

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

**PARALELNÍ EVOLUČNÍ ALGORITMUS EDA
VYUŽÍVAJÍCÍ TEORII KOPULÍ**

PARALLEL EVOLUTIONARY ALGORITHM EDA BASED ON COPULAS

TEZE DISERTAČNÍ PRÁCE - AUTOREFERÁT

AUTOR PRÁCE

Ing. MARTIN HYŘŠ

ŠKOLITEL

doc. Ing. JOSEF SCHWARZ, CSc.

BRNO 2020

Abstrakt

Ve své disertační práci se zabývám návrhem, implementací a testováním pokročilého paralelního algoritmu EDA (Estimation of Distribution Algorithm) využívajícího teorii kopulí pro tvorbu pravděpodobnostního modelu. Nová populace se vytváří v procesu vzorkování sdružené distribuční funkce, která modeluje aktuální rozložení subpopulace slibných jedinců. Použití kopulí umožňuje zefektivnit proces učení a vzorkování pravděpodobnostního modelu. Lze jej separovat na vzájemně nezávislá marginální rozdělení a kopuli, která reprezentuje korelace mezi proměnnými řešeného problému. Tato koncepce iniciovala použití paralelní ostrovní struktury, v níž bylo použito místo migrace jedinců migrace pravděpodobnostních modelů příslušejících jednotlivým ostrovním subpopulacím. Statistické testy použité při komparaci navrženého algoritmu (mCEDA = migrating Copula-based Estimation of Distribution Algorithm) a algoritmů jiných autorů potvrdily efektivnost navržené koncepce.

Klíčová slova

Algoritmy EDA, optimalizace, teorie kopulí, vzorkování vícerozměrných kopulí, paralelismus, ostrovní model, migrace pravděpodobnostních modelů.

Citace

HYRŠ, Martin. *Paralelní evoluční algoritmus EDA využívající teorii kopulí*. Brno, 2020. Teze disertační práce – autoreferát. Vysoké učení technické v Brně, Fakulta informačních technologií. Školitel doc. Ing. Josef Schwarz, CSc.

Abstract

In my thesis I deal with the design, implementation and testing of the advanced parallel Estimation of Distribution Algorithm (EDA) utilizing copula theory to create a probabilistic model. A new population is created by the process of sampling the joint distribution function, which models the current distribution of the subpopulation of promising individuals. The usage of copulas increases the efficiency of the learning process and sampling the probabilistic model. It can be separated into mutually independent marginal distributions and the copula, which represents the correlations between the variables of the solved problem. This concept initiated the usage of the parallel island architecture, in which the migration of probabilistic models belonging to individual islands' subpopulations was used instead of the migration of individuals. The statistical tests used in the comparison of the proposed algorithm (mCEDA = migrating Copula-based Estimation of Distribution Algorithm) and the algorithms of other authors confirmed the effectiveness of the proposed concept.

Keywords

EDA, Estimation of Distribution Algorithms, Optimization, Copula Theory, Multivariate Copula Sampling, Parallelisation, Parallel EDA, Island-based Model, Migration of Probabilistic models.

1 Úvod

Algoritmy EDA (Estimation of Distribution Algorithms) patří do skupiny evolučních algoritmů pro nalezení optimálních či suboptimálních řešení různých matematických či inženýrských úloh. Jsou navrženy zejména pro řešení komplikovaných úloh se silným stupněm závislosti mezi parametry (proměnnými). Vývoj hledaného řešení probíhá opakovaně během velkého počtu generací. V každé generaci jsou vybráni jedinci, kteří svojí kvalitou splňují výběrové kritérium, a tito jedinci jsou použiti k vytvoření pravděpodobnostního modelu. Jeho vzorkováním je získána nová populace.

Existuje řada různě sofistikovaných metod pro tvorbu pravděpodobnostních modelů, od nejjednodušších, které neuvažují žádné závislosti mezi proměnnými, přes různé formy jednoduchých závislostí, až po složité varianty, jako jsou například Bayesovské grafové modely. Tato problematika je v literatuře zpracována zejména pro řešení celočíselných a kombinatorických úloh. Problematika optimalizačních úloh v reálné doméně je propracována méně.

V posledních letech došlo k většímu rozpracování koncepce tvorby pravděpodobnostních modelů pomocí matematického prostředku zvaného *kopule*, který byl poprvé představen A. Sklarem v roce 1959 [48]. Přínos koncepce kopulí spočívá v tom, že vícerozměrnou sdruženou distribuční funkci lze reprezentovat separátně nezávislými jednorozměrnými marginálními distribučními funkcemi a kopulí, která reprezentuje závislosti mezi proměnnými.

V této práci půjde především o aplikační rozpracování této koncepce pro využití v optimalizačních úlohách reálné domény s využitím algoritmů EDA.

Dalším aspektem zkoumaným v disertační práci je použití paralelizace při návrhu EDA algoritmu. Je použit tzv. „ostrovni model“ – celková populace je rozdělena na několik částí (tzv. ostrovy), na nichž evoluce probíhá nezávisle. Výměna informací mezi ostrovy probíhá prostřednictvím migrace pravděpodobnostních modelů. Klíčovým problémem je, jak na každém ostrově efektivně zkombinovat rezidentní a příchozí pravděpodobnostní model. U klasických algoritmů EDA využívajících grafové modely je to netriviální úloha. Využitím kopulí může být tato úloha výrazně zjednodušena.

1.1 Cíle disertační práce

Definuji následující cíle výzkumu:

Cíl 1 Navrhnout koncept paralelního algoritmu EDA založeného na teorii kopulí a migraci pravděpodobnostních modelů (mCEDA = migrating Copula-based Estimation of Distribution Algorithm).

Cíl 1.1 Navrhnout sekvenční část tohoto algoritmu.

Cíl 1.2 Navrhnout paralelní část tohoto algoritmu se specifikací komunikace mezi ostrovy. Navrhnout techniku kombinace migrujících a rezidentních modelů.

Cíl 2 Implementovat navržený algoritmus mCEDA a vyhodnotit jeho efektivitu.

Cíl 2.1 Navrhnout a uskutečnit soubor experimentů pro vybranou sadu řídicích parametrů, zahrnující klasické parametry sekvenčního kopulového algoritmu EDA a parametry jeho paralelního verze.

Cíl 2.2 Experimentálně porovnat finální podobu mCEDA vůči algoritmům publikovaných jinými autory.

2 Úvod do teorie kopulí

V této kapitole bude uvedena matematická definice kopule a vysvětlen význam Sklarova teorému. Pro jednotlivé rodiny kopulí je také popsána metodika vzorkování kopulí, která bude využívána v navazujících kapitolách.

2.1 Definice kopule

Existuje několik ekvivalentních definic kopule. Na tomto místě uvedu dvě definice, které jsou pro tento text nejvíce relevantní. První zde uvedená definice výslovně definuje kopuli jako distribuční funkci [24, 38]:

Definice 2.1. Kopule je taková vícerozměrná distribuční funkce, jejíž všechna marginální rozdělení jsou rovnoměrná rozdělení na intervalu $\langle 0; 1 \rangle$.

Druhá zde uvedená definice názorně popisuje konkrétní vlastnosti kopule [49, 33]:

Definice 2.2. Kopule je funkce $C : \langle 0; 1 \rangle^D \rightarrow \langle 0; 1 \rangle$, které splňuje tyto vlastnosti:

1. $C(u_1, u_2, \dots, u_D) = 0$ pro alespoň jedno $u_d = 0$,
2. $C(1, 1, \dots, 1, u_d, 1, \dots, 1) = u_d$ pro všechna $d = 1, 2, \dots, D$,
3. $\forall (u_1, \dots, u_D), (v_1, \dots, v_D) \in \langle 0; 1 \rangle^D, u_d \leq v_d :$
$$\sum_{(w_1, \dots, w_D) \in \prod_{d=1}^D \{u_d, v_d\}} (-1)^{|\{d | w_d \equiv u_d\}|} C(w_1, \dots, w_D) \geq 0.$$

Mimo to existují i definice kopule pomocí funkce přežití či Laplaceovy transformace (viz [33, 38]).

2.2 Sklarův teorém

Sklarův teorém [48] má v teorii kopulí velký význam. Popisuje vztah mezi sdruženou distribuční funkcí a mezi kopulí spolu s marginálními rozděleními. Lze říci, že při reprezentaci sdružené distribuční funkce lze oddělit efekt marginálních rozdělení od efektu závislosti mezi proměnnými. Právě důsledky Sklarova teorému jsou důvodem, proč teorie kopulí stále více nachází uplatnění při modelování sdružených pravděpodobnostních rozložení.

Teorém 2.1 (Sklarův teorém). *Nechť F je D -rozměrná sdružená distribuční funkce s marginálními distribučními funkcemi F_1, \dots, F_D . Pak existuje taková D -rozměrná kopule C , že pro $\forall(x_1, \dots, x_D) \in \mathbb{R}^D$ platí*

$$F(x_1, x_2, \dots, x_D) = C(F_1(x_1), F_2(x_2), \dots, F_D(x_D)). \quad (2.1)$$

Pokud jsou funkce F_1, \dots, F_D spojité, pak je kopule C určena jednoznačně.

Je-li C D -rozměrná kopule a F_1, \dots, F_D jsou jednorozměrné distribuční funkce, pak F je D -rozměrná sdružená distribuční funkce.

Hustota pravděpodobnosti vícerozměrného rozdělení je za využití kopule popsána vztahem

$$f(x_1, x_2, \dots, x_D) = c(F_1(x_1), \dots, F_D(x_D)) \prod_{d=1}^D f_d(x_d), \quad (2.2)$$

kde $c(\cdot)$ je hustota kopule a $f_d(\cdot)$ je hustota marginálního rozdělení.

Důsledkem Sklarova teorému je, že kopule nabízí cestu, jak transformovat náhodný vektor $(\mathbb{X}_1, \dots, \mathbb{X}_D)$ na jiný náhodný vektor $(\mathbb{U}_1, \dots, \mathbb{U}_D) = (F_1(\mathbb{X}_1), \dots, F_D(\mathbb{X}_D))$ mající uniformní marginální rozdělení na intervalu $(0; 1)$ a přitom zachovat závislost (korelace) mezi jeho komponentami [24].

2.3 Vzorkování kopulí

Pro využití kopulí v algoritmech EDA jsou stěžejní postupy pro jejich vzorkování. Ve své disertační práci se podrobně zabývám třemi různými přístupy.

2.3.1 Využití podmíněné pravděpodobnosti

Na základě obecné metodologie – viz [33] – jsem odvodil postup a výsledné rovnice pro vzorkování dvourozměrné Gaussovy kopule, založené na podmíněné pravděpodobnosti. Jedna proměnná je navzorkována nezávisle, druhá je dopočítána tak, aby byla uplatněna existující korelace ρ mezi proměnnými, viz alg. 2.1.

Tento postup jsem publikoval jako součást článku [G] a částečně i [F].

Algoritmus 2.1 Vzorkování dvourozměrné Gaussovy kopule.

1. vygeneruj nezávislou náhodnou hodnotu $u \sim U(0; 1)$
 2. vygeneruj pomocnou náhodnou proměnnou $\omega \sim U(0; 1)$
 3. dopočítej hodnotu $v := \Phi\left(\sqrt{1 - \rho^2} \Phi^{-1}(\omega) + \rho \Phi^{-1}(u)\right)$
 4. výsledek je dvojice (u, v)
-

2.3.2 Vzorkování Gaussovy kopule pomocí vylučovací metody

Ve své disertaci jsem odvodil postup pro vzorkování vícerozměrné Gaussovy kopule pomocí vylučovací metody, zde je uveden jako alg. 2.2.

Tento algoritmus jsem publikoval v článku [F].

Výhodou vzorkování pomocí vylučovací metody je jeho poměrně snadná modifikace pro jiná rozdělení. Nevýhodou je výpočetní náročnost (jedná se o iterativní výpočet).

