



Politechnika Łódzka

Wydział Elektrotechniki, Elektroniki,
Informatyki i Automatyki

Instytut Informatyki Stosowanej



Rozprawa doktorska

Analiza własności kwantowo inspirowanych algorytmów ewolucyjnych

ROBERT NOWOTNIAK

Promotor: dr hab. inż. Jacek Kucharski, prof. PŁ

Łódź, 2015

PODZIĘKOWANIA

Serdeczne podziękowania za poświęcony czas, cierpliwość, wyrozumiałość i zaufanie oraz za wszelkie bezcenne uwagi, przekazane mi w trakcie powstawania niniejszej pracy, pragnę złożyć promotorowi mojej rozprawy **dr. hab. inż. Jackowi Kucharskiemu, prof. PŁ.**

Dziękuję również wszystkim moim nauczycielom i wykładowcom, w tym kadrze Wydziału Elektrotechniki, Elektroniki, Informatyki i Automatyki oraz Wydziału Fizyki Technicznej, Informatyki i Matematyki Stosowanej Politechniki Łódzkiej. Ze szczególną wdzięcznością wspominam śp. prof. Marka Rudnickiego, którego wykłady wzbudziły moje zainteresowanie Sztuczną Inteligencją na długie lata.

STRESZCZENIE

Prezentowana rozprawa dotyczy nowej klasy metod ewolucyjnych Sztucznej Inteligencji, czerpiących inspirację zarówno z ewolucji biologicznej, jak i z teoretycznych podstaw Informatyki Kwantowej (kubity, superpozycja stanów, kwantowe bramki logiczne).

W rozprawie wykonano adaptację wybranych, znanych metod teoretycznej analizy algorytmów ewolucyjnych do obszaru algorytmów kwantowo inspirowanych. Przeprowadzono analizę zbieżności kwantowo inspirowanego algorytmu ewolucyjnego, wykorzystującą twierdzenie Banacha o punkcie stałym odwzorowania zwężającego. Zostało również rozważone pojęcie dopasowania schematów w kwantowo inspirowanym algorytmie genetycznym. W wyniku przeprowadzonej analizy możliwe było poznanie zależności pomiędzy binarnymi chromosomami kwantowymi i schematami oraz sformułowanie nowych pojęć rzędu kwantowości oraz współczynników kwantowości algorytmu. Zdefiniowano przestrzeń, pozwalającą na przejście od znanych dotychczas algorytmów kwantowo inspirowanych (rzędu I) do pełnych algorytmów kwantowych. W oparciu o wprowadzone pojęcia teoretyczne opracowano autorski kwantowo inspirowany algorytm genetyczny rzędu II. Wykazano również, że wykorzystanie heurystycznych technik metaoptymalizacji oraz implementacji algorytmów w środowisku obliczeń równoległych pozwala na dobór właściwych wartości parametrów, zapewniających efektywność algorytmów.

W części doświadczalnej zweryfikowano efektywność autorskiego kwantowo inspirowanego algorytmu genetycznego rzędu II oraz wykazano, że wykorzystanie II rzędu kwantowości jest metodą poprawy efektywności kwantowo inspirowanego algorytmu genetycznego w szczególności w zwodniczych zadaniach optymalizacji kombinatorycznej. Dzięki implementacji algorytmów w środowisku obliczeń równoległych oraz zastosowaniu technik metaoptymalizacji zidentyfikowany został zakres parametrów kwantowo inspirowanego algorytmu ewolucyjnego, zapewniających efektywność tego algorytmu dla zadań optymalizacji numerycznej.

SPIS TREŚCI

Streszczenie

Wykaz ważniejszych skrótów i oznaczeń	4
---------------------------------------	---

Wstęp	8
-------	---

Cel, zakres i tezy rozprawy	14
-----------------------------	----

I Aktualny stan wiedzy	18
-------------------------------	-----------

1 Algorytmy ewolucyjne	19
-------------------------------	-----------

1.1 Optymalizacja obliczeniowa	20
1.2 Rodzaje algorytmów ewolucyjnych	21
1.3 Wybrane elementy algorytmów ewolucyjnych	22
1.4 Metody analiz teoretycznych	25
1.5 Techniki metaoptymalizacji	27

2 Teoretyczne podstawy Informatyki Kwantowej	31
---	-----------

2.1 Kubity, zbiory kubitów i rejestry kwantowe	33
2.2 Kwantowe stany splątane	39
2.3 Podstawowe kwantowe bramki logiczne	40

3 Kwantowo inspirowane algorytmy genetyczne i ewolucyjne	45
---	-----------

3.1 Elementy kwantowe w algorytmie ewolucyjnym	45
3.2 Algorytmy QIEA – wybrane pojęcia podstawowe	47
3.2.1 Binarny gen kwantowy i jego przestrzeń stanów	49
3.2.2 Populacja kwantowa Q	50
3.2.3 Podstawowe kwantowe operatory genetyczne	50
3.3 Przegląd literatury tematu	52
3.4 Algorytmy pierwotne	60
3.4.1 Kwantowo inspirowany algorytm genetyczny QIGA1	60
3.4.2 Kwantowo inspirowany algorytm ewolucyjny iQIEA	64

II	Analiza teoretyczna	70
4	Własności podstawowe	71
4.1	Zbieżność algorytmów QIEA	71
4.1.1	Założenia twierdzenia Banacha dla algorytmów QIEA	71
4.1.2	Przestrzeń metryczna $\langle S, \delta \rangle$ populacji kwantowych Q	72
4.1.3	Zupełność przestrzeni $\langle S, \delta \rangle$	73
4.1.4	Odwzorowanie w przestrzeni $\langle S, \delta \rangle$	73
4.1.5	Generacja populacji kwantowej Q jako kontrakcja . .	74
4.1.6	Ogólne warunki zbieżności algorytmu QIEA	74
4.2	Schematy w algorytmach QIGA	77
4.2.1	Schematy w binarnym chromosomie kwantowym . .	77
4.2.2	Warunkowe dopasowanie do schematu	78
4.2.3	Wartość oczekiwana liczby dopasowań	80
4.2.4	Wariancja dopasowań schematu	81
4.3	Podsumowanie i wnioski	83
5	Algorytmy kwantowo inspirowane wyższych rzędów	85
5.1	Przestrzenie działania algorytmów	85
5.1.1	Przestrzeń ciągów binarnych	87
5.1.2	Przestrzeń schematów	87
5.1.3	Przestrzeń stanów binarnego chromosomu kwantowego	90
5.1.4	Przestrzeń stanów rejestru kwantowego	93
5.2	Podstawy algorytmów wyższych rzędów	96
5.2.1	Struktura algorytmu wyższego rzędu	96
5.2.2	Rząd kwantowości algorytmu	97
5.2.3	Współczynniki kwantowości algorytmu	98
5.2.4	Przestrzenie działania algorytmów wyższych rzędów	103
5.2.5	Algorytmy wybranych rzędów dla rozmiaru zadania N = 10	106
5.3	Autorski algorytm QIGA rzędu II	111
5.3.1	Reprezentacja rozwiązań	111
5.3.2	Kwantowe operatory genetyczne rzędu II	113
5.4	Podsumowanie i wnioski	117

III Badania doświadczalne	119
6 Algorytmy QIGA dla zadań optymalizacji kombinatorycznej	120
6.1 Procedura badawcza	120
6.2 Zestawy zadań testowych	122
6.3 Efektywna implementacja QIGA1 w technologii GPGPU . .	125
6.4 Metaoptymalizacja algorytmu QIGA1	128
6.5 Ocena efektywności algorytmu QIGA1*	133
6.6 Efektywność autorskiego algorytmu QIGA2	136
6.7 Podsumowanie i wnioski	139
7 Algorytmy QIEA dla zadań optymalizacji numerycznej	140
7.1 Procedura badawcza	141
7.2 Efektywna implementacja algorytmu iQIEA na klastrze . . .	142
7.3 Metaoptymalizacja algorytmu iQIEA	144
7.4 Wyniki – Porównanie efektywności algorytmów	151
7.5 Podsumowanie i wnioski	157
Podsumowanie i wnioski	158
Załączniki	162
A Postulaty mechaniki kwantowej	163
B Twierdzenie Banacha o punkcie stałym	165
C Wybrane szczegóły implementacyjne algorytmów	167
D Wyniki szczegółowe badań doświadczalnych	175
Bibliografia	178
Spis tabel	209
Spis rysunków	211
Skorowidz	211

WYKAZ WAŻNIEJSZYCH SKRÓTÓW I OZNACZEŃ

Przyjęte oznaczenia algorytmów oraz symboli matematycznych, stosowane w rozprawie, zostały zaprezentowane odpowiednio w Tabeli 1 oraz 2.

TABELA 1: Oznaczenie algorytmów

Skrót	Znaczenie
EA	Algorytm ewolucyjny (ang. <i>Evolutionary Algorithm</i>)
SGA	Prosty algorytm genetyczny (Holland, 1975) (ang. <i>Simple Genetic Algorithm</i>)
PSO	Metoda roju cząstek (Kennedy, Eberhart, 1995) (ang. <i>Particle Swarm Optimization</i>)
ACO	algorytm mrówkowy (Coloni et al., 1991) (ang. <i>Ant Colony Optimization</i>)
ES	Strategia ewolucyjna (ang. <i>Evolutionary Strategy</i>)
LUS	algorytm <i>Local Unimodal Sampling</i> (Pedersen, Chipperfield, 2008)
Nelder-Mead	sympleksowa metoda spadku (Nelder, Mead, 1965)
FA	Algorytm świetlika (Yang, 2008) (ang. <i>Firefly Algorithm</i>)
CMAES	Strategia ewolucyjna z adaptacyjną macierzą kowariancji (Hansen, Ostermeier, 1996) (ang. <i>Covariance Matrix Adaptation Evolutionary Strategy</i>)
QEA	Kwantowy algorytm ewolucyjny (wymagający komputera kwantowego)
QIEA	Kwantowo inspirowany algorytm ewolucyjny (ang. <i>Quantum-Inspired Evolutionary Algorithm</i>)
QIGA	Kwantowo inspirowany algorytm genetyczny (ang. <i>Quantum-Inspired Genetic Algorithm</i>)

TABELA 1: Oznaczenie algorytmów c.d.

Skrót	Znaczenie
QIGA1	Pierwotny kwantowo inspirowany algorytm genetyczny (Han, Kim, 2000)
QIGA1*	Dostrojony pierwotny kwantowo inspirowany algorytm genetyczny (Nowotniak, Kucharski, 2012b)
QIGA2	Autorski kwantowo inspirowany algorytm genetyczny rzędu II (Nowotniak, Kucharski, 2014)
iQIEA	Algorytm QIEA z „kodowaniem impulsowym” (da Cruz <i>et al.</i> , 2006)

TABELA 2: Oznaczenie symboli

Symbol	Znaczenie
$\mathbb{N}, \mathbb{R}, \mathbb{C}, \mathcal{H}$	zbiory liczb: naturalnych (z zerem), rzeczywistych, zespolonych; przestrzeń Hilberta
X	przestrzeń rozwiązań zadania optymalizacji lub przeszukiwania
$x \in X$	dopuszczalne rozwiązanie zadania (fenotyp)
$x^* \in X$	najlepsze istniejące rozwiązanie zadania optymalizacji lub przeszukiwania, $x^* = \arg \max_{x \in X} f(x)$
$f : X \mapsto \mathbb{R}$	funkcja celu zadania optymalizacji
$\tilde{f}_{\mathcal{A}} : \mathbb{R}^n \mapsto \mathbb{R}$	wartość metaprzystosowania algorytmu $\mathcal{A}$ dla danego n -elementowego wektora parametrów
$\epsilon \in \mathbb{R}$	precyzja obliczeń (domyślnie $\epsilon = 10^{-6}$)
$t \in \mathbb{N}$	numer generacji w algorytmie ewolucyjnym
$t_{\max} \in \mathbb{N}^+$	maksymalna liczba generacji w algorytmie
q	chromosom kwantowy (genotyp monoploidalny)
A	zbiór alleli (domyślnie $A = \{0, 1\}$)
$\mathcal{F}_A$	σ -ciało zbioru A ; Dla $A = \{0, 1\}$: $\mathcal{F}_A = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$
$\Omega_0 = \Omega = \{0, 1\}^N$	przestrzeń N -elementowych ciągów binarnych

TABELA 2: Oznaczenie symboli c.d.

Symbol	Znaczenie
$\Omega_H = \{0, 1, *\}^N$	przestrzeń schematów długości N
$H \in \Omega_H$	schemat, tj. napis nad alfabetem ternarym $\{0, 1, *\}$
$o(H) \in \mathbb{N}$	rzęd schematu H (ang. <i>schema order</i>)
$d(H) \in \mathbb{N}$	rozpiętość schematu H
$M(q, H)$	prawdopodobieństwo dopasowania binarnego chromosomu kwantowego q do schematu H
Q	populacja kwantowa (ang. <i>quantum population</i>)
P	zaobserwowana w danej generacji t populacja klasyczna (ang. <i>classical population</i>)
$ Q , P \in \mathbb{N}^+$	liczność populacji Q i P
$N \in \mathbb{N}^+$	rozmiar zadania / długość chromosomów
$\mathbb{S}$	zbiór wszystkich możliwych populacji kwantowych Q przy ustalonej reprezentacji rozwiązań
$\delta : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}$	miara odległości między populacjami kwantowymi
$G : \mathbb{S} \mapsto \mathbb{S}$	odwzorowanie przekształcające daną populację kwantową $Q(t)$ w jej kolejną generację $Q(t+1)$
$r \in \mathbb{N}^+$	rzęd kwantowości algorytmu (ang. <i>quantum order</i>)
$w \in [0, 1]$	względny rzęd kwantowości algorytmu (ang. <i>relative quantum order</i>)
$\lambda \in [0, 1]$	współczynnik kwantowości algorytmu (ang. <i>quantum factor</i>)
$\lambda_O \in [0, 1]$	współczynnik kwantowości operatorowej algorytmu
$ 0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \odot \\ 0 \end{pmatrix}$	stan bazowy kubitów (binarnego genu kwantowego)
$ 1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \uparrow \end{pmatrix}$	stan bazowy kubitów (binarnego genu kwantowego)
$\alpha, \beta \in \mathbb{R}$	amplitudy prawdopodobieństw binarnego genu kwantowego (ang. <i>probability amplitudes</i>)
HAD	kwantowa bramka Hadamarda

TABELA 2: Oznaczenie symboli c.d.

Symbol	Znaczenie
NOT	kwantowa bramka negacji
ROT($\Delta\theta$)	kwantowa bramka obrotu
$\Delta\theta \in \mathbb{R}$	kąt obrotu w przestani stanów kubitów
$P_c \in [0, 1]$	prawdopodobieństwo rekombinacji
$P_m \in [0, 1]$	prawdopodobieństwo mutacji
$\xi \in [0, 1]$	współczynnik krzyżowania w algorytmie iQIEA
$\delta \in [0, 1]$	współczynnik kontrakcji w algorytmie iQIEA
$\mu \in [0, 1]$	współczynnik kontrakcji amplitud w QIGA2
NoFE	liczba wywołań funkcji oceny
$E[L]$	Wartość oczekiwania zmiennej losowej L
$V[L]$	Wariancja zmiennej losowej L

WSTĘP

Sztuczna Inteligencja i Mechanika Kwantowa to dwie niezależne dziedziny informatyki i fizyki, będące wielkimi osiągnięciami nauki XX wieku i jednocześnie wciąż kryjące w sobie liczne tajemnice oraz ciągłe wyzwania dla przyszłego rozwoju ludzkiej cywilizacji. Przedmiotem badań podjętych w prezentowanej rozprawie są **kwantowo inspirowane algorytmy genetyczne i ewolucyjne**, które sytuują się na pograniczu tych dwóch dziedzin wiedzy. Niniejsza praca wpisuje się w dążenie do poprawy efektywności metod Sztucznej Inteligencji poprzez wprowadzenie do nich pojęć i reguł, wynikających z teorii Mechaniki Kwantowej.

Mechanika Kwantowa (ang. *Quantum Mechanics* – QM) powstała na początku minionego wieku i jest uznawana za jedno z największych i przełomowych osiągnięć współczesnej fizyki. Dostarczając najdokładniejszego znanego obecnie opisu fizycznej rzeczywistości (w skali mikroświata), pozwoliła w spójny sposób wyjaśnić wiele niezrozumiałych wcześniej zjawisk, np. promieniowanie ciała doskonale czarnego (Planck, 1901), efekt fotoelektryczny zewnętrzny (Einstein, 1905), zjawisko Comptona (Compton, 1923). Na bazie Mechaniki Kwantowej powstały nowe hipotezy i teorie, dzięki którym m.in. możliwe było ujednoczenie wybranych podstawowych oddziaływań fizycznych (teoria oddziaływań elektroslabych (Salam, 1980), teoria wielkiej unifikacji (Buras *et al.*, 1978), koncepcja pola Higgsa (Higgs, 1964), Model Standardowy (Novaes, 2000)). Mechanika kwantowa wywarła również wpływ na rozwój poglądów filozoficznych, m.in. podważając koncepcję *determinizmu* fizycznej rzeczywistości (Laplace, 1814) i przybliżając nas do zrozumienia natury Wszechświata (Hawking, 2007).

Ponad 50 lat od powstania i powszechnego zaakceptowania teorii kwantowej zaczęto rozważać możliwość wykorzystania kwantowomechanicznego opisu fizycznej rzeczywistości w celu uzyskania zupełnie nowego, innego spojrzenia na teorię informacji i na procesy obliczeniowe (Feynman, 1982). W ten sposób powstała nowa dziedzina, **Informatyka Kwantowa** (ang. *Quantum Computing*

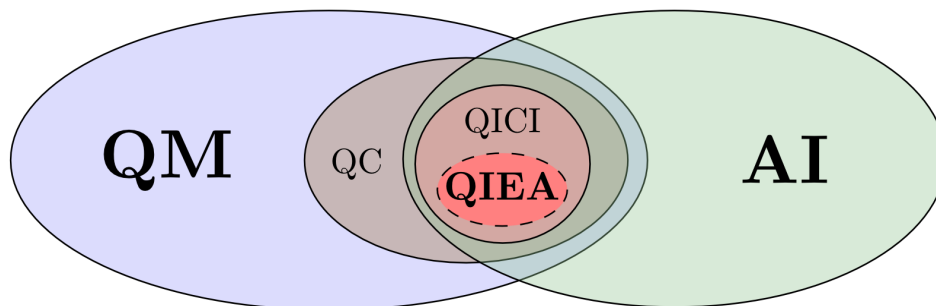
– QC) (Nielsen, Chuang, 2000), zajmująca się wykorzystaniem potencjalnych możliwości obliczeniowych miniaturowych układów (o skali atomowej lub subatomowej) oraz praw, którym one podlegają. Pomimo ciągłych badań i regularnych doniesień o kolejnych postępach (Xu et al., 2011; Bao et al., 2012), do tej pory nie udało się zbudować rzeczywistego skalowalnego komputera kwantowego i zadanie to pozostaje wciąż wielkim wyzwaniem. Według niektórych aktualnych prognoz może to jednak nastąpić w ciągu najbliższej dekady (Pla et al., 2012). Tworzenie algorytmów dla komputerów kwantowych jest wymagającym zadaniem ze względu na nieintuicyjne efekty Mechaniki Kwantowej, na których opierałaby się praca takiej maszyny obliczeniowej. Jak dotąd znanych jest tylko kilka użytecznych algorytmów kwantowych (Hirvensalo, 2004). Ponieważ komputer kwantowy m.in. redukowałby klasy złożoności niektórych problemów obliczeniowych (np. problem faktoryzacji liczb (Shor, 1994), przeszukiwanie zbiorów (Grover, 1996), sukces w jego budowie wniósłby nową jakość do informatyki i teorii obliczeń. Jednocześnie taki postęp prawdopodobnie otworzyłby nowy obszar kolejnych wyzwań dla nauki. Jeśli jednak prawdziwego komputera kwantowego nie uda się nigdy zbudować (bo być może definitywnie uniemożliwiają to fizyczne prawa Wszechświata), dotychczasowe badania nad obliczeniami kwantowymi pozostają przydatne same w sobie (Penrose, 2006, rozdział 23), dostarczając wielu inspirujących pojęć i zjawisk (amplitudy prawdopodobieństwa, kwantowe splątanie, teleportacja kwantowa, kodowanie supergęste, algorytmy kwantowe), oraz nowego spojrzenia na możliwości wykorzystania losowości, stochastyki i równoległości w wielu ważnych współcześnie obszarach informatyki.

Podobnie jak w fizyce tak samo w **Sztucznej Inteligencji** (ang. *Artificial Intelligence* – AI) (Russell et al., 2010), interdyscyplinarnym obszarze informatyki, wyraża się *dążenie człowieka do opisania, wykorzystania i wyjaśnienia natury*. Jednak w odróżnieniu od fizyki dziedzina ta skupia się na badaniu oraz próbie zrozumienia i naśladowania zachowania istot inteligentnych oraz różnorodności złożonych procesów obserwowanych w przyrodzie, np. ewolucji biologicznej (Holland, 1975), kolektywnego zachowania zwierząt (Eberhart et al., 2001), sposobu działania ludzkiego umysłu (Kurzweil,

2012) i wielu innych. W pierwszych latach badań nad Sztuczną Inteligencją czołowi badacze przewidywali, że w ciągu kilku dekad pojawią się maszyny dorównujące ludziom (Minsky, 1967). Kolejne dziesięciolecia nie przyniosły jednak spodziewanego przełomu i do chwili obecnej aktualne jest stwierdzenie, że Sztucznej Inteligencji w jej aktualnym stanie zaawansowania w sensie jakościowym *wciąż czegoś brakuje* (Minsky, 1988; Moravec, 1988). Problem ten, sytuujący się na bardzo złożonym tle, pozostaje nierozwiązany do dziś, jest przedmiotem ciągłej dyskusji (Penrose, Hameroff, 2011; Grygiel, Hohol, 2009), a jego różne aspekty były podejmowane m.in. przez przedstawicieli nauk filozoficznych (Lucas, 1961), fizycznych (Penrose, 1989) i biologicznych (Hameroff, 2006). Według niektórych hipotez, dla których podstawą jest m.in. fundamentalne dla logiki matematycznej twierdzenie Gödla o niezupełności formalnych systemów logicznych (Gödel, 1931), fenomen ludzkiego rozumowania, inteligencji i świadomości ma charakter *nieobliczeniowy* i *niealgorytmiczny*. Niektórzy badacze sugerują (Penrose, 1994; Penrose *et al.*, 1997), że istota inteligencji nie może być odtworzona wyłącznie na zasadzie stworzenia warunków do *emergentnego wyłonienia się* funkcji umysłu takich jak pojmowanie i świadomość, poprzez sam wzrost mocy obliczeniowych i stopnia skomplikowania algorytmów, ale wymagane może być włączenie innych jakościowo, wyższego rzędu zjawisk i mechanizmów, dostępnych np. na poziomie kwantowym.

Ten pogląd wraz z wpływem, jaki Mechanika Kwantowa wywarła na zrozumienie natury rzeczywistości, stanowiły motywację dla podejmowania prób tworzenia metod Sztucznej Inteligencji rozszerzonych o dodatkowe czynniki, wzorowane na pojęciach i regułach, którymi rządzą się systemy kwantowe. Dziesiątki przykładów (Manju, Nigam, 2012), zaprezentowanych w ciągu minionej dekady, wykazały, że nawet drobna „kwantowa inspiracja” (dodatkowe elementy losowości, oparte na teorii kwantowej) pozwala na poprawę efektywności metod inteligencji obliczeniowej w różnorodnych zastosowaniach.

Informatyka Kwantowa pozwoliła na stworzenie nowych i rozszerzonych metod, takich jak: kwantowo inspirowane algorytmy ewolucyjne (Zhang, 2010a), kwantowo inspirowane sieci neuronowe (Menneer, Narayanan, 1995; Allauddin *et al.*, 2008), kwantowo inspirowane metody inteli-



RYS. 1: Relacje pomiędzy poszczególnymi obszarami wiedzy na styku Mechaniki Kwantowej i Sztucznej Inteligencji.

QM – Mechanika Kwantowa, **QC** – Informatyka Kwantowa, **AI** – Sztuczna Inteligencja, **QICI** – kwantowo inspirowane metody inteligencji obliczeniowej, **QIEA** – kwantowo inspirowane algorytmy ewolucyjne.

gencji roju (Coelho *et al.*, 2008; Luitel, Venayagamoorthy, 2010), kwantowo inspirowane sztuczne systemy immunologiczne (Li *et al.*, 2011c) i wiele innych (Manju, Nigam, 2012; Nedjah *et al.*, 2008). Wyodrębniony został w ten sposób nowy obszar *quantum-inspired computational intelligence* (QICI) (Nedjah *et al.*, 2008; Zak, 2011), zaznaczony na Rysunku 1, w którym nie jest niezbędny komputer kwantowy, a poprawa efektywności metod inteligencji obliczeniowej jest możliwa dzięki wprowadzeniu do algorytmów nawet drobnych elementów, bazujących na teorii kwantowej. W szczególności Informatyka Kwantowa dostarcza nowych sposobów wykorzystania losowości i stochastyki dla poprawy efektywności zrandomizowanych, populacyjnych metaheurystyk przeszukiwania, do których należą algorytmy ewolucyjne.

Identyfikuje się obecnie następujące, traktowane dotąd jako rozłączne¹, możliwości wzajemnego uzupełniania się i łączenia metod ewolucyjnych

¹Zaprezentowane w Rozdziale 5 kwantowo inspirowane algorytmy wyższych rzędów są próbą unifikacji klasy algorytmów kwantowo inspirowanych oraz kwantowych — podejście odpowiednio 2) i 3) — i pozwala na przejście pomiędzy tymi klasami algorytmów.

Ponadto, możliwe są również innego rodzaju mieszane podejścia hybrydowe: np. projektowanie elementów „prawdziwych” algorytmów kwantowych (np. optymalizacja macierzy unitarnych) za pomocą kwantowo inspirowanego algorytmu ewolucyjnego. Takim podejściom nie poświęcano jednak dotąd uwagi w literaturze.

i Informatyki Kwantowej ([Zhang, 2010a](#); [Nowotniak, 2010](#)):

1) ALGORYTMY KWANTOWE PROJEKTOWANE EWOLUCYJNIE

– Projektowanie algorytmów dla rzeczywistych komputerów kwantowych, których użyteczne konstrukcje być może powstaną w przyszłości, jest trudnym zadaniem ze względu na nieintuicyjne efekty Mechaniki Kwantowej (losowość, superpozycja stanów, ewolucja unitarna). Wybrane elementy takich algorytmów i ich implementacji mogą być projektowane w sposób automatyczny za pomocą algorytmów ewolucyjnych, co zostało zaprezentowane w wielu pracach teoretycznych ([Nowotniak, 2008](#); [Giraldi et al., 2004](#)).

2) KWANTOWO INSPIROWANE ALGORYTMY EWOLUCYJNE – to nowa klasa metod ewolucyjnych, w których na wybranych etapach realizacji algorytmu (reprezentacja rozwiązań, ocena osobników, operatory genetyczne) występują dodatkowe elementy losowości, inspirowane Informatyką Kwantową ([Zhang, 2010a](#)). W tym podejściu nie zakłada się dostępności maszyn obliczeniowych, działających na poziomie kwantowym, jednak nie jest wykluczona potencjalna możliwość implementacji tych algorytmów w strukturze komputera kwantowego w przyszłości ([Udrescu et al., 2006](#); [Malossini et al., 2004](#)).

3) PRAWDZIWE KWANTOWE ALGORYTMY EWOLUCYJNE – w literaturze określane jako tzw. „true” *quantum evolutionary algorithms* ([Sofge, 2006, 2008](#); [Shannon, 2006](#)), wymagają rzeczywistej maszyny obliczeniowej, która w przyszłości mogłaby działać na poziomie kwantowym (komputer kwantowy). To podejście opiera się na bezpośrednim wykorzystaniu w pełni unikalnych zjawisk typowych dla Mechaniki Kwantowej (kwantowe splątanie, interferencja stanów, paralelizm) w algorytmie ewolucyjnym. Działanie takich algorytmów można rozważać w sposób teoretyczny oraz symulować na klasycznym komputerze, jednak taki proces charakteryzuje się wykładniczą złożonością obliczeniową.

Rozwój badań nad **Kwantowo Inspirowanymi Algorytmami Ewolucyjnymi** (ang. *Quantum Inspired Evolutionary Algorithms* – QIEA), których do-

tyczy niniejsza dysertacja, rozpoczął się na przełomie XX i XXI wieku (Narayanan, Moore, 1996; Han, Kim, 2000) i stanowią one jeden z przykładów bardzo skutecznego wykorzystania elementów inspirowanych teorią kwantową dla ulepszenia metod Sztucznej Inteligencji. Potencjał aplikacyjny tych metod jest bardzo duży, ponieważ w dziedzinie nauk technicznych istnieje szeroka klasa problemów, które mogą zostać sprowadzone do zadań o charakterze optymalizacji albo przeszukiwania, lub w których można bezpośrednio wyodrębnić aspekty o takim charakterze.

