



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

KVANTOVĚ INSPIROVANÉ OPTIMALIZAČNÍ ALGORITMY

QUANTUM-INSPIRED OPTIMIZATION ALGORITHMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ BÁRTŮ

VEDOUcí PRÁCE

SUPERVISOR

doc. Ing. MICHAL BIDLO, Ph.D.

BRNO 2025

Zadání diplomové práce



164566

Ústav: Ústav počítačových systémů (UPSY)
Student: **Bártů Tomáš, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Kybernetická bezpečnost
Název: **Kvantově inspirované optimalizační algoritmy**
Kategorie: Umělá inteligence
Akademický rok: 2024/25

Zadání:

1. Nastudujte problematiku optimalizačních algoritmů využívajících principy kvantové fyziky.
2. Zvolte vhodný optimalizační algoritmus z této kategorie a implementujte jej ve zvoleném prostředí.
3. Na vhodně zvolené problémové doméně (např. optimalizace trasy obchodního cestujícího) proveďte sadu experimentů demonstrujících schopnosti vybraného algoritmu, případně jeho modifikací, úlohy z dané domény řešit.
4. Porovnejte výsledky z bodu 3 s výsledky získanými pomocí běžných optimalizačních metod, např. simulovaného žihání nebo genetického algoritmu.
5. Vytvořte srovnávací studii s výsledky řešení vybraných úloh pomocí různých variant algoritmů z bodů 2 a 3.
6. Zhodnoťte dosažené výsledky a diskutujte možnosti pokračování projektu.

Literatura:

- Dle pokynů vedoucího projektu.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 a 2 zadání, demonstrace prototypu optimalizačního systému z bodu 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Bídlo Michal, doc. Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1.11.2024
Termín pro odevzdání: 21.5.2025
Datum schválení: 28.4.2025

Abstrakt

Tato práce se zabývá kvantově inspirovanými evolučními algoritmy, které kombinují principy kvantové mechaniky a evolučního počítání. Zaměřuje se na srovnání různých kvantově inspirovaných evolučních algoritmů při řešení problému batohu. Práce popisuje teoretické základy kvantového evolučního počítání, charakteristiky jednotlivých kvantově inspirovaných evolučních algoritmů. V závěru je provedeno jejich experimentální vyhodnocení společně s porovnáním s běžnými optimalizačními metodami. Součástí práce je také návrh vlastního kvantově inspirovaného evolučního algoritmu vycházejícího z principů částicových systémů.

Abstract

This thesis deals with quantum-inspired evolutionary algorithms that combine principles of quantum mechanics and evolutionary computation. It focuses on comparing various quantum-inspired evolutionary algorithms in solving the knapsack problem. The thesis describes the theoretical foundations of quantum evolutionary computation and the characteristics of individual quantum-inspired algorithms. It concludes with an experimental evaluation and a comparison with classical optimization methods. The thesis also presents a custom evolutionary algorithm that combines a quantum-inspired approach with principles of particle systems.

Klíčová slova

kvantově inspirované evoluční algoritmy, kvantově inspirovaný genetický algoritmus, kvantově inspirované simulované žíhání, kvantová evoluce roje, kvantově inspirovaná optimalizace rojem částic, problém batohu, kvantová mechanika

Keywords

quantum inspired evolutionary algorithms, quantum inspired genetic algorithm, quantum swarm evolutionary, quantum inspired particle swarm optimization, knapsack problem, quantum mechanics

Citace

BÁRTŮ, Tomáš. *Kvantově inspirované optimalizační algoritmy*. Brno, 2025. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Michal Bidlo, Ph.D.

Kvantově inspirované optimalizační algoritmy

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doc. Ing. Michal Bidla Ph.D. Pro úpravu jazykového stylu a zlepšení srozumitelnosti některých částí textu jsem využil generativní jazykové modely jako pomocný nástroj při formální úpravě textu. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Tomáš Bártů
19. května 2025

Poděkování

Rád bych tímto poděkoval panu doc. Ing. Michalovi Bidlovi Ph.D. za jeho ochotu a cenné rady, které mi v průběhu mé práce na diplomové práci poskytl.

Obsah

1	Úvod	6
2	Východiska pro kvantově inspirované evoluční algoritmy	7
2.1	Algoritmy inspirované přírodou	7
2.2	Evoluční algoritmy	9
2.3	Základy kvantové fyziky	14
2.4	Kvantová mechanika	15
2.5	Kvantové evoluční počítání	17
3	Kvantově inspirované evoluční algoritmy	21
3.1	Kódování řešení v kvantově inspirovaných evolučních algoritmech	21
3.2	Operátory v kvantově inspirovaných algoritmech	22
3.3	Kvantově inspirovaný genetický algoritmus	24
3.4	Kvantově inspirované simulované žíhání	27
3.5	Kvantová evoluce roje	31
3.6	Kvantově inspirovaná optimalizace rojem částic	33
3.7	Aplikace kvantově inspirovaných evolučních algoritmů	34
4	Optimalizace úlohy batohu pomocí kvantově inspirovaných evolučních al- goritmů	35
4.1	Základní popis problému batohu	35
4.2	Reprezentace problému batohu pro kvantově inspirované evoluční algoritmy	36
4.3	Návrh experimentů	38
5	Experimentální výsledky	44
5.1	Kvantově inspirovaný genetický algoritmus	44
5.2	Kvantově inspirované simulované žíhání	48
5.3	Kvantová evoluce roje	51
5.4	Kvantově inspirovaná optimalizace rojem částic	54
5.5	Srovnání algoritmů a diskuze	58
6	Závěr	62
	Literatura	63
A	Struktura odevzdaných souborů	68
B	Manuál	69

Seznam obrázků

2.1	Výsledkem sjednocení mnoha oblastí výzkumu přírodních procesů jsou nové způsoby řešení problémů. Obrázek byl převzat s úpravami z [10].	8
2.2	Mapování z prostoru genotypů na prostor fenotypů, kde každý fenotyp je následně mapován na hodnotu fitness. Obrázek byl převzat s úpravami z [6].	11
2.3	Populace je multimnožina genotypů, kde se mohou stejné genotypy vyskytovat vícekrát, přičemž složení populace se může měnit s každou novou generací pomocí selekce, rekombinace a mutace.	12
2.4	Proces mutace rodiče, kde funkce $p(x)$ určuje pravděpodobnostní rozhodnutí, zda bude provedena mutace konkrétní části genotypu. Na základě této série náhodných rozhodnutí vznikne potomek.	12
2.5	Ilustrace jednobodové binární rekombinace, kde se na základě náhodně vybraného bodu řezu kombinují informace z rodičovských genotypů do nově vzniklých potomků. Obrázek byl převzat s úpravami z [6]	13
2.6	Vazby mezi fyzikálními principy a algoritmy. Diagram byl převzat s úpravami z [6].	14
2.7	Kvantové evoluční počítání je kombinací kvantových a evolučních výpočtů, přičemž obě tyto oblasti spadají do výpočtů inspirovaných přírodou. Obrázek byl převzat s úpravami z [30].	18
2.8	Blochova sféra reprezentující qubit. Obrázek byl převzat s úpravami z [30]. .	19
3.1	Kvantové rotační hradlo. Obrázek byl převzat s úpravami z [6].	23
3.2	Znázornění principu kvantové rotace, kde je rotací o úhel $\Delta\theta_{j_i}$ počáteční stav $\psi_{j_i}(t-1) = [\alpha_{j_i}(t-1) \beta_{j_i}(t-1)]$ aktualizován na nový stav $\psi_{j_i}(t) = [\alpha_{j_i}(t) \beta_{j_i}(t)]$. Obrázek převzat s úpravami z [12].	26
3.3	Pokud $e_j^P(t) \leq e_j(t-1)$ pro $j = 1, 2, \dots, n$, kde uvažujeme i -tý kvantový bit $\psi_{j_i}(t-1)$ j -tého jedince a binární hodnotu $x_{j_i}(t)$, pak je kvantový bitu $\psi_{j_i}(t)$ nastaven na novou hodnotu $[1 - x_{j_i}(t) x_{j_i}(t)]$. Obrázek převzat s úpravami z [12].	29
3.4	Pokud $e_j^P(t) > e_j(t-1)$ pro $j = 1, 2, \dots, n$, kde uvažujeme i -tý kvantový bit $\psi_{j_i}(t-1)$ j -tého jedince a binární hodnotu $x_{j_i}(t)$, pak je nový kvantový bit $\psi_{j_i}(t)$ získán aktualizací koeficientů o úhel $b \cdot \Delta\theta_{j_i}(t)$ směrem k $x_{j_i}(t)$ pomocí kvantového rotačního hradla. Obrázek převzat s úpravami z [12]. . .	30
5.1	Vliv hodnot parametru $\Delta\theta$ při velikosti populace 1. Data prezentovaná na grafu byla získána algoritmem <i>QIGA</i> na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.	45

5.2	Závislost kvality řešení na velikosti populace pro různé hodnoty parametru $\Delta\theta$. Data prezentovaná na grafu byla získána algoritmem <i>QIGA</i> na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.	46
5.3	Závislost kvality řešení na velikosti populace pro různé hodnoty parametru $\Delta\theta$ u instance s 500 položkami. Data prezentovaná na grafu byla získána algoritmem <i>QIGA</i> na instanci velikosti 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.	46
5.4	Porovnání kvality řešení při hodnotě $\Delta\theta = 0,002$ a jednočlenné populaci na velkých instancích problému. Data prezentovaná na grafu byla získána algoritmem <i>QIGA</i> na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	47
5.5	Konvergenční křivky pro $\Delta\theta = 0,002$ a jednočlennou populaci. Data prezentovaná na grafu byla získána algoritmem <i>QIGA</i> na instancích, po řadě, velikosti 500 a 2 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	47
5.6	Porovnání vlivu různých zahřívacích funkcí a chladicích plánů pro různé míry ochlazování při jednočlenné populaci. Data prezentovaná na grafu byla získána algoritmem <i>QISA</i> na instancích, po řadě, velikosti 100 a 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.	49
5.7	Porovnání vlivu různých chladicích plánů a zahřívacích funkcí, jenž nevyužívají parametr míry ochlazování. Data prezentovaná na grafu byla získána algoritmem <i>QISA</i> na instancích, po řadě, velikosti 100 a 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.	49
5.8	Porovnání kvality nalezených řešení při rekursivně-logaritmickém chladicím plánu, sigmoidní zahřívací funkci při velkých instancích problému a jednočlenné populaci. Data prezentovaná na grafu byla získána algoritmem <i>QISA</i> na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	51
5.9	Konvergenční křivky pro konfiguraci algoritmu, jež byla nastavena na rekursivně-logaritmický chladicí plán a sigmoidní zahřívací funkci. Data prezentovaná na grafu byla získána algoritmem <i>QISA</i> na instancích, po řadě, velikosti 500 a 2 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	51
5.10	Porovnání vlivu různých velikostí populací a hodnot počátečních rychlostí na kvalitu řešení. Data prezentovaná na grafu byla získána algoritmem <i>QSE</i> na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.	52
5.11	Výsledky experimentů algoritmu <i>QSE</i> na instanci velikosti 250. Vlevo jsou znázorněny krabicové grafy dosažených hodnot fitness pro různé velikosti populace při počáteční rychlosti 1, vpravo pak konvergenční křivka vývoje fitness při populaci čítající 5 jedinců a počáteční rychlosti 1. Data byla získána pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.	52
5.12	Porovnání kvality nalezených řešení při hodnotě počáteční rychlosti 1 a pětičlenné populaci při velkých instancích problému. Data prezentovaná na grafu byla získána algoritmem <i>QSE</i> na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	53

5.13	Konvergenční křivky algoritmu, jehož parametr počáteční rychlosti byl nastaven na hodnotu 1 při pětičlenné populaci. Data prezentovaná na grafu byla získána algoritmem <i>QSE</i> na instancích, po řadě, velikosti 500 a 2000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	54
5.14	Závislost kvality řešení na velikosti populace pro různé hodnoty parametru ω . Data prezentovaná na grafu byla získána algoritmem <i>QIPSO</i> na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci testovaných parametrů.	54
5.15	Závislost kvality řešení na velikosti populace pro různé hodnoty, po řadě, kognitivního koeficientu a sociálního koeficientu. Data prezentovaná na grafu byla získána algoritmem <i>QIPSO</i> na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci testovaných parametrů.	55
5.16	Závislost kvality řešení na velikosti populace pro různé hodnoty, po řadě, kognitivního koeficientu, sociálního koeficientu a parametru tření. Data prezentovaná na grafu byla získána algoritmem <i>QIPSO</i> na instanci velikosti 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci testovaných parametrů.	55
5.17	Závislost kvality řešení na velikosti populace pro různé hodnoty počáteční rychlosti. Data prezentovaná na grafu byla získána algoritmem <i>QIPSO</i> na instanci velikosti 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci testovaných parametrů.	56
5.18	Porovnání kvality nalezených řešení při hodnotě počáteční rychlosti 100, kognitivního koeficientu 0,5, sociálního koeficientu 0,25, parametru tření 0,01 a pětičlenné populaci při velkých instancích problému. Data prezentovaná na grafu byla získána algoritmem <i>QIPSO</i> na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	57
5.19	Konvergenční křivky algoritmu, jehož parametr počáteční rychlosti byl nastaven na 100, kognitivního koeficientu na 0,5, sociálního koeficientu na 0,25, parametru tření na 0,01 při pětičlenné populaci. Data prezentovaná na grafu byla získána algoritmem <i>QIPSO</i> na instancích, po řadě, velikosti 500 a 2000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	58
5.20	Data prezentovaná na grafu byla získána, po řadě, algoritmy <i>QIGA</i> , <i>QISA</i> , <i>QSE</i> a <i>QIPSO</i> na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.	59
5.21	Data prezentovaná na grafu byla získána, po řadě, algoritmy <i>QIGA</i> , <i>QISA</i> , <i>QSE</i> a <i>QIPSO</i> na instanci velikosti a 10 000 pomocí 30 nezávislých běhů s 100 000 evaluacemi na běh.	60

Seznam tabulek

3.1	Vyhledávací tabulka pro parametr $\Delta\theta_i$	23
3.2	Vyhledávací tabulka pro parametr $\Delta\theta_i$ a $s(\alpha_i, \beta_i)$, kde je parametr δ obvykle nastaven na malou hodnotu (běžně 0.01π).	24
4.1	Přehled datových sad, optimálních hodnot a parametrů experimentů.	39
4.2	Parametry testování algoritmu <i>QIGA</i>	39
4.3	Hodnoty parametrů zahřívacích funkcí [12].	40
4.4	Parametry testování algoritmu <i>QISA</i>	41
4.5	Význam a hodnoty parametrů <i>QSE</i> [13, 38].	42
4.6	Parametry testování algoritmu <i>QSE</i>	42
4.7	Parametry testování algoritmu <i>QIPSO</i>	42
5.1	Nejlepší dosažené fitness hodnoty algoritmem <i>QIGA</i> pro prezentované hodnoty parametru $\Delta\theta$ a velikosti instancí.	48
5.2	Nejlepší dosažené fitness hodnoty algoritmem <i>QISA</i> pro prezentované kombinace nastavení chladičů plánu, zahřívací funkce a míry ochlazování při malých instancích problému.	50
5.3	Nejlepší dosažené fitness hodnoty algoritmem <i>QISA</i> pro prezentované nastavení při velkých instancích problému.	50
5.4	Nejlepší dosažené fitness hodnoty algoritmem <i>QSE</i> pro prezentované nastavení při velkých instancích problému při populaci čítající 5 jedinců.	53
5.5	Nejlepší dosažené fitness hodnoty algoritmem <i>QIPSO</i> pro prezentované nastavení při velkých instancích problému při populaci čítající 5 jedinců.	57
5.6	Nejlepší nalezené hodnoty parametrů pro jednotlivé algoritmy.	58
5.7	Nejlepší dosažené fitness hodnoty jednotlivými kvantově inspirovanými evolučními algoritmy při konfiguraci jejich parametrů dle tabulky 5.6 pro 10 000 evaluací.	59
5.8	Nejlepší dosažené fitness hodnoty jednotlivými kvantově inspirovanými evolučními algoritmy při konfiguraci jejich parametrů dle tabulky 5.6 pro 100 000 evaluací.	60
5.9	Nejlepší dosažené fitness hodnoty jednotlivými algoritmy.	61

Kapitola 1

Úvod

Na každém kroku se v běžném životě setkáváme s problémy, jako je například plánování tras, správa financí nebo organizace času, a algoritmy nám pomáhají tato rozhodnutí automatizovat pomocí výpočetní techniky. Ať už se jedná o nalezení nejkratší cesty v navigačním systému [11] nebo optimalizaci jízdního řádu metra [40], algoritmy se snaží najít optimální řešení z velkého množství možností. Některé z nich však spadají do skupiny problémů, u nichž je nalezení nejlepšího možného řešení výpočetně velmi náročné. Takové řešení není možné nalézt v rozumném čase, a proto se využívají přístupy, které se snaží vyhledat dostatečně kvalitní řešení v přijatelném čase.

Jedním z těchto přístupů je počítání podle přírody, což je disciplína zaměřená na návrh algoritmů, které se inspiřují přírodou a jejími procesy.

Do podoblasti počítání podle přírody spadá evoluční počítání, které zahrnuje algoritmy inspirované biologickou evolucí, souhrnně označované jako evoluční algoritmy [15]. Tyto algoritmy při výpočtu využívají evoluční mechanismy selekce, mutace a křížení za účelem efektivního prohledávání stavového prostoru. Přestože evoluční algoritmy našly uplatnění v široké škále aplikací, například při návrhu neuronových sítí nebo v medicínských diagnostických systémech [28], potýkají se s určitými omezeními, jako je uváznutí v lokálním minimu/maximu nebo pomalé konvergenci k řešení. Je proto důležité udržovat správnou rovnováhu mezi průzkumem stavového prostoru a zužitkováním oblastí se slibnými výsledky.

Do výpočetních metod inspirovaných přírodou spadá také fyzikální počítání, které využívá principy z různých oblastí fyziky. Jedním z příkladů je kvantová mechanika, která pracuje s fenoménem superpozice, kdy částice může být v současné době ve více stavech, dokud není pozorována. Tento princip ilustroval Erwin Schrödinger ve slavném myšlenkovém experimentu s kočkou v krabici, která je zároveň živá i mrtvá, dokud není krabice otevřena [26]. Na podobném principu umožňují kvantové výpočty zpracovávat více možností současně, což může výrazně urychlit řešení některých výpočetně náročných úloh. Navzdory tomu, že kvantové počítače zatím nejsou běžně dostupné, lze simulovat některé z kvantových principů i na klasických počítačích, což vedlo k vývoji kvantově inspirovaných algoritmů.

Kombinace principů zmíněných podoblastí počítání podle přírody vedla ke vzniku kvantově inspirovaných evolučních algoritmů [27, 47]. Tyto algoritmy využívají principy kvantové mechaniky ke zvýšení efektivity evolučního počítání při prohledávání prostoru řešení.

Cílem práce je analyzovat různé typy kvantově inspirovaných evolučních algoritmů při řešení vybraného optimalizačního problému, vyhodnotit získané výsledky a porovnat je s výsledky běžných optimalizačních metod. Součástí analýzy je také identifikace případných omezení a nedostatků zkoumaných přístupů. Na základě provedeného zhodnocení je dále navržen, implementován a posouzen vlastní kvantově inspirovaný evoluční algoritmus.

Kapitola 2

Východiska pro kvantově inspirované evoluční algoritmy

Přírodní procesy představují významný zdroj inspirace při návrhu algoritmů, neboť se ukázalo, že jejich využitím lze vytvořit účinné metody pro efektivní řešení složitých výpočetních úloh. Podněty pro tvorbu algoritmů lze čerpat nejen z biologických systémů a procesů, ale také z různých oblastí fyziky. Oba tyto přístupy je navíc možné kombinovat a vytvářet algoritmy, které spojují principy biologické evoluce s fyzikálními principy [6, 36].

Tato kapitola nejprve v krátkosti představí základní oblasti přírody, jimiž je možné se inspirovat ve výpočetních metodách. Dále budou ve stručnosti uvedeny základy evolučních algoritmů a v závěru kapitoly budou popsány vybrané principy kvantové fyziky a kvantové mechaniky, které slouží jako teoretický základ pro návrh kvantově inspirovaných evolučních algoritmů.

2.1 Algoritmy inspirované přírodou

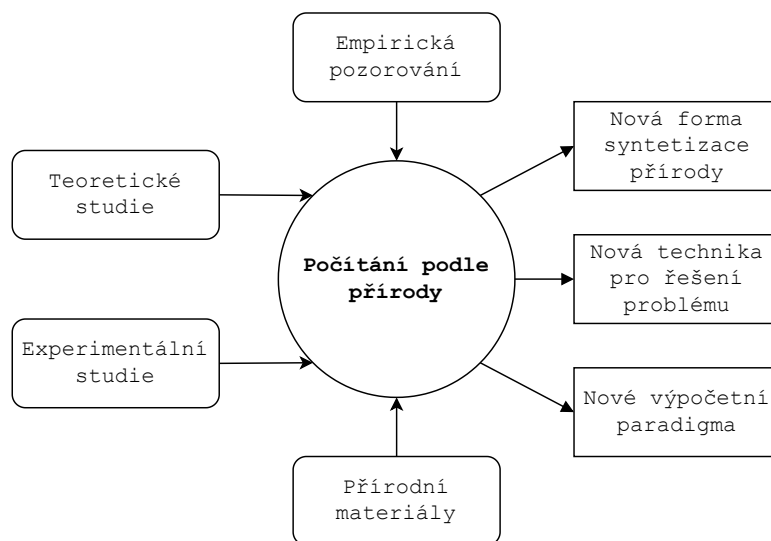
Počítání podle přírody (*natural computing*) se zaměřuje na objevování fundamentálních zákonů a mechanismů, na jejichž principu fungují procesy v přírodě, a na jejich následné využití při řešení různých výpočetních problémů. To vede k návrhu nových způsobů výpočtu a vývoji alternativních řešení různých typů úloh. Aby bylo možné efektivně modelovat systémy s velkým počtem prvků, je nutné zvolit vhodnou úroveň abstrakce, jež je závislá na charakteru řešené úlohy, neboť algoritmy inspirované přírodou vycházejí ze zjednodušeného pojetí přírodních procesů. Abstrakce zároveň umožňuje zdůraznit klíčové aspekty systému, jež jsou zásadní pro jeho reprodukovatelnost a analýzu vývojových procesů. Zvolená úroveň abstrakce je závislá na charakteru řešeného problému, typu modelovaného jevu a cílů simulace. Počítání podle přírody spojuje znalosti z teoretické a experimentální teorie fungování přírody společně s empirickými pozorováními za účelem návrhu nových způsobů pro řešení různých problémů, vizte obrázek 2.1 [10].

Počítání podle přírody je možné rozdělit do následujících tří oblastí [10]:

1. **Výpočty inspirované přírodou** vycházejí z předpokladu, že přírodní procesy dokáží efektivně řešit složité problémy, přičemž tyto přístupy lze rozdělit na dva hlavní směry. První vychází z teoretických modelů, pomocí nichž lze modelovat přírodní jevy, což pomáhá k lepšímu porozumění fungování různých procesů v přírodě. Druhým směrem je návrh algoritmů, které pomáhají řešit složité úlohy, u nichž tradiční přístupy selhávají, přičemž tyto algoritmy využívají určitou úroveň abstrakce nad

vybranými přírodními procesy. Mezi nejvýznamnější oblasti výpočtů inspirovaných přírodou patří [10, 6]:

- umělé neuronové sítě (*artificial neural networks*) inspirované nervovými systémy,
- evoluční algoritmy (*evolutionary algorithms*) vycházející z Darwinovy teorie evoluce,
- inteligence roje (*swarm intelligence*) založené na kolektivním chování společenských organismů a
- umělé imunitní systémy (*artificial immune systems*) vycházející z principů fungování imunitního systému.



Obrázek 2.1: Výsledkem sjednocení mnoha oblastí výzkumu přírodních procesů jsou nové způsoby řešení problémů. Obrázek byl převzat s úpravami z [10].

Příklady výše zmíněných oblastí zahrnují například aplikace umělých neuronových sítí při rozpoznávání hlasu, klasifikaci objektů nebo v oblasti počítačového vidění [2, 1]. Evoluční algoritmy je možné uplatnit při řešení různých optimalizačních problémů, zatímco inteligence roje je aplikovatelná například při návrhu autonomních robotických systémů nebo při prohledávání stavového prostoru [6, 24]. Umělé imunitní systémy nacházejí uplatnění u komplexních problémů v oblastech od biologie až po robotiku.

2. **Simulace a emulace přírody pomocí výpočetní techniky** se zaměřuje na syntézu a studium přírodních fenoménů, známých vzorců či chování a to prostřednictvím simulace, jež je realizována na výpočetních systémech. Tato oblast poskytuje nástroje pro testování biologických teorií, které by bylo obtížné ověřit pomocí experimentálních a analytických metod. Existují dva hlavní přístupy [10]:

- Fraktální geometrie v přírodě (*fractal geometry of nature*) poskytuje možnost vizualizovat různé přírodní struktury a procesy, jež se vyznačují nekonečnou úrovní detailu, nekonečnou délkou nebo soběpodobností. Struktury, na nichž je fraktální geometrie viditelná, se nacházejí například na kapradinách, horách nebo

brokolici a dokonce je patrná na organismech v podobě fraktální struktury plic, oběhového systému či mozku.

- Umělý život (*artificial life*) je oblast, která se snaží simulovat organismy v umělých prostředích. Hlavním cílem není řešení konkrétního problému, ale pochopení konceptů přírody, včetně vývoje nových forem života.

Aplikace fraktální geometrie lze využít při simulaci růstu rostlin nebo formování přírodních struktur, zatímco umělý život lze použít ke studiu evoluce a chování organismů ve virtuálním prostředí či k analýze počítačových virů [10].

3. Výpočty s využitím přírodních materiálů nepracují s křemíkem jako je tomu u tradičních výpočetních systémů, ale využívají odlišné materiály, jež umožňují překonat některé limity tradičních počítačů, jelikož poskytují odlišné přístupy k provádění výpočetních operací. Systémy založené na těchto principech nabízejí efektivní řešení problému limit miniaturizace elektroniky založené na křemíku. Alternativní výpočetní prostředky jsou [10]:

- Molekulární počítání (*molecular computing*) využívající biologické molekuly, jako je například DNA, k uchování informace, přičemž samotný paralelní výpočet probíhá manipulací s těmito molekulami. Výpočty založené na molekulární úrovni nabízejí vysokou výpočetní rychlost, energetickou efektivitu a levné úložiště informací.
- Kvantové počítání (*quantum computing*) využívající principy kvantové mechaniky, kde je informace uchována na mikroskopické úrovni.

Výše uvedené prostředky lze využít k efektivnímu řešení specifických problémů. Molekulární počítání například umožnilo efektivně řešit problém Hamiltonovské cesty [3], zatímco kvantové počítání se ukázalo jako efektivní například při faktorizaci čísel [39].

Tato kapitola se dále zaměří na kvantové evoluční počítání, které kombinuje principy kvantového výpočtu a evolučních algoritmů.

2.2 Evoluční algoritmy

Ačkoliv existuje mnoho variant evolučních algoritmů, všechny vycházejí ze stejné myšlenky. V prostředí s omezenými zdroji dochází mezi jednotlivci, jejichž schopnosti jsou určeny kvalitativní funkcí, k soutěži o přežití, jejímž výsledkem je přirozený výběr jedinců, jež během vývoje zvyšují kvalitu celé populace. Proces spočívá v aplikaci kvalitativní funkce na náhodně vygenerovanou populaci kandidátních řešení. Následně dochází k opakovanému výběru jedinců podstupujících reprodukci aplikací variančních operátorů (typicky křížení a mutace). Tento cyklus pokračuje až do nalezení dostatečně kvalitního řešení nebo do dosažení předem nastaveného počtu opakování. Evoluční proces, znázorněný v algoritmu 1, lze chápat jako průběžnou optimalizaci, jejímž cílem je nalezení co nejlepšího řešení prostřednictvím postupného přibližování se k optimálnímu stavu [15].

