



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**PARALELNÍ EVOLUČNÍ ALGORITMUS EDA
VYUŽÍVAJÍCÍ TEORII KOPULÍ**

PARALLEL EVOLUTIONARY ALGORITHM EDA BASED ON COPULAS

DISERTAČNÍ PRÁCE

PHD THESIS

AUTOR PRÁCE

AUTHOR

Ing. MARTIN HYRŠ

ŠKOLITEL

SUPERVISOR

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.

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.

Klíčová slova

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

Keywords

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

Citace

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

Paralelní evoluční algoritmus EDA využívající teorii kopulí

Prohlášení

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně pod vedením svého školitele doc. Ing. Josefa Schwarze, CSc. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Martin Hyrš
17. srpna 2020

Poděkování

Kamarádům a kolegům z UPSY za spoustu skvělých zážitků, přátelskou atmosféru a cenné postřehy. Svému školiteli za neskonalou trpělivost zejména během dokončování. Rodičům za nekončící podporu.

Výzkum prováděný v rámci disertační práce byl podporován z projektů FIT-S-14-2297 a FIT-S-17-3994.

Obsah

1	Úvod	8
1.1	Cíle disertační práce	9
1.2	Struktura disertační práce	9
2	Úvod do teorie kopulí	10
2.1	Rozdělení spojitě náhodné veličiny	10
2.1.1	Vzorkování náhodných čísel	12
2.2	Kopule	13
2.2.1	Definice kopule	13
2.2.2	Sklarův teorém	13
2.2.3	Vizualizace kopulí	14
2.2.4	Typy kopulí	15
2.3	Vzorkování sdruženého rozdělení pomocí kopulí	18
2.4	Vzorkování kopulí	19
2.4.1	Využití podmíněné pravděpodobnosti	19
2.4.2	Vzorkování Gaussovy kopule pomocí vylučovací metody	21
2.4.3	Algoritmy pro efektivní vzorkování kopulí	23
3	Přehled existujících algoritmů EDA	25
3.1	Přehled sekvenčních algoritmů typu EDA	26
3.1.1	Modely s nezávislými proměnnými	26
3.1.2	Párové závislosti	27
3.1.3	Vícenásobné závislosti	28
3.1.4	Algoritmy EDA pro spojitě problémy	28
3.2	Algoritmy EDA využívající kopule (CEDA)	29
3.2.1	Li-Fang Wang et al.	30
3.2.2	Rogelio Salinas-Gutiérrez et al.	30
3.2.3	Marta Soto et al.	30
3.2.4	Další autoři	31
3.2.5	Shrnutí algoritmů CEDA	31
3.3	Paralelní evoluční algoritmy	32
3.3.1	Algoritmy EDA s migrací pravděpodobnostní modelů	32
3.4	Porovnávání variant algoritmů	33
3.4.1	Klasické testovací úlohy	34
3.4.2	Sada CEC 2013	37
3.5	Testování statistických hypotéz	37
3.5.1	Dvouvýběrový t-test	39
3.5.2	Wilcoxonův dvouvýběrový test	40

3.5.3	Vícenásobné testy typu Friedman	40
4	Návrh efektivního kopulového algoritmu EDA – mCEDA	42
4.1	Sekvenční jádro navrženého algoritmu	42
4.1.1	Parametry marginálních rozdělání	42
4.1.2	Parametry kopule	43
4.1.3	Vzorkování kopule – implementace	47
4.1.4	Vzorkování marginálního rozdělání – implementace	47
4.1.5	Ošetření neplatných výpočetních operací	48
4.2	Paralelní část navrženého algoritmu	49
4.2.1	Kombinace pravděpodobnostních modelů	50
4.2.2	Komunikační topologie	51
5	Provedené experimenty a jejich vyhodnocení	53
5.1	Nastavení řídicích parametrů	53
5.1.1	Postup vyhodnocení	54
5.1.2	Porovnání vhodnosti užití různých typů kopulí	56
5.1.3	Vliv počtu jedinců v populaci	59
5.1.4	Vliv velikosti selekce	62
5.1.5	Vliv obnovy populace	65
5.1.6	Porovnání různých počtů ostrovů	68
5.1.7	Vliv periody migrací	71
5.1.8	Porovnání různých topologií	74
5.2	Porovnání mCEDA s algoritmy jiných autorů	77
5.2.1	Porovnání mCEDA s existujícími kopulovými algoritmy EDA	77
5.2.2	Porovnání mCEDA s klasickými evolučními algoritmy na sadě CEC 2013	80
6	Závěr	84
6.1	Teorie kopulí	84
6.2	Přehled algoritmů EDA	84
6.3	Návrh paralelního algoritmu mCEDA	85
6.4	Provedené experimenty	85
6.5	Další možná rozšíření práce	86
	Autorovy publikace	87
	Literatura	88
	A Obsah přiloženého CD	95

Seznam obrázků

2.1	Příklad distribuční funkce a funkce hustoty pravděpodobnosti.	11
2.2	Příklad kopulí.	14
2.3	Příklady grafů pro vizualizaci kopulí.	15
2.4	Vzorky získané z dvojrozměrných archimédovských kopulí s korelací ρ	17
2.5	Vzorky získané z dvojrozměrných eliptických kopulí s korelací ρ	17
3.1	Schematické znázornění grafových modelů užívaných v EDA.	27
3.2	Zobrazení testovací funkce Sphere.	35
3.3	Zobrazení testovací funkce Ackley.	35
3.4	Zobrazení testovací funkce Schwefel.	35
3.5	Zobrazení testovací funkce Rosenbrock.	36
3.6	Zobrazení testovací funkce Rastrigin.	36
3.7	Zobrazení testovací funkce Griewank.	36
4.1	Vytváření pravděpodobnostního modelu pomocí kopule.	44
4.2	Ilustrace použití marginálních rozdělení a kopule v algoritmech EDA.	45
4.3	Schéma algoritmu EDA s migrací modelů.	49
4.4	Příklad kombinace modelů	51
4.5	Ilustrace různých topologií.	52
5.1	Schéma vyhodnocení výsledků na jednotlivých úlohách.	55
5.2	Vliv různých typů kopulí – výsledky pro klasické testovací úlohy.	57
5.3	Vliv různých typů kopulí – výsledky pro sadu CEC 2013.	57
5.4	Vliv různých velikostí populace – výsledky pro klasické testovací úlohy.	60
5.5	Vliv různých velikostí populace – výsledky pro sadu CEC 2013.	60
5.6	Vliv různých velikostí selekce – výsledky pro klasické testovací úlohy.	63
5.7	Vliv různých velikostí selekce – výsledky pro sadu CEC 2013.	63
5.8	Vliv různých obnov populace – výsledky pro klasické testovací úlohy.	66
5.9	Vliv různých obnov populace – výsledky pro sadu CEC 2013.	66
5.10	Vliv různých počtů ostrovů – výsledky pro klasické testovací úlohy.	69
5.11	Vliv různých počtů ostrovů – výsledky pro sadu CEC 2013.	69
5.12	Vliv různých period migrací – výsledky pro klasické testovací úlohy.	72
5.13	Vliv různých period migrací – výsledky pro sadu CEC 2013.	72
5.14	Vliv různých topologií – výsledky pro klasické testovací úlohy.	75
5.15	Vliv různých topologií – výsledky pro sadu CEC 2013.	75
5.16	Průběh evoluce algoritmu mCEDA na sadě klasických testovacích úloh.	79
5.17	Histogram výsledků testu statistické významnosti rozdílu mezi navrženým algoritmem mCEDA a algoritmy jiných autorů.	80
5.18	Průběh evoluce algoritmu mCEDA na sadě CEC 2013, 10 dimenzí.	83

Seznam tabulek

2.1	Příklad archimédovských kopulí a jejich generátorů.	16
3.1	Seznam funkcí testovací sady CEC 2013.	38
5.1	Vliv různých typů kopulí – výsledky pro klasické testovací úlohy.	58
5.2	Vliv různých typů kopulí – výsledky pro sadu CEC 2013.	58
5.3	Vliv různých velikostí populace – výsledky pro klasické testovací úlohy.	61
5.4	Vliv různých velikostí populace – výsledky pro sadu CEC 2013.	61
5.5	Vliv různých velikostí selekce – výsledky pro klasické testovací úlohy.	64
5.6	Vliv různých velikostí selekce – výsledky pro sadu CEC 2013.	64
5.7	Vliv různých obnov populace – výsledky pro klasické testovací úlohy.	67
5.8	Vliv různých obnov populace – výsledky pro sadu CEC 2013.	67
5.9	Vliv různých počtů ostrovů – výsledky pro klasické testovací úlohy.	70
5.10	Vliv různých počtů ostrovů – výsledky pro sadu CEC 2013.	70
5.11	Vliv různých period migrací – výsledky pro klasické testovací úlohy.	73
5.12	Vliv různých period migrací – výsledky pro sadu CEC 2013.	73
5.13	Vliv různých topologií – výsledky pro klasické testovací úlohy.	76
5.14	Vliv různých topologií – výsledky pro sadu CEC 2013.	76
5.15	Porovnání algoritmu mCEDA vůči algoritmům jiných autorů pro klasické testovací úlohy.	78
5.16	Výsledky navrženého algoritmu mCEDA na sadě klasických testovacích úloh pro 10 dimenzí, výsledky z 51 náhodných běhů.	78
5.17	Porovnání algoritmu mCEDA vůči algoritmům jiných autorů na sadě CEC 2013.	81
5.18	Výsledky navrženého algoritmu mCEDA na sadě CEC 2013.	82

Seznam algoritmů

2.1	Pseudokód algoritmu pro vylučovací metodu.	12
2.2	Vzorkování sdruženého rozdělení.	18
2.3	Vzorkování dvourozměrné Gaussovy kopule.	21
2.4	Vzorkování vícerozměrné Gaussovy kopule pomocí vylučovací metody.	23
2.5	Vzorkování archimédovských kopulí.	24
2.6	Vzorkování Gaussovy kopule.	24
2.7	Vzorkování Studentovy kopule.	24
3.1	Pseudokód klasického algoritmu EDA.	25
4.1	Pseudokód sekvenčního kopulového algoritmu EDA.	43
4.2	Vzorkování normálního rozdělení $\mathcal{N}(\mu, \sigma)$ s ošetřením mezí proměnných $x \in \langle MIN; MAX \rangle$	47
4.3	Pseudokód algoritmu EDA s migrací modelů.	50

Seznam symbolů

- B počet úloh, kdy je referenční algoritmus statisticky významně lepší
- $C, C(u_1, \dots, u_D)$ kopule
- $c(\cdot)$ hustota kopule
- D maximální počet dimenzí
- d index dimenze, $1 \leq d \leq D$
- $F(\cdot)$ distribuční funkce
- $F_d(\cdot)$ marginální distribuční funkce
- $F^{-1}(\cdot)$ inverzní distribuční funkce
- F_f testová statistika Friedmanova testu
- $f(\cdot)$ hustota pravděpodobnosti
- $f_d(\cdot)$ hustota pravděpodobnosti marginálního rozdělení
- g číslo generace
- I počet úloh bez statisticky významného rozdílu mezi algoritmy
- i, j univerzálně použitelné indexy bez specifického významu
- L dolní trojúhelníkovitá matice
- M, M_R, M_I model, rezidentní model, imigrantský model
- $\langle MIN; MAX \rangle^D$ prohledávaný prostor, $\langle MIN; MAX \rangle \subset \mathbb{R}$
- N velikost populace (počet jedinců)
- N' velikost (počet jedinců) vybrané subpopulace
- N'' počet jedinců zachovaných při obnově populace
- $\mathcal{N}(\mu, \sigma)$ normální rozdělení
- $O(g)$ potomci v generaci g
- $P(g)$ populace v generaci g

- p index jedince v populaci, $1 \leq p \leq N$
- $p(\cdot)$ pravděpodobnost náhodného jevu
- RA referenční algoritmus
- \mathbb{R} množina všech reálných čísel
- R korelační matice
- R_j průměrné pořadí ve Friedmanově testu
- $S(P(g))$ vybraná populace v generaci g
- $U(a, b)$ rovnoměrné rozdělení
- u, v, u_d argumenty pro kopuli
- W počet úloh, kdy je referenční algoritmus statisticky významně horší
- \mathbb{X}, \mathbb{Y} náhodné veličiny (obecně)
- $X(d)$ náhodná veličina popisující proměnnou x_d
- \mathbf{x} jedinec
- \mathbf{x}^S jedinec ve vybrané subpopulaci, $\mathbf{x}^S \in S(P(g))$
- x_d proměnná v jedinci, $\mathbf{x} = (x_1, \dots, x_D)$
- $(x_d)_p$ d -tá proměnná v p -tém jedinci
- β učící koeficient
- θ parametr archimédovské kopule
- μ střední hodnota
- ν počet stupňů volnosti
- ρ korelace
- $\rho_{i,j}$ korelace mezi proměnnými s indexy i, j
- σ směrodatná odchylka
- τ korelační koeficient Kendallovo τ
- Φ normované normální rozdělení $\Phi \equiv \mathcal{N}(0, 1)$
- ▼ referenční algoritmus je lepší než alternativní varianta
- △ referenční algoritmus je horší než alternativní varianta
- mezi algoritmy (referenční a alternativní) není statisticky významný rozdíl

Kapitola 1

Úvod

V této disertační práci se zabývám návrhem, implementací a testováním paralelního algoritmu EDA s kopulemi s využitím migrace parametrů pravděpodobnostního modelu mezi distribuovanými subpopulacemi tvořícími celkovou populaci. Tento algoritmus by dle předpokladu měl výrazně profitovat z užití kopulí.

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).

Evoluční algoritmy obecně jsou výpočetním mechanismem inspirovaným přírodními procesy. Podobně jako u přirozeného výběru v živé přírodě, i v evolučních algoritmech 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 další generace, typicky pomocí vzájemného křížení a mutace.

V algoritmech EDA je k vytvoření nové populace použit modifikovaný přístup: z vybraných jedinců je odvozen pravděpodobnostní model a nová populace je vytvářena vzorkováním jedinců ze získaného rozložení pravděpodobnosti. V procesu evoluce je model zpřesňován z aktuálně navzorkované 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, jako jsou například Bayesovské grafové modely. Tato problematika je propracována 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 [67]. V této práci půjde především o aplikační rozpracování této koncepce pro využití v optimalizačních úlohách reálné domény s využitím algoritmů EDA. Kopule jsou už dlouho využívány ve finančnictví zejména pro modelování investičních rizik. 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.

Dalším aspektem zkoumaným v této práci je použití paralelizace při návrhu EDA algoritmu. Přímočarou aplikací paralelismu (kterou se ale v této práci nezabývám) je použití paralelní vyhodnocování kandidátních jedinců za účelem snížení výpočetního času. Pokročilejší variantou je použití tzv. „ostrovního modelu“ – celková populace je rozdělena na několik

částí (tzv. ostrovy), na nichž evoluce probíhá nezávisle. Výměna informací mezi ostrovy probíhá prostřednictvím migrace – posílání jedinců mezi sousedními ostrovy. Cílem je primárně zlepšit konvergenci a celkovou dobu výpočtu.

Méně prozkoumanou koncepcí je migrace pravděpodobnostních modelů – tedy mezi ostrovy je namísto jedinců přenášen pravděpodobnostní model. 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.

1.2 Struktura disertační práce

Text práce je rozdělen do sedmi kapitol. V kapitole 2 seznamuji čtenáře se základními teoretickými východisky této práce. Uvádím základní pojmy z oblasti teorie pravděpodobnosti, popisují metody vzorkování náhodných čísel a představuji teorii kopulí.

Kapitola 3 se zaměřuje na přehled algoritmů EDA. Jsou zde stručně popsány klasické varianty algoritmů EDA, dále algoritmy EDA užívající migraci diskrétních modelů a také existující varianty algoritmů EDA využívající kopule.

V kapitole 4 se zabývám vlastním návrhem paralelního algoritmu mCEDA využívajícího kopule a migraci pravděpodobnostních modelů.

Kapitola 5 popisuje experimentální výsledky. V první řadě popisuje vliv jednotlivých řídicích parametrů na kvalitu konvergence testovaných variant algoritmu mCEDA pro standardní sadu testovacích úloh a sadu CEC 2013. Finální algoritmus mCEDA je pak porovnáván s EDA algoritmy jiných autorů na standardní sadě úloh a s algoritmy využívající jiná evoluční paradigmatata na sadě CEC 2013.

Závěrečná kapitola 6 shrnuje nejdůležitější přínosy této práce.

Kapitola 2

Úvod do teorie kopulí

V první části této kapitoly shrnuji základní poznatky o teorii pravděpodobnosti a statistice, které budou používány v dalším textu. Jádro této kapitoly tvoří popis teorie kopulí, což je matematický nástroj umožňující efektivně reprezentovat vícerozměrná rozdělení pravděpodobnosti. Pro jednotlivé rodiny kopulí je také popsána metodika vzorkování kopulí, která bude využívána v navazujících kapitolách.

2.1 Rozdělení spojitě náhodné veličiny

Pro text disertační práce jsou důležité pojmy *distribuční funkce*, *hustota pravděpodobnosti* a *marginální distribuční funkce*, které budou nyní definovány.

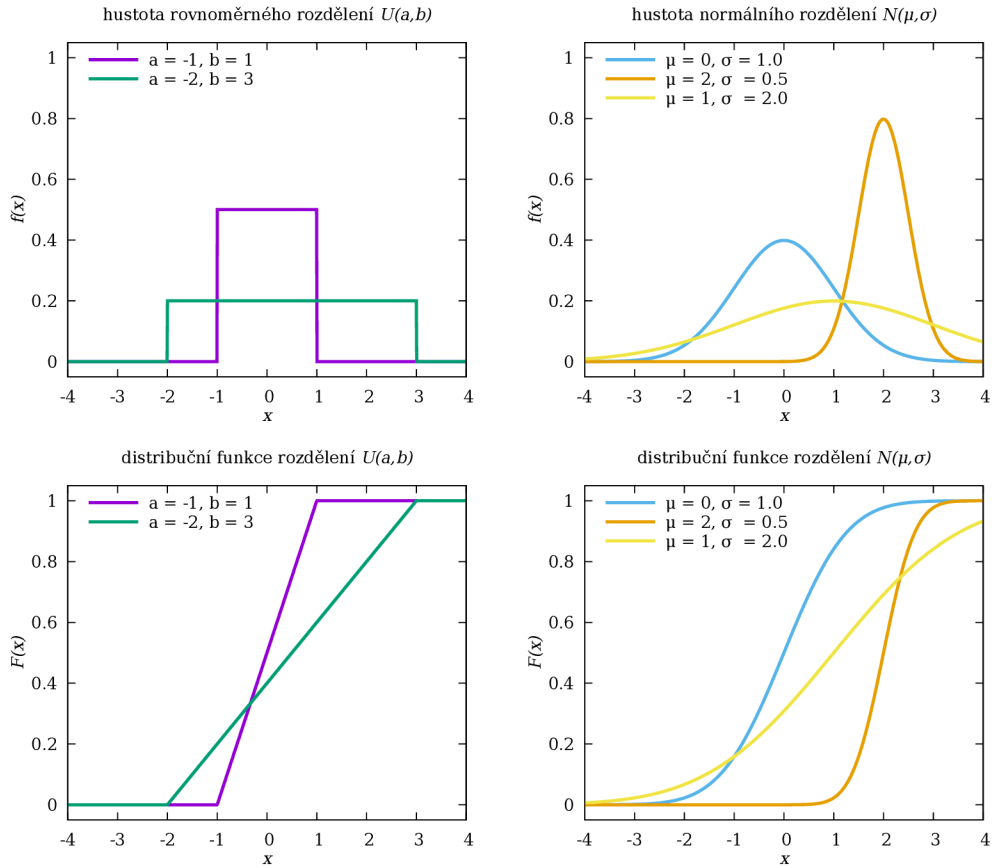
Definice 2.1. Necht \mathbb{X} je náhodná veličina. Funkci $F(x)$ definovanou vztahem $F(x) = p(\mathbb{X} < x)$ nazýváme *distribuční funkcí* náhodné veličiny \mathbb{X} . Pro všechna $x \in \mathbb{R}$ platí

1. $0 \leq F(x) \leq 1$,
2. $F(x)$ je neklesající funkce,
3. $F(x)$ je zleva spojitá,
4. $\lim_{x \rightarrow -\infty} F(x) = 0 \quad \lim_{x \rightarrow +\infty} F(x) = 1$,
5. $p(\mathbb{X} = x) = \lim_{h \rightarrow 0^+} F(x+h) - F(x)$,
6. $p(x_1 \leq \mathbb{X} < x_2) = F(x_2) - F(x_1)$.

Definice 2.2. Necht \mathbb{X} je spojitá náhodná veličina s distribuční funkcí $F(x)$. Reálnou funkci $f(x)$, pro kterou platí

1. $f(x) \geq 0 \quad \forall x \in \mathbb{R}$,
2. $\int_{-\infty}^{\infty} f(x) dx = 1$,
3. $F(x) = \int_{-\infty}^x f(t) dt$,

nazýváme *funkcí hustoty pravděpodobnosti* náhodné veličiny \mathbb{X} .



Obrázek 2.1: Příklad distribuční funkce a funkce hustoty pravděpodobnosti pro rovnoměrné a normální rozdělení pravděpodobnosti.

Příkladem velmi často užívaných rozdělení pravděpodobnosti jsou rovnoměrné rozložení a normální rozložení, viz obrázek 2.1. Hustota rovnoměrného rozdělení $U(a, b)$ nabývá v daném intervalu konstantní hodnoty, jinde je jeho hustota nulová

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{pro } x \in (a; b), \\ 0 & \text{pro } x \notin (a; b). \end{cases} \quad (2.1)$$

Normální rozdělení $\mathcal{N}(\mu, \sigma)$ je definováno vztahem

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (2.2)$$

Normální rozdělení se střední hodnotou $\mu = 0$ a směrodatnou odchylkou $\sigma = 1$ se nazývá normované a jeho distribuční funkce se označuje $\Phi(x)$.

Definice 2.3. Mějme D -rozměrnou náhodnou veličinu $(\mathbb{X}_1, \dots, \mathbb{X}_D)$ se sdruženou distribuční funkcí $F(x_1, \dots, x_D)$. Každá veličina \mathbb{X}_d má svojí distribuční funkci $F_d(x_d)$. Pak platí

1. $p(\mathbb{X}_d < x_d) = p(\mathbb{X}_d < x_d, \forall \mathbb{X}_j \in (-\infty; +\infty), j \neq d)$,
2. $F_d(x_d) = \lim_{\forall x_j \rightarrow +\infty, j \neq d} F(x_1, \dots, x_D)$.

Distribuční funkce $F_d(x_d)$ se nazývají *marginální distribuční funkce*.

2.1.1 Vzorkování náhodných čísel

Vzorkováním náhodných čísel rozumíme takový proces, kdy jsou generována (pseudo)náhodná čísla takovým způsobem, aby splňovala zadané rozdělení pravděpodobnosti. Dva základní přístupy jsou inverzní transformace a vylučovací metoda.

Metoda inverzní transformace

Metoda inverzní transformace je výpočetně jednoduchá metoda, ale pro její použití je nutné znát inverzní funkci $F^{-1}(u)$ k distribuční funkci požadovaného rozdělení.

Při použití metody je vygenerováno náhodné číslo u z rovnoměrného rozdělení $u \sim U(0, 1)$. Z náhodné hodnoty u je vypočítána hodnota požadovaného rozdělení

$$x = F^{-1}(u). \quad (2.3)$$

Vylučovací metoda

Vylučovací metoda, též známá jako acceptance-rejection sampling, je metoda založená na přístupu Monte Carlo. Jejím základem je iterativní generování náhodných čísel, dokud není splněno žádané rozdělení.

Předpokladem pro její použití je omezení rozdělení na určitý interval $\langle MIN; MAX \rangle$, znalost funkce hustoty $f(x)$ tohoto rozdělení a znalost maximální hodnoty m , které funkce $f(x)$ na intervalu $\langle MIN; MAX \rangle$ nabývá. Za těchto podmínek je metoda popsána následujícím algoritmem, viz alg. 2.1.

Algoritmus 2.1 Pseudokód algoritmu pro vylučovací metodu.

1. REPEAT
 2. \dashrightarrow generuj náhodné $x \sim U(MIN, MAX)$
 3. \dashrightarrow generuj náhodné $u \sim U(0, 1)$
 4. UNTIL $u \leq \frac{f(x)}{m}$
 5. výsledek je x
-

Významnou výhodou oproti předešlé metodě je skutečnost, že není potřeba inverzní funkce. Zásadní nevýhodou této metody je její výpočetní náročnost způsobená opakovaným cyklem.

Specifické přístupy

Některá rozložení lze generovat přístupem specifickým pouze pro toto konkrétní rozdělení. Známým příkladem je získávání normálně rozdělených čísel pomocí součtu alespoň dvanácti rovnoměrně rozdělených náhodných čísel. Jinou ukázkou specifického přístupu bude vzorkování kopule, ukázané v kapitole 2.4.1

2.2 Kopule

V této kapitole bude uvedena matematická definice kopule, vysvětlen význam Sklarova teorému a popsány základní typy kopulí.

2.2.1 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 [36, 55]:

Definice 2.4. 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 [68, 49]:

Definice 2.5. 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. $\forall (u_1, \dots, u_D), (v_1, \dots, v_D) \in \langle 0; 1 \rangle^D, u_d \leq v_d :$

$$\sum_{(w_1, \dots, w_D) \in \prod_{d=1}^D \{u_d, v_d\}} (-1)^{|\{d | w_d = u_d\}|} C(w_1, \dots, w_D) \geq 0.$$

Mimo to existují i definice kopule pomocí funkce přežití či Laplaceovy transformace (viz [49, 55]). Ať už kopuli definujeme kterýmkoliv uvedeným způsobem, vždy platí to, že kopule je distribuční funkce. Pro kopuli tedy musí platit všechny vlastnosti (například metody pro vzorkování nebo vztah mezi hustotou pravděpodobnosti a distribuční funkcí), jako pro každou distribuční funkci.

Na obrázku 2.2 je ukázka základních kopulí, které modelují případy, kdy proměnné jsou: zcela negativně závislé – nezávislé – zcela pozitivně závislé.

2.2.2 Sklarův teorém

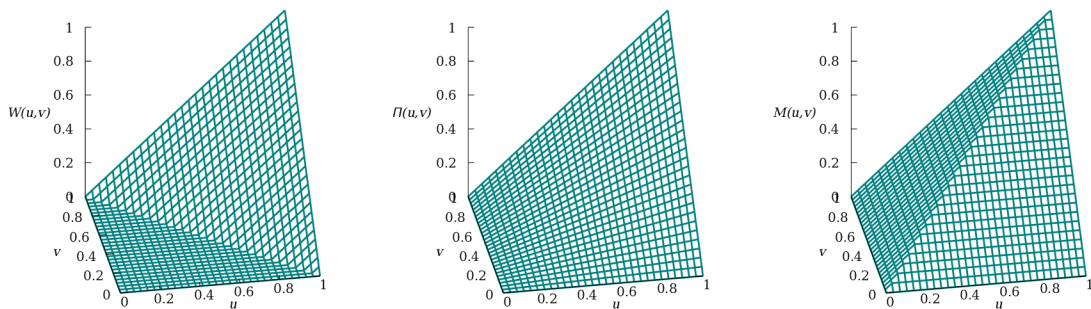
Sklarův teorém [67] má v teorii kopulí velký význam. Popisuje vztah mezi sdruženou distribuční funkcí a mezi kopulí spolu s marginálními rozděleními. Lze říci, že při reprezentaci sdružené distribuční funkce lze oddělit efekt marginálních rozdělení od efektu závislosti mezi proměnnými. Právě důsledky 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*

$$F(x_1, x_2, \dots, x_D) = C(F_1(x_1), F_2(x_2), \dots, F_D(x_D)). \quad (2.4)$$

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.



Obrázek 2.2: Příklad kopulí, $W(u, v) = \max(u+v-1, 0)$, $\Pi(u, v) = uv$, $M(u, v) = \min(u, v)$.

Hustota pravděpodobnosti vícerozměrného rozdělení je za využití kopule popsána vztahem

$$f(x_1, x_2, \dots, x_D) = c(F_1(x_1), \dots, F_D(x_D)) \prod_{d=1}^D f_d(x_d), \quad (2.5)$$

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 [36].

Nicméně Sklarův teorém nijak nepopisuje, jak získat kopuli pro konkrétní rozdělení. Často používaný přístup je proto ten, že pevně zvolíme typ kopule (takový, o němž se domníváme, že vyhoví našim potřebám) a z daných dat pouze odvozujeme parametry zvolené kopule.

2.2.3 Vizualizace kopulí

Jedním z nejnázornějších způsobů, jak ukázat vlastnosti rozdělení pravděpodobnosti, je pomocí grafů; toto platí i pro kopule. Obvykle se zobrazují dvojrozměrné kopule [49, 55].

Prostorový graf

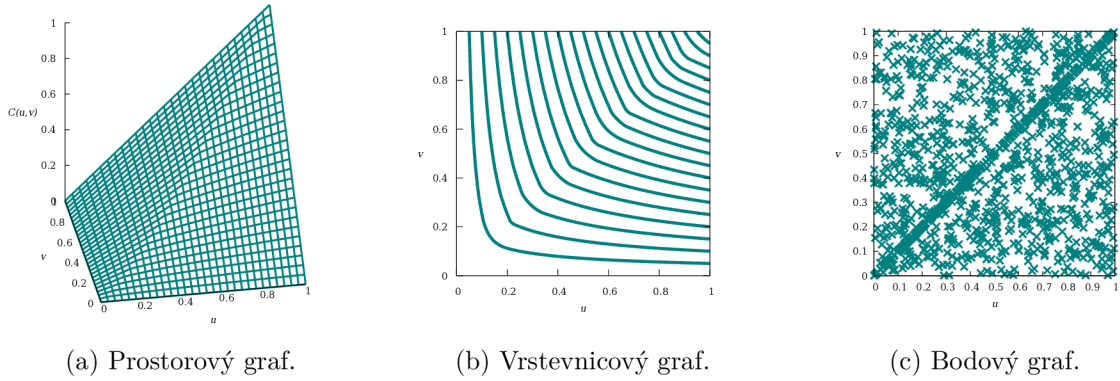
Prostorový graf dvojrozměrné kopule $C(u, v)$ má na vodorovných osách vyneseny hodnoty u, v a na svislé ose je vynášena funkční hodnota $C(u, v)$. Všechny osy nabývají rozsahu $\langle 0; 1 \rangle$. Pro příklad prostorového grafu viz obr. 2.3a.

Vrstevnicový graf

Vrstevnicový graf s n vrstevnicemi zachycuje body o stejné funkční hodnotě, tj. množiny $L_{k,n}$, kde

$$L_{k,n} = \left\{ (u, v) \in \langle 0; 1 \rangle^2 \mid C(u, v) = \frac{k}{n} \right\}, \quad k = 0, 1, \dots, n. \quad (2.6)$$

Pro příklad vrstevnicového grafu viz obr. 2.3b.



Obrázek 2.3: Příklady grafů pro vizualizaci kopulí (kopule typu Cuadras-Augé, $\theta = 0.5$).

Bodový graf

Bodový graf je velmi často používán pro názorné zobrazení závislosti mezi proměnnými náhodného vektoru. V případě kopule body získáme vzorkováním realizace náhodného vektoru asociovaného s kopulí C . Bodový graf vnímáme i jako reprezentaci hustoty ([49]). Pro příklad bodového grafu viz obr. 2.3c.

2.2.4 Typy kopulí

Typů kopulí existuje velké množství, často používané kopule se dělí do dvou rodin (archimédovské a eliptické kopule), ale existují i kopule, která nepatří do žádné z těchto rodin (např. Marshall-Olkinovy kopule).

Archimédovské kopule

Archimédovské kopule jsou popsány pomocí svého generátoru a jeho inverze, podrobnosti viz [49].

Definice 2.6. Generátor je funkce $\varphi: \langle 0; \infty \rangle \rightarrow \langle 0; 1 \rangle$ s těmito vlastnostmi:

1. $\varphi(0) = 1$ a $\lim_{x \rightarrow \infty} \varphi(x) = 0$,
2. φ je spojitá,
3. φ je klesající na $\langle 0; \infty \rangle$ a ostře klesající na $\langle 0; \inf\{x > 0 : \varphi(x) = 0\}$, přičemž buď $\inf(\emptyset) := 0$.¹

Definice 2.7. Archimédovské kopule jsou definovány touto rovnicí

$$C_\varphi = \varphi(\varphi^{-1}(u_1) + \varphi^{-1}(u_2) + \dots + \varphi^{-1}(u_D)), \quad (2.7)$$

kde $\varphi(x)$ je generátorová funkce.