Podjęte w niniejszej rozprawie zagadnienie badawcze odpowiada bezpośrednio bieżącym problemom identyfikowanym i regularnie podkreślanym w literaturze tematu i aktualnych publikacjach przeglądowych (Zhang, 2010a; Manju, Nigam, 2012; Nowotniak, 2010):

1. W bardzo nielicznych pracach podejmowane były dotychczas próby zbadania własności algorytmów QIEA na drodze analiz teoretycznych.
2. Brakuje usystematyzowania możliwości wprowadzania czynników kwantowych do algorytmu ewolucyjnego – w blisko 300 znanych autorowi pracach nt. algorytmów QIEA elementy kwantowe wprowadzane były zazwyczaj w sposób arbitralny.
3. Nie rozważano dotąd możliwości oraz skutków zwiększania „elementu kwantowego”² w algorytmie oraz wpływu takiego działania na efektywność algorytmu w trudnych zadaniach optymalizacji numerycznej i kombinatorycznej.

²W Rozdziale 5 wprowadzone zostaje w sposób formalny pojęcie współczynnika kwantowości $\lambda \in [0, 1]$, charakteryzującego dany algorytm

CEL, ZAKRES I TEZY ROZPRAWY

Celem niniejszej rozprawy jest analiza własności nowej klasy algorytmów ewolucyjnych – kwantowo inspirowanych algorytmów ewolucyjnych (QIEA), do których „nowy wymiar” wprowadzony jest za pomocą dodatkowych elementów, inspirowanych teorią Informatyki Kwantowej. Takie podejście przybliża te algorytmy do niepewnego z natury charakteru wielu problemów świata rzeczywistego, a liczne przykłady aplikacyjne z minionej dekady pokazały, że umożliwia to poprawę efektywności algorytmów w różnorodnych zadaniach optymalizacji numerycznej i kombinatorycznej. W rozprawie zwrócono szczególną uwagę na możliwość analizy własności kwantowo inspirowanych algorytmów ewolucyjnych poprzez wykorzystanie metod teoretycznych, co wymagało adaptacji wybranych istniejących metod do nowego obszaru algorytmów kwantowo inspirowanych. Rozważono także możliwość wprowadzania elementów inspirowanych Informatyką Kwantową na różnych etapach realizacji algorytmu ewolucyjnego, oraz zaproponowano zwiększanie wpływu elementów kwantowych na działanie algorytmu.

TEZY ROZPRAWY

Na podstawie przeglądu literatury tematu, obejmującego zarówno dostępne autorowi publikacje naukowe z obszaru algorytmów QIEA, jak również ogólną literaturę naukową z zakresu heurystycznych metod optymalizacji i przeszukiwania, w tym techniki metaoptymalizacji i zrównoleglenia algorytmów ewolucyjnych, sformułowano następujące tezy rozprawy:

TEZA I: Wykorzystanie *II rzędu kwantowości*³ w kwantowo inspirowanym algorytmie genetycznym jest metodą poprawy efektywności algorytmu w szczególności w zadaniach zwodniczych.

³Definicja nowego pojęcia *rzędu kwantowości algorytmu* (rozmiar największego rejestru kwantowego w algorytmie) została wprowadzona w dalszej części rozprawy (**Rozdział 5**).

TEZA II: Heurystyczne algorytmy metaoptymalizacji oraz implementacja algorytmów w środowisku obliczeń równoległych stanowią efektywną metodę doboru wartości parametrów kwantowo inspirowanych algorytmów ewolucyjnych.

W celu dowiedzenia postawionych tez, zrealizowano następujące zadania cząstkowe:

1. Usystematyzowano możliwości wprowadzania czynników kwantowych do algorytmu ewolucyjnego (Nowotniak, 2010) poprzez wyodrębnienie tych etapów realizacji algorytmu, na których możliwe jest wykorzystanie dodatkowych elementów losowości.
2. Przeanalizowano w sposób teoretyczny wybrane, podstawowe własności algorytmów, w tym zbieżność w szerokiej klasie kwantowo inspirowanych algorytmów ewolucyjnych (Nowotniak, Kucharski, 2012a) oraz dopasowanie schematów w kwantowo inspirowanych algorytmach genetycznych (Nowotniak, Kucharski, 2010a). W tym celu wybrane znane metody analizy zostały zaadaptowane do obszaru kwantowo inspirowanych algorytmów ewolucyjnych.
3. Sformułowano pojęcia rzędu i współczynników kwantowości algorytmu, pozwalające zdefiniować nową klasę kwantowo inspirowanych algorytmów genetycznych wyższych rzędów (Nowotniak, Kucharski, 2014). Na tej podstawie opracowano autorski kwantowo inspirowany algorytm genetyczny rzędu II.
4. Przedstawiono autorski sposób efektywnego zrównoleglenia algorytmów oraz procedur badawczych (Nowotniak, Kucharski, 2011b).
5. Porównano efektywność prostego algorytmu genetycznego (SGA), pierwotnego kwantowo inspirowanego algorytmu genetycznego (QIGA1), dostrojonego w procesie metaoptymalizacji kwantowo inspirowanego algorytmu genetycznego (QIGA1*) oraz autorskiego kwantowo inspirowanego algorytmu genetycznego (QIGA2), wykorzystując zestaw zwodniczych zadań optymalizacji kombinatorycznej (Nowotniak, Kucharski, 2010b), (Nowotniak, Kucharski, 2011a).

-
6. Zaprezentowano dobór właściwych wartości parametrów algorytmu iQIEA, dzięki wykorzystaniu efektywnej autorskiej implementacji w środowisku obliczeń równoległych oraz heurystycznych metody metaoptymalizacji. Badania wykonano na zestawie 28 wielowymiarowych funkcji testowych optymalizacji numerycznej CEC2013. Wykazano w ten sposób przewagę dostrojonego algorytmu iQIEA nad innymi populacyjnymi heurystykami przeszukiwania (Nowotniak, Kucharski, 2010b).

UKŁAD TREŚCI ROZPRAWY

Treść rozprawy została podzielona na trzy zasadnicze części:

Część I – Wiadomości wstępne. Przedstawiono podstawowe wiadomości z zakresu metod ewolucyjnych Sztucznej Inteligencji (**Rozdział 1**), oraz niezbędne dla potrzeb rozprawy teoretyczne podstawy Informatyki Kwantowej (**Rozdział 2**): kubity, rejestry oraz bramki kwantowe. **Rozdział 3** prezentuje aktualny stan wiedzy w zakresie algorytmów QIEA.

Część II – Analiza teoretyczna. Zbadano podstawowe własności algorytmów poprzez adaptację wybranych metod analizy do obszaru algorytmów kwantowo inspirowanych oraz wprowadzono nowe pojęcia algorytmów kwantowo inspirowanych wyższych rzędów. W **Rozdziale 4.1** przeprowadzono analizę zbieżności ogólnego kwantowo inspirowanego algorytmu ewolucyjnego z wykorzystaniem twierdzenia Banacha o punkcie stałym odwzorowania zwężającego. W **Rozdziale 4.2** została zaprezentowana analiza propagacji bloków budujących w kwantowo inspirowanych algorytmach genetycznych. W **Rozdziale 5** wprowadzono podstawowe pojęcia algorytmów kwantowo inspirowanych wyższych rzędów, co pozwoliło na opracowanie kwantowo inspirowanego algorytmu genetycznego rzędu II.

Część III – Badania doświadczalne. Zweryfikowano efektywność autorskiego algorytmu rzędu II w zadaniach optymalizacji kombinato-

rycznej (**Rozdział 6**). Zbadano także wpływ wartości parametrów kwantowo inspirowanego algorytmu ewolucyjnego iQIEA, wykorzystując technikę metaoptymalizacji oraz implementację algorytmu w środowisku obliczeń równoległych (**Rozdział 7**).

Informacje uzupełniające, wykraczające tematycznie poza zasadniczą część niniejszej pracy, zostały przedstawione w załącznikach. W **Załączniku A** zostały przedstawione podstawowe postulaty Mechaniki Kwantowej, na których oparte są pojęcia oraz reguły Informatyki Kwantowej. W **Załączniku B** zostało podane twierdzenie Banacha o punkcie stałym odwzorowania zwężającego wraz z niezbędnymi definicjami. W **Załączniku C** zostały przedstawione wybrane szczegóły implementacyjne opracowanego przez autora oprogramowania, za pomocą którego zrealizowano badania doświadczalne. W **Załączniku D** przedstawiono wyniki szczegółowe przeprowadzonych eksperymentów numerycznych.

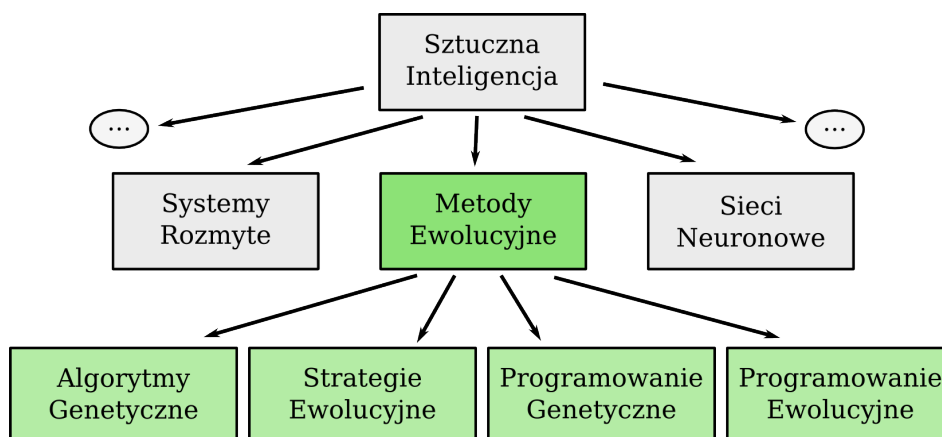
Część I

AKTUALNY STAN WIEDZY

Rozdział 1

ALGORYTMY EWOLUCYJNE

Algorytmy ewolucyjne są jedną z intensywnie rozwijanych metod Sztucznej Inteligencji i znajdują one szerokie zastosowanie w rozwiązywaniu zadań optymalizacji i przeszukiwania. Aktualny stan wiedzy w tym obszarze zaprezentowany jest w sposób wyczerpujący w licznych pracach m.in. (Goldberg, 1998; Michalewicz, 2003; Michalewicz, Fogel, 2006; Rutkowski, 2006; Arabas, 2004; Gwiazda, 2009).



RYS. 1.1: Wybrane klasy metod Sztucznej Inteligencji

Sztuczna Inteligencja (Russell *et al.*, 2010) zajmuje się badaniem, modelowaniem i próbą naśladowania wielu złożonych zjawisk obserwowanych w naturze, m.in. sposobu ludzkiego rozumowania, ewolucji biologicznej, kolektywnego zachowania zwierząt. Do podstawowych i najbardziej znanych metod Sztucznej Inteligencji (Rysunek 1.1) należą m.in. sztuczne sieci neuronowe (Tadeusiewicz, 1993), systemy rozmyte (Zadeh, 1965; Yager, Zadeh, 1992), systemy ekspertowe (Russell *et al.*, 2010), uczenie maszynowe (Samuel, 1967), algorytmy mrówkowe (Colorni *et al.*, 1991; Dorigo *et al.*, 1996) i wiele innych metod. Do najważniejszych metod Sztucznej Inteligencji należą również w szczególności algorytmy ewolucyjne, których nowa klasa jest przedmiotem niniejszej pracy.

1.1 OPTYMALIZACJA OBLICZENIOWA

Powszechność problemów o charakterze optymalizacji w dziedzinie nauk technicznych powoduje, że metody ewolucyjne Sztucznej Inteligencji znajdują praktyczne zastosowanie w rozwiązywaniu wielu problemów świata rzeczywistego. W szczególności okazują się one być wysoce skuteczne w rozwiązywaniu trudnych zadań **optymalizacji** i **przeszukiwania**, również w problemach NP-trudnych (Goldberg, 1989).

Definicja 1.1. Zakładając, że dany jest niepusty zbiór X (przestrzeń rozwiązań) oraz funkcja celu $f : X \mapsto \mathbb{R}$, zadaniem **optymalizacji**, nazywanej też programowaniem matematycznym, nazywamy poszukiwanie takiego elementu $x^* \in X$, że:

$$\forall_{x \in X} f(x^*) \leq f(x) \quad (\text{zadanie minimalizacji}) \quad (1.1)$$

lub

$$\forall_{x \in X} f(x^*) \geq f(x) \quad (\text{zadanie maksymalizacji}) \quad (1.2)$$

Każde zadanie maksymalizacji może być w sposób trywialny sprowadzone do zadania minimalizacji, ponieważ zawsze:

$$\min\{f(x)\} = \max\{-f(x)\} \quad (1.3)$$

Ze względu na charakter przeszukiwanej przestrzeni X zadania optymalizacji można podzielić na dwie kategorie:

- 1) OPTYMALIZACJA KOMBINATORYCZNA – poszukiwanie optymalnego rozwiązania x^* dokonywane jest w skończonej dyskretnej przestrzeni rozwiązań X . Przykładem optymalizacji kombinatorycznej jest zadanie szeregowania, polegające na znalezieniu optymalnej permutacji elementów danego zbioru.
- 2) OPTYMALIZACJA NUMERYCZNA – przeszukiwana jest N -wymiarowa przestrzeń zazwyczaj liczb rzeczywistych $X = \mathbb{R}^N$ lub jej podzbiór, zwany *obszarem dopuszczalnym*. Przykładami zadań optymalizacji numerycznej są: optymalizacja funkcji $\mathbb{R}^N \mapsto \mathbb{R}$, optymalizacja kształtu, znalezienie optymalnej trajektorii ruchu.

Wiele problemów w dziedzinie nauk technicznych może być sprowadzonych do zadań optymalizacji kombinatorycznej lub numerycznej, lub daje się w nich wyodrębnić zagadnienia o takim charakterze.

Do klasycznych metod optymalizacji należą m.in. dobrze znane z literatury: metoda najszybszego spadku (Fortuna *et al.*, 2005), metoda Hooke'a-Jeevesa (Hooke, Jeeves, 1961), metoda Newtona (Kusiak *et al.*, 2009, str. 106) metody quasi-Newtonowskie (Broyden, 1967), metoda gradientów sprzężonych (Fletcher, Reeves, 1964), metoda Neldera-Meada (Nelder, Mead, 1965), symulowane wyżarzanie (Kirkpatrick *et al.*, 1983).

1.2 RODZAJE ALGORYTMÓW EWOLUCYJNYCH

Wspólną cechą **algorytmów ewolucyjnych** (ang. *Evolutionary Algorithms* – EA) jest przetwarzanie populacji, będącej zbiorem zakodowanych postaci rozwiązań zadań optymalizacji lub przeszukiwania. W zależności od danego algorytmu reprezentacja rozwiązań może opierać się na różnego rodzaju strukturach np. ciągi binarne, liczby rzeczywiste, drzewa lub grafy. Do podstawowych rodzajów algorytmów ewolucyjnych należą **algorytmy genetyczne** (ang. *Genetic Algorithms* – GA) (Holland, 1975), strategie ewolucyjne (ang. *Evolutionary Strategies* – ES) (Rechenberg, 1973), programowanie genetyczne (ang. *Genetic Programming* – GP) (Koza, 1992), programowanie ewolucyjne (ang. *Evolutionary Programming* – EP) (Fogel *et al.*, 1966). Dużą popularnością w ostatnim czasie cieszą się również innego rodzaju heurystyczne metody populacyjne takie jak metoda roju cząstek (ang. *Particle Swarm Optimization* – PSO) (Kennedy, Eberhart, 1995), algorytm świetlika (ang. *Firefly Algorithm* – FA) (Yang, 2008), strategia ewolucyjna z adaptacyjną macierzą kowariancji (ang. *Covariance Matrix Adaptation Evolutionary Strategy* – CMAES) (Hansen, Ostermeier, 1996; Hansen, Kern, 2004). Jednym z pierwszych, podstawowych i najlepiej zbadanych algorytmów ewolucyjnych jest tzw. **prosty algorytm genetyczny** (ang. *Simple Genetic Algorithm* – SGA) (Holland, 1975; Goldberg, 1989; Michalewicz, 2003). Algorytm genetyczny jest szczególnym przypadkiem algorytmu ewolucyjnego i w jego klasycznej postaci populacja rozwiązań składa się ze zbioru ciągów binarnych, podlegających w kolejnych generacjach

operacjom krzyżowania i mutacji.

Od czasu opracowania podstawowych wymienionych rodzajów algorytmów ewolucyjnych prowadzono również badania nad ich zmodyfikowanymi wersjami, co miało na celu znalezienie kompromisu pomiędzy globalną *eksploracją* (ang. *exploration*) przestrzeni rozwiązań przez algorytm (np. przez zapewnienie wysokiej różnorodności genetycznej) a *eksploatacją* (ang. *exploitation*) wiedzy odkrytej we wcześniejszych generacjach (np. przez różne sposoby krzyżowania ze sobą dotychczas znalezionych najlepszych osobników). W badaniach dotyczących algorytmów ewolucyjnych rozważano w minionych dekadach ogromną różnorodność możliwych rozszerzeń i modyfikacji, mających na celu modelowanie różnych dodatkowych zjawisk występujących w przyrodzie, mogących potencjalnie poprawiać efektywność algorytmów np. poprzez zwiększenie różnorodności genetycznej populacji i przeciwdziałanie przedwczesnej zbieżności algorytmu. Niektóre z takich rozszerzeń w algorytmach ewolucyjnych to m.in. niszowanie (Streichert *et al.*, 2003; Hu, Goodman, 2004; Petrowski, 1996), deterioryzacja (Obuchowicz, 1997), poliploidalność (Hadad, Eick, 1997; Petko, Werner, 2007), modele wyspowe (Skolicki, De Jong, 2004; Whitley *et al.*, 1999; Whitley, 1994), koewolucja (Koza, 1991; Michalewicz, Nazhiyath, 1995; Pollack, Blair, 1998), wprowadzenie elementów memetycznych do algorytmu ewolucyjnego (Moscato *et al.*, 1989; Ong *et al.*, 2006).

1.3 WYBRANE ELEMENTY ALGORYTMÓW EWOLUCYJNYCH

Algorytmy ewolucyjne bazują na pojęciach i procesach obserwowanych w naturze: na selekcji naturalnej, dziedziczeniu i adaptacji gatunków do środowiska naturalnego. W algorytmach ewolucyjnych wykorzystywane są następujące pojęcia, wzorowane na pojęciach biologicznych: *gen*, *allel*, *chromosom*, *populacja*, *generacja*, *locus*, *fenotyp*, *genotyp*. Pojęcia te są powszechnie znane w obszarze obliczeń ewolucyjnych i przedstawione szczegółowo m.in. w pracach (Rutkowski, 2006; Michalewicz, 2003; Goldberg, 1998), dlatego w niniejszym rozdziale podano jedynie wybrane definicje ważne z punktu widzenia rozprawy. Do podstawowych pojęć algorytmów ewolucyjnych należą:

GENOTYP – zespół chromosomów danego osobnika. Najczęściej przyjmuje się, że genotyp składa się z pojedynczego chromosomu (genotyp monoploidalny).

FENOTYP – zestaw wartości (rozwiązanie), odpowiadających danemu genotypowi; fenotyp jest bezpośrednio elementem przestrzeni poszukiwań X . Przy tworzeniu fenotypu poprzez dekodowanie informacji zawartej w genotypie następuje tzw. ekspresja wartości genów.

LOCUS – pozycja wskazująca położenie danego genu w chromosomie.

ALLEL – zbiór dopuszczalnych wartości danego genu, w przypadku ciągów binarnych zbiór alleli to $A = \{0, 1\}$.

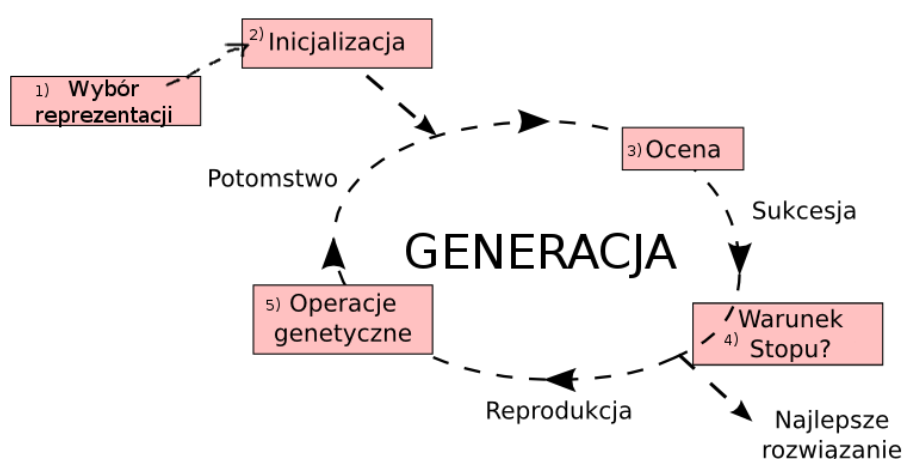
FUNKCJA DOPASOWANIA $f : X \mapsto \mathbb{R}$ – ilościowa miara, pozwalająca na ocenę jakości osobnika (rozwiązania), reprezentowanego przez element $x \in X$, za pomocą rzeczywistej wartości liczbowej $f(x)$. Pojedyncze obliczenie wartości funkcji dopasowania jest z założenia traktowane jako operacja dominująca w algorytmie ewolucyjnym. W zależności od charakteru zadania rozwiązywanego przez algorytm, funkcja dopasowania osobników może być oparta bezpośrednio lub pośrednio (np. przez dodatkowe przeskalowanie) na funkcji celu (w przypadku optymalizacji numerycznej), funkcji kosztu, funkcji błędu etc. Funkcja dopasowania jest też nazywana funkcją przystosowania lub funkcją oceny.

Do istotnych pojęć, związanych z algorytmami genetycznymi oraz możliwością ich zastosowania w rozwiązywaniu zadań przeszukiwania i optymalizacji, należą:

EPISTAZA – pozycyjna nieliniowość, współdziałanie genów zlokalizowanych na różnych *loci*; sytuacja taka występuje gdy na dopasowanie osobnika do środowiska składają się oddziałujące wzajemnie ze sobą podzbiory genów, zwłaszcza usytuowane na odległych od siebie pozycjach w chromosomie (Davidor, 1990).

ZWODNICZOŚĆ – własność zadań przeszukiwania oraz sposobów ich kodowania, polegająca na występowaniu izolowanych optimów w prze-

strzeni rozwiązań, gdzie punkty najlepsze są otoczone przez najgorsze (Goldberg, 1998, str.62), (Naudts *et al.*, 1999). Problemy zwodnicze sprawiają trudności algorytmom genetycznym, ponieważ optymalne rozwiązanie nie może być znalezione poprzez zestawianie ze sobą bloków budujących, występujących w rozwiązaniach dobrych lecz suboptymalnych.



RYS. 1.2: Ogólny schemat działania algorytmu ewolucyjnego z zaznaczonymi etapami, na których mogą być wprowadzone elementy kwantowe

Typowy schemat algorytmu ewolucyjnego, obejmujący kolejno etapy inicjalizacji, selekcji, rekombinacji, reprodukcji i oceny osobników, zaprezentowany jest na **Rysunku 1.2**. W algorytmie ewolucyjnym symulowany jest w sztucznym środowisku proces ewolucji biologicznej. W naturze osobniki rywalizują o przetrwanie, dopasowując się jak najlepiej w kolejnych generacjach do środowiska, w którym żyją. Na możliwość przetrwania i rozmnażania się osobników mają wpływ czynniki takie jak ich inteligencja, siła lub inne cechy. Ewolucja biologiczna może być zatem traktowana jako naturalny mechanizm optymalizacji. Należy zauważyć, że pomiędzy rzeczywistą ewolucją gatunków, obserwowaną w naturze, a symulowanym procesem algorytmicznym, imitującym ewolucję, istnieją jednak zasadnicze różnice. W naturalnej ewolucji funkcja dopasowania nie jest zdefiniowana bezpośrednio – nie ma z góry wyznaczonego celu, na który proces ewolucji

byłby ukierunkowany. W ewolucji biologicznej nie jest potrzebna żadna funkcja oceny, lecz jedynie relacja liniowego porządku, pozwalająca porównywać osobniki pomiędzy sobą. W algorytmie ewolucyjnym definiuje się natomiast zazwyczaj funkcję dopasowania osobników $f : X \mapsto \mathbb{R}$, której sposób zdefiniowania wpływa na działanie algorytmu.

Należy również podkreślić, iż algorytmów ewolucyjnych nie należy traktować jedynie jako metod optymalizacji, co było podkreślane m.in. już w pracy De Jonga, *Genetic algorithms are NOT function optimizers* (De Jong, 1993). Algorytmy ewolucyjne znajdują zastosowanie w szerszym zakresie problemów i należy je traktować jako ogólne metaheurystyki przeszukiwania. W literaturze rozważa się trzy ogólne obszary zastosowań algorytmów ewolucyjnych: (1) zadania optymalizacji (2) zadania przeszukiwania oraz (3) systemy uczące się. Te trzy obszary częściowo pokrywają się ze sobą (*Genetic Algorithms in Search, Optimization and Machine Learning* (Goldberg, 1989)). Wykorzystanie algorytmów ewolucyjnych w optymalizacji jest zatem szczególnym przypadkiem zastosowania tych algorytmów.

1.4 METODY ANALIZ TEORETYCZNYCH

Jednym z najważniejszych i wciąż aktualnych problemów w obszarze algorytmów genetycznych jest wyjaśnienie za pomocą metod matematycznych skuteczności mechanizmu wykorzystywanego w algorytmie, modelującym ewolucję biologiczną. Były w tym celu podejmowane próby stosowania różnych metod teoretycznej analizy (Eiben, Rudolph, 1999; Reeves, Rowe, 2003; Schmitt, 2004; Weise, 2008; Wolpert, Macready, 1997; Eremeev, 2000).

Pierwszą próbą wyjaśnienia efektywności algorytmów genetycznych była **teoria Hollanda** (Holland, 1975), jednak wielu badaczy próbowało stosować również inne metody. Była to m.in. analiza zbieżności z wykorzystaniem twierdzenia Banacha (Michalewicz, 2003), analiza algorytmów jako układów dynamicznych (Socółka *et al.*, 2005; Kotowski, 2008; Neubauer, 2008), zastosowanie teorii łańcuchów Markova (Paszynska, 2005; Rudolph, 1994), wykorzystanie transformacji Walsh'a (Vose, Wright, 1998a), wykorzystanie metody funkcji dolnej (Socółka, Kosiński, 2007). Do najistotniejszych aktualnych prac, odnoszących się do teorii schematów, należą (Holland,

1975; Vose, Wright, 1998a,b; Vose, 1999; Stephens, Waelbroeck, 1999; Ming *et al.*, 2004; Poli, 1999, 2001, 2000; Bäck, 1996; Yang, Li, 2003; Haijun *et al.*, 2003). Jedną z prac, prezentujących możliwość analizy propagacji schematów w algorytmach genetycznych z uwzględnieniem wyników badań doświadczalnych jest (Kusztelak, Rudnicki, Wiak, 2004).

Do podstawowych pojęć, pozwalających na analizę teoretyczną własności algorytmów genetycznych za pomocą schematów, należą:

Definicja 1.2. Schematem $H \in \{0, 1, *\}^N$ nazywamy napis nad alfabetem ternarnym $\{0, 1, *\}$ o długości $N \in \mathbb{N}^+$.

Definicja 1.3. Rzędem schematu $o(H) \in \mathbb{N}$ nazywamy liczbę ustalonych symboli (zer lub jedynek w przypadku ciągu binarnego) w schemacie H .

Definicja 1.4. Rozpiętością schematu $d(H) \in \mathbb{N}$ nazywamy odległość między dwiema skrajnymi pozycjami ustalonymi w schemacie H .

Zgodnie z powyższymi definicjami przykładowy schemat $H = **01**0*$ ma długość $N = 8$ i charakteryzowany jest rzędem $o(H) = 3$ oraz rozpiętością $d(H) = 4$. Dany ciąg binarny pasuje do schematu H , jeśli kolejnym elementom ciągu odpowiadają elementy schematu, za wyjątkiem symboli nieoznaczonych $*$ w schemacie.

Podstawowym twierdzeniem, określającym własność prostego algorytmu genetycznego SGA, jest tzw. **twierdzenie Hollanda o schematach** (Holland, 1975) (Goldberg, 1998, str. 50) oraz związana z nim **hipoteza bloków budujących** (Rutkowski, 2006, str. 249):

Twierdzenie 1.1 (Hollanda o schematach). Krótkie, niskiego rzędu i oceniające powyżej średniej schematy uzyskują wykładniczo rosnącą liczbę swoich reprezentantów w kolejnych generacjach algorytmu genetycznego.

$$E[m(H, t + 1)] \geq m(H, t) \frac{f(H)}{\bar{f}} \left[1 - P_c \frac{d(H)}{N - 1} - o(H)P_m \right] \quad (1.4)$$

gdzie:

- $P_c \in [0, 1]$ – prawdopodobieństwo krzyżowania
- $P_m \in [0, 1]$ – prawdopodobieństwo mutacji
- $f(H) \in \mathbb{R}$ – wartość dopasowania schematu H
- $\bar{f} \in \mathbb{R}$ – wartość dopasowania całej populacji
- $N \in \mathbb{N}^+$ – długość chromosomów

Twierdzenie Hollanda o schematach odegrało dużą rolę w rozwoju metod teoretycznej analizy działania algorytmów genetycznych, ale ze względu na swoje ograniczenia obecnie traktowane jest jako historyczna, pierwsza próba wyjaśnienia sposobu działania tych algorytmów. Teoria Hollanda ma poważne ograniczenia i trudne do spełnienia założenia (nieskończony rozmiar populacji) i nie sprawdza się ona dla zadań, w których optymalne rozwiązanie nie jest zbudowane ze schematów, spełniających kryteria „cegiełek Hollanda” (problemy zwodnicze, tzw. GA-trudne).