Základní elementy evolučního procesu jsou [15]:

- variační operátory (rekombinace a mutace), jež vytvářejí nové jedince a udržují diverzitu v populaci a
- selekce, která napodobuje přirozený výběr preferováním kvalitnějších jedinců.

Evoluční proces je založen na stochastickém chování, přičemž během selekce, rekombinace a mutace figuruje prvek náhody [15].

Algoritmus 1: Obecné schéma evolučního algoritmu [15]

```
1 Inicializace populace náhodně vygenerovanými kandidátními řešeními;  
2 Ohodnocení každého kandidátního řešení;  
3 while není splněna ukončovací podmínka do  
4   Selekce rodičů;  
5   Rekombinace vybraných rodičů;  
6   Mutace vzniklých potomků;  
7   Ohodnocení nových kandidátních řešení;  
8   Selekce jedinců pro další generaci;
```

Podle způsobu reprezentace kandidátních řešení a způsobu realizace některých operátorů lze evoluční algoritmy rozdělit do následujících kategorií [15]:

- evoluční strategie (*evolution strategies – ES*) pracující s reálnými vektory,
- evoluční programování (*evolutionary programming – EP*) využívají stavy automatů,
- genetické programování (*genetic programming – GP*) využívající stromové struktury a
- genetické algoritmy (*genetic algorithms – GA*) reprezentující kandidátní řešení pomocí binárních řetězců.

Rozdíly mezi výše uvedenými kategoriemi jsou především historického charakteru. Výběr vhodného typu totiž závisí na povaze řešeného problému, neboť některé typy reprezentací mohou být výhodnější než jiné, například pokud lépe odpovídají charakteru problému, usnadňují reprezentaci kandidátních řešení nebo jsou pro řešený problém přirozenější [15].

Pro definici konkrétního evolučního algoritmu je potřeba specifikovat jeho následující části:

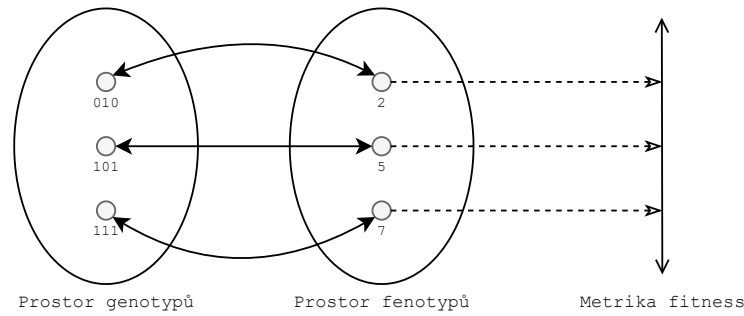
Reprezentace

Prvním krokem je definice reprezentace možných řešení problému, která umožňuje efektivní manipulaci s těmito řešeními v rámci algoritmu. Možná řešení problému, označovaná jako fenotypy, jsou zakódována do tvarů zvaných jako genotypy, s nimiž algoritmus dokáže pracovat. Dochází tak k mapování prostoru fenotypů na prostor genotypů, přičemž ohodnocovací funkce hodnotí fenotypy, ale ohodnocení je přiřazeno genotypům. Význam tohoto principu spočívá v tom, že pro algoritmus může být efektivnější pracovat například s binární reprezentací řešení, i když samotný problém má řešení v oblasti celých čísel, vizte obrázek 2.2 [15, 6].

V literatuře se v kontextu řešeného problému fenotyp označuje také jako kandidátní řešení nebo jedinec. Na straně evolučního algoritmu se pro genotyp často používají označení jako chromozom nebo opět jedinec [15].

Inicializace populace

V evolučních algoritmech je populace většinou inicializována pomocí náhodně generovaných jedinců. Pro inicializaci počáteční populace je možné využít také heuristiku, která, pokud je výpočetně přijatelná, vytvoří kvalitnější populaci [15].



Obrázek 2.2: Mapování z prostoru genotypů na prostor fenotypů, kde každý fenotyp je následně mapován na hodnotu fitness. Obrázek byl převzat s úpravami z [6].

Ohodnocovací funkce (fitness funkce)

Hlavním účelem ohodnocovací funkce je řízení evolučního procesu tím, že hodnotí, jak dobře jedinci splňují určené požadavky. Jedná se o funkci, jež přiřazuje míru kvality genotypům prostřednictvím ohodnocení odpovídajících fenotypů. Ohodnocovací funkce je běžně označována jako fitness funkce. V případě optimalizačního problému je často používán pojem objektivní funkce, který vyjadřuje specifické požadavky úlohy (například maximalizaci nebo minimalizaci určité hodnoty). Ohodnocovací (fitness) funkce $F(x)$ může být identická s objektivní funkcí nebo se může jednat o její transformaci:

$$F(x) = g(f(x)),$$

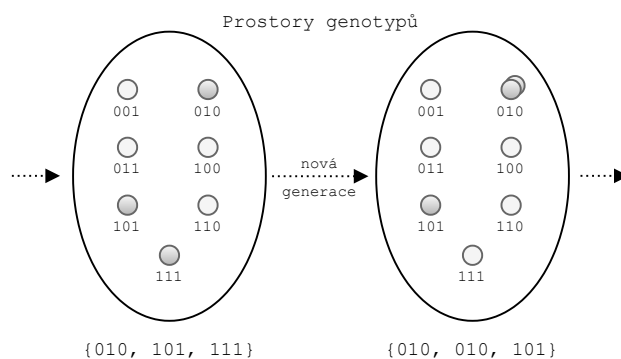
kde x je fenotyp, $y = f(x)$ je objektivní funkce a $g(y)$ je transformační funkce, jež upravuje hodnotu objektivní funkce pro potřeby selekce a hodnocení v evolučním algoritmu [15, 6].

Populace

Hlavní rolí populace v evolučních algoritmech je uchování možných řešení problému. Populace je chápána jako multimnožina genotypů, která se v průběhu evoluce vyvíjí, přičemž genotypy jsou pouze statické objekty, jenž se samy nemění ani nepřizpůsobují, protože změny nastávají až na úrovni populace, vizte obrázek 2.3. Velikost populace je obvykle konstantní, což vytváří podmínky pro přirozený výběr jedinců vlivem omezených zdrojů a nutnosti soutěžení o přežití [15].

Selekce rodičů

Rodičem se stává jedinec, jenž byl vybrán k tomu, aby podstoupil proces reprodukce, ať už rekombinací nebo mutací, čímž jsou vytvářeni potomci. Obvykle se rodičem stává kvalitní jedinec, avšak vzhledem k tomu, že proces selekce bývá stochastický, může se rodičem stát také méně kvalitní jedinec, což poskytuje vyšší diverzitu, jež pomáhá předcházet uvíznutí v lokálním optimu [15].

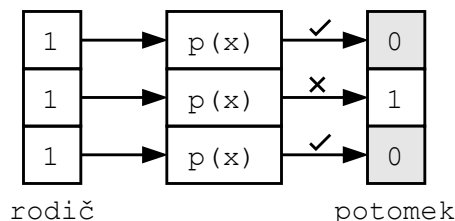


Obrázek 2.3: Populace je multimnožina genotypů, kde se mohou stejné genotypy vyskytovat vícekrát, přičemž složení populace se může měnit s každou novou generací pomocí selekce, rekombinace a mutace.

Variační operátory

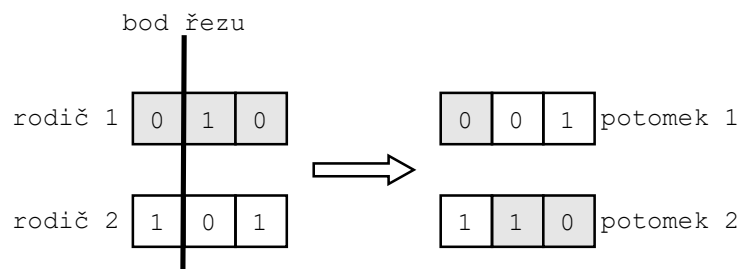
Cílem variačních operátorů je vytvořit z existujících jedinců jedince nové. V kontextu fenotypového prostoru dochází ke generování nových kandidátních řešení. Podle arity se variační operátory dělí na následující dva typy:

- **Mutace** je unární operátor, jenž je aplikován na rodiče a jehož výsledkem je potomek. Tento potomek vzniká na základě série náhodných voleb, která vede k úpravám genotypu, pokud jsou splněny určité podmínky, vizte obrázek 2.4 [15, 6].



Obrázek 2.4: Proces mutace rodiče, kde funkce $p(x)$ určuje pravděpodobnostní rozhodnutí, zda bude provedena mutace konkrétní části genotypu. Na základě této série náhodných rozhodnutí vznikne potomek.

- **Rekombinace**, někdy označovaná jako křížení, je n -ární operátor, jenž kombinuje informace z vícero rodičů. Standardně se v přírodě objevuje pouze binární rekombinace, ale v evolučním počítání je možné uvažovat i křížení s větší aritou. Podobně jako mutace je také rekombinace založena na stochastickém principu, což znamená, že výběr částí, které budou z každého rodiče kombinovány, probíhá na základě náhodné volby. Pro příklad jednoduché jednobodové rekombinace vizte obrázek 2.5 [15].



Obrázek 2.5: Ilustrace jednobodové binární rekombinace, kde se na základě náhodně vybraného bodu řezu kombinují informace z rodičovských genotypů do nově vzniklých potomků. Obrázek byl převzat s úpravami z [6]

Mutace v evolučních algoritmech zajišťuje neustálý vývoj evolučního procesu, jelikož v každé iteraci s určitou pravděpodobností umožňuje objevování nových vlastností. Na rozdíl od rekombinace, která, pokud je jediným zdrojem diverzity, přestává vytvářet nová řešení v případě, že populace konverguje k jednomu genotypu [6].

Selekce přeživších

Podobně jako u selekce rodičů jsou i u selekce přeživších jedinci vybíráni na základě jejich kvality, avšak až poté, co jsou vytvořeni potomci z vybraných rodičů. Jelikož velikost populace je ve většině případů konstantní, je nutné rozhodnout, kteří jedinci budou figurovat v další generaci. Toto rozhodnutí může záviset nejen na hodnotě fitness, ale může být zohledněn i věk jedinců, což značí, že selekce jedinců do nové generace bývá ve většině případů deterministická [15].

Příkladem jedné z nejběžnějších strategií selekce je sjednocení množiny rodičů a potomků, jejich seřazení na základě hodnoty fitness a výběr nejlepších jedinců pro novou generaci [15].

Ukončovací podmínka

Evoluční proces je ukončen splněním ukončovací podmínky, kterou lze rozdělit do dvou hlavních případů. V prvním případě je evoluční algoritmus ukončen po nalezení jedince, jehož hodnota fitness se shoduje s dříve známou optimální hodnotou. Ve druhém případě, kdy je řešený problém zjednodušen nebo obsahuje šum, je akceptováno řešení, které se dostatečně blíží optimálnímu řešení v rámci požadované přesnosti. Jelikož jsou evoluční algoritmy stochastické, není vždy zaručeno dosažení optimálního řešení, což může vést k nekonečnému evolučnímu procesu. Proto je nutné zavést následující podmínky, jejichž splnění zajistí ukončení evoluce [15]:

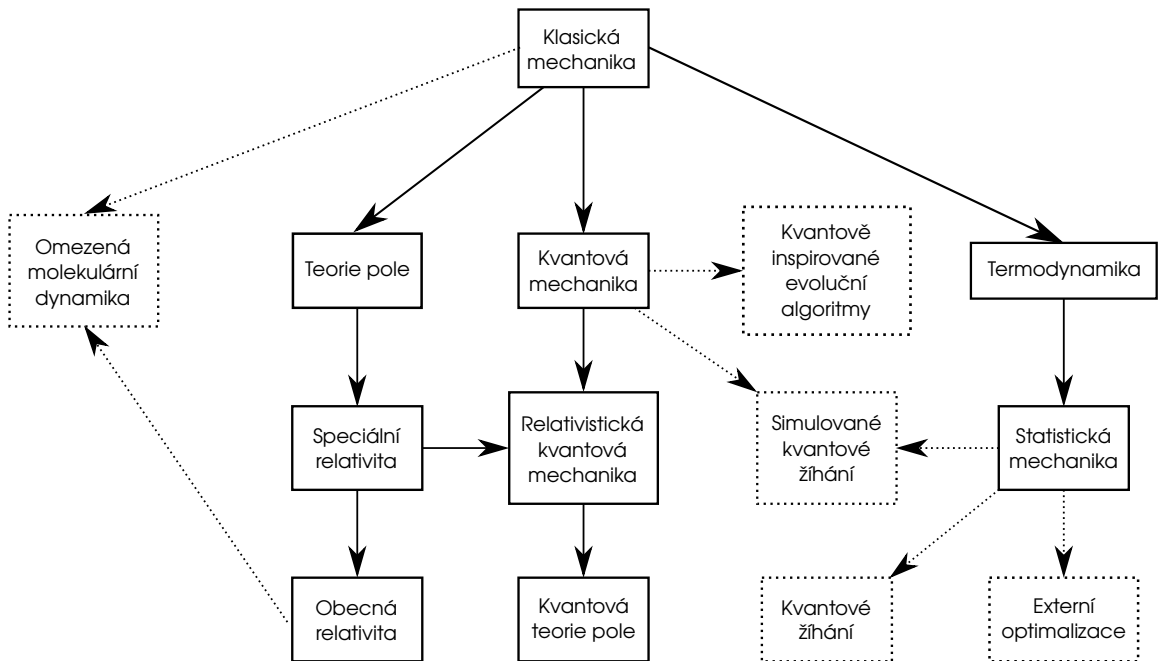
1. Uplyne maximální povolený procesorový čas.
2. Je dosaženo maximálního počtu vyhodnocení fitness funkce.
3. Po určitou dobu nedochází ke zlepšení fitness nad určitou prahovou hodnotu.
4. Diverzita populace klesne pod vybraný práh.

Pokud není známo optimální řešení problému, postačí pro ukončení evolučního algoritmu použít kteroukoli z výše uvedených podmínek [15].

2.3 Základy kvantové fyziky

Tato sekce poskytne stručný popis základních oblastí fyziky a jejich vzájemných souvislostí. Následně se zaměří na podrobnější popis vybrané oblasti, konkrétně kvantové mechaniky, jejíž popis je nutný pro hlubší pochopení kapitoly 3 věnované algoritmům inspirovaným touto oblastí fyziky.

Obrázek 2.6 vyobrazuje základní dělení moderní fyziky na elementární části (nepřerušované obdélníky) společně s jejich vzájemnými vztahy. Nepřerušované šipky ukazují nové fyzikální teorie, které vznikly úpravou těch starších a to buď odstraněním některých z jejich předpokladů nebo přidáním nových principů. Přerušované obdélníky značí nejznámější fyzikálně inspirované algoritmy, přičemž přerušované šipky znázorňují z jakých fyzikálních konceptů tyto algoritmy vycházejí [6].



Obrázek 2.6: Vazby mezi fyzikálními principy a algoritmy. Diagram byl převzat s úpravami z [6].

Jednotlivé základní oblasti fyziky společně s jejich stručným popisem jsou následující [6]:

- **Klasická mechanika (*Classical Mechanics*)**: Podle Galileiho a Newtona působí na tělesa síly, které jsou vnímány jako síly působící okamžitě a na dálku. Euler, Laplace, Lagrange, Hamilton a další vytvořili alternativní formulace pomocí skalární energie, které jsou s původní formulací ekvivalentní.
- **Teorie pole (*Field Theory*)**: Koncept formulovaný Maxwellem a dalšími popisující, jak mohou objekty generovat skalární potencionální pole, které vytváří odpovídající vektorové pole (například gravitační nebo elektromagnetické), jež následně ovlivňuje ostatní objekty. Tato pole mohou přenášet energii prostorem ve formě vln, což omezuje

rychlost šíření signálů, čímž odstraňuje okamžité působení síly na dálku. Tato teorie mimo jiné ukazuje, že se světlo šíří ve vakuu konstantní konečnou rychlostí.

- **Speciální teorie relativity (*Special Theory of Relativity*):** Dílo Einsteina, vycházející z toho, že fyzikální zákony jsou ve všech inerciálních vztažných soustavách stejné, společně s faktem, že rychlost světla ve vakuu je vždy konstantní.
- **Obecná teorie relativity (*General Theory of Relativity*):** Rozšířená speciální teorie relativity samotným Einsteinem, která zavádí princip ekvivalence mezi rovnoměrným gravitačním polem a zrychlením.
- **Kvantová mechanika (*Quantum Mechanics*):** Koncept podle Plancka, Bohra, Schrödingera, Heisenberga, Pauliho, Einsteina a dalších, zavádějící mimo jiné pojem kvantování, který nahrazuje pozorovatelné fyzikální veličiny, jako je například poloha či energie, operátory v Hilbertově prostoru vlnových funkcí.
- **Relativistická kvantová mechanika (*Relativistic Quantum Mechanics*):** Teorie, kde Schrödingerova rovnice pro elektron byla upravena do relativisticky invariantní podoby Diracem tak, že vlnové vektory mají přidáné stupně volnosti, které přesně modelují spin elektronu. Tato teorie předpověděla existenci antihmoty, respektive pozitronu.
- **Kvantová teorie pole (*Quantum Field Theory*):** Dílo Feynmana a dalších, které za pomoci Diracovy práce rozšiřuje relativistickou kvantovou mechaniku tím, že kvantizuje samotné pole včetně vlnové funkce. Tento postup umožňuje interpretovat interakci sil prostřednictvím výměny částic. Praktickým příkladem je Feynmanova kvantová elektrodynamika.
- **Termodynamika (*Thermodynamics*):** Teorie formulovaná Carnotem, Boltzmannem a dalšími, která se zabývá makroskopickými fyzikálními veličinami, jako je například tlak a teplota, u systémů složených z mikroskopických částic. Popisuje mimo jiné koncept termodynamické rovnováhy, termodynamických procesů a entropie.
- **Statistická mechanika (*Statistical Mechanics*):** Moderní pohled na termodynamiku, který popisuje makroskopické veličiny pomocí statistického chování částic.

Následující sekce se podrobněji zaměří na odnož klasické mechaniky, konkrétně na kvantovou mechaniku, která slouží jako inspirace pro kvantově inspirované algoritmy.

2.4 Kvantová mechanika

Model chování přírodních systémů pozorovaných na velmi krátkých časových a délkových měřítkách popisuje kvantová mechanika, která je rozšířením klasické mechaniky. Kvantový systém se může skládat z jedné nebo více částic, jako je například volný elektron či foton [6].

Použitím kvantizace dochází k nahrazení proměnných reprezentujících pozorovatelné fyzikální veličiny, jako jsou poloha, hybnost či energie, lineárními operátory ve vhodném vektorovém prostoru. Tento vektorový prostor nese název Hilbertův prostor (*Hilbert space*) a je definován jako úplný unitární prostor nad komplexními čísly. Jeho prvky jsou funkce časových a prostorových souřadnic, které reprezentují kvantové stavy systému. Kvantový stav systému ψ je někdy rovněž označován jako stavový vektor, vlnová funkce nebo vlnový vektor. Na prvky Hilbertova prostoru působí lineární operátory (*observables*), které

reprezentují pozorovatelné fyzikální veličiny. Vlastní hodnoty těchto operátorů odpovídají možným výsledkům pozorování daných fyzikálních veličin [6].

Následující části stručně popisují klíčové koncepty kvantové mechaniky [6]:

Pozorování v kvantové mechanice

Pozorování kvantového systému vede k určení hodnoty pozorovatelné veličiny, přičemž pravděpodobnost pozorování konkrétní hodnoty odpovídá kvadrátu absolutní hodnoty vlnové funkce ψ . Kvadrát absolutní hodnoty $|\psi|^2$ představuje hustotu pravděpodobnosti, která udává pravděpodobnost nalezení částice na dané pozici v daném čase v prostoru [6].

Časový vývoj vlnové funkce a tedy i hustoty pravděpodobnosti v každém bodě prostoru je popsán lineární Schrödingerovou rovnicí. V čase se kvantový stav vyvíjí deterministicky podle této rovnice, což způsobuje rozptyl hustoty pravděpodobnosti v prostoru. Kvantový stav se před provedením pozorování nachází v superpozici, tedy v lineární kombinaci všech možných vlastních stavů (*eigenstates*) operátoru odpovídajícího pozorované veličině [6].

S postupem času se hustota pravděpodobnosti stále více rozprostírá, což vede ke zvyšování neurčitosti pozice částice. Tento proces pokračuje, dokud není provedeno pozorování, při kterém vlnová funkce nelineárně „kolabuje“ do jednoho z vlastních stavů měřené veličiny. Pravděpodobnost nalezení systému v konkrétním vlastním stavu je dána hodnotou $|\psi|^2$, přičemž pravděpodobnost nalezení částice je nejvyšší v oblastech, kde je hodnota $|\psi|$ největší, a nulová tam, kde $|\psi| = 0$ [6].

Normalizované vlnové funkce splňují podmínku:

$$\int_{\mathbb{R}^3} |\psi(x)|^2 dx = 1.$$

Tato rovnost vyjadřuje, že celková pravděpodobnost nalezení částice v celém prostoru musí být rovna jedné. Z toho plyne, že kvantové stavy leží na povrchu jednotkové koule Hilbertova prostoru, tedy v množině všech vektorů s normou 1 a operátory, které na ně působí, musí být unitární, aby tuto normu zachovávaly [6].

Kvantové provázání

Pojem kvantové provázání (*entanglement*) je jeden z nejdůležitějších rozdílů mezi klasickou a kvantovou fyzikou. V kvantovém systému složeného z více částic existuje pouze jedna vlnová funkce, kterou nelze rozdělit na nezávislé vlnové funkce jednotlivých částic. Podle Schrödingerovy rovnice se tato vlnová funkce vyvíjí v čase do té doby, dokud není provedeno pozorování. Kolaps vlnové funkce celého systému nastává při jeho pozorování a okamžitě ovlivňuje stav všech provázaných částic, které následně přecházejí do jednoho z vlastních stavů měřené veličiny [6].

V klasickém systému složeném z m částic, kde každá může nabývat n různých stavů, roste dimenze prostoru lineárně jako $n \times m$, neboť celkový prostor lze chápat jako m -násobný přímý součet stavových prostorů jednotlivých částic. V kvantovém systému je však správným prostorem tenzorový součin, což způsobuje, že jeho dimenze roste exponenciálně jako n^m [6, 30].

Dekoherence

V případě, že dojde ke kontaktu makroskopického prostředí (například měřící zařízení) s kvantovým systémem, dochází k nevratnému úniku kvantových vlastností do okolí v ter-

modynamicky nevratném procesu. Dekoherece se projevuje tím, že složky kombinované vlnové funkce systému a prostředí přestanou efektivně interferovat. Na makroskopické úrovni tento proces vede k rozpadu superpozic stavů, což se jeví jako zdánlivý kolaps vlnové funkce systému [6].

Nekomutující operátory

Pokud jakékoli dva operátory A a B komutují, platí pro ně vztah $[A, B] = A \times B - B \times A = 0$ ¹. V kvantové mechanice však operátory obecně nekomutují. Příkladem nekomutujících operátorů jsou operátory polohy a hybnosti, jejichž vzájemná nekomutativita vede k Heisenbergově principu neurčitosti. Tento princip říká, že není možné současně měřit hodnoty obou veličin s požadovanou přesností, respektive měření hodnoty jedné veličiny vede ke snížení přesnosti, s jakou lze určit druhou veličinu [6, 35].

Kvantové tunelování

Kvantové tunelování je jev, při kterém může být částice s konečnou, nenulovou pravděpodobností pozorována i za bariérou, přestože standardně by ji neměla být schopna překonat. Očekávaný čas tunelování závisí nejen na šířce bariéry, ale také na její výšce, přičemž čím širší je bariéra, tím nižší je pravděpodobnost, že částice projde touto bariérou. Tento efekt je důsledkem vlnové povahy částice, protože její vlnová funkce je rozprostřena v prostoru, a tudíž je nenulová i na opačné straně bariéry [6].

2.5 Kvantové evoluční počítání

Hlavní rozdíl mezi kvantovými a klasickými výpočetními systémy spočívá v jejich výpočetních schopnostech. Kvantové systémy využívají principy superpozice a provázání, což jim umožňuje efektivněji řešit některé problémy, které jsou pro klasické počítače výpočetně náročné. Existují dva hlavní přístupy pro kvantové počítače [6]:

- **Digitální kvantové počítače:** Oproti klasickým počítačům místo bitů využívají tzv. kvantové bity neboli qubity.
- **Adiabatické kvantové počítače:** Hledají optimální řešení problémů prostřednictvím postupné evoluce kvantového systému.

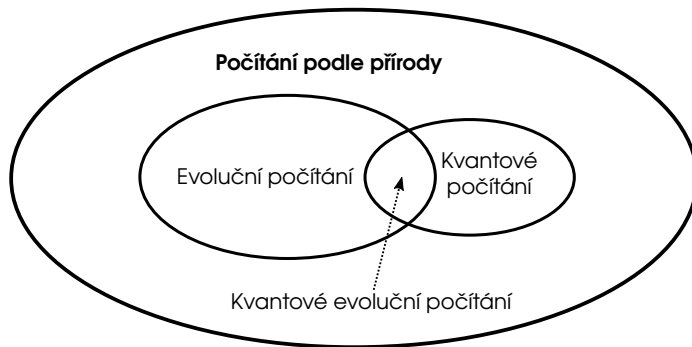
Jelikož kvantové výpočty vykazují výhody při řešení určitých typů optimalizačních úloh, vedly k návrhu kvantově inspirovaných evolučních algoritmů, které jsou zařazeny do kvantové evolučního počítání, viz obrázek 2.7. Z tohoto důvodu bude tato sekce věnována právě těmto kvantovým výpočtům [6].

2.5.1 Kvantový bit

Kvantový bit neboli qubit se na rozdíl od klasického bitu, který může nabývat pouze hodnot 0 nebo 1, nachází v superpozici těchto stavů. Avšak při měření qubitu dojde ke kolapsu jeho superpozice do jednoho ze stavů klasického bitu [22].

Superpozice v kontextu klasické fyziky značí situaci, kdy součtem dvou fyzikálních veličin vznikla odlišná fyzikální veličina. Případem aplikace této superpozice je výpočet celkové

¹Analogie k Poissonovým závorkám v klasické mechanice.



Obrázek 2.7: Kvantové evoluční počítání je kombinací kvantových a evolučních výpočtů, přičemž obě tyto oblasti spadají do výpočtů inspirovaných přírodou. Obrázek byl převzat s úpravami z [30].

velikosti a směru veličiny, jako je například síla či elektrické pole. Na rozdíl od klasické fyziky v kvantové mechanice superpozice značí stav, kdy může být systém současně ve více stavech. Tento systém zůstává ve stavu superpozice do té doby, dokud není provedeno jeho pozorování, při kterém se superpozice zhroutí a systém bude v jedné konkrétní hodnotě [22].

Ve standardním zápisu² je kvantový stav qubitu $|\psi\rangle$ vyjádřen jako superpozice stavů $|0\rangle$ a $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (2.1)$$

kde koeficienty (amplitudy) $\alpha, \beta \in \mathbb{C}$ umožňují matematicky reprezentovat všechny možné superpozice. Tyto koeficienty musí splňovat podmínku normalizace:

$$|\alpha|^2 + |\beta|^2 = 1, \quad (2.2)$$

kde $|\alpha|^2$, respektive $|\beta|^2$, udává pravděpodobnost nalezení částice ve stavu $|0\rangle$, respektive $|1\rangle$, po provedeném měření. Podmínka normalizace zajišťuje, že superpozice zkolabuje s jistotou do jednoho ze stavů $|0\rangle$ nebo $|1\rangle$ [6, 22].