Algoritmus 2.2 Vzorkování vícerozměrné Gaussovy kopule pomocí vy-
lučovací metody.

1. sestav pomocnou matici $A = R^{-1} - I$
(R je daná korelační matice, I je jednotková matice)
 2. REPEAT
 3. \rightarrow generuj náhodná $u_d \sim U(0, 1)$, $d = 1, \dots, D$
 4. \rightarrow vypočítej hodnoty $v_d := \Phi^{-1}(u_d)$
 5. \rightarrow generuj pomocnou náhodnou hodnotu $u \sim U(0, 1)$
 6. UNTIL $u \leq e^{(-\frac{1}{2} \sum_{i=1}^D \sum_{j=1}^D A_{ij} v_i v_j)}$
 7. výsledek je D -tice (u_1, \dots, u_D)
-

2.3.3 Algoritmy pro efektivní vzorkování kopulí

Výpočetně mnohem efektivnější, v porovnání s algoritmy z předešlé kapitoly, jsou algoritmy specificky navržené pro vzorkování kopulí.

Algoritmus pro vzorkování **archimédovských kopulí** využívá náhodnou hodnotu J , která je obecně získána z rozdělení pravděpodobnosti určeného podle inverze Laplaceovy transformace \mathcal{L}^{-1} generátoru φ [33, 1, 34, 16], viz alg. 2.3.

Vzorkování **Gaussovy a Studentovy kopule** [33] (viz alg. 2.4, 2.5) je založeno na Choleského dekompozici dané korelační matice R , čímž je získána dolní trojúhelníková matice L , taková že $LL^T = R$. Studentova kopule je dále specifikována stupni volnosti ν .

Algoritmus 2.3 Vzorkování archimédovských kopulí.

- vygeneruj náhodnou hodnotu $J \sim \mathcal{L}^{-1}[\varphi(t)]$:
 - pro Claytonovu kopuli: $J \sim \text{Gamma}\left(\frac{1}{\theta}, \theta\right)$
 - pro Gumbelovu kopuli: $J \sim \text{Stable}\left(\frac{1}{\theta}, 1, (\cos \frac{\pi}{2\theta})^\theta, 0\right)$
 - pro Frankovu kopuli: $J \sim \text{Logarithmic}(1 - e^{-\theta})$
 - vygeneruj hodnoty (z rovnoměrného rozdělení) $z_d \sim U(0, 1)$ (pro $d = 1, \dots, D$)
 - spočítej $u_d := \varphi\left(\frac{-\log(z_d)}{J}\right)$ (pro $d = 1, \dots, D$)
 - výsledek je D -tice (u_1, \dots, u_D)
-

Algoritmus 2.4 Efektivní vzorkování Gaussovy kopule.

- urči L — dolní trojúhelníková matice, $LL^T = R$
 - vygeneruj náhodná čísla $z_d \sim \mathcal{N}(0, 1)$ z normovaného normálního rozdělení (pro $d = 1, \dots, D$)
 - vypočítej $s_d := \sum_{j=1}^d L_{d,j} z_j$ (pro $d = 1, \dots, D$)
 - spočítej $u_d := \Phi(s_d)$ (pro $d = 1, \dots, D$)
 - výsledek je D -tice (u_1, \dots, u_D)
-

Algoritmus 2.5 Efektivní vzorkování Studentovy kopule.

- urči L — dolní trojúhelníková matice, $LL^T = R$
 - vygeneruj $V \sim \chi^2(\nu)$
 - vygeneruj náhodná čísla $z_d \sim \mathcal{N}(0, 1)$ z normovaného normálního rozdělení (pro $d = 1, \dots, D$)
 - vypočítej $s_d := \sqrt{\frac{\nu}{V}} \sum_{j=1}^d L_{d,j} z_j$ (pro $d = 1, \dots, D$)
 - spočítej $u_d := t_\nu(s_d)$ (pro $d = 1, \dots, D$)
 - výsledek je D -tice (u_1, \dots, u_D)
-

3 Přehled existujících algoritmů EDA

Algoritmy EDA (Estimation of Distribution Algorithm) patří do skupiny evolučních algoritmů pro hledání optimálních či suboptimálních řešení různých matematických či inženýrských úloh. Jsou navrženy zejména pro řešení komplikovaných úloh se silným stupněm závislosti mezi parametry (proměnnými).

Formálně lze popsat optimalizační úlohu ve tvaru

$$\mathbf{x}_{opt} = \arg \min_{\mathbf{x} \in \mathbb{R}^D} fitness(\mathbf{x}). \quad (3.1)$$

Jedná se o evoluční mechanismus, který využívá teorie pravděpodobnosti. Vývoj hledaného řešení probíhá opakovaně během velkého počtu generací. V každé generaci jsou vybráni jedinci, kteří svojí kvalitou splňují výběrové kritérium, z nich je odvozen pravděpodobnostní model a nová populace je vytvářena vzorkováním jedinců ze získaného rozložení pravděpodobnosti, viz algoritmus 3.1.

Algoritmus 3.1 Pseudokód klasického algoritmu EDA.

1. náhodně vygeneruj počáteční populaci $P(0)$
 2. WHILE (nejsou splněny ukončovací podmínky) DO
 3. $|\rightarrow$ vyhodnoť stávající populaci $P(g)$
 4. $|\rightarrow$ vyber kandidátní řešení $S(P(g))$
 5. $|\rightarrow$ vytvoř pravděpodobnostní model M z $S(P(g))$
 6. $|\rightarrow$ navzorkuj nové jedince $O(g)$ podle modelu M
 7. $|\rightarrow$ vytvoř novou populaci $P(g+1)$, $P(g+1) \subset O(g) \cup P(g)$
 8. $|\rightarrow$ zvyš počítadlo generací g
 9. END WHILE
-

3.1 Přehled sekvenčních algoritmů EDA

Jednotlivé varianty EDA se liší zejména tím, jak složitý pravděpodobnostní model je použit. Tyto algoritmy byly původně navrženy pro řešení problémů s binární reprezentací, tj. proměnné mohou nabývat pouze binárních hodnot 0 nebo 1 . Položka $p(X_d)$ pravděpodobnostního modelu tak vyjadřuje pravděpodobnost, že se v proměnné x_d vyskytne hodnota 1 .

Nejjednoduššími algoritmy EDA jsou ty, ve kterých nejsou uvažovány žádné závislosti mezi proměnnými. Jejich pravděpodobnostní model je popsán rovnicí

$$p(X_1, \dots, X_D) = \prod_{d=1}^D p(X_d). \quad (3.2)$$

Příkladem jsou algoritmy UMDA (Univariate Marginal Distribution Algorithm) [36], PBIL (Population-Based Incremental Learning) [2], či cGA (Compact Genetic Algorithm) [21].

Pokročilejší varianty EDA již předpokládají existenci závislostí mezi dvojicemi proměnných, např. algoritmy MIMIC (Mutual-Information-Maximizing Input Clustering) [8], COMIT (Combining Optimizers with Mutual Information Trees) [3], či BMDA (Bivariate Marginal Distribution Algorithm) [40].

Nejkomplexnější pravděpodobnostní model mají algoritmy EDA s vícenásobnými závislostmi, je popsán obecným vztahem

$$p(X) = \prod_{d=1}^D p(X_d | \Pi_{X_d}), \quad (3.3)$$

kde Π_{X_d} označuje množinu uzlů, na nichž závisí X_d . Příkladem jsou algoritmy BOA (Bayesian Optimization Algorithm) [39] nebo EBNA (Estimation of Bayesian Networks Algorithm) [14].

3.1.1 Algoritmy EDA pro spojité problémy

Algoritmy EDA byly primárně navrženy pro řešení problémů s diskretní reprezentací, ale existují i pro spojité problémy. Jsou to jednak spojité varianty předešlých algoritmů, např. $UMDA_c^G$, $MIMIC_c^G$ [27, 22], a rovněž varianty určené specificky pro spojité úlohy – př. SHCLVND (Stochastic Hill Climbing with Learning by Vectors of Normal Distributions)

[42], EGNA (Estimation of Gaussian Network Algorithm) [27], IDEA (Iterated Density Estimation Evolutionary Algorithm) [4], EMNA (Estimation Multivariate Normal Algorithm) [28], ESTDA (Estimation of Student's t Distribution Algorithm) [31].

3.2 EDA využívající kopule (CEDA)

V algoritmech EDA založených na kopulích (CEDA = Copula-based Estimation of Distribution Algorithm) je pravděpodobnostní model rozdělen na dvě části:

- na jednorozměrná vzájemně nezávislá marginální rozdělení $F_d(x_d)$ (tato část je obdobná pravděpodobnostnímu modelu v algoritmech typu UMDA)
- a na kopuli $C(\cdot)$, která definuje strukturu závislostí.

Hustota pravděpodobnosti v takovém modelu je dána vztahem

$$p(\mathbf{x}) = c(F_1(x_1), \dots, F_D(x_D)) \prod_{d=1}^D f_d(x_d), \quad (3.4)$$

kde $c(\cdot)$ je hustota kopule a $f_d(\cdot)$ je hustota marginálního rozdělení.

Tématikou CEDA se zabývá jen relativně málo prací. Mnoho významných publikací lze přiřadit do jednoho ze tří tvůrčích kolektivů: Marta Soto et al. ([50, 17, 18]), Rogelio Salinas-Gutiérrez et al. ([43, 44, 45, 46]), Li-Fang Wang et al. ([53, 52, 54, 6, 20, 25]).

Řada algoritmů CEDA používá klasické eliptické nebo archimédovské kopule, popř. jejich kombinace. Často se pro vícerozměrné modely používají hierarchické modely typu C-vine či D-vine na bázi dvoudimenzionálních archimédovských kopulí. Vzhledem k navržené koncepci migrace modelů se jeví jako přínosnější použít v mojí disertaci vícerozměrné kopule.

Společným problémem publikovaných algoritmů je testování výkonnosti jen na malém počtu úloh. V jednotlivých člancích bývají použity odlišné typy testovacích úloh i odlišné řídicí parametry (např. ukončující podmínky) – je proto obtížné uskutečnit korektní porovnání rozdílných algoritmů. Převažují testovací úlohy s 10 dimenzemi.

3.3 Paralelní evoluční algoritmy

Moje disertace se zaměřuje na dekompoziční přístup k paralelizaci evolučního algoritmu, využívající migraci spojitých pravděpodobnostních modelů.

Princip migrace diskrétních pravděpodobnostních modelů byl poprvé navržen v [9]. Jako sekvenční algoritmus na každém ostrově byl použit algoritmus UMDA. Později byl proces kombinace pravděpodobnostní modelů ze dvou různých ostrovů vylepšen o metody lokálního prohledávání [10].

V článku [47] je navržen diskrétní paralelní algoritmus BMMA s migrací modelů; v [23] je navrženo další zjednodušení agregace pravděpodobnostních modelů.

V článku [11] je navržen hybridní migrující algoritmus na bázi EDA a PSO (Particle Swarm Optimization).

Studie [37] porovnává migraci jedinců a migraci pravděpodobnostních modelů v algoritmech EDA pro spojitě úlohy, konkrétně použili algoritmy UMDA^G a EMNA.

Obecně lze říci, že kombinování pravděpodobnostních modelů je netriviální úloha zejména při použití grafových modelů (kromě parametrů je nutné kombinovat i strukturu grafu). Možná právě proto se tento přístup nedočkal velkého rozšíření. Domnívám se, že efektivnější řešení může přinést použití kopulí díky separabilitě pravděpodobnostního modelu.

4 Návrh efektivního kopulového algoritmu EDA – mCEDA

V této kapitole se budu zabývat návrhem paralelního kopulového algoritmu EDA využívajícího migraci modelů (mCEDA = migrating Copula-based Estimation of Distribution Algorithm). Algoritmus podle mého návrhu je tvořen dvěma hlavními částmi, sekvenční a paralelní. Sekvenční část vychází z principů klasických algoritmů EDA a využívá pravděpodobnostní model založený na kopulích. Paralelní část zavádí nový koncept migrace pravděpodobnostních modelů v kopulovém algoritmu EDA.