1.5 TECHNIKI METAOPTYMALIZACJI

Efektywność zrandomizowanych algorytmów heurystycznych, do których należą algorytmy ewolucyjne, jest w znacznym stopniu uzależniona od licznych rzeczywistych i dyskretnych parametrów tychże algorytmów. W zależności od konkretnego algorytmu, do takich parametrów należą np. prawdopodobieństwo mutacji i rekombinacji, prawdopodobieństwa stosowania innych operatorów genetycznych, współczynniki uczenia, kryterium stopu, współczynniki migracji, nacisku selekcyjnego etc.

Metody analiz teoretycznych pozwoliły na określenie właściwych wartości parametrów algorytmów ewolucyjnych jedynie w nielicznych i bardzo uproszczonych przypadkach ([Chaiyaratana, Zalzala, 1997](#); [Schmitt, 2001](#); [Yin, Khoo, 2011](#)). Z tego powodu często niezbędne są wielokrotnie powtarzane, czasochłonne badania doświadczalne, polegające na uruchamianiu algorytmu z różnymi wartościami parametrów. Z uwagi na duże trudności w prowadzeniu skutecznych analiz teoretycznych, uniemożliwiające zazwyczaj bezpośrednie ustalenie optymalnych wartości parametrów algorytmów, od pewnego czasu zaczęto rozważać możliwość doboru parametrów na drodze tzw. **metaoptymalizacji**, polegającej na potraktowaniu zadania doboru parametrów jako na zadaniu optymalizacji samego w sobie ([Goldberg, 1998](#), str. 113).

Metaoptymalizacja jest skuteczną metodą rozwiązywania problemu doboru wartości parametrów algorytmu i jej ogólna idea została zilustrowana na [Rysunku 1.3](#). W procesie metaoptymalizacji efektywność algorytmu jest oceniana poprzez wartość jego **metaprzystosowania** $\tilde{f}$ dla danego ze-

stawu parametrów. Takie metaprzystosowanie opisuje za pomocą miary ilościowej efektywność algorytmu o danych parametrach. Metadopasowanie $\tilde{f}$ może być oparte np. na średniej wartości najlepszego rozwiązania wybranego zadania testowego $x \in X$, znajdowanego przez wielokrotne uruchamianie badanego algorytmu.

Definicja 1.5. Dla danego algorytmu o $\tilde{N} \in \mathbb{N}^+$ parametrach oraz funkcji metaprzystosowania $\tilde{f}$, **zadanie metaoptymalizacji** polega na znalezieniu wektora parametrów algorytmu p^* , zapewniającego optymalną efektywność algorytmu:

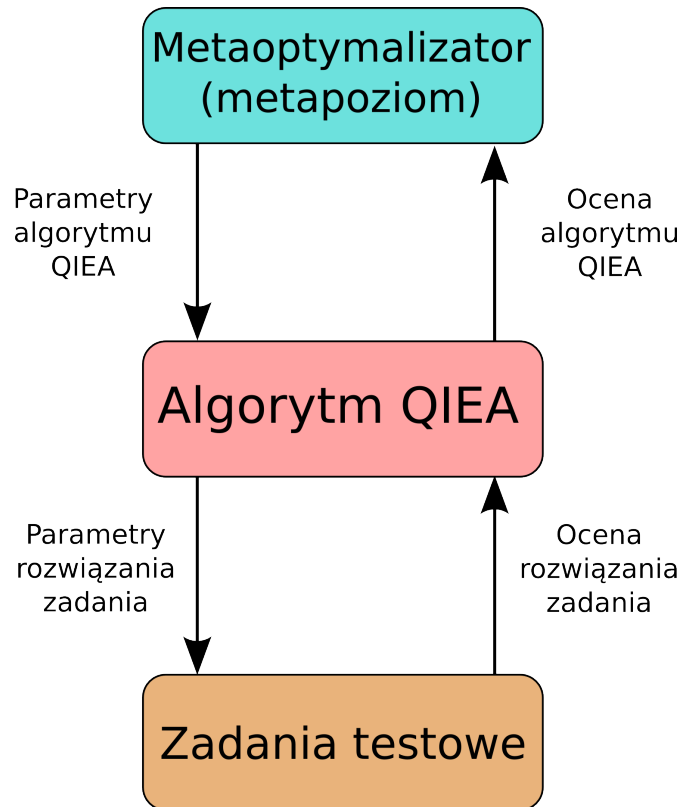
$$\forall_p \tilde{f}(p^*) \leq \tilde{f}(p) \quad (\text{minimalizacja}) \quad (1.5)$$

lub

$$\forall_p \tilde{f}(p^*) \geq \tilde{f}(p) \quad (\text{maksymalizacja}) \quad (1.6)$$

Po raz pierwszy takie podejście zostało rozważone teoretycznie już w pracy (Weinberg, 1970), gdzie zaproponowano tzw. dwuwarstwowy algorytm genetyczny z algorytmem adaptacyjnym, rozwiązującym zadany problem, oraz algorytmem nieadaptacyjnym na metapoziomie, którego zadaniem był dobór właściwych wartości algorytmu adaptacyjnego. Jednak ze względu na zbyt dużą jak na ówczesne czasy niezbędną moc obliczeniową nie zostały w tej pracy przedstawione wyniki żadnej symulacji. Pierwsze proste implementacje oraz wyniki bardzo ograniczonych eksperymentów z algorytmem ewolucyjnym, działającym na metapoziomie, zaprezentowane zostały po raz pierwszy w (Mercer, Sampson, 1978; Grefenstette, 1986).

Wraz ze wzrostem liczby parametrów algorytmu $\tilde{N}$ rośnie w tempie wykładniczym złożoność znalezienia optymalnych wartości (ang. *Curse of dimensionality*) (Pedersen, 2010). Ponadto, ze względu na stochastyczną naturę algorytmów ewolucyjnych, wartość metadopasowania powinna być oparta na kilkudziesięciokrotnym uruchomieniu algorytmu (Luke, 2009). Obliczanie wartości metaprzystosowania (w celu poznania ukształtowania tzw. *krajobrazu metaprzystosowania*) jest zatem czasochłonnym i kosztowym obliczeniowo zadaniem. Zbadanie efektywności algorytmu dla wszystkich możliwych wartości parametrów (co odpowiadałoby wygenerowaniu



Rys. 1.3: Ogólny schemat metaoptymalizacji

pełnej mapy krajobrazu metaprzystosowania) zazwyczaj nie jest możliwe lub jest bardzo czasochłonne. Ze względu na elementy losowości w algorytmach ewolucyjnych, aby ocenić algorytm o określonych wartościach parametrów, algorytm powinien być uruchomiony kilkudziesięciokrotnie, np. 50-krotnie (Luke, 2009). Zatem dla populacji składającej się ze 100 osobników, które ewoluują przez 100 generacji, obliczenie pojedynczej wartości metaprzystosowania $\tilde{f}$ wymaga $50 \cdot 100 \cdot 100 = 500000$ wywołań funkcji oceny w algorytmie ewolucyjnym. Pokazuje to, że ocena metado-
pasowania wiąże się z kosztem obliczeniowym, będącym kilka rzędów wielkości większym niż ocena rozwiązań przestrzeni X . Z tego względu do wykonania procesu metaoptymalizacji niezbędne są wysoce efektywne implementacje algorytmów, które można zrealizować wykorzystując np. środowiska obliczeń równoległych (klastry obliczeniowe lub technologie GPGPU).

Jedną z nowoczesnych technik metaoptymalizacji jest np. metoda Pedersena ([Pedersen, 2010](#)). W porównaniu z pełnym przeglądem przestrzeni metaprzyrostosowania metaoptymalizacja wymaga jedynie niewielkiej części czasu obliczeniowego dla znalezienia parametrów algorytmu optymalizacji, zapewniających jego efektywność. Propozycja Pedersena opiera się na wykorzystaniu algorytmu Local Unimodal Sampling (LUS) ([Pedersen, Chipperfield, 2008](#); [Pedersen, 2010](#)) do zrandomizowania przeszukiwania krajobrazu metaprzyrostosowania. Wiele innych nowoczesnych podejść do zadania metaoptymalizacji, wykorzystujących techniki uczenia maszynowego, zaprezentowano m.in. w ([Birattari, 2009](#)).

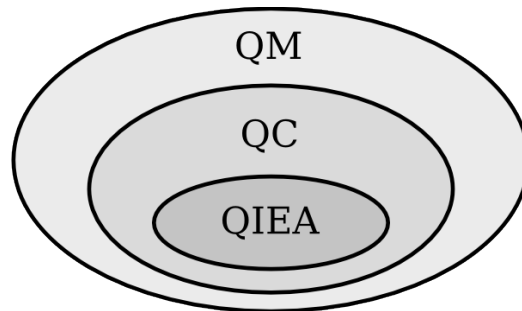
Rozdział 2

TEORETYCZNE PODSTAWY INFORMATYKI KWANTOWEJ

Informatyka Kwantowa to dziedzina wiedzy, zajmująca się wykorzystaniem potencjalnych możliwości obliczeniowych miniaturowych układów, podlegających prawom Mechaniki Kwantowej (Nielsen, Chuang, 2000; Hirvensalo, 2004). W niniejszym rozdziale zostały przedstawione podstawowe pojęcia i prawa Informatyki Kwantowej, na których bazują kwantowo inspirowane algorytmy genetyczne i ewolucyjne. Wynikają one z tzw. postulatów Mechaniki Kwantowej, przedstawionych w **Załączniku A** (str. 163).

W klasycznej informatyce wartości zmiennych (zapisywane w pamięci operacyjnej komputera w postaci bajtów, słów maszynowych etc.) przyjmują konkretne i ustalone wartości, podobnie jak w mechanice klasycznej obiekty fizyczne mogą posiadać ustalone masy, położenia lub prędkości. W Mechanice Kwantowej natomiast wielkości fizyczne określone są w sposób stochastyczny, np. z każdym punktem przestrzeni może być związana liczba, zwana amplitudą prawdopodobieństwa, określająca prawdopodobieństwo zaobserwowania danego obiektu w danym punkcie przestrzeni. Rozkłady te ulegają zmianom w czasie zgodnie z tzw. ewolucją unitarną. Jest to podstawowa różnica pomiędzy mechaniką klasyczną i kwantową, oraz informatyką klasyczną i kwantową (*obliczenia wykonywane na konkretnych wartościach* i *obliczenia polegające na przetwarzaniu rozkładów prawdopodobieństw*). Pomiedzy zwykłym rachunkiem prawdopodobieństwa a Mechaniką Kwantową istnieje jednak kilka istotnych różnic. W rachunku prawdopodobieństwa wartość prawdopodobieństwa jest liczbą rzeczywistą z przedziału $[0, 1]$, zaś w mechanice kwantowej z każdym elementem przestrzeni związana jest amplituda prawdopodobieństwa, która może być liczbą zespoloną. Dzięki temu w algorytmach kwantowych różne rozwiązania problemów obliczeniowych mogą interferować ze sobą, np. wzajemnie

się znosząc, co jest wykorzystywane w niektórych opracowanych dotychczas algorytmach kwantowych (Grover, 1996).



RYS. 2.1: Kwantowo inspirowane algorytmy ewolucyjne (QIEA) opierają się na ograniczonym podzbiorze pojęć Informatyki Kwantowej (QC) oraz ograniczonym fragmencie teorii Mechaniki Kwantowej (QM).

Na **Rysunku 2.1** zostały zilustrowane związki pomiędzy poszczególnymi obszarami wiedzy. Należy mieć na uwadze, że Mechanika Kwantowa to teoria fizyczna i definiuje ona wiele pojęć w sposób bardzo ogólny, często zbyt ogólny z punktu widzenia informatyki oraz z punktu widzenia obliczeniowego aspektu działania komputera kwantowego. Dla potrzeb niniejszej rozprawy niezbędne jest wprowadzenie jedynie elementarnych pojęć wraz z ograniczonym podzbiorem teorii Informatyki Kwantowej. W mechanice kwantowej stan układu określany jest w ∞ -wymiarowej, zespolonej przestrzeni Hilberta, natomiast w komputerze kwantowym rejestry kwantowe zawsze mają określony rozmiar ($N \in \mathbb{N}^+$), więc ich przestrzeń stanów ma określony skończony wymiar. W mechanice kwantowej ewolucja stanu układu określana jest ciągłym równaniem Schrödingera (**Postulat A.1**), natomiast w Informatyce Kwantowej następują po sobie kolejne dyskretne etapy obliczeń, których liczba jest skończona, i które są definiowane przez macierze unitarne. Dodatkowo w rozważanej klasie algorytmów QIEA pojęcia Informatyki Kwantowej są jeszcze bardziej zawężone i ograniczone: np. nie występują częściowe pomiary stanów rejestrów kwantowych, ani składniki urojone w amplitudach prawdopodobieństw.

Dzięki temu teoria przedstawiona w niniejszym rozdziale mogła zostać ograniczona jedynie do wybranego zestawu pojęć wykorzystywanych w

zasadniczej części rozprawy. Pełen zakres teorii Informatyki Kwantowej można znaleźć w monograficznej pracy (Nielsen, Chuang, 2000).

2.1 KUBITY, ZBIORY KUBITÓW I REJESTRY KWANTOWE

Kubity

Podstawową jednostką danych w informatyce jest bit, mogący przyjmować jedną spośród dwóch wartości logicznych: 0 albo 1. W Informatyce Kwantowej podstawową jednostką informacji są **kubity** (ang. *qubits*) i **rejestry kwantowe** (ang. *quantum registers*). Kubit (podobnie jak bit) może przyjmować wartości 0 albo 1, oznaczane w tzw. notacji Diraca odpowiednio $|0\rangle$ (czyt. *ket 0*) i $|1\rangle$ (czyt. *ket 1*), i są to wielkości wektorowe a nie skalarne. Kubit może przyjmować także wartość będącą dowolną superpozycją tych dwóch stanów bazowych $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ i $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Definicja 2.1. Kubit (ang. *qubit*) to wektor znormalizowany w dwuwymiarowej, zespolonej przestrzeni Hilberta. Korzystając z notacji Diraca, stan kubitów można także zapisać jako wektor kolumnowy:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = [\alpha \ \beta]^T \quad (2.1)$$

gdzie: $\alpha, \beta \in \mathbb{C}$ oraz $|\alpha|^2 + |\beta|^2 = 1$. Liczby α i β nazywa się **amplitudami prawdopodobieństwa**. Wektory $|0\rangle$ i $|1\rangle$ tworzą **bazę standardową** (lub obliczeniową) dwuwymiarowej, zespolonej przestrzeni stanów kubitów.

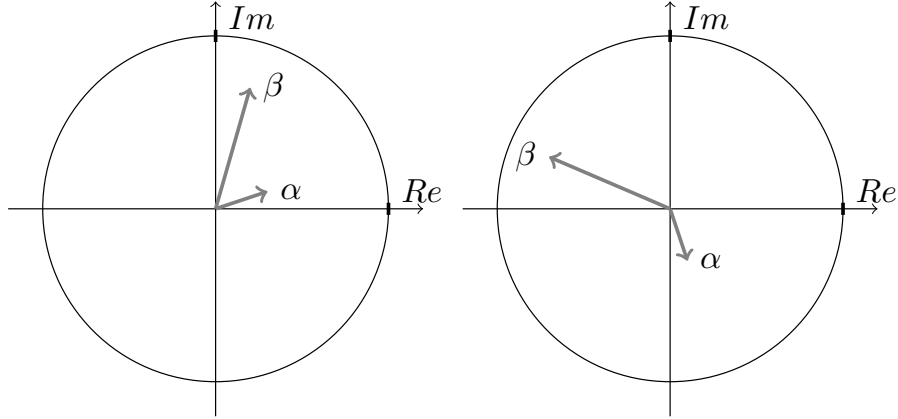
Dwie możliwe geometryczne reprezentacje kubitów przedstawione zostały na **Rysunku 2.2**. Ponieważ w ogólnym przypadku amplitudy prawdopodobieństwa reprezentowane są przez pojedyncze wektory na płaszczyźnie zespolonej $\mathbb{C}$, na **Rysunku 2.2a** stan kubitów reprezentuje para wektorów α i β . Dzięki możliwości ograniczenia amplitud w kubicie do ciała liczb rzeczywistych stan kubitów może być reprezentowany jako pojedynczy, znormalizowany wektor na płaszczyźnie (**Rysunek 2.2b**). W takim przypadku stan kubitów opisywany jest w dwuwymiarowej przestrzeni liniowej.

Własność 2.2. W wyniku **pomiaru** wartości kubitów¹ (tzw. **obserwacja stanu kubitów**), kubit przyjmuje wartość $|0\rangle$ (*ket 0*) z prawdopodobieństwem

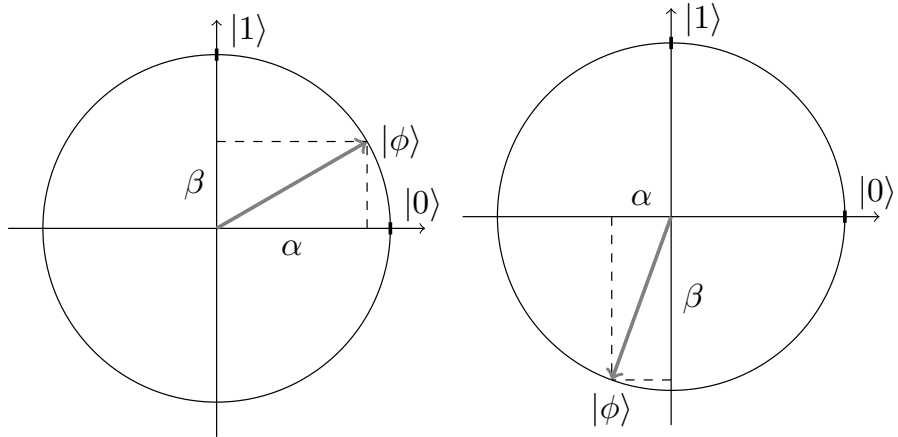
¹dotyczy to pomiaru wykonywanego w bazie standardowej $\{|0\rangle, |1\rangle\}$

$|\alpha|^2$ lub wartość $|1\rangle$ (ket 1) z prawdopodobieństwem $|\beta|^2$.

Własność 2.3. W Informatyce Kwantowej odczyt wartości kubitów nieodwracalnie niszczy jego stan. Stan kubitów zostaje ustalony na jeden z wektorów bazy pomiarowej $\{|0\rangle, |1\rangle\}$.



(A) Ilustracja stanów kubitów w ogólnym przypadku,
 $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, gdy $\alpha, \beta \in \mathbb{C}$



(B) Ilustracja stanów kubitów ograniczonego do ciała $\mathbb{R}$

RYS. 2.2: Kubit w ogólnym przypadku może być reprezentowany jako para wektorów na płaszczyźnie zespolonej (A). W kwantowo inspirowanych algorytmach genetycznych wykorzystywane są wyłącznie składniki rzeczywiste, więc stan można przedstawić jako pojedynczy znormalizowany wektor (B).

Pomimo że w „prawdziwym” komputerze kwantowym stan układu kwantowego ulegałby niszczeniu wraz z operacją pomiaru stanu układu, to zarówno w przypadku symulacji pracy komputera kwantowego (Gawron *et al.*, 2010; Nowotniak, 2008) jak i w kwantowo inspirowanych algorytmach ewolucyjnych stan układu kwantowego może być zachowany, a odpowiadający mu rozkład prawdopodobieństwa może podlegać wielokrotnemu próbkowaniu.

Zbiory niezależnych kubitów

Jeśli pomiędzy N kubitami nie zachodzą jakiejkolwiek wzajemne interakcje i pozostają one całkowicie odseparowane od siebie, ich stany mogą być reprezentowane jako N niezależnych jednostkowych wektorów na płaszczyźnie.

Zakładając, że przestrzeń stanów kubitów zostanie ograniczona do ciała liczb rzeczywistych $\mathbb{R}$ ($\alpha, \beta \in \mathbb{R}$), możliwe jest przyjęcie następującego, uproszczonego sposobu zapisu. Niech stan bazowy kubit $|\phi\rangle = |0\rangle$ będzie oznaczany symbolem $\ominus$, zaś stan $|\phi\rangle = |1\rangle$ symbolem $\oplus$. Wówczas dowolny stan kubitów można krótko zapisać w przybliżony sposób np. $|\phi\rangle = \frac{\sqrt{2}}{2}|0\rangle + \frac{\sqrt{2}}{2}|1\rangle = \otimes$. Podobnie w zwięzły i przybliżony sposób można przedstawić stan zbioru niezależnych kubitów np. $\ominus\oplus\ominus\oplus\oplus$ jest równoważne 5 kubitom w stanach: $|0\rangle, |1\rangle, |0\rangle, |1\rangle, |1\rangle$, natomiast zapis $\otimes\oplus\ominus\otimes$ odpowiada czterem kubitom, spośród których każdy znajduje się w pewnej superpozycji stanów bazowych $|0\rangle$ i $|1\rangle$. Ten uproszczony sposób zapisu został wykorzystany w kolejnych rozdziałach pracy.

Rejestry kwantowe

Wspólny układ wielu kubitów tworzy **rejestr kwantowy** (ang. *quantum register*), który — zgodnie z jednym z postulatów Mechaniki Kwantowej (Załącznik A, str. 163) — może być rozpatrywany jako układ izolowany złożony z wielu układów składowych (poszczególne kubity należące do rejestru).

Dla zdefiniowania pojęcia rejestru kwantowego potrzebne jest uprzednie wprowadzenie **iloczynu Kroneckera**, zwanego również **iloczynem**

tensorowym (Hirvensalo, 2004, str. 18).

Definicja 2.4. Dla dowolnych macierzy

$$A_{r \times s} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1s} \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rs} \end{bmatrix} \quad B_{t \times u} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1u} \\ b_{21} & b_{22} & \cdots & b_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ b_{t1} & b_{t2} & \cdots & b_{tu} \end{bmatrix}$$

definiuje się **Iloczyn Kroneckera** (ang. *Kronecker product*) jako macierz blokową

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1s}B \\ a_{21}B & a_{22}B & \cdots & a_{2s}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1}B & a_{r2}B & \cdots & a_{rs}B \end{bmatrix}$$

Przykład 2.5. Niech będą dane macierze A oraz B :

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Wówczas iloczyn Kroneckera tych macierzy wynosi:

$$A \otimes B = \begin{bmatrix} 1 \cdot 5 & 1 \cdot 6 & 2 \cdot 5 & 2 \cdot 6 \\ 1 \cdot 7 & 1 \cdot 8 & 2 \cdot 7 & 2 \cdot 8 \\ 3 \cdot 5 & 3 \cdot 6 & 4 \cdot 5 & 4 \cdot 6 \\ 3 \cdot 7 & 3 \cdot 8 & 4 \cdot 7 & 4 \cdot 8 \end{bmatrix} = \begin{bmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{bmatrix}$$

Własność 2.6. Iloczyn tensorowy stanów bazowych $|\psi\rangle$ i $|\phi\rangle$ bazy kanonicznej, zapisanych w postaci binarnej, jest stanem bazowym w postaci konkatencji $|\psi\phi\rangle$.

Przykład 2.7. $|1\rangle \otimes |101\rangle = |1\rangle|101\rangle = |1101\rangle$, ponieważ (z definicji iloczynu tensorowego 2.4):

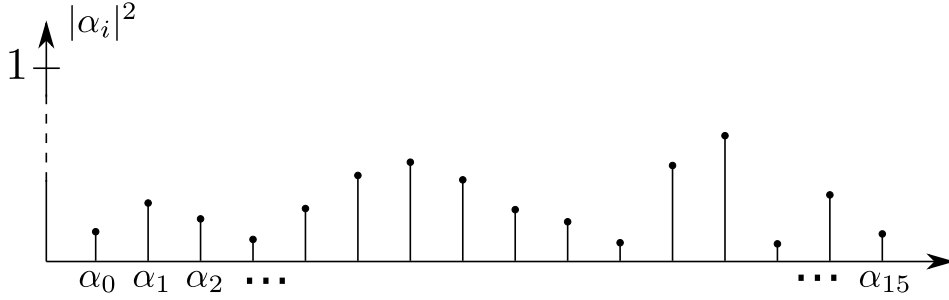
$$|1\rangle \otimes |101\rangle = |1\rangle|101\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot 0 \\ 0 \cdot 0 \\ 0 \cdot 0 \\ 0 \cdot 0 \\ 0 \cdot 0 \\ 0 \cdot 1 \\ 0 \cdot 0 \\ 1 \cdot 0 \\ 1 \cdot 0 \\ 1 \cdot 0 \\ 1 \cdot 0 \\ 1 \cdot 0 \\ 1 \cdot 1 \\ 1 \cdot 0 \\ 1 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = |1101\rangle$$

Definicja 2.8. Przestrzenią stanów **rejestru kwantowego** jest przestrzeń, będąca iloczynem tensorowym przestrzeni stanów poszczególnych kubitów należących do rejestru (**Postulat A.2**). Stan N-kubitowego rejestru kwantowego jest zatem opisywany jako wektor znormalizowany w 2^N -wymiarowej przestrzeni Hilberta. Pozwala to zapisać w rejestrze superpozycję 2^N różnych wartości.

W przypadku rejestru kwantowego, składającego się jedynie z dwóch kubitów $|q_0\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ i $|q_1\rangle = \alpha_2|0\rangle + \alpha_3|1\rangle$, stan rejestru jest opisywany jako iloczyn tensorowy (gdzie $\alpha_0, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{C}$):

$$\begin{aligned} |q_0\rangle \otimes |q_1\rangle &= (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\alpha_2|0\rangle + \alpha_3|1\rangle) \\ &= \alpha_0\alpha_2|0\rangle \otimes |0\rangle + \alpha_0\alpha_3|0\rangle \otimes |1\rangle + \alpha_1\alpha_2|1\rangle \otimes |0\rangle + \alpha_1\alpha_3|1\rangle \otimes |1\rangle \\ &= \alpha_0\alpha_2|00\rangle + \alpha_0\alpha_3|01\rangle + \alpha_1\alpha_2|10\rangle + \alpha_1\alpha_3|11\rangle \end{aligned}$$

Pojedynczy kubit $|\phi\rangle \in \mathcal{H}_2$ reprezentowany jest jako para liczb zespolonych, układ dwukubitowy reprezentowany jest wektorem z przestrzeni $\mathcal{H}_4$. Stan trzykubitowego rejestru jest opisany przez wektor znormalizowany z przestrzeni $\mathcal{H}_8$ itd.



RYS. 2.3: Ilustracja stanu 4-kubitowego rejestru. Występuje tu $2^4 = 16$ amplitud prawdopodobieństwa.

Dowolny stan $|\phi\rangle$ 4-kubitowego rejestru kwantowego może być zawsze zapisany w bazie standardowej $\{|0000\rangle, |0001\rangle, \dots, |1111\rangle\}$ jako:

$$|\phi\rangle = \alpha_0|0000\rangle + \alpha_1|0001\rangle + \dots + \alpha_{15}|1111\rangle$$

Zostało to zilustrowane na **Rysunku 2.3**. Występuje tu $2^4 = 16$ amplitud prawdopodobieństw, spełniających warunek $\sum_{i=0}^{15} |\alpha_i|^2 = 1$.

Powyższy zapis w notacji Diraca jest równoważny wektorowi kolumnowemu:

$$|\phi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{15} \end{bmatrix} = \alpha_0 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \dots + \alpha_{15} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (2.2)$$

gdzie: $|\alpha_0|^2 + |\alpha_1|^2 + \dots + |\alpha_{15}|^2 = 1$.

W wyniku pomiaru stanu takiego trzykubitowego rejestru $|\phi\rangle$ jego stan zostaje ustalony na wektor bazowy $|000\rangle$ z prawdopodobieństwem $|\alpha_0|^2$, $|001\rangle$ z prawdopodobieństwem $|\alpha_1|^2$ itd.

Z własności iloczynu tensorowego (2.4) wynika bezpośrednio:

Własność 2.9. Wraz z liniowym wzrostem liczby kubitów w rejestrze kwantowym rośnie w tempie wykładniczym wymiar przestrzeni stanów rejestru.

Z powyższej własności wynika brak możliwości efektywnego symulowania pracy komputera kwantowego na klasycznej maszynie obliczeniowej. (Nowotniak, 2008; Gawron *et al.*, 2010).