Rovnici 2.1 lze díky platnosti rovnosti 2.2 přepsat do podoby:

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle,$$

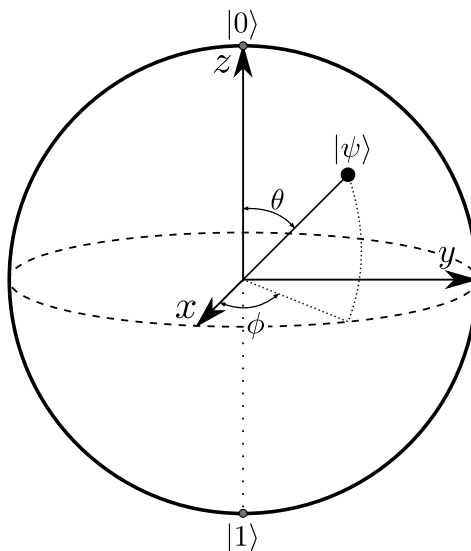
kde hodnoty θ a ϕ určují pozici qubitu na povrchu tzv. Blochovy sféry, viz obrázek 2.8. Tato sféra umožňuje vizualizaci stavu pouze jednoho qubitu. Pro více qubitů již nelze využít tuto geometrickou reprezentaci [22, 30].

2.5.2 Kvantová informace

Princip kvantové provázanosti určuje hlavní rozdíl mezi kvantovým a klasickým bitem, neboť udává silné korelace mezi qubity, což umožňuje pracovat s mnoha stavy současně. Tento jev, známý jako kvantový paralelismus, poskytuje kvantovým systémům výpočetní výhodu oproti klasickým systémům, protože umožňuje existenci stavu v podobě superpozice mnoha klasických stavů [6].

Vlastnosti kvantové informace způsobující problémy při provádění kvantových výpočtů jsou [6]:

²Standardně se využívá Diracova nebo „bra-ket“ notace.



Obrázek 2.8: Blochova sféra reprezentující qubit. Obrázek byl převzat s úpravami z [30].

- **Princip neurčitosti:** V případě nekomutujících pozorovaných veličin ovlivní měření jedné veličiny výsledek měření jiné veličiny.
- **Princip nemožnosti klonování:** Kvantová informace nemůže být dokonale zkopírována, jinak by byl narušen princip neurčitosti.
- **Princip provázanosti:** Kvantová informace je díky své provázanosti rozložena mezi více částí systému, což ji typicky znemožňuje rozdělit na části (jednotlivé qubity). Tento princip je hlavním důvodem, proč obecně nelze efektivně simulovat kvantové systémy na těch klasických.
- **Dekoherence:** Při interakci informací kvantového systému s informacemi z okolního prostředí dochází k jejich vzájemnému provázání, což snižuje schopnost obnovit původní kvantovou informaci, neboť většina provázaných informací pochází z okolního prostředí.

2.5.3 Digitální kvantové počítače

Digitální kvantové počítače pracují s qubity a manipulují s nimi pomocí kvantových hradel (kvantový protějšek logických hradel). V průběhu výpočtu jsou tyto hradla aplikovány na systém qubitů a měření qubitů je provedeno až na konci samotného výpočtu. Paul Benioff uvažoval Turingův stroj pracující s ekvivalentem qubitů a později Richard Feynman dokázal, že klasické počítačové systémy nejsou schopny efektivně simulovat kvantové systémy, zejména kvantové provázání, protože by jejich simulace vyžadovala exponenciální časovou a paměťovou složitost, respektive $\mathcal{O}(2^n)$ [6, 34].

Kvantové počítače prokazatelně umožňují oproti těm klasickým řešit některé druhy problémů efektivněji. Příklady algoritmů, které jsou schopné řešit problém v kvantovém počítači efektivněji, jsou [6]:

- **Shorův algoritmus:** Umožňuje faktorizaci velkých čísel v časové složitosti $\mathcal{O}(n^2)$ místo $\mathcal{O}(e^{\sqrt[3]{n}})$, jako je tomu u nejlepšího známého algoritmu na klasických systémech.

- **Groverův algoritmus:** Zrychluje vyhledávání v nestrukturovaných databázích klasických systémů z časové složitosti $\mathcal{O}(n)$ na $\mathcal{O}(\sqrt{n})$.

Mimo jiné umožňují kvantové počítače simulovat samy sebe, přičemž však stále nebylo prokázáno zda dokáží řešit NP-těžké problémy v polynomiálním čase³ [6].

2.5.4 Adiabatické kvantové počítače

Oproti digitálním kvantovým počítačům se adiabatické kvantové počítače neskládají z qubitů. Místo toho využívají postupnou evoluci kvantového systému v čase. Přístup je podobný analogovým počítačům, kde je vytvořen fyzikální systém odpovídající řešenému problému a jeho stav je ponechán, aby se vyvíjel v čase. Jejich výpočetní síla je ekvivalentní výpočetní síle kvantových počítačů založených na qubitech, což znamená, že dokáží řešit stejné problémy se srovnatelnou polynomiální složitostí [6].

Adiabatické kvantové počítače splňují kvantový adiabatický teorém, jehož hlavní myšlenkou je, že pokud jsou na systém aplikovány vnější jevy dostatečně pomalu, zůstává ve svém relativním vlastním stavu. Pokud jsou však změny vnějších jevů aplikovány na systém příliš rychle, stav systému se nezmění, jelikož nemá dostatek času, aby se přizpůsobil a skončí v superpozici stavů [6].

Princip evoluce v adiabatických kvantových počítačích je hlavní myšlenkou adiabatického kvantového počítání (*Adiabatic Quantum Computing – AQC*). Tento princip svým charakterem připomíná tepelné žíhání kovů, a proto se rovněž označuje jako kvantové žíhání (*Quantum Annealing – QA*) [6].

³Obecně se věří, že nedokážou řešit NP-těžké problémy v polynomiálním čase.

Kapitola 3

Kvantově inspirované evoluční algoritmy

Evoluční algoritmy, které vycházejí z principů kvantové mechaniky, se souhrnně označují jako kvantově inspirované evoluční algoritmy (*Quantum Inspired Evolutionary Algorithms – QIEA*) a jsou navrženy především pro řešení optimalizačních problémů. Není však možné, aby tyto algoritmy využívaly všechny náležitosti vyplývající z kvantové mechaniky. Jedná se zejména o kvantové provázání, které nelze efektivně simulovat na klasických počítačích. Nicméně použití kvantově inspirované reprezentace v evolučních algoritmech se v některých případech ukázalo jako vhodný kompromis mezi průzkumem a exploatací, přičemž často umožňuje pracovat s menší populací jedinců, než by bylo nutné u klasických evolučních algoritmů [6].

Tato kapitola nejdříve popíše, jak lze reprezentovat jedince v populaci, a následně se zaměří na dva způsoby transformace těchto jedinců pomocí kvantového rotačního hradla a kvantové mutace. Poté charakterizuje následující kvantově inspirované evoluční algoritmy:

- kvantově inspirovaný genetický algoritmus,
- kvantově inspirované simulované žíhání,
- kvantová evoluce roje a
- kvantově inspirovaná optimalizace rojem částic.

V poslední řadě jsou souhrnně vylíčeny zajímavé studie popisující situace, kde byly *QIEA* využity.

3.1 Kódování řešení v kvantově inspirovaných evolučních algoritmech

Jeden z možných způsobů reprezentace jedince v populaci v kvantově inspirovaných evolučních algoritmech vychází z konceptu kvantového bitu. Binární kvantově inspirované evoluční algoritmy využívají qubity k reprezentaci řešení, přičemž manipulace s nimi probíhá prostřednictvím kvantově inspirovaných operátorů [6].

V binárním kvantově inspirovaném evolučním algoritmu je qubit popsán dvojicí koeficientů α a β , přičemž systém sestávající z m qubitů lze zapsat v podobě matice jako:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix}, \quad (3.1)$$

kde $\alpha_i, \beta_i \in \mathbb{R}$ a přičemž pro normalizovaný systém musí platit:

$$\forall i \in \{1, 2, \dots, m\} : \alpha_i^2 + \beta_i^2 = 1. \quad (3.2)$$

Tento způsob umožňuje efektivní zápis qubitů, ale je však důležité mít na paměti, že kvantový systém o m qubitech dokáže současně reprezentovat všechny bitové řetězce o délce 2^m bitů, zatímco klasický bitový registr umožňuje reprezentovat pouze jeden z 2^m možných stavů [6].

Binární kvantově inspirované evoluční algoritmy využívají kvantovou reprezentaci, vizte matici 3.1, k vyjádření pravděpodobnosti výsledného řešení. V případě splnění podmínky normalizace z rovnice 3.2, hodnota α_i^2 , respektive β_i^2 , určuje s jakou pravděpodobností bude ve výsledném řetězci na i -té pozici binární hodnota 0, respektive 1, s ohledem na rovnici 2.1.

3.2 Operátory v kvantově inspirovaných algoritmech

Pro generování diverzity populace se běžně využívají dva přístupy, konkrétně kvantové rotační hradlo a kvantová mutace, jenž jsou popsány níže, přičemž se nevyužívají standardní operátory křížení a mutace, jak je tomu u klasických evolučních algoritmů [6].

3.2.1 Kvantové rotační hradlo

V kvantových systémech se s qubity manipuluje pomocí kvantových hradel (Hadamardovo, CNOT, Pauli-X, aj.). Tato hradla umožňují provádět paralelní výpočet nad všemi qubity najednou bez změření jejich hodnoty, přičemž jejich výstupem je nová superpozice stavů systému [6, 22, 30].

V kvantově inspirovaných evolučních algoritmech našla kvantová hradla uplatnění pro rotační hradlo $R()$ popsané následující maticí:

$$R(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix}, \quad (3.3)$$

kde $\Delta\theta_i$ je úhel rotace, kterým je upravován stav i -tého qubitu. Pravděpodobnostní koeficienty α_i a β_i i -tých qubitů v chromozomu jsou modifikovány na nové koeficienty α'_i a β'_i pomocí kvantového rotačního hradla $R()$ podle vzorce:

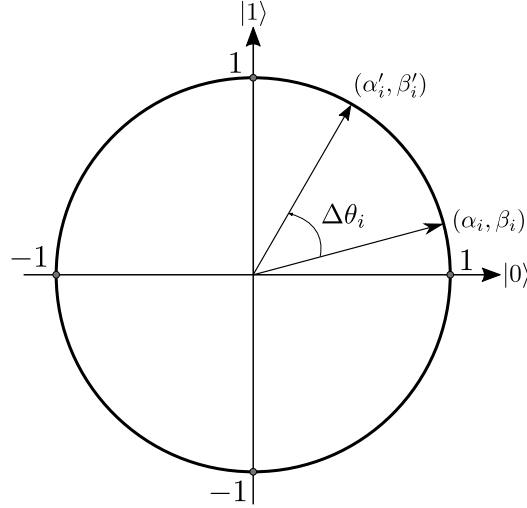
$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = R(\Delta\theta_i) \cdot \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}, \quad (3.4)$$

přičemž výsledné nové koeficienty musí splňovat podmínku normalizace z rovnice 2.2. Tato podmínka je vždy splněna, jelikož kvantové rotační hradlo odpovídá unitární matici. Unitární vlastnost rotačního hradla zaručuje, že součet pravděpodobností bitového stavu 0 a 1 po pozorování zůstává roven 1. Výsledný stav qubitu se tedy po aplikaci hradla vždy nachází na jednotkové kružnici, vizte obrázek 3.1 [6].

Aby mutace postupně přizpůsobovala hodnoty kvantového chromozomu směrem k nejlepšímu nalezenému jedinci v aktuální pozorované populaci, lze využít různé přístupy. Jeden z možných způsobů spočívá ve využití tabulky 3.1, kde postup pro výběr hodnoty parametru $\Delta\theta_i$ je následující [6]:

1. Pomocí aktuálního kvantového chromozomu $q = (q_1 \ q_2 \ \dots \ q_m)$ složeného z kvantových bitů $q_i = [\alpha_i, \beta_i]^T$ pro $i = 1, 2, \dots, m$, respektive

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix},$$



Obrázek 3.1: Kvantové rotační hradlo. Obrázek byl převzat s úpravami z [6].

je vytvořen binární chromozom $x = (x_1 \ x_2 \ \dots \ x_m)$. Hodnota x_i je určena na základě qubitu q_i , respektive jeho parametru α_i , následujícím způsobem:

$$x_i = \begin{cases} 0 & \text{pokud } \alpha_i^2 \leq 0,5, \\ 1 & \text{jinak.} \end{cases}$$

2. Z populace je vybráno nejlepší binární řešení $b = (b_1 \ b_2 \ \dots \ b_m)$.
3. Na základě hodnot x_i a b_i je z vyhledávací tabulky 3.1 vybrána hodnota $\Delta\theta_i$ následovně:

x_i	b_i	$\Delta\theta_i$
1	1	0
0	1	δ
0	0	0
1	0	$-\delta$

Tabulka 3.1: Vyhledávací tabulka pro parametr $\Delta\theta_i$.

- Pokud $x_i = b_i$, pak $\Delta\theta_i = 0$, respektive parametry α_i a β_i qubitu q_i zůstanou zachovány.
- Pokud $x_i = 1 \wedge b_i = 0$, pak $\Delta\theta_i = -\delta$, respektive dojde ke snížení pravděpodobnosti pozorování binární hodnoty 1 na pozici i chromozomu q .
- Pokud $x_i = 0 \wedge b_i = 1$, pak $\Delta\theta_i = \delta$, respektive dojde ke zvýšení pravděpodobnosti pozorování binární hodnoty 1 na pozici i chromozomu q .

Tabulka 3.1 nebere v úvahu kvadranty z obrázku 3.1, v nichž koeficienty α_i nebo β_i mohou být záporné, tudíž je možné upravit rovnici 3.4 tak, aby aktualizovala i -tý kvantový bit $\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$ následovně:

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = R(\xi(\Delta\theta_i)) \cdot \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}, \quad (3.5)$$

kde $\xi(\Delta\theta_i) = s(\alpha_i, \beta_i) \cdot \Delta\theta_i$, přičemž $s(\alpha_i, \beta_i)$ a $\Delta\theta_i$ určují směr rotace a úhel, vizte vyhledávací tabulku 3.2.

x_i	b_i	$f(x) > f(b)$	$\Delta\theta_i$	$s(\alpha_i, \beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	Nepravda	0	0	0	0	0
0	0	Pravda	0	0	0	0	0
0	1	Nepravda	δ	+1	-1	0	± 1
0	1	Pravda	δ	-1	+1	± 1	0
1	0	Nepravda	δ	-1	+1	± 1	0
1	0	Pravda	δ	+1	-1	0	± 1
1	1	Nepravda	0	0	0	0	0
1	1	Pravda	0	0	0	0	0

Tabulka 3.2: Vyhledávací tabulka pro parametr $\Delta\theta_i$ a $s(\alpha_i, \beta_i)$, kde je parametr δ obvykle nastaven na malou hodnotu (běžně 0.01π).

V kvantově inspirovaných evolučních algoritmech se v současné době nevyužívají komplexní koeficienty z kvantové mechaniky, ale výlučně se používají reálné koeficienty, což vede na reprezentaci kvantového stavu pomocí jednotkové kružnice namísto Blochovy sféry, jak je zřejmé z obrázku 3.1 [6].

3.2.2 Kvantová mutace

Kvantová mutace je inspirována mutací ze standardních evolučních algoritmů v podobě [6]:

$$Q^*(t) = a \cdot B_{\text{best}}(t) + (1 - a) \cdot (1 - B_{\text{best}}(t))$$

$$Q(t+1) = Q^*(t) + b \cdot r,$$

kde

- $B_{\text{best}}(t)$ reprezentuje nejlepší nalezené řešení v iteraci t ,
- $Q^*(t)$ je dočasný kvantový chromozom,
- r je náhodné číslo pocházející z normálního rozdělení $N(0, 1)$,
- a a b jsou parametry řídící poměr průzkumu a exploatace.

Případně existují i další možné metody pro mutaci kvantového chromozomu [6].

3.3 Kvantově inspirovaný genetický algoritmus

Kanonický příklad kvantově inspirovaného genetického algoritmu (*Quantum Inspired Genetic Algorithm – QIGA*), jenž je principiálně podobný ostatním evolučním algoritmům, je popsán algoritmem 2, kde parametr t_{max} udává maximální počet iterací (generací) algoritmu a parametr t označuje aktuálně prováděnou iteraci [6].

Uvažujme počáteční generaci $t = 0$, kdy algoritmus začíná inicializací počáteční populace $Q(t)$ čítající n kvantových chromozomů:

$$Q(t) = \{q_1(t), q_2(t), \dots, q_n(t)\}, \quad (3.6)$$

Algoritmus 2: Kvantově inspirovaný genetický algoritmus [6]

```
1  $t \leftarrow 0$ ;  
2 Inicializace populace  $Q(t)$  kvantových chromozomů;  
3 Vytvoření množiny řešení  $P(t)$  pozorováním  $Q(t)$ ;  
4 Ohodnocení  $P(t)$  a vybrání nejlepšího řešení z populace;  
5 Uložení nejlepšího řešení do  $B(t)$ ;  
6 while  $t < t_{max}$  do  
7    $t \leftarrow t + 1$ ;  
8   Vytvoření  $P(t)$  pozorováním  $Q(t-1)$ ;  
9   Ohodnocení populace  $P(t)$ ;  
10  if Nejlepší řešení z  $P(t) > B(t-1)$  then  
11    | Uložení nejlepších řešení z  $P(t)$  do  $B(t)$ ;  
12  else  
13    | Uložení  $B(t-1)$  do  $B(t)$ ;  
14  Vytvoření  $Q(t)$  pomocí  $B(t)$ ;
```

přičemž každý z chromozomů $q_j(t)$ pro $j = 1, 2, \dots, n$ je tvořen m kvantovými bity $\psi_{ji}(t)$ složenými z koeficientů α a β jako:

$$q_j(t) = [\psi_{j1}(t) \quad \psi_{j2}(t) \quad \dots \quad \psi_{jm}(t)] = \begin{bmatrix} \alpha_{j1}(t) & \alpha_{j2}(t) & \dots & \alpha_{jm}(t) \\ \beta_{j1}(t) & \beta_{j2}(t) & \dots & \beta_{jm}(t) \end{bmatrix},$$

kde jsou koeficienty $\alpha_{ji}(t)$ a $\beta_{ji}(t)$ pro $i = 1, 2, \dots, m$ v každém kvantovém bitu $q_j(t)$ nastaveny na běžně používanou hodnotu $\frac{1}{\sqrt{2}}$. Respektive počáteční pravděpodobnost výsledného stavu 0 nebo 1 je po provedeném pozorování rovna $\left(\frac{1}{\sqrt{2}}\right)^2 = 0,5$. Případně mohou být koeficienty α a β nastaveny na libovolné hodnoty, které lépe vyhovují řešenímu problému, avšak stále musí splňovat podmínku normalizace danou rovnicí 2.2 [6, 20].

Nad vytvořenou počáteční populací $Q(t)$ může být provedeno pozorování, čímž dojde k vygenerování množiny řešení:

$$P(t) = \{p_1(t), p_2(t), \dots, p_n(t)\}, \quad (3.7)$$

kde je každé řešení p_j pro $j = 1, 2, \dots, n$ reprezentováno binárním řetězcem složeného z m bitů:

$$p_j(t) = (x_{j1}(t) \quad x_{j2}(t) \quad \dots \quad x_{jm}(t)).$$

Při jedné z možných metod generování množiny řešení $P(t)$ je na i -té pozici j -tého bitu x_{ji} nastavena hodnota 1 v případě, že náhodně vygenerované číslo $r_i \in \langle 0, 1 \rangle$ splňuje podmínku $r_i > |\alpha_i(t)|^2$, jinak je na tuto pozici nastavena hodnota 0 [6, 20].

Vygenerování množiny řešení $P(t)$ může být provedeno několika způsoby [6]:

- Použitím jednoho kvantového chromozomu, kdy je množina řešení $P(t)$ vygenerována pomocí n -násobného pozorování tohoto jediného chromozomu.
- Použitím malé populace kvantových chromozomů, kdy je každý z nich pozorován tolikrát, aby celkový počet pozorování odpovídal velikosti výsledné množiny řešení $P(t)$.

- Použitím stejného počtu kvantových chromozomů jako je velikost populace $P(t)$.

Posléze je každé binární řešení $p_j(t)$ ohodnoceno fitness funkcí. Nejlepší řešení $p_k(t)$ je vybráno na základě tohoto ohodnocení a použito k určení aktuálně nejlepšího řešení:

$$B(t) = (b_1(t) \ b_2(t) \ \dots \ b_m(t)),$$

kde $b_i(t) = x_{k_i}(t)$ pro $i = 1, 2, \dots, m$ [6, 20].

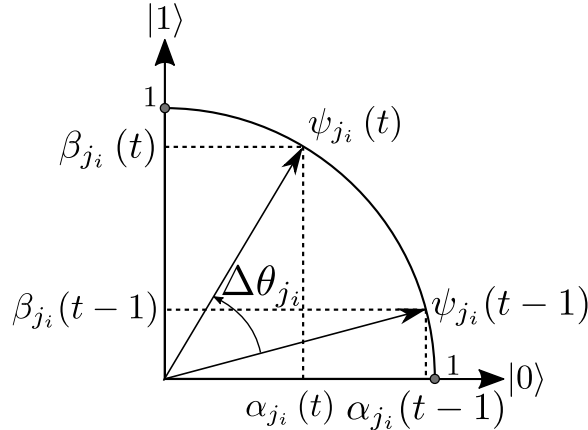
Uvažujme generaci t pro $t > 0 \wedge t < t_{\max}$ v rámci smyčky *QIGA*. V iteraci t je vytvořena množina řešení $P(t)$ pozorováním předchozí populace $Q(t-1)$, jenž je následně ohodnocena fitness funkcí. Nejlepší řešení napříč $P(t)$ a $B(t-1)$ je uloženo do $B(t)$. Hlavní částí smyčky algoritmu je vytvoření populace $Q(t)$ na základě $B(t)$ s cílem její evoluce, která může být provedena několika způsoby a to například [6]:

- použitím některé z variant kvantové mutace, vizte podsekcí 3.2.2, nebo
- aplikací kvantových rotačních hradel, vizte podsekcí 3.2.1 a popis níže.

Uvažujme i -tý kvantový bit $\psi_{j_i}(t-1) = \begin{bmatrix} \alpha_{j_i}(t-1) \\ \beta_{j_i}(t-1) \end{bmatrix}$ j -tého jedince. Nová hodnota kvantového bitu $\psi_{j_i}(t)$ je spočtena pomocí kvantového rotačního hradla $R()$ následujícím způsobem:

$$\psi_{j_i}(t) = \begin{bmatrix} \alpha_{j_i}(t) \\ \beta_{j_i}(t) \end{bmatrix} = R(\Delta\theta_{j_i}) \cdot \begin{bmatrix} \alpha_{j_i}(t-1) \\ \beta_{j_i}(t-1) \end{bmatrix},$$

kde $\alpha_{j_i}(t)$ a $\beta_{j_i}(t)$ jsou nové pravděpodobnostní koeficienty aktualizovaného kvantového bitu. Proces je znázorněn na obrázku 3.2.



Obrázek 3.2: Znázornění principu kvantové rotace, kde je rotací o úhel $\Delta\theta_{j_i}$ počáteční stav $\psi_{j_i}(t-1) = \begin{bmatrix} \alpha_{j_i}(t-1) \\ \beta_{j_i}(t-1) \end{bmatrix}$ aktualizován na nový stav $\psi_{j_i}(t) = \begin{bmatrix} \alpha_{j_i}(t) \\ \beta_{j_i}(t) \end{bmatrix}$. Obrázek převzat s úpravami z [12].

Kvantové chromozomy jsou v tomto aktualizacním kroku upraveny tak, aby v následující generaci docházelo k pravděpodobnějším generování dosud nejlepšího nalezeného řešení. Iterace evolučního procesu se opakují, dokud není splněna ukončující podmínka, přičemž se algoritmus postupně blíží k nejlepšímu nalezenému řešení, jednotlivé qubity kvantového chromozomu konvergují k hodnotě 0 nebo 1 [6, 20].

3.4 Kvantově inspirované simulované žíhání

Tato sekce se zaměří na představení kvantově inspirovaného simulovaného žíhání (*Quantum Inspired Simulated Annealing – QISA*), což je algoritmus, jenž kombinuje principy klasického simulovaného žíhání (*Simulated Annealing – SA*) a kvantového počítání. Algoritmus *QISA*, znázorněný algoritmem 3, vychází principiálně z algoritmu *QIGA* ze sekce 3.3, jenž je rozšířen o principy *SA* a jehož klasické pozorování kvantové populace je upraveno na tepelně-řízené pozorování (*heated observation*). Obdobně jako v pseudokódu algoritmu 2 *QIGA* proměnná t reprezentuje aktuálně prováděnou iteraci a hodnota t_{\max} určuje maximální počet generací evolučního procesu [12].

Algoritmus 3: Kvantově inspirované simulované žíhání [12]

```

1 Inicializace počáteční teploty  $T_0$ ;
2 Selektce chladicího plánu;
3  $t \leftarrow 0$ ;
4 Inicializace populace  $Q(t)$  kvantových chromozomů;
5 Vytvoření množiny řešení  $P(t)$  tepelně-řízeným pozorováním  $Q(t)$ ;
6 Získání energie  $E(t)$  populace  $P(t)$ ;
7 while  $t < t_{\max}$  do
8    $t \leftarrow t + 1$ ;
9   Vytvoření množiny řešení  $P(t)$  tepelně-řízeným pozorováním  $Q(t-1)$ ;
10  Získání energie  $E_P(t)$  množiny řešení  $P(t)$ ;
11  if  $E_P(t) \leq E(t-1)$  then
12    Vytvoření  $Q(t)$  na základě  $\begin{bmatrix} 1 - P(t) \\ P(t) \end{bmatrix}$ ;
13    Vytvoření  $E(t)$  na základě  $E_P(t)$ ;
14  else
15    Vytvoření  $Q(t)$  na základě  $R_Q(Q(t-1), P(t))$ ;
16    Vytvoření  $E(t)$  na základě  $U_E(E(t-1), E_P(t))$ ;
17  Aktualizace  $T_t$  pomocí chladicího plánu;
```

Na začátku algoritmu je inicializována počáteční teplota T_0 pomocí standardní odchylky následovně:

$$T_0 = \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}, \quad (3.8)$$

kde

- x_i je i -té náhodně vygenerované řešení,
- \bar{x} je průměr vygenerovaných řešení a
- N je počet vygenerovaných řešení.

Tato inicializace teploty T_0 je vhodná pro řešení numerických optimalizačních problémů, kde jsou řešením reálné hodnoty, přičemž se standardně při výpočtu počáteční teploty uvažuje 1000 náhodně vygenerovaných řešení, respektive $N = 1000$ [12, 18].

Následuje výběr chladicího plánu. Příkladem takového plánu je exponenciální chladicí plán, jenž je definován jako:

$$T(t) = T_0 \cdot a^t,$$

kde T_0 je počáteční teplota, a je chladicí koeficient a t je chladicí krok (iterace) [12].