Archimédovských kopulí existuje velké množství, příklad neznámějších viz tabulka 2.1 a příklad na obrázku 2.4. Uvedené kopule jsou dvojrozměrné, v takovéto situaci se často místo obecného označení $C(u_1, u_2)$ používá zápis $C(u, v)$.

¹Na tom intervalu, kde nabývá kladné hodnoty, je generátor $\varphi(x)$ ostře klesající funkce.

Tabulka 2.1: Příklad archimédovských kopulí a jejich generátorů.

kopule	$C(u, v) = \varphi(\varphi^{-1}(u) + \varphi^{-1}(v))$	
součinnová	$C(u, v) = uv$	
Claytonova	$C(u, v) = (\max\{u^{-\theta} + v^{-\theta} - 1; 0\})^{-1/\theta}$	
Gumbelova	$C(u, v) = \exp\left(-\left((-\log(u))^\theta + (-\log(v))^\theta\right)^{1/\theta}\right)$	
Frankova	$C(u, v) = -\frac{1}{\theta} \log\left(1 + \frac{(\exp(-\theta u) - 1)(\exp(-\theta v) - 1)}{\exp(-\theta) - 1}\right)$	
generátor	$\varphi(t)$	$\varphi^{-1}(t)$
součinnová	$\varphi(t) = e^{-t}$	$\varphi^{-1}(t) = -\log(t)$
Claytonova	$\varphi(t) = (1 + \theta t)^{-1/\theta}$	$\varphi^{-1}(t) = \frac{1}{\theta}(t^{-\theta} - 1)$
Gumbelova	$\varphi(t) = \exp(-t^{1/\theta})$	$\varphi^{-1}(t) = (-\log(t))^\theta$
Frankova	$\varphi(t) = -\frac{1}{\theta} \log(1 + \exp(-t)(\exp(-\theta) - 1))$	$\varphi^{-1}(t) = -\log\left(\frac{\exp(-\theta t) - 1}{\exp(-\theta) - 1}\right)$

V některé literatuře (př. [55]) je zaměněn generátor a inverze generátoru. Definiční rovnice má pak tvar $C_\varphi = \varphi^{-1}(\varphi(u_1) + \varphi(u_2) + \dots + \varphi(u_D))$. Je ale také zaměněna definice generátoru a jeho inverze a také je zaměněna dvojice konkrétních rovnic generátor – inverze generátoru pro konkrétní kopuli. Daná kopule je tedy popsána vždy stejnou rovnicí, bez ohledu na variantu použité definice.

Archimédovské kopule mají často jen jeden parametr θ (bez ohledu na počet dimenzí), jejich schopnosti popisovat vícerozměrné závislosti jsou proto v takových případech omezené.

Jednotlivé dimenze archimédovských kopulí jsou tzv. zaměnitelné, př. $C_\varphi(u, v) = C_\varphi(v, u)$. Arch. kopule je možné asociativně skládat. Tyto vlastnosti mají díky tomu, že jsou definovány pomocí součtu (a sčítání je komutativní i asociativní). V důsledku těchto vlastností je možné je použít jako stavební bloky ve vícerozměrných modelech, což vede na tzv. C-vine a D-vine kopule či hierarchické modely.

Eliptické kopule

Eliptické kopule jsou definovány pomocí eliptických rozdělení pravděpodobnosti.

Definice 2.8. Eliptické kopule jsou určeny rovnicí ve tvaru

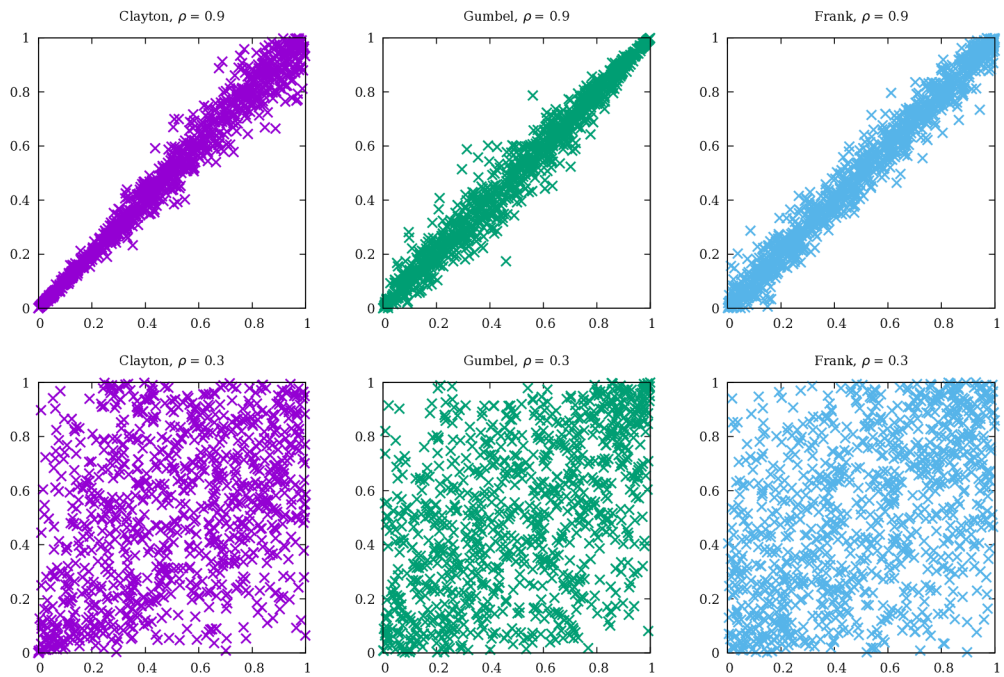
$$C_R(u_1, \dots, u_d) = F_R^\mathcal{E}(F_\mathcal{E}^{-1}(u_1), \dots, F_\mathcal{E}^{-1}(u_d)), \quad (2.8)$$

kde $F_R^\mathcal{E}(x_1, \dots, x_D)$ je eliptické rozdělení s korelační maticí R a $F_\mathcal{E}^{-1}(u)$ je jeho kvantilová funkce.

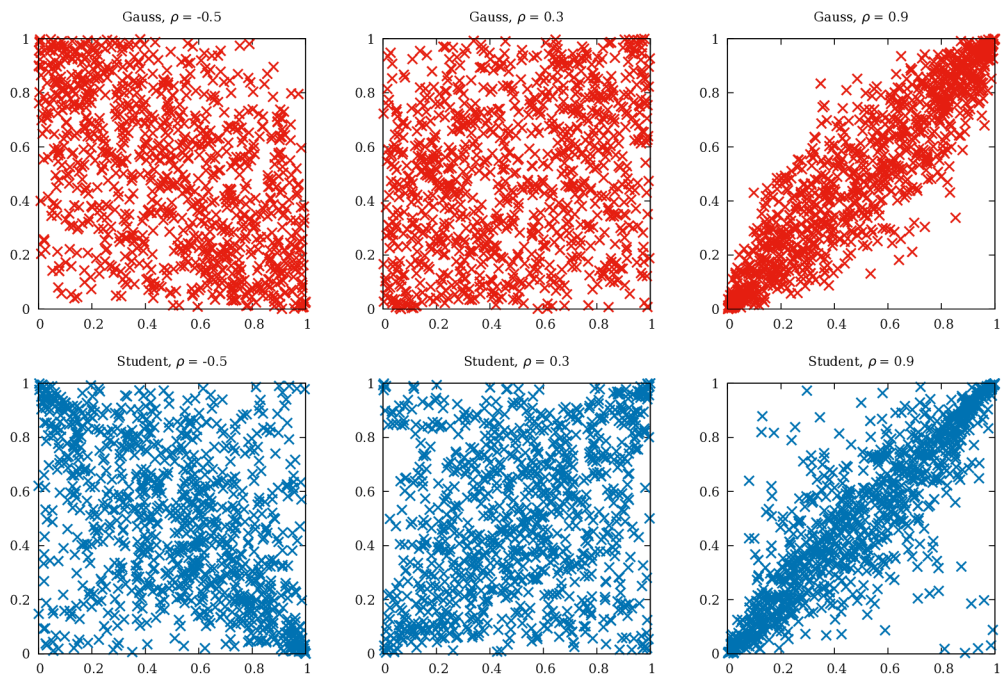
Příkladem eliptické kopule je Gaussova kopule odvozená od normálního rozdělení nebo t -kopule odvozená od Studentova t -rozdělení:

$$C_R^{\text{Gauss}} = \Phi_R(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)), \quad (2.9)$$

$$C_{R,\nu}^t = t_{\nu,R}(t_\nu^{-1}(u_1), \dots, t_\nu^{-1}(u_d)). \quad (2.10)$$



Obrázek 2.4: Vzorčky získané z dvojrozměrných archimédovských kopulí s korelací ρ .



Obrázek 2.5: Vzorčky získané z dvojrozměrných eliptických kopulí s korelací ρ .

Eliptické kopule jsou vždy symetrické.

Na rozdíl od standardních archimédovských kopulí nejsou možnosti použití eliptických kopulí omezeny malým počtem parametrů, eliptické kopule jednoduše modelují vícerozměrná rozdělení. Korelační matice R je tvořena dílčími parametry mezi dvojicemi proměnných ve tvaru

$$R = \begin{pmatrix} 1 & \rho_{1,2} & \cdots & \rho_{1,D} \\ \rho_{2,1} & 1 & \cdots & \rho_{2,D} \\ \vdots & \cdots & \ddots & \vdots \\ \rho_{D,1} & \cdots & \rho_{D,D-1} & 1 \end{pmatrix} \quad (2.11)$$

(korelační matice je symetrická, prvky na hlavní diagonále mají hodnotu jedna).

Eliptické kopule dokáží postihnout všechny hodnoty korelace, kladné i záporné, příklad eliptických kopulí s různými korelacemi je uveden na obrázku 2.5.

2.3 Vzorkování sdruženého rozdělení pomocí kopulí

Vzorkování pravděpodobnostního modelu je stěžejním krokem algoritmů EDA, během něhož dochází k vytváření nové populace pro další generaci.

Postup pro vzorkování sdruženého rozdělení pravděpodobnosti $F(x_1, \dots, x_D) = C(F_1(x_1), \dots, F_D(x_D))$ popsaného pomocí kopule a marginálních funkcí má dva samostatné kroky, viz algoritmus 2.2. Nejprve samostatně navzorkujeme kopuli (vhodnou metodou podle typu kopule, viz následující kapitola). Vzorkem kopule je náhodný vektor vzájemně závislých hodnot (u_1, \dots, u_D) , kde každý prvek vektoru u_d je hodnota z intervalu $\langle 0; 1 \rangle$. Ve druhém kroku pak inverzní transformací navzorkujeme jednotlivá marginální rozdělení.

Algoritmus 2.2 Vzorkování sdruženého rozdělení popsaného pomocí kopule a marginálních rozdělení.

1. získej vzorek kopule $(u_1, \dots, u_D) \sim C$, $u_d \in \langle 0; 1 \rangle$
 2. inverzí marginálních rozdělení vypočítej hodnoty $x_d := F_d^{-1}(u_d)$, $x_d \in \mathbb{R}$
 3. výsledek je D-tice $(x_1, \dots, x_D) \sim F$
-

Hlavním přínosem tohoto postupu je to, že vzorkování sdružené distribuční funkce je rozděleno na dva jednodušší kroky. A dále, při volbě marginálního rozdělení používáme pouze jednorozměrná rozdělení, to je velké zjednodušení celého procesu vytváření modelu a vzorkování. Také můžeme pro každou dimenzi použít marginální rozdělení jiného typu, toto by při použití klasických vícerozměrných rozdělení bylo jen těžko realizovatelné.

Určitým problémem této metody je to, že musíme být schopni vypočítat hodnotu inverzní distribuční funkce požadovaného marginálního rozdělení.

2.4 Vzorkování kopulí

V této kapitole se budu zabývat algoritmy na získání náhodného vzorku z rozdělení popsání kopulí. Nejprve představím obecnou metodologii pro dvě dimenze a ukážu její aplikaci na příkladu Gaussovy kopule. Poté odvodím algoritmus pro vzorkování vícerozměrné Gaussovy kopule pomocí vylučovací metody. Ve třetí části této kapitoly popíšu algoritmy pro efektivní vzorkování kopulí.

2.4.1 Využití podmíněné pravděpodobnosti

V knize [49] je popsána obecná metodologie pro vzorkování kopulí s malým počtem dimenzí založená na podmíněné pravděpodobnosti. Podle této metodologie je jedna proměnná navzorkována nezávisle, $u \sim U(0; 1)$. Dále je nutné spočítat podmíněnou pravděpodobnost $F_{V|U=u}(v) = \frac{\partial}{\partial u} C(u, v)$ a stanovit inverzní funkci tohoto rozdělení $F_{V|U=u}^{-1}(\omega)$. Následně je vygenerována nezávislá hodnota $\omega \sim U(0; 1)$ a proměnná v je do počítána pomocí $v = F_{V|U=u}^{-1}(\omega)$. Dvojice (u, v) pak tvoří požadovaný vzorek z kopule C .

Na základě této metodologie jsem odvodil postup a výsledné rovnice pro vzorkování dvouřozměrné Gaussovy kopule.

Příklad vzorkování Gaussovy kopule

Dvojřozměrná Gaussovská kopule je definována jako $C(u, v; \rho) = \Phi_\rho(\Phi^{-1}(u), \Phi^{-1}(v))$, kde $\Phi_\rho(x, y) = P(\mathbb{X} < x, \mathbb{Y} < y) = \int_{-\infty}^x \int_{-\infty}^y \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{s^2+t^2-2\rho st}{2(1-\rho^2)}} dt ds$ je distribuční funkce dvojřozměrného normovaného normálního rozložení s korelačním koeficientem ρ a $\Phi(x) = \int_{-\infty}^x \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$ je distribuční funkce normovaného normálního rozložení, $\Phi^{-1}(x)$ je její kvantilová funkce.

$$C(u, v; \rho) = \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{s^2+t^2-2\rho st}{2(1-\rho^2)}} dt ds. \quad (2.12)$$

Derivací podle základní věty integrálního počtu $\frac{d}{dx} \left(\int_c^{g(x)} f(t) dt \right) = f(g(x))g'(x)$ získáme:

$$\omega = \frac{\partial}{\partial u} C(u, v; \rho) = \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{(\Phi^{-1}(u))^2+t^2-2\rho(\Phi^{-1}(u))t}{2(1-\rho^2)}} (\Phi^{-1})'(u) dt. \quad (2.13)$$

Vhodnými úpravami (úprava na čtverec v čitateli zlomku, následným zjednodušením a přesunutím konstantní části před integrál) postupně získáme:

$$\omega = (\Phi^{-1})'(u) \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{(t-\rho(\Phi^{-1}(u)))^2+(\Phi^{-1}(u))^2-\rho^2(\Phi^{-1}(u))^2}{2(1-\rho^2)}} dt, \quad (2.14)$$

$$\omega = (\Phi^{-1})'(u) \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\left(\frac{(t-\rho(\Phi^{-1}(u)))^2}{2(1-\rho^2)} + \frac{(\Phi^{-1}(u))^2-\rho^2(\Phi^{-1}(u))^2}{2(1-\rho^2)} \right)} dt, \quad (2.15)$$

$$\omega = (\Phi^{-1})'(u) \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{(t-\rho(\Phi^{-1}(u)))^2}{2(1-\rho^2)}} e^{-\frac{(\Phi^{-1}(u))^2(1-\rho^2)}{2(1-\rho^2)}} dt, \quad (2.16)$$

$$\omega = (\Phi^{-1})'(u) e^{-\frac{(\Phi^{-1}(u))^2}{2}} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{(t-\rho(\Phi^{-1}(u)))^2}{2(1-\rho^2)}} dt. \quad (2.17)$$

Nyní vypočítáme derivaci $(\Phi^{-1})'(u)$. Pro derivaci inverzní funkce existuje vztah $(f^{-1})'(y) = \frac{1}{f'(f^{-1}(y))}$, tedy v našem případě

$$(\Phi^{-1})'(u) = \frac{1}{\Phi'(\Phi^{-1}(u))}. \quad (2.18)$$

Derivaci Φ' určíme pomocí základní věty integrálního počtu

$$\Phi'(x) = \frac{d}{dx} \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (2.19)$$

a po dosazení do vztahu pro derivaci inverze dostaneme

$$(\Phi^{-1})'(u) = \frac{1}{\frac{1}{\sqrt{2\pi}} e^{-\frac{(\Phi^{-1}(u))^2}{2}}} = \frac{\sqrt{2\pi}}{e^{-\frac{1}{2}(\Phi^{-1}(u))^2}}. \quad (2.20)$$

Po dosazení do původní rovnice pro ω (2.17) získáme

$$\omega = \frac{\sqrt{2\pi}}{e^{-\frac{1}{2}(\Phi^{-1}(u))^2}} e^{-\frac{(\Phi^{-1}(u))^2}{2}} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{(t-\rho(\Phi^{-1}(u)))^2}{2(1-\rho^2)}} dt, \quad (2.21)$$

$$\omega = \sqrt{2\pi} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{(t-\rho(\Phi^{-1}(u)))^2}{2(1-\rho^2)}} dt, \quad (2.22)$$

$$\omega = \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^2}} e^{-\frac{(t-\rho(\Phi^{-1}(u)))^2}{2(1-\rho^2)}} dt. \quad (2.23)$$

Nyní integrál odpovídá distribuční funkci normálního rozdělení $F(x) = \int_{-\infty}^x \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt = \Phi\left(\frac{x-\mu}{\sigma}\right)$ s parametry $\sigma = \sqrt{1-\rho^2}$ a $\mu = \rho(\Phi^{-1}(u))$, přičemž za x je dosazeno $\Phi^{-1}(v)$; můžeme tedy rovnici přepsat do tvaru

$$\omega = \Phi\left(\frac{\Phi^{-1}(v) - \rho\Phi^{-1}(u)}{\sqrt{1-\rho^2}}\right). \quad (2.24)$$

Inverzí a následnou ekvivalentní úpravou rovnice získáme v :

$$\Phi^{-1}(\omega) = \frac{\Phi^{-1}(v) - \rho\Phi^{-1}(u)}{\sqrt{1-\rho^2}}, \quad (2.25)$$

$$\sqrt{1-\rho^2} \Phi^{-1}(\omega) + \rho\Phi^{-1}(u) = \Phi^{-1}(v), \quad (2.26)$$

$$v = \Phi\left(\sqrt{1-\rho^2} \Phi^{-1}(\omega) + \rho\Phi^{-1}(u)\right). \quad (2.27)$$

Jedna proměnná je navzorkována nezávisle, druhá je dopočítána tak, aby byla uplatněna existující korelace ρ mezi proměnnými, viz alg. 2.3.

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

Algoritmus 2.3 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)
-

2.4.2 Vzorkování Gaussovy kopule pomocí vylučovací metody

Jako další jsem odvodil postup pro vzorkování vícerozměrné Gaussovy kopule pomocí vylučovací metody.

Cílem vylučovací metody je získat náhodné číslo podle daného rozdělení pravděpodobnosti; toto rozdělení je ohraničeno na určitém intervalu, pro dané rozdělení jsme schopni stanovit funkci hustoty pravděpodobnosti a na onom intervalu známe maximální možnou hodnotu funkce hustoty. Tento postup může být jednoduše rozšířen do více dimenzí.

Tento algoritmus potřebuje popis kopule pomocí hustoty pravděpodobnosti, zatímco kopule jsou běžně definovány jako distribuční funkce. Hustotu z distribuce získáme pomocí derivace, např. pro Gaussovu kopuli:

$$c_R(u_1, \dots, u_D) = \frac{\partial^{D-1}}{\partial u_2 \dots \partial u_D} \frac{\partial}{\partial u_1} C_R(u_1, u_2, \dots, u_D), \quad (2.28)$$

$$c_R(u_1, \dots, u_D) = \frac{\partial^{D-1}}{\partial u_2 \dots \partial u_D} \frac{\partial}{\partial u_1} \int_{-\infty}^{\Phi^{-1}(u_1)} \int_{-\infty}^{\Phi^{-1}(u_2)} \dots \int_{-\infty}^{\Phi^{-1}(u_D)} \frac{1}{\sqrt{(2\pi)^D |R|}} e^{\left(-\frac{1}{2}(t_1 \ t_2 \ \dots \ t_D) R^{-1} \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_D \end{pmatrix}\right)} dt_D \dots dt_2 dt_1. \quad (2.29)$$

Opakovaným užitím pravidla $\frac{d}{dx} \left(\int_c^{g(x)} f(t) dt \right) = f(g(x))g'(x)$ postupně získáme

$$c_R(u_1, \dots, u_D) = \frac{\partial^{D-1}}{\partial u_2 \dots \partial u_D} \int_{-\infty}^{\Phi^{-1}(u_2)} \dots \int_{-\infty}^{\Phi^{-1}(u_D)} \frac{1}{\sqrt{(2\pi)^D |R|}} e^{\left(-\frac{1}{2}(\Phi^{-1}(u_1) \ t_2 \ \dots \ t_D) R^{-1} \begin{pmatrix} \Phi^{-1}(u_1) \\ t_2 \\ \vdots \\ t_D \end{pmatrix}\right)} dt_D \dots dt_2 \cdot \left((\Phi^{-1})'(u_1) \right), \quad (2.30)$$

$$c_R(u_1, \dots, u_D) = \frac{1}{\sqrt{(2\pi)^D |R|}} e^{\left(-\frac{1}{2}(\Phi^{-1}(u_1) \ \dots \ \Phi^{-1}(u_D)) R^{-1} \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_D) \end{pmatrix}\right)} \cdot \prod_{d=1}^D \left((\Phi^{-1})'(u_d) \right). \quad (2.31)$$

Po dosazení za výraz $(\Phi^{-1})'(u_d)$ podle rovnice (2.20) a postupným vykrácením a zjednodušením

$$c_R(u_1, \dots, u_D) = \frac{1}{\sqrt{(2\pi)^D |R|}} e^{\left(-\frac{1}{2} (\Phi^{-1}(u_1) \dots \Phi^{-1}(u_D)) R^{-1} \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_D) \end{pmatrix} \right)} \cdot \prod_{d=1}^D \frac{\sqrt{2\pi}}{e^{-\frac{1}{2}(\Phi^{-1}(u_d))^2}}, \quad (2.32)$$

$$c_R(u_1, \dots, u_D) = \frac{1}{\sqrt{|R|}} e^{\left(-\frac{1}{2} \left((\Phi^{-1}(u_1) \dots \Phi^{-1}(u_D)) R^{-1} \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_D) \end{pmatrix} - \Phi^{-1}(u_1)^2 - \dots - \Phi^{-1}(u_D)^2 \right) \right)}, \quad (2.33)$$

$$c_R(u_1, \dots, u_D) = \frac{1}{\sqrt{|R|}} e^{\left(-\frac{1}{2} (\Phi^{-1}(u_1) \dots \Phi^{-1}(u_D)) (R^{-1} - I) \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_D) \end{pmatrix} \right)}. \quad (2.34)$$

Zavedme označení matice $A = (R^{-1} - I)$ a řádkového vektoru $\mathbf{v} = (\Phi^{-1}(u_1) \dots \Phi^{-1}(u_D))$. Při opakovaném vzorkování dané kopule je matice A konstantní; mění se pouze hodnoty vektoru \mathbf{v} počítané podle aktuálně vygenerovaných hodnot u_1, \dots, u_D . Pomocí takto uvedené notace vyjádříme hustotu Gaussovy kopule jako

$$c_R(u_1, \dots, u_D) = \frac{1}{\sqrt{|R|}} e^{(-\frac{1}{2} \mathbf{v} A \mathbf{v}^T)}. \quad (2.35)$$

Maticové násobení $\mathbf{v} A \mathbf{v}^T$ může být efektivně spočítáno jako: $\mathbf{v} A \mathbf{v}^T = \sum_{i=1}^D \sum_{j=1}^D A_{ij} v_i v_j$. Hustotu Gaussovy kopule vyjádříme – pomocí tohoto výpočtu –

$$c_R(u_1, \dots, u_D) = \frac{1}{\sqrt{|R|}} e^{(-\frac{1}{2} \sum_{i=1}^D \sum_{j=1}^D A_{ij} v_i v_j)}. \quad (2.36)$$

Maximální možná hodnota hustoty nastává tehdy, pokud je výraz v exponentu roven nula, tedy $m = \frac{1}{\sqrt{|R|}} e^0 = \frac{1}{\sqrt{|R|}}$. Díky tomu může být vylučovací podmínka ze vzorkovacího algoritmu vyjádřena jako

$$u \leq e^{(-\frac{1}{2} \sum_{i=1}^D \sum_{j=1}^D A_{ij} v_i v_j)}. \quad (2.37)$$

Celý postup pro vzorkování vícerozměrné Gaussovy kopule pomocí vylučovací metody je zde uveden jako alg. 2.4.

Tento algoritmus jsem publikoval v článku [F].

Výhodou tohoto algoritmu je jeho poměrně snadná modifikace pro jiná rozdělení. Zásadní nevýhodou je výpočetní náročnost, protože algoritmus je iterativní a může trvat velmi mnoho iterací, než vzorkování uspěje.

Algoritmus 2.4 Vzorkování vícerozměrné Gaussovy kopule pomocí vylučovací metody.

1. sestav pomocnou matici $A = R^{-1} - I$ (R je daná korelační matice, I je jednotková matice)
 2. REPEAT
 3. \dashrightarrow generuj náhodná $u_d \sim U(0, 1)$, $d = 1, \dots, D$
 4. \dashrightarrow vypočítej hodnoty $v_d := \Phi^{-1}(u_d)$
 5. \dashrightarrow 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)
-

2.4.3 Algoritmy pro efektivní vzorkování kopulí

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

Algoritmy uvedené v této kapitole jsem publikoval v článkách [E] a [D].

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 [49, 1, 50, 27], viz alg. 2.5.

Konkrétně pro zde uvažované typy archimédovských kopulí je hodnota J dána [1, 50]:

- pro Claytonovu kopuli Gamma rozdělením $J \sim \text{Gamma}(\frac{1}{\theta}, \theta)$,
- pro Gumbelovu kopuli Levyho skew alpha-Stable rozdělením $J \sim \text{Stable}(\frac{1}{\theta}, 1, (\cos \frac{\pi}{2\theta})^\theta, 0)$,
- pro Frankovu kopuli logaritmickým rozdělením $J \sim \text{Logarithmic}(1 - e^{-\theta})$.

Vzorkování **Gaussovy a Studentovy kopule** [49] (viz alg. 2.6, 2.7) 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 2.5 Vzorkování archimédovských kopulí.

1. vygeneruj náhodnou hodnotu $J \sim \mathcal{L}^{-1}[\varphi(t)]$
 2. vygeneruj hodnoty (z rovnoměrného rozdělení) $z_d \sim U(0, 1)$ (pro $d = 1, \dots, D$)
 3. spočítej $u_d := \varphi\left(\frac{-\log(z_d)}{J}\right)$ (pro $d = 1, \dots, D$)
 4. výsledek je D -tice (u_1, \dots, u_D)
-

Algoritmus 2.6 Vzorkování Gaussovy kopule.

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

Algoritmus 2.7 Vzorkování Studentovy kopule.

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

Kapitola 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. \dashrightarrow vyhodnoť stávající populaci $P(g)$
 4. \dashrightarrow vyber kandidátní řešení $S(P(g))$
 5. \dashrightarrow vytvoř pravděpodobnostní model M z $S(P(g))$
 6. \dashrightarrow navzorkuj nové jedince $O(g)$ podle pravděpodobnostního modelu M
 7. \dashrightarrow vytvoř novou populaci $P(g+1)$, $P(g+1) \subset O(g) \cup P(g)$
 8. \dashrightarrow zvyš počítadlo generací g
 9. END WHILE
-

3.1 Přehled sekvenčních algoritmů typu EDA

Jednotlivé varianty EDA se liší zejména tím, jak složitý pravděpodobnostní model je použit. Mnoho z nich používá model ve formě určitého typu grafu, jsou proto označovány jako metody s grafovým modelem. Na obrázku 3.1 je uvedeno schematické znázornění nejvýznamnějších typů grafových modelů.

Podrobný popis algoritmů EDA je dostupný zejména v [41] nebo [34].

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.

3.1.1 Modely s nezávislými proměnnými

Algoritmy, ve kterých nejsou uvažovány žádné závislosti mezi proměnnými, jsou ze všech algoritmů typu EDA ty nejjednodušší. Jejich pravděpodobnostní model je popsán rovnicí

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

Algoritmus UMDA

Základním algoritmem bez závislostí mezi proměnnými je algoritmus UMDA (Univariate Marginal Distribution Algorithm) [53][58]. Jeho proces učení modelu je dán vztahem

$$p_{g+1}(X_d) = \frac{1}{N'} \sum_{\mathbf{x}^S \in S(P(g))} (x_d)^S \quad (3.3)$$

(model je učen pouze z aktuální vybrané populace $S(P(g))$).

Algoritmus PBIL

Algoritmus PBIL (Population-Based Incremental Learning) [3] je velmi podobný algoritmu UMDA, také předpokládá, že jednotlivé proměnné řešeného problému jsou navzájem nezávislé.

Rozdíl spočívá v tom, že zatímco UMDA vytváří model v každé generaci zcela znovu, PBIL svůj pravděpodobnostní model na základě vybraných jedinců inkrementálně mění

$$p_{g+1}(X_d) = (1 - \gamma)p_g(X_d) + \gamma \frac{1}{N'} \sum_{\mathbf{x}^S \in S(P(g))} (x_d)^S, \quad (3.4)$$

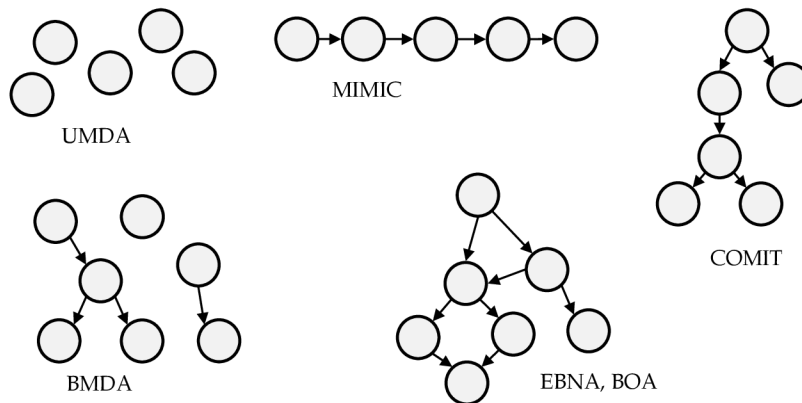
kde γ označuje učicí koeficient pro vytváření nového modelu.

Algoritmus cGA

Velmi minimalistickým algoritmem je cGA (Compact Genetic Algorithm) [33]. V každé generaci jsou vygenerováni jen dva jedinci a pravděpodobnostní model je zpřesněn podle rozdílu mezi nimi

$$p_{g+1}(X_d) = p_g(X_d) + \gamma(x_d^{best} - x_d^{worst}), \quad (3.5)$$

kde x_d^{best} , resp. x_d^{worst} je hodnota genu x_d z nejlepšího, resp. nejhoršího jedince.



Obrázek 3.1: Schematické znázornění grafových modelů užívaných v EDA.

3.1.2 Párové závislosti

V těchto algoritmech se již předpokládá existence závislostí mezi dvojicemi proměnných.

Algoritmus MIMIC

Algoritmus MIMIC (Mutual-Information-Maximizing Input Clustering) [13] zavádí řetězový model, proměnné jsou seřazeny podle množství vzájemné informace a je určena míra vzájemné závislosti sousedních proměnných. Vlastností tohoto grafového modelu je to, že k jeho zakódování slouží pouhá permutace $\pi = \pi_1 \pi_2 \dots \pi_D$ pořadí proměnných:

$$p_\pi(X_1, \dots, X_d) = p(X_{\pi_1} | X_{\pi_2}) p(X_{\pi_2} | X_{\pi_3}) \dots p(X_{\pi_{D-1}} | X_{\pi_D}) p(X_{\pi_D}). \quad (3.6)$$

Algoritmus COMIT

V algoritmu COMIT (Combining Optimizers with Mutual Information Trees) [4] jsou proměnné podle vzájemné závislosti uspořádány do stromu.