2.2 KWANTOWE STANY SPLĄTANE

Definicja 2.10. Złożony stan kwantowy $|\phi\rangle$ nazywany jest **stanem splątany** (ang. *quantum entanglement state*), jeśli nie można przedstawić go w postaci iloczynu tensorowego jego części składowych. W przypadku rejestru kwantowego $|\phi\rangle$, składającego się z dwóch kubitów:

$$\nexists_{a,b,c,d \in \mathbb{C}} |\phi\rangle = (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) \quad (2.3)$$

gdzie: $|a|^2 + |b|^2 = 1, |c|^2 + |d|^2 = 1$.

Taki złożony stan kwantowy, składający się z dwóch lub większej liczby układów, musi być traktowany jako całość – jako tzw. pojedyncza jednostka holistyczna (Penrose, 2006, str. 552). Rozkład prawdopodobieństwa określony jest w takim przypadku dla całego układu i ta własność jest zachowana, niezależnie od odległości, jaka dzieli elementy składowe, podlegające splątaniu (Hirvensalo, 2004; Penrose, 2006). Ta unikalna własność stała się podstawą do opracowania niektórych algorytmów kwantowych, np. protokołu kwantowej teleportacji (Vaidman, 1994; Boschi *et al.*, 1998; Riebe *et al.*, 2004), algorytmów kryptografii kwantowej (Brassard *et al.*, 1993).

Przykładami dwukubitowych stanów splątanych są tzw. **pary EPR** (Einstein, Podolsky, Rosen, 1935), tworzące bazę Bella (Bell *et al.*, 1964):

$$\begin{aligned} |\Phi^+\rangle &= \frac{\sqrt{2}}{2}|00\rangle + \frac{\sqrt{2}}{2}|11\rangle \\ |\Phi^-\rangle &= \frac{\sqrt{2}}{2}|00\rangle - \frac{\sqrt{2}}{2}|11\rangle \\ |\Psi^+\rangle &= \frac{\sqrt{2}}{2}|01\rangle + \frac{\sqrt{2}}{2}|10\rangle \\ |\Psi^-\rangle &= \frac{\sqrt{2}}{2}|01\rangle - \frac{\sqrt{2}}{2}|10\rangle \end{aligned} \quad (2.4)$$

Przykład 2.11. Stan $|\Phi^+\rangle = \frac{\sqrt{2}}{2}|00\rangle + \frac{\sqrt{2}}{2}|11\rangle$ jest stanem splątany, ponieważ nie można go przedstawić jako iloczynu tensorowego:

$$|\Phi^+\rangle \neq (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$$

Wymnażając prawą część nierówności:

$$\frac{\sqrt{2}}{2}|00\rangle + \frac{\sqrt{2}}{2}|11\rangle \neq (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$$

otrzymuje się:

$$\frac{\sqrt{2}}{2}|00\rangle + \frac{\sqrt{2}}{2}|11\rangle \neq ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

Z powyższego wynika układ równań:

$$\begin{cases} ac = \frac{\sqrt{2}}{2} \\ bd = \frac{\sqrt{2}}{2} \\ ad = 0 \\ bc = 0 \end{cases} \quad (2.5)$$

Z pierwszego i drugiego równania w układzie wynika, że wszystkie zmienne a, b, c, d muszą być niezerowe, zaś z trzeciego i czwartego równania wynika, że co najmniej dwie spośród zmiennych muszą być zerem. Układ ten jest zatem sprzeczny, co dowodzi, że $|\Phi^+\rangle$ jest stanem splątany. Analogiczny dowód można przeprowadzić dla pozostałych par EPR we wzorach (2.4).

2.3 PODSTAWOWE KWANTOWE BRAMKI LOGICZNE

Stany kubitów i rejestrów kwantowych przekształcane są przez kwantowe bramki logiczne, opisywane za pomocą macierzy unitarnych.

Definicja 2.12. N -kubitowa **bramka kwantowa** (ang. *quantum gate*) to dowolne odwzorowanie 2^N -wymiarowej przestrzeni Hilberta w 2^N -wymiarową przestrzeń Hilberta, opisywane za pomocą macierzy unitarnej i wymiarze $2^N \times 2^N$.

Podczas ewolucji czasowej układu kwantowego musi być zachowany warunek normalizacyjny wektora stanu (**Postulat A.1**), co odpowiada jednostkowej sumie prawdopodobieństw w statystycznej interpretacji Mechaniki Kwantowej. Jednocześnie obliczenia wykonywane przez bramkę kwantową muszą być obliczeniami odwracalnymi, co wynika z wymaganej odwracalności względem czasu ewolucji układu kwantowego (**Feynman**,

1996). Dowodzi się (Hirvensalo, 2004), że jedynie odwzorowania unitarne spełniają powyższe wymogi. Bramki kwantowe są zatem tworzone przez macierze unitarne.

W niniejszym podrozdziale zostaną przedstawione podstawowe bramki kwantowe. Za pomocą bramek kwantowych wykonuje się operacje na kubitach i rejestrach kwantowych. Rezultat użycia bramki kwantowej na rejestrze kwantowym określa własność (2.13):

Własność 2.13. Stan rejestru kwantowego $|\phi\rangle$, po przetworzeniu przez bramkę kwantową opisaną macierzą unitarną U , jest wyrażony w postaci iloczynu macierzy U oraz wektora kolumnowego odpowiadającego rejestrowi $|\phi\rangle$:

$$|\phi'\rangle = U \cdot |\phi\rangle \quad (2.6)$$

Aby mnożenie we własności 2.13 było wykonalne, bramka i rejestr kwantowy muszą być odpowiednich rozmiarów. Przykładowo na trzykubitowy rejestr kwantowy, opisywany przez znormalizowany wektor w ośmiowymiarowej przestrzeni, można działać tylko bramkami trzykubitowymi, opisywanymi przez macierze unitarne rozmiaru 8×8 .

Do podstawowych kwantowych bramek logicznych, kluczowych dla niniejszej pracy, należą m.in. bramka negacji NOT, bramka obrotu $ROT(\Delta\theta)$, bramka Hadamarda HAD.

Bramka negacji NOT

Bramka negacji NOT zamienia współczynniki stanów bazowych $|0\rangle$ i $|1\rangle$ i jest określona macierzą:

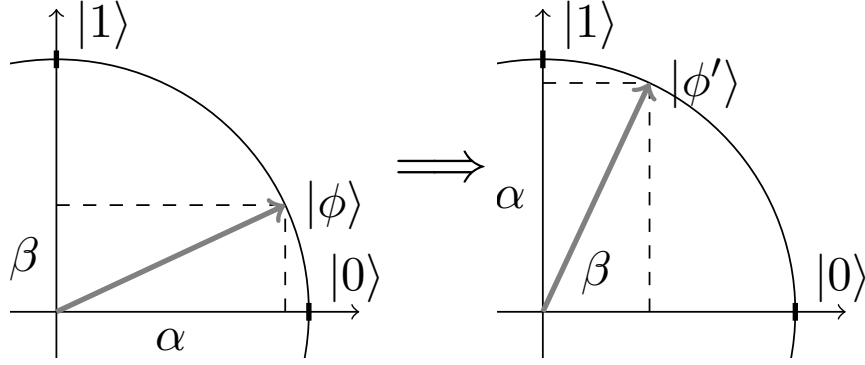
$$NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (2.7)$$

Dla stanów bazowych $|0\rangle$ i $|1\rangle$ bramka NOT działa tak jak klasyczna operacja negacji:

$$NOT|0\rangle = |1\rangle$$

$$NOT|1\rangle = |0\rangle$$

Natomiast dla dowolnej superpozycji tych stanów bazowych bramka powoduje zamianę związanych z nimi amplitud prawdopodobieństw, co zostało przedstawione na **Rysunku 2.4**.



RYS. 2.4: Bramka NOT zamienia amplitudy związane ze stanami bazowymi $|0\rangle$ i $|1\rangle$

Przykład 2.14. Działanie bramki NOT na stan kubit $|\phi\rangle = \frac{1}{3}|0\rangle + \frac{2\sqrt{2}}{3}|1\rangle$:

$$\begin{aligned} \text{NOT}|\phi\rangle &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \left(\frac{1}{3}|0\rangle + \frac{2\sqrt{2}}{3}|1\rangle \right) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{3} \\ \frac{2\sqrt{2}}{3} \end{bmatrix} = \\ &= \begin{bmatrix} \frac{2\sqrt{2}}{3} \\ \frac{1}{3} \end{bmatrix} = \frac{2\sqrt{2}}{3}|0\rangle + \frac{1}{3}|1\rangle \quad (2.8) \end{aligned}$$

Bramka obrotu $\text{ROT}(\Delta\theta)$

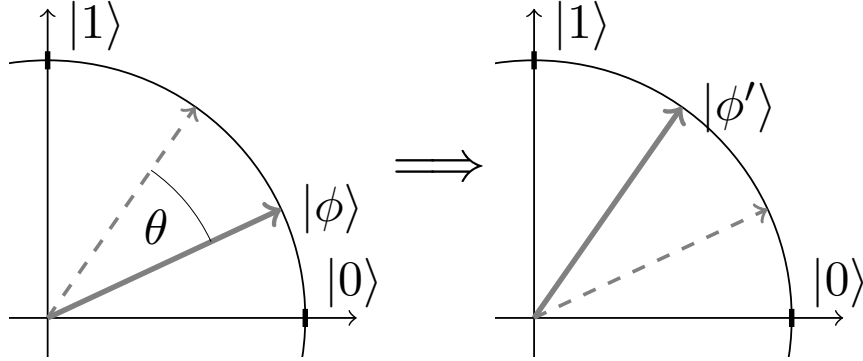
Kwantowa bramka $\text{ROT}(\Delta\theta)$ jest bramką kluczową dla niniejszej pracy, ponieważ opierają się na niej podstawowe kwantowe operatory genetyczne, które obracają wektor stanu w przestrzeni genu kwantowego:

$$\text{ROT}(\Delta\theta) = \begin{bmatrix} \cos \Delta\theta & -\sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{bmatrix} \quad \begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \text{ROT}(\Delta\theta) \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2.9)$$

Działanie bramki $\text{ROT}(\Delta\theta)$ zostało zilustrowane na **Rysunku 2.5**.

Uwaga: Używana tutaj bramka obrotu $\text{ROT}(\Delta\theta)$ NIE jest tą samą bramką obrotu, która jest wykorzystywana zazwyczaj w Informatyce Kwantowej ([Nielsen, Chuang, 2000](#); [Hirvensalo, 2004](#))! Typowa bramka obrotu w obliczeniach kwantowych dokonuje obrotu wektora na płaszczyźnie

zespolonej $\mathbb{C}$, natomiast w algorytmach kwantowo inspirowanych QIGA i QIEA w ogóle nie wykorzystywany wymiar urojony kubitów.



Rys. 2.5: Działanie bramki obrotu $\text{ROT}(\Delta\theta)$

Bramka Hadamarda HAD

Bramka Hadamarda określona jest za pomocą macierzy unitarnej:

$$\text{HAD} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.10)$$

Najczęściej wykorzystywaną własnością bramki Hadamarda jest to, że przekształca ona stan bazowy $|0\rangle$ w równomierną superpozycję stanów $|0\rangle$ i $|1\rangle$, z tego względu bramka ma istotne znaczenie dla obliczeń kwantowych.

$$\text{HAD}|0\rangle = \frac{\sqrt{2}}{2} (|0\rangle + |1\rangle)$$

$$\text{HAD}|1\rangle = \frac{\sqrt{2}}{2} (|0\rangle - |1\rangle)$$

ponieważ:

$$\text{HAD}|0\rangle = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} = \frac{\sqrt{2}}{2}|0\rangle + \frac{\sqrt{2}}{2}|1\rangle$$

$$\text{HAD}|1\rangle = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{bmatrix} = \frac{\sqrt{2}}{2}|0\rangle - \frac{\sqrt{2}}{2}|1\rangle$$

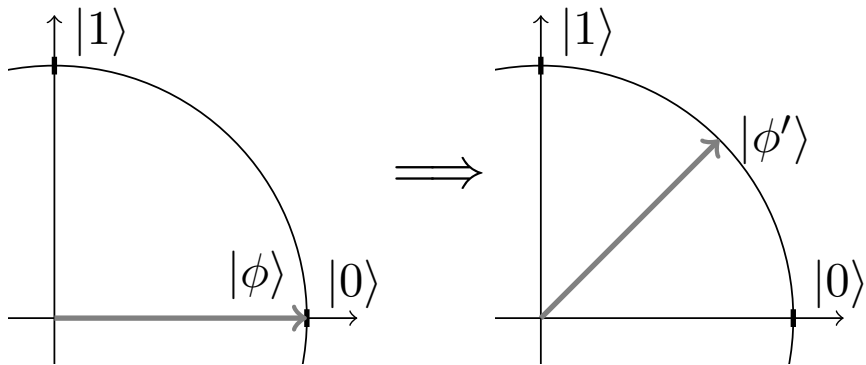
Działanie bramki HAD zostało zilustrowane na [Rysunku 2.6](#).

Wektory

$$\text{HAD}|0\rangle = \frac{\sqrt{2}}{2} (|0\rangle + |1\rangle)$$

$$\text{HAD}|1\rangle = \frac{\sqrt{2}}{2} (|0\rangle - |1\rangle)$$

stanowią ortonormalną bazę w przestrzeni stanów jednego kubit, nazywaną *bazą Hadamarda*.



Rys. 2.6: Działanie kwantowej bramki logicznej HAD

Rozdział 3

KWANTOWO INSPIROWANE ALGORYTMY GENETYCZNE I EWOLUCYJNE

Tak jak wspomniano wcześniej, kwantowo inspirowane algorytmy ewolucyjne (ang. *Quantum Inspired Evolutionary Algorithms* – **QIEA**) stanowią nową klasę metod w obszarze obliczeń ewolucyjnych. Podobnie jak do klasycznych algorytmów ewolucyjnych zalicza się kilka różnych metod (w szczególności algorytmy genetyczne, programowanie genetyczne, strategie ewolucyjne), tak samo szczególną podklasę w algorytmach QIEA stanowią kwantowo inspirowane algorytmy genetyczne (ang. *Quantum Inspired Genetic Algorithms* – **QIGA**), które charakteryzują się wykorzystaniem pojęcia kubitów do reprezentacji rozwiązań w genotypie kwantowym. Jednym z pierwszych i najbardziej typowych algorytmów tego typu jest algorytm **QIGA1** (Han, Kim, 2000). Ogólna klasa QIEA obejmuje również algorytmy wykorzystujące inne rodzaje reprezentacji rozwiązań jak np. algorytm **iQIEA** (da Cruz *et al.*, 2006), służący do optymalizacji numerycznej i niewykorzystujący pojęcia kubitów do reprezentacji rozwiązań.

W niniejszym rozdziale usystematyzowano możliwości wprowadzania elementów kwantowych na różnych etapach realizacji algorytmu ewolucyjnego, przedstawiono definicje podstawowych pojęć kwantowo inspirowanych algorytmów genetycznych i ewolucyjnych, oraz wykonano przegląd bieżącego stanu wiedzy w tym obszarze.

3.1 ELEMENTY KWANTOWE W ALGORYTMIE EWOLUCYJNYM

Elementy kwantowe mogą zostać wprowadzone do algorytmu ewolucyjnego na różnych etapach jego realizacji (Rysunek 1.2, str. 24):

- 1) REPREZENTACJA ROZWIĄZAŃ – elementy kwantowe mogą być wprowadzone do algorytmu ewolucyjnego przede wszystkim na po-

ziomie reprezentacji rozwiązań (np. ciągu kubitów zamiast ciągów binarnych). O ile w klasycznym algorytmie ewolucyjnym genotyp danego osobnika reprezentuje konkretny punkt w przestrzeni poszukiwań X , tak w kwantowo inspirowanym algorytmie genetycznym i ewolucyjnym każdy osobnik w populacji kwantowej Q reprezentuje rozkład prawdopodobieństwa próbkowania przestrzeni rozwiązań X .

- 2) INICJALIZACJA POPULACJI POZATKOWEJ – nowy rodzaj reprezentacji wiąże się z koniecznością innego sposobu inicjalizacji populacji bazowej $Q(0)$. W tym celu może być wykorzystana np. kwantowa bramka Hadamarda (**Rozdział 2.3**), generująca równomierną superpozycję stanów bazowych $|0\rangle$ i $|1\rangle$.
- 3) OCENA ROZWIĄZAŃ – nowy rodzaj reprezentacji wymaga zastosowania odpowiednich miar oceny osobników kwantowych. Oprócz miar deterministycznych, stosowanych typowo w klasycznym algorytmie ewolucyjnym, w algorytmach QIEA mogą być stosowane również miary stochastyczne, które przypisują genotypowi kwantowemu wartość po dokonaniu próbowania przestrzeni zgodnie z zapisanym w nim rozkładem prawdopodobieństwa.
- 4) KRYTERIUM STOPU – obecność elementu kwantowego w algorytmie ewolucyjnym pozwala również w nowy sposób definiować kryterium stopu, np. zatrzymanie algorytmu może następować po całkowitym zaniku superpozycji stanów we wszystkich genach kwantowych.
- 5) OPERATORY GENETYCZNE – mogą być tutaj wyodrębnione dwie grupy operatorów dla genotypów kwantowych:
 1. Możliwe jest uogólnienie wielu klasycznych operatorów do nowego rodzaju reprezentacji (np. ciągi kubitów mogą być krzyżowane ze sobą na takiej samej zasadzie jak zwykłe ciągi binarne).
 2. Możliwe jest również zastosowanie zupełnie nowych operatorów genetycznych, modelujących zjawiska kwantowe i opierających się na zasadzie ewolucji unitarnej układu kwantowego

(np. obroty wektorów w przestrzeni stanów kubitu, interferencja amplitud prawdopodobieństwa etc.).

Ogólny schemat algorytmu QIEA został przedstawiony na [Listing 1](#) i nie różni się on znacząco od ogólnego schematu klasycznego algorytmu ewolucyjnego. Oprócz klasycznej populacji osobników P , w algorytmie przetwarza się w sposób iteracyjny populację osobników kwantowych Q . Rozkłady prawdopodobieństw, zapisane w Q są modyfikowane za pomocą odpowiednich, nowych operatorów genetycznych, aby w kolejnych generacjach zwiększać prawdopodobieństwo znalezienia najlepszego rozwiązania w przestrzeni X . Iteracje te są powtarzane aż do spełnienia kryterium stopu (np. osiągnięcie maksymalnego numeru generacji $t_{\max}$).

3.2 ALGORYTMY QIEA – WYBRANE POJĘCIA PODSTAWOWE

Niech X oznacza przeszukiwaną przestrzeń rozwiązań, Ω – przestrzeń zdarzeń elementarnych, odpowiadających wybieraniu pojedynczych elementów przestrzeni X przez algorytm, oraz niech $\mathcal{F}_\Omega$ stanowi σ -ciało zdarzeń z Ω . Do podstawowych pojęć kwantowo inspirowanych algorytmów ewolucyjnych należą:

- *gen kwantowy* $|g\rangle$ – pojedynczy element chromosomu kwantowego. Koduje rozkład prawdopodobieństwa $\Pr_g : \mathcal{F}_A \mapsto [0, 1]$ ekspresji wartości genu, spośród zbioru dopuszczalnych wartości A .
- *binarny gen kwantowy* – w przypadku kwantowego kodowania binarnego (dwuelementowy zbiór możliwych wartości $A = \{0, 1\}$), binarny gen kwantowy $|g\rangle$, kodujący 2-punktowy rozkład prawdopodobieństwa $\Pr_g : \mathcal{F}_{\{0,1\}} \mapsto [0, 1]$, jest reprezentowany w postaci kubitu (z pominięciem jego części urojonej).
- *binarny chromosom kwantowy* $|g_1\rangle|g_2\rangle \dots |g_N\rangle$ – uporządkowany N -elementowy łańcuch binarnych genów kwantowych, spośród których każdy może przyjąć jedną z dwóch wartości z ustalonym prawdopodobieństwem. Taki binarny chromosom kwantowy odpowiada zatem ciągowi N 2-punktowych rozkładów prawdopodobieństw.

- *genotyp kwantowy* q – zbiór chromosomów (w szczególnym przypadku jednoelementowy, $q = \{|g_1\rangle|g_2\rangle \dots |g_N\rangle\}$) lub inny rodzaj struktury, zapisującej prawdopodobieństwa ekspresji cech osobnika. Genotyp kwantowy q koduje rozkład prawdopodobieństwa $\Pr_q : \mathcal{F}_\Omega \mapsto [0, 1]$ wyboru poszczególnych elementów przestrzeni rozwiązań X przez algorytm.
- *populacja kwantowa* $Q = \{q_1, q_2, \dots, q_{|Q|}\}$ – zbiór osobników w postaci genotypów kwantowych o określonej liczebności. Populacja kwantowa Q jest zatem izomorficzna z $|Q|$ -elementowym zbiorem przestrzeni probabilistycznych $\{(\Omega, \mathcal{F}_\Omega, \Pr_1), (\Omega, \mathcal{F}_\Omega, \Pr_2), \dots, (\Omega, \mathcal{F}_\Omega, \Pr_{|Q|})\}$.
- *obserwacja stanu osobnika* q – wygenerowanie pojedynczego elementu przestrzeni poszukiwań $x \in X$, poprzez wylosowanie elementu zgodnie z rozkładem prawdopodobieństwa zapisanym w q (próbkiwanie przestrzeni X zgodnie z tym rozkładem). Poprzez obserwację stanu i wygenerowanie osobnika klasycznego $x \in X$ dokonuje się ukształtowanie fenotypu i tzw. ekspresja wartości genów.
- *populacja klasyczna* P – populacja utworzona poprzez $|P|$ -krotne próbkowanie przestrzeni rozwiązań X zgodnie z rozkładami prawdopodobieństw, zapisanymi w populacji kwantowej Q .

Listing 1: Ogólny schemat algorytmu QIEA

```

1 begin
2    $t \leftarrow 0$ 
3   Inicjalizuj  $Q(0)$ 
4   Zaobserwuj populację klasyczną  $P(0)$ 
5   Oceń osobniki w populacji  $P(0)$ 
6   while (nie zachodzi kryterium stopu) do
7      $t \leftarrow t + 1$ 
8     Zaobserwuj populację klasyczną  $P(t)$ 
9     Oceń osobniki w populacji  $P(t)$ 
10    Zastosuj operatory genetyczne na  $Q(t - 1)$ 
11    ...
12  end
13 end

```

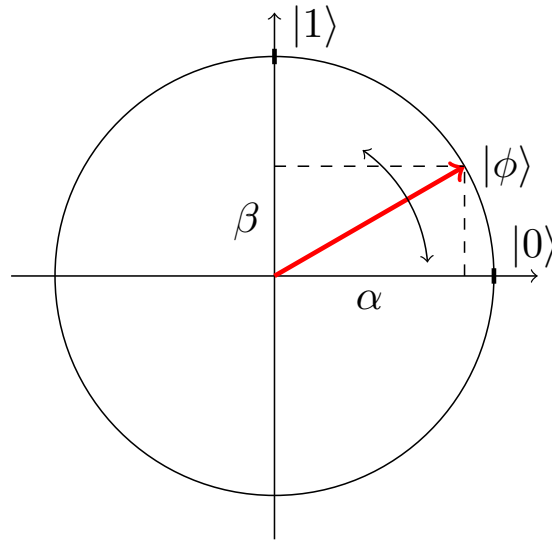
3.2.1 Binarny gen kwantowy i jego przestrzeń stanów

W algorytmach QIGA (Han, Kim, 2000) reprezentacja rozwiązań w genach kwantowych opiera się bezpośrednio na pojęcie kubitów:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (3.1)$$

gdzie: $\alpha, \beta \in \mathbb{R}$, $|\alpha|^2 + |\beta|^2 = 1$.

Gen kwantowy $|\phi\rangle$ przyjmuje wartość 0 z prawdopodobieństwem $|\alpha|^2$ i wartość 1 z prawdopodobieństwem $|\beta|^2$. Pomimo iż podstawowe pojęcia Informatyki Kwantowej zdefiniowane są w oparciu o ciało liczb zespolonych, w kwantowo inspirowanych algorytmach genetycznych i ewolucyjnych przyjmuje się ograniczenie do zbioru liczb rzeczywistych. Ograniczenie przestrzeni stanu binarnego genu kwantowego do zbioru liczb rzeczywistych pozwala na reprezentowanie stanu genów jako pojedynczych jednostkowych wektorów na płaszczyźnie (Rysunek 3.1). Podobnie, w „prawdziwym” komputerze kwantowym odczyt stanu kubitów niszczy nieodwracalnie zapisaną w nim superpozycję stanów. Natomiast w przypadku symulacji kubitów, odpowiadający mu wektor stanu może być zapisany i podlegać wielokrotnemu próbkowaniu.



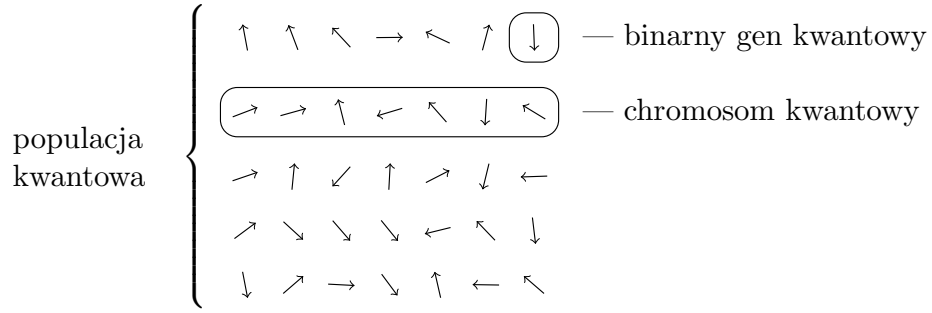
RYS. 3.1: Ilustracja stanu binarnego genu kwantowego

3.2.2 Populacja kwantowa Q

Posługując się uproszczoną notacją, wprowadzoną w [Rozdziale 2.1](#), w związku i przybliżony stan zbioru niezależnych genów kwantowych w chromosomie można przedstawić za pomocą ciągu wektorów np. $\rightarrow\uparrow\rightarrow\uparrow\uparrow$ jest równoważne 5 binarnym genom kwantowym w stanach: $|0\rangle, |1\rangle, |0\rangle, |1\rangle, |1\rangle$, natomiast zapis $\rightarrow\uparrow\leftarrow\searrow$ odpowiada czterem genom kwantowym, spośród których każdy znajduje się w określonej superpozycji stanów bazowych $|0\rangle$ i $|1\rangle$. W ten sposób na [Rysunku 3.2](#) zilustrowano w uproszczony sposób pewien stan populacji kwantowej Q, składającej się z 5 osobników kwantowych (5 chromosomów), spośród których każdy zbudowany jest z 7 binarnych genów kwantowych.

3.2.3 Podstawowe kwantowe operatory genetyczne

Stany genów kwantowych modyfikowane są przez kwantowe bramki logiczne, opisywane macierzami unitarnymi i przedstawione w [Rozdziale 2.3](#) na [str. 40](#). Wybrane etapy algorytmów QIGA i wykorzystywane w nich bramki kwantowe to:



RYS. 3.2: Ilustracja stanu binarnej populacji kwantowej

INICJALIZACJA – kwantowa bramka Hadamarda, przetwarzająca stan bazowy $|0\rangle$ w stan superpozycji $[\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2}]^T$:

$$\text{HAD} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Ta własność kwantowej bramki Hadamarda, pozwalająca na wygenerowanie stanu superpozycji $[\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2}]^T$ ze stanu $|0\rangle$, ma znaczenie dla ewentualnej możliwości realizacji algorytmu w strukturze komputera kwantowego w przyszłości. W algorytmach QIGA, implementowanych na klasycznym komputerze, zmienne mogą być inicjalizowane bezpośrednio stanem superpozycji.

OPERATOR MUTACJI – kwantowa bramka NOT, zamieniająca amplitudy prawdopodobieństw $\alpha \leftrightarrow \beta$, związane z obserwacją wartości genu 0 i 1 w chromosomie kwantowym:

$$\text{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

POZOSTAŁE OPERATORY – kwantowa bramka obrotu o kąt $\Delta\theta$, powodująca zmianę rozkładu prawdopodobieństwa obserwacji wartości genów 0 i 1 poprzez obrót wektora stanu na płaszczyźnie (Rysunek 3.1):

$$R(\Delta\theta) = \begin{bmatrix} \cos \Delta\theta & -\sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{bmatrix}$$

3.3 PRZEGLĄD LITERATURY TEMATU

Badania nad obszarem *quantum-inspired computational intelligence* rozpoczęły się w 1995 od dwóch pionierskich prac (Menneer, Narayanan, 1995), (Moore, Narayanan, 1995) i do roku obecnego opublikowano na temat tego rodzaju metod łącznie około kilkuset publikacji¹. Zdecydowana większość to prace o charakterze aplikacyjnym, pokazujące możliwość skutecznego zastosowania algorytmów QIGA oraz QIEA w szerokim zakresie problemów, oraz przewagę tych algorytmów nad innymi nowoczesnymi heurystycznymi metodami przeszukiwań. Menneer oraz Narayanan rozważali w pierwszej kolejności możliwość wprowadzania elementów inspirowanych kwantowo do poprawy efektywności działania sztucznych sieci neuronowych (Menneer, Narayanan, 1995). Możliwość wprowadzenia dodatkowych elementów charakterystycznych dla systemów Informatyki Kwantowej do algorytmu ewolucyjnego została po raz pierwszy zasygnalizowana rok później w pracy (Narayanan, Moore, 1996), a w kolejnych latach idea ta została spopularyzowana przez Hana oraz Kima w licznych pracach (Han, Kim, 2000, 2001; Han *et al.*, 2001; Han, Kim, 2002a,b; Kim *et al.*, 2003). W pracy (Han, Kim, 2000) przedstawiony został pierwotny algorytm **QIGA1**, który badany był następnie w licznych pracach i który jest wykorzystywany w dalszej części rozprawy. Zainteresowanie algorytmami QIGA i QIEA wzrosło znacząco po roku 2004 (kilkukrotnie większa liczba publikacji w ciągu roku, Rysunek 3.3), gdy da Cruz, Barbosa, Pacheco, Vellasco przedstawili pierwszą propozycję algorytmu **iQIEA** (da Cruz *et al.*, 2004) dla problemów kodowanych liczbami rzeczywistymi (optymalizacja numeryczna). Od roku 2000 opublikowano łącznie blisko 300 publikacji, których autorzy wprowadzali różnego rodzaju modyfikacje i dodatkowe elementy do oryginalnych algorytmów QIGA i QIEA, mające na celu dalszą poprawę efektywności tych algorytmów.