Uvažujme $t = 0$. Po výpočtu počáteční teploty T_0 a výběru chladicího plánu je vygenerována počáteční populace $Q(t)$, vizte rovnici 3.6, jež je pozorována pomocí tepelně-řízeného pozorování. Necht $q_j(t)$ je aktuálně pozorovaný kvantový chromozom, pak jsou pravděpodobnosti pozorování binárních stavů i -tého kvantového bitu dány jako [12]:

$$\begin{aligned} \left| \alpha_{j_i}^h(t) \right|^2 &= |\alpha_{j_i}(t)|^2 + \left(0,5 - |\alpha_{j_i}(t)|^2\right) \cdot h(T_t) \\ \left| \beta_{j_i}^h(t) \right|^2 &= |\beta_{j_i}(t)|^2 + \left(0,5 - |\beta_{j_i}(t)|^2\right) \cdot h(T_t), \end{aligned}$$

kde

- $\left| \alpha_{j_i}^h(t) \right|^2$ je pravděpodobnost pozorování stavu 0,
- $\left| \beta_{j_i}^h(t) \right|^2$ je pravděpodobnost pozorování stavu 1 a
- $h(T_t)$ pro $0 \leq h(T_t) \leq 1$ je zahřívací funkce (*heating function*).

Na rozdíl od klasického pozorování, kde je náhodně vygenerovaná hodnota r_i porovnávána s $|\alpha_{j_i}(t)|^2$ nebo s $|\beta_{j_i}(t)|^2$, u tepelně-řízeného pozorování se porovnává hodnota r_i s $\left| \alpha_{j_i}^h(t) \right|^2$ nebo s $\left| \beta_{j_i}^h(t) \right|^2$. Tyto pravděpodobnosti vycházejí z klasických koeficientů kvantového bitu, jež jsou navíc upraveny tepelným procesem. Tento proces zajišťuje vyrovnaní pravděpodobností pro pozorování stavu 0 a 1 u každého kvantového bitu, čímž nedochází k rychlé konvergenci k jednomu stavu a tudíž nejsou jednotlivé kvantové bity vázány na jeden konkrétní stav. Síla tohoto zahřívání určuje, jak moc budou pozorované bity závislé na konkrétním stavu, respektive když $h(T) = 0$, jsou pozorované kvantové bity zcela určeny aktuálním řešením, zatímco když $h(T) = 1$, je výsledek pozorování zcela náhodný [12].

Uvažujme dvě zahřívací metody. První z nich se nazývá konstantní zahřívání a má tvar:

$$h(T) = w, \tag{3.9}$$

kde w je konstanta. Druhá metoda je tzv. sigmoidní funkce, která má tvar:

$$h(T) = \frac{w_3}{1 + e^{-w_1\left(\frac{T}{T_0} - w_2\right)}} + w_4, \tag{3.10}$$

kde w_1 , w_2 , w_3 a w_4 jsou kladné kontrolní parametry a T_0 je počáteční teplota. Princip sigmoidního zahřívání spočívá v tom, že při vyšších teplotách je zahřívání silnější, zatímco při nižších teplotách je slabší, což znamená, že na počátku evoluce bude mít pozorované řešení větší závislost na aktuálním kvantovém bitu, což pomáhá globálnímu prohledávání. Naopak na konci evoluce bude závislost menší, a tudíž se prohledávání více zaměří na lokální oblasti [12].

Po prvním provedeném pozorování je z počáteční množiny řešení $P(t)$, vizte rovnici 3.7, získána prvotní energie řešení ve tvaru:

$$E(t) = \{e_1(t), e_2(t), \dots, e_n(t)\},$$

kde $e_j(t) = f(p_j(t))$ pro $j = 1, 2, \dots, n$, přičemž $p_j(t)$ je řešení získané tepelně-řízeným pozorováním a $f()$ je fitness funkce [12].

Uvažujme generaci t pro $t > 0 \wedge t < t_{\max}$ v rámci smyčky *QISA*. V iteraci t je vytvořena množina řešení $P(t)$ tepelně-řízeným pozorováním populace $Q(t-1)$, ze které je následně získána energie $E_P(t)$ [12].

Pokud je splněna akceptační podmínka

$$e_j^P(t) \leq e_j(t-1), \quad (3.11)$$

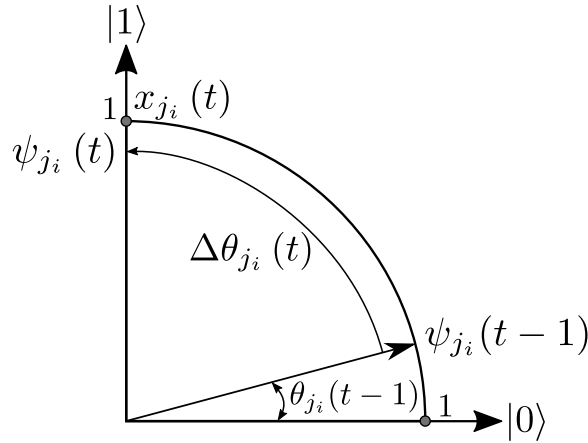
kde $j = 1, 2, \dots, n$ pak:

- je vytvořena kvantová populace $Q(t)$ jedinců jako $\psi_{j_i}(t) = \begin{bmatrix} 1 - x_{j_i}(t-1) \\ x_{j_i}(t-1) \end{bmatrix}$ pro $i = 1, 2, \dots, m$, vizte obrázek 3.3, a následně
- je vytvořena energie $E(t)$ jako $e_j(t) = e_j(t-1)$.

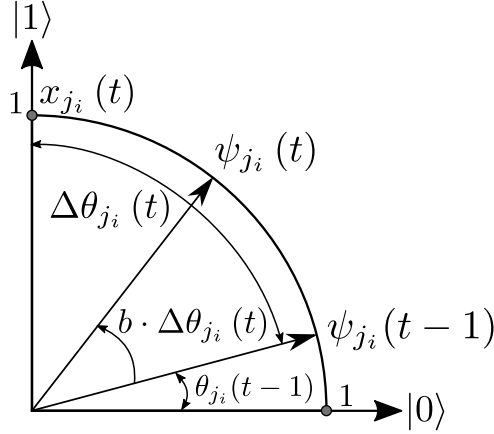
jinak

- je vytvořena kvantová populace $Q(t)$ pomocí $R_Q(Q(t-1), P(t))$, vizte obrázek 3.4, a následně
- je vytvořena energie $E(t)$ jako $U_E(E(t-1), E_P(t))$,

kde $R_Q()$ je rotační funkce, vizte podsekcí 3.4.1, a $U_E()$ je funkce pro aktualizaci energie, vizte podsekcí 3.4.2. Po provedené aktualizaci kvantové populace a energie je aktualizována teplota T_t v závislosti na vybraném chladicím plánu. Iterace evolučního procesu *QISA* jsou opakovány, dokud není splněna ukončující podmínka [12].



Obrázek 3.3: Pokud $e_j^P(t) \leq e_j(t-1)$ pro $j = 1, 2, \dots, n$, kde uvažujeme i -tý kvantový bit $\psi_{j_i}(t-1)$ j -tého jedince a binární hodnotu $x_{j_i}(t)$, pak je kvantový bitu $\psi_{j_i}(t)$ nastaven na novou hodnotu $\begin{bmatrix} 1 - x_{j_i}(t) \\ x_{j_i}(t) \end{bmatrix}$. Obrázek převzat s úpravami z [12].



Obrázek 3.4: Pokud $e_j^P(t) > e_j(t-1)$ pro $j = 1, 2, \dots, n$, kde uvažujeme i -tý kvantový bit $\psi_{j_i}(t-1)$ j -tého jedince a binární hodnotu $x_{j_i}(t)$, pak je nový kvantový bit $\psi_{j_i}(t)$ získán aktualizací koeficientů o úhel $b \cdot \Delta\theta_{j_i}(t)$ směrem k $x_{j_i}(t)$ pomocí kvantového rotačního hradla. Obrázek převzat s úpravami z [12].

3.4.1 Rotační funkce

Rotační funkce $R_Q()$ odpovídá kvantovému rotačnímu hradlu, vizte podsekcí 3.2.1. Uvažujeme i -tý kvantový bit $\psi_{j_i}(t-1) = \begin{bmatrix} \alpha_{j_i}(t-1) \\ \beta_{j_i}(t-1) \end{bmatrix}$ j -tého jedince $q_j(t-1)$ kvantové populace $Q(t-1)$. Nechť $\Delta\theta_{j_i}(t)$ je úhel mezi $\psi_{j_i}(t-1)$ a $x_{j_i}(t)$, pak jsou koeficienty $\psi_{j_i}(t-1)$ upraveny pomocí kvantového rotačního hradla o úhel $b \cdot \Delta\theta_{j_i}(t)$ směrem k $x_{j_i}(t)$. Faktor b splňující $0 \leq b \leq 1$ je definován jako:

$$b = e^{\left(\frac{e_j(t-1) - e_j^P(t)}{T_t} \right)}, \quad (3.12)$$

kde $e_j^P(t)$ je energie j -tého prvku množiny řešení $P(t)$ a T_t je aktuální teplota. Výsledný úhel $\Delta\theta_j(t)$ pro rotaci je spočten podle vzorce:

$$\Delta_{j_i}(t) = b \cdot \Delta\theta_{j_i}(t) = b \cdot \left(\frac{\pi}{2} \cdot p_{j_i}(t) - \theta_{j_i}(t-1) \right),$$

kde $\theta_{j_i}(t-1) = \arctan\left(\frac{\beta_{j_i}(t-1)}{\alpha_{j_i}(t-1)}\right)$. Spočítaný úhel je následně aplikován na kvantové rotační hradlo $R()$:

$$\begin{bmatrix} \alpha_{j_i}(t) \\ \beta_{j_i}(t) \end{bmatrix} = R(\Delta_{j_i}(t)) \begin{bmatrix} \alpha_{j_i}(t-1) \\ \beta_{j_i}(t-1) \end{bmatrix},$$

kde $\alpha_{j_i}(t)$ a $\beta_{j_i}(t)$ jsou nové koeficienty kvantového bitu $\psi_{j_i}(t)$ [12].

3.4.2 Funkce pro aktualizaci energie

Funkce pro aktualizaci energie $U_E()$ není nutná v případě, že platí $e_j^P(t) \leq e_j(t-1)$, jelikož je nová energie $e_j(t)$ jedince nastavena na energii $e_j(t-1)$. Pokud však platí $e_j^P(t) > e_j(t-1)$, měla by být energie $e_j(t)$ nastavena na hodnotu energie odpovídající novému

stavu $q_j(t)$. Přesná energie tohoto stavu je dána jako vážený součet energií všech 2^n pozorovaných stavů jedince, a to následovně:

$$E_Q = \sum_{i=0}^{2^n-1} p_i \cdot e_{j_i}$$

kde

- e_{j_i} je energie i -tého pozorovaného stavu j -tého jedince a
- p_i je pravděpodobnost s jakou je pozorován i -tý stav,

Tento výpočet je pro klasické výpočetní systémy náročný, neboť jeho výpočetní složitost roste exponenciálně s počtem kvantových bitů. Proto se pro výpočet energie nového řešení využívá aproximace, která je vyjádřena následující rovnicí:

$$e_j(t) = U_E(e_j(t-1), e_j^P(t)) = \cos^2\left(b \cdot \frac{\pi}{2}\right) \cdot e_j(t-1) + \sin^2\left(b \cdot \frac{\pi}{2}\right) \cdot e_j^P(t),$$

kde b je faktor definován rovnicí 3.12. Přestože tato aproximace není zcela přesná, poskytuje efektivní způsob, jak algoritmu umožnit opustit lokální optima [12].

3.5 Kvantová evoluce roje

Tato sekce pojednává o kvantové evoluci roje (*Quantum Swarm Evolutionary – QSE*), což je algoritmus kombinující principy kvantového počítání a částicového systému (*Particle Swarm Optimization*) a jež je znázorněn na algoritmu 4. Obdobně jako u algoritmů *QIGA* a *QISA*, proměnná t reprezentuje aktuální iteraci a hodnota t_{\max} určuje maximální počet iterací evolučního procesu.

Algoritmus 4: Kvantová evoluce roje [44]

```

1  $t \leftarrow 0$ ;
2 Inicializace populace  $Q(t)$  kvantových chromozomů;
3 Inicializace počátečních rychlostí  $V(t)$ ;
4 Inicializace nejlepších řešení populace  $B(t)$  a globálně nejlepších řešení  $G(t)$ ;
5 while  $t < t_{\max}$  do
6    $t \leftarrow t + 1$ ;
7   Vytvoření množiny řešení  $P(t)$  pozorováním populace  $Q(t-1)$ ;
8   if Nejlepší řešení z  $P(t) > B(t-1)$  then
9     Uložení nejlepších řešení z  $P(t)$  do  $B(t)$ ;
10  else
11    Uložení  $B(t-1)$  do  $B(t)$ ;
12  if Nejlepší řešení z  $B(t) > G(t-1)$  then
13    Uložení nejlepšího řešení z  $B(t)$  do  $G(t)$ ;
14  else
15    Uložení  $G(t-1)$  do  $G(t)$ ;
16  Vytvoření  $V(t)$  pomocí  $B(t)$  a  $G(t)$ ;
17  Vytvoření  $Q(t)$  pomocí  $V(t)$ ;
```

Na počátku algoritmu, kdy $t = 0$, je inicializována počáteční populace $Q(t)$, která je složena z n kvantových chromozomů, jež jsou reprezentovány tzv. kvantovými úhly (*quantum angles*) místo dříve používaných pravděpodobnostních koeficientů α a β , a to následovně:

$$Q(t) = \{q_1(t), q_2(t), \dots, q_n(t)\},$$

kde je každý z kvantových chromozomů $q_j(t)$ pro $j = 1, 2, \dots, n$ tvořen m kvantovými úhly $\varphi_{j_i}(t)$ jako:

$$q_j(t) = [\varphi_{j_1}(t) \quad \varphi_{j_2}(t) \quad \dots \quad \varphi_{j_m}(t)],$$

jejichž hodnota je nastavena na $\frac{\pi}{4}$. Současně dochází také k inicializaci počátečních rychlostí pro každý kvantový chromozom:

$$V(t) = \{v_1(t), v_2(t), \dots, v_n(t)\},$$

kde je vektor rychlosti $v_j(t)$ jedince pro $j = 1, 2, \dots, n$ definován jako:

$$v_j(t) = (y_{j_1}(t) \quad y_{j_2}(t) \quad \dots \quad y_{j_m}(t)),$$

kde je každá rychlost y_{j_i} pro $i = 1, 2, \dots, m$ nastavena na předem danou hodnotu [44].

Uvažujme generaci $t > 0 \wedge t < t_{\max}$. Z populace $Q(t-1)$ je za pomoci pozorovací funkce $\zeta()$ vygenerována množina řešení $P(t)$ jako

$$P(t) = \{\zeta(q_1(t-1)), \zeta(q_2(t-1)), \dots, \zeta(q_n(t-1))\} = \{p_1(t), p_2(t), \dots, p_n(t)\}, \quad (3.13)$$

kde každé $p_j(t)$ pro $j = 1, 2, \dots, n$ reprezentuje binární řešení problému:

$$p_j(t) = (x_{j_1}(t) \quad x_{j_2}(t) \quad \dots \quad x_{j_m}(t)),$$

kde pro každou ze složek platí:

$$x_{j_i}(t) = \begin{cases} 1 & \text{je-li } r_{j_i}(t) > \cos^2 \varphi_{j_i}(t-1), \quad \text{kde } r_{j_i}(t) \sim U(0, 1) \\ 0 & \text{jinak.} \end{cases}$$

Jedinci ze získané množiny řešení $P(t)$ jsou ohodnoceni pomocí fitness funkce $f()$ jako [44]:

$$F(t) = \{f(p_1(t)), f(p_2(t)), \dots, f(p_n(t))\}.$$

Následně je dle ohodnocení provedena aktualizace nejlepších osobních kvantových úhlů jedinců [44]:

$$B_i(t) = \begin{cases} q_i(t) & \text{je-li } f(p_i(t)) > f(B_i(t-1)), \\ B_i(t-1) & \text{jinak.} \end{cases} \quad (3.14)$$

Pro výpočet globálního nejlepšího kvantového úhlu $G(t)$ je nejprve určen index c aktuálně nejlepšího osobního řešení všech jedinců jako:

$$c = \arg \max_{i=1, \dots, n} f(\zeta(B_i(t))),$$

jenž je využit pro aktualizaci globálního úhlu $G(t)$ jako [44]:

$$G(t) = \begin{cases} B_c(t) & \text{pokud } f(\zeta(B_c(t))) > f(\zeta(G(t-1))) \\ G(t-1) & \text{jinak.} \end{cases} \quad (3.15)$$

Předposledním krokem je aktualizace rychlosti jednotlivých kvantových chromozomů pomocí vylepšeného vzorce *PSO* algoritmu [44]:

$$y_{j_i}(t) = \chi \cdot (\omega \cdot y_{j_i}(t-1) + c_1 \cdot r_1 \cdot (B_{j_i}(t) - \varphi_{j_i}(t-1)) + c_2 \cdot r_2 \cdot (G_i(t) - \varphi_{j_i}(t-1))), \quad (3.16)$$

kde χ, ω, c_1, c_2 jsou koeficienty setrvačnosti a učení a $r_1, r_2 \sim U(0, 1)$.

Při závěrečném kroku iterace dochází k aktualizaci kvantových úhlů jedinců v populaci následovně:

$$\varphi_{j_i}(t) = \varphi_{j_i}(t-1) + y_{j_i}(t).$$

Tento proces se opakuje do té doby, dokud není splněna ukončující podmínka [44].

3.6 Kvantově inspirovaná optimalizace rojem částic

Tato sekce představí algoritmus, který byl navržen v rámci práce po analýze dříve představených technik, jenž byl pojmenován jako kvantově inspirovaná optimalizace rojem částic (*Quantum Inspired Particle Swarm Optimization – QIPSO*). Navržený algoritmus obdobně jako algoritmus *QSE* využívá principy používané v *PSO*. Hlavní rozdílem mezi algoritmem *QSE* a vlastním algoritmem *QIPSO* je to, že zatímco *QSE* reprezentuje jedince kvantovými úhly, algoritmus *QIPSO* používá dvojici pravděpodobnostních koeficientů, jenž jsou aktualizovány kvantovým rotačním hradlem, podobně jako u algoritmů *QIGA* a *QISA*. Obdobně jako u výše uvedených algoritmů proměnná t označuje aktuální iteraci algoritmu a proměnná t_{\max} značí maximální počet iterací.

Algoritmus 5: Kvantově inspirovaná optimalizace rojem částic

```

1  $t \leftarrow 0$ ;
2 Inicializace populace  $Q(t)$  kvantových chromozomů;
3 Inicializace počátečních rychlostí  $V(t)$ ;
4 Inicializace nejlepších řešení populace  $B(t)$  a globálně nejlepších řešení  $G(t)$ ;
5 while  $t < t_{\max}$  do
6    $t \leftarrow t + 1$ ;
7   Vytvoření množiny řešení  $P(t)$  pozorováním populace  $Q(t-1)$ ;
8   if Nejlepší řešení z  $P(t) > B(t-1)$  then
9     | Uložení nejlepších řešení z  $P(t)$  do  $B(t)$ ;
10  else
11    | Uložení  $B(t-1)$  do  $B(t)$ ;
12  if Nejlepší řešení z  $B(t) > G(t-1)$  then
13    | Uložení nejlepšího řešení z  $B(t)$  do  $G(t)$ ;
14  else
15    | Uložení  $G(t-1)$  do  $G(t)$ ;
16  Vytvoření  $V(t)$  pomocí  $B(t)$  a  $G(t)$ ;
17  Vytvoření  $Q(t)$  pomocí  $V(t)$ ;
```

Uvažujme $t = 0$. Algoritmus začíná inicializací počáteční populace $Q(t)$ čítající n kvantových chromozomů jako:

$$Q(t) = \{q_1(t), q_2(t), \dots, q_n(t)\},$$

kde každý jedinec $q_j(t)$ pro $j = 1, 2, \dots, n$ je složen z m dvojic pravděpodobnostních koeficientů $\alpha_{j_i}(t)$ a $\beta_{j_i}(t)$ pro $i = 1, 2, \dots, m$, jejichž hodnota je natavena na $\frac{1}{\sqrt{2}}$. Společně s inicializací populace je pro každého jedince $q_j(t)$ vytvořen vektor rychlostí jako:

$$v_j(t) = (y_{j_1}(t) \quad y_{j_2}(t) \quad \dots \quad y_{j_m}(t)),$$

Uvažujme iteraci algoritmu $t > 0 \wedge t < t_{\max}$. Nejdříve je provedeno pozorování populace $Q(t-1)$, čímž vznikne množina řešení $P(t)$, která je následně ohodnocena fitness funkcí $f()$. Pomocí získaného ohodnocení jsou aktualizovány nejlepší řešení jednotlivých kvantových chromozomů, vizte rovnici 3.14, a globálně nejlepší řešení, vizte rovnici 3.15.

Následuje aktualizace vektorů rychlostí pomocí rovnice:

$$y_{j_i}(t) = \omega \cdot y_{j_i}(t-1) + c_1 \cdot r_1 \cdot (b_{j_i}(t) - x_{j_i}(t)) + c_2 \cdot r_2 \cdot (g_i(t) - x_{j_i}(t)),$$

kde ω je koeficient tření (*friction*) nebo inerciální váha (*intercal weight*) [38] a hodnoty c_1, c_2 reprezentují, po řadě, vliv osobního a globálně nejlepšího řešení. Koeficienty r_1, r_2 obdobně jako u *QSE* představují náhodné proměnné.

Následně jsou pomocí aktualizovaných rychlostí upraveny pravděpodobnostní koeficienty $\alpha_{j_i}(t-1), \beta_{j_i}(t-1)$ jednotlivých kvantových chromozomů pomocí kvantového rotačního hradla $R()$ jako:

$$\begin{bmatrix} \alpha_{j_i}(t) \\ \beta_{j_i}(t) \end{bmatrix} = R(y_{j_i}(t)) \cdot \begin{bmatrix} \alpha_{j_i}(t-1) \\ \beta_{j_i}(t-1) \end{bmatrix}.$$

Iterace algoritmu jsou opakovány do té doby, dokud není splněna ukončující podmínka.

3.7 Aplikace kvantově inspirovaných evolučních algoritmů

Kvantově inspirované evoluční algoritmy nacházejí uplatnění v různých oblastech strojového učení a optimalizace. Příkladem může být jejich využití v návrhu konvolučních neuronových sítí, kde tyto algoritmy umožňují robustně vyhledat silný klasifikátor [46]. Kvantově inspirované evoluční algoritmy mohou být rovněž použity při optimalizaci přepojování elektrických distribučních sítí [33]. Dále se například uvažuje jejich aplikace v analogově evolvovatelném hardwaru [45].

Souhrnný přehled vybraných aplikací *QIEA* v různých oblastech, od kombinatorické optimalizace po numerickou optimalizaci, společně s jejich komentářem přinášejí následující dvě studie [27, 47]. Kvantově inspirované evoluční algoritmy jsou v současné době značně zkoumány a stále vznikají nové studie rozšiřující jejich praktické využití.

Kapitola 4

Optimalizace úlohy batohu pomocí kvantově inspirovaných evolučních algoritmů

Problém batohu patří mezi základní kombinatorické optimalizační úlohy, jež nachází široké uplatnění v oblastech, jako jsou logistika, ekonomie nebo plánování zdrojů a patří do skupiny NP-těžkých problémů. Jednou z možností, jak dosáhnout dostatečně kvalitního řešení v rozumném čase, je použití kvantově inspirovaných evolučních algoritmů, které využívají principy kvantové mechaniky a biologické evoluce.

Kapitola nejdříve poskytne úvod do konceptu řešeného problému batohu, následně popíše reprezentaci tohoto problému v kontextu kvantově inspirovaných evolučních algoritmů a v poslední části se bude věnovat návrhu jednotlivých experimentů, přičemž jejich výsledky budou vyhodnoceny v kapitole 5.

4.1 Základní popis problému batohu

Problém batohu (*knapsack problem*) patří do třídy NP-těžkých problémů, což znamená, že neexistuje žádný algoritmus, který by ho dokázal řešit v polynomiálním čase, ale zároveň není dokázáno, že takový algoritmus neexistuje. Problém batohu je kombinatorický optimalizační problém, jehož cílem je maximalizovat celkovou hodnotu položek v batohu, přičemž každá z položek má svoji určitou hodnotu a váhu. Samotná kapacita batohu je omezena na určitou hodnotu a tudíž součet vah jednotlivých položek v batohu nesmí překročit jeho kapacitu. Pro každou z položek je možné rozhodnout, zda bude do batohu vložena či nikoli, přičemž každá z položek může být v batohu nejvýše jednou [43, 14].

Formálněji řečeno, mějme množinu m položek, přičemž každá položka i je popsána dvojicí (v_i, w_i) , kde

- v_i reprezentuje hodnotu položky i a
- w_i udává její váhu.

Hodnota rozhodovací proměnné x_i je určena následovně:

$$x_i = \begin{cases} 1 & \text{pokud je položka } i \text{ umístěna do batohu,} \\ 0 & \text{jinak.} \end{cases}$$

Hlavním cílem je maximalizovat celkovou hodnotu položek v batohu:

$$\max \sum_{i=1}^m v_i x_i,$$

přičemž součet vah položek nesmí překročit celkovou kapacitu C batohu:

$$\sum_{i=1}^m w_i x_i \leq C, \quad (4.1)$$

kde:

$$x_i \in \{0, 1\} \text{ pro } \forall i = 1, 2, \dots, m.$$

Varianta problém batohu popsaná výše je známá jako *0-1 knapsack problem* [49]. Další možné varianty problému batohu jsou:

- **Vícerozměrný problém batohu (*multi-dimensional knapsack problem*):** Jedná se o zobecněný 0-1 problém batohu, kde má batoh více různých omezení [17].
- **Zlomkový problém batohu (*fractional knapsack problem*):** Jednotlivé položky lze dělit na části, tudíž je možné do batohu umístit i pouze část položky [14].
- **Neomezený problém batohu (*unbounded knapsack problem*):** Každá položka může být do batohu umístěna v neomezeném počtu [4].
- **Vícevýběrový problém batohu (*multiple-choice knapsack problem*):** Položky jsou rozděleny do skupin, přičemž do batohu může být z každé skupiny umístěna nejvýše jedna položka [25].

Seznam výše obsahuje pouze několik vybraných variant problému batohu. Existuje jeho mnoho dalších variant, například omezený problém batohu (*bounded knapsack problem*), kvadratický problém batohu (*quadratic knapsack problem*) a další [25, 8, 9].

4.2 Reprezentace problému batohu pro kvantově inspirované evoluční algoritmy

V klasických evolučních algoritmech je řešení problému batohu reprezentováno jako bitový vektor, kde hodnota 1 reprezentuje stav, kdy byla položka do batohu vložena a hodnota 0 značí nevložení položky do batohu [20].

V kvantově inspirovaných evolučních algoritmech je reprezentace řešení inspirována kvantovým bitem. Každé řešení problému batohu je utvářeno řetězcem m kvantových bitů, přičemž každý kvantový bit q_i je složen z dvojice koeficientů α_i a β_i dle rovnice 3.1, kde $|\alpha_i|^2$ představuje pravděpodobnost, že položka nebude do batohu vložena, zatímco $|\beta_i|^2$ udává pravděpodobnost jejího vložení. Tento řetězec kvantových bitů reprezentuje lineární superpozici všech možných řešení, přičemž jeho délka m odpovídá počtu položek v instanci problému batohu [20].

Tato kvantová reprezentace řešení určuje pravděpodobnosti vložení či nevložení položky do batohu. Aby bylo možné získat konkrétní binární řešení $x = (x_1 \ x_2 \ \dots \ x_m)$, je nutné provést pozorování kvantového stavu, při kterém je každý bit x_i vektoru x pro $i = 1, 2, \dots, m$ získán tak, že je vygenerováno náhodné číslo $r \sim U(0, 1)$. Pokud je splněna podmínka

$r < |\beta_i|^2$ je na i -té pozici binárního řetězce nastavena 1, jinak je nastavena 0. Tento postup je popsán v algoritmu 6 [20].