4.1 Sekvenční jádro navrženého algoritmu

Sekvenční jádro mnou navrženého algoritmu odpovídá klasickému schématu algoritmů EDA, přičemž fáze vytváření a vzorkování modelu jsou užitím kopulí rozděleny do dvou nezávislých dílčích kroků, jak ukazuje alg. 4.1.

Vytváření dekomponovaného modelu je ilustrováno na obrázku 4.1. Z populace jsou vybráni slibní jedinci a pouze tito jsou použiti pro vytvoření modelu. Pro první část modelu jsou specifikována vzájemně nezávislá marginální rozdělení pro každou proměnnou řešené úlohy. Používám normální rozdělení a příslušné parametry jsou stanoveny s použitím MLE odhadu. Druhou částí modelu je kopule. Její parametry jsou odvozeny pouze z korelací mezi proměnnými, není třeba přihlížet k volbě typu marginálních rozdělení.

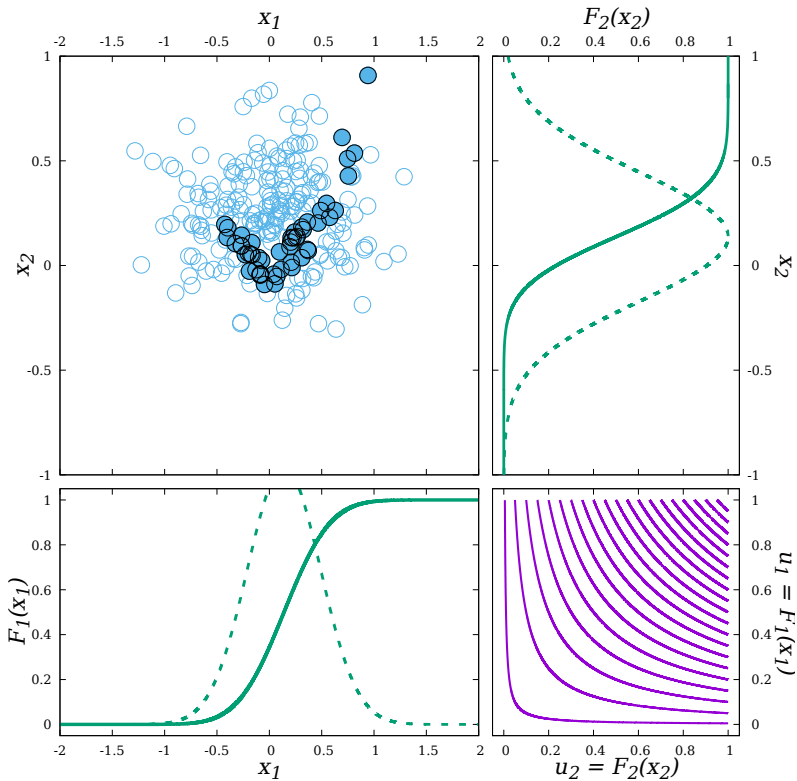
Vzorkováním pravděpodobnostního modelu dochází k vytváření nových jedinců do další populace. Vzorkování je dvoufázové, je schematicky znázorněno na obrázku 4.2. Prvním krokem je navzorkování kopule, vzorkem kopule je D -tice $(u_1, \dots, u_D) \in \langle 0; 1 \rangle^D$, druhým je získání hodnot marginálních rozdělení pomocí inverzní distribuční funkce $x_d = F_d^{-1}(u_d)$, čímž postupně získáme nové jedince jako D -tici (x_1, \dots, x_D) .

Algoritmus 4.1 Pseudokód sekvenčního kopulového algoritmu EDA.

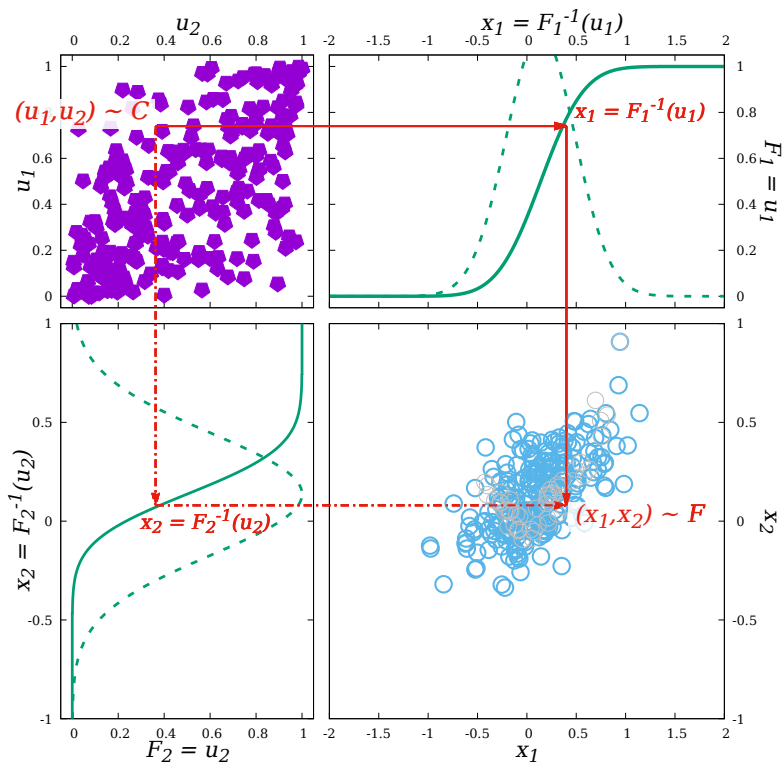
1. náhodně vygeneruj počáteční populaci $P(0) = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$,
 $|P(0)| = N$, $\mathbf{x}_p \sim U(MIN, MAX)$ $p = 1, \dots, N$
 2. WHILE (nejsou splněny ukončovací podmínky) DO
 3. |--> vyber kandidátní řešení $S(P(g))$, $|S(P(g))| = N'$

 // vytvoř pravděpodobnostní model M :
 4. |--> urči parametry marginálních rozdělání F_d
 5. |--> urči parametry kopule C

 // navzorkuj novou populaci $O(g)$ podle pravděpodobnostního
 modelu M , $|O(g)| = N - N''$:
 6. |--> FOR EACH nový jedinec DO
 7. |--> |--> získej vzorek kopule $(u_1, \dots, u_D) \sim C$
 8. |--> |--> vytvoř nového jedince
 $(x_1, \dots, x_D) := (F_1^{-1}(u_1), \dots, F_D^{-1}(u_D))$
 9. |--> END FOR EACH
 10. |--> vytvoř novou populaci $P(g+1)$ sjednocením $O(g)$
 s N'' nejlepšími jedinci z $P(g)$
 11. |--> inkrementuj g
 12. END WHILE
-



Obrázek 4.1: Ilustrace modelování sruženého rozdělení pomocí marginálních rozdělení a kopule – vytváření pravděpodobnostního modelu. Na počátku procesu je populace kandidátních jedinců, z níž jsou vybráni slibní jedinci (na grafu zvýrazněni) – levá horní část obrázku. Z těchto vybraných jedinců je odvozen pravděpodobnostní model tvořený parametry pro marginální rozdělení $F_1(x_1)$ a $F_2(x_2)$ (jsou použita normální rozdělení) – schematicky znázorněno v pravé horní a levé dolní části obrázku – a hodnotou korelace, pomocí níž je určena hodnota parametru θ resp. R pro zvolenou kopuli $C(u_1, u_2)$ – kopule je schematicky znázorněna pomocí vrstevnic v pravé dolní části obrázku.



Obrázek 4.2: Ilustrace použití marginálních rozdělání a kopule v algoritmech EDA – vzorkování nové populace (dvourozměrný příklad). Prvním krokem vzorkování je získání náhodného vzorku $(u_1, u_2) \sim C$ z kopule – vzorky kopule jsou zobrazeny v levé horní části obrázku. Následně jsou pomocí inverzí marginálních rozdělání dopočítány hodnoty $x_1 = F_1^{-1}(u_1)$ a $x_2 = F_2^{-1}(u_2)$ – pravá horní a levá dolní část obrázku. Dvojice (x_1, x_2) tvoří jednoho nového jedince – v pravé dolní části obrázku. Tento postup je opakován, dokud není vytvořena celá nová populace.

4.2 Paralelní část navrženého algoritmu

V předchozí části je popis sekvenčního jádra (alg. 4.1), které je implementované na každém uzlu (ostrově) v paralelní topologické struktuře. Paralelní část realizující migraci pravděpodobnostních modelů odpovídá základní ostrovní topologii:

1. Celková superpopulace je rozdělena na několik ostrovů. Populace na každém ostrově se většinu doby vyvíjejí nezávisle na sobě.
2. Samotná migrace modelů znamená to, že v určenou chvíli (např. po uskutečnění stanoveného počtu generací) dojde ke komunikaci mezi ostrovními populacemi a pravděpodobnostní model vyvinutý na jednom ostrově je přenesen na jiný ostrov (volba cílového ostrova závisí na topologii komunikační sítě). Na cílovém ostrově je poté provedena kombinace původního (rezidentního) modelu s příchozím (imigrantským) modelem.

Takto utvořený algoritmus má potenciál dosahovat lepší konvergence než sekvenční verze algoritmu.

Celková struktura navrženého paralelního algoritmu je popsána v algoritmu 4.2, jeho datové toky jsou znázorněny na obrázku 4.3. Stěžejní jsou dva procesy: aktualizace modelu na každém ostrově a komunikace mezi ostrovy tak, aby každý ostrov odesílal a přijímal modely ve vhodný čas. Je použita asynchronní neblokující komunikace, kdy nevznikají při posílání zpráv žádná vynucená čekání. Je-li přijato více modelů naráz, jsou cyklicky zpracovány v rámci jedné generace.

4.2.1 Kombinace pravděpodobnostních modelů

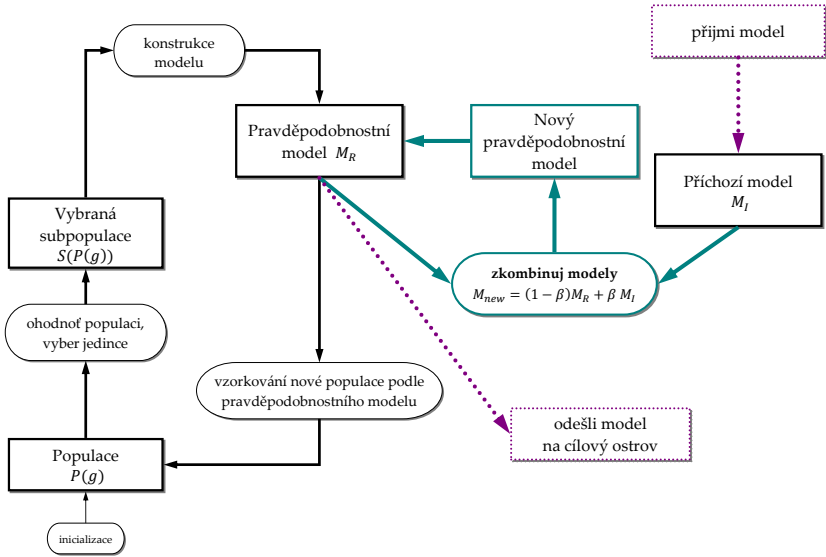
V souladu s navrženou strukturou algoritmu je migrační proces dekomponován na jednosměrné interakce dvou ostrovů. Ostrov, který přijímá poslaný pravděpodobnostní model, je označen jako rezidentní, jeho pravděpodobnostní model je rezidentní model M_R . Ostrov, jehož pravděpodobnostní model je odesílán, je označen jako imigrantský, k němu příslušný pravděpodobnostní model je M_I . (Označení je provedeno z pohledu rezidenta, tedy M_I je příchozí model.)