Do najbardziej znaczących publikacji w rozważanym obszarze należą według autora rozprawy następujące prace: (Narayanan, Moore, 1996; Han,

¹Usystematyzowane zestawienie wszystkich publikacji z tego obszaru, które były dostępne autorowi, zamieszczone jest w postaci elektronicznej pod adresem: <http://rnowotniak.kis.p.lodz.pl/quantum/>

Kim, 2000, 2001; Bin *et al.*, 2003; Han, Kim, 2003b, 2004; Giraldi *et al.*, 2004; da Cruz *et al.*, 2004; Han, Kim, 2006; da Cruz *et al.*, 2006; Zhang, Rong, 2007; Defoin-Platel *et al.*, 2007; Khosraviani *et al.*, 2008; Liu *et al.*, 2008; Platel *et al.*, 2009; Li *et al.*, 2009b; Mahdabi *et al.*, 2009; Zhang *et al.*, 2010; Zhang, 2010a; Li *et al.*, 2011b; Chen, Ding, 2011; Manju, Nigam, 2012). Najistotniejsze spośród tych prac zostały przedstawione w sposób chronologiczny w Tabeli 3.1, w której uwzględniono również publikacje autora rozprawy. Na Rysunku 3.3 przedstawiono liczbę publikacji naukowych dotyczących obszaru QIEA kolejno w latach 1996-2014. Zestawienie to uwzględnia prace indeksowane w bazach Google Scholar, IEEE, Elsevier oraz ScienceDirect. Próby usystematyzowania wszystkich zaprezentowanych modyfikacji algorytmów QIEA zostały dotychczas podjęte tylko w nielicznych pracach przeglądowych (Zhang, 2010a; Manju, Nigam, 2012).

TABELA 3.1: Najważniejsze prace w kolejności chronologicznej, z uwzględnieniem prac autora

Rok	Publikacje
1995	(Menneer, Narayanan, 1995) – pierwsza publikacja sugerująca możliwość wykorzystania pojęć Informatyki Kwantowej w sztucznych sieciach neuronowych
1996	(Narayanan, Moore, 1996) – pierwsze zasygnalizowanie możliwości wykorzystania elementów kwantowych do poprawy efektywności algorytmu genetycznego
2000	(Han, Kim, 2000) – pierwotny algorytm QIGA1 , zaproponowany przez Hana i Kima
2003	(Bin <i>et al.</i> , 2003) – pierwsza propozycja modyfikacji algorytmu QIGA1 (algorytm GAQPR), nowe operatory krzyżowania i mutacji
	(Han, Kim, 2003b,a) – prace doświadczalne, dotyczące ustalenia właściwych wartości parametrów algorytmów QIGA
	(Han, 2003) – rozprawa doktorska Hana na temat algorytmów QIGA i ich wybranych własności
	(Han, Kim, 2004) – propozycja zmodyfikowanego algorytmu QIGA, wykorzystującego nowe bramki kwantowe H_c oraz nowe kryterium stopu

TABELA 3.1: Najważniejsze prace oraz prace autora c.d.

Rok	Publikacje
2004	(Giraldi <i>et al.</i> , 2004) – znacząca praca przeglądowa, prezentująca różne możliwe podejścia łączenia metod ewolucyjnych i obliczeń kwantowych
	(da Cruz <i>et al.</i> , 2004) – pierwsza propozycja kwantowo inspirowanego algorytmu ewolucyjnego dla zadań kodowanych liczbami rzeczywistymi iQIEA
2005	(Khorsand, Akbarzadeh-T, 2005) – pierwsza praca sygnalizująca możliwość zastosowania metaoptymalizacji do doboru parametrów algorytmów QIEA
2007	(Zhang, Rong, 2007) – propozycja Zhanga algorytmu rQIEA, kwantowo inspirowanego algorytmu ewolucyjnego dla zadań kodowanych liczbami rzeczywistymi, kodującego liczby rzeczywiste we współrzędnych amplitud prawdopodobieństwa
2008	(Khosraviani <i>et al.</i> , 2008) – analiza zbieżności algorytmu QIGA o bardzo uproszczonych założeniach (pojedynczy osobnik w populacji), wykorzystująca teorię łańcuchów Markova
	(Sofge, 2008) – praca przeglądowa, analiza różnych możliwości łączenia obliczeń ewolucyjnych z obliczeniami kwantowymi
2009	(Jopek, Nowotniak, Postolski, Babout, Janaszewski, 2009) – przykład skutecznego zastosowania algorytmu QIGA1 w wybranym obszarze aplikacyjnym (rozpoznawanie obrazów)
2010	(Zhang, 2010a) – wyczerpująca praca przeglądowa Zhanga wraz z wynikami licznych eksperymentów numerycznych, porównujących efektywność wybranych algorytmów QIGA i QIEA
	(Nowotniak, Kucharski, 2010b) – metaoptymalizacja algorytmu iQIEA dla czterech prostych funkcji rzeczywistych
	(Nowotniak, Kucharski, 2010a) – teoretyczna analiza propagacji bloków budujących w algorytmach QIGA
2011	(Chen, Ding, 2011) – teoretyczna analiza zbieżności uproszczonych i zmodyfikowanych algorytmów QIGA wraz z eksperymentami numerycznymi

TABELA 3.1: Najważniejsze prace oraz prace autora c.d.

Rok	Publikacje
2012	(Nowotniak, Kucharski, 2012a) – teoretyczna analiza zbieżności algorytmów QIEA z wykorzystaniem twierdzenia Banacha o punkcie stałym odwzorowania zwężającego
	(Nowotniak, 2012) – efektywna implementacja algorytmu iQIEA za pomocą obliczeń równoległych z wykorzystaniem Polskiej Infrastruktury Gridowej PL-GRID
	(Izadinia, Ebadzadeh, 2012) – propozycja algorytmu kwantowo inspirowanej strategii ewolucyjnej oraz doświadczalne porównanie efektywności algorytmu z algorytmem CMAES
	(Nowotniak, Kucharski, 2012b) – metaoptymalizacja algorytmu QIGA1 za pomocą implementacji w technologii GPGPU. Wykazanie przewagi algorytmu dostrojonego QIGA1* nad algorytmem pierwotnym
	(Manju, Nigam, 2012) – wyczerpująca praca przeglądowa, obejmująca 748 prac z obszaru kwantowo inspirowanych metod inteligencji obliczeniowej
2014	(Zhou, Cao, 2014) – propozycja zmodyfikowanego algorytmu QIEA oraz sposobu jego efektywnej realizacji w środowisku obliczeń równoległych; badania dla wybranych przykładów aplikacyjnych
	(Nowotniak, Kucharski, 2014) – teoretyczne podstawy kwantowo inspirowanych algorytmów genetycznych wyższych rzędów; wykazanie możliwości wykorzystania drugiego rzędu kwantowości kwantowo inspirowanego algorytmu genetycznego QIGA jako metody poprawy efektywności algorytmu; autorska propozycja algorytmu QIGA2 , wykazanie jego znacznej przewagi nad pierwotnym algorytmem QIGA1 oraz dostrojonym QIGA1*

Analizy teoretyczne

W nielicznych publikacjach podejmowano próby teoretycznej analizy własności algorytmów QIGA i QIEA. Do takich prac należą (Han, Kim, 2006; Khosraviani *et al.*, 2008; Li *et al.*, 2011b; Chen, Ding, 2011; Malossini *et al.*,

2008). Do prac o takim charakterze należą również prace autora (Nowotniak, Kucharski, 2012a), (Nowotniak, Kucharski, 2010a), (Nowotniak, Kucharski, 2014).

Sposoby wprowadzania elementów kwantowych

Elementy inspirowane Informatyką Kwantową wprowadzane były do algorytmów ewolucyjnych często w sposób bardzo arbitralny. Autorzy wielu prac wykazywali możliwość wykorzystania dodatkowych elementów kwantowych do poprawy efektywności algorytmów za pomocą różnych przykładów aplikacyjnych. Przedmiotem badań były m.in.:

NOWE SPOSOBY INICJALIZACJI – (Fan *et al.*, 2008a; Lv, Li, 2008; Chung *et al.*, 2011)

ZMIENIONY SPOSÓB REPREZENTACJI ROZWIĄZAŃ – (da Cruz *et al.*, 2004, 2005; Zhou *et al.*, 2006; da Cruz *et al.*, 2006; Fan *et al.*, 2007; Layeb, Saidouni, 2007; Xiao *et al.*, 2008; Huo *et al.*, 2008; Li, Li, 2008; Zhang *et al.*, 2008b; Qin *et al.*, 2008; Fan *et al.*, 2008b,a; Shen, Liu, 2008; Xiao *et al.*, 2009; Xiong *et al.*, 2009; Cao, Shang, 2009; Zhao *et al.*, 2009a; Izadinia, Ebadzadeh, 2009; de Pinho *et al.*, 2009; Lin, Waller, 2009; da Cruz *et al.*, 2010; Ashish Mani, 2010; Wang, Zhou, 2010; Li *et al.*, 2011b; Zongyao, Zhou, 2011; Silveira *et al.*, 2012; Liao *et al.*, 2010)

KWANTOWE OPERATORY GENETYCZNE – (Bin *et al.*, 2003; Zhang *et al.*, 2003; Han, Kim, 2004; Hui *et al.*, 2004; Wang *et al.*, 2005a; da Cruz *et al.*, 2005; Zhang *et al.*, 2006; Layeb *et al.*, 2006; Xiao *et al.*, 2008; Dai, Li, 2008; Mahdabi *et al.*, 2008; Xiao *et al.*, 2009; Fu *et al.*, 2009; Wang *et al.*, 2009; Imabeppu *et al.*, 2009; Xing *et al.*, 2009c; Narayan, Patvardhan, 2009; JEONG *et al.*, 2009; Gu *et al.*, 2009b; Tao *et al.*, 2010; Teng *et al.*, 2010; Li *et al.*, 2011b; Zongyao, Zhou, 2011; Hao *et al.*, 2010; Imabeppu *et al.*, 2008; Liu *et al.*, 2010; Tayarani, Akbarzadeh. T, 2008; Teng *et al.*, 2008; Yin *et al.*, 2012). W pracy (Zhang *et al.*, 2010) została porównana efektywność sześciu różnych bramek obrotu jako operatorów genetycznych w algorytmach QIGA.

Prace o charakterze aplikacyjnym

Zdecydowana większość prac na temat algorytmów QIEA, opublikowanych w ciągu minionej dekady, miała charakter różnorodnych przykładów aplikacyjnych. W licznych publikacjach została wykazana skuteczność algorytmów QIGA i QIEA w najróżniejszych obszarach, oraz przewaga tych algorytmów nad klasycznymi metodami optymalizacji oraz nad innymi znanymi metodami ewolucyjnymi. Obszary aplikacyjne, w których wykazywano skuteczność i efektywność algorytmów QIGA i QIEA to m.in.:

ZADANIA SZEREGOWANIA – (Kim *et al.*, 2003; Wang *et al.*, 2005c,a; Gu *et al.*, 2009a; Niu *et al.*, 2010; Gong *et al.*, 2009; Wang *et al.*, 2005b; Gu *et al.*, 2009c; Yasin *et al.*, 2012)

ROZPOZNAWANIE OBRAZÓW – (Jang *et al.*, 2003, 2004; Jopek *et al.*, 2009; Bhattacharyya *et al.*, 2010; Hao, Shiyong, 2012; Yu, Fan, 2008; Melo *et al.*, 2008; Song, Liu, 2013; Zhang *et al.*, 2012)

PRZETWARZANIE SYGNAŁÓW – (Bin *et al.*, 2003; Fan *et al.*, 2007; Bhattacharyya *et al.*, 2010; Talbi *et al.*, 2004a, 2007)

SELEKCJA CECH – (Zhang *et al.*, 2006; Jopek *et al.*, 2009)

KLASTERYZACJA – (Zhou *et al.*, 2006; Ramdane *et al.*, 2010; Xiao *et al.*, 2009)

ENERGETYKA – (Chung *et al.*, 2011; Al-Othman *et al.*, 2007; Alfares, Esat, 2006; dos Santos Nicolau *et al.*, 2012; Ibrahim *et al.*, 2011; Liu *et al.*, 2009; Miao *et al.*, 2009; Pareek *et al.*, 2011; Vlachogiannis, Lee, 2008; Vlachogiannis, Østergaard, 2009)

SIECI KOMPUTEROWE – (Ali *et al.*, 2011; Gong *et al.*, 2009; Radha, Rughooputh, 2010; Wang-xin *et al.*, 2006; Xing *et al.*, 2010; Li *et al.*, 2009a)

EKONOMIA – (Araújo *et al.*, 2010; Neto *et al.*, 2010; Araujo *et al.*, 2008a,b; Neto *et al.*, 2011)

SYSTEMY UCZĄCE SIĘ – (Li *et al.*, 2011b,a; Arpaia *et al.*, 2011; Coelho *et al.*, 2008; Dias, Pacheco, 2009; Gharipour *et al.*, 2011; Shill *et al.*, 2011;

Wang *et al.*, 2010; Luo *et al.*, 2008; Dias, Pacheco, 2012; Cho *et al.*, 2011; Araujo *et al.*, 2008c; Arpaia *et al.*, 2007; Zhang *et al.*, 2008c)

ZADANIA STEROWANIA – (Arpaia *et al.*, 2007, 2011; Park *et al.*, 2010)

OPTYMALIZACJA WIELOKRYTERIALNA – (Kim *et al.*, 2006; Li, Wang, 2006; Talbi *et al.*, 2007; Charan Kumari *et al.*, 2013; Wei *et al.*, 2008)

I WIELE INNYCH – (Guo *et al.*, 2007, 2009b; Hu, 2010; Khan, Akter, 2009; Lv, Liu, 2007; Muramoto *et al.*, 2012; Popa *et al.*, 2010; Ryu *et al.*, 2012; Wang, Li, 2009; Wang *et al.*, 2010, 2012b; Yan *et al.*, 2009; Zhang, 2010b; Zhao *et al.*, 2009b; Zheng, Liu, 2011; Chen, 2010; Wang, Shi, 2010)

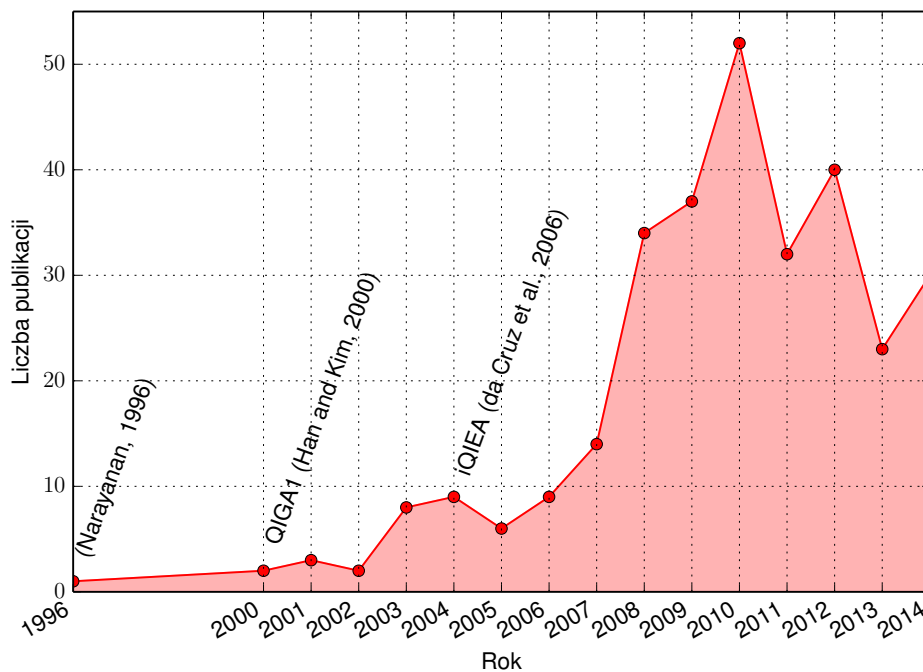
W wielu publikacjach wykorzystywane były również znane typowe zadania optymalizacji kombinatorycznej, które wielokrotnie wykorzystywano do porównywania efektywności algorytmów, wykorzystywano m.in. **problem plecakowy** (Chen *et al.*, 2010; Patvardhan *et al.*, 2007; Wang *et al.*, 2012a; Zhang *et al.*, 2008a; Koppaka, Hota, 2010; Liu *et al.*, 2010; Mahdabi *et al.*, 2009; Li *et al.*, 2009b; Narayan, Patvardhan, 2009; Shu, 2009; Imabeppu *et al.*, 2009, 2008; Zhang *et al.*, 2008a; Shen, Liu, 2008), **problem komiwojażera** (Silveira *et al.*, 2012; Feng *et al.*, 2006; Talbi *et al.*, 2004b; Yu *et al.*, 2009; Dai, Li, 2008; Wang *et al.*, 2007), **problem spełnialności SAT** (Layeb, Saidouni, 2008a,b). Jedną z bardzo nielicznych prac krytycznych, której autorzy argumentują, że przewaga algorytmu QIGA jest nieznaczna w porównaniu z odpowiednio zmodyfikowanymi klasycznymi algorytmami GA, jest (Toosi, Mashhadi, 2005).

Rozszerzenia, metody hybrydowe, zrównoleglenie algorytmów

Podejmowano również próby tworzenia hybrydowych algorytmów QIEA, połączonych z algorytmami kulturowymi (da Cruz *et al.*, 2005; Li *et al.*, 2012), sztucznymi systemami immunologicznymi (Jiao *et al.*, 2008; Li *et al.*, 2004) oraz metodami inteligencji roju (Wang *et al.*, 2007; Xiao *et al.*, 2009; Fang *et al.*, 2009; Hamed *et al.*, 2009; Luitel, Venayagamoorthy, 2010; Hossain *et al.*, 2005). W licznych pracach rozważano również modele wyspowe w algorytmie (migracyjność, subpopulacje) (Guo *et al.*, 2009a; Wang, Zhou, 2010; Liu, You, 2009; Malossini *et al.*, 2008), modele selekcji elitarniej (Lv,

Li, 2008; Kumari *et al.*, 2012; Wu, Li, 2011; Hossain *et al.*, 2005), koewolucję (Gu *et al.*, 2009b; Liu, You, 2009) oraz adaptacyjne metody doboru wartości parametrów algorytmów (Xing *et al.*, 2009a; Izadinia, Ebadzadeh, 2012; Ashish Mani, 2010; Mani, Patvardhan, 2010; Xing *et al.*, 2009b,c,a).

W wielu pracach proponowano również wprowadzenie elementów inspirowanych kwantowo do metod zaliczanych do obliczeń ewolucyjnych, jednak innych niż algorytmy ewolucyjne. Było to m.in. kwantowo inspirowane programowanie genetyczne (Dias *et al.*, 2011), kwantowo inspirowana ewolucja różnicowa (Draa *et al.*, 2010; Su, Yang, 2008; Zheng, Yamashiro, 2010a,b), kwantowo inspirowane losowe błędzenie (Koppaka, Hota, 2010), kwantowo inspirowane symulowane wyżarzanie (Shu, 2009), kwantowo inspirowane metody inteligencji roju (Xue *et al.*, 2010; Xiao, Liu, 2009), kwantowo inspirowane programowanie ewolucyjne (Yang, Jiao, 2003).



RYS. 3.3: Wykres przedstawiający liczbę publikacji, dotyczących algorytmów QIEA, kolejno w latach 1996-2014. Źródło: Opracowanie własne.

3.4 ALGORYTMY PIERWOTNE

W dwóch kolejnych podrozdziałach opisane zostały algorytmy pierwotne QIGA1 (Han, Kim, 2000) oraz iQIEA (da Cruz *et al.*, 2006), analizowane szczegółowo oraz modyfikowane w kolejnych rozdziałach rozprawy.

3.4.1 Kwantowo inspirowany algorytm genetyczny QIGA1

Pierwotny algorytm QIGA1 został zaprezentowany w pracy (Han, Kim, 2000) wraz z wynikami eksperymentów numerycznych, wykonanych dla problemu plecakowego (optymalizacja kombinatoryczna). Bardzo szczegółowy opis tego algorytmu można znaleźć m.in. w (Zhang, 2010a, str. 7). Reprezentacja rozwiązań w algorytmie QIGA1 oparta jest na niezależnych binarnych genach kwantowych, modelowanych za pomocą osobnych kubitów. Dzięki temu nie ma zastosowania własność (Własność 2.9 na str. 38, Rozdział 2), mówiąca o wykładniczym wzroście wymiaru przestrzeni w komputerze kwantowym. Ponieważ w reprezentacji wykorzystywane są niezależne kubity, nie jest potrzebny komputer kwantowy do efektywnego działania tego algorytmu.

Populacja kwantowa Q w algorytmie QIGA1 składa się z chromosomów kwantowych:

$$q = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \dots & \alpha_N \\ \beta_1 & \beta_2 & \dots & \beta_N \end{array} \right] \quad (3.2)$$

Operatory genetyczne w algorytmie oparte są natomiast na kwantowych bramkach obrotu. Pseudokod algorytmu został przedstawiony na Listingu 2 i przebiega on według następujących kroków:

- (2) Wykonywana jest inicjalizacja licznika generacji $t = 0$.
- (3) Stan wszystkich binarnych genów kwantowych w populacji $Q(0)$ jest początkowo ustawiany na stan superpozycji, co może być zrealizowane za pomocą zastosowania **kwantowej bramki Hadamarda** (wzór (2.10) na str. 43) na stanie bazowym $|0\rangle$:

$$\text{HAD}|0\rangle = \left(\frac{\sqrt{2}}{2}|0\rangle + \frac{\sqrt{2}}{2}|1\rangle \right) = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

Listing 2: Pierwotny algorytm QIGA1 (Han, Kim, 2000)

```

1 begin
2    $t \leftarrow 0$ 
3   Inicjalizuj  $Q(0)$ 
4   Utwórz populację  $P(0)$  poprzez obserwację stanów  $Q(0)$ 
5   Uruchom procedurę naprawczą na  $P(0)$ 
6   Oceń osobniki w populacji  $P(0)$ , obliczając wartości  $f(x)$ 
7   Zapisz najlepszego osobnika z populacji  $P(0)$  w  $b$ 
8   while  $t \leq t_{\max}$  do
9      $t \leftarrow t + 1$ 
10    Utwórz populację  $P(t)$  poprzez obserwację stanów  $Q(t - 1)$ 
11    Uruchom procedurę naprawczą na  $P(t)$ 
12    Oceń osobniki w populacji  $P(t)$ , obliczając wartości  $f(x)$ 
13    Aktualizuj populację  $Q(t)$  za pomocą bramek kwantowych
14    Zapisz najlepszego osobnika z populacji  $P(t)$  w  $b$ 
15  end
16 end

```

Stan takiej superpozycji wszystkich genów kwantowych odpowiada możliwości wyboru przez algorytm każdego elementu z przestrzeni X z takim samym prawdopodobieństwem.

- (4) Tworzona jest początkowa populacja klasyczna $P(0)$ poprzez obserwację stanów z populacji kwantowej $Q(0)$ (losowanie elementów przestrzeni X zgodnie z rozkładami prawdopodobieństw zapisanymi w Q).
- (5) Zastosowanie procedury naprawczej na ciągach binarnych w populacji P . Jest to krok opcjonalny i specyficzny dla zadania optymalizacji, rozwiązywanego przez algorytm, np. dla problemu plecakowego procedura naprawcza polega na zastąpieniu rozwiązań, które przekraczają dopuszczalną wagę, rozwiązaniami dopuszczalnymi.
- (6) Osobniki z zaobserwowanej populacji P są poddawane ocenie, tzn. dla każdego rozwiązania w $x \in P$ obliczana jest wartość $f(x)$.

- (7) Najlepszy osobnik z populacji P zapamiętywany jest w zmiennej b .
- (8-15) Analogiczne operacje wykonywane są w głównej pętli algorytmu dla kolejnych generacji t , aż do spełnienia kryterium stopu np. osiągnięcia maksymalnego numer generacji $t_{\max}$.
- (13) Wykonywana jest aktualizacja stanu populacji kwantowej Q . Nowa populacja kwantowa $Q(t+1)$ tworzona jest poprzez zastosowanie odpowiednich **kwantowych bramek obrotu** ROT (wzór (2.9) na str. 42) na wszystkich genach kwantowych w dotychczasowej populacji Q . Nowe wartości amplitud α' i β' wszystkich genów przekształcane są za pomocą zależności:

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = \text{ROT}(\Delta\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$$

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$$

Wybór wartości i kierunków kąta obrotu $\Delta\theta_i$ dokonywany jest z wykorzystaniem tzw. **tablic Lookup Table**. **Tabela 3.2** określa wartości kątów obrotów $\Delta\theta_i$ w zależności od tego, jaka wartość binarna była zaobserwowana na danej pozycji i aktualnie przetwarzanego osobnika, jaka wartość binarna występowała na tej samej pozycji w najlepszym znalezionym dotychczas rozwiązaniu b oraz w zależności od tego, czy zaobserwowane rozwiązanie $x \in P$ ma wyższą wartość oceny niż najlepsze znalezione dotychczas rozwiązanie, tzn. $f(x) \geq f(b)$. W zależności od spełnienia tych warunków ustalana jest wartość kąta obrotu $\Delta\theta_i$ (czwarta kolumna tabeli). **Tabela 3.3** określa natomiast kierunek obrotu, którego wybór zależy od tych samych warunków, jak również od znaku iloczynu amplitud $\alpha_i\beta_i$. Ma to na celu wybór właściwego kierunku obrotu, niezależnie od tego, w której ćwiartce układu współrzędnych znajduje się stan aktualnie przetwarzanego genu kwantowego.

x_i	b_i	$f(x) \geq f(b)$	$\Delta\theta_i$
0	0	Nie	0
0	0	Tak	0
0	1	Nie	0
0	1	Tak	0.05π
1	0	Nie	0.01π
1	0	Tak	0.025π
1	1	Nie	0.005π
1	1	Tak	0.025π

TABELA 3.2: Tabela *Lookup Table* dla algorytmu QIGA1 (kąty obrotów)

x_i	b_i	$f(x) \geq f(b)$	$s(\alpha_i \beta_i)$			
			$\alpha_i \beta_i > 0$	$\alpha_i \beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	Nie	0	0	0	0
0	0	Tak	0	0	0	0
0	1	Nie	0	0	0	0
0	1	Tak	-1	+1	± 1	0
1	0	Nie	-1	+1	± 1	0
1	0	Tak	+1	-1	0	± 1
1	1	Nie	+1	-1	0	± 1
1	1	Tak	+1	-1	0	± 1

TABELA 3.3: Tabela *Lookup Table* dla algorytmu QIGA1 (kierunki obrotów)

3.4.2 Kwantowo inspirowany algorytm ewolucyjny iQIEA

Algorytm iQIEA został zaprezentowany po raz pierwszy w roku 2006 i jego szczegółowy opis można znaleźć m.in. w pracach (da Cruz *et al.*, 2004, 2006; Nowotniak, Kucharski, 2010b).