Algoritmus 6: Proces generování binárního řešení problému batohu [20]

```

1  $i \leftarrow 0$ ;
2 while  $i < m$  do
3    $i \leftarrow i + 1$ ;
4    $r \leftarrow U(0, 1)$ ;
5   if  $r < |\beta_i|^2$  then
6      $x_i \leftarrow 1$ ;
7   else
8      $x_i \leftarrow 0$ ;

```

Po provedeném pozorování může řešení x porušovat kapacitní omezení popsané vzorcem 4.1 batohu a proto je nutné jej opravit. Proces opravy je popsán algoritmem 7, přičemž algoritmus nejdříve určí, zda bylo překročeno kapacitní omezení batohu. V případě jeho překročení jsou odstraňovány položky z batohu do té doby, dokud je batoh přeplněn.

Po opravě případného překročení kapacitního omezení v něm vzniká místo, které může být opět zaplněno. Následuje tedy cyklus, který vybírá položky a vkládá je do batohu, dokud batoh nepřekročí svoji kapacitu. V závěrečném kroku algoritmus odstraní poslední přidanou položku do batohu, aby zajistil nepřekročení jeho kapacity [20].

Algoritmus 7: Obecný proces opravy nevalidního řešení [20]

```

1  $knapsack-overfilled \leftarrow \text{nepravda}$ ;
2 if  $\sum_{i=1}^m w_i x_i > C$  then
3    $knapsack-overfilled \leftarrow \text{pravda}$ ;
4 while  $knapsack-overfilled$  do
5   Výběr  $i$ -té položky batohu;
6    $x_i \leftarrow 0$ ;
7   if  $\sum_{i=1}^m w_i x_i \leq C$  then
8      $knapsack-overfilled \leftarrow \text{nepravda}$ ;
9 while not  $knapsack-overfilled$  do
10  Výběr  $j$ -té položky batohu;
11   $x_j \leftarrow 1$ ;
12  if  $\sum_{i=1}^m w_i x_i > C$  then
13     $knapsack-overfilled \leftarrow \text{pravda}$ ;
14  $x_j \leftarrow 0$ ;

```

Případně je možné místo náhodného výběru položek využít princip, kdy jsou voleny položky na základě poměru ceny a váhy.

Při experimentech byla místo obecného procesu opravy řešení využita heuristika, jež místo náhodného odstraňování a přidávání položek volí vždy tu položku, jejíž poměr ceny a váhy je nejvýhodnější pro odstranění nebo pro zpětné doplnění, jelikož se dle článku [21] ukázala jako potencionálně efektivnější. Tento postup je popsán algoritmem 8.

Algoritmus 8: Proces opravy nevalidního řešení pomocí poměru váha/hodnota [21, 42, 48]

```
1 while  $\sum_{i=1}^m w_i x_i > C$  do
2   Pro každé  $i$  se spočte  $r_i \leftarrow \frac{w_i}{v_i} x_i$ ;
3    $k \leftarrow \arg \max_i r_i$ ;
4    $x_k \leftarrow 0$ ;
5 while  $\sum_{i=1}^m w_i x_i \leq C$  do
6   Pro každé  $i$  se spočte  $r_i \leftarrow \frac{w_i}{v_i} (1 - x_i)$ ;
7    $k \leftarrow \arg \max_i r_i$ ;
8    $x_k \leftarrow 1$ ;
9  $x_k \leftarrow 0$ ;
10 return  $x$ ;
```

Algoritmus zajišťuje, že výsledné řešení bude vždy splňovat kapacitní omezení a zároveň minimalizuje ztrátu kvality řešení. Nejprve jsou odstraňovány položky s nejhorším poměrem váhy a ceny a následně jsou doplňovány položky s nejlepším poměrem, čímž je efektivně opraven kvantový chromozom.

4.3 Návrh experimentů

Tato sekce podrobně popisuje návrh a parametrizaci experimentů, které slouží k analýze a ladění hodnot parametrů jednotlivých algoritmů. V následujících sekcích proměnná j označuje j -tého jedince v populaci, i odpovídá i -té položce batohu a t reprezentuje aktuální iteraci evolučního procesu.

V tabulce 4.1 jsou uvedeny použité datové sady, velikosti instancí m datových sad, počet evaluací fitness funkce a počet nezávislých opakování každého nastavení algoritmu. Datové sady jsou rozděleny do dvou skupin:

- datové sady čítající nízký počet položek problému batohu, tj. 100, 250 a 500¹ sloužící pro ladění parametrů a
- datové sady s vysokým počtem položek problému batohu, respektive 1 000, 2 000, 5 000 a 10 000², jenž byly využity pro ověření schopností *QIEA* při řešení větších instancí problému.

Počet evaluací fitness byl pro většinu experimentů nastaven na 10 000. U největších instancí s $m = 10\,000$ však byly navíc provedeny běhy se 100 000 evaluacemi, aby bylo možné posoudit chování algoritmů v případě poskytnutí delšího výpočetního času.

Pevně stanovený počet evaluací a počet nezávislých opakování jednotlivých nastavení algoritmů umožňuje vyhodnotit vliv velikosti problému, počtu jedinců v populaci a parametrů algoritmů na rychlost konvergence a kvalitu dosažených řešení.

V následující části jsou popsány jednotlivé testované algoritmy a jejich nastavení, přičemž výsledky experimentů spolu s komentářem budou prezentovány v kapitole 5.

¹Zdroj: <https://github.com/mjBM/Quantum-Evolutionary-Algorithm-Knapsack-Python->.

²Zdroj: http://artemisa.unicauca.edu.co/~johnyortega/instances_01_KP/

Datová sada	Instance n	Optimum	Evaluace	Opakování
Malé instanční sady				
—	100	3 967	10 000	30
—	250	10 424	10 000	30
—	500	20 925	10 000	30
Velké instanční sady				
knapPI_1_1000_1000_1	1 000	54 503	10 000	30
knapPI_1_2000_1000_1	2 000	110 625	10 000	30
knapPI_1_5000_1000_1	5 000	276 457	10 000	30
knapPI_1_10000_1000_1	10 000	563 647	10 000 a 100 000	30

Tabulka 4.1: Přehled datových sad, optimálních hodnot a parametrů experimentů.

Kvantově inspirovaný genetický algoritmus

Kvantově inspirovaný genetický algoritmus popsáný v sekci 3.3 společně s jeho pseudokódem (alg. 2, str. 25), umožňuje ladění hodnoty úhlu rotace $\Delta\theta_{j_i}$ pravděpodobnostních koeficientů $\alpha_{j_i}(t)$ a $\beta_{j_i}(t)$ po jednotkové kružnici, vizte obrázek 3.2. Závislost parametru rotace $\Delta\theta_{j_i}$ na velikosti populace byla experimentálně vyhodnocena tak, že každá hodnota rotace byla pro danou instanci s velikostí m testována se všemi zvolenými velikostmi populace. Jednotlivá testovaná nastavení jsou uvedena v tabulce 4.2.

Instance n	Úhel rotace $\Delta\theta_{j_i}$	Velikost populace
100	0,002; 0,01; 0,05; 0,1; 0,2; 0,5; 1; 2	1; 5; 10; 20; 30; 40; 50; 100
250, 500	0,002; 0,01; 0,05; 0,1	1; 5; 10; 20
1 000, 2 000, 5 000, 10 000	0,002	1

Tabulka 4.2: Parametry testování algoritmu *QIGA*.

Vzhledem k tomu, že algoritmus *QIGA* neobsahuje mechanismus pro vzájemnou interakci jedinců v populaci, je z hlediska efektivity nejvýhodnější použití jednočlenné populace. V takovém případě je totiž jediný jedinec vystaven největšímu počtu generací evolučního procesu. Přesto byly provedeny experimenty i s většími populacemi, aby bylo možné posoudit, do jaké míry velikost populace ovlivňuje rychlost konvergence a kvalitu dosažených výsledků.

Výsledky experimentů pro *QIGA* jsou vyhodnoceny v následující kapitole v sekci 5.1.

Kvantově inspirované simulované žíhání

Algoritmus pro kvantově inspirované simulované žíhání, popsáný v sekci 3.4 spolu s pseudokódem (alg. 3, str. 27) podle článku [12], je navržen pro minimalizaci energie. Pro jeho aplikaci na problém batohu bylo nutné upravit některé části algoritmu tak, aby docházelo k maximalizaci hodnoty řešení.

Úprava algoritmu spočívala v obrácení nerovnostmi v akceptační podmínce 3.11, která byla upravena na následující tvar:

$$e_j(t) > e_j(t-1).$$

Společně s touto změnou byl upraven také výpočet faktoru b , a to tak, aby odpovídal směru optimalizace problému:

$$b = e^{\left(\frac{e_j(t) - e_j(t-1)}{T_t}\right)}.$$

Jedním z parametrů algoritmu *QISA* je počáteční teplota, která se standardně určuje podle vzorce 3.8, jenž je však navržen primárně pro numerické problémy. V případě kombinatorických problémů, kde jsou řešení reprezentována diskrétními objekty, je nutné provést úpravu výpočtu počáteční teploty. Nechť $f(x)$ je fitness funkce. Výpočet počáteční teploty T_0 pak probíhá následovně:

$$T_0 = \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (f(x_i) - \bar{f})^2},$$

kde

- x_i představují jednotlivá náhodně vygenerovaná řešení,
- \bar{f} je průměrná hodnota fitness funkcí $f(x_i)$ a
- N je počet vygenerovaných řešení (v tomto případě $N = 1000$).

Tento postup umožňuje použití algoritmu *QISA* i pro řešení kombinatorického problému batohu [12, 18].

Další parametry využívané v algoritmu *QISA* souvisejí s dvěma popsány metodami zahřívací funkce $h(T_t)$. Jedná se o konstantní zahřívání a sigmoidní zahřívání, popsané, po řadě, rovnicemi 3.9 a 3.10. Hodnoty parametrů těchto metod byly převzaty z článku [12] a jsou uvedeny v tabulce 4.3, kde proměnná m označuje velikost instance problému batohu.

Parametr	Zahřívací funkce	
	Konstantní	Sigmoidní
w	$\frac{9}{m}$	—
w_1	—	50
w_2	—	0, 2
w_3	—	$\frac{9}{m}$
w_4	—	$\frac{9}{m}$

Tabulka 4.3: Hodnoty parametrů zahřívacích funkcí [12].

Mezi parametry, které bylo možné dále ladit v algoritmu *QISA*, patří počet jedinců v populaci, volba zahřívací funkce, výběr chladičho plánu a míra ochlazování teploty v průběhu iterací algoritmu. Vzájemné závislosti těchto parametrů byly experimentálně vyhodnoceny prostřednictvím jejich kombinací na různých instancích problému batohu. Konkrétní hodnoty použitých parametrů jsou uvedeny v tabulce 4.4.

Instance n	Zahřívací funkce	Chladicí schéma	Míra ochlazování η	Velikost populace
100, 250, 500	sigmoid, constant	exp, lin	0,90; 0,95; 0,98; 0,99	1, 5, 10, 20, 30, 40, 50, 100
		log, rec-log	—	
1 000, 2 000, 5 000, 10 000	sigmoid	rec-log	—	1

Tabulka 4.4: Parametry testování algoritmu *QISA*.

Obdobně jako *QIGA*, ani algoritmus *QISA* principiálně nevyužívá mechanismus vzájemné interakce jedinců. Přesto byly provedeny experimenty s různými velikostmi populací, jejichž cílem bylo posouzení vlivu počtu jedinců na kvalitu řešení a rychlost konvergence.

Hodnoty pro ochlazování byly převzaty z článku [12], který uvažoval jediný chladicí plán, a to:

- exponenciální (*exponential-exp*) [12, 31]

$$T_t = T_0 \cdot \eta^t \iff T_t = T_{t-1} \cdot \eta, \quad (4.2)$$

přičemž tento plán byl v rámci experimentů rozšířen o další alternativní varianty:

- lineární (*linear-lin*) [31]

$$T_t = T_0 - \eta \cdot t \iff T_t = T_{t-1} - \eta \quad (4.3)$$

- logaritmický (*logarithmic-log*) [41, 31, 5]

$$T_t = \frac{T_0}{\ln(t+1)} \quad (4.4)$$

- rekurzivně-logaritmický (*recursive-logarithmic-rec-log*)

$$T_t = \frac{T_{t-1}}{\ln(t+1)} \quad (4.5)$$

V exponenciálním a lineárním chladicím plánu označuje proměnná η míru ochlazování. Naopak logaritmické a rekurzivně-logaritmické chladicí plány tento parametr nevyužívají. Poslední zmíněný chladicí plán představuje vlastní rozšíření, navržené v této práci. Na rozdíl od logaritmického schématu, které využívá statickou počáteční teploty T_0 , pracuje s dynamicky se měnící teplotou, jež je definována rekurzivně, přičemž nová hodnota T_t se v tomto případě počítá z předchozí teploty T_{t-1} .

Výsledky experimentů pro *QISA* jsou vyhodnoceny v následující kapitole v sekci 5.2.

Kvantová evoluce roje

Kvantová evoluce roje, popsaná v sekci 3.5 a uvedená v algoritmu 4, neumožňuje nastavení ladicích parametrů, neboť koeficienty pro výpočet rychlosti podle rovnice 3.16 jsou pevně dané a uvedené v tabulce 4.5. Tyto koeficienty ω , c_1 a c_2 splňují konvergenční podmínku částic:

$$\omega > \frac{c_1 + c_2}{2} - 1.$$

Koeficient	Hodnota	Význam
χ	0,99	koeficient zúžení (omezení rychlosti částice)
ω	0,7298	inerciální váha (vyvážení explorační a exploatační)
c_1	1,42	kognitivní koeficient (vliv nejlepší osobní pozice)
c_2	1,57	sociální koeficient (vliv nejlepší globální pozice)

Tabulka 4.5: Význam a hodnoty parametrů *QSE* [13, 38].

Zároveň platí, že částice budou rychleji konvergovat ke globálnímu optimu roje než k lokálnímu optimu jednotlivých částic, neboť je splněna nerovnost $c_2 > c_1$ [44].

Vzhledem k tomu, že ladicí parametry algoritmu byly pevně dané, byly v rámci experimentů porovnány různé velikosti populací částic spolu s jejich různými počátečními rychlostmi. Důvodem je skutečnost, že původní článek [44] explicitně neuváděl výchozí hodnotu pro inicializaci počáteční rychlosti jednotlivých částic. Přehled všech testovaných nastavení je uveden v tabulce 4.6.

Instance n	Počáteční rychlost	Velikost populace
100, 250, 500	0, 1, 2, 5, 10, 25, 50, 100	1, 5, 10, 20, 30, 40, 50, 100
1 000, 2 000, 5 000, 10 000	1	5

Tabulka 4.6: Parametry testování algoritmu *QSE*.

Na rozdíl od algoritmů *QIGA* a *QISA*, které neumožňují vzájemnou komunikaci jedinců v populaci, algoritmus *QSE* tento mechanismus již využívá. Díky tomu mohou jedinci sdílet informace o kvalitě nalezených řešení.

Výsledky experimentů pro *QSE* jsou vyhodnoceny v následující kapitole v sekci 5.3.

Kvantově inspirovaná optimalizace rojem částic

Algoritmus kvantově inspirované optimalizace rojem částic, jenž vychází z kombinace principů algoritmů *QIGA* a *QSE*, je popsán v sekci 3.6 a uveden v algoritmu 5. Na rozdíl od algoritmu *QSE* již nevyužívá pevně dané hodnoty koeficientů.

Instance n	Počáteční rychlost	c_1	c_2	ω	Velikost populace
100	0, 1, 2, 5, 10, 25, 50, 100, 150, 200	0,1; 0,25; 0,5; 0,75	0,1; 0,25; 0,5; 0,75	0,002; 0,01; 0,05; 0,1; 0,2	1, 5, 10, 20, 30, 40, 50, 100
250, 500	0, 1, 2, 5, 10, 25, 50, 100, 150, 200	0,5; 0,75	0,1; 0,25	0,01; 0,05	1, 5, 10, 20, 30, 40, 50, 100
1 000, 2 000, 5 000, 10 000	100	0,5	0,25	0,01	5

Tabulka 4.7: Parametry testování algoritmu *QIPSO*.

Z tohoto důvodu byl experimentálně ověřován vliv různých kombinací počáteční rychlosti jednotlivých částic, kognitivního koeficientu c_1 , sociálního koeficientu c_2 , koeficientu zúžení (tření) ω při různých velikostech populace [37, 38, 13]. Přehled všech testovaných parametrů a jejich hodnot je uveden v tabulce 4.7.

Obdobně jako algoritmus *QSE*, také algoritmus *QIPSO* využívá mechanismu vzájemné interakce částic v populaci, čímž je umožněno sdílení informací o kvalitě dosažených řešení.

Výsledky experimentů pro *QIPSO* jsou vyhodnoceny v následující kapitole v sekci 5.4.

Kapitola 5

Experimentální výsledky

Tato kapitola prezentuje výsledky experimentů popsanych v sekci 4.3. Cílem těchto experimentů je zhodnotit výkonnost jednotlivých kvantově inspirovaných evolučních algoritmů a provést jejich porovnání s klasickými evolučními algoritmy.

Nejprve jsou uvedeny výsledky ladění parametrů pro jednotlivé kvantově inspirované evoluční algoritmy. Tyto optimalizované parametry jsou následně aplikovány na rozsáhlejší instance problému batohu, jejichž výsledky jsou rovněž prezentovány. Každý ze zobrazených grafů je navíc doplněn o informaci z tabulky 4.1, která uvádí optimální hodnotu řešení problému batohu pro jednotlivé velikosti testovaných instancí.

Na základě získaných dat je v závěru kapitoly vedena diskuze porovnávající klasické evoluční algoritmy s běžnými optimalizačními metodami, a to včetně analýzy jejich silných a slabých stránek. Zvláštní pozornost je věnována rovněž vzájemnému srovnání jednotlivých kvantově inspirovaných evolučních algoritmů.

Ve všech grafech prezentujících výsledky experimentů je použito označení ve formátu (alg-inst), kde *alg* označuje použitý algoritmus a *inst* velikost instance problému.

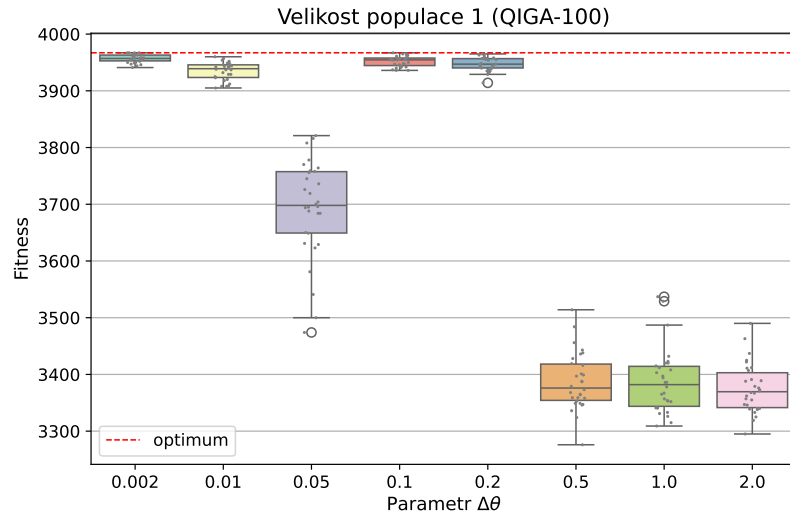
5.1 Kvantově inspirovaný genetický algoritmus

V této části jsou představeny experimentální výsledky týkající se kvantově inspirovaného genetického algoritmu (*QIGA*). Pozornost je věnována zejména vlivu různých hodnot parametru úhlu rotace $\Delta\theta$ a velikosti populace na kvalitu nalezených řešení.

Na obrázku 5.1 jsou zobrazeny výsledky experimentů pro problém o 100 položkách s populací tvořenou jediným jedincem. Sledován je vliv různých hodnot parametru $\Delta\theta$, jejichž konkrétní nastavení vychází z tabulky 4.2. Prezentovány jsou pouze výsledky pro jednoho jedince, neboť algoritmus *QIGA*, jak bylo uvedeno v sekci 4.3, neobsahuje mechanismus pro sdílení informací mezi jedinci v populaci.

Z dat na obrázku 5.1 vyplývá, že parametr $\Delta\theta$ má významný vliv na kvalitu řešení. Nejlepších řešení bylo dosaženo při $\Delta\theta = 0,002$, kde se průměrná kvalita řešení pohybovala velmi blízko optimu a současně vykazovala nejnižší rozptyl.

Obecně platí, že nižší hodnoty parametru $\Delta\theta$ vedly k lepším výsledkům s výjimkou anomálie při $\Delta\theta = 0,05$. Naopak vyšší hodnoty (zejména od 0,5 a výše) vedly ke značnému zhoršení kvality řešení, doprovázené vyšší směrodatnou odchylkou, což naznačuje nižší stabilitu algoritmu. Z výsledků lze usuzovat, že vyšší hodnoty $\Delta\theta$ způsobují velkou změnu kvantového stavu, což má za následek spíše chaotické a méně efektivní prohledávání prostoru možných řešení.



Obrázek 5.1: Vliv hodnot parametru $\Delta\theta$ při velikosti populace 1. Data prezentovaná na grafu byla získána algoritmem *QIGA* na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.

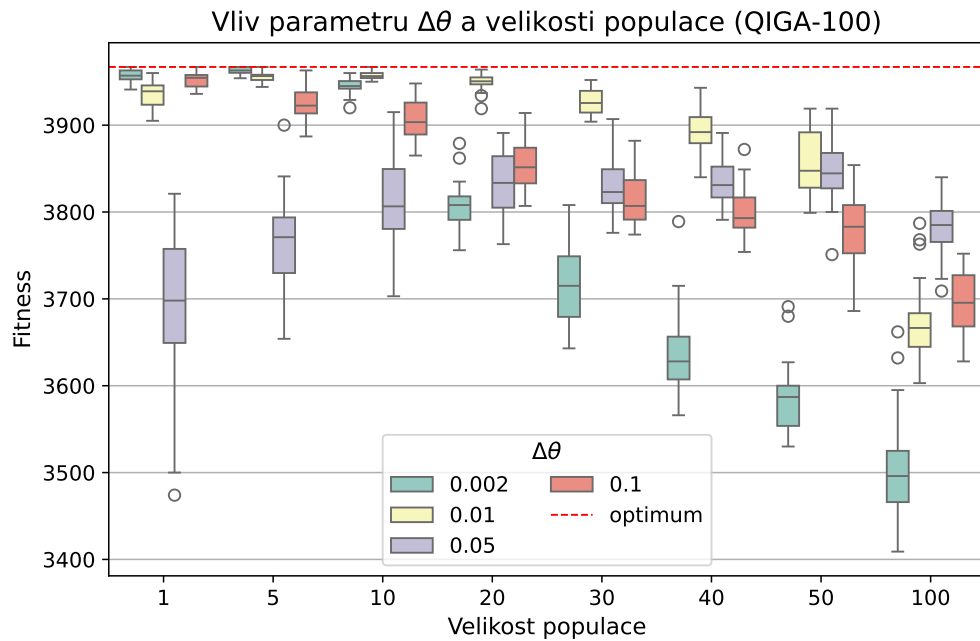
Na základě předchozích výsledků byly z dalších experimentů vyřazeny hodnoty parametru $\Delta\theta = 0,5, 1$ a 2 , neboť opakovaně vedly k výrazně horším výsledkům. Hodnota $\Delta\theta = 0,05$ však byla ponechána, ačkoliv také nepatřila mezi nejúspěšnější, neboť spadá do oblasti hodnot, které v jiných případech vedly k velmi kvalitním řešením. Právě z tohoto důvodu byla tato anomálie zachována i v následujících experimentech, aby mohla být dále analyzována ve vztahu k větším instancím problému.

Tato anomálie byla dále zkoumána v kontextu různých velikostí populace, respektive v závislosti na celkovém počtu vyhodnocení fitness funkce. Výsledky zobrazené na obrázku 5.2 ukazují, že vliv velikosti populace na kvalitu řešení při nastavení parametru $\Delta\theta = 0,05$ není přímočarý. Nejhorší výsledky byly dosaženy při populaci tvořené jediným jedincem. Se zvyšujícím se počtem jedinců se však kvalita zlepšovala a nejlepších výsledků bylo dosaženo při desítkách jedinců v populaci. Při dalším zvyšování velikosti populace však kvalita opět začala mírně klesat.

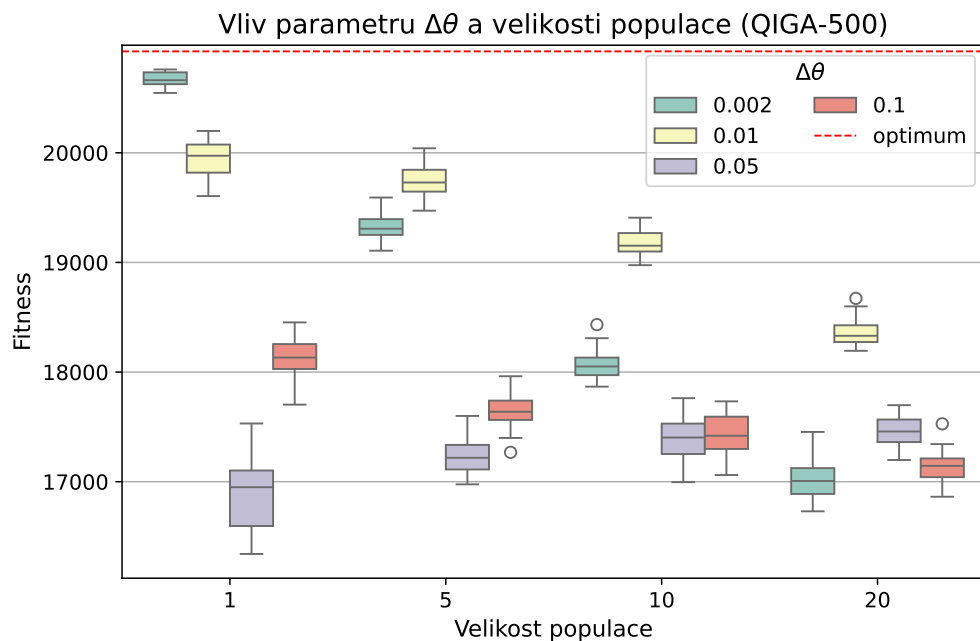
Z grafu 5.2 je rovněž patrné, že algoritmus dosahoval lepších výsledků při populaci velikosti 5, ve srovnání s jedním jedincem, a to při nastavení parametru $\Delta\theta = 0,02$. Tato anomálie může být vysvětlena tím, že i když algoritmus *QIGA* nepodporuje sdílení informací mezi jedinci, větší počet jedinců zvyšuje šanci na nalezení kvalitního řešení. To je zvláště pravděpodobné v tomto případě, kdy instance problému obsahuje 100 položek a počet evaluací je 10 000, což poskytuje jednotlivým jedincům dostatek prostoru pro jejich vývoj.

Toto tvrzení lze ověřit pomocí obrázku 5.3, kde byla velikost instance problému navýšena na 500 položek, zatímco celkový počet evaluací zůstal zachován. Z grafu je patrné, že v tomto případě dosahovala nejlepších výsledků populace tvořena pouze jedním jedincem, protože větší populace neměly dostatek prostoru pro svůj rozvoj, což je dáno tím, že při zachování stejného počtu evaluací se jedinci vyvíjí kratší dobu.