Algoritmus BMDA

Algoritmus BMDA (Bivariate Marginal Distribution Algorithm) [57] je založen na párových závislostech, každý uzel grafu může záviset na nejvýše jednom rodičovském uzlu (grafem je les). Pravděpodobnostní model je popsán vztahem

$$p(x) = \prod_{r \in R} p_r(x_r) \prod_{i \in V \setminus R} p_{i, e(i)}(x_i | x_{e(i)}), \quad (3.7)$$

kde V je množina všech uzlů v grafu, R je množina kořenových uzlů, $e(i)$ vrací rodičovský uzel uzlu i .

Graf závislostí je budován hladovým algoritmem na základě Pearsonovy χ^2 statistiky.

3.1.3 Vícenásobné závislosti

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

$$p(X) = \prod_{d=1}^D p(X_d | \Pi_{X_d}), \quad (3.8)$$

kde Π_{X_d} označuje množinu uzlů, na nichž závisí X_d .

Algoritmus BOA

V algoritmu BOA (Bayesian Optimization Algorithm) [56] je struktura závislostí zakódovaná ve formě acyklické Bayesovy sítě. K jejímu sestavení je použita Bayes-Dirichletova metrika (BD). Protože BD má tendenci preferovat složitější modely, obvykle bývá stanoven maximální počet rodičovských uzlů [34].

Algoritmus EBNA

Také algoritmus EBNA (Estimation of Bayesian Networks Algorithm) [22] používá k popisu závislostí Bayesovu síť. K jejímu sestavení je ale použito Bayesovské informační kritérium (BIC) a hladový algoritmus. Výhodou BIC oproti BD je to, že preferuje jednoduché modely [34].

3.1.4 Algoritmy EDA pro spojitě problémy

Algoritmy EDA byly primárně navrženy pro řešení problémů s diskretní reprezentací, ale existují i varianty určené specificky pro spojitě úlohy.

Spojitě varianty předešlých algoritmů (př. UMDA_c^G, MIMIC_c^G)

Je použit algoritmus původně popsáný pro binární problémy, avšak jako rozdělení pravděpodobnosti je použito spojitě rozdělení, typicky normální [40, 34]. Jako příklad uvedu pravděpodobnostní model UMDA_c^G (indexy udávají, že se jedná o c -spojitý (*continuous*) algoritmus s G -normálním (*Gaussian*) rozdělením)

$$p(\mathbf{x}) = \prod_{d=1}^D n(x_d; \mu_d, \sigma_d). \quad (3.9)$$

Algoritmus SHCLVND

Také algoritmus SHCLVND (Stochastic Hill Climbing with Learning by Vectors of Normal Distributions) [61] používá normální rozdělení. Střední hodnoty μ_d^g jsou v každé generaci aktualizovány o rozdíl mezi středem vybrané populace b_d^g , škálovaný pevně zvoleným koeficientem $\theta_{move} \in (0; 1)$. Směrodatná odchylka σ_d^g je snížena pevně zvoleným koeficientem $\theta_{reduce} \in (0; 1)$:

$$\mu_d^{g+1} = \mu_d^g + \theta_{move}(b_d^g - \mu_d^g), \quad (3.10)$$

$$\sigma_d^{g+1} = \theta_{reduce}\sigma_d^g. \quad (3.11)$$

Algoritmus EGNA

Algoritmus EGNA (Estimation of Gaussian Network Algorithm) [40] využívá Gaussovské sítě,

$$\mu_d = m_d + \sum_{x_j \in \Pi_{X_d}} b_{jd}(x_j - m_j), \quad (3.12)$$

kde m_d je nepodmíněná střední hodnota X_d , b_{jd} je lineární koeficient reflektující sílu vazby mezi X_j a X_d .

Algoritmus IDEA

Framework IDEA (Iterated Density Estimation Evolutionary Algorithm) [5] zobecňuje a formalizuje popis předešlých algoritmů pomocí operátorů selekce, obnovy populace, ukončovací podmínky, hledání struktury závislostí, hledání hustoty rozdělení pravděpodobnosti a generování nových vzorků.

Algoritmus EMNA

EMNA (Estimation Multivariate Normal Algorithm) [42] je algoritmus využívající jako model vícerozměrné normální rozdělení:

$$P(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mu, \Sigma), \quad (3.13)$$

kde μ je vektor středních hodnot a Σ kovarianční matice.

Algoritmus ESTDA

Alternativa k obvykle užívanému normálnímu rozdělení – Studentovo t rozdělení – je navržena v algoritmu ESTDA (Estimation of Student's t Distribution Algorithm) [45]. Protože Studentovo rozdělení umí vyjádřit závislosti pro mezní hodnoty proměnných, předpokládají, že takový algoritmus bude mít větší diverzitu.

3.2 Algoritmy 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

$$p(\mathbf{x}) = c(F_1(x_1), \dots, F_D(x_D)) \prod_{d=1}^D f_d(x_d), \quad (3.14)$$

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., Rogelio Salinas-Gutiérrez et al., Li-Fang Wang et al.

3.2.1 Li-Fang Wang et al.

Kolektiv okolo Li-Fang Wang se zaměřuje hlavně na různá použití CEDA s archimédovskou kopulí:

Hojně citovaným článkem je [74]. Popisují zde CEDA algoritmus, zabývají se vzorci pro vzorkování dvojrozměrné Gaussovy kopule.

V [73] a [75] prezentují CEDA algoritmus s Gumbelovou, resp. s Claytonovou kopulí. Není zde použito učení parametru θ pro kopuli, pro každou testovací funkci ho autoři pevně zvolili.

V článku [8] zkoumají hybridní model CEDA s Gumbelovou a Claytonovou kopulí. Celá populace je rozdělena na dvě subpopulace, v každé subpopulaci je použita jiná kopule. Sdílení informace mezi subpopulacemi je dosaženo pomocí migrace jedinců.

Ve [32] je testováno použití CEDA (využívající Claytonovu kopuli a Gaussovo marginální rozdělení) k vektorovému kvantování digitálního obrazu. Pomocí evoluce jsou vyvíjena jednotlivá kódová slova.

V [37] navrhuje použití statistického testu ke zlepšení přesnosti pravděpodobnostního modelu. V každé generaci vytvářejí několik pravděpodobnostních modelů a pomocí testu S_n vybírají ten nejvhodnější. Používají archimédovské kopule.

3.2.2 Rogelio Salinas-Gutiérrez et al.

Kolektiv Rogelia Salinas-Gutiérrez se zaměřuje hlavně na grafové modely využívající dvojrozměrné kopule.

Například v [62] je ukázána praktická aplikace – pravděpodobnostní klasifikátor založený na Gaussově kopuli, který je použit ke klasifikaci pixelů v obrázku. Je použito učení se supervizorem.

V článku [63] je prezentován algoritmus CEDA založený na řetězové struktuře typu MIMIC a dvojrozměrných Frankových a Gaussových kopulích. Jako marginální rozdělení používají beta-rozdělení.

V [64] je jako grafový model v algoritmu CEDA použit strom závislostí, jehož uzly obsahují dvojrozměrné Gaussovy kopule.

V [65] popisují použití D-vine kopulí pro CEDA.

3.2.3 Marta Soto et al.

Také Marta Soto a kol. se v oblasti CEDA zabývají grafovými pravděpodobnostními modely, zejména C-vine (*canonical*) a D-vine (*drawable*).¹

Vine jsou závislostní modely vícerozměrné distribuční funkce založené na dekompozici $f(x_1, \dots, x_n)$ na dvojrozměrné kopule a marginální hustoty. Vine n proměnných je zanořená množina stromů T_1, \dots, T_{n-1} , kde hrany stromu T_j jsou uzly stromu T_{j+1} pro $j = 1, \dots, n - 2$. Regulární vine tvoří speciální třídu, ve které jsou dvě hrany stromu j spojeny hranou ve stromu $j + 1$, pouze když tyto hrany sdílejí společný uzel. — [70]

Článek [70] se zabývá CEDA se strukturami C-vine a D-vine, používají několik variant dvojrozměrných kopulí (Gaussova, Studentova, Claytonova, Gumbelova). Autoři toto řešení považují za flexibilnější, než by bylo použití vícerozměrné kopule.

¹ *Vine* anglicky znamená *popínavá rostlina*.

V technické zprávě [29] je přehled důležitých prací zaměřených na kopulové EDA algoritmy publikovaných v letech 2007 – 2012. Je zde uvedeno, že celé téma využití kopulí je stále velmi málo prozkoumáno a zejména je třeba klást důraz na otázku, jak vlastnosti kopule ovlivňují optimalizační proces, a na reprodukovatelnost publikovaných výsledků. Také se domnívají, že velmi přínosná by byla existence veřejně dostupné implementace EDA založené na kopulích (později uveřejnili balíček `copulaedas` pro jazyk R [30]).

3.2.4 Další autoři

V článku [52] je použita EDA založená na Bayesovské síti a archimédovských kopulích. Protože jejich struktura dokáže zachytit pouze pozitivní závislosti, uchovávají pouze kladné hodnoty korelace. Využívají korelační koeficient Kendallovo τ ; aby mohli parametrizovat vícerozměrnou archimédovskou kopuli, počítají průměrné τ všech dvojic proměnných vybraných k utvoření kopule. Konkrétní typ kopule (Claytonova, Frankova, Gumbelova) je vybrán adaptivně podle aktuálních dat. Pro marginální rozdělení je použito beta-rozdělení, přičemž jeho parametry (střední hodnotu μ a rozptyl σ^2) ovlivňují i hodnotami z předešlé generace.

V článku [28] je použit algoritmus CEDA s vícerozměrnou t -kopulí na multikriteriální optimalizaci.

Článek [51] se zaměřuje na algoritmy CEDA užívající vícerozměrné eliptické kopule. Pro podpoření diverzity v populaci dále využili rebelujících jedinců (cílené generování několika nových jedinců, kteří podle aktuální pravděpodobnostního modelu mají jen velmi malou pravděpodobnost výskytu) a částečné restarty algoritmu (za určitých podmínek je část populace nahrazena náhodně vygenerovanými jedinci).

V [10] je použit algoritmus CEDA k urychlení vícekriteriální optimalizace. CEDA (archimédovské kopule) slouží k modelování Pareto fronty, díky tomu je algoritmus schopen navrhovat nové jedince bez dalších výpočtů náročného multikriteriálního algoritmu.

V [78] použili CEDA (Gaussova kopule) ke zlepšení výkonnosti algoritmu AEA použitým k návrhu Butterworthových filtrů. Algoritmus AEA (Alopex-based evolutionary algorithm; Alopex = Algorithm of Pattern Extraction – algoritmus extrakce vzorů) je principiálně podobný simulovanému žíhání. CEDA je použit na makroúrovni ke stavbě pravděpodobnostního modelu, zatímco AEA řídí evoluci na mikroúrovni pomocí extrakce vzorů.

V článku [23] bylo zkoumáno použití několika evolučních technik, včetně CEDA, na kalibraci mikroskopických dopravních modelů. Použili CEDA založený na eliptických kopulích. Bylo zjištěno, že při použití CEDA jsou výsledky kalibrace méně citlivé vůči použití různých metrik, celkový proces kalibrace je tak robustnější.

3.2.5 Shrnutí algoritmů CEDA

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 pro tuto práci 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

řídící parametry (např. ukončující podmínky) – je proto obtížné uskutečnit korektní porovnání rozdílných algoritmů. Převažují testovací úlohy s 10 dimenzemi.

3.3 Paralelní evoluční algoritmy

Existují dva základní přístupy k paralelizaci evolučního algoritmu, a to (viz [2, 69])

- standardní paralelní přístup,
- dekompoziční přístup.

Při **standardním paralelním přístupu** uvažujeme pouze jednu běžící instanci evolučního algoritmu a cílem je snížit výpočetní čas souběžným výpočtem některých částí algoritmu (např. výpočet fitness funkce každého jedince) — toto není náplní mé práce.

Tento text se zaměřuje na **dekompoziční přístup**, konkrétně na *ostrovní model*. Základní charakteristikou dekompozičního přístupu je rozdělení populace mezi více procesorů. Podle typu rozdělení dále rozlišujeme

- jemnozrný paralelní evoluční algoritmus,
- hrubozrný paralelní evoluční algoritmus.

V případě *jemnozrného algoritmu* je na každý procesor přiřazen obvykle pouze jeden jedinec. Jedinec může rekombinovat pouze v rámci svého okolí, velikost tohoto okolí spolu s topologií jsou stěžejními parametry algoritmu. Tyto algoritmy jsou obzvláště vhodné pro architektury s velkým počtem dobře propojených procesorů a pro řešení úloh s velmi náročným výpočtem fitness funkce.

U *hrubozrných algoritmů* – obvykle se pro ně používá označení *ostrovní model* [48] – je globální superpopulace rozdělena na několik menších částí – populací na jednotlivých ostrovech. Každý ostrov vykonává sekvenční evoluční algoritmus pouze nad svou populací. Populace z různých ostrovů jsou od sebe izolovány, toto napomáhá udržení diverzity v globální superpopulaci. Ve stanoveném čase (např. po uskutečnění stanoveného počtu generací nebo po zlepšení fitness o danou hodnotu) dochází ke komunikaci mezi ostrovy – někteří jedinci z jednoho ostrova jsou zkopírováni na jiný ostrov. Tento proces se označuje jako migrace. Způsob výběru a počet migrujících jedinců, to, jak jsou zapojeni do rezidentní populace na cílovém ostrově, způsob volby cílového ostrova, počet ostrovů a podmínka uskutečnění migrace – toto vše jsou dodatečné parametry ostrovního evolučního algoritmu, které má navíc oproti své sekvenční verzi.

3.3.1 Algoritmy EDA s migrací pravděpodobnostní modelů

Doposud málo testovanou úpravou algoritmů EDA je použití migrace části nebo celého pravděpodobnostního modelu místo migrace jedinců.

Paralelní diskrétní UMDA

Princip migrace diskrétních pravděpodobnostních modelů byl poprvé navržen v [14]. Jako sekvenční algoritmus na každém ostrově byl použit algoritmus UMDA; nový model byl ovlivněn sousedním ostrovem v procesu konvexní kombinace. Tento postup byl shledán přínosným ve srovnání s tradiční migrací jedinců.

Později byl proces kombinace pravděpodobnostní modelů ze dvou různých ostrovů vylepšen o metody lokálního prohledávání [15]. Motivací je rozhodnout, které části příchozího modelu mají vliv na zlepšení rezidentního modelu a při kombinaci použít pouze je.

Paralelní diskrétní BMDA

V článku [66] je navržen diskrétní paralelní algoritmus BMDA s migrací modelů. Pravděpodobnostní model M se skládá ze dvou částí,

$$M = (G, \Theta), \quad (3.15)$$

kde G je graf závislostí a Θ množina parametrů.

Množiny parametrů Θ_R (rezident), Θ_I (imigrant) jsou kombinovány konvexní kombinací přes adaptační koeficient β

$$\Theta = \beta\Theta_R + (1 - \beta)\Theta_I. \quad (3.16)$$

Proces kombinování grafů závislostí G_R, G_I není tak přímočarý, je založen na kombinaci hodnot χ^2 statistiky

$$\chi_{i,j}^2 = \beta\chi_{i_R,j_R}^2 + (1 - \beta)\chi_{i_I,j_I}^2 \quad (3.17)$$

a podle aktualizovaných hodnot χ^2 je hladovým algoritmem vytvořen nový graf závislostí.

V [35] je navrženo další zjednodušení agregace pravděpodobnostních modelů v paralelním algoritmu BMDA. Při migraci je posílána kontingenční tabulka. Na cílovém ostrově jsou kontingenční tabulky zkombinovány, ze zkombinované kontingenční tabulky je spočítána nová tabulka hodnot χ^2 . Podle aktualizovaných hodnot χ^2 je vygenerován nový model závislostí a nová množina parametrů.

Paralelizace algoritmů EDA pro spojité úlohy

V článku [19] je navržen hybridní migrující algoritmus na bázi EDA a PSO (Particle Swarm Optimization). Autoři uvažují tři přístupy k migraci: posílání jedinců; nebo posílání modelů s kombinací modelů; nebo posílání modelu, z něhož jsou po doručení navzorkováni jedinci nahrazující nejhorší část rezidentní populace.

Studie [54] porovnává migraci jedinců a migraci pravděpodobnostních modelů v algoritmech EDA pro spojité 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.

3.4 Porovnávání variant algoritmů

Všechny evoluční algoritmy patří mezi stochastické optimalizační metody. S přihlédnutím k „No free lunch“ teorému [76] je zřejmá nutnost při porovnávání jednotlivých algoritmů vybrat vhodnou sadu testovacích úloh a specifikovat dostatečný počet běhů algoritmů a ukončující podmínku optimalizace.

Obvykle se používá alespoň 30 nebo 50 běhů.

Používají se zejména dva typy ukončujících podmínek: fixní počet evaluací fitness funkce vykonaných během jednotlivých běhů, nebo dosažení požadované chyby vzhledem ke známé hodnotě řešení, popř. jejich kombinace.

U diskrétních optimalizačních úloh bývá sledován počet evaluací fitness funkce, potřebný k dosažení globálního optima.

U optimalizačních úloh se spojitým definičním oborem je obvykle měřena velikost chyby fitness funkce nalezeného kandidátního vůči známému optimálnímu řešení po dosažení stanoveného počtu evaluací fitness funkce.

3.4.1 Klasické testovací úlohy

Jde o základní testovací spojitě funkce více proměnných často používané v literatuře. Vzhledem k dobře známým typickým tvarům těchto funkcí lze snadno posoudit silné a slabé stránky testovaného algoritmu.

Sphere Nejjednodušší testovací funkce, je tvořena hladkou plochou s jedním optimem. Viz obrázek 3.2.

$$f_{Sphere}(\mathbf{x}) = \sum_{d=1}^D x_d^2, \quad x_d \in \langle -100; 100 \rangle. \quad (3.18)$$

Ackley Velmi mnoho nevýrazných lokálních optim, uprostřed výrazné globální optimum. Viz obrázek 3.3.

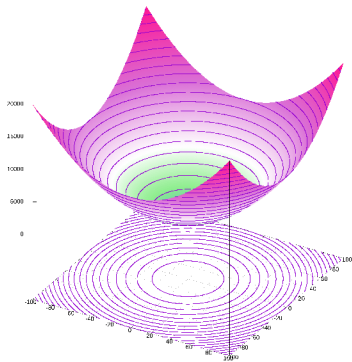
$$f_{Ackley}(\mathbf{x}) = -20 e^{-0.2 \sqrt{\frac{1}{D} \sum_{d=1}^D x_d^2}} - e^{\frac{1}{D} \sum_{d=1}^D \cos(2\pi x_d)} + 20 + e, \quad x_d \in \langle -32; 32 \rangle. \quad (3.19)$$

Schwefel Klamná úloha, optimum se nachází výrazně mimo střed stanoveného prostoru. Druhé nejlepší lokální optimum se nachází na zcela opačné straně prohledávaného prostoru. Extrémy na okrajích stanoveného prostoru jsou výrazné a daleko od sebe, ve středu se odstup mezi nimi zmenšuje a jsou stále méně výrazné. Viz obrázek 3.4.

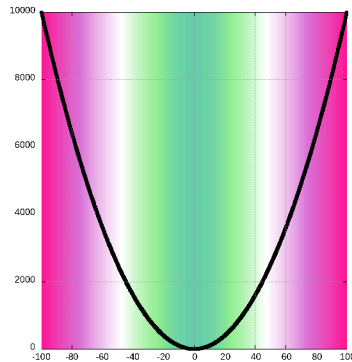
$$f_{Schwefel}(\mathbf{x}) = 418.982888 D - \sum_{d=1}^D x_d \sin \sqrt{|x_d|}, \quad x_d \in \langle -500; 500 \rangle. \quad (3.20)$$

Rosenbrock Velmi obtížná funkce, má jedno optimum, které se ale nachází v parabolicky zakřiveném úzkém údolí. Algoritmy, které používají pohyb jedinců ve směru přímk, mívají velký problém udržet populaci uvnitř údolí. Na svazích údolí se fitness velmi rychle zhoršuje. Viz obrázek 3.5.

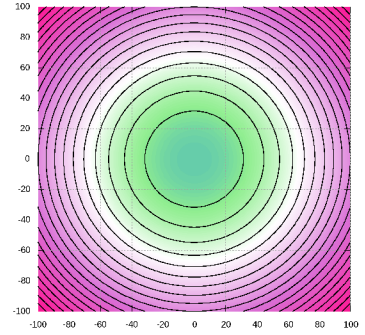
$$f_{Rosenbrock}(\mathbf{x}) = \sum_{d=1}^{D-1} (100(x_d^2 - x_{d+1})^2 + (x_d - 1)^2), \quad x_d \in \langle -10; 10 \rangle. \quad (3.21)$$



(a) Prostorový graf (funkce dvou proměnných).

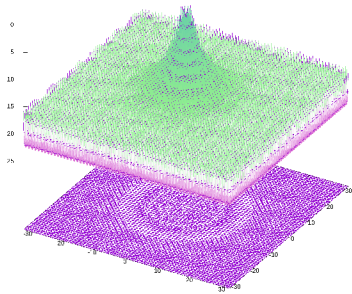


(b) Graf funkce jedné proměnné.

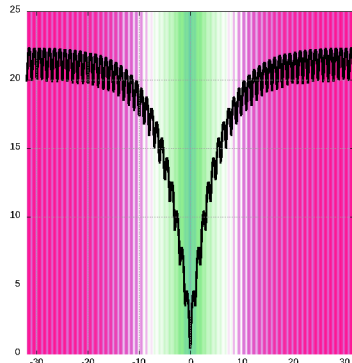


(c) Vrstevnicový graf (funkce dvou proměnných).

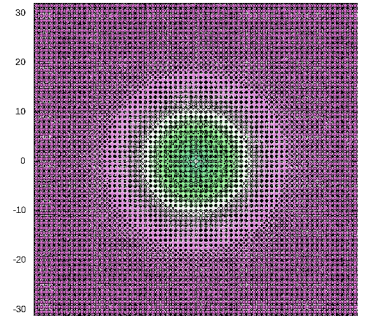
Obrázek 3.2: Zobrazení testovací funkce Sphere.



(a) Prostorový graf (funkce dvou proměnných).

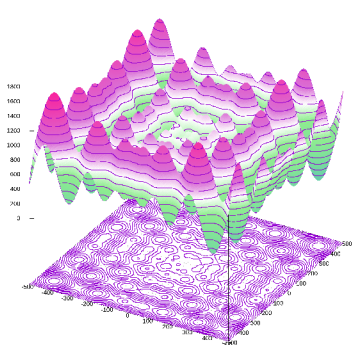


(b) Graf funkce jedné proměnné.

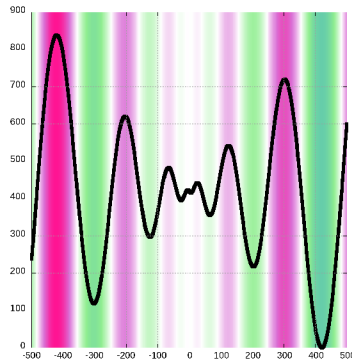


(c) Vrstevnicový graf (funkce dvou proměnných).

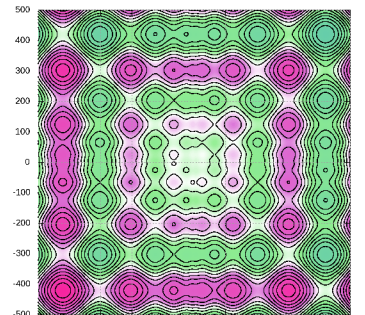
Obrázek 3.3: Zobrazení testovací funkce Ackley.



(a) Prostorový graf (funkce dvou proměnných).

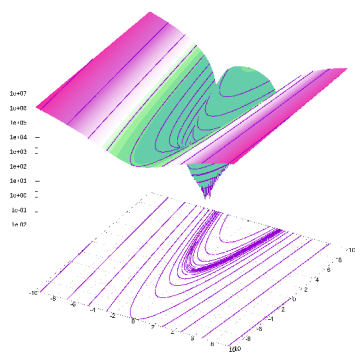


(b) Graf funkce jedné proměnné.

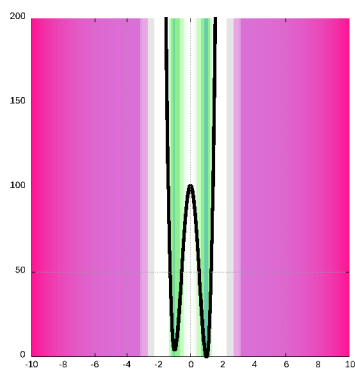


(c) Vrstevnicový graf (funkce dvou proměnných).

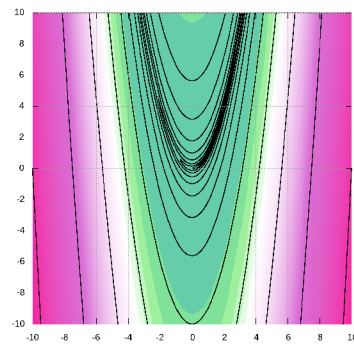
Obrázek 3.4: Zobrazení testovací funkce Schwefel.



(a) Prostorový graf (funkce dvou proměnných).

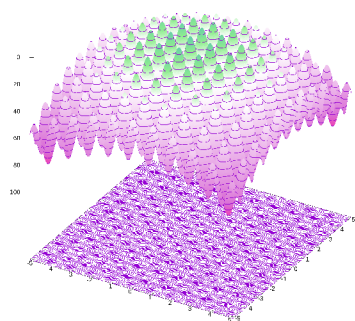


(b) Řez grafem pro $x_2 = 1$.

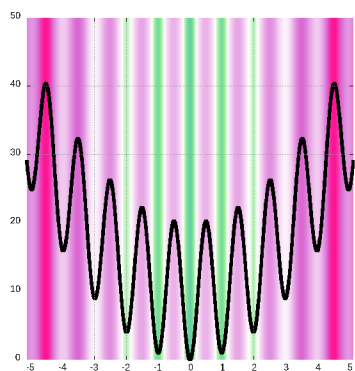


(c) Vrstevnicový graf (funkce dvou proměnných).

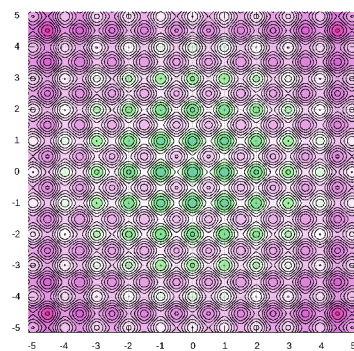
Obrázek 3.5: Zobrazení testovací funkce Rosenbrock.



(a) Prostorový graf (funkce dvou proměnných).

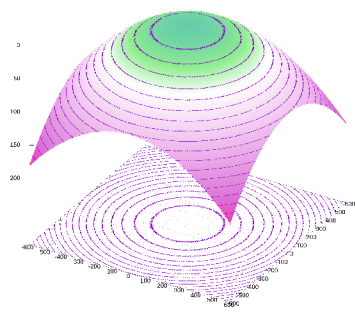


(b) Graf funkce jedné proměnné.

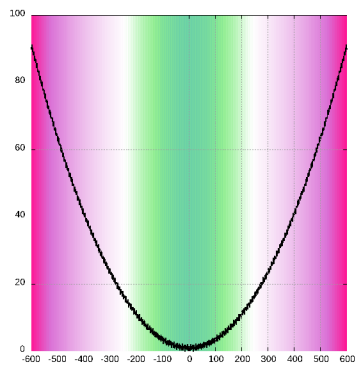


(c) Vrstevnicový graf (funkce dvou proměnných).

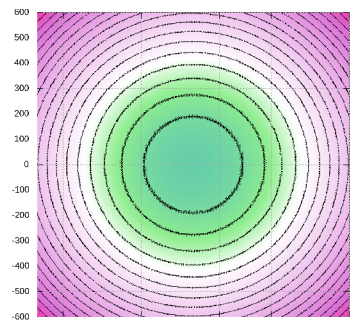
Obrázek 3.6: Zobrazení testovací funkce Rastrigin.



(a) Prostorový graf (funkce dvou proměnných).



(b) Graf funkce jedné proměnné.



(c) Vrstevnicový graf (funkce dvou proměnných).

Obrázek 3.7: Zobrazení testovací funkce Griewank.

Rastrigin Úloha tvořená množstvím lokálních extrémů, ve středu stanovené definiční oblasti proměnných obsahuje několik výraznějších optim a globální extrém. Viz obrázek 3.6.

$$f_{Rastrigin}(\mathbf{x}) = \sum_{d=1}^D (x_d^2 - 10 \cos(2\pi x_d) + 10), \quad x_d \in \langle -5.12; 5.12 \rangle. \quad (3.22)$$

Griewank Relativně jednoduchá úloha, její tvar lze popsat jako sférickou plochu s velkým počtem malých lokálních extrémů. Viz obrázek 3.7.

$$f_{Griewank}(\mathbf{x}) = 1 + \frac{1}{4000} \sum_{d=1}^D x_d^2 + \prod_{d=1}^D \frac{\cos x_d}{\sqrt{d}}, \quad x_d \in \langle -600; 600 \rangle. \quad (3.23)$$

3.4.2 Sada CEC 2013

Sada testovacích úloh CEC 2013 byla původně určena pro porovnání optimalizačních algoritmů v rámci konference CEC '13 [43], ale díky svým vlastnostem bývá často používána i v dalších letech. V jejím použití spatřuji dva benefity – jedním je to, že jde o testovací funkce s různou složitostí ve třech třídách komplexity. Druhým benefitem je to, že součástí sady je i přesný popis, jak prezentovat získané výsledky, je tedy snadné spravedlivě porovnat jakékoliv dva algoritmy, které byly na tuto sadu úloh použity (oproti tomu u klasických testovacích úloh je situace složitější – i když autoři použijí stejnou úlohu, stále se někdy vyskytují rozdíly v počtu evaluací fitness funkce nebo v definičním oboru úlohy). Obdobná sada testovacích úloh byla navržena i v jiných letech, největšího rozšíření se z nich dočkala sada z roku 2005, nicméně ta už se dnes používá méně často.

Sadu CEC 2013 tvoří 28 funkcí, rozdělených do tří kategorií: unimodální, základní multimodální a složené funkce, viz tabulka 3.1.

Součástí pravidel pro tuto sadu testovacích úloh je i stanovení počtu nezávislých optimalizačních běhů. Každý optimalizační běh je limitován počtem evaluací fitness funkce, které může vykonat. Povolený počet evaluací závisí na velikosti řešeného problému, pro 10 D to je 100000 evaluací, pro 30 D 300000 evaluací a pro 50 D je limitem 500000 evaluací.

Počet běhů je stanoven na 51. Domnívám se, že toto číslo nebylo zvoleno samoúčelně. Kromě očekávaných vlastností – aby byl počet běhů dostatečný pro statistickou významnost výsledků, ale současně aby se nevykonávalo zbytečně mnoho výpočtů – má i jednu příjemnou vlastnost navíc. Je snadné určit medián – je to přímo 26. člen seřazené posloupnosti. Proto (a pro jednotnost postupu) budu tento počet nezávislých běhů používat ve všech svých testech.

3.5 Testování statistických hypotéz

Pro porovnání účinnosti dvou či více evolučních algoritmů mezi sebou je adekvátní použít inferenční statistiky. Jde o techniku, která umožňuje provádět zobecněné závěry nad základními soubory (populacemi) dat na základě výběrových souborů, které jsou k dispozici. Je možné použít dvě metodiky – testování hypotéz a intervalový odhad parametrů základních souborů. Nejvíce se používá testování hypotéz – umožňují formulovat tvrzení o vlastnostech rozdělení náhodného výběru. Pokud je rozdělení známé, je možné přímo pracovat s hodnotami parametrů uvažovaného rozdělení.

Tabulka 3.1: Seznam funkcí testovací sady CEC 2013 [43].