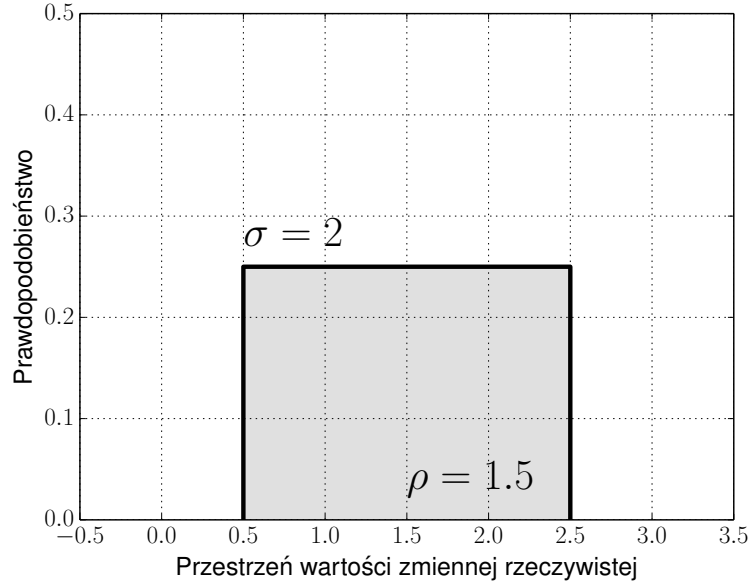
Niech $Q(t) = \{q_1, q_2, \dots, q_{|Q|}\}$ oznacza generację t populacji kwantowej, składającej się z $|Q|$ osobników. Każdy osobnik q_i tworzony jest przez N genów kwantowych $q_i = [g_{i1} \ g_{i2} \ \dots \ g_{iN}]$. W odróżnieniu od kwantowo inspirowanych algorytmów genetycznych QIGA, w kwantowo inspirowanym algorytmie ewolucyjnym iQIEA reprezentacja rozwiązań nie jest oparta na kubitach, lecz na ciągłych 1-wymiarowych rozkładach prawdopodobieństw (tzw. kodowanie impulsowe). W algorytmie iQIEA gen kwantowy modelowany jest jako ciągły, jednostajny rozkład prawdopodobieństwa na zbiorze $\mathbb{R}$. Taki gen kwantowy może być zatem zapisany za pomocą pary liczb rzeczywistych $g_{ij} = (\rho_{ij}, \sigma_{ij})$, gdzie ρ_{ij} oznacza środek j -tego przedziału, tzn. wartość oczekiwaną rozkładu (w pierwotnej pracy nazywanego *square pulse*, stąd propozycja nazywania tego rodzaju reprezentacji *kodowaniem impulsowym*), natomiast σ_{ij} to szerokość przedziału. Przykład genu kwantowego w algorytmie iQIEA został zaprezentowany na **Rysunku 3.4**.

Wysokość każdego impulsu określona jest zależnością:

$$h_{ij} = \frac{1/\sigma_{ij}}{|Q|} \quad (3.3)$$

co gwarantuje, że w dalszym etapie algorytmu suma pól impulsów w genach, występujących na tym samym locusie, będzie jednostkowa. W rezultacie suma może być traktowana bezpośrednio jako funkcja rozkładu prawdopodobieństwa.

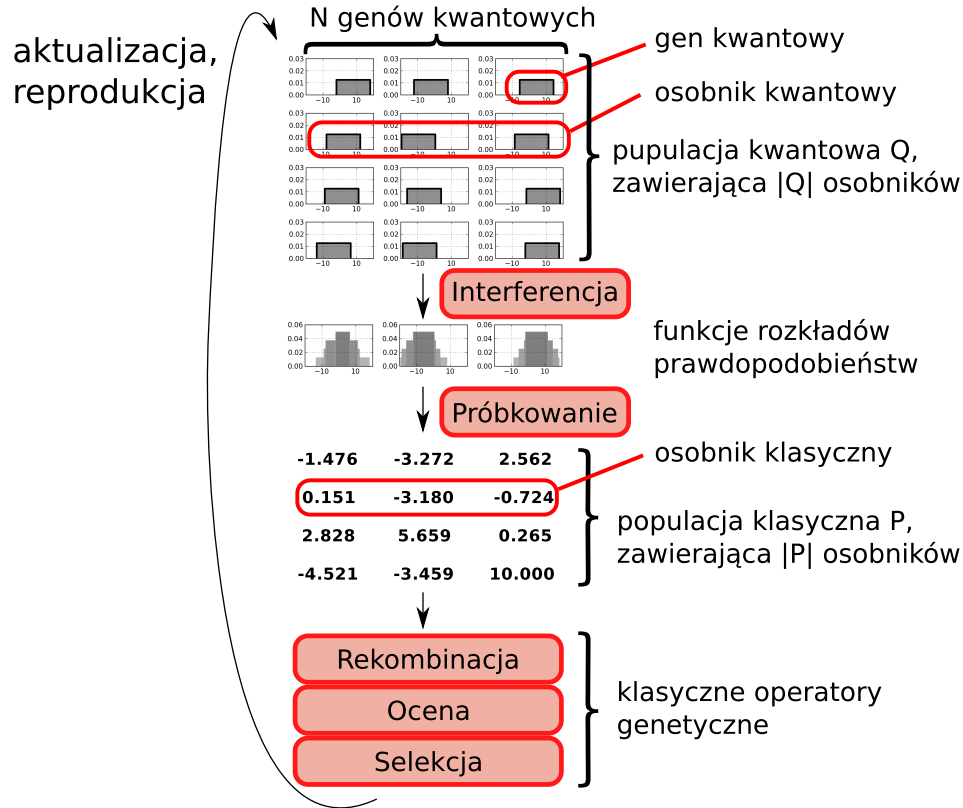
Ogólny schemat algorytmu iQIEA został przedstawiony na **Rysunku 3.5**. Populacja kwantowa $Q(t)$ reprezentuje rozkład prawdopodobieństwa próbkowania przestrzeni $\mathbb{R}$, odpowiadającej danej zmiennej rzeczywistej. N rozkładów prawdopodobieństw tworzonych jest poprzez sumowanie ze sobą impulsów, występujących na tych samych pozycjach w genotypach kwantowych. Etap sumowania rozkładów nazywany jest w algorytmie iQIEA **operatorem interferencji**. Algorytm próbkuje przestrzeń rozwiązań



RYS. 3.4: Przykład genu kwantowego $g_{ij} = (1.5, 2)$, kodującego liczbę rzeczywistą w algorytmie iQIEA

$X = \mathbb{R}^N$ zgodnie z tymi rozkładami. Następnie realizowane są typowe dla algorytmu ewolucyjnego operacje: rekombinacja, ocena oraz selekcja klasycznych osobników wytworzonych na etapie próbkowania. Ostatecznie tworzona jest kolejna generacja populacji $Q(t+1)$ poprzez taką modyfikację rozkładów w genach kwantowych Q , aby prawdopodobieństwa próbkowania otoczenia wysoko ocenianych osobników było zwiększane w kolejnych generacjach. Pełny pseudokod algorytmu iQIEA został zaprezentowany na [Listing 3](#).

Na początku algorytmu (linia 3), tworzona jest populacja początkowa $Q(0)$, składająca się z osobników kwantowych, w których środki ρ rozkładów prawdopodobieństw w poszczególnych genach ustalane są losowo, natomiast szerokości σ poszczególnych genów ustalane są na wartości równe rozmiarowi zakresu dopuszczalnego danej zmiennej. Na początku głównej pętli algorytmu (linia 6 i 7), następuje **obserwacja populacji** i wygenerowanie populacji klasycznej, składającej się z $|P| \geq |Q|$ klasycznych osobników (tzn. punktów w przestrzeni $\mathbb{R}^N$). Etap ten realizowany jest w dwóch krokach. Najpierw następuje interferencja, polegająca na su-

Rys. 3.5: Ogólny schemat algorytmu iQIEA (da Cruz *et al.*, 2006)

mowania impulsów w genach, występujących na tych samych pozycjach w poszczególnych osobnikach kwantowych. Następnie dla każdego z $|Q|$ wygenerowanych rozkładów prawdopodobieństw tworzone są dystrybuanty (CDF), co pozwala na wylosowanie N liczb rzeczywistych, zgodnie z wygenerowanymi rozkładami prawdopodobieństw:

$$x = \text{CDF}^{-1}(r) \quad (3.4)$$

gdzie r jest losową liczbą z przedziału $[0, 1]$, CDF^{-1} jest funkcją odwrotną dla danej dystrybuanty, natomiast x jest zaobserwowaną wartością rzeczywistą, którą przyjmie dany gen klasycznego osobnika (da Cruz *et al.*, 2004).

W pierwszej generacji algorytmu zaobserwowana populacja klasyczna $E(t)$ kopiowana jest bezpośrednio do $C(t)$ (krok 7). W kolejnych generacjach stosowane są dodatkowe operatory genetyczne. Wykorzystywane jest

Listing 3: Pierwotny algorytm iQIEA (da Cruz *et al.*, 2006)

```

1 begin
2    $t \leftarrow 0$ 
3    $Q(0) \leftarrow$  Utwórz  $|Q|$ -elementową populację początkową
4   while  $t \leq t_{\max}$  do
5      $t \leftarrow t + 1$ 
6     Utwórz dystrybuanty (CDF) dla populacji  $Q(t)$ 
7      $E(t) \leftarrow$  próbkuj  $Q(t)$ , korzystając z dystrybuant CDF
8     if  $t = 1$  then
9        $C(t) \leftarrow E(t)$ 
10    else
11       $E(t) \leftarrow$  krzyżowanie pomiędzy  $E(t)$  i  $C(t)$ 
12      Oceń osobniki w  $E(t)$ 
13       $C(t) \leftarrow |P|$  najlepszych osobników z  $E(t) \cup C(t)$ 
14    end
15    with  $|Q|$  najlepszych osobników z  $C(t)$  do
16       $Q(t+1) \leftarrow$  zastosuj operator translacji na  $Q(t)$ 
17       $Q(t+1) \leftarrow$  zastosuj operator skalowania na  $Q(t)$ 
18    endw
19  end
20 end

```

krzyżowanie jednorodne pomiędzy populacjami osobników klasycznych $E(t)$ oraz $C(t)$. Pseudokod tego operatora został przedstawiony na [Listingu 4](#).

Parametr $\xi \in [0, 1]$ (w linii 7 pseudokodu) to **współczynnik krzyżowania**. Wartość $\xi = 1$ powoduje przenoszenie wartości genów wszystkich osobników z zaobserwowanej populacji klasycznej $E(t)$ do potomstwa. Wartość współczynnika $\xi = 0$ powoduje, że osobniki z populacji $E(t)$ i $C(t)$ nie są krzyżowane ze sobą.

Ostatnim etapem głównej pętli algorytmu iQIEA jest aktualizacja populacji kwantowej $Q(t)$. Operacja ta składa się z dwóch etapów. Środki ρ_{ij} każdego genu kwantowego modyfikowane są w taki sposób, aby od-

Listing 4: Operator krzyżowania w iQIEA (da Cruz *et al.*, 2006)

```

1 begin
2   for i = 1 to K do
3     wybierz osobnika  $e_i$  z  $E(t)$ 
4     wybierz osobnika  $c_i$  z  $C(t)$ 
5     for j = 1 to N do
6        $r \leftarrow$  losowa liczba z przedziału  $[0,1)$ 
7       if  $r < \xi$  then
8          $e'_{ij} \leftarrow e_{ij}$ 
9       else
10         $e'_{ij} \leftarrow c_{ij}$ 
11      end
12    end
13  end
14 end

```

powiadały wartościom, zapisanym kolejno w $|Q|$ najlepszych osobników klasycznych z populacji $C(t)$. Następnie stosowana jest operacja skalowania impulsów według wzoru:

$$\sigma_{ij} = \begin{cases} \sigma_{ij} \cdot \delta & \text{jeżeli } \varphi < 1/5 \\ \sigma_{ij} / \delta & \text{jeżeli } \varphi > 1/5 \\ \sigma_{ij} & \text{jeżeli } \varphi = 1/5 \end{cases} \quad (3.5)$$

gdzie: φ to współczynnik określający, ile osobników z bieżącej populacji $C(t)$ poprawiło swoje dopasowanie w porównaniu z poprzednią generacją, natomiast $\delta \in (0, 1)$ to **współczynnik kontrakcji**. Wzór (3.5) oparty jest na prostej heurystyce znanej jako "zasada 1 / 5": Jeżeli więcej niż 20% osobników poprawia swoje dopasowanie, zakres przedziału próbkowania przestrzeni rozwiązań jest zwiększany. Jeżeli mniej niż 20% osobników poprawia swoje dopasowanie, zakres ulega zmniejszeniu. Mówiąc inaczej, jeśli niewiele osobników poprawia swoje dopasowanie w kolejnych generacjach, populacja zbiega się i dokonuje lokalnego przeszukiwania.

Gdy modyfikowane są szerokości impulsów w genach kwantowych, muszą również być aktualizowane wysokości impulsów zgodnie z rów-

naniem (3.3), tak aby ich suma tworzyła funkcję rozkładu prawdopodobieństwa. Operacja skalowania może być realizowana w każdej lub co kilka generacji, określonych poprzez dodatkowy parametr $k \in \mathbb{N}^+$ algorytmu. Na przykład, jeśli parametr $k = 5$, to szerokości impulsów w genach kwantowych są skalowane co pięć generacji.

Część II

ANALIZA TEORETYCZNA

Rozdział 4

WŁASNOŚCI PODSTAWOWE

4.1 ZBIEŻNOŚĆ ALGORYTMÓW QIEA

W rozdziale przeanalizowano zbieżność ogólnego kwantowo inspirowanego algorytmu ewolucyjnego z wykorzystaniem twierdzenia Banacha o punkcie stałym odwzorowania zwężającego (Nowotniak, Kucharski, 2012a). Zaprezentowana tutaj analiza stanowi adaptację rozumowania przedstawionego w (Michalewicz, 2003, str. 96) do algorytmu QIEA. Wszystkie niezbędne definicje wraz z twierdzeniem Banacha zostały zawarte w **Załączniku B** (str. 165).

4.1.1 Założenia twierdzenia Banacha dla algorytmów QIEA

Do przeprowadzenia analizy zbieżności algorytmu z wykorzystaniem twierdzenia Banacha (**Twierdzenie B.1**) niezbędne jest sprawdzenie wszystkich jego założeń:

1. Działanie algorytmu wymaga rozważenia w pewnej przestrzeni $\mathbb{S}$ populacji kwantowych, będącej przestrzenią metryczną (**Definicja B.2**), przy ustalonej reprezentacji rozwiązań.
2. Wybrana przestrzeń $\mathbb{S}$ musi być przestrzenią zupełną (**Definicja B.3**).
3. Pewna liczba $n \in \mathbb{N}^+$ iteracji algorytmu QIEA musi stanowić odwzorowanie $G : \mathbb{S} \mapsto \mathbb{S}$ w rozważanej przestrzeni populacji.
4. Wybrane odwzorowanie G musi być kontrakcją (**Definicja B.4**).

W kolejnych podpunktach przeanalizowano powyższe założenia. Rozważono przestrzeń metryczną oraz odwzorowanie, będące złożeniem pewnej liczby generacji algorytmu QIEA. Przyjęto, że analiza zbieżności będzie wykonana dla zadania optymalizacji unimodalnej funkcji celu $f : X \mapsto \mathbb{R}$ oraz — bez utraty ogólności — przyjęto, że będzie to zadanie maksymalizacji. Niech $x^* \in X$ oznacza optymalne rozwiązanie w przestrzeni rozwiązań:

$$x^* = \arg \max_{x \in X} f(x) \quad (4.1)$$

4.1.2 Przestrzeń metryczna $\langle \mathbb{S}, \delta \rangle$ populacji kwantowych Q

Niech $\mathbb{S}$ oznacza zbiór wszystkich możliwych $|Q|$ -elementowych populacji kwantowych Q , przy ustalonej reprezentacji rozwiązań. Do utworzenia przestrzeni metrycznej niezbędne jest wybranie miary odległości $\delta : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}$ między populacjami z $\mathbb{S}$.

Aby możliwe było dowiedzenie zbieżności algorytmu, metryka δ musi opierać się na mierze jakości populacji Q , bowiem ze wzrostu tej miary będzie wynikać zmniejszanie odległości między elementami przestrzeni $\langle \mathbb{S}, \delta \rangle$ (warunek zwężania). Kluczowe jest zatem wybranie odpowiedniej funkcji miary $\text{eval} : Q \mapsto \mathbb{R}$ oceny jakości genotypów kwantowych $q \in Q$ oraz miary $F : \mathbb{S} \mapsto \mathbb{R}$ oceny całych populacji kwantowych Q . Należy zauważyć, że miara F musi być miarą deterministyczną, aby (wykorzystując ją) odległość δ mogła spełniać warunki metryki (Definicja B.1).

Przyjmijmy następującą miarę jakości genotypów oraz populacji kwantowych. Załóżmy, że $\text{eval} : Q \mapsto [0, 1]$ przyporządkowuje genotypom kwantowym $q \in Q$ prawdopodobieństwo wylosowania (spróbkowania), z rozkładem prawdopodobieństwa $\text{Pr}_q : \mathcal{F}_X \mapsto [0, 1]$ (zapisanym w q), rozwiązania x^* , dla którego wartość funkcji celu f jest optymalna:

$$\text{eval}(q) = \text{Pr}_q \left(\left\{ \arg \max_{x \in X} f(x) \right\} \right) \quad (4.2)$$

Należy zauważyć, że zaproponowana miara osiąga wartość maksymalną tylko dla genotypu kwantowego q , pozbawionego niepewności, tj. wybierającego jednoznacznie element x^* z przestrzeni rozwiązań X .

Niech funkcja F oznacza średnią wartość eval dla całej populacji kwantowej, tzn.:

$$F(Q) = \frac{1}{|Q|} \sum_{q \in Q} \text{eval}(q) \quad (4.3)$$

Wykorzystując miarę oceny populacji kwantowych F , niech odległość pomiędzy populacjami kwantowymi $Q_1, Q_2 \in \mathbb{S}$ będzie zdefiniowana analo-

gicznie jak w (Michalewicz, 2003):

$$\delta(Q_1, Q_2) = \begin{cases} 0 & \text{jeżeli } Q_1 = Q_2 \\ |2 - F(Q_1)| + |2 - F(Q_2)| & \text{jeżeli } Q_1 \neq Q_2 \end{cases} \quad (4.4)$$

Miara odległości $\delta(Q_1, Q_2)$ spełnia warunki metryki, ponieważ z definicji $\delta(Q_1, Q_2)$:

- $\delta(Q_1, Q_2) \geq 0$ dla dowolnych $Q_1, Q_2 \in \mathbb{S}$ oraz
 $\delta(Q_1, Q_2) = 0 \iff Q_1 = Q_2$
- $\delta(Q_1, Q_2) = \delta(Q_2, Q_1)$
- $\delta(Q_1, Q_2) + \delta(Q_2, Q_3) = |2 - F(Q_1)| + |2 - F(Q_2)| + |2 - F(Q_2)| + |2 - F(Q_3)| \geq |2 - F(Q_1)| + |2 - F(Q_3)| = \delta(Q_1, Q_3)$

Para $\langle \mathbb{S}, \delta \rangle$ tworzy zatem przestrzeń metryczną.

4.1.3 Zupełność przestrzeni $\langle \mathbb{S}, \delta \rangle$

Wybrana przestrzeń metryczna populacji musi być przestrzenią metryczną zupełną, tzn. każdy zbieżny ciąg elementów przestrzeni $\langle \mathbb{S}, \delta \rangle$ musi posiadać granicę należącą do tej przestrzeni. W komputerowej implementacji algorytmu warunek ten jest zawsze spełniony, ponieważ populacje zawsze należą do pewnej dyskretnej, przeliczalnej i skończonej przestrzeni.

4.1.4 Odwzorowanie w przestrzeni $\langle \mathbb{S}, \delta \rangle$

Przyjmijmy, że jako przekształcenie $G : S \mapsto S$ populacji $Q(t)$ będzie traktowany taki ciąg $n \in \mathbb{N}^+$ iteracji algorytmu, po którym nastąpiła poprawa oceny populacji $Q(t+n)$ w stosunku do $Q(t)$, tzn. kiedy $F(Q(t+n)) > F(Q(t))$. Zbieżność może być zatem dowiedziona tą metodą tylko dla takich algorytmów, dla których można wykazać, że taka poprawa zachodzi – tzn. że zachodzi wzrost prawdopodobieństwa próbkowania optymalnego rozwiązania.

W odróżnieniu od rozumowania przedstawionego w (Michalewicz, 2003), dotyczącego zbieżności klasycznego „algorytmu genetycznego z odwzorowaniem zwiężającym”, numer iteracji t nie jest inkrementowany

jedynie po wystąpieniu poprawy jakości populacji $Q(t)$, ale jako odwzorowanie G traktowane jest złożenie odpowiedniej liczby iteracji. Pozwala to uniknąć wprowadzania modyfikacji do schematu algorytmu QIEA do przeprowadzenia rozumowania.

4.1.5 Generacja populacji kwantowej Q jako kontrakcja

Można wykazać, że jeżeli odwzorowanie G poprawia ocenę populacji, to G jest odwzorowaniem zwężającym, tzn:

$$\exists \epsilon \in (0,1) \forall x,y \in X \delta(G(x), G(y)) \leq \epsilon \cdot \delta(x, y) \quad (4.5)$$

Zgodnie z definicją odwzorowania G , wprowadzonego w punkcie 4.1.4:

$$F(Q1(t)) < F(G(Q1(t))) = F(Q1(t+n))$$

$$F(Q2(t)) < F(G(Q2(t))) = F(Q2(t+n))$$

zatem:

$$\delta(G(Q1(t)), G(Q2(t))) = \delta(Q1(t+n), Q2(t+n)) \quad (4.6)$$

$$= |2 - F(Q1(t+n))| + |2 - F(Q2(t+n))|$$

$$< |2 - F(Q1(t))| + |2 - F(Q2(t))| = \delta(Q1(t), Q2(t))$$

W przypadku komputerowej implementacji algorytmu można podać wielkość stałą $\epsilon \in [0, 1]$, przez którą może zostać pomnożona prawa strona powyższej nierówności, zachowując jej prawdziwość. Wielkość tej stałej związana jest z najmniejszą liczbą rzeczywistą w słowie komputerowym i określa minimalną wielkość poprawy oceny F . Odwzorowania G zmniejsza odległość δ pomiędzy dowolnymi populacjami kwantowymi $Q_1, Q_2 \in \mathbb{S}$, zatem G jest odwzorowaniem zwężającym.

4.1.6 Ogólne warunki zbieżności algorytmu QIEA

Przeprowadzona analiza założeń twierdzenia Banacha w odniesieniu do ogólnego kwantowo inspirowanego algorytmu ewolucyjnego pozwala określić, dla jakich warunków taki algorytm jest zbieżny do jednoznacznie wybranego punktu stałego x^* . W celu przeprowadzenia analizy zbieżności

wprowadzona została odpowiednia miara oceny genotypów kwantowych $eval$ (4.2), uwzględniającą element losowości występujących w reprezentacji rozwiązań w algorytmach QIEA. Następnie w odpowiedni sposób zaadaptowane zostały pozostałe założenia twierdzenia Banacha (B.1) w stosunku do rozumowania z (Michalewicz, 2003).

Jak wynika z wprowadzonej funkcji oceny $eval$, aby algorytm był zbieżny niezależnie od stanu początkowej populacji kwantowej $Q(0)$, musi zachodzić wzrost miary oceny F populacji kwantowych Q w kolejnych generacjach algorytmu. Oznacza to, że algorytm jest zbieżny, jeśli nigdy nie traci możliwości spróbkowania optymalnego rozwiązania $x^* \in X$ i gdy działanie operatorów genetycznych powoduje, że przy t dążącym do nieskończoności rośnie prawdopodobieństwo próbkowania najlepszego znajdującego dotąd rozwiązania. Powyższy wniosek wynika bezpośrednio z przyjętej wcześniej definicji odwzorowania G , która zapewnia, że po odpowiednio wielu generacjach algorytmu nastąpi poprawa średniej oceny populacji kwantowej Q . Spełnienie warunków z powyższego wniosku gwarantuje wzrost miary F , odpowiadającej średniemu prawdopodobieństwu spróbkowania optymalnego rozwiązania w kolejnych generacjach.

Przedstawione w tym podrozdziale rozumowanie prowadzi do bardzo intuicyjnego wniosku: Jeżeli wraz z liczbą generacji t , dążącą do nieskończoności, zachodzi ciągły wzrost średniej oceny całej populacji, a optymalizowana funkcja posiada optimum, to populacja na pewno będzie zbieżna do tego optimum. Przedstawione tu rozumowanie jest jedynie matematycznym zapisem i formalnym dowodem tego intuicyjnego faktu.

Jak zauważono w (De Jong, 2006, str. 206), stwierdzenie faktu zbieżności algorytmów ewolucyjnych np. poprzez tego rodzaju adaptację rozumowania z (Michalewicz, 2003) nie jest zadaniem trudnym. Wynika to z występujących w algorytmach ewolucyjnych elementów losowości i możliwości przyjęcia założenia o dążącej do nieskończoności liczbie generacji. Zadaniem dużo trudniejszym jest natomiast określenie szybkości zbieżności w ogólnym przypadku i w obszarze algorytmów ewolucyjnych wciąż uznawane jest to za problem nietrywialny. Dla niektórych bardzo uproszczonych kwantowo inspirowanych algorytmów ewolucyjnych zosta-

ły określone nie tylko warunki ich zbieżności, ale została obliczona również szybkość zbieżności (wyrażona jako funkcja wartości parametrów algorytmów), poprzez wykonanie analizy wykorzystującej teorię łańcuchów Markova ([Khosraviani et al., 2008](#); [Yi et al., 2011](#); [Chen, Ding, 2011](#)).

4.2 SCHEMATY W ALGORYTMACH QIGA

Jedną z pierwszych prób wyjaśnienia skuteczności algorytmów genetycznych było znane w tym obszarze twierdzenie Hollanda o schematach (Holland, 1975), zgodnie z którym liczba krótkich schematów, niskiego rzędu, o ponadprzeciętnym przystosowaniu rośnie w tempie wykładniczym w kolejnych generacjach algorytmu genetycznego. Podstawowe pojęcia, dotyczące teorii schematów, podane zostały w Rozdziale 1.4 na str. 25. Niniejszy podrozdział stanowi analizę klasy kwantowo inspirowanych algorytmów genetycznych w tym kontekście. Analiza ma na celu uogólnienie pojęcia dopasowania chromosomu kwantowego do schematu, określenie prawdopodobieństwa dopasowania chromosomu do schematu oraz określenie podstawowych miar zmiennej losowej, odpowiadającej liczbie chromosomów kwantowych, pasujących do danego schematu w populacji kwantowej.

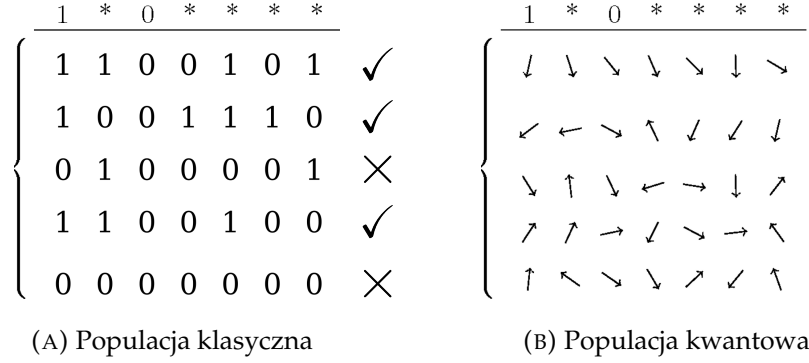
Zaprezentowana analiza została po raz pierwszy opublikowana w pracy (Nowotniak, Kucharski, 2010a), do której odnoszono się później m.in. w (Manju, Nigam, 2012).

4.2.1 Schematy w binarnym chromosomie kwantowym

Analizę rozpoczęto od rozważenia pojęcia schematu w kontekście binarnego chromosomu kwantowego, co zostało zilustrowane dla przykładowej populacji klasycznej i kwantowej oraz dla wybranego schematu na **Rysunku 4.1**. W przypadku populacji klasycznej chromosomy pasują lub nie pasują do schematu w zależności od tego, czy kolejne elementy chromosomu odpowiadają kolejnym symbolom w schemacie za wyjątkiem tych pozycji, na których występują symbole nieoznaczone *. W przypadku populacji kwantowej dopasowanie do schematu uzależnione jest natomiast od wartości binarnych, obserwowanych w kolejnych genach kwantowych w danym pomiarze.

Niech H oznacza schemat o długości N , tzn. N -elementowy napis nad alfabetem ternarnym $\{0, 1, *\}$, gdzie $*$ oznacza dowolny element ze zbioru alleli $A = \{0, 1\}$. Niech $\text{Pr}_g : \mathcal{F}_A \mapsto [0, 1]$ oznacza rozkład prawdopodobieństwa, kodowany przez binarny gen kwantowy $|g\rangle = \alpha|0\rangle + \beta|1\rangle$. Podczas kształ-

owania się fenotypu zachodzi proces obserwacji stanu genu $|g\rangle$ i następuje ekspresja wartości 0 z prawdopodobieństwem $|\alpha|^2$ oraz wartości 1 z prawdopodobieństwem $|\beta|^2$, tzn. $\Pr_g(\{0\}) = |\alpha|^2$, $\Pr_g(\{1\}) = |\beta|^2$. Ponadto niech $\ast = \{0, 1\}$, wówczas $\Pr_g(\ast) = \Pr_g(\{0, 1\}) = 1$. Niech $|q\rangle = |g_1\rangle|g_2\rangle \dots |g_N\rangle$ oznacza binarny chromosom kwantowy o długości N .