Anomálie pozorovaná při hodnotě parametru $\Delta\theta = 0,005$ lze vysvětlit vlivem zaokrouhlovacích chyb a konvergence, kdy kvantové pravděpodobnosti postupně směřují k hodnotám velmi blízkým nule nebo jedné (např. $0,0000 \dots$ nebo $0,9999 \dots$). V takovém případě se prakticky zastaví změny pozorovaného chromozomu, čímž se výrazně omezí možnost dal-

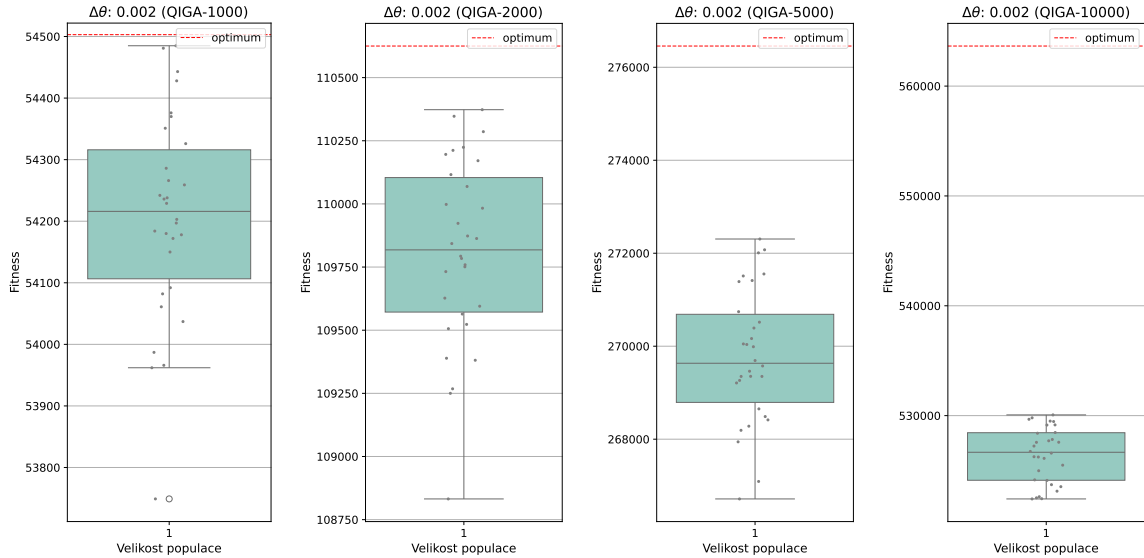


Obrázek 5.2: Závislost kvality řešení na velikosti populace pro různé hodnoty parametru $\Delta\theta$. Data prezentovaná na grafu byla získána algoritmem *QIGA* na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.



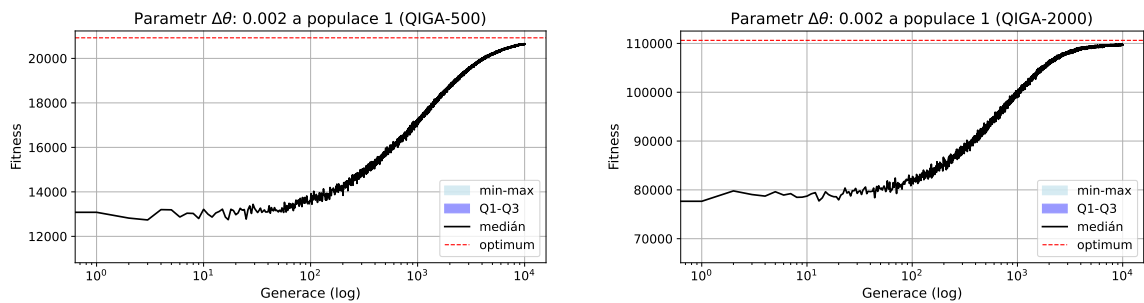
Obrázek 5.3: Závislost kvality řešení na velikosti populace pro různé hodnoty parametru $\Delta\theta$ u instance s 500 položkami. Data prezentovaná na grafu byla získána algoritmem *QIGA* na instanci velikosti 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.

šího průzkumu prostoru řešení. Naopak při mírně odlišných hodnotách parametru (např. $\Delta\theta = 0,051$) zůstávají pravděpodobnosti méně extrémní (např. 0,0002 a 0,9998), což zachovává možnost dalšího zlepšení. Podobně jako v předchozím případě se i zde ukázalo, že u větších instancí problému vedla větší velikost populace k lepším výsledkům, neboť přítomnost více jedinců v populaci zvyšovala šanci na nalezení kvalitnějšího řešení.



Obrázek 5.4: Porovnání kvality řešení při hodnotě $\Delta\theta = 0,002$ a jednočlenné populaci na velkých instancích problému. Data prezentovaná na grafu byla získána algoritmem *QIGA* na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

Obrázek 5.4 shrnuje výsledky experimentů provedených na rozsáhlejších instancích problému batohu, při doladěné hodnotě parametru $\Delta\theta = 0,002$. Z výsledků je patrné, že u instancí o velikosti 1 000 a 2 000 se algoritmus *QIGA* stále dokáže přiblížit k optimálnímu řešení. Naopak u větších instancí, konkrétně pro 5 000 a 10 000 položek batohu, dochází k poklesu kvality řešení, což lze přičíst pevně nastavenému počtu evaluací.



Obrázek 5.5: Konvergenční křivky pro $\Delta\theta = 0,002$ a jednočlennou populaci. Data prezentovaná na grafu byla získána algoritmem *QIGA* na instancích, po řadě, velikosti 500 a 2 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

Obrázek 5.5 zachycuje rychlost konvergence při velikostech instancí 500 a 2 000. V obou případech byla použita jednočlenná populace a doladěná hodnota parametru $\Delta\theta = 0,002$.

$\Delta\theta$	Instance						
	100	250	500	1 000	2 000	5 000	10 000
0,002	3 967	10 411	20 760	54 485	110 373	272 306	530 066
0,01	3 967	10 402	20 199				
0,05	3 919	9 541	17 762				
0,1	3 967	9 986	18 453				
0,2	3 965						
0,5	3 618						
1,0	3 603						
2,0	3 544						
Optimum	3 967	10 424	20 925	54 503	110 625	276 457	563 647

Tabulka 5.1: Nejlepší dosažené fitness hodnoty algoritmem *QIGA* pro prezentované hodnoty parametru $\Delta\theta$ a velikosti instancí.

Tabulka 5.1 prezentuje nejlepší dosažené fitness hodnoty řešení pro různé velikosti instancí a hodnoty parametru $\Delta\theta$. Z výsledků je patrné, že nejnižší hodnota $\Delta\theta = 0,002$ vedla napříč instancemi k nejkvalitnějším řešením, přičemž u instance velikosti 100 bylo dokonce dosaženo optimálního řešení, zatímco u ostatních instancí se nejlepší nalezené hodnoty pohybovaly velmi blízko optimu.

5.2 Kvantově inspirované simulované žíhání

V této sekci jsou prezentovány výsledky experimentů zaměřených na kvantově inspirovaného simulovaného žíhání (*QISA*). Cílem experimentů bylo analyzovat vliv různých konfigurací zahřívací funkce, chladicího plánu a míry ochlazování na kvalitu nalezených řešení. Konkrétní nastavení použitých hodnot parametrů je uvedeno v tabulce 4.4.

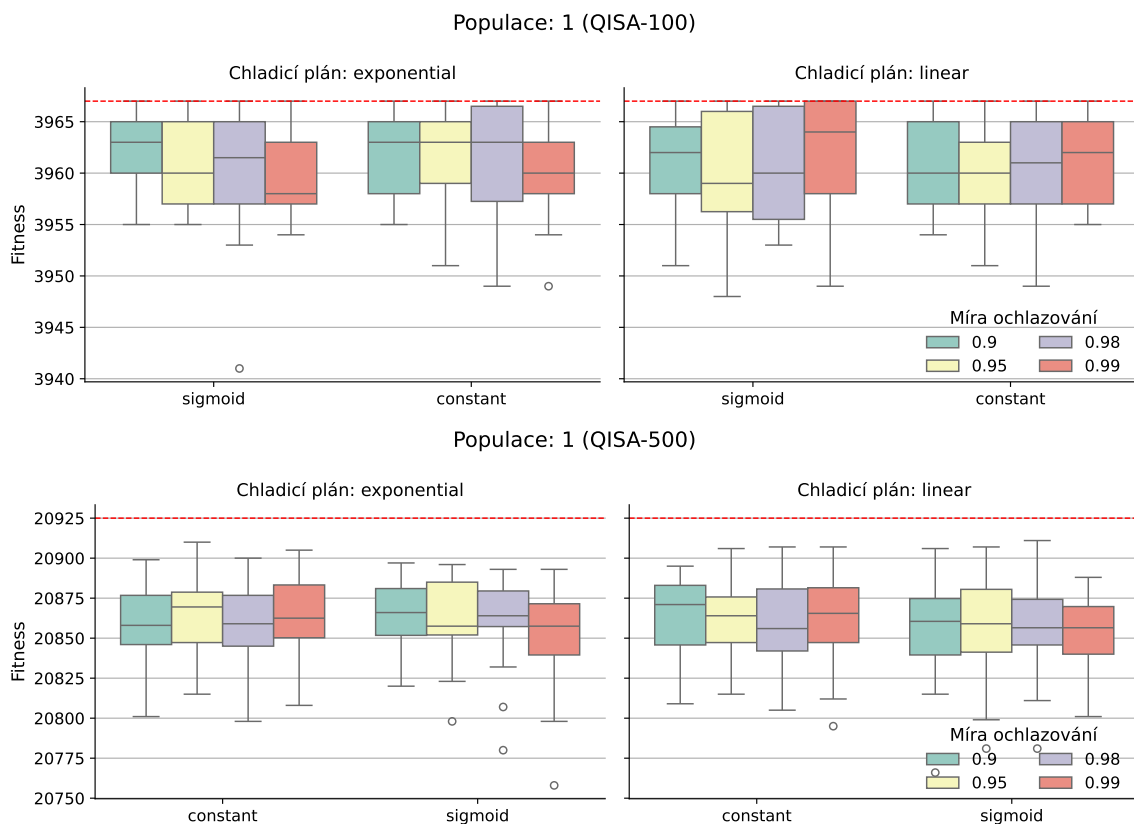
V první fázi experimentů byla provedeno ladění parametrů na menších instancích problému s cílem nalézt optimální konfiguraci algoritmu. Následně byl algoritmus s doladěnými hodnotami parametrů aplikován na větší instance problému s účelem ověřit jeho výkonnost při řešení náročnějších úloh.

Na obrázku 5.6 jsou znázorněny výsledky experimentů pro různá nastavení zahřívací funkce a chladicího plánu, který využívají parametr míry ochlazování, přičemž grafy zobrazují výsledky pro instance problému o velikostech 100 a 500. Z výsledků je patrné, že míra ochlazování nemá výrazný vliv na kvalitu nalezených řešení a obdobně ani volba zahřívacího plánu zásadně neovlivňuje kvalitu výsledků.

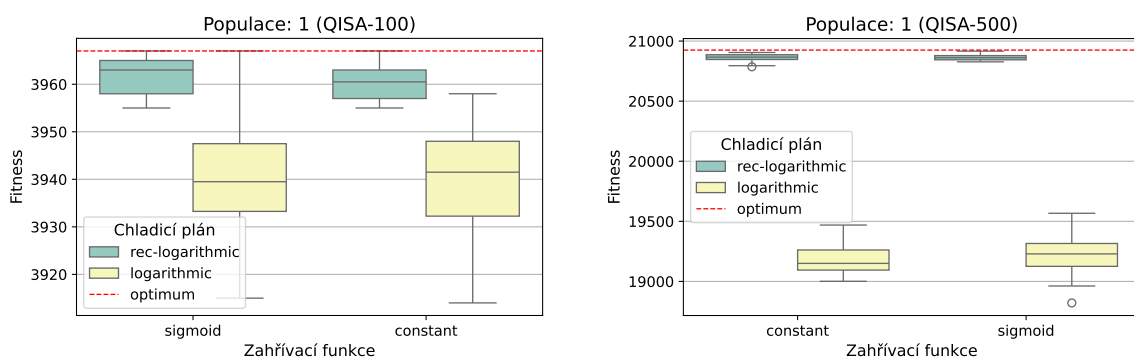
Výsledky experimentů s chladicími plány, jež nevyužívají parametr míry ochlazování, jsou zobrazeny na obrázku 5.7. Z výsledků na uvedeném grafu je patrné, že logaritmický chladicí plán dosahuje výrazně horších výsledků ve srovnání s rekurzivně-logaritmickým plánem.

Z dat v tabulce 5.2 je patrné, že většina kombinací parametrů dosahuje srovnatelných hodnot fitness, přičemž rozdíly mezi různými zahřívacími funkcemi a mírami ochlazování jsou minimální. Výrazně horších výsledků dosahuje pouze logaritmický chladicí plán, jenž ve všech velikostech instancí poskytuje nejnižší kvalitu řešení. Na základě těchto výsledků bylo pro další experimenty na větších instancích zvoleno nastavení s rekurzivně-logaritmickým

chladicím plánem a sigmoidní zahřívací funkcí, jelikož tato kombinace dosáhla nejlepšího výsledku pro největší testovanou instanci.



Obrázek 5.6: Porovnání vlivu různých zahřívacích funkcí a chladicích plánů pro různé míry ochlazování při jednočlenné populaci. Data prezentovaná na grafu byla získána algoritmem *QISA* na instancích, po řadě, velikosti 100 a 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.



Obrázek 5.7: Porovnání vlivu různých chladicích plánů a zahřívacích funkcí, jenž nevyužívají parametr míry ochlazování. Data prezentovaná na grafu byla získána algoritmem *QISA* na instancích, po řadě, velikosti 100 a 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.

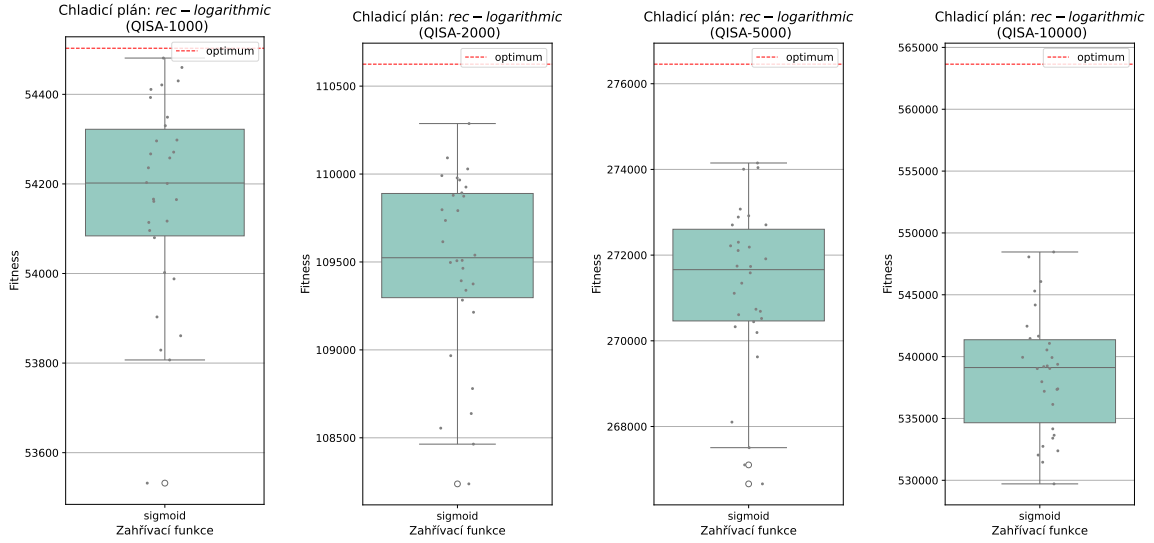
Chladicí plán	Zahřívací funkce	Míra ochlazování	Instance		
			100	250	500
exp	konstantní	0,9	3 967	10 421	20 899
exp	konstantní	0,95	3 967	10 421	20 910
exp	konstantní	0,98	3 967	10 420	20 900
exp	konstantní	0,99	3 967	10 421	20 905
exp	sigmoidní	0,9	3 967	10 419	20 897
exp	sigmoidní	0,95	3 967	10 420	20 896
exp	sigmoidní	0,98	3 967	10 421	20 893
exp	sigmoidní	0,99	3 967	10 420	20 893
lin	konstantní	0,9	3 967	10 421	20 895
lin	konstantní	0,95	3 967	10 420	20 906
lin	konstantní	0,98	3 967	10 422	20 907
lin	konstantní	0,99	3 967	10 421	20 907
lin	sigmoidní	0,9	3 967	10 421	20 906
lin	sigmoidní	0,95	3 967	10 421	20 907
lin	sigmoidní	0,98	3 967	10 420	20 911
lin	sigmoidní	0,99	3 967	10 420	20 888
log	konstantní	—	3 958	10 201	19 469
log	sigmoidní	—	3 967	10 221	19 567
rec-log	konstantní	—	3 967	10 420	20 905
rec-log	sigmoidní	—	3 967	10 420	20 916
Optimum			3 967	10 424	20 925

Tabulka 5.2: Nejlepší dosažené fitness hodnoty algoritmem *QISA* pro prezentované kombinace nastavení chladicího plánu, zahřívací funkce a míry ochlazování při malých instancích problému.

Na obrázku 5.8 jsou prezentovány výsledky experimentů provedených na velkých instancích problému při konfiguraci algoritmu využívající rekurzivně-logaritmický chladicí plán a sigmoidní zahřívací funkci. Z grafů je patrné, že i při velkých instancích problému (1 000, 2 000 a 5 000) si algoritmus udržuje výsledky blízko optimu. Výraznější pokles kvality řešení je patrný až u instance velikosti 10 000, kde algoritmus, již nedosahuje kvalitních výsledků. Výsledky dosažené tímto nastavením jsou dále detailněji uvedeny v tabulce 5.3.

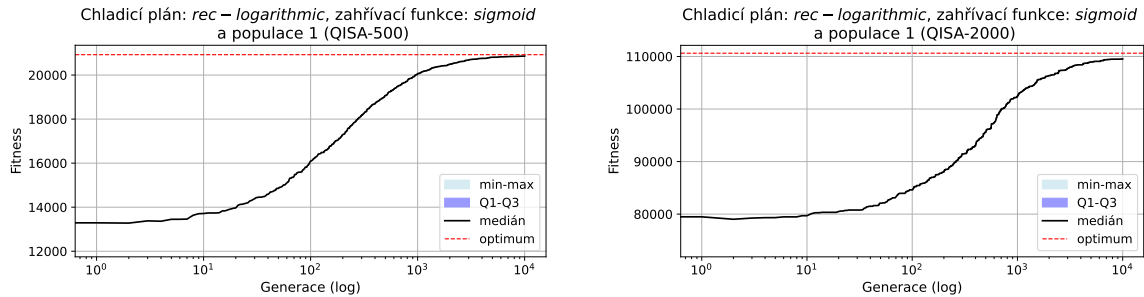
Chladicí plán	Zahřívací funkce	Instance			
		1 000	2 000	5 000	10 000
rec-log	sigmoidní	54 481	110 287	274 150	548 460
Optimum		54 503	110 625	276 457	563 647

Tabulka 5.3: Nejlepší dosažené fitness hodnoty algoritmem *QISA* pro prezentované nastavení při velkých instancích problému.



Obrázek 5.8: Porovnání kvality nalezených řešení při rekurzivně-logaritmickém chladicím plánu, sigmoidní zahřívací funkci při velkých instancích problému a jednočlenné populaci. Data prezentovaná na grafu byla získána algoritmem *QISA* na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

Obrázek 5.5 zachycuje průběhy zlepšování fitness algoritmu *QISA* při řešení instancí problému o velikosti 500 a 2 000. V obou případech byla použita jednočlenná populace spolu s rekurzivně-logaritmický chladicí plánem a sigmoidní zahřívací funkcí.

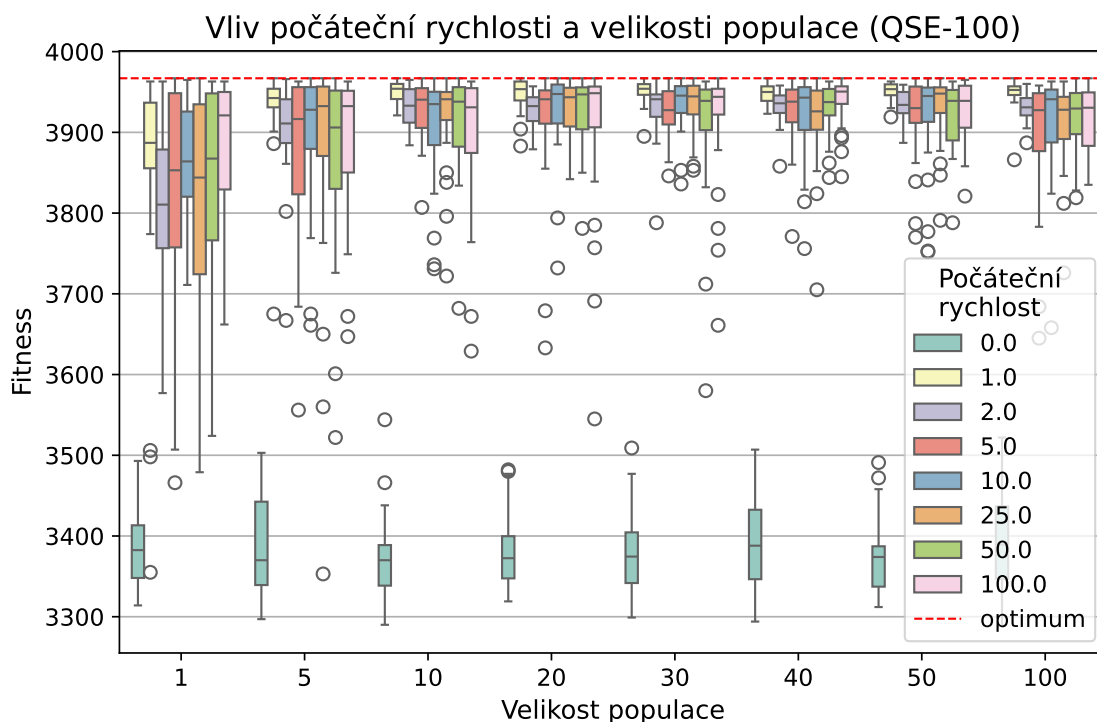


Obrázek 5.9: Konvergenční křivky pro konfiguraci algoritmu, jež byla nastavena na rekurzivně-logaritmický chladicí plán a sigmoidní zahřívací funkci. Data prezentovaná na grafu byla získána algoritmem *QISA* na instancích, po řadě, velikosti 500 a 2 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

5.3 Kvantová evoluce roje

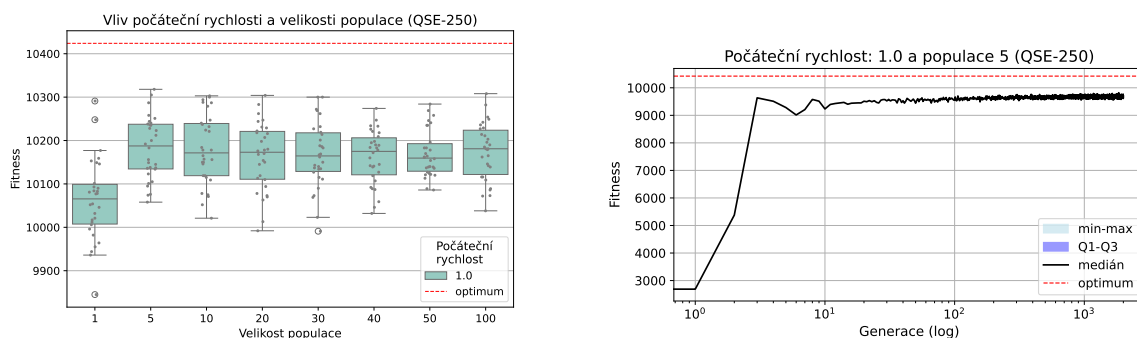
V této sekci jsou prezentovány výsledky experimentů zaměřených na kvantovou evoluci roje (*QSE*). Cílem experimentů bylo analyzovat vliv různých konfigurací počátečních rychlostí a velikostí populací na kvalitu nalezených řešení. Na rozdíl od algoritmů *QIGA* a *QISA*, které nedisponují mechanismem vzájemné interakce jedinců, algoritmus *QSE* již tento mechanismus využívá, což může mít vliv na průběh evoluce a výslednou kvalitu řešení. Z to-

hoto důvodu bude při interpretaci výsledků kladen důraz právě na tyto aspekty. Konkrétní nastavení použitých hodnot parametrů je uvedeno v tabulce 4.6.



Obrázek 5.10: Porovnání vlivu různých velikostí populací a hodnot počátečních rychlostí na kvalitu řešení. Data prezentovaná na grafu byla získána algoritmem *QSE* na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.

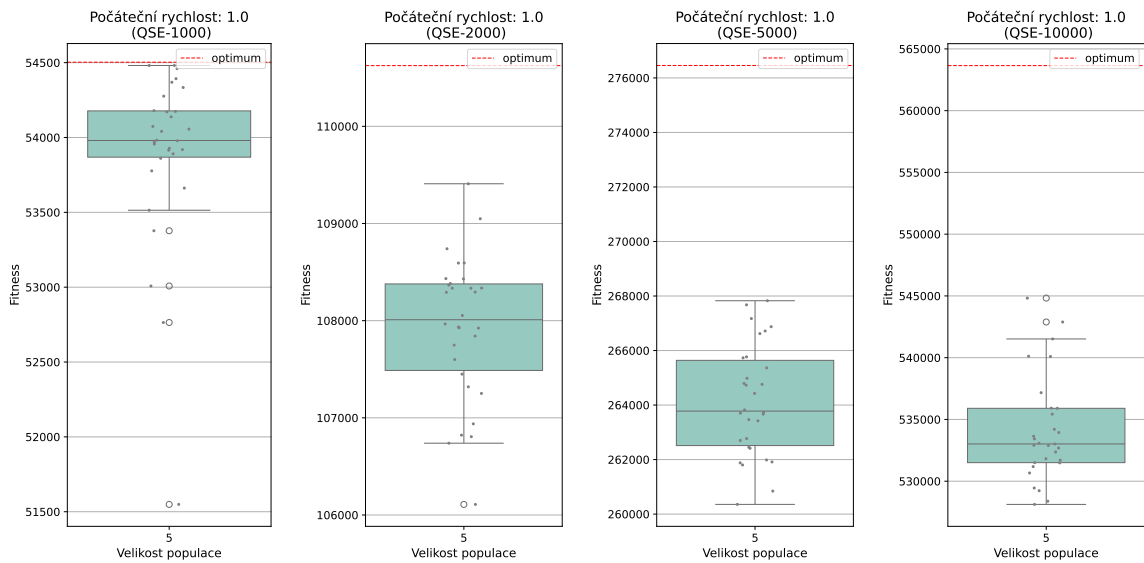
Z výsledků experimentů na instanci velikosti 100 na obrázku 5.10 je patrné, že nejstabilnější a zároveň nejvyšší výsledky poskytuje hodnota počáteční rychlosti 1, zatímco nulová počáteční rychlost generuje nejméně kvalitní výsledky.



Obrázek 5.11: Výsledky experimentů algoritmu *QSE* na instanci velikosti 250. Vlevo jsou znázorněny krabicové grafy dosažených hodnot fitness pro různé velikosti populace při počáteční rychlosti 1, vpravo pak konvergenční křivka vývoje fitness při populaci čítající 5 jedinců a počáteční rychlosti 1. Data byla získána pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci prezentovaných parametrů.