	Funkce
f1	Sphere Function
f2	Rotated High Conditioned Elliptic Function
f3	Rotated Bent Cigar Function
f4	Rotated Discus Function
f5	Different Powers Function
f6	Rotated Rosenbrock's Function
f7	Rotated Schaffers F7 Function
f8	Rotated Ackley's Function
f9	Rotated Weierstrass Function
f10	Rotated Griewank's Function
f11	Rastrigin's Function
f12	Rotated Rastrigin's Function
f13	Non-Continuous Rotated Rastrigin's Function
f14	Schwefel's Function
f15	Rotated Schwefel's Function
f16	Rotated Katsuura Function
f17	Lunacek bi-Rastrigin Function
f18	Rotated Lunacek bi-Rastrigin Function
f19	Expanded Griewank's plus Rosenbrock's Function
f20	Expanded Scaffer's F6 Function
f21	Composition Function 1 (n=5, Rotated) — (Rosenbrock + Different Powers + Bent Cigar + Discus + Sphere)
f22	Composition Function 2 (n=3, Unrotated) — (Schwefel + Schwefel + Schwefel)
f23	Composition Function 3 (n=3, Rotated) — (Schwefel + Schwefel + Schwefel)
f24	Composition Function 4 (n=3, Rotated) — (Schwefel + Rastrigin + Weierstrass)
f25	Composition Function 5 (n=3, Rotated) — (Schwefel + Rastrigin + Weierstrass)
f26	Composition Function 6 (n=5, Rotated) — (Schwefel + Rastrigin + High Conditioned Elliptic + Weierstrass + Griewank)
f27	Composition Function 7 (n=5, Rotated) — (Griewank + Rastrigin + Schwefel + Weierstrass + Sphere)
f28	Composition Function 8 (n=5, Rotated) — (Griewank & Rosenbrock + Schaffers F7 + Schwefel + Expanded Scaffer's F6 + Sphere)
	Stavový prostor: $\langle -100, 100 \rangle^D$

Formulují se dvě statistické hypotézy [18]:

- nulová hypotéza H_0 , která vyjadřuje shodnost srovnávaných souborů nebo jejich parametrů (obecně hypotézu, o které se předpokládá, že bude odmítnuta),
- alternativní hypotéza H_1 předpokládá odlišnost srovnávaných souborů nebo jejich parametrů (je to tedy hypotéza, jejíž platnost se očekává).

Pro rozhodnutí o zamítnutí hypotéz se používá různých statistických procedur – testů – pro něž se musí předem stanovit tzv. hladina významnosti α , na níž je toto zamítnutí platné. Je třeba zvolit vhodné testové kritérium a získat aktuální hodnotu testového kritéria. H_0 se zamítá, pokud hodnota testového kritéria padne do kritické oblasti vymezené parametrem α . Je možný i alternativní postup, kdy se stanovuje skutečná pravděpodobnost zamítnutí H_0 (tzv. čistý test významnosti): p -hodnota – pokud je menší jako α , H_0 se zamítá.

Jinak řečeno statistická významnost je pravděpodobnost, s jakou bychom – za předpokladu pravdivosti nulové hypotézy – mohli obdržet data odporující nulové hypotéze stejně či ještě více než pozorovaná data [83, 82]. Jde tedy o podmíněnou pravděpodobnost získání dat s testovou statistikou stejnou, jako je naše, nebo „horší“ při platnosti nulové hypotézy v základním souboru $P(D|H_0)$ [11, 46]. Naše uvažování je běžně řízeno následující logikou: je-li statistická významnost nízká (většinou menší než 5 %), nulová hypotéza pro základní soubor nejspíš neplatí (v praxi při malé hodnotě vypočtené statistické významnosti, nejčastěji pod 0.05, říkáme, že výsledek je statisticky významný, a naopak při větší nebo rovné 0.05 říkáme, že je statisticky nevýznamný), protože získat náš výběr z takového základního souboru je velmi nepravděpodobné (ale ne nemožné). [71]

Následuje výčet často používaných statistických testů (dvouvýběrové testy mají pro moji práci větší význam, budou následně popsány podrobněji):

Parametrické testy

- testy normality — zda lze rozdělení dat považovat za normální,
- F-test — testuje významnost rozdílu mezi dvěma rozptyly,
- jednovýběrový t-test — testuje, zda se výběrový průměr liší od zadané hodnoty,
- párový t-test — testuje velikost rozdílu párových hodnot (typicky se provádí dvě měření u jednoho výběrového souboru – před pokusným zásahem a po něm),
- dvouvýběrový t-test — testuje významnost rozdílu dvou výběrových průměrů.

Neparametrické testy

- Wilcoxonův test pro párové hodnoty (Wilcoxon signed-rank test) — alternativa párového t-testu pro vzorky nevyhovující normálnímu rozdělení,
- dvouvýběrový Wilcoxonův test (Wilcoxon rank-sum test) – alternativa dvouvýběrového t-testu pro vzorky nevyhovující normálnímu rozdělení.

3.5.1 Dvouvýběrový t-test

Dvouvýběrový t-test slouží k ověření, zda dvě normální rozdělení, z nichž pocházejí dva zadané nezávislé náhodné výběry, mají stejné střední hodnoty. Tento test existuje ve dvou

variantách, podle toho, jestli zkoumaná normální rozdělení mají, nebo nemají stejný rozptyl. Shodnost rozptylů bývá obvykle testována pomocí F-testu. Testovacím kritériem t-testu je hodnota t , v případě shodných rozptylů platí rovnice

$$t = \frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2} \frac{n_1+n_2}{n_1n_2}}}, \quad (3.24)$$

kde \bar{x}_1 , \bar{x}_2 jsou aritmetické průměry, s_1^2 , s_2^2 výběrové rozptyly a n_1 , n_2 počet členů obou náhodných souborů.

V případě, že náhodné soubory mají různé rozptyly, je hodnota testovacího kritéria t udána rovnicí

$$t = \frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}. \quad (3.25)$$

Platí-li, že oba náhodné soubory jsou stejně velké ($n_1 = n_2 = n$), potom se vztah pro výpočet testovacího kritéria t (v případě stejných i nestejných rozptylů) dále zjednodušuje na

$$t = \frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{s_1^2 + s_2^2}{n}}}. \quad (3.26)$$

Hodnota testovacího kritéria t je následně porovnána s tabelovanou kritickou hodnotou na zvolené hladině významnosti $t_{1-\alpha/2}(\nu)$. Je-li

- $t \leq t_{1-\alpha/2}(\nu)$, potom je rozdíl statisticky nevýznamný (platí H_0),
- $t > t_{1-\alpha/2}(\nu)$, potom se jedná o rozdíl statisticky významný (platí H_1).

3.5.2 Wilcoxonův dvouvýběrový test

Wilcoxonův dvouvýběrový test (jinak označován též jako Mann–Whitney U test, Mann–Whitney–Wilcoxon, Wilcoxonův rank-sum test) ověřuje, zda dva náhodné výběry mají shodné rozdělení pravděpodobnosti. Je citlivý zejména vůči posunu, méně vůči různým rozptylům. Tento test má podobné použití jako dvouvýběrový t-test, ale na rozdíl od něho se jedná o test neparametrický, který nevyžaduje, aby data pocházela z konkrétního typu rozdělení.

Za předpokladu stejného tvaru obou rozdělení je tento test testem shody mediánů.

Výpočet tohoto testu je založen na hodnotách U_1 a U_2 . Tyto hodnoty jsou vztaheny ke vzájemnému srovnání pořadí hodnot z testových souborů, viz [80].

Testovým kritériem je hodnota $U = \min(U_1, U_2)$. Tato hodnota je porovnána vůči tabelovaným kritickým hodnotám $U_{n_1, n_2, \alpha}^{Crit}$.

Pokud

- $U \leq U_{n_1, n_2, \alpha}^{Crit}$, pak je odlišnost mezi soubory statisticky významná (platí H_1),
- $U > U_{n_1, n_2, \alpha}^{Crit}$, pak je odlišnost nevýznamná a platí H_0 .

3.5.3 Vícenásobné testy typu Friedman

Friedmanův test [24, 25] je neparametrická statistická procedura, která umožňuje detekovat významné rozdíly mezi dvěma a více soubory. Dává také představu v pořadové metrice o kandidátu na nejlepší algoritmus.

Postup výpočtu Uvažujme p různých algoritmů porovnávaných na n úlohách. Prvním krokem Friedmanova testu je pro každou úlohu i , $1 \leq i \leq n$, stanovit pořadí r_j^i algoritmu j (nejlepšímu náleží $r_j^i = 1$, nejhoršímu $r_j^i = p$, v případě shody je pořadí dělené). Následně je pro každý algoritmus j vypočteno průměrné pořadí R_j :

$$R_j = \frac{1}{n} \sum_{i=1}^n r_j^i. \quad (3.27)$$

Hodnota testové statistiky F_f je dána rovnicí

$$F_f = \frac{12n}{p(p+1)} \sum_{j=1}^p R_j^2 - 3n(p+1). \quad (3.28)$$

Statistika F_f má rozdělení χ^2 s $p-1$ stupni volnosti (pro dostatečně vysoká n a p).

Pokud je F_f větší jak kritická hodnota, zamítáme hypotézu H_0 , že se testované algoritmy vzájemně neliší, a přijímáme alternativní hypotézu H_1 , že se algoritmy vzájemně liší. Pokud je použita hladina významnosti α , platí pro tento případ, že α je větší jak hodnota p -hodnota.

Pokud bychom případě potřebovali zjistit statisticky významné difference mezi nejlepším algoritmem a ostatními algoritmy, je třeba použít navazující post-hoc testy, například proceduru Bonferroni–Dunn nebo proceduru podle Holma, viz [17].

Kapitola 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 rozděleny do dvou nezávislých dílčích etap, jak ukazuje alg. 4.1.

Vytváření dekomponovaného modelu je ilustrováno na obrázku 4.1. Z populace jsou vybráni slibní jedinci a pouze ti 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) .

4.1.1 Parametry marginálních rozdělení

Jako marginální rozdělení používám normální rozdělení (v souladu s doporučením v literatuře [41]). Toto rozdělení má pro každou dimenzi $d = 1, \dots, D$ dva parametry, střední hodnotu μ_d a směrodatnou odchylku σ_d . Tyto parametry jsou spočítány z hodnot jedinců ve vybrané subpopulaci $\mathbf{x}_p^S \in S(P(g))$:

$$\mu_d = \frac{1}{N'} \sum_{p=1}^{N'} (x_d)_p^S, \quad (4.1)$$

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. \dashrightarrow vyber kandidátní řešení $S(P(g))$, $|S(P(g))| = N'$

 // vytvoř pravděpodobnostní model M :
 4. \dashrightarrow urči parametry marginálních rozdělání F_d
 5. \dashrightarrow urči parametry kopule C

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

$$\sigma_d = \sqrt{\frac{1}{N'} \sum_{p=1}^{N'} ((x_d)_p^S - \mu_d)^2}. \quad (4.2)$$

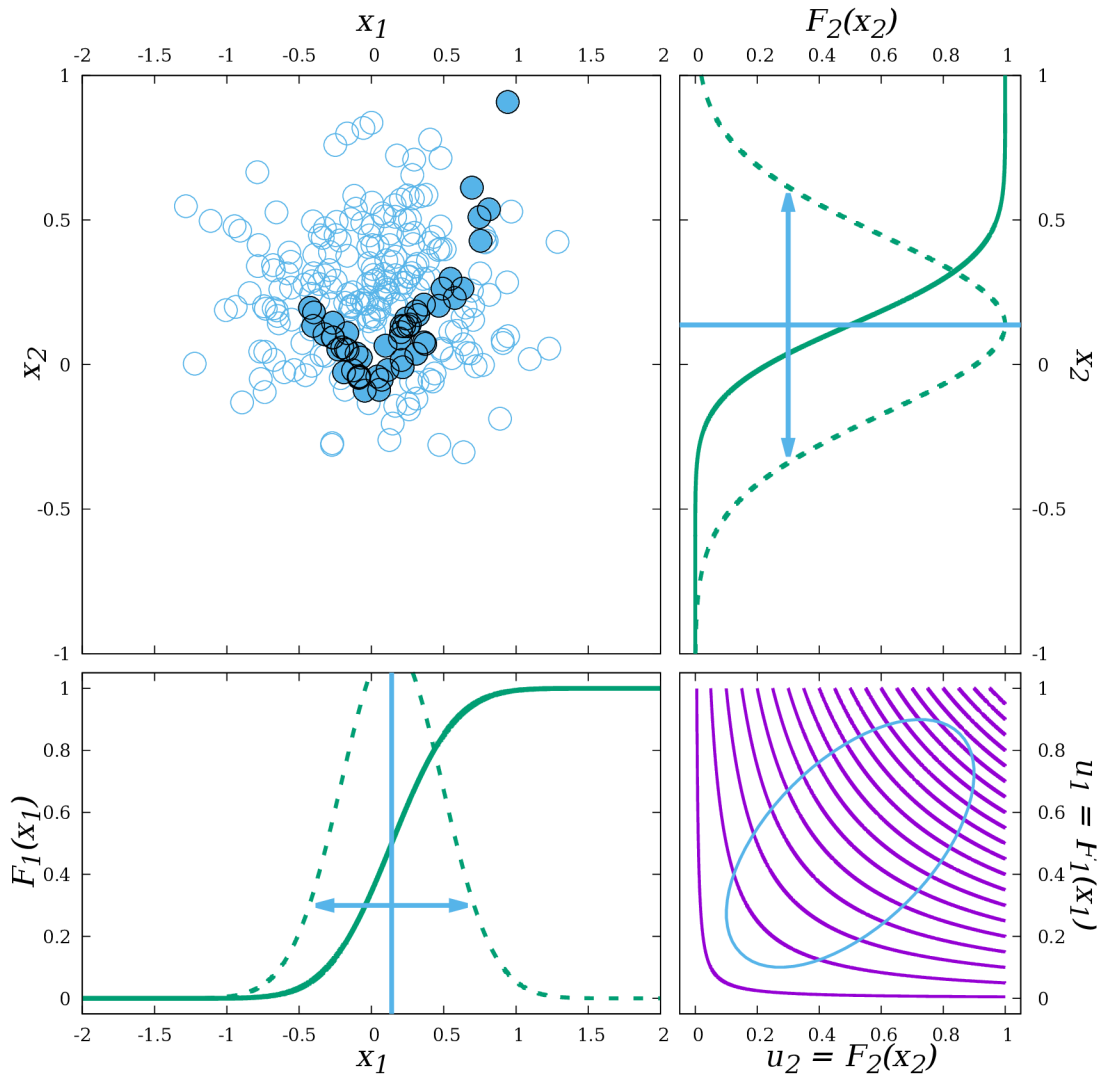
4.1.2 Parametry kopule

Parametry kopule jsou z vybraných jedinců $\mathbf{x}_p^S \in S(P(g))$ odvozeny pomocí korelačního koeficientu Kendallovo τ .

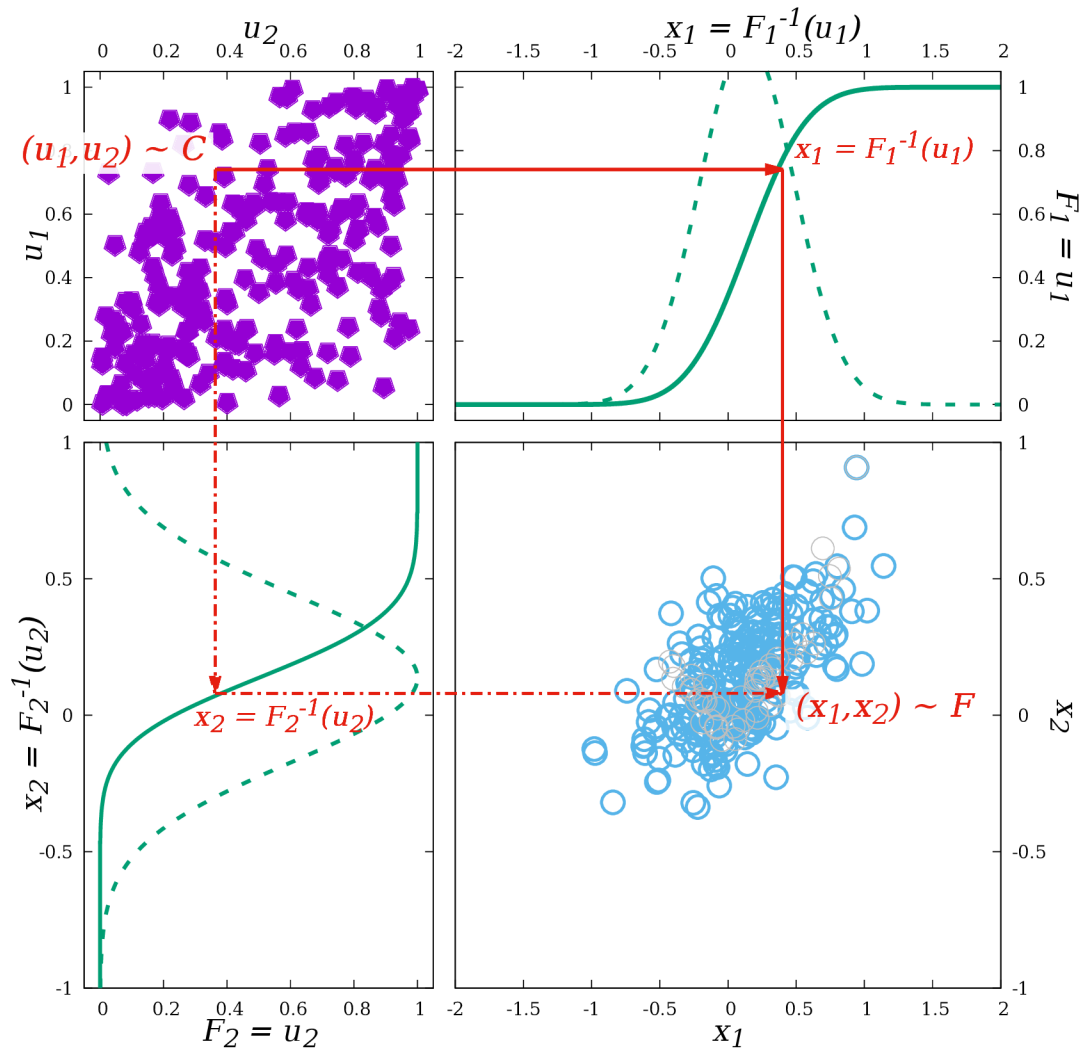
Kendalovo τ

Mějme dvě náhodné veličiny \mathbb{X} a \mathbb{Y} se vzorky x_i, y_i , počet vzorků je N' . Dvojici pozorování (x_i, y_i) a (x_j, y_j) nazveme *souhlasný pár*, pokud $x_i < x_j$ a zároveň $y_i < y_j$ nebo $x_i > x_j$ a $y_i > y_j$. Pár je *nesouhlasný*, pokud $x_i < x_j$ a zároveň $y_i > y_j$ nebo $x_i > x_j$ a $y_i < y_j$. Pokud $x_i = x_j$ nebo $y_i = y_j$, není pár ani souhlasný, ani nesouhlasný. Kendallovo τ je [39]:

$$\tau = \frac{(\text{počet souhlasných párů}) - (\text{počet nesouhlasných párů})}{\frac{N'(N'-1)}{2}}. \quad (4.3)$$



Obrázek 4.1: Ilustrace modelování sdruženého 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 rozdělení 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 rozdělení 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.

Pro implementaci se často užívá vztah pomocí funkce signum [12], tento vztah už uvedu přímo ve tvaru pro výpočet korelace mezi proměnnými $var1$, $var2$ v populaci jedinců \mathbf{x}^S (kde $(x_d)_p^S$ je d -tá proměnná v p -tém vybraném jedinci):

$$\tau_{var1,var2} = \frac{2}{N'(N' - 1)} \sum_{i=2}^{N'} \sum_{j=1}^{i-1} \text{sgn}((x_{var1})_i^S - (x_{var1})_j^S) \cdot \text{sgn}((x_{var2})_i^S - (x_{var2})_j^S). \quad (4.4)$$

Parametry archimédovských kopulí

Archimédovské kopule používají parametr θ , který se odvozuje z korelace ve vstupních datech. Konkrétně je vypočítán korelační koeficient Kendallovo τ (pro tři- a více-rozměrné kopule jsem použil průměrnou hodnotu τ mezi všemi dvojicemi dimenzí, v souladu s [52]).

V literatuře [49] jsou uvedeny vztahy mezi korelací τ a parametrem θ :

- pro Claytonovu kopuli:

$$\tau_{Clayton} = \frac{\theta}{\theta + 2}, \quad (4.5)$$

- pro Gumbelovu kopuli:

$$\tau_{Gumbel} = \frac{\theta - 1}{\theta}, \quad (4.6)$$

- pro Frankovu kopuli:

$$\tau_{Frank} = 1 + \frac{4}{\theta} \left(\int_0^\theta t (\theta e^t - \theta)^{-1} dt - 1 \right). \quad (4.7)$$

Z těchto vztahů lze odvodit následující vzorce pro výpočet parametru θ ze zjištěné korelace τ :

- pro Claytonovu kopuli:

$$\theta_{Clayton} = \frac{2\tau}{1 - \tau}, \quad (4.8)$$

- pro Gumbelovu kopuli:

$$\theta_{Gumbel} = \frac{1}{1 - \tau}, \quad (4.9)$$

- pro Frankovu kopuli jsem vytvořil tuto aproximaci:

$$\theta_{Frank} \doteq (10^{\arcsin(\tau)} - 1) e. \quad (4.10)$$

Parametry eliptických kopulí

Gaussova kopule je parametrizována pomocí korelační matice R s prvky ρ_{ij} . Hodnota každého prvku ρ_{ij} v matici je odvozena z párové korelace mezi každým párem proměnných i, j za využití Kendallova τ pomocí rovnice (podle [16, 6])

$$\rho_{ij} = \sin \frac{1}{2} \pi \tau_{ij}. \quad (4.11)$$

Pro Studentovu kopuli je také použita korelační matice R , navíc je ještě potřeba specifikovat počet stupňů volnosti ν . Hodnota ν se standardně volí jako $\nu = N' - 1$, kde N' je počet vzorků (zde počet jedinců ve vybrané populaci $S(P(g))$).

4.1.3 Vzorkování kopule – implementace

Vzorkem kopule je D -tice

$$(u_1, \dots, u_D) \in \langle 0; 1 \rangle^D. \quad (4.12)$$

K získání tohoto vzorku pro zvolenou kopuli se známými parametry jsou použity techniky podrobně popsané v kapitole 2.4.3 — alg. 2.5 (vzorkování archimédovských kopulí), alg. 2.6 (vzorkování Gaussovy kopule) a alg. 2.7 (vzorkování Studentovy kopule).

4.1.4 Vzorkování marginálního rozdělení – implementace

Vzorkování marginálních rozdělení F_d je dáno vztahem

$$x_d = F_d^{-1}(u_d) \quad (\text{pro } d = 1, \dots, D), \quad (4.13)$$

kde $u_d \in \langle 0; 1 \rangle$ je vzorkem kopule.

V souladu s doporučením v literatuře [41] používám jako marginální rozdělení normální rozdělení $F_d \equiv \mathcal{N}(\mu, \sigma)$. Dílčí nevýhodou tohoto rozdělení je skutečnost, že poskytuje vzorky z celého reálného oboru, zatímco optimalizační problém je z definice omezen na interval $\langle MIN; MAX \rangle$. Jak tedy rozdělení omezit, aby poskytovalo vzorky pouze z tohoto intervalu?

Konvenčním řešením tohoto problému by bylo nevyhovující vzorek zahodit a vzorkování opakovat, ale protože jsou vlivem korelací mezi sebou provázané všechny dimenze, bylo by nutné (aby nedošlo k porušení korelací) opakovat vzorkování celého jedince, ne jen té proměnné, kde nastal problém. Je evidentní, že tímto přístupem by utrpěla efektivita algoritmu. Navíc není zaručen úspěch vzorkování, protože výskyt nechtěné hodnoty v jedné proměnné znamená, že se vzorkování opakuje, ovšem v novém vzorku se opět může vyskytnout neplatná hodnota.

Dobrym řešením není ani oříznutí na hranicích intervalu $\langle MIN; MAX \rangle$, tím by docházelo k nežádoucímu hromadění jedinců na hranicích prohledávaného prostoru. Navíc i tímto způsobem by došlo k porušení korelací.

Proto jsem navrhl postup, který při vzorkování normálního rozdělení pomocnou hodnotu $u \in \langle 0; 1 \rangle$ transformuje do takového podintervalu, aby bylo zaručeno, že po navzorkování normálního rozdělení bude získaná hodnota x náležet právě do zadaného intervalu $\langle MIN; MAX \rangle$, viz algoritmus 4.2.

Algoritmus 4.2 Vzorkování normálního rozdělení $\mathcal{N}(\mu, \sigma)$ s ošetřením mezí proměnných $x \in \langle MIN; MAX \rangle$.

1. vygeneruj náhodné $u \sim U(0, 1)$
 2. urči $A := \Phi\left(\frac{MIN - \mu}{\sigma}\right)$
 3. urči $B := \Phi\left(\frac{MAX - \mu}{\sigma}\right)$
 4. $\omega := u(B - A) + A$
 5. urči kvantil $z := \Phi^{-1}(\omega)$ z normovaného normálního rozdělení
 6. výsledek je hodnota proměnné $x := z\sigma + \mu$
-

4.1.5 Ošetření neplatných výpočetních operací

Při práci na počítači máme pro reprezentaci reálných čísel k dispozici jen konečný počet cifer. Pracujeme proto s přibližnými (zaokrouhlenými) hodnotami čísel. Zaokrouhlovací chyby vznikají už při zadávání reálných čísel, další pak vznikají v průběhu číselných výpočtů. Není možné jim zabránit, ale můžeme výpočet organizovat tak, abychom jejich vliv minimalizovali.

U eliptických kopulí je prvním krokem vzorkování provedení Choleského dekompozice korelační matice R . Podmínkou Choleského dekompozice je to, že rozkládaná matice musí být pozitivně definitní. Bohužel tato vlastnost není u experimentálně získaných korelačních matic zcela zaručena. Aby mohl výpočet úspěšně pokračovat, je nutné toto zajistit. K tomu používám metodu vlastních čísel (*Eigenvalue Method* podle [60]): korelační matice R je rozložena na vlastní čísla a vlastní vektory, případná nekladná vlastní čísla jsou nahrazena malými kladnými hodnotami, tato čísla a vektory jsou zpětně složeny do nové matice R .

Během výpočtu Choleského dekompozice matice jsou použity i dvě matematické operace, které nejsou definovány pro všechny jejich vstupy, a sice dělení (nulou) a výpočet odmocniny (ze záporného čísla). Po aplikaci opravy na pozitivně definitní matici¹ jsou tyto dva problematické stavy způsobeny pouze numerickými nepřesnostmi při počítání s reálnými čísly. Jako výsledek odmocniny ze záporného čísla lze použít nulu (protože máme zaručeno, že tento stav byl způsoben numerickými nepřesnostmi a číslo pod odmocninou se tak nule velmi blíží). Pro dělení nulou žádná takováto jednoduchá oprava není možná. Důsledkem čehož je to, že nelze zaručit úspěšné vypočítání Choleského dekompozice a tedy ani navzorkování eliptické kopule. Aby ani tehdy algoritmus neuvázl, potřebný vzorek $(u_1, \dots, u_D) \in \langle 0; 1 \rangle^D$ generuji náhodně – nezávisle, bez korelací.

Podobně jako u vzorkování kopule, i vzorkování marginálních rozdělání může vlivem neplatných matematických operací dojít k selhání výpočtu. V takovém případě, aby nedošlo k přerušení celého algoritmu, navzorkuji x_d podle rovnoměrného rozdělání $x_d \sim U(MIN, MAX)$.

¹Jedním ze způsobů, jak rozpoznat, že matice není pozitivně definitní, já právě nemožnost provést Choleského dekompozici z důvodu nedefinovaných operací.

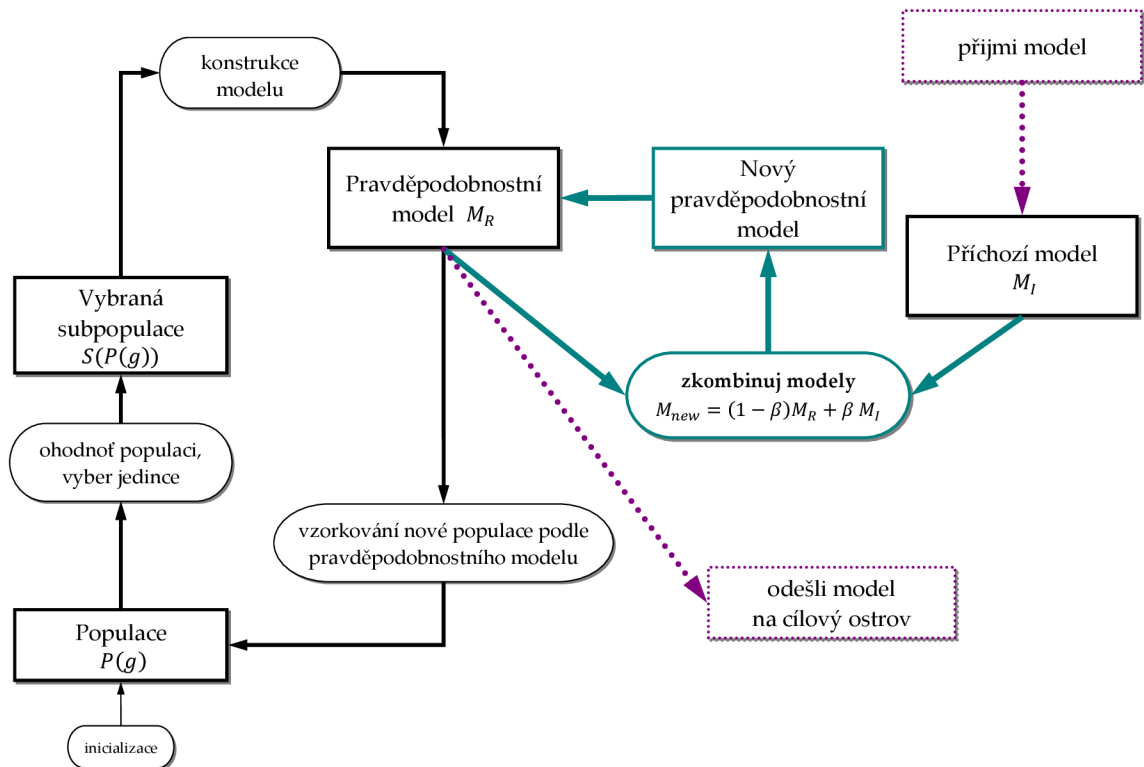
4.2 Paralelní část navrženého algoritmu

V předchozí části je popis sekvenčního jádra, 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 4.3, 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 zpracovávány v rámci jedné generace.



Obrázek 4.3: Schéma algoritmu EDA s migrací modelů.

Algoritmus 4.3 Pseudokód algoritmu EDA s migrací modelů.

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

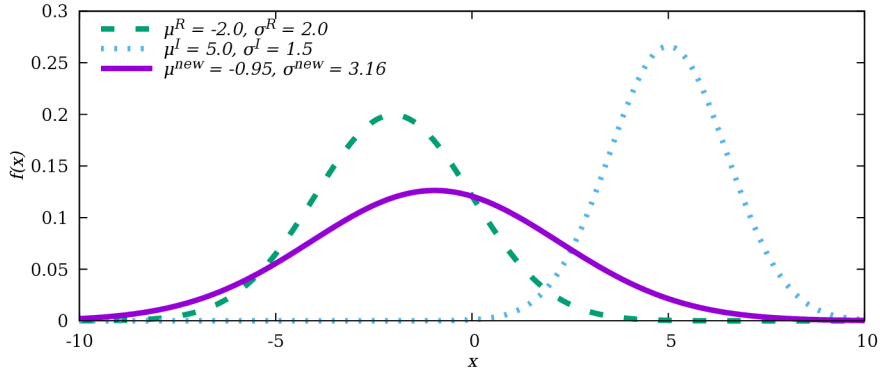
4.2.1 Kombinace pravděpodobnostních modelů

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 odeslán, 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

$$M_R^{new} = (1 - \beta)M_R \circ \beta M_I, \quad (4.14)$$

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.



Obrázek 4.4: Příklad kombinace modelů – normální marginální rozdělení: $M_R \sim \mathcal{N}(-2, 2)$, $M_I \sim \mathcal{N}(5, 1.5)$, $\beta = 0.15$. Vidíme, že v nově vzniklém modelu M_R^{new} dochází oproti výchozímu M_R k mírnému posunu střední hodnoty směrem k M_I a k adekvátnímu zvětšení rozptylu.

