBOA) the partial knowledge (prior information) about the problem was used to modify the probabilistic model. In addition, an injection of building blocks was used to improve the quality of initial population. This approach resulted in optimization process acceleration.

Important results were also gained through extending the single-objective BOA to the multiobjective one. The experiments show that the realized Pareto-strength BOA algorithm which utilizes a promising Pareto-strength niching technique, outperforms the weighting-based and classical constraint approaches, they are comparable or better than the best standard multiobjective genetic algorithms NSGA and SPEA [Zitz99].