RYS. 4.1: Ilustracja dopasowania do wybranego schematu chromosomów w populacji klasycznej (A) oraz kwantowej (B)

4.2.2 Warunkowe dopasowanie do schematu

Rozważmy na początku prostą sytuację, kiedy binarny chromosom kwantowy $|q\rangle = |g_1\rangle|g_2\rangle|g_3\rangle$ pasuje do schematu $H = 10\ast$. Takie zdarzenie zachodzi, gdy odpowiednio: wartość 1 zostaje zaobserwowana w genie g_1 , wartość 0 zostanie zaobserwowana w genie g_2 , a w genie g_3 może zostać zaobserwowana dowolna wartość. Ze względu na element losowości w reprezentacji rozwiązań w algorytmach QIGA, binarny chromosom kwantowy pasuje do schematu z określonym prawdopodobieństwem. Ponieważ geny kwantowe w algorytmach QIGA są niezależne względem siebie (tzn. są modelowane poprzez niezależne dwupoziomowe układy kwantowe), prawdopodobieństwo dopasowania binarnego chromosomu kwantowego $|q\rangle$ do schematu H równe jest iloczynowi prawdopodobieństw obserwacji w kolejnych genach kolejnych elementów schematu H .

Zatem prawdopodobieństwo $M(q, H)$ dopasowania binarnego chromosomu kwantowego q do schematu H wyraża się jako:

$$M(q, H) = \prod_{i=1}^N \Pr_{g_i}(\{H[i]\}) = \prod_{i=1}^N \begin{cases} |\alpha_i|^2 & \text{gdy } H[i] = 0 \\ |\beta_i|^2 & \text{gdy } H[i] = 1 \\ 1 & \text{gdy } H[i] = * \end{cases} \quad (4.7)$$

gdzie:

- q – binarny chromosom kwantowy o długości N
- H – schemat o długości N
- $H[i]$ – element w schemacie H na pozycji i

q	1	0	$\frac{1}{2} 0\rangle + \frac{\sqrt{3}}{2} 1\rangle$	$M(q, H)$
H_1	*	*	*	1
H_2	*	0	*	1
H_3	1	0	*	1
H_4	*	*	0	$\frac{1}{4}$
H_5	1	*	1	$\frac{3}{4}$

TABELA 4.1: Przykład dla binarnego chromosomu kwantowego $|q\rangle = |g_1\rangle|g_2\rangle|g_3\rangle = 10(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle)$ oraz pięciu różnych schematów. Prawdopodobieństwo $M(q, H_1)$ dopasowania $|q\rangle$ do $H_1 = ***$ wynosi 1, prawdopodobieństwo $M(q, H_5)$ dopasowania $|q\rangle$ do $H_5 = 1*1$ wynosi $\frac{3}{4}$ etc.

Dla przykładu rozważmy binarny chromosom kwantowy $|q\rangle = |g_1\rangle|g_2\rangle|g_3\rangle = 10(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle)$ i jego dopasowanie do pięciu schematów: $***, *0*, 10*, **0$ i $1*1$. Te schematy i odpowiadające im prawdopodobieństwa dopasowania zostały przedstawione w Tabeli 4.1. Chromosom $|q\rangle$ pasuje do schematu $H_1 = ***$ niezależnie od wartości zaobserwowanych w genach kwantowych $|g_1\rangle, |g_2\rangle, |g_3\rangle$, zatem prawdopodobieństwo $M(q, H_1)$ dopasowania wynosi 1, zgodnie ze wzorem (4.7). Podobnie, niezależnie od wartości zaobserwowanej w genie $|g_3\rangle$, prawdopodobieństwo dopasowania $|q\rangle = 10(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle)$ do schematów $*0*$ i $10*$ wynosi $M(q, H_2) = 1$ i $M(q, H_3) = 1$. Chromosom $|q\rangle$ pasuje do schematu $**0$,

wtedy i tylko wtedy gdy w binarnym genie kwantowym g_3 zostanie zaobserwowana wartość 0. Prawdopodobieństwo takiego zdarzenia to $(\frac{1}{2})^2$, zatem $M(q, H_4) = \frac{1}{4}$. Chromosom kwantowy $|q\rangle$ pasuje do schematu $1*1$, jeśli wartość 1 jest zaobserwowana w genach $|g_1\rangle$ i $|g_3\rangle$. Następuje to z prawdopodobieństwem $M(q, H_5) = 1 \cdot |\frac{\sqrt{3}}{2}|^2 = \frac{3}{4}$.

4.2.3 Wartość oczekiwana liczby dopasowań

Niech zmienna losowa L określa liczbę chromosomów kwantowych, pasujących do schematu H w populacji $Q = \{q_1, q_2, \dots, q_{|Q|}\}$. Dodatkowo, dla uproszczenia notacji, w dalszej części rozumowania oznaczmy prawdopodobieństwo zdarzenia, że chromosom kwantowy q_i pasuje lub nie pasuje do schematu H odpowiednio jako $D_i^+ = M(q_i, H)$ oraz $D_i^- = (1 - M(q_i, H))$.

Wartość oczekiwana liczby chromosomów w populacji Q , pasujących do schematu H (tzn. wartość oczekiwaną zmiennej losowej L), wyraża się jako:

$$\begin{aligned}
 E(L) = & 0 \cdot [D_1^- D_2^- \dots D_{|Q|}^-] \\
 & + 1 \cdot \underbrace{[D_1^+ D_2^- \dots D_{|Q|}^- + D_1^- D_2^+ \dots D_{|Q|}^- + \dots + D_1^- D_2^- \dots D_{|Q|}^+]}_{\binom{|Q|}{1} \text{ elementów}} \\
 & + 2 \cdot \underbrace{[D_1^+ D_2^+ \dots D_{|Q|}^- + D_1^+ D_2^- \dots D_{|Q|}^- + \dots + D_1^- D_2^- \dots D_{|Q|}^+]}_{\binom{|Q|}{2} \text{ elementów}} \\
 & + \dots \\
 & + |Q| \cdot [D_1^+ D_2^+ \dots D_{|Q|}^+] \tag{4.8}
 \end{aligned}$$

Kolejne wiersze w równaniu (4.8) odpowiadają iloczynom liczb możliwych wystąpień chromosomów pasujących do schematu H oraz prawdopodobieństw wystąpienia takiej liczby pasujących chromosomów. $D_1^- D_2^- \dots D_{|Q|}^-$ jest prawdopodobieństwem tego, że żaden element Q nie pasuje do H . Wyrażenie z drugiego wiersza we wzorze (4.8) jest prawdopodobieństwem zdarzenia, że dokładnie jeden (którykolwiek spośród $q_1, \dots, q_{|Q|}$) element w populacji Q pasuje do H . Ponieważ dopasowanie $w \in \{0, 1, \dots, |Q|\}$ chromosomów z $|Q|$ -elementowej populacji Q odpowiada wyborowi w

elementowych podzbiorów zbioru $|Q|$ -elementowego, w kolejnych wierszach we wzorze (4.8) następuje sumowanie $\binom{|Q|}{w}$ elementów.

To pozwala przepisać równanie (4.8) bardziej zwięźle, jednocześnie w sposób pozwalający na łatwą implementację w języku programowania¹:

$$E(L) = \sum_{w=0}^{|Q|} w \cdot \sum_{C \in \{X \in 2^{\{1, \dots, |Q|\}} : |X|=w\}} \prod_{j \in C} D_j^+ \prod_{k \in \{1, \dots, |Q|\} \setminus C} D_k^- \quad (4.9)$$

Równanie (4.9) pokazuje, w jaki sposób można łatwo obliczyć wartość oczekiwaną $E(L)$ w komputerowej implementacji w czterech pętlach, niezależnie od rozmiaru populacji oraz długości chromosomów (Załącznik C). Pierwszy znak sumy odpowiada sumowaniu po możliwych liczbach dopasowań. Drugi znak sumy odpowiada sumowaniu po w -elementowych podzbiorach zbioru $\{1, 2, \dots, |Q|\}$. Iloczyn we wzorze odpowiadają natomiast prawdopodobieństwom liczb dopasowań wybranych binarnych chromosomów kwantowych do schematu H .

Podstawiając $M(q_j, H)$ i $(1 - M(q_j, H))$ odpowiednio za D_j^+ i D_j^- , otrzymuje się postać:

$$E(L) = \sum_{w=0}^{|Q|} w \cdot \sum_{C \in \{X \in 2^{\{1, \dots, |Q|\}} : |X|=w\}} \prod_{j \in C} M(q_j, H) \prod_{k \in \{1, \dots, |Q|\} \setminus C} (1 - M(q_j, H)) \quad (4.10)$$

Wzór (4.10) wyraża wartość oczekiwaną liczby binarnych chromosomów kwantowych, pasujących do schematu H , w $|Q|$ -elementowej populacji kwantowej Q .

4.2.4 Wariancja dopasowań schematu

W celu obliczenia wariancji liczby dopasowań, wykorzystane mogą być następujące własności zmiennej losowej:

$$\begin{aligned} V(L) &= E(L^2) - (E(L))^2 \\ E(L^2) &= \sum_k x_k^2 p(x_k) \end{aligned} \quad (4.11)$$

¹ Należy pamiętać, że iloczyn elementów zbioru pustego wynosi 1, tzn. $\prod \emptyset = 1$, natomiast symbol $2^{\{1, \dots, |Q|\}}$ oznacza w matematyce zbiór wszystkich podzbiorów zbioru $\{1, 2, \dots, |Q|\}$.

Wartość oczekiwana zmiennej losowej L^2 wyraża się jako:

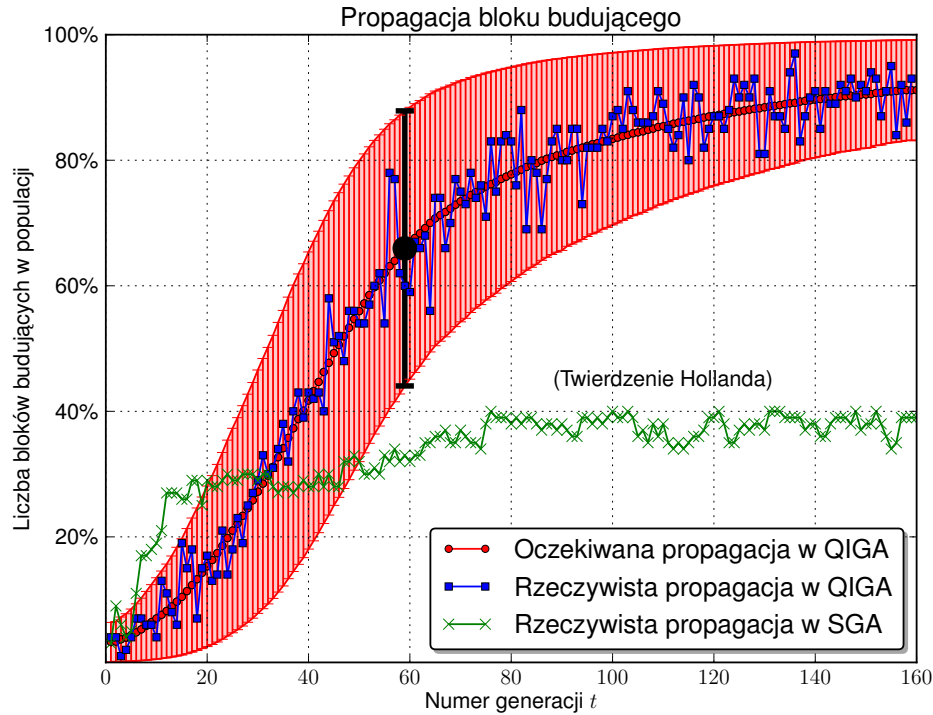
$$\begin{aligned}
 E(L^2) &= 0^2 \cdot [D_1^- D_2^- \dots D_{|Q|}^-] \\
 &+ 1^2 \cdot [D_1^+ D_2^- \dots D_{|Q|}^- + D_1^- D_2^+ \dots D_{|Q|}^- + \dots + D_1^- D_2^- \dots D_{|Q|}^+] \\
 &+ 2^2 \cdot [D_1^+ D_2^+ \dots D_{|Q|}^- + D_1^+ D_2^- \dots D_{|Q|}^- + \dots + D_1^- D_2^- \dots D_{|Q|}^+] \\
 &+ \dots \\
 &+ |Q|^2 \cdot [D_1^+ D_2^+ \dots D_{|Q|}^+] \\
 &= \sum_{w=0}^{|Q|} w^2 \cdot \sum_{C \in \{X \in 2^{\{1, \dots, |Q|\}} : |X|=w\}} \prod_{j \in C} D_j^+ \prod_{k \in \{1, \dots, |Q|\} \setminus C} D_k^-
 \end{aligned}$$

Zatem wariancja liczby binarnych chromosomów kwantowych, pasujących do schematu H w $|Q|$ -elementowej populacji Q , wyraża się jako:

$$\begin{aligned}
 V(L) &= \sum_{w=0}^{|Q|} w^2 \cdot \sum_{C \in \{X \in 2^{\{1, \dots, |Q|\}} : |X|=w\}} \prod_{j \in C} M(q_j, H) \prod_{k \in \{1, \dots, |Q|\} \setminus C} (1 - M(q_k, H)) \\
 &- \left(\sum_{w=0}^{|Q|} w \cdot \sum_{C \in \{X \in 2^{\{1, \dots, |Q|\}} : |X|=w\}} \prod_{j \in C} M(q_j, H) \prod_{k \in \{1, \dots, |Q|\} \setminus C} (1 - M(q_k, H)) \right)^2
 \end{aligned} \tag{4.12}$$

Wyprowadzone wzory (4.10) i (4.12) pozwalają określić wartość oczekiwaną oraz wariancję liczby reprezentantów danego schematu w aktualnej generacji t w algorytmie QIGA. Wzory te zostały zaimplementowane przez autora w języku programowania Python, a ilustracja wprowadzonej zmiennej losowej dla przykładowego uruchomienia algorytmu QIGA i wybranego bloku budującego przedstawiona została na [Rysunku 4.2](#). Wybór bloku budującego został wykonany za pomocą tej samej metody, jaka została zaproponowana w pracy ([Kusztelak, Rudnicki, Wiak, 2004](#)). Linia zieloną zaznaczono procentowy udział reprezentantów danego bloku budującego w populacji w algorytmie SGA (wartość ta podlega twierdzeniu Hollanda). Kolorem czerwonym zaznaczono wartość oczekiwaną i wariancję liczby reprezentantów bloku budującego w algorytmie QIGA w danej generacji i te wielkości zostały obliczone za pomocą wyprowadzonych wzorów. Przebieg rzeczywistej propagacji (linia niebieska

na wykresie), sytuujący się wewnątrz zakresu wyznaczonego przez wartość oczekiwaną i wariancję, potwierdza prawidłowość wyprowadzeń końcowych postaci wzorów (4.10) i (4.12).



RYS. 4.2: Porównanie propagacji wybranego schematu w algorytmach SGA i QIGA dla testowego uruchomienia algorytmu. Wyprowadzone wielkości $E(L)$ oraz $V(L)$ pozwalają określić wartość oczekiwaną oraz wariancję liczby reprezentantów danego schematu w aktualnej generacji t .

4.3 PODSUMOWANIE I WNIOSKI

W niniejszym rozdziale poddano analizie pojęcie schematu w kontekście kwantowo inspirowanego algorytmu genetycznego oraz chromosomów kwantowych. Uogólnione zostało pojęcie dopasowania binarnego chromosomu kwantowego do schematu, określone zostało prawdopodobieństwo $M(q, H)$ (4.7) dopasowania chromosomu do schematu, zostały także wyprowadzone wzory (4.10) i (4.12), pozwalające na obliczenie podsta-

wowych miar zmiennej losowej (wartości oczekiwanej i wariancji), odpowiadającej liczbie chromosomów w populacji kwantowej, pasujących do danego schematu.

Przeprowadzona analiza stanowi jedną z nielicznych do tej pory prób teoretycznego zbadania własności kwantowo inspirowanych algorytmów genetycznych i została po raz pierwszy zaprezentowana w artykule (Nowotniak, Kucharski, 2010a). Zostało to podkreślone m.in. w we wnioskach końcowych artykułu (Manju, Nigam, 2012), w którym dokonano przekrojowego przeglądu 748 publikacji naukowych, sytuujących się w obszarze *quantum-inspired computational intelligence*.

Ograniczenia wykonanej analizy mają charakter podobny do ograniczeń teorii Hollanda dla prostego algorytmu genetycznego. Analiza działania algorytmu za pomocą pojęcia schematów oraz związanej z nimi propagacji bloków budujących nie pozwala wyciągać wniosków na temat działania algorytmu w przypadku trudnych zadań optymalizacji kombinatorycznej czyli zadań zwodniczych, w których optymalne rozwiązania nie są tworzone z bloków budujących. Jednakże dzięki przeprowadzeniu tej analizy możliwe było zidentyfikowanie i zrozumienie nowych związków pomiędzy schematami i binarnymi chromosomami kwantowymi, omówionych szczegółowo w kolejnym rozdziale.

Rozdział 5

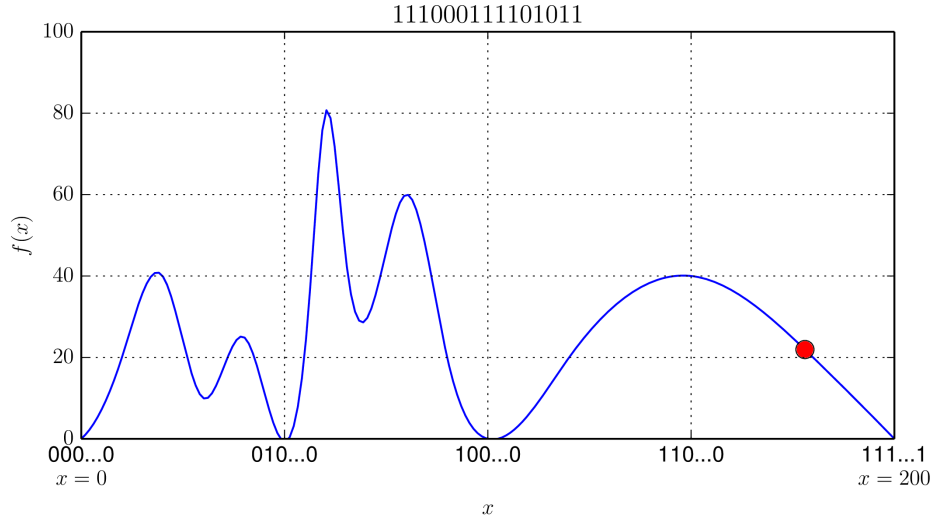
ALGORYTMY KWANTOWO INSPIROWANE WYŻSZYCH RZĘDÓW

W rozdziale wprowadzone zostały podstawy kwantowo inspirowanych algorytmów genetycznych wyższych rzędów, zaprezentowane w pracy (Nowotniak, Kucharski, 2014). Sformułowano pojęcia rzędu i współczynników kwantowości algorytmu, omówiono wybrane przestrzenie działania algorytmów oraz przeanalizowano reprezentację rozwiązań i podstawowe własności algorytmów wyższych rzędów.

5.1 PRZESTRZENIE DZIAŁANIA ALGORYTMÓW

Analizę własności algorytmów, prowadzącą do zdefiniowania pojęć kwantowo inspirowanych algorytmów wyższych rzędów, rozpoczęto od zbadania własności wybranych przestrzeni, w których działają algorytmy przeszukiwania (przeszukiwanie binarne, klasyczny algorytm SGA, pierwotny algorytm QIGA oraz teoretyczny algorytm w prawdziwym komputerze kwantowym).

Podobnie jak zostało to przedstawione w pracy (Goldberg, 1989, str. 61), sposoby działania algorytmów oraz przestrzenie ich działania mogą być analizowane na tle prostej jednowymiarowej funkcji celu $f : X \mapsto \mathbb{R}$. Wybrana w tym celu funkcja $f(x)$ została zaprezentowana na **Rysunku 5.1**. Dziedzina funkcji jest przedział liczbowy $X = [x_{\min}, x_{\max}] = [0, 200]$ (*przestrzeń fenotypów* tworzą więc bezpośrednio liczby rzeczywiste z tego przedziału), a sama funkcja została utworzona poprzez interpolację krzywymi sklejonymi zbioru węzłów, wybranymi dla nadania funkcji odpowiedniego kształtu. Przy wykorzystaniu bardzo typowego dla algorytmów genetycznych kodowania binarnego, przestrzeń fenotypów $X = [0, 200]$ może być odwzorowana na przestrzeń ciągów binarnych za pomocą prostej bijekcji.



RYS. 5.1: Przestrzeń ciągów binarnych odwzorowana na dziedzinę wybranej funkcji celu $f(x) : [0, 200] \mapsto \mathbb{R}$

W takim odwzorowaniu początkowi przedziału ($x = 0$) odpowiada ciąg binarny $00 \dots 0$, zaś końcowi przedziału ($x = 200$) odpowiada ciąg $11 \dots 1$. Długość ciągów N wynika natomiast z oczekiwanej precyzji obliczeń ϵ i wyraża się zależnością (Kusiak *et al.*, 2009, str. 193):

$$N = \left\lceil \log_2 \left(\frac{x_{\max} - x_{\min}}{\epsilon} + 1 \right) \right\rceil \quad (5.1)$$

Dla precyzji obliczeń $\epsilon = 10^{-6}$, wymagana długość ciągów binarnych wynosi $N = \left\lceil \log_2 \left(\frac{200}{10^{-6}} + 1 \right) \right\rceil = 28$. W przypadku prostego algorytmu genetycznego SGA, zbiór ciągów binarnych stanowi jednocześnie *przestrzeń genotypów*. Oprócz przestrzeni rozwiązań X , można jednak wyróżnić dodatkowo abstrakcyjną **przestrzeń działania algorytmu**, np. prosty algorytm genetyczny SGA, pomimo że bezpośrednio przetwarza ciągi binarne, to w sposób niejawnny (tzw. ukryta równoległość) działa w przestrzeni schematów, z czego wynika jego efektywność (Holland, 1975).

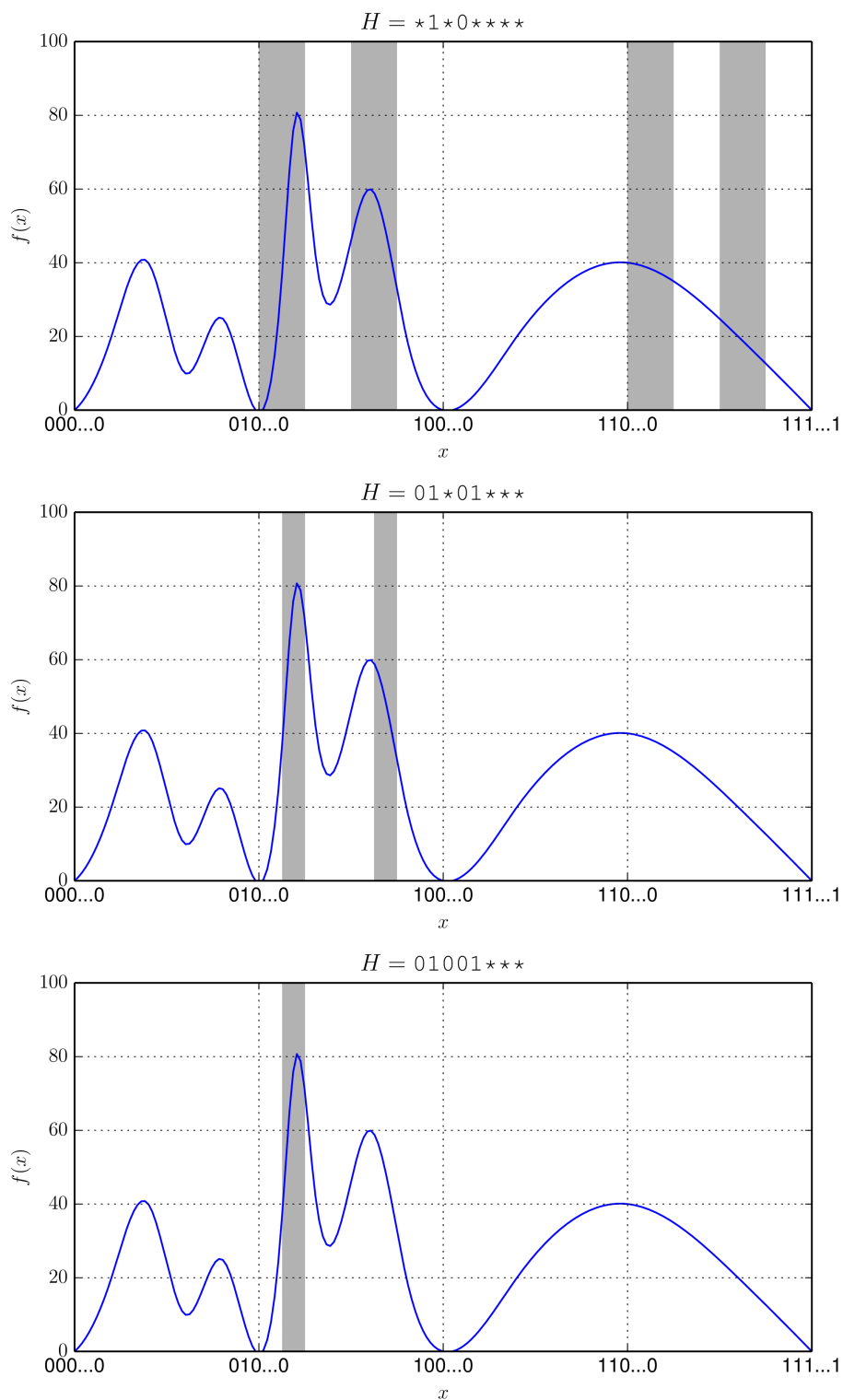
W kolejnych podpunktach przedstawiono i porównano przestrzenie działania wybranych algorytmów. Zilustrowano również na tle funkcji $f(x)$, w jaki sposób przestrzeń działania danego algorytmu odwzorowywana jest na przestrzeń rozwiązań X .

5.1.1 Przestrzeń ciągów binarnych

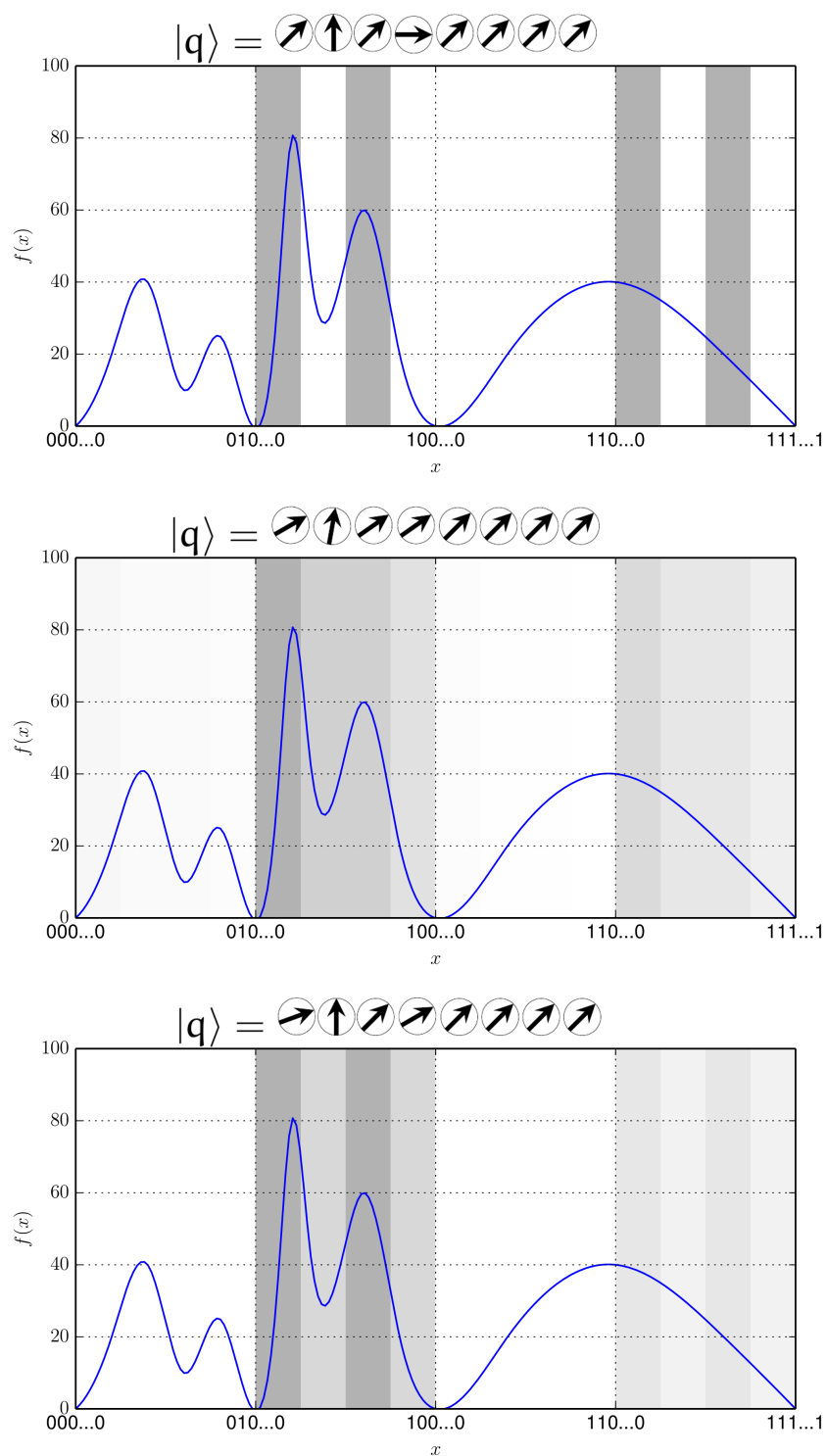
Dyskretna, skończona przestrzeń ciągów binarnych $\Omega_0 = \{ 00 \dots 0, 00 \dots 1, \dots, 11 \dots 1 \}$. Dla N-elementowych ciągów taka przestrzeń Ω_0 składa się z 2^N elementów. W tej przestrzeni działają bezpośrednio „prymitywne” algorytmy przeszukiwania (np. algorytm złotego podziału, przeszukiwanie binarne, metody enumeracyjne, pełen przegląd). Poszczególne elementy przestrzeni Ω_0 odpowiadają elementom przestrzeni rozwiązań X (odwzorowanie za pomocą prostej bijekcji), co zostało zilustrowane na **Rysunku 5.1**.