Podle [26] jsem pro kombinaci modelů navrhl následující pravidla:

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

$$\mu_d^{new} = (1 - \beta)\mu_d^R + \beta\mu_d^I. \quad (4.15)$$

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

$$\sigma_d^{new} = \sqrt{(1 - \beta) \left((\mu_d^{new} - \mu_d^R)^2 + (\sigma_d^R)^2 \right) + \beta \left((\mu_d^{new} - \mu_d^I)^2 + (\sigma_d^I)^2 \right)}. \quad (4.16)$$

- Aktualizace hodnot prvků korelační matice ρ_{ij} :

$$\rho_{ij}^{new} = (1 - \beta)\rho_{ij}^R + \beta\rho_{ij}^I. \quad (4.17)$$

Hodnotu aktualizacího koeficientu β jsem zvolil jako

$$\beta = \begin{cases} \frac{fit^R}{fit^R + fit^I} & fit^I \leq fit^R, \\ 0.1 & \text{jinak,} \end{cases} \quad (4.18)$$

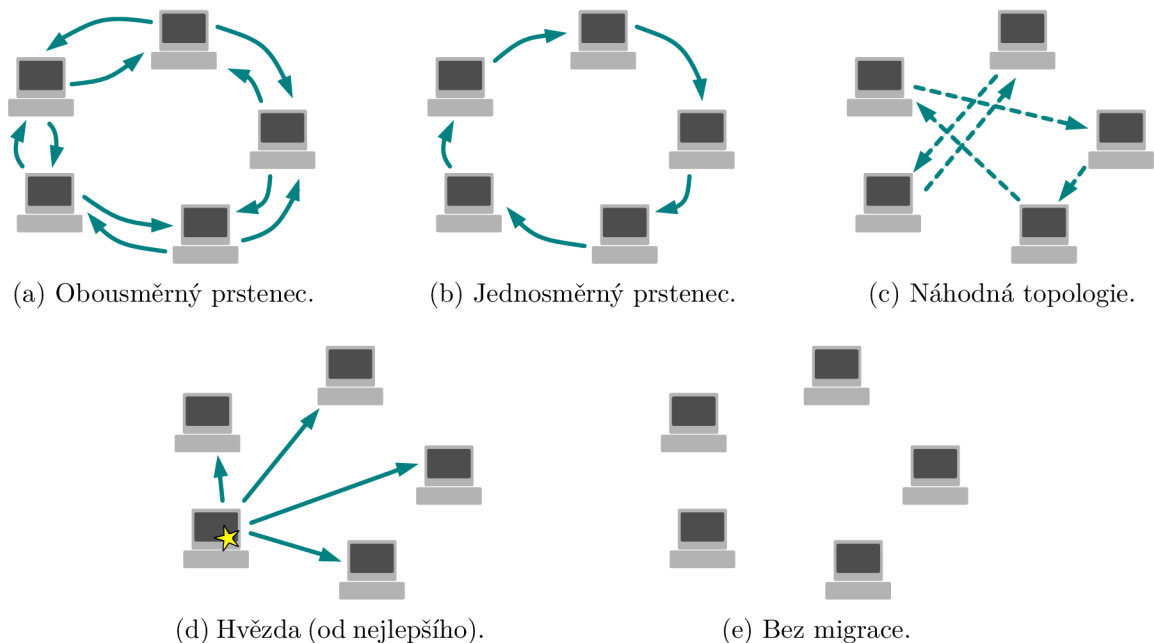
kde fit^R resp. fit^I reprezentují kvalitu subpopulace $S(P(g))$ (vyjádřenou střední hodnotou fitness hodnot vybraných jedinců) na rezidentském resp. imigrantském ostrově.

Na obrázku 4.4 je uveden příklad kombinace dvou normálních marginálních rozdělení podle uvedených rovnic.

4.2.2 Komunikační topologie

Komunikační topologie určuje, jakým způsobem dochází k šíření informace mezi ostrovy. V navrženém systému probíhá komunikace vždy od imigrantského ostrova k rezidentnímu – tyto role nejsou trvalé, každý ostrov může vstupovat do obou rolí.

Například u topologie jednosměrný prstenec je ostrov v roli rezidenta vůči svému předchůdci na prstenci a současně je v roli imigranta vůči svému následníkovi. Naopak u obousměrného prstence je daný ostrov současně v obou rolích vůči každému ze svých sousedů.



Obrázek 4.5: Ilustrace různých topologií.

Zároveň vidíme, že u obousměrného prstence se sousední ostrovy ovlivňují navzájem, zatímco u jednosměrného se informace šíří pouze jedním směrem.

Příliš pomalé šíření informace (úspěšných modelů) mezi ostrovy, může způsobit pomalou konvergenci algoritmu; ale možná naopak právě díky pomalému šíření úspěšných modelů bude udržena vysoká diverzita a díky tomuto bude snížena šance, že algoritmus uváže v lokálním optimu.

Mnou zkoumané topologie jsou znázorněny na obrázku 4.5.

Obousměrný prstenec (biring) Ostrovy jsou uspořádány do kruhu, každý ostrov v rámci jednoho migračního kroku odesílá model svému předchůdci a svému následníkovi (viz obrázek 4.5a).

Jednosměrný prstenec (uniring) Ostrovy jsou uspořádány do kruhu, každý ostrov v rámci jednoho migračního kroku odesílá model svému následníkovi (viz obrázek 4.5b).

Náhodná topologie (random) V čase proměnlivá topologie, cílový ostrov je při každé migraci volen náhodně (viz obrázek 4.5c).

Hvězda (star) V čase adaptivní topologie, migrace je na všech ostrovech aktivována současně (na rozdíl od všech ostatních uvažovaných topologií, v tomto případě je použita synchronní komunikace): nejprve je vybrán aktuálně nejlepší ostrov (ten, který obsahuje globálně nejlepšího jedince), jeho model je následně rozeslán na všechny ostatní ostrovy. V tomto případě je tedy topologie obohacena o další informaci (ostatní zkoumané topologie jsou neinformované). Viz obrázek 4.5d.

Bez migrace (zero) Migrace a tedy ani kombinace modelů není aktivována (viz obrázek 4.5e). Tato topologie je velmi významná jako reference při zkoumání vlivu ostatních topologií a kombinace modelů, neboť ponechává pouze základní ostrovní model bez uplatnění kombinace.

Kapitola 5

Provedené experimenty a jejich vyhodnocení

Významnou část mého výzkumu tvořily experimentální testy s navrženým algoritmem. První část této kapitoly (kap. 5.1) je věnována vhodnému nastavení 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

V této části se blíže zaměřím na vliv sedmi parametrů určujících typ algoritmu:

- typ kopule,
- velikost populace na každém ostrově,
- velikost selekce,
- obnova populace,
- počet ostrovů,
- perioda migrací,
- komunikační topologie.

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. Vzhledem k tomu, že v práci [A] nebyl dostatečný prostor pro publikování konceptu vícenásobné komparace vyšetřovaných variant algoritmů a také vzhledem k částečné změně konceptu obnovy populace, použil jsem analýzu citlivosti jednotlivých parametrů. V každém experimentu je použito jednotné výchozí nastavení parametrů, mění se vždy jen hodnoty daného zkoumaného parametru.

5.1.1 Postup vyhodnocení

Všechny experimenty jsem prováděl na sadě CEC 2013 a také na sadě klasických úloh, viz kap. 3.4. Aby byly výsledky dostatečně průkazné, vždy jsem využíval opakované běhy zkoumaného algoritmu. Pro každou zkoumanou variantu algoritmu a pro každou testovací úlohu bylo provedeno 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 označovanou jako chybová funkce). Těchto 51 hodnot je použito při výpočtu 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. Srovnání je provedeno samostatně pro každou testovací funkci. Jsou použity dva soubory testovacích funkcí: standardní soubor se šesti klasickými funkcemi a soubor CEC 2013 s 28 funkcemi. Pro finální doladění parametrů jsem použil výsledky získané pro soubor CEC 2013.

Výsledky statistických testů prezentuji ve dvou formách, v tabulkách (tab. 5.1 až tab. 5.14) a grafech (obr. 5.2 až obr. 5.15).

Tabulky v každé buňce obsahují (každé testovací funkci odpovídá jeden řádek a každé variantě algoritmu jeden sloupec):

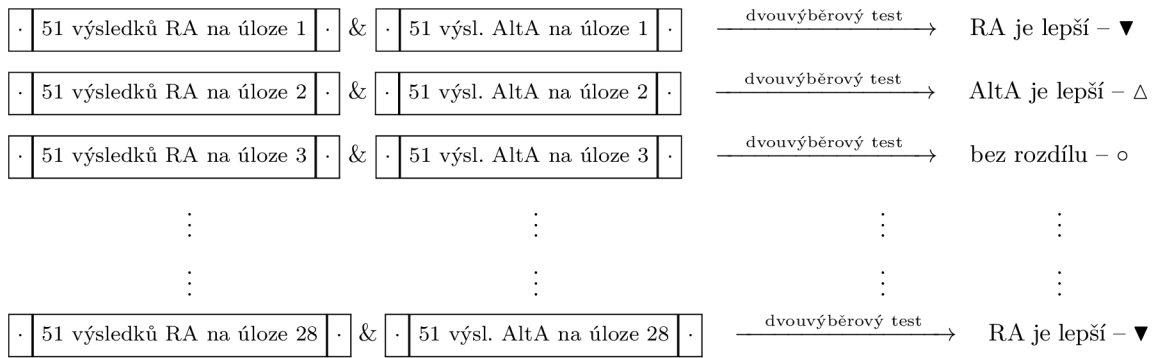
- medián chybové funkce příslušné varianty algoritmu pro příslušnou úlohu,
- výsledek statistického testu – porovnání příslušné verze algoritmu vůči RA získané dvouvýběrovým Wilcoxonovým testem ve třech kategoriích \blacktriangledown , \circ , nebo Δ :
 - \blacktriangledown = RA je lepší než alternativní varianta, počet takových úloh je označen jako B ,
 - Δ = RA je horší, počet takových úloh je označen jako W ,
 - \circ = mezi variantami není statisticky významný rozdíl, počet těchto úloh je označen jako I .

V zápatí tabulky jsou uvedeny celkové počty (B, W, I) jednotlivých kategorií pro každou variantu algoritmu.

Ve spodní (přídavné) části tabulky jsou uvedeny výsledky Friedmanova testu. Pro každou variantu algoritmu je uvedeno průměrné pořadí R_j dosažené na testovaných úlohách a hodnota absolutního pořadí podle dosaženého R_j , nejlepší hodnoty jsou tučně. Poslední řádek každé přídavné tabulky obsahuje spočtenou hodnotu testové statistiky F_f , p -hodnotu, α hodnotu a údaj o platnosti H_0 nebo H_1 .

V **grafech** jsou pro větší přehlednost a rychlou orientaci reprezentovány tabelované výsledky B, W, I . Tyto hodnoty jsou odlišeny barevně, složenými histogramy pro každou variantu algoritmu:

- zelená (horní) část histogramu – B (počet úloh, kdy je referenční algoritmus statisticky významně lepší),



Celkové vyhodnocení (př.):	$B = 22$ (počet ▼)
	$I = 3$ (počet ○)
	$W = 3$ (počet Δ)

Obrázek 5.1: Schéma vyhodnocení výsledků na jednotlivých úlohách pomocí dvouvýběrových testů mezi referenčním algoritmem (RA) a alternativním algoritmem (AltA). Schéma odpovídá sadě CEC 2013 s 28 úlohami.

- žlutá (prostřední) část – I (počet úloh bez statisticky významného rozdílu mezi porovnávanými algoritmy),
- červená (spodní) část histogramu – W (počet úloh, kdy je referenční algoritmus statisticky významně horší).

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

5.1.2 Porovnání vhodnosti užití různých typů kopulí

V první fázi svého výzkumu jsem se zabýval otázkou, jaký typ kopule je nejvhodnější použít. V doposud existujících publikacích si autoři často zvolili (bez další diskuze) jen jednu konkrétní kopuli a tu používali. V zásadě tento postup chybný není, ale bylo by vhodné zdůvodnit, proč používat zrovna tento jeden typ kopule. Na výběr konkrétního typu kopule se zaměřuji v této části své práce.

Jak již bylo zmíněno v kapitole 2.2, 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.

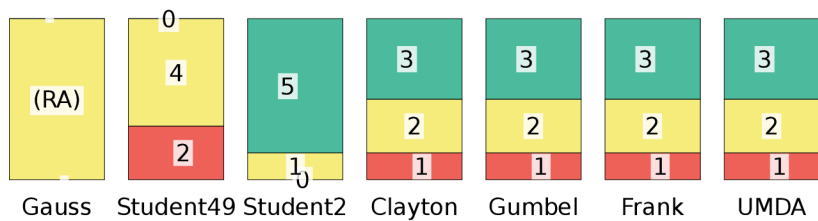
Pro tento pokus jsem zvolil dvě eliptické kopule, Gaussovu a Studentovu (s různými počty stupňů volnosti ν); a dále Claytonu, Gumbelovu a Frankovu kopuli jako nejčastěji používané zástupce rodiny archimédovských kopulí. Jako referenční algoritmus pro srovnání vlivu korelací jsem ještě přidal i algoritmus 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*).

Pro experiment jsem použil toto nastavení:

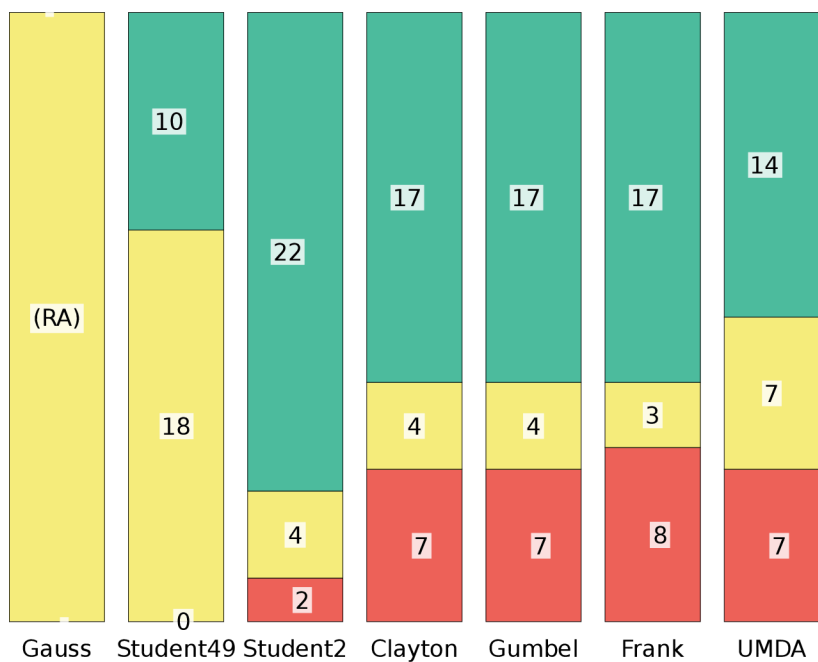
- velikost populace na každém ostrově: 250 jedinců,
- výběr: nejlepších 50 jedinců,
- obnova populace: zachování 3 nejlepší jedinci,
- kopule: Gaussova, Studentova ($\nu = 49$), Studentova ($\nu = 2$), Claytonova, Gumbelova, Frankova, nebo součinná,
- počet ostrovů: 4,
- topologie: obousměrný prstenec,
- perioda migrací: každých 5 generací,
- počet dimenzí: 10,
- počet evaluací fitness: 100000.

Souhrnné výsledky Wilcoxonových testů jsou na obrázku 5.2 (pro klasické testovací úlohy) a na obrázku 5.3 (pro sadu CEC 2013), v tabulkách 5.1 a 5.2 jsou uvedeny mediány pro každou testovací úlohu a jednotlivé výsledky Wilcoxonových testů.

Získané výsledky potvrdily variantu s **Gaussovou kopulí** jako tu nejlepší. Na sadě CEC 2013 toto platí jednoznačně, na sadě klasických testovacích úloh je Studentova s $\nu = 49$ mírně lepší. Dále je prezentováno, že mezi Gaussovou a Studentovou se stupni volnosti $\nu = 49$ velmi často není významný rozdíl – toto je zdůvodnitelné tím, že pro vysoké hodnoty ν je t-rozdělení velmi blízké normálnímu rozdělení, lze proto očekávat, že obě kopule se budou podobat a podají obdobný výkon. To potvrzují i blízké hodnoty pořadí získané Friedmanovým testem.



Obrázek 5.2: Výsledky experimentu zkoumajícího vliv různých typů **kopulí** – souhrnné porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.



Obrázek 5.3: Výsledky experimentu zkoumajícího vliv různých typů **kopulí** – souhrnné porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

Tabulka 5.1: Výsledky experimentu zkoumajícího vliv různých typů **kopulí** – mediány a porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.

	Gauss	Student49	Student2	Clayton	Gumbel	Frank	UMDA
Sphere	1.17E-30 ◦	2.45E-31 △	6.88E-06 ▼	1.69E-28 ▼	3.13E-28 ▼	1.14E-26 ▼	2.32E-29 ▼
Ackley	4.00E-15 ◦	4.00E-15 △	6.99E-04 ▼	1.11E-14 ▼	1.11E-14 ▼	4.00E-15 ▼	7.55E-15 ▼
Schwefel	4.74E+02 ◦	4.74E+02 ◦	6.71E+02 ▼	4.55E-13 △	3.98E-13 △	3.98E-13 △	4.56E-10 △
Rastrigin	0.00 ◦	0.00 ◦	1.77E-05 ▼	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦
Rosenbrock	6.65E+00 ◦	6.59E+00 ◦	6.77E+00 ◦	7.30E+00 ▼	7.67E+00 ▼	7.53E+00 ▼	7.45E+00 ▼
Griewank	0.00 ◦	0.00 ◦	1.05E-02 ▼	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦
	(RA)	$B = 0$ $I = 4$ $W = 2$	$B = 5$ $I = 1$ $W = 0$	$B = 3$ $I = 2$ $W = 1$	$B = 3$ $I = 2$ $W = 1$	$B = 3$ $I = 2$ $W = 1$	$B = 3$ $I = 2$ $W = 1$
R_j	3.1	2.8	6.3	3.9	4.3	3.8	3.8
Pořadí	2	1	7	5	6	3	4
$F_f = 10.36$, p -hodnota = 0.11040, ($\alpha = 0.05$) platí H_0							

Tabulka 5.2: Výsledky experimentu zkoumajícího vliv různých typů **kopulí** – mediány a porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

	Gauss	Student49	Student2	Clayton	Gumbel	Frank	UMDA
f1	3.91E-27 ◦	3.85E-27 ◦	4.78E-02 ▼	2.67E-27 △	2.60E-27 △	2.83E-27 △	2.25E-27 △
f2	2.09E+03 ◦	4.05E+03 ▼	1.45E+06 ▼	3.87E+06 ▼	4.51E+06 ▼	4.28E+06 ▼	1.61E+06 ▼
f3	7.22E+05 ◦	1.95E+06 ◦	6.34E+06 ▼	1.28E+08 ▼	5.50E+06 ▼	1.66E+07 ▼	7.92E+06 ▼
f4	2.36E+02 ◦	2.37E+02 ◦	6.33E+03 ▼	9.19E+03 ▼	6.91E+03 ▼	7.37E+03 ▼	5.80E+03 ▼
f5	1.98E-07 ◦	7.32E-04 ▼	3.17E+00 ▼	2.80E-15 △	1.94E-15 △	2.44E-15 △	2.19E-15 △
f6	1.38E-13 ◦	4.39E-16 ◦	8.10E+00 ▼	9.94E+00 ▼	9.88E+00 ▼	9.87E+00 ▼	9.84E+00 ▼
f7	1.22E-02 ◦	2.42E-02 ▼	3.48E+00 ▼	1.67E+01 ▼	8.08E+00 ▼	1.12E+01 ▼	3.90E+00 ▼
f8	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦
f9	9.84E-01 ◦	9.84E-01 ◦	2.48E+00 ▼	2.58E+00 ▼	2.25E+00 ▼	2.35E+00 ▼	2.15E+00 ▼
f10	0.00 ◦	9.99E-16 ▼	3.93E-01 ▼	7.02E-01 ▼	6.43E-01 ▼	1.08E+00 ▼	2.22E-01 ▼
f11	9.95E-01 ◦	9.95E-01 ◦	8.00E+00 ▼	9.95E-01 ◦	5.33E-15 △	9.82E-10 △	1.33E-13 △
f12	9.95E-01 ◦	1.99E+00 ◦	1.11E+01 ▼	2.99E+00 ▼	2.98E+00 ▼	2.98E+00 ▼	1.99E+00 ▼
f13	1.99E+00 ◦	1.99E+00 ◦	1.57E+01 ▼	4.40E+00 ▼	3.14E+00 ▼	3.87E+00 ▼	1.99E+00 ◦
f14	6.54E+01 ◦	1.70E+01 ◦	2.53E+02 ▼	9.16E+01 ▼	3.68E+02 ▼	3.66E+02 ▼	6.20E+01 ◦
f15	1.35E+02 ◦	1.34E+02 ◦	1.56E+02 ▼	4.74E+02 ▼	3.47E+02 ▼	4.92E+02 ▼	1.48E+02 ▼
f16	1.18E+00 ◦	1.18E+00 ◦	1.19E+00 ◦	1.23E+00 ◦	1.15E+00 ◦	1.18E+00 ◦	1.15E+00 ◦
f17	2.25E+01 ◦	2.13E+01 ◦	1.22E+01 △	1.30E+01 △	1.23E+01 △	1.25E+01 △	2.60E+01 ▼
f18	2.76E+01 ◦	2.73E+01 ◦	2.32E+01 △	1.26E+01 △	1.32E+01 △	1.22E+01 △	2.71E+01 ◦
f19	9.60E-01 ◦	8.86E-01 ◦	9.48E-01 ◦	4.23E-02 △	9.20E-02 △	1.62E-01 △	1.13E+00 ▼
f20	1.87E+00 ◦	1.95E+00 ◦	2.48E+00 ▼	3.49E+00 ▼	3.38E+00 ▼	3.47E+00 ▼	2.47E+00 ▼
f21	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ▼	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦
f22	2.86E+01 ◦	2.91E+01 ◦	1.30E+02 ▼	1.15E+02 ▼	4.37E+02 ▼	3.76E+02 ▼	2.41E+01 △
f23	3.46E+01 ◦	4.99E+01 ▼	3.57E+02 ▼	3.16E+02 ▼	3.80E+02 ▼	3.81E+02 ▼	2.18E+02 ▼
f24	2.00E+02 ◦	2.00E+02 ▼	2.05E+02 ▼	2.02E+02 ▼	2.02E+02 ▼	2.00E+02 ▼	2.00E+02 ◦
f25	2.00E+02 ◦	2.00E+02 ▼	2.03E+02 ▼	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 △
f26	1.00E+02 ◦	1.01E+02 ▼	1.04E+02 ▼	1.06E+02 ▼	1.07E+02 ▼	1.07E+02 ▼	1.02E+02 ▼
f27	3.00E+02 ◦	3.00E+02 ▼	3.20E+02 ▼	3.00E+02 △	3.00E+02 ◦	3.00E+02 △	3.00E+02 △
f28	3.05E+02 ◦	3.18E+02 ▼	3.02E+02 ◦	3.00E+02 △	3.00E+02 △	3.00E+02 △	3.00E+02 △
	(RA)	$B = 10$ $I = 18$ $W = 0$	$B = 22$ $I = 4$ $W = 2$	$B = 17$ $I = 4$ $W = 7$	$B = 17$ $I = 4$ $W = 7$	$B = 17$ $I = 3$ $W = 8$	$B = 14$ $I = 7$ $W = 7$
R_j	3.0	3.1	5.0	4.9	4.1	4.6	3.3
Pořadí	1	2	7	6	4	5	3
$F_f = 26.90$, p -hodnota = 0.00015, ($\alpha = 0.05$) platí H_1							

5.1.3 Vliv počtu jedinců v populaci

Dalším významným parametrem evolučního algoritmu je počet jedinců tvořících populaci. V případě ostrovního modelu tento parametr primárně vyjadřuje velikost populace na ostrově. Velikost globální superpopulace je pak odvozeně dána součinem počtu ostrovů a velikosti populace.

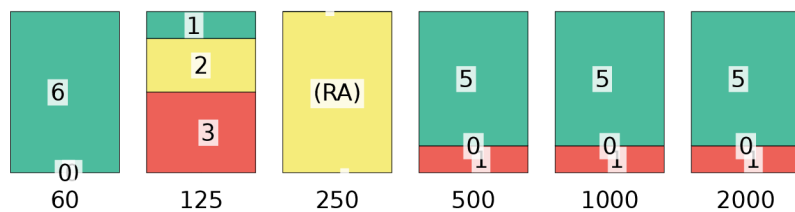
Pro experiment jsem použil toto nastavení:

- velikost populace na každém ostrově: 60, 125, 250, 500, 1000, nebo 2000 jedinců,
- výběr: nejlepších 50 jedinců,
- obnova populace: zachování 3 nejlepší jedinci,
- kopule: Gaussova,
- počet ostrovů: 4,
- topologie: obousměrný prstenec,
- perioda migrací: každých 5 generací,
- počet dimenzí: 10,
- počet evaluací fitness: 100000.

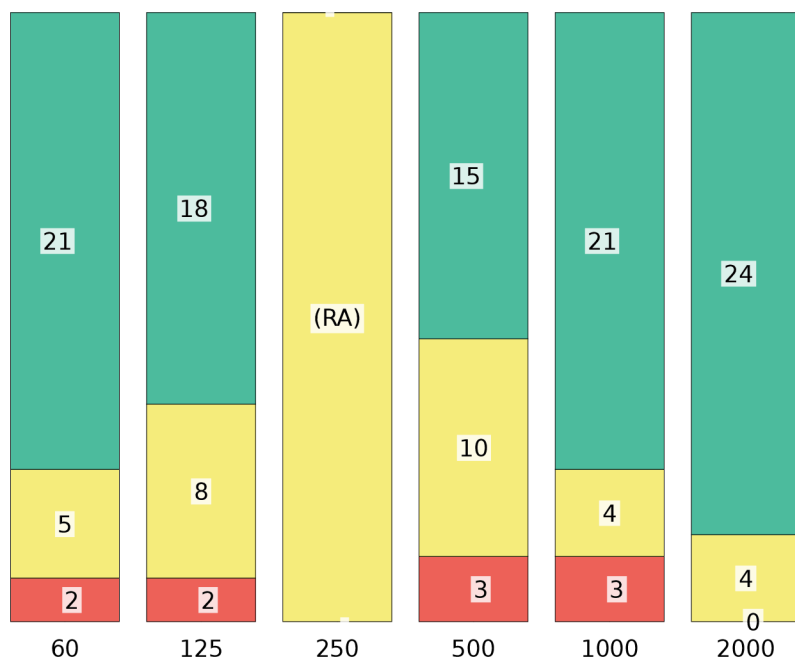
Souhrnné výsledky Wilcoxonových testů jsou na obrázku 5.4 (pro klasické testovací úlohy) a na obrázku 5.5 (pro sadu CEC 2013), v tabulkách 5.3 a 5.4 jsou uvedeny mediány pro každou testovací úlohu a jednotlivé výsledky Wilcoxonových testů.

Nejlepších výsledků dosahuje populace tvořená **250 jedinci** (resp. 125 pro klasické testovací úlohy). Úměrně s tím, jak se hodnota parametru vzdaluje od této hodnoty, se zhoršuje kvalita konvergence dané varianty algoritmu. To potvrzují i výsledky získané Friedmanovým testem.

Při zvětšování populace 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.



Obrázek 5.4: Výsledky experimentu zkoumajícího vliv různých **velikostí populace** – souhrnné porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.



Obrázek 5.5: Výsledky experimentu zkoumajícího vliv různých **velikostí populace** – souhrnné porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

Tabulka 5.3: Výsledky experimentu zkoumajícího vliv různých **velikostí populace** – mediány a porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.

	60	125	250	500	1000	2000
Sphere	2.81E-27 ▼	6.66E-43 △	1.35E-28 ○	5.06E-17 ▼	2.34E-08 ▼	2.21E-03 ▼
Ackley	1.47E-14 ▼	4.44E-16 △	4.00E-15 ○	3.07E-09 ▼	6.01E-05 ▼	2.52E-02 ▼
Schweffel	6.71E+02 ▼	4.74E+02 △	5.92E+02 ○	6.32E+02 ▼	1.24E+03 ▼	1.56E+03 ▼
Rastrigin	9.95E-01 ▼	0.00 ○	0.00 ○	2.23E-09 ▼	4.09E-01 ▼	2.03E+01 ▼
Rosenbrock	7.50E+00 ▼	7.13E+00 ▼	6.69E+00 ○	6.19E+00 △	6.13E+00 △	6.44E+00 △
Griewank	0.00 ▼	0.00 ○	0.00 ○	5.05E-14 ▼	5.58E-04 ▼	4.04E-01 ▼
	$B = 6$ $I = 0$ $W = 0$	$B = 1$ $I = 2$ $W = 3$	(RA)	$B = 5$ $I = 0$ $W = 1$	$B = 5$ $I = 0$ $W = 1$	$B = 5$ $I = 0$ $W = 1$
R_j	3.8	1.9	2.3	3.3	4.2	5.5
Pořadí	4	1	2	3	5	6
$F_f = 14.83$, p -hodnota = 0.01110, ($\alpha = 0.05$) platí H_1						

Tabulka 5.4: Výsledky experimentu zkoumajícího vliv různých **velikostí populace** – mediány a porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

	60	125	250	500	1000	2000
f1	2.86E-02 ▼	5.80E+00 ▼	3.18E-27 ○	1.16E-16 ▼	2.28E-08 ▼	2.49E-03 ▼
f2	9.92E+03 ▼	3.67E+03 ▼	2.09E+03 ○	4.77E+03 ▼	1.90E+04 ▼	6.73E+04 ▼
f3	2.31E+06 ▼	7.11E+06 ▼	7.86E+05 ○	9.31E+04 △	1.78E+05 △	1.05E+07 ▼
f4	7.61E+01 △	6.97E+01 △	2.76E+02 ○	8.04E+02 ▼	1.82E+03 ▼	3.26E+03 ▼
f5	4.30E-02 ▼	2.79E+00 ▼	6.33E-09 ○	2.22E-11 △	6.09E-06 ○	3.44E-02 ▼
f6	2.25E+00 ▼	1.92E-02 ○	7.02E-04 ○	2.04E-09 ○	9.19E-04 ○	1.19E-01 ▼
f7	1.22E+00 ▼	4.70E-02 ▼	7.06E-03 ○	4.06E-02 ▼	5.15E-01 ▼	4.64E+00 ▼
f8	2.04E+01 ○	2.04E+01 ○	2.04E+01 ○	2.04E+01 ○	2.04E+01 △	2.04E+01 ○
f9	1.46E+00 ▼	9.85E-01 ▼	9.84E-01 ○	9.86E-01 ▼	1.50E+00 ▼	2.84E+00 ▼
f10	3.20E-02 ▼	2.40E-02 ▼	0.00 ○	9.41E-11 ▼	5.94E-02 ▼	5.67E-01 ▼
f11	4.97E+00 ▼	2.98E+00 ▼	9.95E-01 ○	7.48E-02 ○	3.18E+00 ▼	2.06E+01 ▼
f12	5.97E+00 ▼	2.98E+00 ▼	9.95E-01 ○	9.95E-01 ○	8.99E+00 ▼	2.12E+01 ▼
f13	1.10E+01 ▼	1.99E+00 ▼	1.99E+00 ○	1.04E+00 ○	5.55E+00 ▼	2.01E+01 ▼
f14	2.77E+02 ▼	2.52E+01 ○	2.78E+01 ○	5.72E+02 ▼	1.05E+03 ▼	1.28E+03 ▼
f15	1.54E+02 ▼	1.36E+02 ○	1.34E+02 ○	6.23E+02 ▼	9.22E+02 ▼	1.11E+03 ▼
f16	1.16E+00 ○	1.18E+00 ○	1.15E+00 ○	1.15E+00 ○	1.11E+00 ○	1.19E+00 ○
f17	2.35E+01 ○	1.96E+01 △	2.25E+01 ○	2.70E+01 ▼	2.88E+01 ▼	3.13E+01 ▼
f18	2.85E+01 ▼	2.70E+01 ○	2.70E+01 ○	2.79E+01 ○	2.90E+01 ▼	3.18E+01 ▼
f19	8.15E-01 ○	8.95E-01 ○	9.03E-01 ○	1.17E+00 ▼	1.59E+00 ▼	1.80E+00 ▼
f20	3.62E+00 ▼	2.36E+00 ▼	1.88E+00 ○	1.90E+00 ○	2.05E+00 ▼	2.44E+00 ▼
f21	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○
f22	6.10E+01 ▼	3.19E+01 ▼	2.40E+01 ○	1.39E+02 ▼	6.93E+02 ▼	1.16E+03 ▼
f23	2.90E+02 ▼	1.10E+02 ▼	4.21E+01 ○	1.06E+02 ▼	5.28E+02 ▼	9.79E+02 ▼
f24	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ○	2.00E+02 ▼	2.00E+02 ▼	2.03E+02 ▼
f25	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ○	2.00E+02 ▼	2.00E+02 ▼	2.02E+02 ▼
f26	1.05E+02 ▼	1.02E+02 ▼	1.00E+02 ○	1.00E+02 ○	1.23E+02 ▼	1.56E+02 ▼
f27	4.00E+02 ▼	3.72E+02 ▼	3.00E+02 ○	3.00E+02 ▼	3.02E+02 ▼	3.13E+02 ▼
f28	3.00E+02 △	3.21E+02 ▼	3.08E+02 ○	3.00E+02 △	3.00E+02 △	3.04E+02 ○
	$B = 21$ $I = 5$ $W = 2$	$B = 18$ $I = 8$ $W = 2$	(RA)	$B = 15$ $I = 10$ $W = 3$	$B = 21$ $I = 4$ $W = 3$	$B = 24$ $I = 4$ $W = 0$
R_j	3.9	3.2	2.0	2.5	3.9	5.5
Pořadí	4	3	1	2	5	6
$F_f = 59.55$, p -hodnota = 1.50E-11, ($\alpha = 0.05$) platí H_1						

5.1.4 Vliv velikosti selekce

Velikost selekce je parametr, který je do určité míry provázán s velikostí populace zkoumanou v předešlé kapitole. Jsou dva možné způsoby, jak na tuto hodnotu můžeme nahlížet: buď jako na poměr (procento) vybrané subpopulace vůči celkové, nebo jako na absolutní počet vybraných jedinců.