Obecně můžeme modifikaci rezidentského modelu modelem příchozím formalizovat jako

$$M_R^{new} = (1 - \beta)M_R \circ \beta M_I, \quad (4.1)$$

Algoritmus 4.2 Pseudokód algoritmu EDA s migrací modelů.

1. FOR EACH ostrov DO IN PARALLEL
 2. | \rightarrow náhodně vygeneruj počáteční populaci $P(0)$
 3. | \rightarrow WHILE (nejsou splněny ukončovací podmínky) DO
 4. | \rightarrow | \rightarrow vyber kandidátní řešení $S(P(g))$
 5. | \rightarrow | \rightarrow vytvoř pravděpodobnostní model M_R
 6. | \rightarrow | \rightarrow IF (splněny podmínky pro odeslání) THEN
 7. | \rightarrow | \rightarrow | \rightarrow odešli model M_R
 8. | \rightarrow | \rightarrow END IF
 9. | \rightarrow | \rightarrow WHILE (obdržen příchozí model M_I) DO
 10. | \rightarrow | \rightarrow | \rightarrow zkombinuj modely $M_R^{new} := (1 - \beta)M_R + \beta M_I$
 11. | \rightarrow | \rightarrow | \rightarrow $M_R := M_R^{new}$
 12. | \rightarrow | \rightarrow END WHILE
 13. | \rightarrow | \rightarrow navzorkuj populaci $O(g)$ podle modelu M_R
 14. | \rightarrow | \rightarrow vytvoř novou populaci $P(g + 1)$ sjednocením $O(g)$ s nejlepšími jedinci z $P(g)$
 15. | \rightarrow | \rightarrow inkrementuj g
 16. | \rightarrow END WHILE
 17. END IN PARALLEL
-



Obrázek 4.3: Schéma algoritmu EDA s migrací modelů.

kde koeficient $\beta \in \langle 0; 1 \rangle$ specifikuje míru vlivu imigrantského modelu. Vzhledem k vlastnostem kopulí lze nezávisle aktualizovat parametry marginálních distribucí a korelační koeficienty kopule.

Podle [15] jsem pro kombinaci modelů navrhl následující pravidla:

- Aktualizace střední hodnoty μ_d pro každé jednorozměrné marginální rozdělení $F_d(x_d)$:

$$\mu_d^{new} = (1 - \beta)\mu_d^R + \beta\mu_d^I. \quad (4.2)$$

- Aktualizace směrodatné odchylky σ pro každé jednorozměrné marginální rozdělení $F_d(x_d)$:

$$\begin{aligned} (\sigma_d^{new})^2 = (1 - \beta) \left((\mu_d^{new} - \mu_d^R)^2 + (\sigma_d^R)^2 \right) + \\ + \beta \left((\mu_d^{new} - \mu_d^I)^2 + (\sigma_d^I)^2 \right). \end{aligned} \quad (4.3)$$

- Aktualizace hodnot prvků korelační matice ρ_{ij} :

$$\rho_{ij}^{new} = (1 - \beta)\rho_{ij}^R + \beta\rho_{ij}^I. \quad (4.4)$$

Hodnotu aktualizačního koeficientu β jsem zvolil jako

$$\beta = \begin{cases} \frac{fit^R}{fit^R + fit^I} & fit^I \leq fit^R, \\ 0.1 & \text{jinak,} \end{cases} \quad (4.5)$$

kde fit^R resp. fit^I reprezentují kvalitu subpopulace $S(P(g))$ (vyjádřenou střední hodnotou fitness hodnot vybraných jedinců) na rezidentském resp. imigrantském ostrově.

5 Provedené experimenty a jejich vyhodnocení

Významnou část mého výzkumu tvořily experimentální testy s navrženým algoritmem. V první fázi (viz kap. 5.1) jsem se zabýval nastavením řídicích parametrů navrženého algoritmu.

Výsledky získané z finální verze paralelního algoritmu mCEDA jsem pro dva soubory testovacích úloh porovnal s výsledky jiných autorů (kap. 5.2), abych ověřil, že navržený koncept migrace modelů přináší zlepšení konvergence.

5.1 Nastavení řídicích parametrů

Zaměřil jsem se na vliv sedmi parametrů určujících typ algoritmu. Vycházím z experimentů, které jsem postupně publikoval v článcích [D, B, A], a ve kterých jsem popsal proces získání hodnot těchto parametrů, při kterých dosahuje algoritmus nejlepších výsledků:

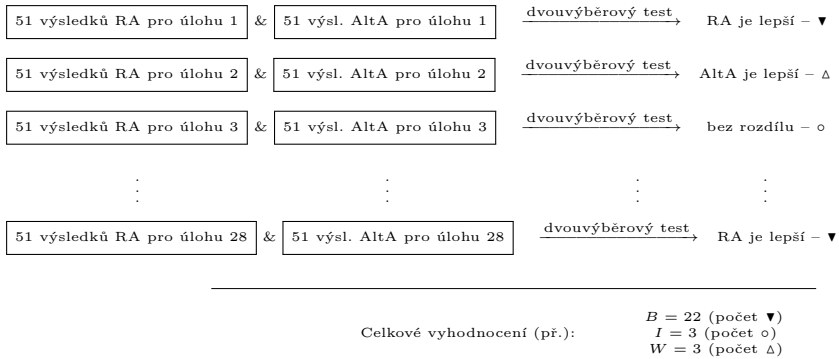
- typ kopule: Gaussova,
- velikost populace: 250 jedinců na každém ostrově,
- velikost selekce: nejlepších 50 jedinců,
- obnova populace: zachování 3 nejlepší jedinci,
- počet ostrovů: 4,
- perioda migrací: 5 generací,
- topologie: obousměrný prstenec.

Provedl jsem sedm experimentů. V každém experimentu měním hodnotu právě jednoho z těchto sedmi parametrů ve vybraném intervalu hodnot.

5.1.1 Postup vyhodnocení

Experimenty jsem prováděl na sadě CEC 2013 [29]¹, sada je tvořena 28 úlohami rozdělených do tří skupin podle složitosti (unimodální, multimodální a složené úlohy). Pro každou zkoumanou variantu algoritmu a pro

¹V disertační práci jsou také prezentovány výsledky experimentů na souboru klasických testovacích úloh.



Obrázek 5.1: Schéma vyhodnocení výsledků pro jednotlivé úlohy pomocí dvouvýběrových testů mezi referenčním algoritmem (RA) a alternativním algoritmem (AltA).

každou testovací úlohu bylo provedeno 51 náhodných běhů, s celkovým počtem 100000 vyhodnocení fitness funkce. Výsledkem každého běhu je nejlepší nalezený jedinec (s nejmenší hodnotou fitness funkce, v našem případě s nejmenší odchylkou funkční hodnoty od známé hodnoty optima označovanou jako chybová funkce). Těchto 51 hodnot je použito při výpočtu následně uvedených statistických testů a pro stanovení příslušných statistik – zejména mediánu, střední hodnoty a rozptylu.

Algoritmus s výchozím nastavením daného parametru je označen jako referenční algoritmus (RA), každá alternativní varianta (s odlišnou hodnotou parametru) je porovnávána vůči němu. Každý tento pár algoritmů (RA vs. alternativní algoritmus) jsem porovnával pro každou úlohu pomocí dvouvýběrového Wilcoxonova testu na hranici významnosti $\alpha = 0.05$, viz schéma na obr. 5.1.

V grafech na obr. 5.2 až obr. 5.8 jsou znázorněny sumarizované výsledky statistických testů:

- zelená (horní) část histogramu – B (počet úloh, kdy je RA statisticky významně lepší než alternativní algoritmus),
- žlutá (prostřední) část histogramu – I (počet úloh bez statisticky významného rozdílu),

- červená (spodní) část histogramu – W (počet úloh, kdy je RA statisticky významně horší než alternativní algoritmus).

Převažujícím typem výsledku je $B > W$ (zelená část histogramu je větší jak červená) – v tomto případě je RA považován za lepší, než alternativní varianta.

5.1.2 Porovnání vhodnosti užití různých typů kopulí

Všeobecně používané kopule spadají do dvou rodin – archimédovských a eliptických. Archimédovských kopulí je mnoho různých typů. Typicky mají pro vyjádření závislosti jen jeden parametr, z toho nutně plyne, že jejich schopnost popsat adekvátní korelace je v případě vícerozměrných úloh omezená. (V praxi se často toto omezení řeší pomocí skládání dvojrozměrných archimédovských kopulí do hierarchických modelů, tato oblast je poměrně dobře prozkoumaná, avšak pro mojí práci zaměřenou na kombinaci modelů nevhodná.)

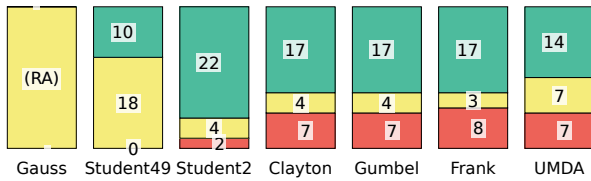
Naproti tomu eliptické kopule popisují závislost proměnných úplným výčtem podvojných závislostí. Lze proto předpokládat, že u vícerozměrných problémů poskytnou mnohem mnohem adekvátnější model korelace než klasické archimédovské kopule s jedním parametrem. Cílem tohoto experimentu je tedy ověřit tento předpoklad.

Pro tento pokus jsem zvolil dvě eliptické kopule, Gaussovu a Studentovu (s různými počty stupňů volnosti ν); a dále Claytonu, Gumbelovu a Frankovu kopuli jako nejčastěji používané zástupce rodiny archimédovských kopulí. Jako referenční algoritmus pro srovnání vlivu korelací jsem ještě přidal i algoritmus UMDA (tedy algoritmus EDA, který nevyužívá žádné korelace – nicméně z formálního hlediska i na něho stále můžeme hledět jako na algoritmus EDA založený na kopulovém modelu, neboť situace, kdy neuvažujeme korelace, proměnné jsou zcela nezávislé, může být popsána součinnou kopulí).

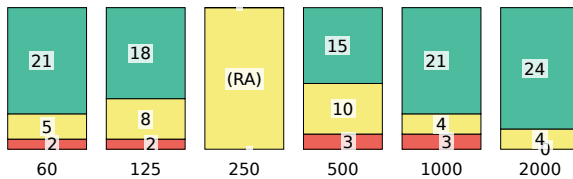
Získané výsledky potvrdily variantu s **Gaussovou kopulí** jako tu nejlepší. Dále je prezentováno, že mezi Gaussovou a Studentovou se stupni volnosti $\nu = 49$ velmi často není významný rozdíl.

5.1.3 Vliv počtu jedinců v populaci

Dalším významným parametrem evolučního algoritmu je počet jedinců tvořících populaci. Při příliš velké i příliš malé populaci je kvalita algoritmu horší.



Obrázek 5.2: Sumarizované výsledky experimentu zkoumajícího vliv různých typů **kopulí**. Legenda grafu viz kap. 5.1.1.



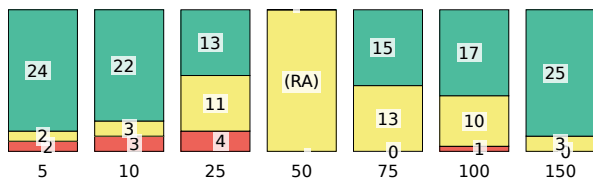
Obrázek 5.3: Sumarizované výsledky experimentu zkoumajícího vliv různých **velikostí populace**. Legenda grafu viz kap. 5.1.1.

Při velké populaci je to dáno hlavně tím, že se vzrůstajícím počtem jedinců, které je třeba vyhodnotit v každé generaci, se snižuje počet generací, které algoritmus může uskutečnit před tím, než vyčerpá povolený počet evaluací fitness funkce. Malá velikost populace sice umožní využít větší počet generací, ale už není dostatek jedinců na to, aby po selekci byla získána kvalitní subpopulace, a rychlost konvergence se velmi sníží.