Ze zobrazeného krabicového grafu na obrázku 5.11 prezentujícího výsledky experimentů pro vybranou hodnotu počáteční rychlosti 1 je patrné, že velikost populace při instanci 250 nemá výrazný vliv na kvalitu nalezených řešení. Zároveň lze z konvergenční křivky na stejném obrázku 5.11 pozorovat, že algoritmus při pětičlenné populaci velmi rychle konverguje k řešení blízkým optimu, přičemž další vývoj je již minimální. Z těchto důvodů byla pro experimenty na větších instancích zvolena právě pětičlenná populace, neboť umožňuje uplatnění mechanismu vzájemné komunikace jedinců a zároveň poskytuje jednotlivým jedincům dostatek prostoru pro jejich vývoj.

Na obrázku 5.12 jsou uvedeny výsledky experimentů provedených na velkých instancích problému s konfigurací algoritmu nastavenou na počáteční rychlost 1 a pětičlennou populaci. Z grafů a z tabulky nejvyšších dosažených hodnot fitness 5.4 je zřejmé, že při velikosti instance 1 000 algoritmus dosáhl řešení velmi blízkého optimu. Se zvyšující se velikostí instance však docházelo k postupnému zhoršování kvality nalezených řešení.

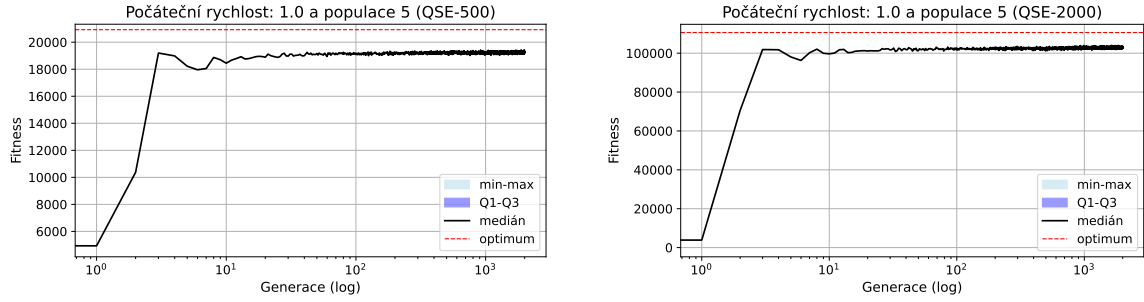


Obrázek 5.12: Porovnání kvality nalezených řešení při hodnotě počáteční rychlosti 1 a pětičlenné populaci při velkých instancích problému. Data prezentovaná na grafu byla získána algoritmem *QSE* na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

Počáteční rychlost	Instance			
	1 000	2 000	5 000	10 000
1	54 481	108 387	267 827	544 824
Optimum	54 503	110 625	276 457	563 647

Tabulka 5.4: Nejlepší dosažené fitness hodnoty algoritmem *QSE* pro prezentované nastavení při velkých instancích problému při populaci čítající 5 jedinců.

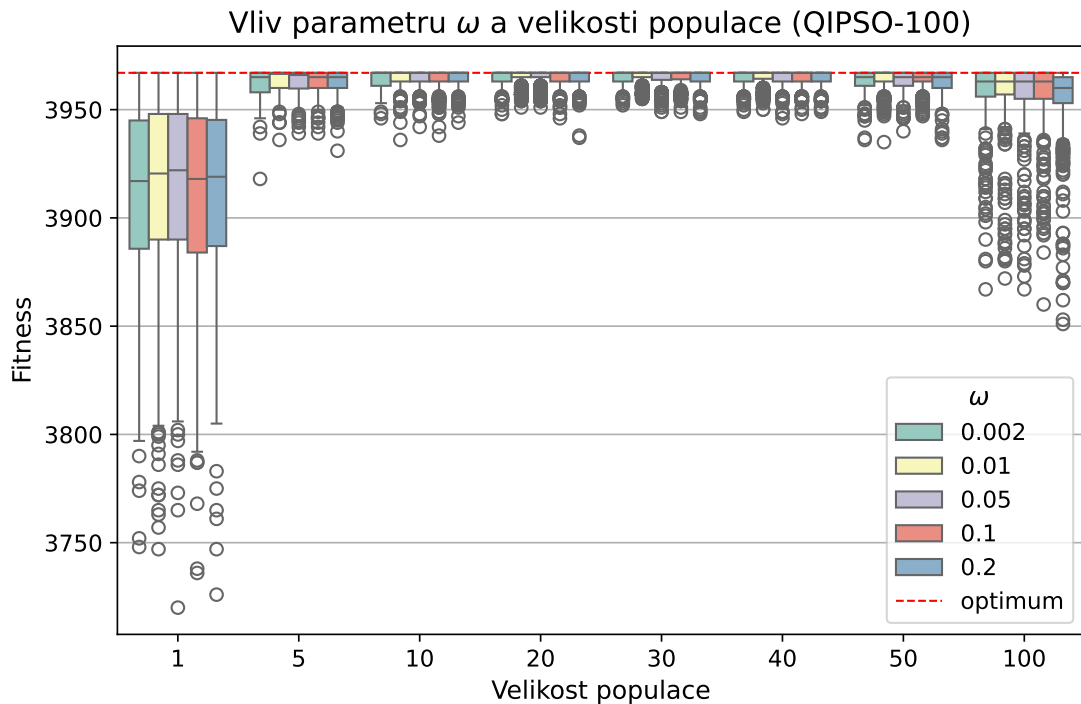
Obrázek 5.13 zachycuje průběhy zlepšování fitness algoritmu při řešení instancí problému velikosti 500 a 2 000. V obou případech byla použita pětičlenná populace spolu s počáteční rychlostí nastavenou na hodnotu 1.



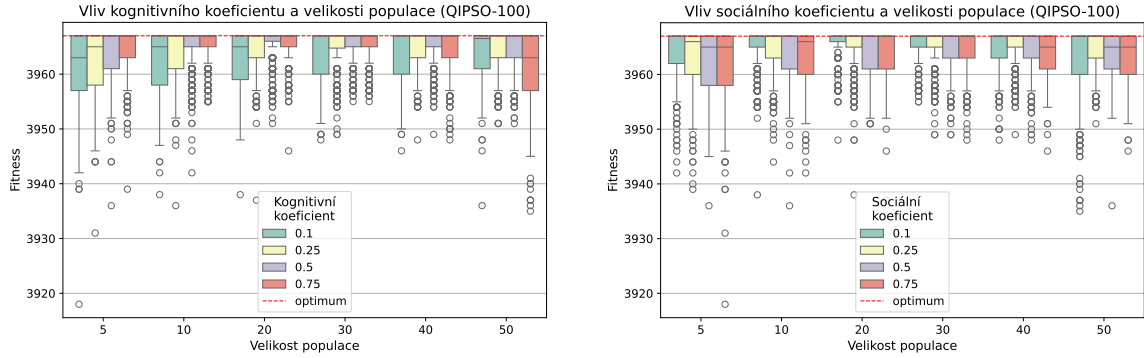
Obrázek 5.13: Konvergenční křivky algoritmu, jehož parametr počáteční rychlosti byl nastaven na hodnotu 1 při pětičlenné populaci. Data prezentovaná na grafu byla získána algoritmem *QSE* na instancích, po řadě, velikosti 500 a 2000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

5.4 Kvantově inspirovaná optimalizace rojem částic

V této sekci jsou prezentovány výsledky experimentů zaměřených na kvantově inspirovanou optimalizaci rojem částic (*QIPSO*). Cílem experimentů bylo analyzovat vliv různých konfigurací počátečních rychlostí, kognitivních koeficientů, sociálních koeficientů, parametrů tření a velikostí populací na kvalitu nalezených řešení. Obdobně jako algoritmus *QSE* i algoritmus *QIPSO* disponuje mechanismem vzájemné interakce jedinců. Konkrétní nastavení použitých hodnot parametrů je uvedeno v tabulce 4.7.

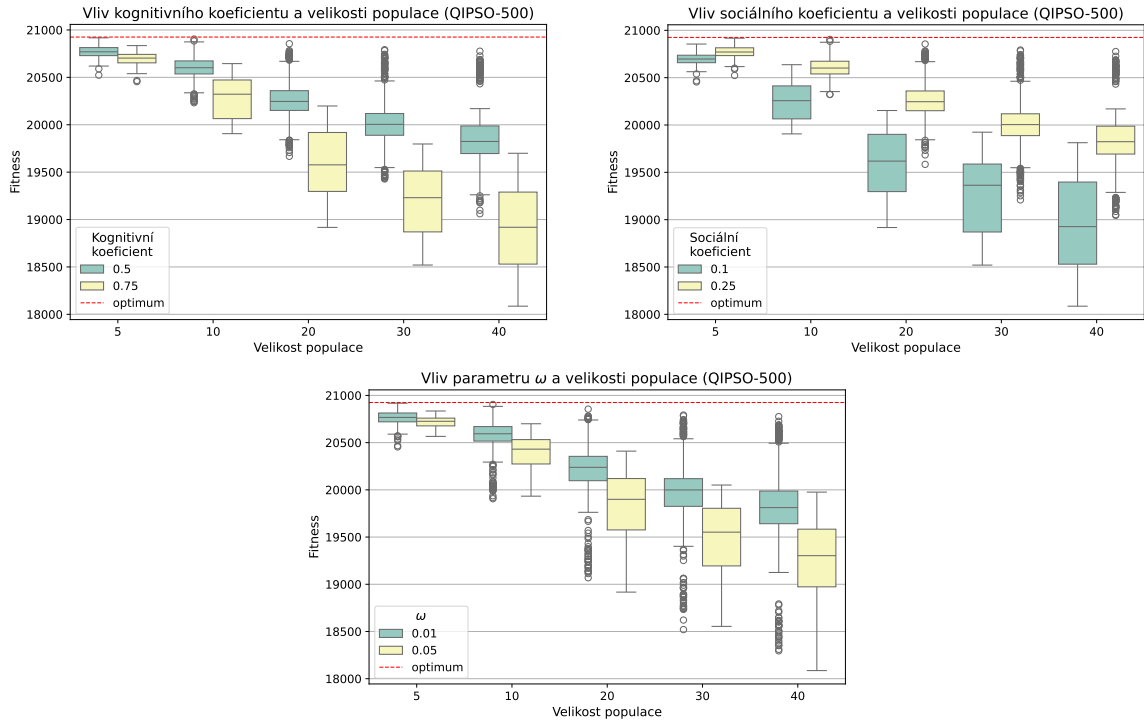


Obrázek 5.14: Závislost kvality řešení na velikosti populace pro různé hodnoty parametru ω . Data prezentovaná na grafu byla získána algoritmem *QIPSO* na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci testovaných parametrů.



Obrázek 5.15: Závislost kvality řešení na velikosti populace pro různé hodnoty, po řadě, kognitivního koeficientu a sociálního koeficientu. Data prezentovaná na grafu byla získána algoritmem *QIPSO* na instanci velikosti 100 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci testovaných parametrů.

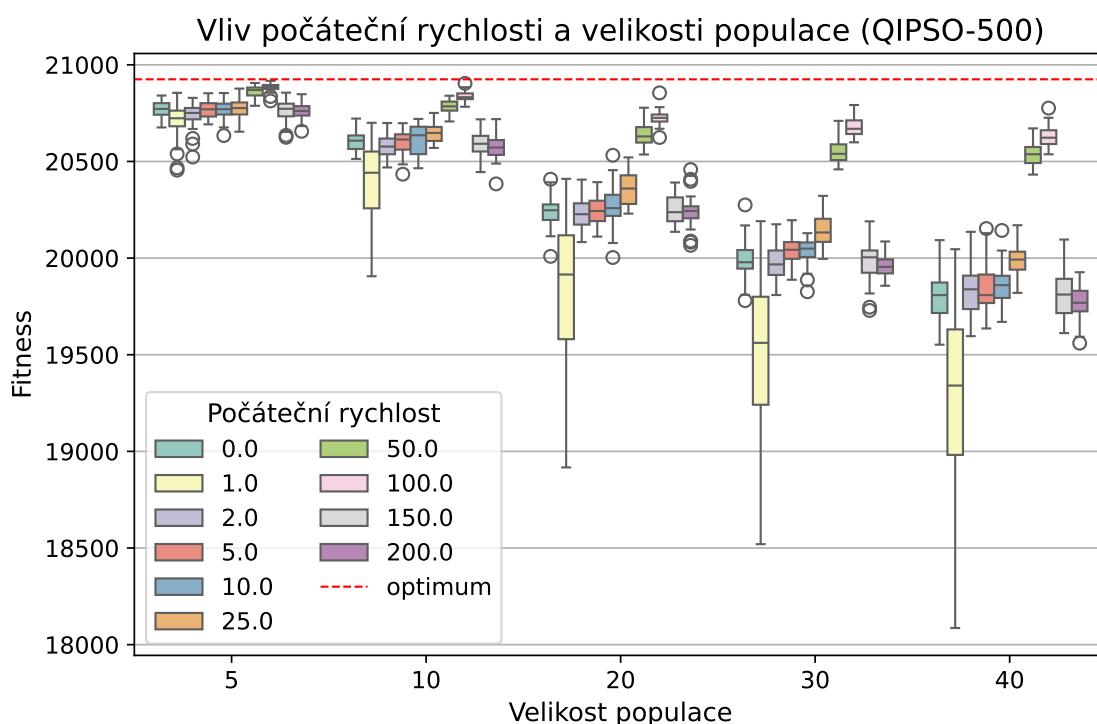
Jak je patrné z obrázku 5.14, populace o velikosti 1 a 100 poskytují nekvalitní řešení, a proto budou v dalších grafech tyto velikosti vynechány, rovněž nebude dále uvažována populace 50, protože při experimentování s většími instancemi problémů by jednotlivci měli omezený prostor pro svůj rozvoj z důvodu omezeného počtu evaluací. Na základě těchto výsledků byly vybrány k další analýze hodnoty parametru ω 0,01 a 0,05, které průměrně dosahovaly kvalitnějších výsledků oproti ostatním testovaným hodnotám.



Obrázek 5.16: Závislost kvality řešení na velikosti populace pro různé hodnoty, po řadě, kognitivního koeficientu, sociálního koeficientu a parametru tréní. Data prezentovaná na grafu byla získána algoritmem *QIPSO* na instanci velikosti 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci testovaných parametrů.

Graf na obrázku 5.15 prezentuje závislost kvality řešení na velikosti populace pro různé hodnoty kognitivního a sociálního koeficientu. Na základě těchto výsledků budou v dalších experimentech uvažovány hodnoty kognitivního koeficientu 0,5 a 0,75 a sociálního koeficientu 0,1 a 0,25, protože tyto kombinace nastavení průměrně poskytovaly nejkvalitnější řešení.

Vybrané hodnoty kognitivního a sociálního koeficientu spolu s parametry tréní byly dále analyzovány na větší instanci o velikosti 500, jak ukazuje obrázek 5.16. Z výsledků je patrné, že nejkvalitnější řešení bylo dosaženo při kognitivním koeficientu 0,5, sociálním koeficientu 0,25 a parametru tréní 0,01. Prezentovaná data na obrázku 5.17 zobrazující vliv počáteční rychlosti na kvalitu řešení se nejvíce odlišovaly u populace velikosti 50 a 100, přičemž lepších výsledků bylo dosaženo při velikosti 100 a tudíž tato počáteční rychlost společně s výše vybranými hodnotami kognitivního koeficientu, sociálního koeficientu a parametru tréní byly vybrány k další analýze při větších instancích problému.

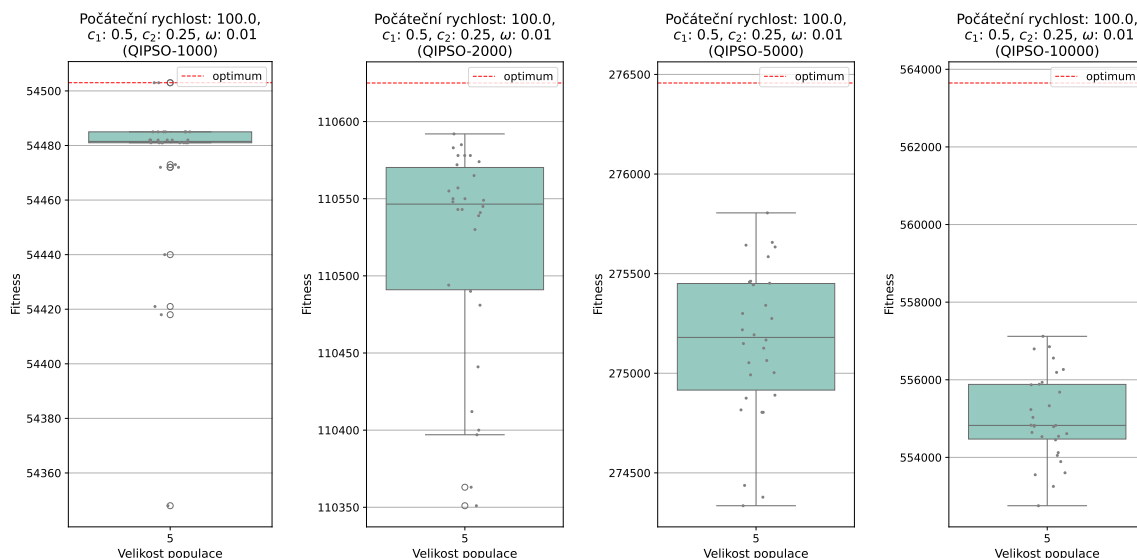


Obrázek 5.17: Závislost kvality řešení na velikosti populace pro různé hodnoty počáteční rychlosti. Data prezentovaná na grafu byla získána algoritmem *QIPSO* na instanci velikosti 500 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh pro každou kombinaci testovaných parametrů.

Na obrázku 5.18 jsou uvedeny výsledky experimentů provedených na velkých instancích problému s konfigurací algoritmu nastavenou na počáteční rychlost 100, kognitivní koeficient 0,5, sociální koeficient 0,25, parametr tréní 0,01 a pětičlennou populaci. Z grafů a z tabulky 5.5 nejvyšších dosažených hodnot fitness je patrné, že při velikosti instance 1 000 algoritmus dosáhl optimálního řešení, zatímco při velikostech instancí 2 000 a 5 000 se výsledná řešení pohybovala velmi blízko optimu.

Počáteční rychlost	c_1	c_2	ω	Instance			
				1 000	2 000	5 000	10 000
100	0,5	0,25	0,01	54 503	110 592	275 805	557 120
Optimum				54 503	110 625	276 457	563 647

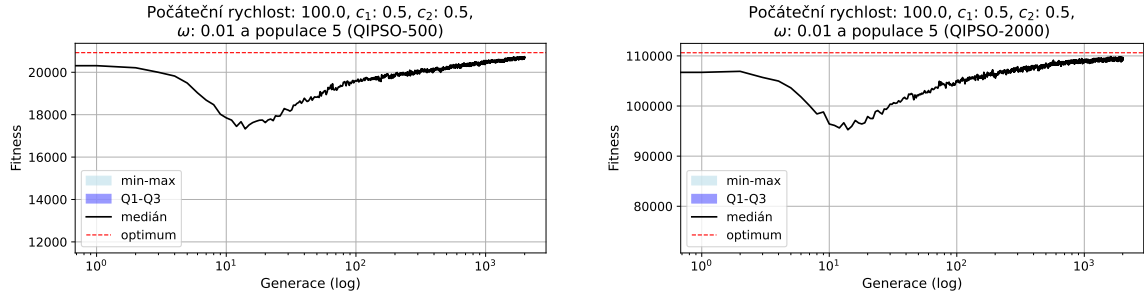
Tabulka 5.5: Nejlepší dosažené fitness hodnoty algoritmem *QIPSO* pro prezentované nastavení při velkých instancích problému při populaci čítající 5 jedinců.



Obrázek 5.18: Porovnání kvality nalezených řešení při hodnotě počáteční rychlosti 100, koefektivního koeficientu 0,5, sociálního koeficientu 0,25, parametru tření 0,01 a pětičlenné populaci při velkých instancích problému. Data prezentovaná na grafu byla získána algoritmem *QIPSO* na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

Konvergenční křivky zobrazené na obrázku 5.19 ukazují, že v počátečních generacích algoritmu dochází k dočasnému poklesu kvality řešení, který je následně překonán a kvalita nalezených řešení opět vzrůstá. Toto chování je způsobeno vysokou počáteční rychlostí částic, kdy dochází k výrazné úpravě pravděpodobnostních koeficientů kvantových chromozomů prostřednictvím kvantového rotačního hradla.

Vlivem vysoké počáteční rychlosti dochází v prvních generacích k rychlému nalezení řešení s relativně vysokou kvalitou. V této fázi však převládá explorace, která vede k výrazným změnám v hodnotách pravděpodobnostních koeficientů a k dočasnému opouštění kvalitních oblastí prostoru řešení, což způsobuje dočasný pokles kvality. Následně, jak se rychlost stabilizuje, přechází algoritmus do fáze exploatace, kdy se populace soustředí na vylepšování nalezených kvalitních oblastí a kvalita řešení opět roste.



Obrázek 5.19: Konvergenční křivky algoritmu, jehož parametr počáteční rychlosti byl nastaven na 100, kognitivního koeficientu na 0,5, sociálního koeficientu na 0,25, parametru tření na 0,01 při pětičlenné populaci. Data prezentovaná na grafu byla získána algoritmem *QIPSO* na instancích, po řadě, velikosti 500 a 2000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

5.5 Srovnání algoritmů a diskuze

V této sekci jsou nejprve navzájem porovnány kvantově inspirované evoluční algoritmy *QIGA*, *QISA*, *QSE* a *QIPSO*, jež byly popsány, po řadě, v sekcích 3.3, 3.4, 3.5 a 3.6 a jejichž doladěné parametry byly představeny, po řadě, v sekcích 5.1, 5.2, 5.3 a 5.4, přičemž přehled těchto parametrů a jejich hodnot pro jednotlivé algoritmy je uveden v tabulce 5.6. Tyto konfigurace byly následně využity ve všech srovnávacích experimentech. Po tomto vzájemném porovnání budou kvantově inspirované algoritmy dále srovnány s vybranými běžně používanými optimalizačními metodami.

Algoritmus	Konfigurace parametrů	Velikost populace
QIGA	$\Delta\theta = 0,002$	1
QISA	rekurzivně-logaritmický chladicí plán a sigmoidní zahřívací funkce	1
QSE	počáteční rychlost = 1	5
QIPSO	počáteční rychlost = 100, $c_1 = 0,5$, $c_2 = 0,25$, $\omega = 0,01$	5

Tabulka 5.6: Nejlepší nalezené hodnoty parametrů pro jednotlivé algoritmy.

Na obrázku 5.20 jsou znázorněny výsledky experimentů porovnávajících jednotlivé kvantově inspirované evoluční algoritmy na větších instancích problému, přičemž jejich konfigurace odpovídá parametrům uvedených v tabulce 5.6.

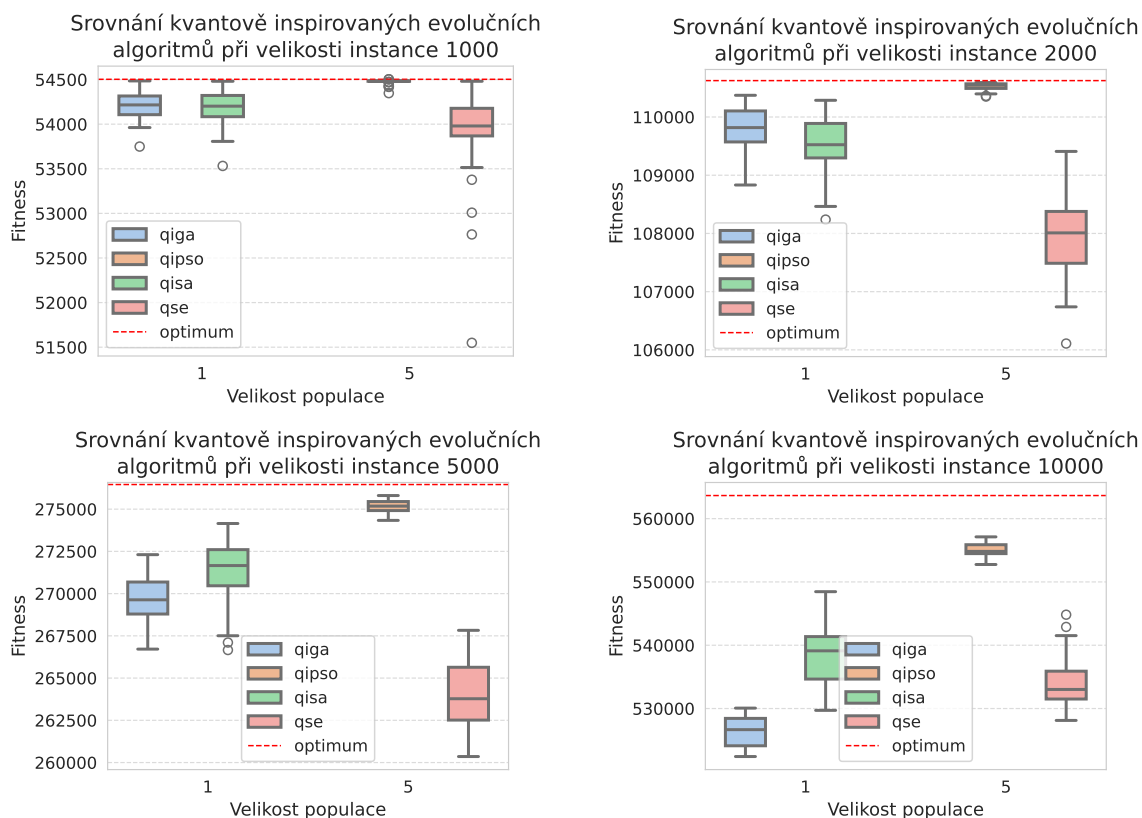
Z výsledků uvedených v tabulce 5.7 a na obrázku 5.20 je patrné, že žádný z kvantově inspirovaných evolučních algoritmů, co se týče kvality vyvinutých výsledků velmi nezaostává. U instance o velikosti 1000 položek se všechny algoritmy blíží optimální hodnotě, přičemž algoritmu *QIPSO* se jako jedinému podařilo nalézt řešení, jehož fitness odpovídá přímo optimální hodnotě.

S rostoucími velikostmi instancí se mezi algoritmy začínají výrazněji projevovat rozdíly v kvalitě nalezených řešení. Algoritmus *QIPSO* dosahuje nejlepších výsledků ve všech testovaných instancích a nejvíce se přibližuje známému optimu i při instancích o velikosti 5 000 a 10 000. Naopak algoritmy *QIGA* a *QSE* vykazují u větších instancí výraznější odchylky od

optimální hodnoty, kdy se algoritmus *QSE* vzdaluje optimu již od velikosti instance 2 000, zatímco algoritmus *QIGA* se začíná mírně vzdalovat až od velikosti instance 5 000.

Algoritmus	Instance			
	1 000	2 000	5 000	10 000
QIGA	54 485	110 373	272 306	530 066
QISA	54 481	110 287	274 150	548 460
QSE	54 481	108 387	267 827	544 824
QIPSO	54 503	110 592	275 805	557 120
Optimum	54 503	110 625	276 457	563 647

Tabulka 5.7: Nejlepší dosažené fitness hodnoty jednotlivými kvantově inspirovanými evolučními algoritmy při konfiguraci jejich parametrů dle tabulky 5.6 pro 10 000 evaluací.