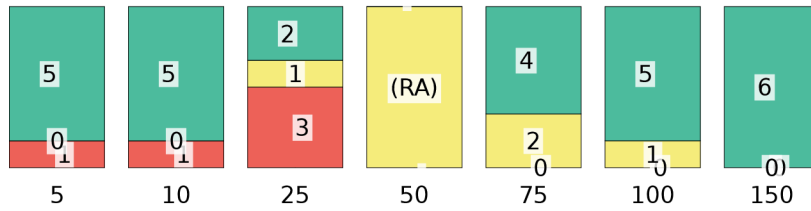
Já v tomto textu používám druhý způsob, lépe postihuje strukturu navrženého algoritmu. Důvodem je nutnost použít dostatečný počet jedinců ve vybrané subpopulaci, aby mohlo dojít k vytvoření kvalitního pravděpodobnostního modelu. Zejména k získání adekvátních korelací (naučení parametrů pro kopuli) je potřebný dostatečně vysoký počet jedinců; pro naučení parametrů marginálních rozdělení počet jedinců až tak kritický není.

Pro experiment jsem použil toto nastavení:

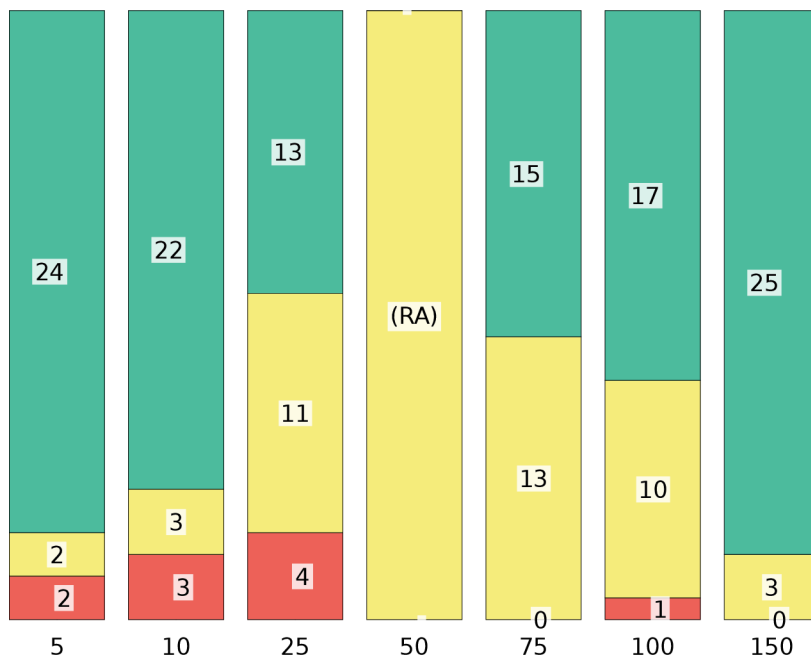
- velikost populace na každém ostrově: 250 jedinců,
- velikost selekce: 5, 10, 25, 50, 75, 100, nebo 150 jedinců,
- obnova populace: zachování 3 nejlepší jedinci,
- kopule: Gaussova,
- počet ostrovů: 4,
- topologie: obousměrný prstenec,
- perioda migrací: každých 5 generací,
- počet dimenzí: 10,
- počet evaluací fitness: 100000.

Souhrnné výsledky Wilcoxonových testů jsou na obrázku 5.6 (pro klasické testovací úlohy) a na obrázku 5.7 (pro sadu CEC 2013), v tabulkách 5.5 a 5.6 jsou uvedeny mediány pro každou testovací úlohu a jednotlivé výsledky Wilcoxonových testů.

Jako nejlepší variantu jsem vyhodnotil tu s velikostí selekce **50 jedinců** (resp. 25 pro klasické testovací úlohy). Tato varianta překonává všechny ostatní testované varianty, přičemž je zřetelné, že se změnou velikosti výběru k mezním hodnotám se výhodnost této varianty stále zvyšuje. To potvrzují i pořadí získaná Friedmanovým testem.



Obrázek 5.6: Výsledky experimentu zkoumajícího vliv různých **velikostí selekce** – souhrnné porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.



Obrázek 5.7: Výsledky experimentu zkoumajícího vliv různých **velikostí selekce** – souhrnné porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

Tabulka 5.5: Výsledky experimentu zkoumajícího vliv různých velikostí selekce – mediány a porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.

	5	10	25	50	75	100	150
Sphere	4.36E+00 ▼	2.45E-01 ▼	1.71E-31 ○	2.09E-30 ○	4.46E-25 ▼	6.63E-19 ▼	1.40E-12 ▼
Ackley	2.03E+00 ▼	5.53E-02 ▼	4.00E-15 △	4.00E-15 ○	8.46E-13 ▼	2.70E-10 ▼	7.96E-07 ▼
Schwefel	3.55E+02 △	3.56E+02 △	3.56E+02 △	4.74E+02 ○	4.74E+02 ○	5.26E+02 ▼	2.02E+03 ▼
Rastrigin	3.00E+00 ▼	2.00E+00 ▼	0.00 ▼	0.00 ○	0.00 ▼	1.42E-11 ▼	1.49E-03 ▼
Rosenbrock	2.13E+01 ▼	7.75E+00 ▼	5.94E+00 △	6.63E+00 ○	6.76E+00 ▼	7.12E+00 ▼	7.48E+00 ▼
Griewank	2.69E-01 ▼	5.67E-02 ▼	0.00 ▼	0.00 ○	0.00 ○	0.00 ○	1.70E-10 ▼
	$B = 5$ $I = 0$ $W = 1$	$B = 5$ $I = 0$ $W = 1$	$B = 2$ $I = 1$ $W = 3$	(RA)	$B = 4$ $I = 2$ $W = 0$	$B = 5$ $I = 1$ $W = 0$	$B = 6$ $I = 0$ $W = 0$
R_j	6.0	5.4	1.8	2.4	3.0	4.1	5.3
Pořadí	7	6	1	2	3	4	5
$F_f = 21.04$, p -hodnota = 0.00181, ($\alpha = 0.05$) platí H_1							

Tabulka 5.6: Výsledky experimentu zkoumajícího vliv různých velikostí selekce – mediány a porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

	5	10	25	50	75	100	150
f1	2.94E+01 ▼	4.28E+00 ▼	4.31E-04 ▼	4.10E-27 ○	5.90E-23 ▼	3.42E-01 ▼	3.01E+01 ▼
f2	1.65E+06 ▼	1.52E+05 ▼	6.99E+03 ▼	3.10E+03 ○	4.78E+03 ▼	1.12E+04 ▼	6.21E+04 ▼
f3	4.37E+08 ▼	4.74E+07 ▼	3.52E+05 ○	6.42E+05 ○	4.30E+06 ▼	6.98E+06 ▼	1.22E+08 ▼
f4	5.41E+03 ▼	2.71E+03 ▼	1.86E+02 ○	2.09E+02 ○	5.27E+02 ▼	1.04E+03 ▼	1.46E+03 ▼
f5	5.93E+01 ▼	7.98E+00 ▼	3.32E-01 ▼	2.58E-07 ○	1.81E-13 ○	2.98E-09 ○	1.30E+00 ▼
f6	1.80E+01 ▼	1.03E+00 ▼	1.06E-14 ○	2.18E-03 ○	5.05E-12 ○	4.22E-10 ○	1.41E-01 ▼
f7	1.74E+01 ▼	3.31E+00 ▼	4.18E-02 ▼	1.58E-02 ○	2.42E-02 ○	8.19E-02 ▼	1.34E+00 ▼
f8	2.03E+01 ○	2.04E+01 ○	2.03E+01 ○	2.04E+01 ○	2.03E+01 ○	2.04E+01 ○	2.04E+01 ○
f9	3.71E+00 ▼	1.91E+00 ▼	9.84E-01 ○	2.93E-01 ○	9.84E-01 ▼	9.86E-01 ▼	1.10E+00 ▼
f10	8.36E+00 ▼	4.37E-01 ▼	7.45E-03 ▼	0.00 ○	0.00 ○	4.24E-12 ▼	5.26E-02 ▼
f11	6.43E+00 ▼	3.99E+00 ▼	1.99E+00 ▼	9.95E-01 ○	9.95E-01 ○	9.95E-01 ○	3.41E+00 ▼
f12	1.20E+01 ▼	6.02E+00 ▼	2.98E+00 ▼	9.95E-01 ○	9.95E-01 ○	9.95E-01 ○	1.09E+01 ▼
f13	1.67E+01 ▼	9.34E+00 ▼	3.15E+00 ▼	1.99E+00 ○	9.97E-01 ○	1.92E+00 ○	3.86E+00 ▼
f14	2.56E+02 ▼	1.41E+02 ▼	2.52E+01 △	1.27E+02 ○	2.32E+02 ▼	5.39E+02 ▼	1.05E+03 ▼
f15	2.83E+02 ▼	1.49E+02 ▼	1.34E+02 ○	1.37E+02 ○	2.23E+02 ▼	6.19E+02 ▼	9.23E+02 ▼
f16	1.16E+00 ○	1.15E+00 ○	1.20E+00 ○	1.13E+00 ○	1.21E+00 ▼	1.09E+00 ○	1.18E+00 ○
f17	1.69E+01 △	1.42E+01 △	1.87E+01 △	2.34E+01 ○	2.44E+01 ○	2.52E+01 ▼	2.78E+01 ▼
f18	1.92E+01 △	2.64E+01 ○	2.64E+01 ○	2.54E+01 ○	2.70E+01 ○	2.68E+01 ○	2.80E+01 ▼
f19	1.10E+00 ▼	7.05E-01 △	7.97E-01 △	9.14E-01 ○	9.98E-01 ▼	1.12E+00 ▼	1.37E+00 ▼
f20	2.49E+00 ▼	1.77E+00 △	1.66E+00 △	2.01E+00 ○	2.33E+00 ▼	2.69E+00 ▼	3.89E+00 ▼
f21	4.01E+02 ▼	4.00E+02 ▼	4.00E+02 ▼	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○
f22	2.64E+02 ▼	1.57E+02 ▼	4.55E+01 ▼	2.52E+01 ○	3.19E+01 ▼	1.35E+02 ▼	7.78E+02 ▼
f23	4.70E+02 ▼	2.21E+02 ▼	5.07E+01 ▼	3.96E+01 ○	4.86E+01 ▼	2.47E+02 ▼	8.72E+02 ▼
f24	2.12E+02 ▼	2.02E+02 ▼	2.00E+02 ○	2.00E+02 ○	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ▼
f25	2.06E+02 ▼	2.01E+02 ▼	2.00E+02 ○	2.00E+02 ○	2.00E+02 ▼	2.00E+02 ▼	2.01E+02 ▼
f26	1.13E+02 ▼	1.04E+02 ▼	1.02E+02 ▼	1.01E+02 ○	1.00E+02 ○	1.00E+02 △	1.11E+02 ▼
f27	3.85E+02 ▼	3.12E+02 ▼	3.00E+02 ○	3.00E+02 ○	3.00E+02 ▼	3.00E+02 ▼	4.00E+02 ▼
f28	4.48E+02 ▼	3.35E+02 ▼	3.14E+02 ▼	3.04E+02 ○	3.04E+02 ○	3.06E+02 ○	3.22E+02 ▼
	$B = 24$ $I = 2$ $W = 2$	$B = 22$ $I = 3$ $W = 3$	$B = 13$ $I = 11$ $W = 4$	(RA)	$B = 15$ $I = 13$ $W = 0$	$B = 17$ $I = 10$ $W = 1$	$B = 25$ $I = 3$ $W = 0$
R_j	5.9	4.7	2.8	2.3	2.8	3.7	5.7
Pořadí	7	5	3	1	2	4	6
$F_f = 76.27$, p -hodnota = 2.10E-14, ($\alpha = 0.05$) platí H_1							

5.1.5 Vliv obnovy populace

Hodnotu obnovy populace lze uvádět několika různými způsoby. Já v tomto textu (obdobně jako u selekce v předešlé kapitole) používám počet nejlepších jedinců, kteří jsou převzati do další generace.

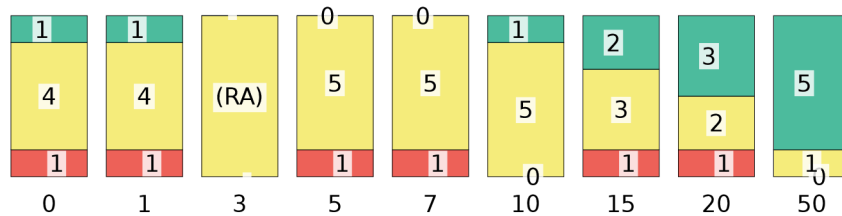
Pro experiment jsem použil toto nastavení:

- velikost populace: 250 jedinců na každém ostrově,
- selekce: nejlepších 50 jedinců,
- obnova populace: zachováno 0, 1, 3, 5, 7, 10, 15, 20, nebo 50 nejlepších jedinců,
- kopule: Gaussova,
- počet ostrovů: 4,
- topologie: obousměrný prstenec,
- perioda migrací: každých 5 generací,
- počet dimenzí: 10,
- počet evaluací fitness: 100000.

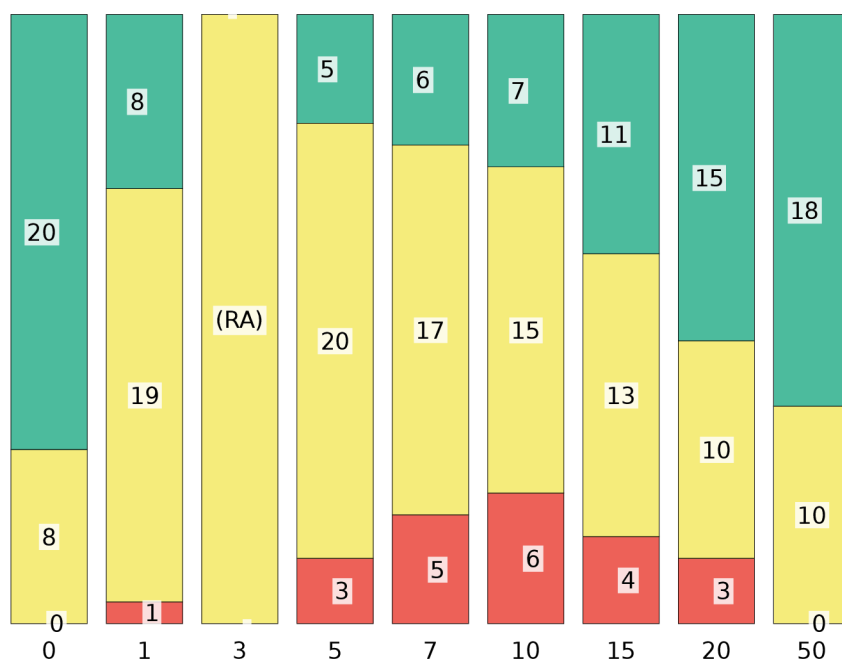
Souhrnné výsledky Wilcoxonových testů jsou na obrázku 5.8 (pro klasické testovací úlohy) a na obrázku 5.9 (pro sadu CEC 2013), v tabulkách 5.7 a 5.8 jsou uvedeny mediány pro každou testovací úlohu a jednotlivé výsledky Wilcoxonových testů.

Z výsledků je zřejmé, že nejlepších výsledků dosáhly varianty, které zachovávají jen **malý počet jedinců, 3–7**. Při větším počtu zachovaných jedinců kvalita konvergence výrazně klesá. Horšího výsledku dosáhla i varianta, která zachovávala pouze jednoho (jen toho nejlepšího) jedince, výrazně horší výsledek podala varianta, která obnovila úplně celou populaci (tzv. koncepce vymírání rodičovské populace).

Podle hodnot B , W , I je nejlepší obnova se zachováním 3 jedinců, Friedmanův test preferuje sousední hodnoty obnovy, 5 nebo 7. Je vidět, že hodnoty B , W ve srovnání s RA jsou přítom na nízkých podobných hodnotách.



Obrázek 5.8: Výsledky experimentu zkoumajícího vliv různých **obnov populace** – souhrnné porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.



Obrázek 5.9: Výsledky experimentu zkoumajícího vliv různých **obnov populace** – souhrnné porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

Tabulka 5.7: Výsledky experimentu zkoumajícího vliv různých **obnov populace** – mediány a porovnáni variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.

	0	1	3	5	7	10	15	20	50
Sph.	7.31E-30 Δ	2.41E-29 Δ	1.71E-28 \circ	7.63E-30 Δ	3.84E-30 Δ	2.36E-28 \circ	2.38E-30 Δ	2.16E-30 Δ	1.59E-26 \blacktriangledown
Ack.	4.00E-15 \blacktriangledown	4.00E-15 \blacktriangledown	4.00E-15 \circ	4.00E-15 \circ	4.00E-15 \circ	4.00E-15 \circ	4.00E-15 \circ	4.00E-15 \blacktriangledown	4.00E-15 \blacktriangledown
Sch.	4.74E+02 \circ	4.74E+02 \circ	4.74E+02 \circ	4.74E+02 \circ	4.74E+02 \circ	4.74E+02 \circ	4.74E+02 \circ	6.71E+02 \blacktriangledown	1.06E+03 \blacktriangledown
Ras.	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \blacktriangledown	1.88E-13 \blacktriangledown	3.14E-03 \blacktriangledown	1.99E+01 \blacktriangledown
Ros.	6.70E+00 \circ	6.67E+00 \circ	6.66E+00 \circ	6.56E+00 \circ	6.59E+00 \circ	6.51E+00 \circ	6.71E+00 \circ	6.45E+00 \circ	6.68E+00 \circ
Gri.	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	3.33E-16 \blacktriangledown
	$B = 1$ $I = 4$ $W = 1$	$B = 1$ $I = 4$ $W = 1$	(RA)	$B = 0$ $I = 5$ $W = 1$	$B = 0$ $I = 5$ $W = 1$	$B = 1$ $I = 5$ $W = 0$	$B = 2$ $I = 3$ $W = 1$	$B = 3$ $I = 2$ $W = 1$	$B = 5$ $I = 1$ $W = 0$
R_j	4.8	4.8	4.8	4.1	3.9	4.4	5.8	4.6	8.0
Poř.	5	5	5	2	1	3	8	4	9
$F_f = 9.82$, p -hodnota = 0.27773, $(\alpha = 0.05)$ platí H_0									

Tabulka 5.8: Výsledky experimentu zkoumajícího vliv různých **obnov populace** – mediány a porovnáni variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

	0	1	3	5	7	10	15	20	50
f1	2.91E-02 \blacktriangledown	3.65E-27 \circ	3.81E-27 \circ	3.22E-27 \circ	2.73E-27 \circ	1.92E-27 Δ	3.86E-27 \circ	3.70E-27 \circ	2.64E-07 \blacktriangledown
f2	1.21E+04 \blacktriangledown	3.06E+03 \circ	3.01E+03 \circ	2.44E+03 \circ	2.01E+03 Δ	2.55E+03 \circ	3.18E+03 \circ	3.39E+03 \circ	1.35E+04 \blacktriangledown
f3	8.88E+06 \blacktriangledown	1.19E+06 \circ	8.56E+05 \circ	4.54E+05 \circ	1.67E+05 Δ	6.93E+05 \circ	3.48E+05 \circ	5.34E+05 \circ	1.43E+06 \circ
f4	1.02E+04 \blacktriangledown	1.10E+03 \blacktriangledown	2.41E+02 \circ	2.27E+02 \circ	2.13E+02 \circ	2.73E+02 \circ	7.04E+02 \blacktriangledown	1.12E+03 \blacktriangledown	2.66E+03 \blacktriangledown
f5	8.26E-02 \blacktriangledown	1.23E-03 \circ	1.11E-05 \circ	9.49E-11 Δ	1.37E-05 \circ	5.45E-08 Δ	4.11E-02 \circ	2.51E-05 \circ	1.01E-01 \blacktriangledown
f6	1.60E-14 \circ	4.86E-14 \circ	4.15E-15 \circ	1.89E-15 \circ	1.44E-16 \circ	9.56E-10 \circ	4.85E-17 Δ	2.90E-05 \circ	3.26E-05 \circ
f7	1.34E-02 \blacktriangledown	6.45E-03 \circ	8.45E-03 \circ	8.11E-03 \circ	7.39E-03 \circ	2.74E-03 Δ	3.12E-03 Δ	3.34E-03 Δ	5.45E-03 \circ
f8	2.08E+01 \blacktriangledown	2.04E+01 \circ	2.04E+01 \circ	2.04E+01 \circ	2.04E+01 \circ	2.04E+01 \circ	2.04E+01 \circ	2.03E+01 \circ	2.04E+01 \circ
f9	9.84E-01 \blacktriangledown	9.84E-01 \blacktriangledown	9.84E-01 \circ	9.84E-01 \circ	9.84E-01 \blacktriangledown	9.84E-01 \circ	2.32E-01 \circ	9.84E-01 \blacktriangledown	3.40E-01 \circ
f10	0.00 \blacktriangledown	0.00 \blacktriangledown	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	0.00 \circ	1.00E-11 \blacktriangledown	3.38E-01 \blacktriangledown
f11	9.95E-01 \circ	9.95E-01 \circ	9.95E-01 \circ	9.95E-01 \circ	9.95E-01 \circ	9.95E-01 \circ	1.01E+00 \blacktriangledown	1.99E+00 \blacktriangledown	2.21E+01 \blacktriangledown
f12	1.99E+00 \blacktriangledown	1.99E+00 \blacktriangledown	9.95E-01 \circ	9.95E-01 \circ	9.95E-01 \circ	9.96E-01 \blacktriangledown	1.99E+00 \blacktriangledown	3.36E+00 \blacktriangledown	2.27E+01 \blacktriangledown
f13	1.99E+00 \circ	1.99E+00 \circ	1.99E+00 \circ	1.99E+00 \circ	1.99E+00 \circ	1.99E+00 \circ	1.99E+00 \circ	3.14E+00 \blacktriangledown	2.08E+01 \blacktriangledown
f14	1.28E+01 \circ	1.63E+01 \circ	1.55E+01 \circ	1.23E+02 \blacktriangledown	2.73E+02 \blacktriangledown	7.53E+02 \blacktriangledown	1.08E+03 \blacktriangledown	1.23E+03 \blacktriangledown	1.36E+03 \blacktriangledown
f15	1.33E+02 \circ	1.30E+02 \circ	1.35E+02 \circ	1.52E+02 \blacktriangledown	2.77E+02 \blacktriangledown	6.81E+02 \blacktriangledown	9.33E+02 \blacktriangledown	1.04E+03 \blacktriangledown	1.25E+03 \blacktriangledown
f16	2.60E+00 \blacktriangledown	1.12E+00 \circ	1.14E+00 \circ	1.13E+00 \circ	1.16E+00 \circ	1.18E+00 \circ	1.16E+00 \circ	1.16E+00 \circ	1.21E+00 \circ
f17	2.71E+01 \blacktriangledown	1.99E+01 Δ	2.29E+01 \circ	2.58E+01 \blacktriangledown	2.70E+01 \blacktriangledown	2.69E+01 \blacktriangledown	2.88E+01 \blacktriangledown	2.91E+01 \blacktriangledown	3.02E+01 \blacktriangledown
f18	4.56E+01 \blacktriangledown	2.69E+01 \circ	2.69E+01 \circ	2.61E+01 \circ	2.72E+01 \circ	2.77E+01 \blacktriangledown	2.89E+01 \blacktriangledown	2.84E+01 \blacktriangledown	2.96E+01 \blacktriangledown
f19	9.56E-01 \blacktriangledown	9.14E-01 \circ	8.05E-01 \circ	9.52E-01 \blacktriangledown	1.06E+00 \blacktriangledown	1.19E+00 \blacktriangledown	1.36E+00 \blacktriangledown	1.46E+00 \blacktriangledown	1.78E+00 \blacktriangledown
f20	2.69E+00 \blacktriangledown	2.16E+00 \blacktriangledown	2.02E+00 \circ	1.84E+00 Δ	1.89E+00 Δ	1.88E+00 Δ	1.97E+00 \circ	1.94E+00 \circ	2.07E+00 \circ
f21	4.00E+02 \circ	4.00E+02 \circ	4.00E+02 \circ	4.00E+02 \circ	4.00E+02 \circ	4.00E+02 \circ	4.00E+02 \circ	4.00E+02 \circ	4.00E+02 \circ
f22	2.41E+01 \circ	2.60E+01 \circ	2.51E+01 \circ	2.80E+01 \circ	2.91E+01 \circ	4.83E+01 \blacktriangledown	3.66E+02 \blacktriangledown	5.71E+02 \blacktriangledown	1.36E+03 \blacktriangledown
f23	4.51E+01 \blacktriangledown	5.80E+01 \blacktriangledown	3.46E+01 \circ	3.35E+01 \circ	3.26E+01 \circ	4.55E+01 \circ	8.88E+01 \blacktriangledown	3.17E+02 \blacktriangledown	1.03E+03 \blacktriangledown
f24	2.00E+02 \blacktriangledown	2.00E+02 \circ	2.00E+02 \circ	2.00E+02 \circ	2.00E+02 \circ	2.00E+02 Δ	2.00E+02 \circ	2.00E+02 Δ	2.00E+02 \circ
f25	2.00E+02 \blacktriangledown	2.00E+02 \blacktriangledown	2.00E+02 \circ	2.00E+02 Δ	2.00E+02 Δ	2.00E+02 Δ	2.00E+02 Δ	2.00E+02 Δ	2.00E+02 \circ
f26	1.01E+02 \circ	1.00E+02 \circ	1.01E+02 \circ	1.00E+02 \circ	1.00E+02 Δ	1.01E+02 \circ	1.01E+02 \blacktriangledown	1.12E+02 \blacktriangledown	1.21E+02 \blacktriangledown
f27	3.00E+02 \blacktriangledown	3.00E+02 \blacktriangledown	3.00E+02 \circ	3.00E+02 \blacktriangledown	3.00E+02 \blacktriangledown	3.00E+02 \blacktriangledown	3.00E+02 \circ	3.00E+02 \blacktriangledown	3.00E+02 \blacktriangledown
f28	3.24E+02 \blacktriangledown	3.12E+02 \circ	3.09E+02 \circ	3.11E+02 \circ	3.04E+02 Δ	3.10E+02 \circ	3.02E+02 Δ	3.06E+02 \circ	3.19E+02 \blacktriangledown
	$B = 20$ $I = 8$ $W = 0$	$B = 8$ $I = 19$ $W = 1$	(RA)	$B = 5$ $I = 20$ $W = 3$	$B = 6$ $I = 17$ $W = 5$	$B = 7$ $I = 15$ $W = 6$	$B = 11$ $I = 13$ $W = 4$	$B = 15$ $I = 10$ $W = 3$	$B = 18$ $I = 10$ $W = 0$
R_j	6.0	4.4	4.0	3.5	3.7	4.5	5.1	6.1	7.6
Poř.	7	4	3	1	2	5	6	8	9
$F_f = 54.17$, p -hodnota = 6.41E-09, $(\alpha = 0.05)$ platí H_1									

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).

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.

Pro experiment jsem použil toto nastavení:

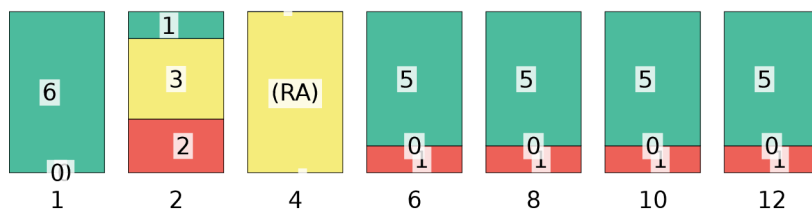
- velikost populace: 250 jedinců na každém ostrově,
- výběr: nejlepších 50 jedinců,
- obnova populace: zachování 3 nejlepší jedinci,
- kopule: Gaussova,
- počet ostrovů: 12, 8, 6, 4, 2, nebo 1,
- topologie: obousměrný prstenec,
- perioda migrací: každých 5 generací,
- počet dimenzí: 10,
- počet evaluací fitness: 100000.

Souhrnné výsledky Wilcoxonových testů jsou na obrázku 5.10 (pro klasické testovací úlohy) a na obrázku 5.11 (pro sadu CEC 2013), v tabulkách 5.9 a 5.10 jsou uvedeny mediány pro každou testovací úlohu a jednotlivé výsledky Wilcoxonových testů.

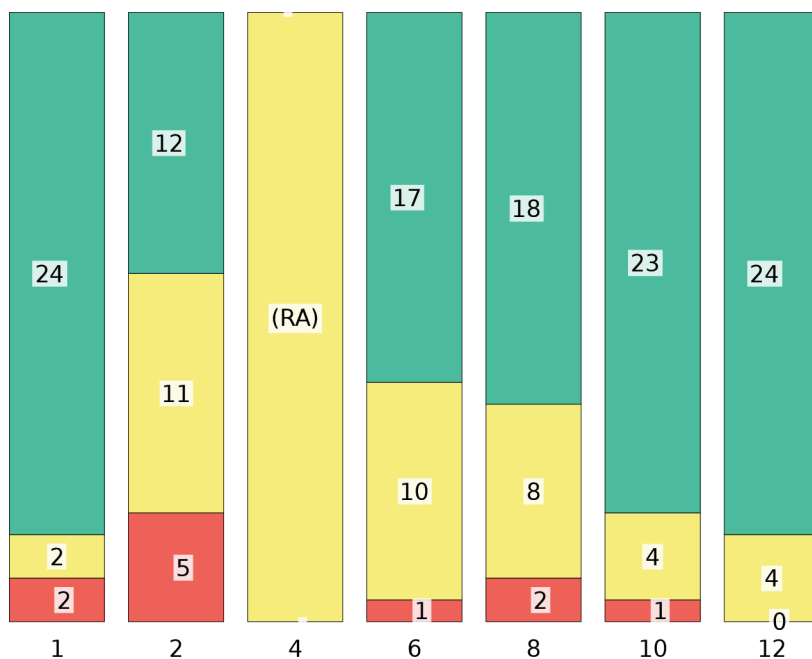
Nejlepších výsledků dosahují varianty s malým počtem ostrovů (vyjma případu s jedním ostrovem), **4 ostrovy** pro sadu CEC 2013 a pouhé 2 ostrovy stačí pro klasické testovací úlohy. Varianty s větším počtem ostrovů jsou vůči referenční variantě vždy výrazně horší.

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 na všech šesti uvažovaných klasických testovacích úlohách ($B = 6, I = 0, W = 0$) a na naprosté většině úloh ze sady CEC 2013 (na 24 úlohách z celkových 28 ($B = 24, I = 2, W = 2$)).



Obrázek 5.10: Výsledky experimentu zkoumajícího vliv různých **počtů ostrovů** – souhrnné porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.