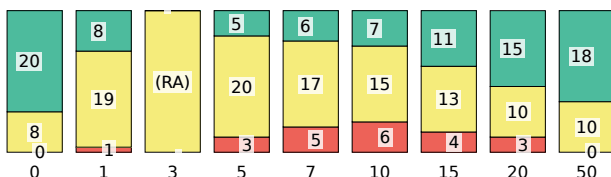
Nejlepších výsledků dosahuje populace tvořená **250 jedinci**.

5.1.4 Vliv velikosti selekce

Velikost selekce je parametr, který je do určité míry provázán s velikostí populace. Ve své disertaci selekci uvádím jako absolutní počet vybraných jedinců – tento způsob lépe postihuje vlastnosti algoritmu. Důvodem je nutnost použít dostatečný počet jedinců ve vybrané subpopulaci, aby mohlo dojít k vytvoření kvalitního pravděpodobnostního modelu. Zejména k získání adekvátních korelací (naučení parametrů pro kopuli) je potřebný dostatečně vysoký počet jedinců; pro naučení parametrů marginálních rozdělání počet jedinců až tak kritický není.



Obrázek 5.4: Sumarizované výsledky experimentu zkoumajícího vliv různých **velikostí selekce**. Legenda grafu viz kap. 5.1.1.



Obrázek 5.5: Sumarizované výsledky experimentu zkoumajícího vliv různých **obnov populace**. Legenda grafu viz kap. 5.1.1.

Jako nejlepší variantu jsem vyhodnotil tu s velikostí selekce **50 jedinců**.

5.1.5 Vliv obnovy populace

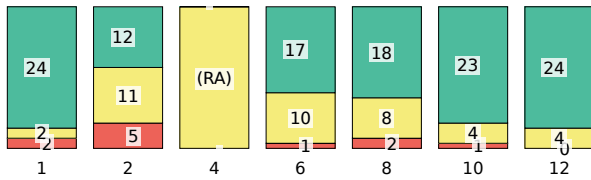
Nejlepších výsledků dosáhly varianty, které zachovávají jen **malý počet jedinců, (3–7)**. Při větším počtu zachovaných jedinců kvalita konvergence klesá.

Výrazně horší výsledek podala varianta, která obnovila úplně celou populaci (tzv. koncepce vymírání rodičovské populace).

5.1.6 Porovnání různých počtů ostrovů

Počet ostrovů je dalším důležitým parametrem u ostrovního modelu. Významným způsobem ovlivňuje kvalitu konvergence.

Pokud se použije jen jeden ostrov, jedná se o klasický sekvenční algoritmus (bez užití migrace).



Obrázek 5.6: Sumarizované výsledky experimentu zkoumajícího vliv různých **počtů ostrovů**. Legenda grafu viz kap. 5.1.1.

Protože je fixován celkový počet evaluací fitness funkce (pro všechny ostrovy), při větším počtu ostrovů se povolený počet evaluací na jeden ostrov sníží. Tedy na jednu stranu dostáváme výsledky z více ostrovů, ale na druhou stranu se počet možných generací každého ostrova snižuje. Kvůli tomu lze předpokládat, že u příliš velkého počtu ostrovů dojde ke zhoršení kvality optimalizace.

Lepších výsledků dosahují varianty s malým počtem ostrovů (vyjma případu s jedním ostrovem), pro sadu CEC 2013 jsou nejlepší **4 ostrovy**. Varianty s větším počtem ostrovů jsou vůči referenční variantě vždy výrazně horší.

Porovnání vůči sekvenční variantě

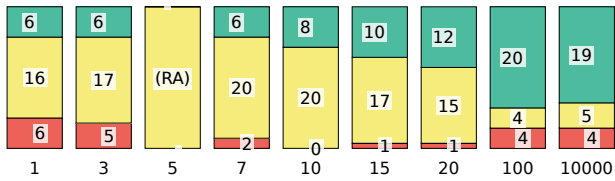
Zajímavé a důležité je porovnání referenční varianty RA (obsahující migraci) se sekvenční verzí algoritmu (použit jeden ostrov). RA je statisticky významně lepší pro naprostou většinu testovaných úloh.

5.1.7 Vliv perioda migrací

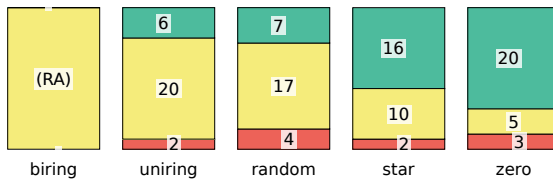
V případě migrace po každé generaci dochází k splývání modelů interagujících ostrovních modelů, čímž se sníží odlišnost mezi oběma modely, což způsobuje snížení diverzity a zhoršenou konvergenci.

V případě krátkých period mezi migracemi nebyl mezi jednotlivými variantami významný rozdíl. S prodlužující se hodnotou periody se začíná ukazovat výhodnost referenční varianty – tj. **5 generací**. Vůbec nejhorší výsledek byl zaznamenán pro nejdelší periody.

Protože varianty s dlouhou prodlevou mezi migracemi dosáhly nejhorších výsledků, je potvrzen význam vzájemného ovlivňování ostrovů prostřednictvím kombinace pravděpodobnostních modelů.



Obrázek 5.7: Sumarizované výsledky experimentu zkoumajícího vliv různých **period migrací**. Legenda grafu viz kap. 5.1.1.



Obrázek 5.8: Sumarizované výsledky experimentu zkoumajícího vliv různých **topologií**. Legenda grafu viz kap. 5.1.1.

5.1.8 Porovnání různých topologií

Jako nejlepší topologii jsem identifikoval **obousměrný prstenec**. Nicméně podobně jako u předešlého parametru, ani zde není volba kritická a i jiné topologie – jednosměrný prstenec a náhodná – dávají jen mírně zhoršený výkon.

5.2 Porovnání mCEDA s algoritmy jiných autorů

Tato kapitola je věnována srovnání navrženého algoritmu mCEDA s algoritmy, které publikovali jiní autoři.

Vycházím ze srovnání, která jsem již publikoval v [F, D, B], avšak ve finálním algoritmu používám aktualizované nastavení řídicích parametrů.

5.2.1 Porovnání mCEDA s existujícími kopulovými algoritmy EDA

V tabulkách 5.1a až 5.1c uvádím srovnání výsledků navrženého algoritmu mCEDA s výsledky uveřejněnými v článcích jiných autorů zabývajících se kopulovými algoritmy EDA na standardní sadě testovacích úloh. Pro každou funkci jsem použil 51 náhodných běhů. Výsledkem každého běhu je nejlepší nalezený jedinec (s nejmenší hodnotou fitness funkce, v našem případě s nejmenší odchylkou funkční hodnoty od známé hodnoty optima označenou jako chybová funkce). Z těchto 51 hodnot je stanovena střední hodnota prezentovaná ve výše uvedených tabulkách.

Z uvedených tabulek je patrné, že pro většinu standardních úloh algoritmus mCEDA ostatní algoritmy výrazně překonává – toto platí pro úlohy Sphere, Rastrigin, Ackley, Griewank. Na úloze Rosenbrock dosahují všechny porovnávané algoritmy podobných výsledků, rozdíly nejsou výrazné. Pouze pro úlohu Schwefel mají lepší výsledky porovnávané algoritmy.

Tabulka 5.1: Porovnání algoritmu mCEDA (střední hodnota, nejlepší je vyznačena tučně) pro klasické testovací úlohy, 10 dimenzí, vůči algoritmům: Copula Bayesian Network (CBN) [35]; Copula EDA (cEDA), Copula EDA of Dynamic K-S test (cEDA-KS) [57]; Clayton, Gumbel, Sn-EDA [25]; UMDAc, MIMICc, cEDA(empirical), cEDA(normal), cEDA(adaptive) [55].

(a) ukončení po 100000 vyhodnocení fitness funkce

	mCEDA	CBN
Rastrigin	0.00	2.39E+00
Ackley	7.34E-15	3.71E-02
Rosenbrock	6.58E+00	1.05E+01

(b) ukončení po 300000 vyhodnocení fitness funkce, část 1

	mCEDA	cEDA	cEDA-KS	Clayton	Gumbel	Sn-EDA
Sphere	2.49E-89	4.62E-08	1.16E-08	1.45E-07	3.59E-09	1.22E-09
Rastrigin	0.00	6.45E-08	2.60E-08	7.00E-08	5.49E-09	9.52E-09
Rosenbrock	6.53E+00	6.52E+00	7.05E+00	8.36E+00	6.62E+00	6.54E+00

(c) ukončení po 300000 vyhodnocení fitness funkce, část 2

	mCEDA	UMDAc	MIMICc	cEDA(empi.)	cEDA(norm.)	cEDA(adap.)
Sph.	2.49E-89	2.54E-23	6.88E-24	2.12E-10	1.17E-14	8.01E-13
Ros.	6.53E+00	7.56E+00	6.54E+00	6.78E+00	6.09E+00	4.24E+00
Sch.	4.56E+02	1.81E-02	1.88E-03	1.14E-04	1.14E-07	4.40E-07
Gri.	0.00	4.26E-03	1.90E-02	3.41E-02	1.69E-10	4.25E-13

5.2.2 Porovnání mCEDA s klasickými evolučními algoritmy na sadě CEC 2013

Druhým zde uvedeným srovnáním je porovnání navrženého algoritmu mCEDA s jinými evolučními algoritmy na sadě CEC 2013 pro 10 dimenzí: IPOP-CMA-ES [32], iCMAES-ILS [30], CMA-ES-RIS [5], SPSO2011 [51], PSO [56], GA-TPC [12], LaF [7], SPAM-AOS [13], DE/rand/1/bin [41], NMS-CS [26].

Srovnání je provedeno pomocí dvouvýběrového t-testu na hranici významnosti $\alpha = 0.05$. Na rozdíl od kap. 5.1 jsou výsledky experimentu tabelovány pro každou testovací úlohu. V tab. 5.2 je výsledek t-testu indikován pomocí symbolu \blacktriangledown (mCEDA je statisticky významně lepší), Δ (mCEDA je statisticky významně horší), nebo \circ (bez významného rozdílu). V prvním sloupci je umístěn algoritmus mCEDA jako referenční (RA), v dalších sloupcích jsou algoritmy jiných autorů. Řádky odpovídají 28 testovacím funkcím, kde je pro každý algoritmus uvedena střední hodnota chybové funkce a výsledek dvouvýběrového t-testu.

V zápatí tabulky jsou jednotlivé typy výsledku sumarizovány do hodnot B (počet úloh, kdy je RA statisticky významně lepší), W (počet úloh, kdy je RA statisticky významně horší), I (počet úloh bez statisticky významného rozdílu).

Z výsledných hodnot B , W , I je zřejmé, že ve srovnání s algoritmy CMA-ES-RIS, SPSO2011, PSO, LaF, SPAM-AOS, DE, NMS-CS je navržený algoritmus lepší ($B > W$). V případě IPOP-CMA-ES a iCMAES-ILS je $B < W$ a jsou proto lepší.

Při bližším pohledu na dosažené výsledky je evidentní, že algoritmus mCEDA je účinnější zejména pro složité úlohy. Pro první části sady CEC 2013 – unimodální úlohy (f1–f5) – jsou ostatní algoritmy lepší. Na druhé části – multimodální úlohy (f6–f20) – už navržený algoritmus překonává téměř všechny ostatní algoritmy, pouze algoritmy typu CMA-ES jsou lepší. Třetí částí srovnání jsou složené úlohy (f21–f28). Na těchto úlohách se navržený algoritmus mCEDA vyrovná i výkonnosti algoritmů typu CMA-ES nebo je překonává; ostatní typy algoritmů jednoznačně překonává.