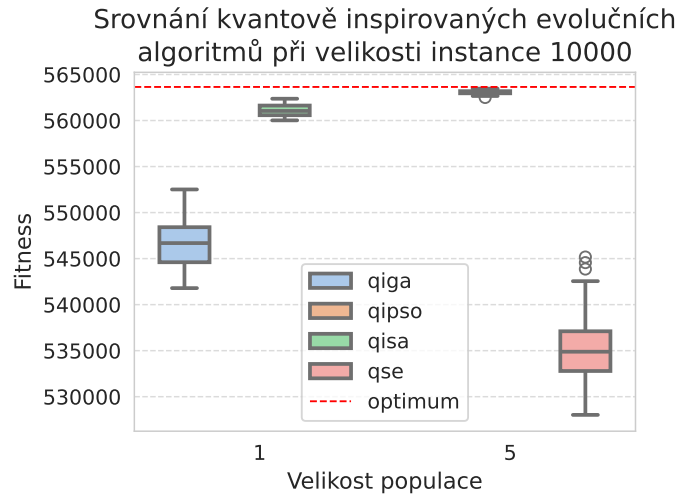


Obrázek 5.20: Data prezentovaná na grafu byla získána, po řadě, algoritmy *QIGA*, *QISA*, *QSE* a *QIPSO* na instancích, po řadě, velikosti 1 000, 2 000, 5 000 a 10 000 pomocí 30 nezávislých běhů s 10 000 evaluacemi na běh.

Na obrázku 5.21 a v tabulce 5.8 jsou prezentovány získané fitness hodnoty pomocí jednotlivých kvantově inspirovaných evolučních algoritmů na instanci o velikosti 10 000 položek, přičemž každý běh měl k dispozici 100 000 evaluací. Tento experiment byl proveden

za účelem ověření, jakého výkonu jsou algoritmy schopny dosáhnout, pokud jim bude na rozsáhlých instancích poskytnut větší výpočetní prostor.

Ve srovnání s předchozími experimenty, kde bylo použito pouze 10 000 evaluací, se ukazuje, že prodloužený běh umožňuje většině algoritmů lépe prozkoumat prostor řešení a nalézt kvalitnější výsledky. Výjimkou je algoritmus *QSE*, který na rozdíl od ostatních algoritmů zaznamenal pouze minimální zlepšení ve kvalitě nalezených řešení. Výraznějšího zlepšení, a to jak z hlediska kvality generovaných řešení, tak jejich stability, dosáhl algoritmus *QISA*, avšak nejlepšího výsledku opět dosáhl algoritmus *QIPSO*, jenž se výrazně přiblížil optimální hodnotě.



Obrázek 5.21: Data prezentovaná na grafu byla získána, po řadě, algoritmy *QIGA*, *QISA*, *QSE* a *QIPSO* na instanci velikosti 10 000 pomocí 30 nezávislých běhů s 100 000 evaluacemi na běh.

Algoritmus	Maximální fitness
QIGA	552 513
QISA	562 368
QSE	545 171
QIPSO	563 464
Optimum	563 647

Tabulka 5.8: Nejlepší dosažené fitness hodnoty jednotlivými kvantově inspirovanými evolučními algoritmy při konfiguraci jejich parametrů dle tabulky 5.6 pro 100 000 evaluací.

Výsledky kvantově inspirovaných evolučních algoritmů byly porovnány s běžnými optimalizačními metodami, které byly představeny ve studii [16]. Tato studie byla zvolena mimo jiné proto, že využívá stejné datové sady, jaké byly použity v této práci, vizte tabulka 4.1, což umožňuje přímé porovnání dosažených výsledků.

Ve zmíněné studii byly experimentálně testovány následující algoritmy:

- simulované žíhání (*Simulated Annealing – SA*) [19],

- genetický algoritmus (*Genetic Algorithm – GA*) [23],
- hladový algoritmus (*Greedy Search Algorithm – GSA*) [32],
- dynamické programování (*Dynamic Programming – DP*) [7] a
- metoda větví a mezí (*Branch and Bound – BB*) [29].

Výsledky těchto metod jsou společně s výsledky kvantově inspirovaných evolučních algoritmů uvedeny v tabulce 5.9.

Algoritmus	Instance			
	1 000	2 000	5 000	10 000
SA	36 179	65 793	150 731	563 647
GA	130	102 340	102 340	562 556
GSA	14 927	25 579	49 306	292 225
DP	54 503	110 625	276 457	106 464
BB	53 397	109 679	275 720	130
QIGA	54 485	110 373	272 306	552 513
QISA	54 481	110 287	274 150	562 368
QSE	54 481	108 387	267 827	545 171
QIPSO	54 503	110 592	275 805	563 464
Optimum	54 503	110 625	276 457	563 647

Tabulka 5.9: Nejlepší dosažené fitness hodnoty jednotlivými algoritmy.

Tabulka 5.9 shrnuje nejlepší dosažené hodnoty fitness jednotlivých algoritmů na instancích různých velikostí. Z výsledků je patrné, že exaktní algoritmy [16], jako jsou *DP* a *BB*, poskytují optimální nebo téměř optimální řešení pro instance velikosti 1 000, 2 000 a 5 000, ale u instance o velikosti 10 000 jejich výkonnost znatelně klesá. Heuristiky *SA* a *GA* obecně dosahují nižší kvality řešení, avšak u instance 10 000 poskytly výsledky blízké optimální hodnotě a v případě algoritmu *SA* se dokonce podařilo nalézt řešení, jehož kvalita odpovídá známému optimu.

Kvantově inspirované evoluční algoritmy dosahují stabilně vysokých hodnot fitness napříč všemi testovanými instancemi. Nejlepší výkonnost opakovaně prokazuje navržený algoritmus *QIPSO*, který se výrazně přibližuje známému optimu i u největších instancí. Také algoritmy *QISA* a *QIGA* poskytují konkurenceschopné výsledky, přičemž *QSE* mírně zůstává, zejména u rozsáhlejších úloh.

Kapitola 6

Závěr

Tato diplomová práce se zabývala problematikou optimalizačních algoritmů využívajících principů kvantové fyziky. Pro další zpracování byla zvolena skupina kvantově inspirovaných evolučních algoritmů, jež kombinují principy biologické evoluce a kvantové mechaniky.

V teoretické části byly podrobně popsány jak základy kvantové fyziky, tak hlavní pojmy z oblasti evolučních algoritmů. Následně byly představeny čtyři konkrétní kvantově inspirované evoluční algoritmy, jež tyto principy kombinují. Konkrétně se jednalo o kvantově inspirovaný genetický algoritmus (*QIGA*), kvantově inspirované simulované žíhání (*QISA*), kvantovou evoluci roje (*QSE*) a kvantově inspirovanou optimalizaci rojem částic (*QIPSO*), přičemž poslední z nich byl navržen v rámci této práce.

Tyto algoritmy byly experimentálně vyhodnoceny na variantě 0-1 problému batohu, který spadá do skupiny NP-těžkých problémů. Nejprve byly na instancích o velikosti 100, 250 a 500 provedeny experimenty za účelem ladění parametrů každého z implementovaných algoritmů. Následně byly tyto parametry použity pro testování algoritmů na rozsáhlejších instancích problému o velikosti 1 000, 2 000, 5 000 a 10 000.

Výsledky ukázaly, že všechny zkoumané algoritmy jsou schopny produkovat řešení, jež se blíží optimální hodnotě. Nejlepšího a zároveň nejstabilnějšího výkonu však dosahoval navržený algoritmus *QIPSO*, který si vedl nejlépe při řešení všech testovaných instancí.

Z experimentální části vyplývá, že kvantově inspirované evoluční algoritmy představují silnou konkurenci klasickým heuristickým metodám, neboť dosahovaly stabilních výkonů a zároveň kvalitních výsledků i u rozsáhlých instancí problému. Jejich výhodou je vyšší diverzita řešení a schopnost vyhnout se lokálním extrémům díky využití kvantových principů, zejména reprezentace jedinců pomocí pravděpodobnostních koeficientů.

Do budoucna by bylo možné práci rozšířit o další kvantově inspirované evoluční přístupy, jako je například kvantově inspirovaná diferenciální evoluce nebo mravenčí algoritmy, a to i při řešení jiných typů optimalizačních problémů.

Tato práce byla prezentována na studentské konferenci inovací, technologií a vědy v IT Excel@FIT 2025, kde byla vybrána a oceněna odborným panelem.

Literatura

- [1] ABIODUN, O. I.; JANTAN, A.; OMOLARA, A. E.; DADA, K. V.; MOHAMED, N. A. et al. State-of-the-art in artificial neural network applications: A survey. *Heliyon* online. 1. vyd., 2018, sv. 4, č. 11, s. e00938. ISSN 2405-8440. Dostupné z: <https://doi.org/10.1016/j.heliyon.2018.e00938>. [cit. 2025-04-29].
- [2] ABIODUN, O. I.; JANTAN, A.; OMOLARA, A. E.; DADA, K. V.; UMAR, A. M. et al. Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access* online. 1. vyd., 2019, sv. 7, č. 1, s. 158820–158846. Dostupné z: <https://doi.org/10.1109/ACCESS.2019.2945545>. [cit. 2025-04-29].
- [3] ADLEMAN, L. M. Molecular Computation of Solutions to Combinatorial Problems. *Science* online. 1. vyd., 1994, sv. 266, č. 5187, s. 1021–1024. Dostupné z: <https://doi.org/10.1126/science.7973651>. [cit. 2025-04-29].
- [4] ANDONOV, R.; POIRRIEZ, V. a RAJOPADHYE, S. Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research* online. 1. vyd., 2000, sv. 123, č. 2, s. 394–407. ISSN 0377-2217. Dostupné z: [https://doi.org/10.1016/S0377-2217\(99\)00265-9](https://doi.org/10.1016/S0377-2217(99)00265-9). [cit. 2025-01-21].
- [5] BANCHS, R. E. *Simulated Annealing*. Rafael E. Banchs, 1997. Dostupné z: https://rbanchs.com/documents/THFEL_PR15.pdf. [cit. 2025-05-08].
- [6] BRABAZON, A.; O'NEILL, M. a MCGARRAGHY, S. *Natural Computing Algorithms*. 1. vyd. Berlin: Springer, 2015. ISBN 978-3-662-43630-1.
- [7] BURASHNIKOV, E. Branch-and-Bound and Dynamic Programming Approaches for the Knapsack Problem. *Operations Research Forum* online. 1. vyd., 2024, sv. 5, č. 4, s. 98. ISSN 2662-2556. Dostupné z: <https://doi.org/10.1007/s43069-024-00372-2>. [cit. 2025-05-12].
- [8] CACCHIANI, V.; IORI, M.; LOCATELLI, A. a MARTELLO, S. Knapsack problems — An overview of recent advances. Part I: Single knapsack problems. *Computers & Operations Research* online. 1. vyd., 2022, sv. 143, č. 1, s. 105692. ISSN 0305-0548. Dostupné z: <https://doi.org/10.1016/j.cor.2021.105692>. [cit. 2025-01-21].
- [9] CACCHIANI, V.; IORI, M.; LOCATELLI, A. a MARTELLO, S. Knapsack problems — An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems. *Computers & Operations Research* online. 1. vyd., 2022, sv. 143, č. 1, s. 105693. ISSN 0305-0548. Dostupné z: <https://doi.org/https://doi.org/10.1016/j.cor.2021.105693>. [cit. 2025-01-21].

- [10] CASTRO, L. N. de. *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. 1. vyd. Boca Raton: CRC Press, 2007. COMPUTER and INFORMATION SCIENCE. ISBN 978-1-4200-1144-9.
- [11] CHEN, Y.; BELL, M. a BOGENBERGER, K. Reliable pre-trip multi-path planning and dynamic adaptation for a centralized road navigation system. In: IEEE. *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005*. online. 2005, sv. 1, č. 1, s. 257–262. ISBN 0-7803-9215-9. Dostupné z: <https://doi.org/10.1109/ITSC.2005.1520057>. [cit. 2025-05-05].
- [12] CHEN, Z. a LUO, P. QISA: Incorporating quantum computation into Simulated Annealing for optimization problems. In: IEEE. *2011 IEEE Congress of Evolutionary Computation (CEC)* online. Institute of Electrical and Electronics Engineers, 2011, s. 2480–2487. ISBN 978-1-4244-7835-4. Dostupné z: <https://doi.org/10.1109/CEC.2011.5949925>. [cit. 2025-05-04].
- [13] CLERC, M. a KENNEDY, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* online. 1. vyd., 2002, sv. 6, č. 1, s. 58–73. Dostupné z: <https://doi.org/10.1109/4235.985692>. [cit. 2025-05-08].
- [14] CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. a STEIN, C. *Introduction to algorithms*. 4. vyd. Cambridge, Massachusetts: The MIT Press, 2022. ISBN 9780262046305.
- [15] EIBEN, A. E. a SMITH, J. E. *Introduction to Evolutionary Computing*. 2. vyd. Heidelberg: Springer Berlin, 2015. Natural Computing Series. ISBN 978-3-662-44874-8.
- [16] EZUGWU, A. E.; PILLAY, V.; HIRASEN, D.; SIVANARAIN, K. a GOVENDER, M. A Comparative Study of Meta-Heuristic Optimization Algorithms for 0 – 1 Knapsack Problem: Some Initial Results. *IEEE Access* online. 1. vyd., 2019, sv. 7, č. 1, s. 43979–44001. Dostupné z: <https://doi.org/10.1109/ACCESS.2019.2908489>. [cit. 2025-05-12].
- [17] FRÉVILLE, A. The multidimensional 0-1 knapsack problem: An overview. *European Journal of Operational Research* online. 1. vyd., 2004, sv. 155, č. 1, s. 1–21. ISSN 0377-2217. Dostupné z: [https://doi.org/10.1016/S0377-2217\(03\)00274-1](https://doi.org/10.1016/S0377-2217(03)00274-1). [cit. 2025-01-21].
- [18] GHAHRAMANI, S. *Fundamentals of probability with stochastic processes*. 3. vyd. Upper Saddle River, New Jersey: Pearson Prentice Hall, 2015. ISBN 0-13-145340-8.
- [19] GUILMEAU, T.; CHOUZENOUX, E. a ELVIRA, V. Simulated Annealing: a Review and a New Scheme. In: IEEE. *2021 IEEE Statistical Signal Processing Workshop (SSP)* online. 2021, s. 101–105. ISBN 978-1-7281-5767-2. Dostupné z: <https://doi.org/10.1109/SSP49050.2021.9513782>. [cit. 2025-05-12].
- [20] HAN, K.-H. a KIM, J.-H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation* online. 1. vyd., 2002, sv. 6, č. 6, s. 580–593. ISSN 1941-0026. Dostupné z: <https://doi.org/10.1109/TEVC.2002.804320>. [cit. 2025-01-22].

- [21] HAN, K.-H. a KIM, J.-H. Quantum-inspired evolutionary algorithms with a new termination criterion, H/sub /spl epsi// gate, and two-phase scheme. *IEEE Transactions on Evolutionary Computation* online. 1. vyd., 2004, sv. 8, č. 2, s. 156–169. Dostupné z: <https://doi.org/10.1109/TEVC.2004.823467>. [cit. 2025-05-10].
- [22] HUGHES, C.; ISAACSON, J.; PERRY, A.; SUN, R. F. a TURNER, J. *Quantum Computing for the Quantum Curious*. 1. vyd. Germany: Springer International Publishing, 2021. ISBN 978-3-030-61601-4.
- [23] IMMANUEL, S. D. a CHAKRABORTY, U. K. Genetic Algorithm: An Approach on Optimization. In: IEEE. *2019 International Conference on Communication and Electronics Systems (ICCES)* online. 2019, s. 701–708. ISBN 978-1-7281-1261-9. Dostupné z: <https://doi.org/10.1109/ICCES45898.2019.9002372>. [cit. 2025-05-12].
- [24] JAMES KENNEDY, R. C. E. a SHI, Y. *Swarm Intelligence*. 1. vyd. Academic Press, 2001. The Morgan Kaufmann Series in Evolutionary Computation. ISBN 978-1-55860-595-4.
- [25] KELLERER, H.; PFERSCHY, U. a PISINGER, D. *Knapsack problems*. 1. vyd. Berlin: Springer, 2004. ISBN 978-3-642-07311-3.
- [26] LANDSMAN, K. *Foundations of Quantum Theory: From Classical Concepts to Operator Algebras*. 1. vyd. Springer Cham, 2017. Fundamental Theories of Physics. ISBN 978-3-319-51777-3.
- [27] LI, Y.; TIAN, M.; LIU, G.; PENG, C. a JIAO, L. Quantum Optimization and Quantum Learning: A Survey. *IEEE Access* online. 1. vyd., 2020, sv. 8, č. 1, s. 23568–23593. Dostupné z: <https://doi.org/10.1109/ACCESS.2020.2970105>. [cit. 2025-01-26].
- [28] MIRJALILI, S.; FARIS, H. a ALJARAHA, I. *Evolutionary Machine Learning Techniques: Algorithms and Applications*. 1. vyd. Singapore: Springer Nature Singapore, 2020. Algorithms for Intelligent Systems. ISBN 978-981-32-9992-4.
- [29] MORRISON, D. R.; JACOBSON, S. H.; SAUPPE, J. J. a SEWELL, E. C. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization* online. 1. vyd., 2016, sv. 19, č. 1, s. 79–102. ISSN 1572-5286. Dostupné z: <https://doi.org/10.1016/j.disopt.2016.01.005>. [cit. 2025-05-12].
- [30] NIELSEN, M. A. a CHUANG, I. L. *Quantum Computation and Quantum Information*. 1. vyd. Cambridge: Cambridge University Press, 2010. ISBN 978-1-107-00217-3.
- [31] NOURANI, Y. a ANDRESEN, B. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General* online. 1. vyd., 1998, sv. 31, č. 41. Dostupné z: <https://doi.org/10.1088/0305-4470/31/41/011>. [cit. 2025-05-08].
- [32] PAULUS, J. a KLAPURI, A. Music Structure Analysis Using a Probabilistic Fitness Measure and a Greedy Search Algorithm. *IEEE Transactions on Audio, Speech, and Language Processing* online. 1. vyd., 2009, sv. 17, č. 6, s. 1159–1170. Dostupné z: <https://doi.org/10.1109/TASL.2009.2020533>. [cit. 2025-05-12].

- [33] RADHA, B. a RUGHOOPUTH, H. C. S. Optimal network reconfiguration of electrical distribution systems using real coded quantum inspired evolutionary algorithm. In: IEEE. *2010 International Conference on Networking, Sensing and Control (ICNSC)* online. 2010, s. 38–43. ISBN 978-1-4244-6453-1. Dostupné z: <https://doi.org/10.1109/ICNSC.2010.5461550>. [cit. 2025-01-26].
- [34] RIEFFEL, E. G. a POLAK, W. H. *Quantum Computing: A Gentle Introduction*. 1. vyd. Cambridge, Massachusetts: MIT Press, 2011. ISBN 978-0-262-01506-6.
- [35] RINGBAUER, M.; BIGGERSTAFF, D. N.; BROOME, M. A.; FEDRIZZI, A.; BRANCIARD, C. et al. Experimental Joint Quantum Measurements with Minimum Uncertainty. *Physical Review Letter*. 1. vyd. American Physical Society, Leden 2014, sv. 112, č. 2, s. 020401. ISSN 1079-7114. Dostupné z: <https://link.aps.org/doi/10.1103/PhysRevLett.112.020401>.
- [36] ROZENBERG, G.; BÄCK, T. a KOK, J. N. *Handbook of Natural Computing*. 1. vyd. Berlin: Springer, 2012. ISBN 978-3-540-92909-3.
- [37] SHAMI, T. M.; EL SALEH, A. A.; ALSWAITTI, M.; AL TASHI, Q.; SUMMAKIEH, M. A. et al. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* online. 1. vyd., 2022, sv. 10, č. 1, s. 10031–10061. Dostupné z: <https://doi.org/10.1109/ACCESS.2022.3142859>. [cit. 2025-05-08].
- [38] SHI, Y. a EBERHART, R. A modified particle swarm optimizer. In: IEEE. *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* online. 1998, s. 69–73. ISBN 0-7803-4871-0. Dostupné z: <https://doi.org/10.1109/ICEC.1998.699146>. [cit. 2025-05-08].
- [39] SHOR, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review* online. 1. vyd., 1999, sv. 41, č. 2, s. 303–332. Dostupné z: <https://doi.org/10.1137/S0036144598347011>. [cit. 2025-05-04].
- [40] SU, S.; LI, X.; TANG, T. a GAO, Z. A Subway Train Timetable Optimization Approach Based on Energy-Efficient Operation Strategy. *IEEE Transactions on Intelligent Transportation Systems* online. 1. vyd., 2013, sv. 14, č. 2, s. 883–893. Dostupné z: <https://doi.org/10.1109/TITS.2013.2244885>. [cit. 2025-05-05].
- [41] TALBI, E. *Metaheuristics: From Design to Implementation*. 1. vyd. Wiley, 2009. Wiley Series on Parallel and Distributed Computing. ISBN 978-0-470-49690-9.
- [42] TAYARANI, N. M. H.; BEHESHTI, M.; SABET, J. a MOHAMMADI, H. A statistical analyzing approach for Quantum Evolutionary Algorithms. In: IEEE. *2011 19th Iranian Conference on Electrical Engineering* online. 2011, sv. 1, č. 1, s. 1–6. ISBN 978-964-463-428-4. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/5955671>. [cit. 2025-05-10].
- [43] VAZIRANI, V. V. *Approximation Algorithms*. 2. vyd. Springer, 2010. ISBN 978-3-642-08469-0.

- [44] WANG, Y.; FENG, X.-Y.; HUANG, Y.-X.; PU, D.-B.; ZHOU, W.-G. et al. A novel quantum swarm evolutionary algorithm and its applications. *Neurocomputing* online. 1. vyd., 2007, sv. 70, č. 4, s. 633–640. ISSN 0925-2312. Dostupné z: <https://doi.org/https://doi.org/10.1016/j.neucom.2006.10.001>. [cit. 2025-05-06].
- [45] WANG, Y. a SHI, Y. The application of quantum-inspired evolutionary algorithm in analog evolvable hardware. In: IEEE. *2010 The 2nd Conference on Environmental Science and Information Application Technology* online. 2010, sv. 2, s. 330–334. ISBN 978-1-4244-7390-8. Dostupné z: <https://doi.org/10.1109/ESIAT.2010.5567359>. [cit. 2025-01-26].
- [46] YE, W.; LIU, R.; LI, Y. a JIAO, L. Quantum-Inspired Evolutionary Algorithm for Convolutional Neural Networks Architecture Search. In: IEEE. *2020 IEEE Congress on Evolutionary Computation (CEC)* online. 2020, s. 1–8. ISBN 978-1-7281-6929-3. Dostupné z: <https://doi.org/10.1109/CEC48606.2020.9185727>. [cit. 2025-01-26].
- [47] ZHANG, G. Quantum-inspired evolutionary algorithms: a survey and empirical study. *Journal of Heuristics* online. 1. vyd., 2011, sv. 17, č. 3, s. 303–351. ISSN 1572-9397. Dostupné z: <https://doi.org/10.1007/s10732-010-9136-0>. [cit. 2025-01-26].
- [48] ZHAO, J.; HUANG, T.; PANG, F. a LIU, Y. Genetic Algorithm Based on Greedy Strategy in the 0-1 Knapsack Problem. In: IEEE. *2009 Third International Conference on Genetic and Evolutionary Computing* online. 2009, sv. 1, č. 1, s. 105–107. ISBN 978-1-4244-5245-3. Dostupné z: <https://doi.org/10.1109/WGEC.2009.43>. [cit. 2025-05-10].
- [49] ZHOU, Y.; SHI, Y.; WEI, Y.; LUO, Q. a TANG, Z. Nature-inspired algorithms for 0-1 knapsack problem: A survey. *Neurocomputing* online. 1. vyd., 2023, sv. 554, č. 1, s. 126630. ISSN 0925-2312. Dostupné z: <https://doi.org/10.1016/j.neucom.2023.126630>. [cit. 2025-01-21].

Příloha A

Struktura odevzdaných souborů

— DIP/	Složka se soubory pro překlad diplomové práce práce
— SRC/	Složka se zdrojovými soubory
— datasets/	Složka s instancemi problému batohu
— outputs/	Složky se získanými daty
— graphs/	Složka s vygenerovanými grafy
— scripts/	Složka se skripty s experimenty
— src/	Složka se zdrojovými soubory
— plots/	Složka se zdrojovými soubory pro generování grafů
— qiea/	Složka se zdrojovými soubory algoritmů
— plots.py	Vstupní bod pro generování grafů
— qiea.py	Vstupní bod pro spuštění algoritmů
— Makefile	Makefile pro jednodušší práci s projektem
— README.md	Soubor popisující základní práci se zdrojovými programy
— thesis.pdf	Text diplomové práce

Příloha B

Manuál

Uvažujme kořenový adresář projektu. Základní nápovědu k programu lze zobrazit příkazem:

```
$ python3 src/qiea.py --help
```

Algoritmus lze spustit s výchozími hodnotami parametrů následovně:

```
$ python3 src/qiea.py qiga datasets/100.txt
```

Pro uložení výsledků do zvoleného výstupního adresáře:

```
$ python3 src/qiea.py qiga datasets/100.txt --output outputs/
```

V případě paralelního spouštění je možné připojovat výsledky k již existujícím výstupům:

```
$ python3 src/qiea.py qiga datasets/100.txt --output outputs/ --append_results
```

Pomocí přiloženého souboru `Makefile` lze snadno spouštět skripty pro experimenty.

```
$ make
```

Pro spuštění experimentů s konkrétním algoritmem:

```
$ make exp-qiga
```

A pro konkrétní instanci:

```
$ make exp-qiga-100
```

Ze získaných dat lze vygenerovat grafy příkazem:

```
$ make graphs
```

Pro generování grafů pouze pro vybrané algoritmy a instance:

```
$ python3 src/plots.py outputs/ --save_dir graphs/ --algorithm qiga qisa --instance 100 250
```

Více informací se nachází v přiloženém souboru `README.md`

Seznam argumentů programu Quantum Inspired Evolutionary Algorithms (`qiea.py`)

- `algorithm` – Typ algoritmu, který se má spustit.
 - Možnosti: `qiga`, `qisa`, `qse`, `qipso`.
- `input` – Cesta k vstupnímu datasetu.
- `--append_results` – Pokud je nastaveno, výsledky budou připojovány do výstupního souboru po každém experimentu.
- `--output` – Cesta k adresáři, kam budou uloženy výsledky.
- `--experiments` – Počet nezávislých spuštění algoritmu (výchozí: 30).
- `--population` – Velikost populace (výchozí: 1).
- `--evaluations` – Počet vyhodnocení fitness funkce (výchozí: 10000).
- `--theta` – Úhel kvantové rotace pro QIEA (výchozí: 0,01).
- `--c1` – Kognitivní koeficient pro QIPSO (výchozí: 0,05).
- `--c2` – Sociální koeficient pro QIPSO (výchozí: 0,15).
- `--omega` – Parametr ω pro QIPSO (výchozí: 0,01).
- `--velocity` – Počáteční rychlost pro QIPSO a QSE. (výchozí: 100).
- `--cooling_rate` – Rychlost ochlazování pro *QISA* (výchozí: 0,98).
- `--cooling` – Metoda ochlazování pro *QISA* (výchozí: `rec-logarithmic`).
 - Možnosti: `exponential`, `linear`, `logarithmic`, `rec-logarithmic`.
- `--observation` – Metoda tepelně-řízeného pozorování pro *QISA*. (výchozí: `sigmoid`).
 - Možnosti: `constant`, `sigmoid`.

Seznam argumentů programu pro vykreslení grafů (`plots.py`)

- `source_dir` – Cesta ke zdrojové složce obsahující výsledky experimentů.
- `--save_dir` – Cesta, kam se mají uložit vykreslené grafy. Pokud není zadáno, grafy se pouze zobrazí.
- `--algorithm` – Typ algoritmu, pro který se mají vykreslit výsledky. Lze zadat více hodnot oddělených mezerou.
 - Možnosti: `qiga`, `qisa`, `qse`, `qipso`.
- `--instance` – Velikosti instancí problému, které se mají zpracovat. Lze zadat více hodnot oddělených mezerou.