Obrázek 5.11: Výsledky experimentu zkoumajícího vliv různých **počtů ostrovů** – souhrnné porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

Tabulka 5.9: Výsledky experimentu zkoumajícího vliv různých **počtů ostrovů** – mediány a porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.

	1	2	4	6	8	10	12
Sphere	6.91E-07 ▼	1.23E-29 ○	3.97E-31 ○	2.96E-18 ▼	2.14E-15 ▼	1.52E-11 ▼	5.48E-10 ▼
Ackley	1.39E-05 ▼	4.44E-16 △	4.00E-15 ○	5.50E-10 ▼	1.20E-07 ▼	1.31E-05 ▼	2.24E-04 ▼
Schwefel	5.95E+02 ▼	4.74E+02 ▼	4.74E+02 ○	5.57E+02 ▼	7.30E+02 ▼	1.04E+03 ▼	1.32E+03 ▼
Rastrigin	9.97E-01 ▼	0.00 ○	0.00 ○	6.81E-12 ▼	1.61E-06 ▼	3.31E-02 ▼	2.42E+00 ▼
Rosenbrock	7.59E+00 ▼	5.99E+00 △	6.57E+00 ○	6.44E+00 △	5.94E+00 △	6.26E+00 △	6.31E+00 △
Griewank	1.97E-07 ▼	0.00 ○	0.00 ○	0.00 ▼	1.02E-11 ▼	1.92E-07 ▼	6.44E-05 ▼
	$B = 6$ $I = 0$ $W = 0$	$B = 1$ $I = 3$ $W = 2$	(RA)	$B = 5$ $I = 0$ $W = 1$	$B = 5$ $I = 0$ $W = 1$	$B = 5$ $I = 0$ $W = 1$	$B = 5$ $I = 0$ $W = 1$
R_j	6.0	1.7	2.3	3.2	3.7	4.8	6.3
Pořadí	6	1	2	3	4	5	7
$F_f = 24.64$, p -hodnota = 0.00040, $(\alpha = 0.05)$ platí H_1							

Tabulka 5.10: Výsledky experimentu zkoumajícího vliv různých **počtů ostrovů** – mediány a porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

	1	2	4	6	8	10	12
f1	1.61E+01 ▼	1.94E-27 △	4.97E-27 ○	1.42E-18 ▼	8.25E-13 ▼	1.14E-08 ▼	1.35E-06 ▼
f2	1.11E+05 ▼	3.29E+03 ○	3.04E+03 ○	5.50E+03 ▼	1.35E+04 ▼	2.68E+04 ▼	5.31E+04 ▼
f3	5.49E+08 ▼	1.71E+06 ○	5.31E+05 ○	1.65E+05 △	2.06E+05 △	1.96E+06 ▼	6.45E+06 ▼
f4	7.68E+02 ▼	7.28E+01 △	2.81E+02 ○	7.84E+02 ▼	1.28E+03 ▼	2.00E+03 ▼	2.40E+03 ▼
f5	2.94E+01 ▼	1.91E-01 ▼	1.21E-04 ○	2.55E-10 ○	1.24E-07 ○	9.62E-06 ○	3.07E-04 ○
f6	1.62E+00 ▼	3.07E-22 △	2.68E-16 ○	1.69E-09 ▼	3.61E-07 ▼	7.45E-05 ▼	1.57E-03 ▼
f7	5.06E+00 ▼	1.34E-01 ▼	1.35E-02 ○	4.70E-02 ▼	2.13E-01 ▼	5.32E-01 ▼	1.64E+00 ▼
f8	2.03E+01 ○	2.03E+01 ○	2.03E+01 ○	2.03E+01 ○	2.03E+01 ○	2.03E+01 ○	2.04E+01 ○
f9	2.00E+00 ▼	9.84E-01 ○	9.84E-01 ○	9.57E-01 ▼	6.47E-01 ○	1.08E+00 ▼	1.45E+00 ▼
f10	2.05E+00 ▼	0.00 ○	0.00 ○	7.74E-12 ▼	1.14E-07 ▼	8.38E-04 ▼	3.69E-02 ▼
f11	2.99E+00 ▼	1.99E+00 ▼	9.95E-01 ○	9.95E-01 ○	9.95E-01 ○	2.50E+00 ▼	1.29E+01 ▼
f12	3.98E+00 ▼	1.99E+00 ▼	9.95E-01 ○	9.95E-01 ○	1.03E+00 ○	3.38E+00 ▼	1.65E+01 ▼
f13	4.27E+00 ▼	2.43E+00 ▼	1.99E+00 ○	1.99E+00 ○	1.99E+00 ○	2.56E+00 ▼	1.13E+01 ▼
f14	1.50E+02 ▼	2.86E+01 ○	5.24E+01 ○	5.74E+02 ▼	7.61E+02 ▼	1.08E+03 ▼	1.16E+03 ▼
f15	1.48E+02 ▼	1.38E+02 ○	1.37E+02 ○	4.88E+02 ▼	7.84E+02 ▼	8.79E+02 ▼	1.04E+03 ▼
f16	1.17E+00 ○	1.18E+00 ○	1.15E+00 ○	1.16E+00 ○	1.21E+00 ○	1.14E+00 ○	1.21E+00 ▼
f17	1.13E+01 △	1.13E+01 △	2.34E+01 ○	2.76E+01 ▼	2.71E+01 ▼	2.79E+01 ▼	2.95E+01 ▼
f18	1.19E+01 △	2.49E+01 △	2.69E+01 ○	2.69E+01 ○	2.85E+01 ▼	2.80E+01 ▼	2.97E+01 ▼
f19	9.50E-01 ▼	8.19E-01 ○	9.13E-01 ○	1.19E+00 ▼	1.23E+00 ▼	1.48E+00 ▼	1.67E+00 ▼
f20	2.32E+00 ▼	1.81E+00 ○	1.89E+00 ○	2.13E+00 ▼	2.25E+00 ▼	2.48E+00 ▼	2.58E+00 ▼
f21	4.00E+02 ▼	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○	4.00E+02 ○
f22	1.20E+02 ▼	4.83E+01 ▼	3.05E+01 ○	1.49E+02 ▼	3.67E+02 ▼	6.56E+02 ▼	9.02E+02 ▼
f23	2.51E+02 ▼	7.01E+01 ▼	5.00E+01 ○	9.43E+01 ▼	2.47E+02 ▼	5.36E+02 ▼	7.67E+02 ▼
f24	2.03E+02 ▼	2.00E+02 ▼	2.00E+02 ○	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ▼	2.01E+02 ▼
f25	2.03E+02 ▼	2.00E+02 ▼	2.00E+02 ○	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ▼	2.01E+02 ▼
f26	1.03E+02 ▼	1.01E+02 ▼	1.00E+02 ○	1.00E+02 ○	1.00E+02 ▼	1.02E+02 ▼	1.17E+02 ▼
f27	3.17E+02 ▼	3.00E+02 ▼	3.00E+02 ○	3.00E+02 ▼	3.00E+02 ▼	3.01E+02 ▼	3.03E+02 ▼
f28	4.26E+02 ▼	3.13E+02 ▼	3.04E+02 ○	3.01E+02 ○	3.00E+02 △	3.00E+02 △	3.01E+02 ○
	$B = 24$ $I = 2$ $W = 2$	$B = 12$ $I = 11$ $W = 5$	(RA)	$B = 17$ $I = 10$ $W = 1$	$B = 18$ $I = 8$ $W = 2$	$B = 23$ $I = 4$ $W = 1$	$B = 24$ $I = 4$ $W = 0$
R_j	5.3	2.8	2.4	2.9	3.6	4.7	6.3
Pořadí	6	2	1	3	4	5	7
$F_f = 77.14$, p -hodnota = 1.39E-14, $(\alpha = 0.05)$ platí H_1							

5.1.7 Vliv periody migrací

Perioda migrací je dalším z parametrů, který může významně ovlivnit kvalitu migrujícího algoritmu. Při jeho volbě se vyskytují dvě mezní hodnoty – uskutečnit migraci každou generaci a neumožnit migraci vůbec.

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

Opačným extrémem při nastavení tohoto parametru je tak dlouhá perioda, že migrace je zcela potlačena. Pokud by tato varianta byla stejně efektivní jako ty, kde k migraci dochází, znamenalo by to, že efektivita algoritmu s migrací nespočívá v migraci, ale pouze v jiném faktoru, nejspíše v použití většího počtu ostrovních populací.

Pro experimenty jsem použil toto nastavení:

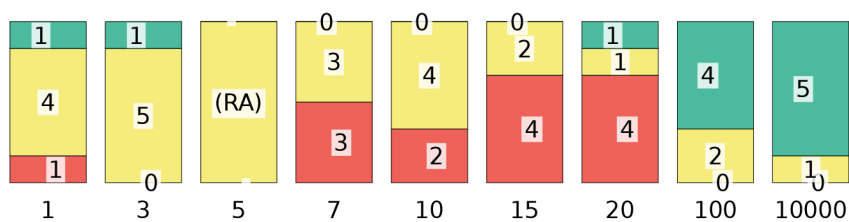
- velikost populace na každém ostrově: 250,
- výběr: nejlepších 50 jedinců,
- obnova populace: zachování 3 nejlepší jedinci,
- kopule: Gaussova,
- počet ostrovů: 4,
- perioda migrací: 1, 3, 5, 7, 10, 15, 20, 100, nebo 10000 generací,
- topologie: obousměrný prstenec,
- počet dimenzí: 10,
- počet evaluací fitness: 100000.

Souhrnné výsledky Wilcoxonových testů jsou na obrázku 5.12 (pro klasické testovací úlohy) a na obrázku 5.13 (pro sadu CEC 2013), v tabulkách 5.11 a 5.12 jsou uvedeny mediány pro každou testovací úlohu a jednotlivé výsledky Wilcoxonových testů.

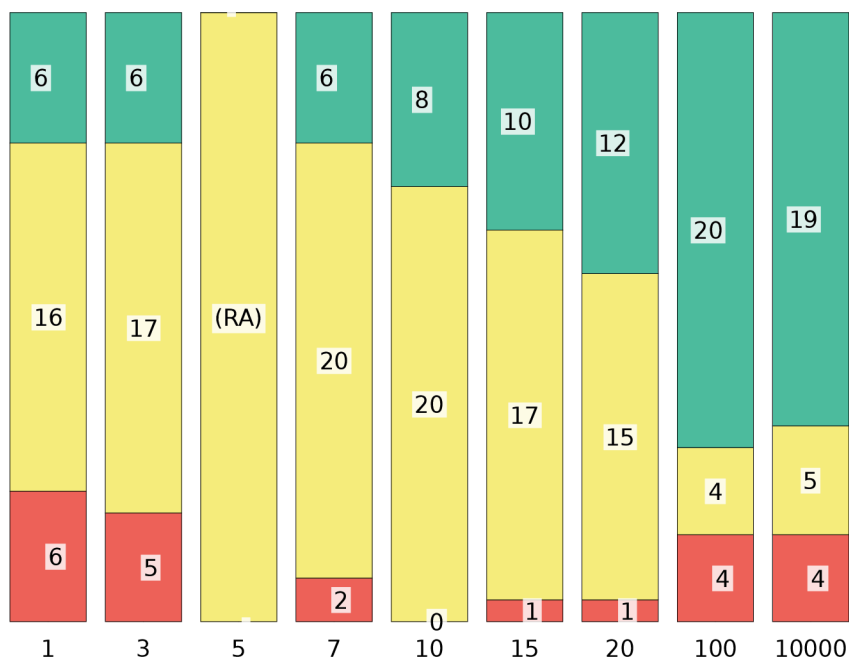
V případě krátkých period mezi migracemi nebyl mezi jednotlivými variantami významný rozdíl (vysoká hodnota I , hodnoty B a W jsou jen malé a vzájemně podobné). S prodlužující se hodnotou periody se začíná ukazovat výhodnost referenční varianty (zvětšuje se B) – tj. **5 generací**. Vůbec nejhorší výsledek byl zaznamenán pro nejdelší periody.

Na sadě klasických testovacích úloh by nejlépe vyhovující velikost periody byla delší, cca 15 generací; ale i zde se ukazuje výhodnost referenční varianty vůči příliš dlouhým periodám.

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



Obrázek 5.12: Výsledky experimentu zkoumajícího vliv různých **period migrací** – souhrnné porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.



Obrázek 5.13: Výsledky experimentu zkoumajícího vliv různých **period migrací** – souhrnné porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

Tabulka 5.11: Výsledky experimentu zkoumajícího vliv různých **period migrací** – mediány a porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.

	1	3	5	7	10	15	20	100	10000
Sph.	9.30E-29 ◦	1.97E-28 ◦	1.99E-28 ◦	3.20E-31 Δ	1.39E-31 Δ	1.69E-30 Δ	4.61E-32 Δ	5.82E-21 ▼	3.60E-21 ▼
Ack.	4.00E-15 ▼	7.55E-15 ▼	4.00E-15 ◦	4.00E-15 Δ	4.00E-15 Δ	4.00E-15 Δ	4.00E-15 Δ	5.30E-10 ▼	6.17E-09 ▼
Sch.	4.74E+02 ◦	4.74E+02 ◦	4.74E+02 ◦	4.74E+02 ◦	4.74E+02 ◦	3.67E+02 Δ	3.55E+02 Δ	4.78E+02 ◦	4.74E+02 ◦
Ras.	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ▼	4.58E-04 ▼	1.72E-04 ▼
Ros.	6.35E+00 Δ	6.57E+00 ◦	6.72E+00 ◦	6.38E+00 Δ	6.59E+00 ◦	6.24E+00 Δ	6.11E+00 Δ	6.71E+00 ◦	7.03E+00 ▼
Gri.	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦	1.03E-14 ▼	1.06E-12 ▼
	$B = 1$ $I = 4$ $W = 1$	$B = 1$ $I = 5$ $W = 0$	(RA)	$B = 0$ $I = 3$ $W = 3$	$B = 0$ $I = 4$ $W = 2$	$B = 0$ $I = 2$ $W = 4$	$B = 1$ $I = 1$ $W = 4$	$B = 4$ $I = 2$ $W = 0$	$B = 5$ $I = 1$ $W = 0$
R_j	4.2	5.3	5.3	4.0	4.2	3.3	2.4	8.3	8.1
Poř.	4	6	7	3	4	2	1	9	8
$F_f = 26.33$, p -hodnota = 0.00092, $(\alpha = 0.05)$ platí H_1									

Tabulka 5.12: Výsledky experimentu zkoumajícího vliv různých **period migrací** – mediány a porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

	1	3	5	7	10	15	20	100	10000
f1	4.85E-27 ◦	4.82E-27 ▼	2.92E-27 ◦	4.22E-27 ▼	4.85E-05 ▼	3.23E-06 ▼	1.54E-03 ▼	8.60E-02 ▼	4.53E-02 ▼
f2	2.60E+03 ◦	2.11E+03 ◦	2.51E+03 ◦	2.16E+03 ◦	2.76E+03 ◦	2.62E+03 ◦	3.41E+03 ▼	1.73E+04 ▼	2.09E+04 ▼
f3	2.26E+05 Δ	1.42E+05 Δ	1.48E+06 ◦	6.26E+05 ◦	3.03E+06 ◦	1.40E+06 ◦	1.96E+06 ◦	1.10E+08 ▼	1.32E+08 ▼
f4	3.11E+02 ◦	2.80E+02 ◦	2.59E+02 ◦	2.29E+02 ◦	3.02E+02 ◦	2.63E+02 ◦	3.58E+02 ▼	1.14E+03 ▼	1.17E+03 ▼
f5	5.68E-15 Δ	5.79E-15 Δ	2.32E-11 ◦	4.76E-03 ▼	2.29E-01 ▼	4.97E-01 ▼	5.55E-01 ▼	5.77E+00 ▼	8.18E+00 ▼
f6	1.85E-02 ▼	1.36E-06 ▼	1.97E-15 ◦	1.99E-16 Δ	1.39E-14 ◦	3.92E-15 ◦	3.21E-16 ◦	6.20E-02 ▼	1.33E-01 ▼
f7	3.96E-03 Δ	2.66E-03 Δ	9.26E-03 ◦	1.59E-02 ◦	1.77E-02 ◦	4.97E-02 ▼	5.91E-02 ▼	1.64E+00 ▼	1.72E+00 ▼
f8	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦	2.03E+01 Δ	2.04E+01 ◦
f9	9.84E-01 ◦	9.84E-01 ◦	9.84E-01 ◦	9.84E-01 ◦	9.84E-01 ◦	4.74E-01 ◦	3.19E-01 ◦	1.09E+00 ▼	1.05E+00 ▼
f10	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ▼	3.91E-07 ▼	1.22E-03 ▼	7.40E-03 ▼	5.42E-02 ▼	4.75E-02 ▼
f11	9.95E-01 Δ	9.95E-01 ◦	9.95E-01 ◦	9.95E-01 ◦	9.95E-01 ◦	9.95E-01 ◦	9.95E-01 ◦	1.99E+00 ▼	1.99E+00 ▼
f12	9.95E-01 Δ	9.95E-01 Δ	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ▼	2.04E+00 ▼
f13	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ▼	1.99E+00 ◦	2.09E+00 ▼	2.00E+00 ▼
f14	4.06E+01 ◦	1.35E+02 ▼	1.52E+01 ◦	1.52E+01 ◦	2.35E+01 ◦	1.52E+01 ◦	1.53E+01 ◦	3.41E+01 ◦	3.47E+01 ◦
f15	1.40E+02 ▼	1.47E+02 ▼	1.31E+02 ◦	1.36E+02 ◦	1.35E+02 ◦	1.30E+02 ◦	1.34E+02 ◦	1.34E+02 ◦	1.30E+02 Δ
f16	1.19E+00 ▼	1.16E+00 ▼	1.09E+00 ◦	1.20E+00 ▼	1.18E+00 ◦	1.12E+00 ◦	1.12E+00 ◦	1.17E+00 ▼	1.18E+00 ◦
f17	2.53E+01 ▼	2.51E+01 ▼	2.23E+01 ◦	2.20E+01 ◦	2.30E+01 ◦	2.00E+01 Δ	1.83E+01 Δ	1.21E+01 Δ	1.19E+01 Δ
f18	2.83E+01 ◦	2.64E+01 ◦	2.71E+01 ◦	2.64E+01 ◦	2.65E+01 ◦	2.67E+01 ◦	2.66E+01 ◦	1.95E+01 Δ	1.88E+01 Δ
f19	9.75E-01 ▼	9.34E-01 ◦	8.61E-01 ◦	8.65E-01 ◦	8.69E-01 ◦	8.03E-01 ◦	8.10E-01 ◦	6.64E-01 Δ	6.78E-01 Δ
f20	1.92E+00 ◦	2.09E+00 ◦	2.02E+00 ◦	1.86E+00 Δ	1.98E+00 ◦	1.93E+00 ◦	2.12E+00 ◦	1.92E+00 ◦	2.13E+00 ◦
f21	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦
f22	2.80E+01 ◦	2.49E+01 ◦	2.42E+01 ◦	2.51E+01 ◦	2.52E+01 ◦	2.73E+01 ◦	2.88E+01 ◦	4.32E+01 ▼	5.57E+01 ▼
f23	3.32E+01 ◦	3.29E+01 ◦	4.08E+01 ◦	4.01E+01 ◦	5.45E+01 ◦	5.21E+01 ◦	5.98E+01 ▼	8.25E+01 ▼	7.53E+01 ▼
f24	2.00E+02 ◦	2.00E+02 ◦	2.00E+02 ◦	2.00E+02 ◦	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ▼	2.01E+02 ▼	2.01E+02 ▼
f25	2.00E+02 ▼	2.00E+02 ◦	2.00E+02 ◦	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ▼	2.00E+02 ▼	2.02E+02 ▼	2.02E+02 ▼
f26	1.00E+02 ◦	1.00E+02 ◦	1.00E+02 ◦	1.01E+02 ▼	1.01E+02 ▼	1.01E+02 ▼	1.01E+02 ▼	1.02E+02 ▼	1.02E+02 ▼
f27	3.00E+02 ◦	3.00E+02 ◦	3.00E+02 ◦	3.00E+02 ◦	3.00E+02 ▼	3.00E+02 ▼	3.00E+02 ▼	3.03E+02 ▼	3.04E+02 ▼
f28	3.01E+02 Δ	3.00E+02 Δ	3.10E+02 ◦	3.14E+02 ◦	3.20E+02 ▼	3.21E+02 ▼	3.20E+02 ▼	3.54E+02 ▼	3.70E+02 ▼
	$B = 6$ $I = 16$ $W = 6$	$B = 6$ $I = 17$ $W = 5$	(RA)	$B = 6$ $I = 20$ $W = 2$	$B = 8$ $I = 20$ $W = 0$	$B = 10$ $I = 17$ $W = 1$	$B = 12$ $I = 15$ $W = 1$	$B = 20$ $I = 4$ $W = 4$	$B = 19$ $I = 5$ $W = 4$
R_j	4.7	4.0	3.7	3.9	5.3	4.4	5.1	6.6	7.2
Poř.	5	3	1	2	7	4	6	8	9
$F_f = 44.65$, p -hodnota = 4.29E-07, $(\alpha = 0.05)$ platí H_1									

5.1.8 Porovnání různých topologií

Posledním parametrem, kterým se chci v této kapitole podrobněji zabývat, je komunikační topologie.

Pro experiment jsem použil toto nastavení:

- velikost populace: 250 jedinců na každém ostrově,
- výběr: nejlepších 50 jedinců,
- obnova populace: zachování 3 nejlepší jedinci,
- kopule: Gaussova,
- počet ostrovů: 4,
- perioda migrací: 5 generací,
- topologie: biring, uniring, star, random, nebo zero,
- počet dimenzí: 10,
- počet evaluací fitness: 100000.

Souhrnné výsledky Wilcoxonových testů jsou na obrázku 5.14 (pro klasické testovací úlohy) a na obrázku 5.15 (pro sadu CEC 2013), v tabulkách 5.13 a 5.14 jsou uvedeny mediány pro každou testovací úlohu a jednotlivé výsledky Wilcoxonových testů.

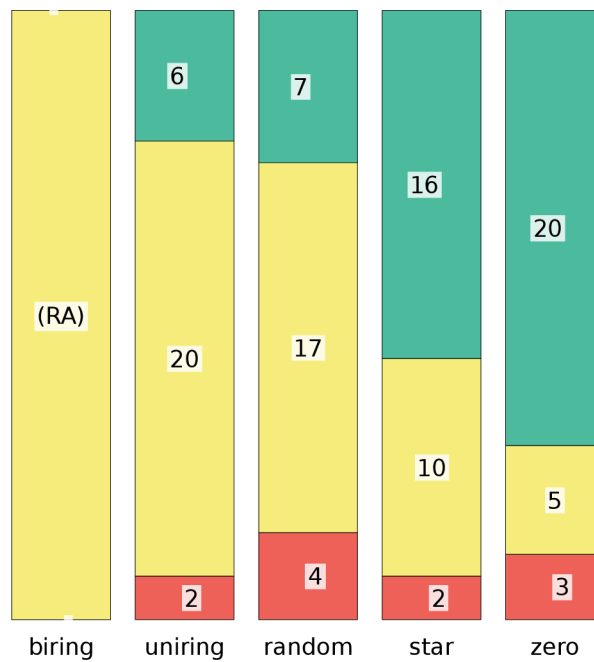
Jako nejlepší topologii jsem identifikoval **obousměrný prstenec**. Nicméně podobně jako u předešlého parametru, ani zde není volba kritická a i jiné topologie – jednosměrný prstenec a náhodná – dávají jen mírně zhoršený výkon. Toto zhoršení lze vysvětlit nižší mírou komunikace mezi ostrovy (u obousměrného prstence v rámci jednoho migračního cyklu každý ostrov odesílá svůj model na dva jiné ostrovy, u jednosměrného prstence i náhodné topologie jen na jeden). To potvrzuje i výsledek Friedmanova testu, který preferuje s minimálním odstupem topologii jednosměrný prstenec.

Výrazně horší výkon měla topologie typu hvězda (aktuálně nejlepší ostrov vybrán jako střed). Zde pravděpodobně došlo k předčasné konvergenci všech populací ke stejnému lokálnímu optimu a tím i příliš velké ztrátě diverzity (kvůli rozesílání stejného modelu na všechny ostrovy).

Zřetelně nejhoršího výsledku bylo dosaženo na topologii s deaktivovanou migrací. Tím je (stejně jako u příliš velké periody mezi migracemi v předešlé kapitole) potvrzen vliv komunikace mezi ostrovy na kvalitu konvergence testovaného algoritmu – nepostačuje jen spustit evoluci na několika izolovaných ostrovech a jako celkový výsledek evoluce vybrat z ostrovů ten nejlepší, zcela nezbytná je i komunikace mezi ostrovy v průběhu evoluce.



Obrázek 5.14: Výsledky experimentu zkoumajícího vliv různých **topologií** – souhrnné porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.



Obrázek 5.15: Výsledky experimentu zkoumajícího vliv různých **topologií** – souhrnné porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

Tabulka 5.13: Výsledky experimentu zkoumajícího vliv různých **topologií** – mediány a porovnání variant, výsledky pro klasické testovací úlohy. Způsob vyhodnocení viz kap. 5.1.1.

	biring	uniring	random	star	zero
Sphere	4.68E-30 ◦	2.82E-31 △	3.43E-29 ▼	1.17E-30 ◦	3.92E-19 ▼
Ackley	4.00E-15 ◦	4.00E-15 △	4.00E-15 ▼	4.00E-15 ◦	3.45E-10 ▼
Schweffel	4.74E+02 ◦	4.74E+02 ◦	3.91E+02 △	4.74E+02 ◦	4.37E+02 △
Rastrigin	0.00 ◦	0.00 ◦	0.00 ▼	0.00 ▼	5.05E-04 ▼
Rosenbrock	6.48E+00 ◦	6.08E+00 △	5.60E+00 △	6.68E+00 ▼	6.79E+00 ▼
Griewank	0.00 ◦	0.00 ◦	0.00 ◦	0.00 ◦	6.41E-13 ▼
	(RA)	$B = 0$ $I = 3$ $W = 3$	$B = 3$ $I = 1$ $W = 2$	$B = 2$ $I = 4$ $W = 0$	$B = 5$ $I = 0$ $W = 1$
R_j	2.9	2.4	2.3	2.9	4.5
Pořadí	3	2	1	3	5
$F_f = 7.60$, p -hodnota = 0.10738, ($\alpha = 0.05$) platí H_0					

Tabulka 5.14: Výsledky experimentu zkoumajícího vliv různých **topologií** – mediány a porovnání variant, výsledky pro sadu CEC 2013. Způsob vyhodnocení viz kap. 5.1.1.

	biring	uniring	random	star	zero
f1	3.42E-27 ◦	3.50E-27 ◦	1.07E-17 ▼	1.43E-05 ◦	6.67E-01 ▼
f2	2.32E+03 ◦	2.73E+03 ◦	2.32E+03 ◦	3.53E+03 ▼	2.48E+04 ▼
f3	1.46E+06 ◦	1.09E+05 △	1.11E+05 △	7.09E+06 ▼	1.21E+08 ▼
f4	3.05E+02 ◦	2.38E+02 ◦	3.71E+02 ▼	4.04E+02 ▼	1.24E+03 ▼
f5	1.10E-05 ◦	4.62E-03 ◦	2.14E-06 ◦	7.73E-01 ▼	7.70E+00 ▼
f6	2.81E-16 ◦	7.05E-14 ◦	1.18E-11 ◦	3.52E-06 ◦	1.61E-01 ▼
f7	7.26E-03 ◦	1.87E-02 ▼	4.83E-02 ▼	1.30E-01 ▼	1.41E+00 ▼
f8	2.04E+01 ◦	2.04E+01 ▼	2.04E+01 ◦	2.04E+01 ◦	2.04E+01 ◦
f9	9.84E-01 ◦	9.84E-01 ◦	5.18E-01 ◦	1.00E+00 ▼	1.12E+00 ▼
f10	0.00 ◦	4.66E-15 ▼	1.33E-15 ▼	1.48E-02 ▼	6.16E-02 ▼
f11	9.95E-01 ◦	9.95E-01 ◦	9.95E-01 ◦	1.99E+00 ▼	1.99E+00 ▼
f12	9.95E-01 ◦	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ▼	1.99E+00 ▼
f13	1.99E+00 ◦	1.99E+00 ◦	1.99E+00 ◦	2.00E+00 ▼	2.52E+00 ▼
f14	4.87E+01 ◦	1.64E+01 △	2.34E+01 ◦	4.06E+01 ◦	2.92E+01 ◦
f15	1.36E+02 ◦	1.34E+02 ◦	1.35E+02 △	1.38E+02 ◦	1.36E+02 ◦
f16	1.11E+00 ◦	1.11E+00 ◦	1.15E+00 ◦	1.12E+00 ◦	1.24E+00 ▼
f17	2.28E+01 ◦	2.18E+01 ◦	1.67E+01 △	1.69E+01 △	1.20E+01 △
f18	2.71E+01 ◦	2.60E+01 ◦	2.68E+01 ◦	2.71E+01 ◦	2.16E+01 △
f19	9.12E-01 ◦	8.91E-01 ◦	7.78E-01 △	7.68E-01 △	7.07E-01 △
f20	1.97E+00 ◦	2.01E+00 ◦	2.03E+00 ◦	2.05E+00 ◦	2.13E+00 ◦
f21	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦	4.00E+02 ◦
f22	2.92E+01 ◦	2.54E+01 ◦	2.66E+01 ◦	4.03E+01 ◦	4.26E+01 ▼
f23	4.19E+01 ◦	5.45E+01 ▼	4.54E+01 ◦	6.96E+01 ▼	6.85E+01 ▼
f24	2.00E+02 ◦	2.00E+02 ◦	2.00E+02 ▼	2.00E+02 ▼	2.01E+02 ▼
f25	2.00E+02 ◦	2.00E+02 ▼	2.00E+02 ◦	2.00E+02 ▼	2.02E+02 ▼
f26	1.00E+02 ◦	1.01E+02 ◦	1.01E+02 ▼	1.02E+02 ▼	1.02E+02 ▼
f27	3.00E+02 ◦	3.00E+02 ▼	3.00E+02 ▼	3.00E+02 ▼	3.02E+02 ▼
f28	3.04E+02 ◦	3.10E+02 ◦	3.01E+02 ◦	3.22E+02 ▼	3.52E+02 ▼
	(RA)	$B = 6$ $I = 20$ $W = 2$	$B = 7$ $I = 17$ $W = 4$	$B = 16$ $I = 10$ $W = 2$	$B = 20$ $I = 5$ $W = 3$
R_j	2.4	2.3	2.4	3.7	4.2
Pořadí	2	1	3	4	5
$F_f = 35.21$, p -hodnota = 4.21E-07, ($\alpha = 0.05$) platí H_1					

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:

- velikost populace: 250 jedinců na každém ostrově,
- výběr: nejlepších 50 jedinců,
- obnova populace: zachování 3 nejlepší jedinci,
- kopule: Gaussova,
- počet ostrovů: 4,
- perioda migrací: 5 generací,
- topologie: biring,
- počet dimenzí: 10,
- počet evaluací fitness: 100000, popř. 300000.

5.2.1 Porovnání mCEDA s existujícími kopulovými algoritmy EDA

V tabulkách 5.15a až 5.15c 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.

Srovnání je provedeno s následujícími algoritmy: Copula Bayesian Network [52]; Copula EDA, Copula EDA of Dynamic K-S test [81]; Clayton, Gumbel, Sn-EDA [37]; UMDAc, MIMICc, cEDA(empirical), cEDA(normal), cEDA(adaptive) [77].

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.

V tabulkách 5.16a a 5.16b jsou uvedeny podrobné výsledky mého algoritmu, použité pro tato srovnání. Průběh evoluce pro 10 dimenzí je také ukázán na obrázku 5.16.

Tabulka 5.15: Porovnání algoritmu mCEDA (střední hodnota, nejlepší je vyznačena tučně) pro klasické testovací úlohy, 10 dimenzí, vůči algoritmům: Copula Bayesian Network (CBN) [52]; Copula EDA (cEDA), Copula EDA of Dynamic K-S test (cEDA-KS) [81]; Clayton, Gumbel, Sn-EDA [37]; UMDAc, MIMICc, cEDA(empirical), cEDA(normal), cEDA(adaptive) [77].