Tabulka 5.2: Výsledky (střední hodnota, nejlepší výsledek tučně) a výsledek t-testu (na hranici významnosti $\alpha = 0.05$) mezi navrženým algoritmem a algoritmy jiných autorů (na sadě CEC 2013 pro 10 dimenzí): IPOP-CMA-ES [32], iCMAES-ILS [30], CMA-ES-RIS [5], SPSO2011 [51], PSO [56], GA-TPC [12], LaF [7], SPAM-AOS [13], DE/rand/1/bin [41], NMS-CS [26].

	mCEDA	ipop-cmaes	icmaes-ils	cmaes-ris	SPSO2011	PSO	GA-TPC	LaF	SPAM-AOS	DE	NMS-CS	
f1	2.8E-01	0.0 Δ	1.0E-08 Δ	0.0 Δ	0.0 Δ	0.0 Δ	0.0 Δ	0.0 Δ	0.0 Δ	1.0E-08 Δ	3.3E-10 Δ	
f2	3.9E+03	0.0 Δ	1.0E-08 Δ	0.0 Δ	1.6E+04 ∇	3.6E+04 ∇	0.0 Δ	1.2E+05 ∇	0.0 Δ	2.4E+03 \circ	1.9E+02 Δ	
f3	4.5E+06	0.0 Δ	1.0E-08 Δ	7.0E-01 Δ	2.3E+03 Δ	2.7E+05 \circ	0.0 Δ	6.0E+06 \circ	1.3E+00 Δ	1.4E+00 Δ	1.1E+05 Δ	
f4	3.3E+02	0.0 Δ	1.0E-08 Δ	0.0 Δ	1.3E+03 ∇	8.9E+03 ∇	0.0 Δ	9.5E+02 ∇	0.0 Δ	2.7E+01 Δ	3.8E-01 Δ	
f5	1.7E+00	0.0 Δ	1.0E-08 Δ	0.0 Δ	0.0 Δ	0.0 Δ	0.0 Δ	0.0 Δ	0.0 Δ	1.1E-13 Δ	1.0E-08 Δ	5.9E-04 Δ
f6	8.0E-01	0.0 Δ	3.9E+00 ∇	1.1E+00 \circ	9.8E+00 ∇	9.8E+00 ∇	0.0 Δ	9.7E+00 ∇	4.1E+00 ∇	3.3E+00 ∇	6.4E+00 ∇	
f7	2.5E-02	0.0 Δ	4.9E-06 Δ	5.3E+01 ∇	4.0E-01 \circ	2.1E+01 ∇	4.4E-02 \circ	1.3E+01 ∇	6.0E+01 ∇	1.4E-03 Δ	3.4E+01 ∇	
f8	2.0E+01	2.0E+01 Δ	2.0E+01 \circ	2.0E+01 Δ	2.0E+01 Δ	2.0E+01 Δ	2.0E+01 Δ	2.0E+01 \circ	2.0E+01 \circ	2.0E+01 \circ	2.0E+01 Δ	
f9	6.6E-01	5.8E-01 \circ	2.9E-01 Δ	3.6E+00 ∇	2.1E+00 ∇	4.8E+00 ∇	3.4E+00 ∇	3.7E+00 ∇	6.7E+00 ∇	1.1E+00 ∇	5.2E+00 ∇	
f10	4.2E-03	0.0 Δ	1.0E-08 Δ	1.2E-02 ∇	1.0E-01 ∇	3.0E-01 ∇	4.0E-02 ∇	3.7E-01 ∇	1.5E-02 ∇	4.9E-02 ∇	3.1E-02 ∇	
f11	9.6E-01	3.3E-01 Δ	4.8E-01 Δ	3.6E+00 ∇	4.8E+00 ∇	1.1E+01 ∇	2.7E-01 Δ	1.6E+00 ∇	6.3E+00 ∇	1.1E+00 \circ	1.6E+01 ∇	
f12	1.1E+00	9.8E-02 Δ	2.3E-01 Δ	1.3E+01 ∇	4.4E+00 ∇	1.4E+01 ∇	6.4E+00 ∇	1.2E+01 ∇	1.8E+01 ∇	8.2E+00 ∇	1.5E+01 ∇	
f13	2.0E+00	3.1E-01 Δ	3.3E-01 Δ	2.6E+01 ∇	5.5E+00 ∇	2.1E+01 ∇	1.0E+01 ∇	2.3E+01 ∇	3.6E+01 ∇	1.2E+01 ∇	3.1E+01 ∇	
f14	7.4E+01	2.7E+01 Δ	5.1E+01 \circ	1.0E+02 \circ	6.4E+02 ∇	8.3E+02 ∇	2.7E+01 Δ	2.1E+02 ∇	2.2E+02 ∇	2.2E+02 ∇	1.0E+03 ∇	
f15	1.5E+02	2.2E+01 Δ	4.4E+01 Δ	6.2E+02 ∇	5.2E+02 ∇	7.7E+02 ∇	8.3E+02 ∇	5.4E+02 ∇	1.0E+03 ∇	1.1E+03 ∇	9.0E+02 ∇	
f16	1.1E+00	1.2E+00 \circ	3.7E-01 Δ	1.6E-01 Δ	7.0E-01 Δ	5.0E-01 Δ	1.3E+00 ∇	1.1E+00 \circ	2.9E-01 Δ	1.0E+00 Δ	5.7E-01 Δ	
f17	2.4E+01	1.1E+01 Δ	1.1E+01 Δ	1.0E+01 Δ	1.7E+01 Δ	1.9E+01 Δ	1.1E+01 Δ	1.1E+01 Δ	1.6E+01 Δ	1.8E+01 Δ	1.9E+01 Δ	
f18	2.6E+01	1.1E+01 Δ	1.1E+01 Δ	3.0E+01 ∇	1.8E+01 ∇	1.8E+01 ∇	1.9E+01 Δ	2.2E+01 Δ	2.2E+01 Δ	3.2E+01 ∇	2.1E+01 Δ	
f19	8.3E-01	6.5E-01 Δ	7.0E-01 Δ	8.1E-01 \circ	8.0E-01 \circ	9.0E-01 \circ	5.3E-01 Δ	6.1E-01 Δ	7.3E-01 \circ	1.1E+00 ∇	8.8E-01 \circ	
f20	2.1E+00	2.8E+00 ∇	2.7E+00 ∇	4.2E+00 ∇	2.5E+00 ∇	3.4E+00 ∇	3.2E+00 ∇	2.8E+00 ∇	4.1E+00 ∇	2.4E+00 ∇	3.6E+00 ∇	
f21	4.0E+02	3.7E+02 Δ	2.2E+02 Δ	1.6E+02 Δ	4.0E+02 \circ	4.0E+02 \circ	2.9E+02 Δ	3.0E+02 Δ	2.9E+02 Δ	3.7E+02 Δ	4.0E+02 \circ	
f22	3.6E+01	7.3E+01 ∇	1.7E+02 ∇	2.4E+02 ∇	4.7E+02 ∇	9.1E+02 ∇	9.4E+01 ∇	2.4E+02 ∇	3.3E+02 ∇	2.2E+02 ∇	1.3E+03 ∇	
f23	4.8E+01	8.6E+01 ∇	4.1E+01 \circ	8.4E+02 ∇	3.5E+02 ∇	9.1E+02 ∇	9.3E+02 ∇	6.9E+02 ∇	1.5E+03 ∇	9.8E+02 ∇	1.3E+03 ∇	
f24	2.0E+02	2.1E+02 ∇	1.3E+02 Δ	1.2E+02 Δ	2.0E+02 \circ	2.1E+02 ∇	2.1E+02 ∇	2.1E+02 ∇	1.9E+02 ∇	2.0E+02 \circ	2.2E+02 ∇	
f25	2.0E+02	2.1E+02 ∇	1.9E+02 Δ	1.9E+02 \circ	2.0E+02 ∇	2.1E+02 ∇	2.2E+02 ∇	2.0E+02 ∇	2.2E+02 ∇	2.0E+02 ∇	2.1E+02 ∇	
f26	1.0E+02	2.0E+02 ∇	1.2E+02 ∇	1.6E+02 ∇	1.1E+02 \circ	2.0E+02 ∇	2.0E+02 ∇	1.4E+02 ∇	1.7E+02 ∇	1.7E+02 ∇	1.8E+02 ∇	
f27	3.0E+02	4.5E+02 ∇	3.3E+02 ∇	3.1E+02 ∇	3.0E+02 ∇	3.4E+02 ∇	4.4E+02 ∇	3.7E+02 ∇	3.8E+02 ∇	3.4E+02 ∇	4.4E+02 ∇	
f28	3.1E+02	3.0E+02 Δ	2.2E+02 Δ	2.1E+02 Δ	3.0E+02 Δ	3.0E+02 \circ	2.9E+02 Δ	3.2E+02 \circ	2.9E+02 \circ	2.9E+02 Δ	4.1E+02 ∇	
	(RA)	$B = 7$ $I = 2$ $W = 19$	$B = 5$ $I = 3$ $W = 20$	$B = 13$ $I = 4$ $W = 11$	$B = 14$ $I = 6$ $W = 8$	$B = 18$ $I = 4$ $W = 6$	$B = 13$ $I = 2$ $W = 13$	$B = 18$ $I = 4$ $W = 6$	$B = 16$ $I = 4$ $W = 8$	$B = 15$ $I = 4$ $W = 9$	$B = 17$ $I = 2$ $W = 9$	

6 Závěr

Hlavním cílem mojí disertační práce bylo prozkoumat, jak účinným nástrojem může být použití kopulí v pravděpodobnostním modelu algoritmů EDA ve spojení s migrací pravděpodobnostních modelů; jak navrhnout efektivní evoluční algoritmus na bázi EDA.

Všechny cíle definované v úvodu (kap. 1.1) jsou v disertaci splněny.

6.1 Teorie kopulí

Počáteční část disertace je věnována popisu teorie kopulí v takovém rozsahu a struktuře, který nepomíjí podstatné informace, ale přitom reflektuje stanovené cíle. V kapitole 2.3 jsem podrobně rozvinul odvození klasických formulí pro proces vzorkování Gaussovy kopule a uvedl efektivní metody pro vzorkování eliptických a archimédovských kopulí využívající Choleského dekompozice, respektive inverzní Laplaceovy transformace.

6.2 Přehled algoritmů EDA

V kapitole 3 je uveden přehled existujících algoritmů EDA. Nejprve je proveden přehled klasických algoritmů EDA s grafovými pravděpodobnostními modely. Dále jsou zde popsány přístupy k paralelnímu evolučnímu algoritmu.

Samostatnou částí je přehled algoritmů EDA využívajících kopule (CEDA) (kap. 3.2). Část algoritmů CEDA se zaměřuje na použití dvoudimenzionálních kopulí jako stavebních prvků v grafových modelech (C-vine, D-vine nebo hierarchické archimédovské kopule). Pro koncepci algoritmu s migrací modelů se jeví jako efektivnější použití vícedimenzionálních kopulí, v této oblasti výrazně převažují eliptické kopule.

6.3 Návrh paralelního algoritmu mCEDA

Jádrem disertace je 4. a 5. kapitola. Kapitola 4 se zabývá návrhem algoritmu mCEDA (migrating Copula-based Estimation of Distribution Algorithm). Paralelní algoritmus mCEDA je rozdělen na dvě části, na sekvenční jádro a na paralelní část s migrací modelů mezi ostrovy.

Sekvenční jádro je navrženo tak, aby při výstavbě pravděpodobnostního modelu bylo možné použít archimédovské i eliptické kopule. S tím souvisela volba vhodné metodiky pro výpočet jejich parametrů – používám korelační koeficient Kendallovo τ , který umožňuje jednoznačnou transformaci na parametry zvoleného typu kopule. Pro výpočet parametrů marginálních rozdělení jsem použil klasickou MLE optimalizaci, jako marginální rozdělení bylo použito normální rozdělení.

V paralelní části je řešena úloha migrace pravděpodobnostních modelů a jejich kombinace na rezidentním ostrově. Kombinace pravděpodobnostních modelů pro spojité úlohy je zcela nová zejména na úrovni kopulí, kde v rámci své práce prezentuji originální řešení. Při tvorbě pravděpodobnostního modelu využívám výhodu jeho separace na model závislosti mezi proměnnými, reprezentovaný kopulí, a model nezávislých marginálních rozdělení. Zatímco pro kombinace korelačních koeficientů τ a pro parametr marginálního rozdělení μ je použit základní adaptační vztah, parametr rozptylu σ je dán složitější funkcí, která je adekvátní vztahu publikovaného v [15].

6.4 Provedené experimenty

Důležitým úkolem bylo také nalezení parametrů navrženého paralelního algoritmu mCEDA, aby dosahoval co nejlepší výkonnosti. V kap. 5.1 jsem prozkoumal vliv sedmi klíčových parametrů, které spadají do dvou skupin – parametry sekvenční části a parametry vyplývající z paralelizace: typ kopule, velikost populace, velikost selekce, obnova populace, počet ostrovů, perioda migrací, komunikační topologie. Jednotlivé varianty algoritmu jsem porovnával pomocí série Wilcoxonových dvouvýběrových testů a Friedmanovy multinásobné komparace, více viz text disertační práce. Tuto analýzu jsem prováděl na sadě šesti klasických úloh a pro soubor CEC 2013 s 28 testovacími úlohami (rozhodný vliv měly experimenty pro rozsáhlejší testovací soubor úloh CEC 2013).

Následně jsem finálně nastavený algoritmus mCEDA porovnal s evolučními algoritmy jiných autorů. Nejprve jsem (v kap. 5.2.1) provedl srovnání s algoritmy EDA využívajícími kopule — algoritmus mCEDA ostatní komparované algoritmy významně překonává. Toto srovnání je uskutečněno jen na sadě klasických úloh.

Provedl jsem též srovnání mCEDA s algoritmy využívajícími rozdílná evoluční paradigmatata (kap. 5.2.2). Pro tyto algoritmy byly do-

stupné výsledky pro známou sadu úloh CEC 2013, což značně přispělo k významnosti statistické komparace s navrženým mCEDA algoritmem. Sada CEC 2013 se skládá ze tří typů úloh. Nejjednodušším typem úloh v sadě jsou unimodální úlohy, pro něž je algoritmus mCEDA překonán ostatními algoritmy. Druhým typem jsou multimodální úlohy – pro něž je navržený algoritmus lepší než většina ostatních, vyjma algoritmů typu CMA-ES. Třetím typem úloh jsou složené úlohy, pro něž dokáže navržený algoritmus mCEDA překonat i algoritmy typu CMA-ES.

Je tedy evidentní, že navržený algoritmus mCEDA je efektivní zejména pro komplexní optimalizační úlohy. Z tohoto zjištění vyplývají návrhy uvedené v kapitole 6.5, které lze chápat jako další možná obohacení použité koncepce mCEDA.

6.5 Další možná rozšíření práce

Koncepce disertační práce je postavena tak, aby se projevila kapacita základní algoritmičké struktury bez případné hybridizace a dalších přidavných technik. Vzhledem k dosaženým výsledkům a problematice konvergence u některého typu úloh je zřejmé, že bude vhodné některé techniky zlepšující konvergenci zavést:

1. Použít kombinaci analytických marginálních rozdělání a neparametrických rozdělání typu kernel, která usnadní řešení problémů zejména s asymetrickým definičním oborem testovaných funkcí.
2. Implementovat upravenou techniku CT-AVS (Correlation-Triggered Adaptive Variance Scaling) [19] pro modifikaci rozptylu marginálních rozdělání na základě korelace mezi tvarem testovací funkce a tvarem marginálního rozdělání.
3. V případě detekce předčasné konvergence zavést fenomén masivní mutace, popř. částečný restart algoritmu.
4. Použít dynamické nastavování některého řídicího parametru, např. populace jako je tomu u IPOP-CMA-ES algoritmu.

Autorovy publikace

2019

- [A] Hyrš, M.; Schwarz, J.: An Analysis of Control Parameters of Copula-based EDA Algorithm with Model Migration. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '19*, New York, NY, USA: ACM, 2019, ISBN 978-1-4503-6748-6, s. 259–260, doi:10.1145/3319619.3321910.

Podíl na publikaci: 80 %.

2016

- [B] Hyrš, M.; Schwarz, J.: Advanced Parallel Copula Based EDA. In *2016 IEEE Symposium Series on Computational Intelligence*, Institute of Electrical and Electronics Engineers, 2016, ISBN 978-1-5090-4239-5, s. 1–8, doi:10.1109/SSCI.2016.7850202.

Podíl na publikaci: 80 %.

- [C] Hyrš, M.; Schwarz, J.: Kopulové EDA algoritmy. In *Počítačové architektury & diagnostika PAD 2016*, Faculty of Information Technology BUT, 2016, ISBN 978-80-214-5376-0, s. 105–108.

Podíl na publikaci: 80 %.

2015

- [D] Hyrš, M.; Schwarz, J.: Elliptical and Archimedean Copulas in Estimation of Distribution Algorithm with Model Migration. In *Proceedings of the 7th International Joint Conference on Computational Intelligence (IJCCI 2015)*, SciTePress - Science and Technology Publications, 2015, ISBN 978-989-758-157-1, s. 212–219.

Podíl na publikaci: 80 %.

- [E] Hyrš, M.; Schwarz, J.: Elliptical and Archimedean Copulas in Estimation of Distribution Algorithm. In *MENDEL 2015 21st International Conference on Soft Computing*, ročník 2015, Faculty of Mechanical Engineering BUT, 2015, ISSN 1803-3814, s. 19–26.

Podíl na publikaci: 80 %.

2014

- [F] Hyrš, M.; Schwarz, J.: Multivariate Gaussian Copula in Estimation of Distribution Algorithm with Model Migration. In *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI) Proceedings*, Institute of Electrical and Electronics Engineers, 2014, ISBN 978-1-4799-4492-7, s. 114–119, doi:10.1109/FOCI.2014.7007815.

Podíl na publikaci: 80 %.

- [G] Hyrš, M.; Schwarz, J.: Estimation of distribution algorithm with copula probabilistic model: A short introduction. In *Proceedings of 20th international conference of softcomputing*, Faculty of Mechanical Engineering BUT, 2014, ISBN 978-80-214-4984-8, ISSN 1803-3814, s. 71–76.

Podíl na publikaci: 60 %.

Citace článků autora

Článek [D] citován v:

1. Chakrabarti, Arnab & Sen, Rituparna. (2019). Copula estimation for nonsynchronous financial data. arXiv: Statistical Finance.
2. Umeorah, Nneka & Mashele, Phillip & Ehrhardt, Matthias. (2019). Elliptical and Archimedean copula models: an application to the price estimation of portfolio credit derivatives.

Článek [F] citován v:

1. de Mello Junior, Harold & Martí, Luis & Abs da Cruz, Andre & Vellasco, Marley. (2016). Evolutionary algorithms and elliptical copulas applied to Continuous optimization problems. Information Sciences. 369. 10.1016/j.ins.2016.07.006.
2. Lane, Fergal & Azad, R. & Ryan, Conor. (2017). DICE: A New Family of Bivariate Estimation of Distribution Algorithms Based on Dichotomised Multivariate Gaussian Distributions. 670-685. 10.1007/978-3-319-55849-3_43.
3. Yang, Niansu & Huang, Y. & Ye, D. & Wang, X. & Li, H. & Li, S. & Dong, B. (2018). Modeling of Output Correlation of Multiple Wind Farms Based on Adaptive Multivariable Nonparametric Kernel Density Estimation. Zhongguo Dianji Gongcheng Xuebao/-Proceedings of the Chinese Society of Electrical Engineering. 38. 3805-3812. 10.13334/j.0258-8013.pcsee.171751.
4. Lane, Fergal & Azad, R. & Ryan, Conor. (2017). DICE: Exploiting All Bivariate Dependencies in Binary and Multary Search Spaces. Memetic Computing. 10.1007/s12293-017-0246-1.

Autorův životopis

Vzdělání

2013 – dosud

doktorské studium – Fakulta informačních technologií Vysokého učení technického v Brně

2011 – 2013

Ing. – Fakulta informačních technologií Vysokého učení technického v Brně,

obor Bioinformatika a biocomputing.

Diplomová práce *Predikce vlivu aminokyselinových mutací na sekundární strukturu proteinů*, vedoucí práce Bendl Jaroslav.

2008 – 2011

Bc. – Fakulta informačních technologií Vysokého učení technického v Brně, obor Informační technologie.

Bakalářská práce *Webová demonstrace Fourierovy transformace pro výuku zpracování signálů*, vedoucí práce Černocký Jan.

2008

maturita – Gymnázium Havlíčkův Brod

Pracovní zkušenosti

2017 – dosud

Scripter, Bohemia Interactive a.s.

Vývoj hry *Vigor* v prostředí Unreal Engine 4 Blueprints.

2013 – 2017

doktorand, Fakulta informačních technologií VUT v Brně.

Vedení cvičení a projektů v předmětech:

- Aplikované evoluční algoritmy,
- Funkcionální a logické programování,
- Prvky počítačů.

Literatura

- [1] AAS, K. *Modelling the dependence structure of financial assets: A survey of four copulas*. December 01, date, 2004.
- [2] BALUJA, S. *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. 1994.
- [3] BALUJA, S. a 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, s. 30–38.
- [4] BOSMAN, P. A. N. a THIERENS, D. Expanding from Discrete to Continuous Estimation of Distribution Algorithms: The IDEA. In: SCHOENAUER, M., DEB, K., RUDOLPH, G., YAO, X., LUTTON, E. et al., ed. *Parallel Problem Solving from Nature PPSN VI*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, s. 767–776. ISBN 978-3-540-45356-7.
- [5] CARAFFINI, F., IACCA, G., NERI, F., PICINALI, L. a MININNO, E. A CMA-ES super-fit scheme for the re-sampled inheritance search. In: *2013 IEEE Congress on Evolutionary Computation*. June 2013, s. 1123–1130. DOI: 10.1109/CEC.2013.6557692. ISSN 1089-778X.
- [6] CHANG, C. a WANG, L.-f. A Multi-population Parallel Estimation of Distribution Algorithms Based on Clayton and Gumbel Copulas. In: DENG, H., MIAO, D., LEI, J. a WANG, F., ed. *Artificial Intelligence and Computational Intelligence*. Springer Berlin Heidelberg, 2011, sv. 7002, s. 634–643. Lecture Notes in Computer Science. ISBN 978-3-642-23880-2.
- [7] CHEN, S. a GONZALEZ-FERNANDEZ, Y. *Leaders and Followers on the CEC2013 Real-Parameter Optimization Benchmark Functionss*. 2015. Technical Report School of Information Technology York University.
- [8] DE BONET, J. S., ISBELL, C. L. a VIOLA, P. A. MIMIC: Finding Optima by Estimating Probability Densities. In: *Advances in Neural Information Processing Systems*. The MIT Press, Cambridge, 1997, sv. 9, s. 424–430.
- [9] DELAOSSA, L., GÁMEZ, J. A. a PUERTA, J. M. Migration of Probability Models Instead of Individuals: An Alternative When Applying the Island Model to EDAs. In: *Parallel Problem Solving from Nature - PPSN VIII*. Springer, 2004, 3242 of LNCS, s. 242–252. Lecture Notes in Computer Science. ISBN 978-3-540-23092-2.
- [10] DELAOSSA, L., GÁMEZ, J. A. a PUERTA, J. M. Improving model combination through local search in parallel univariate EDAs.