(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(empirical)	cEDA(normal)	cEDA(adaptive)
Sphere	2.49E-89	2.54E-23	6.88E-24	2.12E-10	1.17E-14	8.01E-13
Rosenbrock	6.53E+00	7.56E+00	6.54E+00	6.78E+00	6.09E+00	4.24E+00
Schwefel	4.56E+02	1.81E-02	1.88E-03	1.14E-04	1.14E-07	4.40E-07
Griewank	0.00	4.26E-03	1.90E-02	3.41E-02	1.69E-10	4.25E-13

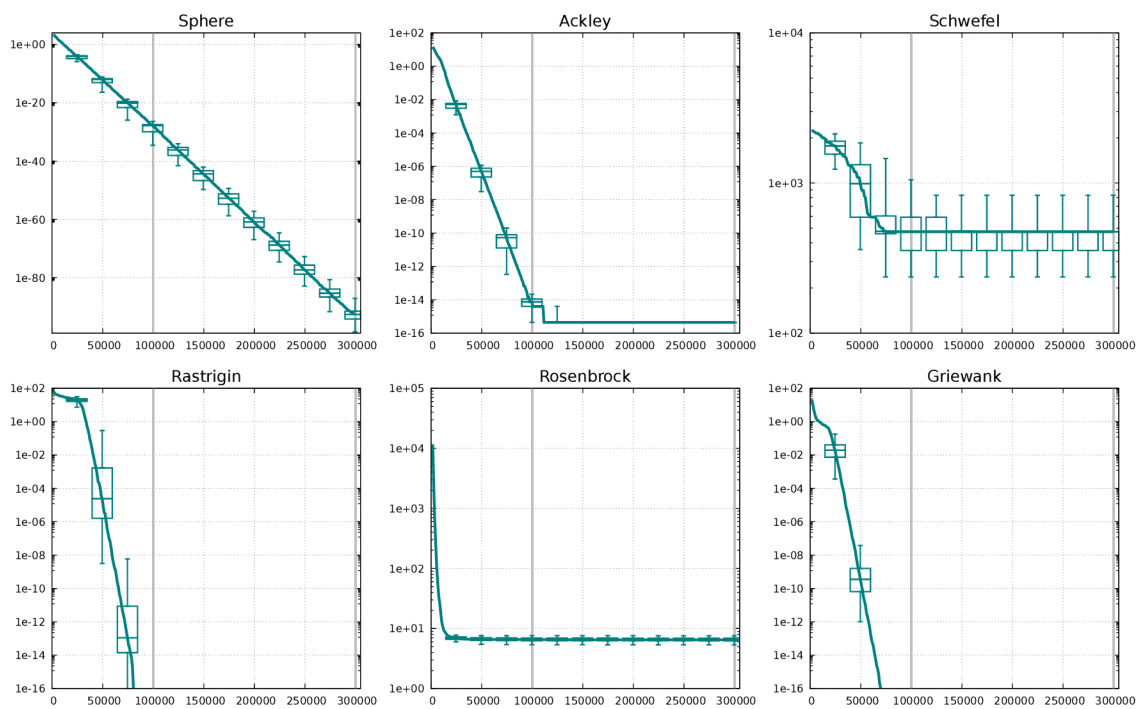
Tabulka 5.16: Výsledky navrženého algoritmu mCEDA na sadě klasických testovacích úloh pro 10 dimenzí, výsledky z 51 náhodných běhů.

(a) po uplynutí 100000 evaluací fitness funkce

	Best	Median	Worst	Average	Std.dev
Sphere	2.72E-35	1.52E-28	4.26E-27	3.34E-28	6.51E-28
Ackley	4.44E-16	7.55E-15	2.18E-14	7.34E-15	4.58E-15
Schwefel	2.37E+02	4.74E+02	1.05E+03	4.83E+02	1.59E+02
Rastrigin	0.00	0.00	0.00	0.00	0.00
Rosenbrock	5.38E+00	6.55E+00	7.62E+00	6.58E+00	5.03E-01
Griewank	0.00	0.00	0.00	0.00	0.00

(b) po uplynutí 300000 evaluací fitness funkce

	Best	Median	Worst	Average	Std.dev
Sphere	2.67E-99	2.49E-93	8.42E-88	2.49E-89	1.28E-88
Ackley	4.44E-16	4.44E-16	4.44E-16	4.44E-16	0.00
Schwefel	2.37E+02	4.74E+02	8.29E+02	4.56E+02	1.21E+02
Rastrigin	0.00	0.00	0.00	0.00	0.00
Rosenbrock	5.34E+00	6.49E+00	7.58E+00	6.53E+00	5.06E-01
Griewank	0.00	0.00	0.00	0.00	0.00



Obrázek 5.16: Průběh evoluce algoritmu mCEDA na sadě klasických testovacích úloh, 10 dimenzí (svislá osa – chybová funkce, vodorovná osa – počet evaluací).

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 [47], iCMAES-ILS [44], CMA-ES-RIS [7], SPSO2011 [72], PSO [79], GA-TPC [20], LaF [9], SPAM-AOS [21], DE/rand/1/bin [59], NMS-CS [38].

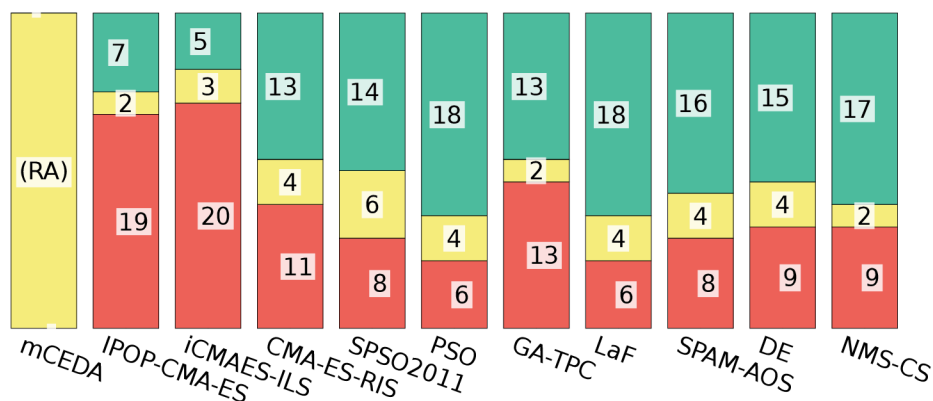
Srovnání je provedeno pomocí dvouvýběrového t-testu na hranici významnosti $\alpha = 0.05$. Wilcoxonův test nebyl použit, protože v citovaných člancích nejsou uvedeny výsledky jednotlivých běhů, nutné pro provedení Wilcoxonova testu.

Výsledky srovnání jsou uvedeny v tab. 5.17, která má stejný styl jako v předešlé kap. 5.1. 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 (místo mediánu) a výsledek dvouvýběrového t-testu. Na obrázku 5.17 jsou souhrnné výsledky tohoto srovnání znázorněny graficky.

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 částí 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 složité 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.

V tabulkách 5.18a až 5.18c jsou uvedeny samostatně kompletní výsledky mého algoritmu podle metodiky pro sadu CEC 2013. Průběh evoluce pro 10 dimenzí je také ukázán na obrázku 5.18.



Obrázek 5.17: Histogram výsledků testu statistické významnosti rozdílu (podle dvouvýběrového t-testu na hranici významnosti $\alpha = 0.05$) mezi navrženým algoritmem mCEDA a algoritmy jiných autorů (na sadě CEC 2013 pro 10 dimenzí): IPOP-CMA-ES [47], iCMAES-ILS [44], CMA-ES-RIS [7], SPSO2011 [72], PSO [79], GA-TPC [20], LaF [9], SPAM-AOS [21], DE/rand/1/bin [59], NMS-CS [38]. Grafický klíč je obdobný jako v předešlé kapitole, viz kap. 5.1.1.

Tabulka 5.17: Porovnání algoritmu mCEDA (střední hodnota, nejlepší je vyznačena tučně a výsledek dvouvýběrového t-testu na hranici významnosti $\alpha = 0.05$) na sadě CEC 2013, 10 dimenzí, vůči algoritmům: IPOP-CMA-ES [47], iCMAES-ILS [44], CMA-ES-RIS [7], SPSO2011 [72], PSO [79], GA-TPC [20], LaF [9], SPAM-AOS [21], DE/rand/1/bin [59], NMS-CS [38]. Grafický klíč je obdobný jako v předešlé kapitole, viz kap. 5.1.1.

	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	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 \circ	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 Δ	1.8E+01 Δ	1.9E+01 Δ	2.2E+01 Δ	4.2E+01 ∇	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 \circ	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 \circ	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)	$B = 7$ $I = 2$ $W = 19$	$B = 5$ $I = 3$ $W = 20$	$B = 13$ $I = 4$ $W = 11$	$B = 14$ $I = 6$ $W = 8$	$B = 18$ $I = 4$ $W = 6$	$B = 13$ $I = 2$ $W = 13$	$B = 18$ $I = 4$ $W = 6$	$B = 16$ $I = 4$ $W = 8$	$B = 15$ $I = 4$ $W = 9$	$B = 17$ $I = 2$ $W = 9$

Tabulka 5.18: Výsledky navrženého algoritmu mCEDA na sadě CEC 2013.

(a) pro 10 dimenzí

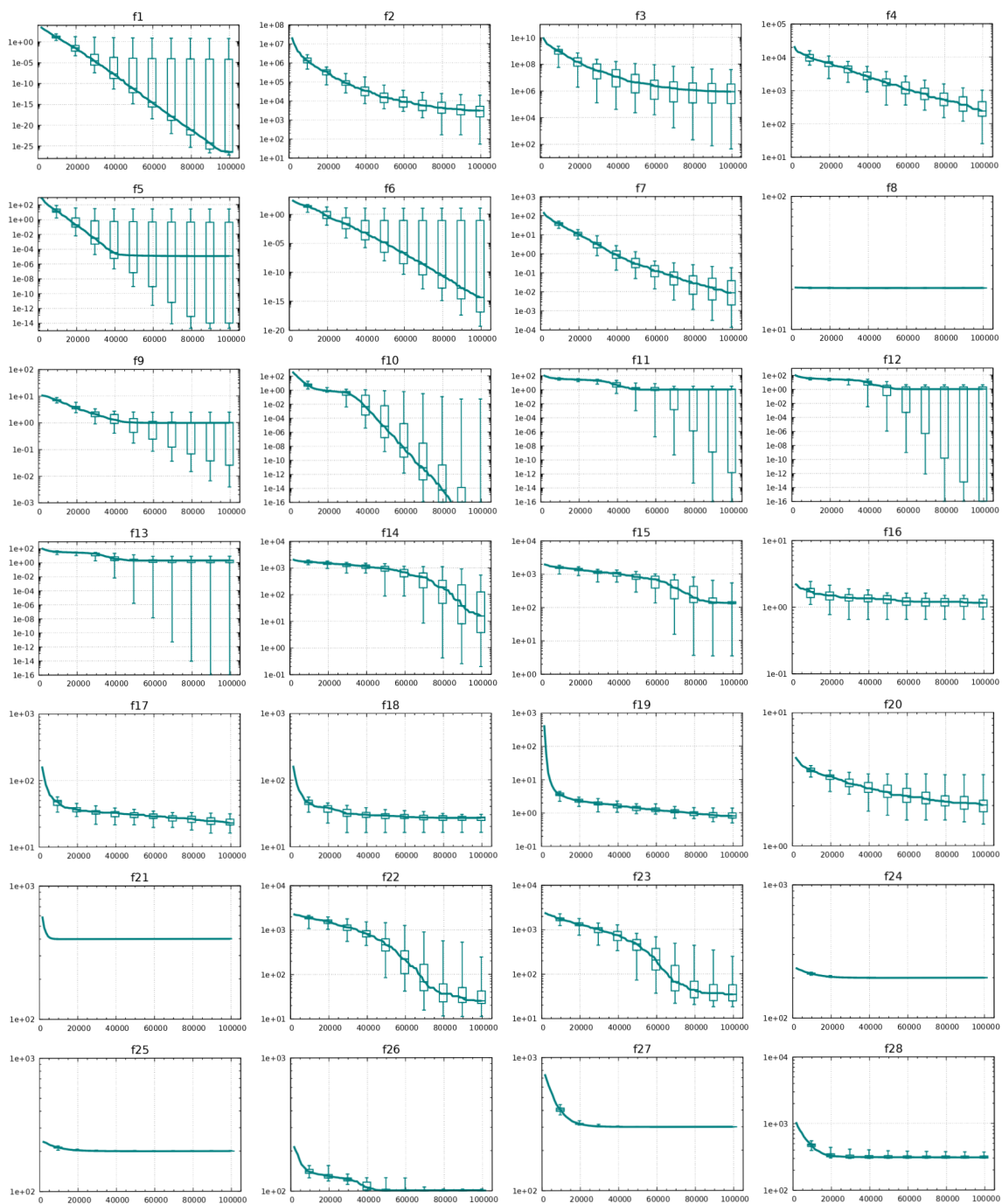
	Best	Median	Worst	Average	Std.dev
f1	0.00	0.00	5.81E+00	2.76E-01	9.86E-01
f2	5.24E+01	3.01E+03	1.93E+04	3.92E+03	3.51E+03
f3	4.29E+01	8.56E+05	3.83E+07	4.54E+06	8.58E+06
f4	2.51E+01	2.41E+02	1.01E+03	3.25E+02	2.22E+02
f5	0.00	1.11E-05	2.51E+01	1.65E+00	4.76E+00
f6	0.00	0.00	1.06E+01	8.00E-01	2.44E+00
f7	1.29E-04	8.45E-03	1.76E-01	2.45E-02	3.81E-02
f8	2.02E+01	2.04E+01	2.05E+01	2.04E+01	6.32E-02
f9	3.86E-03	9.84E-01	2.46E+00	6.60E-01	6.70E-01
f10	0.00	0.00	5.17E-02	4.16E-03	1.04E-02
f11	0.00	9.95E-01	2.98E+00	9.58E-01	7.62E-01
f12	0.00	9.95E-01	3.98E+00	1.13E+00	1.12E+00
f13	0.00	1.99E+00	7.71E+00	2.03E+00	1.27E+00
f14	1.95E-01	1.55E+01	5.28E+02	7.41E+01	1.18E+02
f15	3.48E+00	1.35E+02	5.37E+02	1.51E+02	8.64E+01
f16	6.50E-01	1.14E+00	1.49E+00	1.13E+00	2.11E-01
f17	1.61E+01	2.29E+01	3.14E+01	2.35E+01	3.53E+00
f18	1.63E+01	2.69E+01	3.03E+01	2.62E+01	2.80E+00
f19	4.94E-01	8.05E-01	1.38E+00	8.34E-01	1.84E-01
f20	1.45E+00	2.02E+00	3.41E+00	2.07E+00	4.01E-01
f21	4.00E+02	4.00E+02	4.00E+02	4.00E+02	1.08E-05
f22	1.12E+01	2.51E+01	2.45E+02	3.64E+01	3.39E+01
f23	1.82E+01	3.46E+01	2.47E+02	4.77E+01	3.82E+01
f24	2.00E+02	2.00E+02	2.01E+02	2.00E+02	1.40E-01
f25	2.00E+02	2.00E+02	2.02E+02	2.00E+02	4.16E-01
f26	1.00E+02	1.01E+02	1.03E+02	1.01E+02	7.71E-01
f27	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.54E-02
f28	3.00E+02	3.09E+02	3.70E+02	3.14E+02	1.61E+01

(b) pro 30 dimenzí

	Best	Median	Worst	Average	Std.dev
1	1.75E+03	2.81E+03	4.29E+03	2.87E+03	5.53E+02
2	1.72E+06	5.03E+06	1.20E+07	5.35E+06	1.98E+06
3	5.82E+08	2.37E+09	7.39E+09	2.56E+09	1.42E+09
4	2.53E+02	1.17E+03	5.54E+03	1.31E+03	8.22E+02
5	2.65E+02	5.02E+02	9.38E+02	5.26E+02	1.46E+02
6	6.87E+01	1.17E+02	1.98E+02	1.24E+02	2.89E+01
7	1.17E+01	2.62E+01	4.75E+01	2.75E+01	8.73E+00
8	2.07E+01	2.10E+01	2.10E+01	2.09E+01	5.94E-02
9	6.77E+00	1.03E+01	1.53E+01	1.02E+01	1.70E+00
10	1.32E+02	2.25E+02	3.48E+02	2.29E+02	4.89E+01
11	3.41E+01	5.80E+01	8.67E+01	5.59E+01	1.25E+01
12	3.39E+01	5.84E+01	8.60E+01	5.92E+01	1.15E+01
13	3.87E+01	1.09E+02	1.73E+02	1.13E+02	2.75E+01
14	2.83E+02	5.97E+02	1.39E+03	6.22E+02	2.54E+02
15	3.96E+02	9.50E+02	1.76E+03	9.52E+02	2.49E+02
16	1.76E+00	2.48E+00	3.13E+00	2.46E+00	2.87E-01
17	3.68E+01	4.55E+01	5.99E+01	4.56E+01	5.09E+00
18	8.82E+01	1.74E+02	1.92E+02	1.71E+02	1.81E+01
19	3.37E+01	1.73E+02	4.93E+02	1.84E+02	1.04E+02
20	8.83E+00	1.06E+01	1.14E+01	1.06E+01	5.45E-01
21	3.90E+02	5.21E+02	9.29E+02	5.49E+02	1.08E+02
22	1.82E+02	5.19E+02	1.16E+03	5.45E+02	2.06E+02
23	3.99E+02	1.09E+03	6.14E+03	1.15E+03	7.78E+02
24	2.21E+02	2.30E+02	2.44E+02	2.31E+02	5.68E+00
25	2.09E+02	2.33E+02	3.47E+02	2.43E+02	2.76E+01
26	2.00E+02	2.00E+02	3.13E+02	2.02E+02	1.56E+01
27	4.32E+02	5.25E+02	6.20E+02	5.32E+02	3.64E+01
28	3.56E+02	8.84E+02	1.63E+03	8.73E+02	3.38E+02

(c) pro 50 dimenzí

	Best	Median	Worst	Average	Std.dev
1	4.24E+03	6.77E+03	9.95E+03	6.80E+03	9.73E+02
2	8.21E+06	1.75E+07	3.14E+07	1.84E+07	5.22E+06
3	5.03E+09	8.83E+09	2.16E+10	9.42E+09	3.37E+09
4	5.58E+02	2.03E+03	4.86E+03	2.12E+03	9.23E+02
5	5.83E+02	9.04E+02	1.38E+03	9.23E+02	1.66E+02
6	1.89E+02	2.98E+02	3.88E+02	2.97E+02	3.94E+01
7	2.69E+01	3.99E+01	4.92E+01	3.98E+01	5.07E+00
8	2.09E+01	2.11E+01	2.12E+01	2.11E+01	4.49E-02
9	2.21E+01	2.70E+01	3.30E+01	2.69E+01	2.23E+00
10	4.20E+02	6.62E+02	1.02E+03	6.64E+02	1.08E+02
11	1.09E+02	1.41E+02	1.83E+02	1.42E+02	1.75E+01
12	1.23E+02	1.64E+02	2.16E+02	1.65E+02	1.74E+01
13	2.28E+02	2.74E+02	3.50E+02	2.76E+02	2.70E+01
14	8.11E+02	1.92E+03	3.29E+03	1.88E+03	5.36E+02
15	2.45E+03	3.69E+03	5.18E+03	3.62E+03	5.28E+02
16	2.39E+00	3.29E+00	3.97E+00	3.28E+00	3.30E-01
17	1.01E+02	1.28E+02	1.72E+02	1.29E+02	1.53E+01
18	1.53E+02	3.54E+02	3.82E+02	3.48E+02	4.11E+01
19	6.53E+02	1.62E+03	5.26E+03	1.69E+03	8.03E+02
20	1.49E+01	1.98E+01	2.09E+01	1.93E+01	1.33E+00
21	6.52E+02	1.59E+03	2.73E+03	1.71E+03	4.53E+02
22	6.76E+02	2.07E+03	3.52E+03	2.20E+03	6.12E+02
23	2.25E+03	3.76E+03	5.31E+03	3.75E+03	7.10E+02
24	2.69E+02	2.98E+02	3.20E+02	2.99E+02	9.82E+00
25	3.41E+02	4.42E+02	5.14E+02	4.37E+02	5.02E+01
26	2.01E+02	3.52E+02	3.73E+02	3.42E+02	3.72E+01
27	9.18E+02	1.08E+03	1.27E+03	1.09E+03	8.91E+01
28	1.20E+03	1.73E+03	4.20E+03	1.85E+03	4.50E+02



Obrázek 5.18: Průběh evoluce algoritmu mCEDA na sadě CEC 2013, 10 dimenzí (svislá osa – chybová funkce, vodorovná osa – počet evaluací).

Kapitola 6

Závěr

Hlavním cílem této 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.

V počáteční fázi mého doktorského studia mne motivovala idea využití kopulí v evolučních algoritmech EDA. Druhým inspiračním zdrojem byla paralelizace výpočtu evolučního algoritmu pomocí ostrovního modelu populace.

Jedním z důvodů pro použití kopulí byla úspěšnost jejich použití ve finančnictví při odhadech vývoje akciových trhů a sledování investičních rizik. Jde však nutné poznamenat, že jde o dvě dosti odlišné aplikace kopulí. Zatímco ve finančnictví jde většinou o modelování statických dat a limitovanou predikci vývoje; jde u algoritmů EDA o dynamický proces evoluce datových struktur a jejich modelování s cílem nalezení extrému asociované účelové funkce.

6.1 Teorie kopulí

Počáteční část je věnována popisu teorie kopulí v takovém rozsahu a struktuře, který nepomíjí podstatné informace, ale přitom reflektuje stanovené cíle. V kapitole 2.4 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.

Otevřeným problémem řady publikovaných článků je zdůvodnění volby vhodné kopule a dále nedostatečné testování výkonnosti navržených algoritmů. Často je použit malý počet

testovacích úloh a odlišné testovací podmínky – je proto obtížné uskutečnit spravedlivé porovnání rozdílných algoritmů na základě publikovaných výsledků.

Součástí kapitoly je i přehled používaných testovacích úloh včetně jednotné metodiky a dále popis postupů pro statistické porovnávání výkonnosti různých algoritmů.

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 rozdělání jsem použil klasickou MLE optimalizaci, jako marginální rozdělání bylo použito normální rozdělání.

V paralelní části je řešena úloha migrace pravděpodobnostních modelů a jejich kombinace na rezidentním ostrově. Kombinace pravděpodobnostních modelů pro spojitě 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 rozdělání. Zatímco pro kombinace korelačních koeficientů τ a pro parametr marginálního rozdělání μ je použit základní adaptační vztah, parametr rozptylu σ je dán složitější funkcí, která je adekvátní vztahu publikovaného v [26].

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 jsou porovnávány pomocí série Wilcoxonových dvouvýběrových testů a Friedmanovy multinásobné komparace. Tuto analýzu jsem prováděl pro soubor CEC 2013 s 28 testovacími úlohami a na sadě šesti často používaných klasických testovacích úloh (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 na sadě klasických úloh.

Provedl jsem též srovnání mCEDA s algoritmy využívajícími rozdílná evoluční paradigmaty (kap. 5.2.2). Pro tyto algoritmy byly dostupné výsledky pro známou sadu úloh CEC 2013, což značně přispělo k významnosti statistické komparace s navrženým algoritmem mCEDA. 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 algoritmické struktury bez případné hybridizace a dalších přídatných 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) [31] 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 %.

Literatura

- [1] AAS, K. *Modelling the dependence structure of financial assets: A survey of four copulas*. December 01, date, 2004.
- [2] ADAMIDIS, P. *Parallel Evolutionary Algorithms: A Review*. Březen 1999.
- [3] BALUJA, S. *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. 1994.
- [4] 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.
- [5] 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.
- [6] BOUDT, K., CORNELISSEN, J. a CROUX, C. The Gaussian rank correlation estimator: robustness properties. *Statistics and Computing*. Mar 2012, sv. 22, č. 2, s. 471–483. DOI: 10.1007/s11222-011-9237-0. ISSN 1573-1375. Dostupné z: <https://doi.org/10.1007/s11222-011-9237-0>.
- [7] 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.
- [8] 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.
- [9] CHEN, S. a GONZALEZ-FERNANDEZ, Y. *Leaders and Followers on the CEC2013 Real-Parameter Optimization Benchmark Functions*. 2015. Technical Report School of Information Technology York University.
- [10] CHERIET, A., FOUJIL, C. a AHMED, A. Fast Solutions Enhancing using a Copula-based EDA and SVM for many-objective problems. *IFAC-PapersOnLine*. Prosinac 2016, sv. 49, s. 781–786. DOI: 10.1016/j.ifacol.2016.07.869.

- [11] COHEN, J. The Earth Is Round ($p < 0.5$). *American Psychologist*. 1994, sv. 49, č. 12, s. 997–1003.
- [12] DANIEL, W. *Applied nonparametric statistics*. PWS-Kent Publ., 1990. The Duxbury advanced series in statistics and decision sciences. ISBN 9780534919764.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] DEMARTA, S. a MCNEIL, A. J. The t Copula and Related Copulas. *International Statistical Review*. 2005, sv. 73, č. 1, s. 111–129. DOI: 10.1111/j.1751-5823.2005.tb00254.x. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1751-5823.2005.tb00254.x>.
- [17] DERRAC, J., GARCÍA, S., MOLINA, D. a HERRERA, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithm. *Swarm and Evolutionary Computation*. Březen 2011, sv. 1, s. 3–18. DOI: 10.1016/j.swevo.2011.02.002.
- [18] DUCHOSLAV, M. Úvod do testování statistických hypotéz. Katedra botaniky PřF UP. 2012. Dostupné z: <http://old.botany.upol.cz/prezentace/duch/statistika3.htm>.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] ETXEBERRIA, R. a LARRAÑAGA, P. Global optimization using Bayesian networks. In: *Second Symposium on Artificial Intelligence (CIMA-99)*. 1999, s. 332–339.
- [23] FARD, M. R. a MOHAYMANY, A. S. A copula-based estimation of distribution algorithm for calibration of microscopic traffic models. *Transportation Research Part C: Emerging Technologies*. 2019, sv. 98, s. 449 – 470. DOI:

<https://doi.org/10.1016/j.trc.2018.12.008>. ISSN 0968-090X. Dostupné z:
<http://www.sciencedirect.com/science/article/pii/S0968090X18302468>.

- [24] FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*. Taylor & Francis. 1937, sv. 32, č. 200, s. 675–701.
- [25] FRIEDMAN, M. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*. JSTOR. 1940, sv. 11, č. 1, s. 86–92.
- [26] FRÜHWIRTH SCHNATTER, S. *Finite Mixture and Markov Switching Models*. New York: Springer, 2006. ISBN 978-1-4419-2194-9.
- [27] 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.
- [28] GAO, Y., PENG, L., LI, F., LIU, M. a HU, X. Estimation of Distribution Algorithm with Multivariate T-Copulas for Multi-Objective Optimization. *Intelligent Control and Automation*. 2013, sv. 4, č. 1, s. 63–69.
- [29] GONZALEZ-FERNANDEZ, Y. a SOTO, M. *A Survey of Estimation of Distribution Algorithms Based on Copulas*. 2012. Technical Report ICIMAF 2012-679.
- [30] 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/>.
- [31] 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>.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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>.

- [36] 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.
- [37] 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.
- [38] JOVANOVIĆ, 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>.
- [39] KENDALL, M. G. A New Measure of Rank Correlation. *Biometrika*. [Oxford University Press, Biometrika Trust]. 1938, sv. 30, 1/2, s. 81–93. ISSN 00063444. Dostupné z: <http://www.jstor.org/stable/2332226>.
- [40] 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.
- [41] LARRAÑAGA, P. a LOZANO, J. A. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Norwell, MA, USA: Kluwer Academic Publishers, 2001. ISBN 0792374665.
- [42] 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.
- [43] 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.
- [44] 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.
- [45] 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>.
- [46] LOFTUS, G. R. Psychology will be a much better science when we change the way we analyze data. *Current directions in psychological science*. SAGE Publications Sage CA: Los Angeles, CA. 1996, sv. 5, č. 6, s. 161–171.
- [47] 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.

- [48] MADERA, J., ALBA, E. a OCHOA, A. A Parallel Island Model for Estimation of Distribution Algorithms. In: LOZANO, J. A., LARRAÑAGA, P., INZA, I. a BENGOTXEA, E., ed. *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, s. 159–186. ISBN 978-3-540-32494-2.
- [49] MAI, J. a SCHERER, M. *Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications*. London: Imperial College Press, 2012. Series in quantitative finance. ISBN 9781848168749.
- [50] MELCHIORI, M. R. Tools for sampling multivariate archimedean copulas. *YieldCurve, April*. 2006.
- [51] MELLO JUNIOR, H. D. de, MARTÍ, L., CRUZ, A. V. A. da a VELLASCO, M. M. R. Evolutionary algorithms and elliptical copulas applied to continuous optimization problems. *Information Sciences*. 2016, sv. 369, s. 419 – 440. DOI: <https://doi.org/10.1016/j.ins.2016.07.006>. ISSN 0020-0255. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0020025516304856>.
- [52] 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.
- [53] 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.
- [54] 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>.
- [55] NELSEN, R. B. *An Introduction to Copulas*. New York: Springer, 2006. Springer Series in Statistics. ISBN 0387286594.
- [56] 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.
- [57] 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.
- [58] PELIKAN, M. a MÜHLENBEIN, H. Marginal Distributions in Evolutionary Algorithms. In: *In Proceedings of the International Conference on Genetic Algorithms Mendel 98*. 1999, s. 90–95.
- [59] 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.

- [60] ROUSSEEUW, P. a MOLENBERGHS, G. Transformation of Non Positive Semidefinite Correlation Matrices. *Communications in Statistics - Theory and Methods*. Leden 1993, sv. 22, s. 965–984. DOI: 10.1080/03610928308831068.
- [61] RUDLOF, S. a KÖPPEN, M. Stochastic Hill Climbing with Learning by Vectors of Normal Distributions. In: 1996, s. 60–70.
- [62] 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.
- [63] 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.
- [64] 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.
- [65] 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.
- [66] 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.
- [67] 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.
- [68] SKLAR, A. Random Variables, Joint Distribution Functions, and Copulas. *Kybernetika*. 1973, sv. 9, č. 6, s. 449–460.
- [69] SKOLICKI, Z. M. *An Analysis of Island Models in Evolutionary Computation*. USA, 2007. Disertační práce. George Mason University. ISBN 9780549329619. AAI3289714.
- [70] SOTO, M., GONZÁLEZ-FERNÁNDEZ, Y. a OCHOA, A. Modeling with Copulas and Vines in Estimation of Distribution Algorithms. *CoRR*. 2012, abs/1210.5500.
- [71] SOUKUP, P. Nesprávná užívání statistické významnosti a jejich možná řešení. *Data a výzkum-SDA Info*. Sociologický ústav AV ČR, vvi. 2010, sv. 4, č. 2, s. 77–104.
- [72] 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.

- [73] 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.
- [74] 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.
- [75] 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.
- [76] WOLPERT, D. H. a MACREADY, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. April 1997, sv. 1, č. 1, s. 67–82. DOI: 10.1109/4235.585893. ISSN 1941-0026.
- [77] 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.
- [78] YANG, Y., CHENG, X., CHENG, J., JIANG, D. a LI, S. Improved Alopex-based evolutionary algorithm by Gaussian copula estimation of distribution algorithm and its application to the Butterworth filter design. *International Journal of Systems Science*. Taylor & Francis. 2018, sv. 49, č. 1, s. 160–178. DOI: 10.1080/00207721.2017.1390702. Dostupné z: <https://doi.org/10.1080/00207721.2017.1390702>.
- [79] 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.
- [80] ZAR, J. H. *Biostatistical analysis*. 5th ed. New Jersey: Pearson, 2010. Pearson international edition. ISBN 978-0-13-206502-3.
- [81] 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.
- [82] ZVÁRA, K., ŠTĚPÁN, J. a RIEČAN, B. *Pravděpodobnost' a matematická statistika*. Veda, 2002. ISBN 9788022407366. Dostupné z: <https://books.google.cz/books?id=hXnNPAAACAAJ>.
- [83] ZVÁROVÁ, J. *Základy statistiky pro biomedicínské obory*. 1. vyd. Praha : Karolinum: [b.n.], 2007. ISBN 978-80-7184-786-1.

Příloha A

Obsah přiloženého CD

- HyrsDR.pdf – tento text (elektronická verze)
- HyrsDR-print.pdf – tento text (verze pro tisk)
- publikace/ – elektronické verze publikací
- data/ – výsledky experimentů
- program/ – zdrojové kódy implementovaného algoritmu mCEDA
- latex/ – zdrojové kódy tohoto textu