- In: *Congress on Evolutionary Computation*. IEEE, 2005, sv. 2, s. 1426–1433. ISBN 0-7803-9363-5.
- [11] EL ABD, M. a KAMEL, M. S. A cooperative particle swarm optimizer with migration of heterogeneous probabilistic models. *Swarm Intelligence*. 2010, sv. 4, č. 1, s. 57–89. ISSN 1935-3820.
- [12] ELSAYED, S. M., SARKER, R. A. a ESSAM, D. L. A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization. In: *2013 IEEE Congress on Evolutionary Computation*. June 2013, s. 356–360. DOI: 10.1109/CEC.2013.6557591. ISSN 1089-778X.
- [13] EPITROPAKIS, M. G., CARAFFINI, F., NERI, F. a BURKE, E. K. A Separability Prototype for Automatic Memes with Adaptive Operator Selection. In: *Foundations of Computational Intelligence (FOCI), 2014 IEEE Symposium on*. Dec 2014, s. 70–77. DOI: 10.1109/FOCI.2014.7007809.
- [14] ETXEBERRIA, R. a LARRAÑAGA, P. Global optimization using Bayesian networks. In: *Second Symposium on Artificial Intelligence (CIMA-F-99)*. 1999, s. 332–339.
- [15] FRÜHWIRTH SCHNATTER, S. *Finite Mixture and Markov Switching Models*. New York: Springer, 2006. ISBN 978-1-4419-2194-9.
- [16] GAO, Y. Multivariate Estimation of Distribution Algorithm with Laplace transform Archimedean copula. In: IEEE. *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*. 2009, s. 1–5. ISBN 978-1-4244-4994-1.
- [17] GONZALEZ-FERNANDEZ, Y. a SOTO, M. *A Survey of Estimation of Distribution Algorithms Based on Copulas*. 2012. Technical Report ICIMAF 2012-679.
- [18] GONZALEZ FERNANDEZ, Y. a SOTO, M. copulaedas: An R Package for Estimation of Distribution Algorithms Based on Copulas. *Journal of Statistical Software*. 2014, sv. 58, č. 9, s. 1–34. Dostupné z: <http://www.jstatsoft.org/v58/i09/>.
- [19] GRAHL, J., BOSMAN, P. A. a ROTHLAUF, F. The Correlation-triggered Adaptive Variance Scaling IDEA. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2006, s. 397–404. GECCO '06. DOI: 10.1145/1143997.1144071. ISBN 1-59593-186-4. Dostupné z: <http://doi.acm.org/10.1145/1143997.1144071>.
- [20] GUO, X., WANG, L.-F., ZENG, J.-C. a ZHANG, X. VQ codebook design algorithm based on copula Estimation of Distribution Algorithm. In:

- IEEE. *Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on*. 2011, s. 178–181.
- [21] HARIK, G., LOBO, F. a GOLDBERG, D. The compact genetic algorithm. *Evolutionary Computation, IEEE Transactions on*. Nov 1999, sv. 3, č. 4, s. 287–297. DOI: 10.1109/4235.797971. ISSN 1089-778X.
- [22] HAUSCHILD, M. a PELIKAN, M. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*. 2011, sv. 1, č. 3, s. 111 – 128. ISSN 2210-6502.
- [23] JAROŠ, J. a SCHWARZ, J. Parallel BMDA with an Aggregation of Probability Models. In: *Proceeding of 2009 IEEE Congress on Evolutionary Computation*. IEEE Computational Intelligence Society, 2009, s. 1683–1690. ISBN 978-1-4244-2959-2. Dostupné z: <https://www.fit.vut.cz/research/publication/8948>.
- [24] JAWORSKI, P., DURANTE, F., HARDLE, W. K. a RYCHLIK, T. *Copula theory and its applications : proceedings of the workshop held in Warsaw 25-26 September 2009*. Heidelberg: Springer, 2010. Lecture notes in statistics - proceedings. ISBN 978-3-642-12464-8.
- [25] JIA, B., WANG, L.-F. a CUI, Z. Copula for Estimation of Distribution Algorithm Based on Goodness-of-fit test. *Journal of Theoretical and Applied Information Technology*. 2013, sv. 47, č. 3, s. 1128–1132. ISSN 1992-8645.
- [26] JOVANOVIC, R., KAIS, S. a ALHARBI, F. H. Cuckoo Search Inspired Hybridization of the Nelder-Mead Simplex Algorithm Applied to Optimization of Photovoltaic Cells. *Applied Mathematics & Information Sciences*. Natural Sciences Publishing. May 2016, sv. 10, č. 3, s. 961–973. DOI: 10.18576/amis/100314. ISSN 2325-0399. Dostupné z: <http://dx.doi.org/10.18576/amis/100314>.
- [27] LARRAÑAGA, P., ETXEBERRIA, R., LOZANO, J. A., PEÑA, J. a NA, J. M. P. *Optimization by learning and simulation of Bayesian and Gaussian networks*. 1999.
- [28] LARRAÑAGA, P., LOZANO, J. A. a BENGOTXEA, E. *Estimation of Distribution Algorithms based on Multivariate Normal and Gaussian Networks*. Department of Computer Science and Artificial Intelligence, University of the Basque Country, 2001.
- [29] LIANG, J., QU, B., SUGANTHAN, P. a HERNÁNDEZ DÍAZ, A. G. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*. 2013, sv. 201212.

- [30] LIAO, T. a STÜTZLE, T. Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization. In: *2013 IEEE Congress on Evolutionary Computation*. June 2013, s. 1938–1944. DOI: 10.1109/CEC.2013.6557796. ISSN 1089-778X.
- [31] LIU, B., CHENG, S. a SHI, Y. Student’s t Distribution based Estimation of Distribution Algorithms for Derivative-free Global Optimization. *CoRR*. 2016, abs/1608.03757. Dostupné z: <http://arxiv.org/abs/1608.03757>.
- [32] LOSHCHILOV, I. CMA-ES with restarts for solving CEC 2013 benchmark problems. In: *2013 IEEE Congress on Evolutionary Computation*. June 2013, s. 369–376. DOI: 10.1109/CEC.2013.6557593. ISSN 1089-778X.
- [33] MAI, J. a SCHERER, M. *Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications*. London: Imperial College Press, 2012. Series in quantitative finance. ISBN 9781848168749.
- [34] MELCHIORI, M. R. Tools for sampling multivariate archimedean copulas. *YieldCurve, April*. 2006.
- [35] MÉNDEZ, M. a LANDA, R. An EDA based on Bayesian networks constructed with Archimedean copulas. In: *2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)*. 2012, s. 188–193.
- [36] MÜHLENBEIN, H. The Equation for Response to Selection and Its Use for Prediction. *Evolutionary computation*. Únor 1997, sv. 5, s. 303–46. DOI: 10.1162/evco.1997.5.3.303.
- [37] MUELAS, S., MENDIBURU, A., LATORRE, A. a PEÑA, J.-M. Distributed Estimation of Distribution Algorithms for continuous optimization: How does the exchanged information influence their behavior? *Information Sciences*. 2014, sv. 268, s. 231 – 254. DOI: <https://doi.org/10.1016/j.ins.2013.10.026>. ISSN 0020-0255. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0020025513007536>.
- [38] NELSEN, R. B. *An Introduction to Copulas*. New York: Springer, 2006. Springer Series in Statistics. ISBN 0387286594.
- [39] PELIKAN, M., GOLDBERG, D. a CANTÚ PAZ, E. BOA: The Bayesian optimization algorithm. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*. 1999, I, s. 525–532 also IlliGAL Report no. 99003.
- [40] PELIKAN, M. a MÜHLENBEIN, H. The Bivariate Marginal Distribution Algorithm. In: *Advances in Soft Computing*. Springer London, 1999, s. 521–535. ISBN 978-1-85233-062-0.

- [41] QIN, A. K. a LI, X. Differential evolution on the CEC-2013 single-objective continuous optimization testbed. In: *2013 IEEE Congress on Evolutionary Computation*. June 2013, s. 1099–1106. DOI: 10.1109/CEC.2013.6557689. ISSN 1089-778X.
- [42] RUDLOF, S. a KÖPPEN, M. Stochastic Hill Climbing with Learning by Vectors of Normal Distributions. In: 1996, s. 60–70.
- [43] SALINAS-GUTIÉRREZ, R., HERNÁNDEZ-AGUIRRE, A., RIVERA-MERAZ, M. J. J. a VILLA-DIHARCE, E. R. Using Gaussian Copulas in Supervised Probabilistic Classification. In: *Soft Computing for Intelligent Control and Mobile Robotics*. Springer, 2011, s. 355–372.
- [44] SALINAS-GUTIÉRREZ, R., HERNÁNDEZ-AGUIRRE, A. a VILLA-DIHARCE, E. R. Using Copulas in Estimation of Distribution Algorithms. In: *MICAI 2009: Advances in Artificial Intelligence*. Springer Berlin Heidelberg, 2009, sv. 5845, s. 658–668. Lecture Notes in Computer Science. ISBN 978-3-642-05257-6.
- [45] SALINAS-GUTIÉRREZ, R., HERNÁNDEZ-AGUIRRE, A. a VILLA-DIHARCE, E. R. Dependence Trees with Copula Selection for Continuous Estimation of Distribution Algorithms. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2011, s. 585–592. GECCO '11. ISBN 978-1-4503-0557-0.
- [46] SALINAS-GUTIÉRREZ, R., HERNÁNDEZ-AGUIRRE, A. a VILLA-DIHARCE, E. R. Estimation of Distribution Algorithms Based on Copula Functions. In: *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2011, s. 795–798. GECCO '11. ISBN 978-1-4503-0690-4.
- [47] SCHWARZ, J. a JAROŠ, J. Parallel Bivariate Marginal Distribution Algorithm with Probability Model Migration. In: *Linkage in Evolutionary Computation*. Springer Berlin Heidelberg, 2008, sv. 157, s. 3–23. Studies in Computational Intelligence. ISBN 978-3-540-85067-0.
- [48] SKLAR, A. Fonctions de Répartition à n dimensions et leurs Marges. *Publications de l'Institut de Statistique de l'Université de Paris*. 1959, sv. 8, s. 229–231.
- [49] SKLAR, A. Random Variables, Joint Distribution Functions, and Copulas. *Kybernetika*. 1973, sv. 9, č. 6, s. 449–460.
- [50] SOTO, M., GONZÁLEZ-FERNÁNDEZ, Y. a OCHOA, A. Modeling with Copulas and Vines in Estimation of Distribution Algorithms. *CoRR*. 2012, abs/1210.5500.
- [51] UGOLOTTI, R. a CAGNONI, S. A Fair Comparison Between Standard PSO Versions. In: ROSSI, F., MAVELLI, F., STANO, P. a CAIVANO, D.,

- ed. *Advances in Artificial Life, Evolutionary Computation and Systems Chemistry: 10th Italian Workshop, WIVACE 2015, Bari, Italy, September 22-25, 2015, Revised Selected Papers*. Cham: Springer International Publishing, 2016, s. 3–14. ISBN 978-3-319-32695-5.
- [52] WANG, L.-F., GUO, X., ZENG, J.-C. a HONG, Y. Using Gumbel copula and empirical marginal distribution in Estimation of Distribution Algorithm. In: IEEE. *Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on*. 2010, s. 583–587.
- [53] WANG, L.-F., ZENG, J.-C. a HONG, Y. Estimation of Distribution Algorithm based on copula theory. In: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*. May 2009, s. 1057–1063.
- [54] WANG, L.-F., ZENG, J.-C., HONG, Y. a GUO, X. Copula Estimation of Distribution Algorithm sampling from Clayton copula. *Journal of Computational Information Systems*. 2010, sv. 6, č. 7, s. 2431–2440.
- [55] YANG, R., WANG, L.-F. a ZENG, J.-C. Adaptive Variance Scaling in Clayton Copula EDA. In: YANG, Y. a MA, M., ed. *Green Communications and Networks: Proceedings of the International Conference on Green Communications and Networks (GCN 2011)*. Dordrecht: Springer Netherlands, 2012, s. 1295–1302. ISBN 978-94-007-2169-2.
- [56] ZAMBRANO BIGIARINI, M., CLERC, M. a ROJAS, R. Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements. In: *2013 IEEE Congress on Evolutionary Computation*. June 2013, s. 2337–2344. DOI: 10.1109/CEC.2013.6557848. ISSN 1089-778X.
- [57] ZHAO, H. a WANG, L.-F. Marginal Distribution in Copula Estimation of Distribution Algorithm Based Dynamic K-S test. In: *IJCSI International Journal of Computer Science Issues*. 2012, č. 3, s. 507–514.