5.1.2 Przestrzeń schematów

Dyskretna, skończona przestrzeń schematów $\Omega_H = \{ ** \dots *, ** \dots 0, \dots, 11 \dots 1 \}$. Jak wynika z definicji schematu, przedstawionej wraz z podstawowymi własnościami w **Rozdziale 1.4 (str. 25)**, dla N-elementowych ciągów taka przestrzeń Ω_H składa się z 3^N elementów. Pozycje oraz liczba wystąpień znaku $*$ w schemacie $H \in \Omega_H$ wpływa na rozkład elementów przestrzeni ciągów binarnych Ω_0 , pasujących do danego schematu. Dla przykładowo wybranej funkcji $f(x)$, schemat $0*** \dots$ odpowiada pierwszej połowie dziedziny $[0, 100]$, zaś schemat $\dots ***1$ odpowiada co drugiemu ciągowi binarnemu w Ω_0 (elementy parzyste, zaczynając indeksowanie ciągów w zbiorze Ω_0 od wartości 1). Przestrzeń Ω_H odpowiada przestrzeni generowanej przez jądro transformacji Walsh’a, co analizowano m.in. w pracach (Vose, Wright, 1998a,b; Vose, 1999). Jak zostało wykazane przez Hollanda, w przestrzeni schematów działa algorytm SGA, którego „siła” wynika z przetwarzania dużej liczby schematów (tzw. ukryty paralelizm) (Holland, 1975). Podzbiory przestrzeni X , odpowiadające trzem przykładowym schematom różnych długości $d(H)$ i różnych rzędów $o(H)$, zostały przedstawione na **Rysunku 5.2**. Odpowiadające wybranym schematom przedziały w przestrzeni rozwiązań X zostały zaznaczone na wykresach kolorem szarym. Jak widać na kolejnych wykresach, im niższy rząd schematu $o(H)$, tym większą część przestrzeni rozwiązań X pokrywa dany schemat.



RYS. 5.2: Podzbiory w przestrzeni rozwiązań X , odpowiadające wybranym schematom



Rys. 5.3: Rozkłady prawdopodobieństw w przestrzeni rozwiązań X , odpowiadające zbiorom niezależnych genów kwantowych

5.1.3 Przestrzeń stanów binarnego chromosomu kwantowego

Liniowa przestrzeń stanów zbioru N niezależnych binarnych genów kwantowych $|q\rangle = |g_0\rangle, |g_1\rangle, \dots, |g_N\rangle$ (genotyp kwantowy w algorytmie QIGA). Ponieważ każdy gen kwantowy $|g\rangle$ koduje dwie amplitudy prawdopodobieństwa α i β , jest to przestrzeń o wymiarze $2N$ i o nieskończonej liczbie elementów, ponieważ $\alpha, \beta \in [0, 1]$. Jednak z faktu iż α oraz β są zależne od siebie (zgodnie z własnościami podanymi w [Rozdziale 2.1](#), $|\alpha|^2 + |\beta|^2 = 1$), tylko N zmiennych jest niezależnych. W tej przestrzeni działają dotychczasowe kwantowo inspirowane algorytmy genetyczne QIGA ([Han, Kim, 2000](#)).

Prowadząc analizę schematów i binarnych chromosomów kwantowych, można zauważyć interesującą własność. W tym celu przeanalizowane zostały schematy oraz binarne chromosomy kwantowe, uzyskane poprzez zastąpienie w schemacie symboli nieoznaczonych $*$ genem kwantowym w stanie równomiernej superpozycji $\mathcal{Q}$.

Niech H oznacza N -elementowy schemat i niech $\{v_1, \dots, v_p\}$ oznacza p -elementowy zbiór wszystkich ciągów binarnych pasujących do schematu H . Dopasowanie schematu $f(H)$ określa się jako średnie dopasowanie wszystkich pasujących do niego ciągów binarnych o długości N ([Michalewicz, 2003](#), str. 74), tzn.:

$$f(H) = \sum_{j=1}^p \frac{f(v_j)}{p} \quad (5.2)$$

Należy zauważyć, iż zastępując w schemacie H wszystkie symbole nieoznaczone $*$ binarnymi genami kwantowymi w równomiernej superpozycji $\frac{\sqrt{2}}{2}(|0\rangle + |1\rangle) = \mathcal{Q}$, otrzymuje się binarny chromosom kwantowy $|q\rangle$, którego wielokrotna obserwacja i ocena przybliży w sposób stochastyczny (analogicznie do działania metod Monte Carlo ([Metropolis, 1987](#))) dopasowanie schematu H :

$$\lim_{t \rightarrow \infty} \frac{f(\text{observe}(|q\rangle))}{t} = f(H) \quad (5.3)$$

gdzie $t \in \mathbb{N}$ to numer generacji w algorytmie.

Podzbiory przestrzeni rozwiązań X , odpowiadające wybranym zbiorom niezależnych binarnych genów kwantowych, zostały przedstawione

na **Rysunku 5.3**. Odcienie szarości na wykresach ilustrują prawdopodobieństwo spróbkowania danego punktu w przestrzeni rozwiązań X , przy wylosowaniu punktu zgodnie z rozkładem prawdopodobieństwa zapisanym w $|q\rangle$. Ciemniejsze odcienie szarości oznaczają większe prawdopodobieństwo wylosowania punktów z danego przedziału w X , zaś jaśniejsze oznaczają mniejsze prawdopodobieństwo takiego zdarzenia. Dokonując porównania wykresów na **Rysunku 5.2 i 5.3**, można łatwo zauważyć, iż dopasowanie schematu może być przybliżone poprzez obliczenie średniej wartości dopasowania klasycznych chromosomów, otrzymanych poprzez wielokrotne próbkowanie chromosomu kwantowego, odpowiadającego schematowi (probabilistyczna równoważność schematów i binarnych chromosomów kwantowych).

Dla dwóch przykładowych schematów $H = * * *$ i $H = 0100 * 1$ oraz odpowiadających im binarnych chromosomów kwantowych:

$$f(* * *) = \lim_{t \rightarrow \infty} \frac{\sum_t f(\textcircled{0}\textcircled{0}\textcircled{0})}{t} \quad (5.4)$$

$$f(0100 * 1) = \lim_{t \rightarrow \infty} \frac{\sum_t f(0100\textcircled{0}1)}{t}$$

Równości te pokazują, że dopasowanie schematu $H = * * *$ (z definicji równe średniej wartości wszystkich rozwiązań z przestrzeni X) można przybliżyć poprzez wielokrotną obserwację binarnego chromosomu kwantowego $\textcircled{0}\textcircled{0}\textcircled{0}$ i wyciągnięcie wartości średniej z dopasowań chromosomów otrzymanych w rezultacie dokonywanych obserwacji. Przy liczbie obserwacji genotypu dążącej do nieskończoności ($t \rightarrow \infty$), ta średnia odpowiada dokładnie dopasowaniu schematu $f(* * *)$. Podobnie, dopasowanie schematu $H = 0100 * 1$ może być przybliżone poprzez wielokrotną obserwację chromosomu kwantowego $0100\textcircled{0}1$ i granicą tej średniej przy t dążącym do nieskończoności jest wartość dopasowania schematu H .

Pokazuje to, iż algorytmy QIGA, operując na binarnych chromosomach kwantowych, dokonują stochastycznej oceny schematów. Analogicznie jak zostało wykazane przez Holanda, że „siła SGA tkwi w ukrytym przetwarzaniu dużej liczby schematów”, stwierdzona własność

binarnych chromosomów kwantowych pokazuje, iż „siła” kwantowo inspirowanych algorytmów genetycznych tkwi m.in. w probabilistycznej ocenie dopasowania schematów, które mogą być reprezentowane przez binarne chromosomy kwantowe. Jednak w odróżnieniu od schematów, których dopasowanie może być przybliżane w sposób stochastyczny przez wielokrotną obserwację binarnych chromosomów kwantowych, binarne chromosomy kwantowe, których stan określony jest w przestrzeni liniowej, posiadają wyższy stopień ogólności, co pozwala nie tylko wybierać określone przedziały w przestrzeni rozwiązań X , ale również zapisywać różne prawdopodobieństwo wyboru poszczególnych elementów przestrzeni $x \in X$.

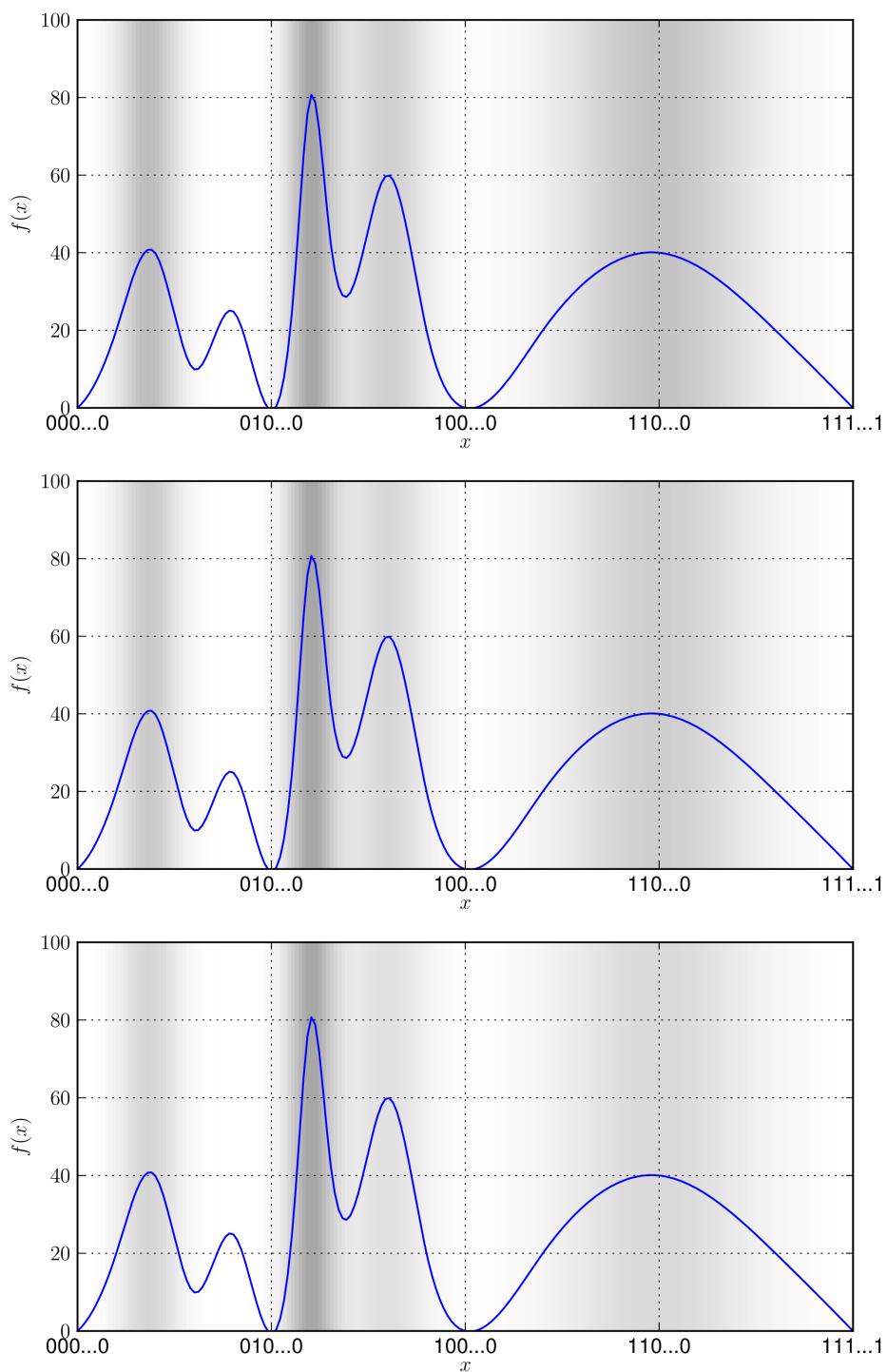
Binarne chromosomy kwantowe, przedstawione na wykresach na [Rysunku 5.3](#), to odpowiednio:

$$\begin{aligned}
 |q_1\rangle &= \begin{array}{cccccccc} \nearrow & \uparrow & \nearrow & \rightarrow & \nearrow & \nearrow & \nearrow & \nearrow \end{array} \\
 &= \left[\begin{array}{c|c|c|c|c|c|c|c} \frac{\sqrt{2}}{2} & 1 & \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \hline \frac{\sqrt{2}}{2} & & \frac{\sqrt{2}}{2} & & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{array} \right] \\
 \\
 |q_2\rangle &= \begin{array}{cccccccc} \nearrow & \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow \end{array} \\
 &= \left[\begin{array}{c|c|c|c|c|c|c|c} 0.866 & 0.174 & 0.819 & 0.819 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \hline 0.500 & 0.985 & 0.574 & 0.574 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{array} \right] \\
 \\
 |q_3\rangle &= \begin{array}{cccccccc} \nearrow & \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow \end{array} \\
 &= \left[\begin{array}{c|c|c|c|c|c|c|c} 0.940 & 0.000 & 0.707 & 0.866 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \hline 0.342 & 1.000 & 0.707 & 0.500 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{array} \right]
 \end{aligned}$$

W pierwszym z tych chromosomów kwantowych kolejne geny mają wartość 0, 1 lub stan superpozycji $[\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2}]^T = \nearrow$. Należy zauważyć, że w tym przypadku rozkład prawdopodobieństwa zakodowany w chromosomie kwantowym odpowiada podzbiorowi przestrzeni X , pasującej do schematu $H = *1*0****$ (pierwsze wykresy od góry na [Rysunku 5.2](#) oraz [5.3](#)). W chromosomach kwantowych $|q_2\rangle$ i $|q_3\rangle$ stany superpozycji zostały nieznacznie zmodyfikowane, co znajduje odzwierciedlenie w rozkładach prawdopodobieństw kodowanych w tych chromosomach (odpowiednio wykres 2. i 3. na [Rysunku 5.3](#)).

5.1.4 Przestrzeń stanów rejestru kwantowego

Liniowa przestrzeń stanów chromosomu kwantowego tworzonego przez pełen rejestr kwantowy $|q\rangle$ o rozmiarze N . Własności tej przestrzeni wynikają z podstaw Informatyki Kwantowej (**Rozdział 2, str. 38**). Ponieważ w komputerze kwantowym z każdym elementem przestrzeni X związana jest amplituda prawdopodobieństwa $\alpha_0, \alpha_1, \dots, \alpha_{2^N-1}$, wymiar tej przestrzeni stanów rejestru $|q\rangle$ to 2^N . Rejestr kwantowy $|q\rangle$ pozwala więc na pełen zapis dowolnego 2^N -punktowego rozkładu prawdopodobieństwa próbkowania przestrzeni X , a komputer kwantowy przetwarzałby lub próbowałby cały taki rozkład w pojedynczym kroku obliczeniowym. Przykładowe rozkłady prawdopodobieństwa próbkowania przestrzeni X zostały zilustrowane na **Rysunku 5.4**. Odcienie szarości na wykresach oznaczają prawdopodobieństwo wyboru danego elementu przestrzeni rozwiązań X , dokonując próbkowania zgodnie z rozkładem zapisanym w q . Im ciemniejszy kolor, tym większe prawdopodobieństwo wyboru danego elementu, zaś im jaśniejszy kolor, tym prawdopodobieństwo tego zdarzenia jest mniejsze.



RYS. 5.4: Przykładowe rozkłady prawdopodobieństw kodowane w rejestrze kwantowym $|q\rangle$

Własności omówionych powyżej kolejno przestrzeni przedstawione zostały w Tabeli 5.1. W trzeciej kolumnie tabeli podano odpowiadające tym przestrzeniom wartości rzędu i współczynnika kwantowości r i λ , które są wprowadzone i wyjaśnione w dalszej części rozdziału. W tabeli zaznaczono również usytuowanie nowej klasy kwantowo inspirowanych algorytmów genetycznych wyższych rzędów, pomiędzy dotychczasowymi algorytmami QIGA (przestrzenie ich działania to trzeci wiersz tabeli), a algorytmami kwantowymi (ostatni wiersz tabeli). Aby przeanalizować algorytmy, sytuujące się „pomiędzy” istniejącymi algorytmami QIGA oraz prawdziwymi algorytmami kwantowymi, sformułowane zostały nowe pojęcia rzędu oraz współczynników kwantowości algorytmu.

TABELA 5.1: Przestrzenie działania algorytmów i ich własności

Przestrzeń	Własności	λ, r
ciągi binarne X	skończona, dyskretna przestrzeń $X = \{0, 1\}^N$	$\lambda = 0$
schematy Ω_H	skończona, dyskretna przestrzeń $\Omega_H = \{0, 1, *\}^N$	
ciągi niezależnych kubitów Ω_{QI}	przestrzeń liniowa, $\dim(\Omega_{QI}) = N$	$\lambda \gtrsim 0$ $r = 1$
Przestrzenie wyższych wymiarów		$\vdots$
przestrzeń stanów rejestrów kwantowego $\mathcal{H}$	zespolona przestrzeń Hilberta, $\dim(\mathcal{H}) = 2^N$	$\lambda = 1$ $r = N$

5.2 PODSTAWY ALGORYTMÓW WYŻSZYCH RZĘDÓW

5.2.1 Struktura algorytmu wyższego rzędu

Ogólny schemat kwantowo inspirowanego algorytmu QIGA wyższego rzędu jest taki sam jak typowego algorytmu ewolucyjnego (**Rysunek 1.2, str. 24**) i odpowiada on pseudokodowi zaprezentowanemu na **Listingu 1 (str. 49)**. Zasadnicze różnice w algorytmach wyższych rzędów występują jednak w następujących elementach:

REPREZENTACJA ROZWIĄZAŃ – w odróżnieniu od dotychczasowych algorytmów QIGA, w których reprezentacja rozwiązań oparta była na niezależnych od siebie binarnych genach kwantowych, modelowanych za pomocą kubitów, w algorytmach wyższych rzędów wykorzystywane są rejestry kwantowe, obejmujące więcej niż pojedynczy gen w chromosomie kwantowym. Ponieważ kubity, należące do tego samego rejestru kwantowego, mogą być w stanie splątany (por. **Rozdział 2.2, str. 39**), pozwala to modelować w algorytmie związki pomiędzy wartościami genów i współdziałanie genów pomiędzy sobą.

INICJALIZACJA – nowy rodzaj reprezentacja wymaga odpowiednio zmodyfikowanej procedury inicjalizacji początkowej populacji kwantowej $Q(0)$. Inicjalizacji podlegają nie pojedyncze kubity, ale rejestry kwantowe, obejmujące kilka genów. Mogą być one inicjalizowane np. za pomocą równomiernej superpozycji wszystkich stanów lub za pomocą losowego stanu rejestru kwantowego (odpowiada to losowemu rozkładowi prawdopodobieństwa zapisanemu w rejestrze).

OPERATORY GENETYCZNE – wykorzystanie rejestrów kwantowych, obejmujących zbiory genów, wymaga również zastosowania odpowiednich operatorów genetycznych. Ponieważ stany r -kubitowych rejestrów kwantowych modyfikowane są przez bramki kwantowe o rozmiarze $2^r \times 2^r$ (por. **Rozdział 2.3, str. 40**), zwiększeniu ulega wymiar przestrzeni operatorów, działających na danym zbiorze genów.

5.2.2 Rząd kwantowości algorytmu

Wprowadzone zostały następujące nowe pojęcia dla kwantowo inspirowanych algorytmów genetycznych QIGA.

Definicja 5.1. Dany algorytm jest **rzędu kwantowości** $r \in \mathbb{N}^+$, jeśli r określa rozmiar największego rejestru kwantowego, wykorzystywanego w algorytmie (np. kodującego ciąg genów). Rząd kwantowości musi spełniać warunki:

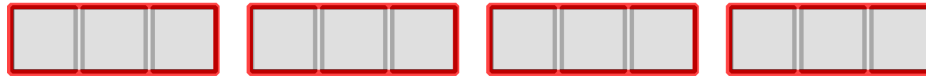
$$1 \leq r \leq N \quad (5.5)$$

gdzie N to długość chromosomów kwantowych w algorytmie.

Wszystkie dotychczasowe kwantowo inspirowane algorytmy genetyczne QIGA, w których reprezentacja rozwiązań jest oparta na ciągach niezależnych kubitów (niezależne binarne geny kwantowe), charakteryzowane są przez rząd kwantowości $r = 1$.



Chromosom w algorytmie rzędu I (12 kubitów)



Chromosom w algorytmie rzędu III (4 rejestry kwantowe)

RYS. 5.5: Geny połączone do wspólnych rejestrów kwantowych

Definicja 5.2. Dla danego algorytmu **względny rząd kwantowości** $w \in [0, 1]$ to stosunek rzędu kwantowości algorytmu r i rozmiaru zadania N (długości chromosomów).

$$w = \frac{r}{N} \quad (5.6)$$

Jeśli w pewnym algorytmie QIGA wykorzystywana jest reprezentacja rozwiązań oparta na 100 niezależnych kubitach (binarnych genach kwantowych), to względny rząd kwantowości tego algorytmu $w = \frac{1}{100} = 0.01$. Jeśli

rozmiar zadania (liczba zmiennych binarnych) to $N = 60$, a reprezentacja oparta jest na 3-kubitowych rejestrach kwantowych, to względny rząd kwantowości wynosi $w = \frac{3}{60} = 0.05$.

Algorytmy charakteryzowane przez $w = 1$ to „prawdziwe” algorytmy kwantowe, w których pojedynczy rejestr kwantowy obejmuje wszystkie zmienne. Dla $w = 1$, przy liniowym wzroście liczby zmiennych (rozmiar zadania N), rośnie w tempie wykładniczym koszt obliczeniowy symulacji (odpowiada to symulacji pełnego komputera kwantowego).

5.2.3 Współczynniki kwantowości algorytmu

Dla uproszczenia dalszych rozważań przyjęto, że grupy, w które połączone są geny, tworząc r -elementowe rejestry kwantowe, będą równoliczne (r dzieli N bez reszty). Na **Rysunku 5.5** przedstawiono dwa przykładowe chromosomy kwantowe w algorytmach o rzędzie odpowiednio $r = 1$ i $r = 3$. W algorytmie rzędu I występuje 12 binarnych genów kwantowych, modelowanych za pomocą niezależnych kubitów. Przestrzeń każdego kubitów jest przestrzenią 2-wymiarową, więc wymiar całego chromosomu wynosi $2N = 24$, co oznacza, że dla zapisania stanu całego chromosomu potrzebne są 24 zmienne rzeczywiste. Jednak ze względu na związek pomiędzy amplitudami $|\alpha|^2 + |\beta|^2 = 1$ tylko 12 zmiennych jest niezależnych. W algorytmie rzędu III dla chromosomu o długości $N = 12$ występują 4 sąsiadujące ze sobą 3-kubitowe rejestry kwantowe. Wymiar przestrzeni każdego z takich rejestrów to $2^r = 2^3 = 8$, a do zapisania stanu całego chromosomu kwantowego potrzebne są $8 \cdot 4 = 32$ zmienne rzeczywiste, zaś liczba zmiennych niezależnych dla całego chromosomu kwantowego to $(2^3 - 1) \cdot 4 = 28$.

Definicja 5.3. Dla danego algorytmu definiuje się **współczynnik kwantowości** $\lambda \in [0, 1]$ jako stosunek wymiaru przestrzeni genotypu kwantowego w danej klasie algorytmów do wymiaru przestrzeni pełnego rejestru kwantowego o rozmiarze równym wielkości zadania N . Dodatkowo przyjmijmy, że jeśli algorytm nie ma w ogóle elementu kwantowego (np. prosty algorytm genetyczny SGA, działający w dyskretnej przestrzeni ciągów binarnych), to $\lambda = 0$.

Liczbowa wartość współczynnika λ wyraża się więc jako:

$$\lambda = \frac{2^{\frac{r}{N}}}{2^N} = \frac{2^r}{w2^N} \quad (5.7)$$

gdzie r to rząd kwantowości algorytmu, a N to rozmiar zadania.

Zapisanie współczynnika λ w ten prosty sposób możliwe jest dzięki przyjętemu wcześniej założeniu, że r dzieli N bez reszty. Jeżeli N nie jest wielokrotnością r , wtedy wartość współczynnika kwantowości λ ma postać:

$$\lambda = \frac{2^{r \lfloor \frac{N}{r} \rfloor} + 2^{(N \bmod r)}}{2^N} \quad (5.8)$$

Czynnik 2^r w liczniku we wzorze (5.7) odpowiada wymiarowi przestrzeni stanów r -kubitowego rejestru kwantowego. Taki rejestr kwantowy koduje 2^r -punktowy rozkład prawdopodobieństwa i opisuje prawdopodobieństwo wyboru jednego spośród 2^r elementów z przestrzeni rozwiązań X .

Należy zauważyć, że dla $r = 1$ (wszystkie dotychczasowe kwantowo inspirowane algorytmy genetyczne QIGA):

$$\lambda = \frac{2^{\frac{1}{N}}}{2^N} = \frac{2 \cdot N}{2^N} \quad (5.9)$$

więc np. w algorytmie, kodującym rozwiązania w postaci 10-elementowych ciągów niezależnych kubitów, $\lambda = \frac{20}{2^{10}} \approx 0.02$, co należy interpretować, że wymiar przestrzeni w tym algorytmie stanowi ok. 2% wymiaru przestrzeni stanów pełnego rejestru kwantowego, który obejmowałby 10 zmiennych binarnych. Wraz ze wzrostem rozmiaru zadania N i stałego rzędu $r = 1$ współczynnik kwantowości λ maleje w tempie zbliżonym do wykładniczego, dla $N = 50$ przyjmując wartość $\lambda < 10^{-10}$.

Wobec powyższego, dla stałego rzędu kwantowości $r = 1$ (algorytmy kwantowo inspirowane QIGA rzędu I) i rosnącej wielkości zadania N , współczynnik kwantowości λ ma granicę równą zero:

$$\lim_{\substack{r=1 \\ N \rightarrow \infty}} \lambda = \lim_{\substack{r=1 \\ N \rightarrow \infty}} \frac{2 \cdot N}{2^N} = 0$$

Natomiast dla $r = N$ (typowe algorytmy kwantowe):

$$\lambda = \frac{2^{\frac{N}{N}}}{2^N} = \frac{2^1}{2^N} = 1 \quad (5.10)$$

Dla $\lambda = 1$, przy liniowym wzroście liczby zmiennych (rozmiaru zadania N), rośnie w tempie wykładniczym koszt obliczeniowy symulacji (odpowiada to symulacji pełnego komputera kwantowego).

W zależności od wartości λ zachodzi więc jeden z następujących przypadków (Tabela 5.1 na str. 95):

1. $\lambda = 0$ – typowy klasyczny algorytm, nie posiadający elementu kwantowego, operujący w dyskretnej, skończonej przestrzeni (np. SGA, operujący w skończonej, dyskretnej przestrzeni ciągów binarnych).
2. $\lambda \in (0, 1)$ – algorytm kwantowo inspirowany: typowy QIGA (rzędu $r = 1$) albo QIGA wyższych rzędów. Dla rozmiarów zadania $N \gg r$, współczynnik kwantowości $\lambda \gtrsim 0$.
3. $\lambda = 1$ – algorytm kwantowy, wymagający komputera kwantowego albo symulacji obciążonej wykładniczym kosztem obliczeniowym.

Ponieważ r -kubitowy rejestr kwantowy pozwala modelować zależności pomiędzy r zmiennymi binarnymi (por. Rozdział 2.2, str. 39), kwantowo inspirowane algorytmy genetyczne rzędu r mają zdolność rozwiązywania problemu epistazy pomiędzy genami, których odległość w chromosomie kwantowym jest nie większa niż r . Pozwala to poprawić efektywność algorytmów w szczególności w zwodniczych zadaniach optymalizacji kombinatorycznej. Eksperymenty numeryczne, będące doświadczalnym dowodem tego stwierdzenia, zostały przedstawione w Rozdziale 6.

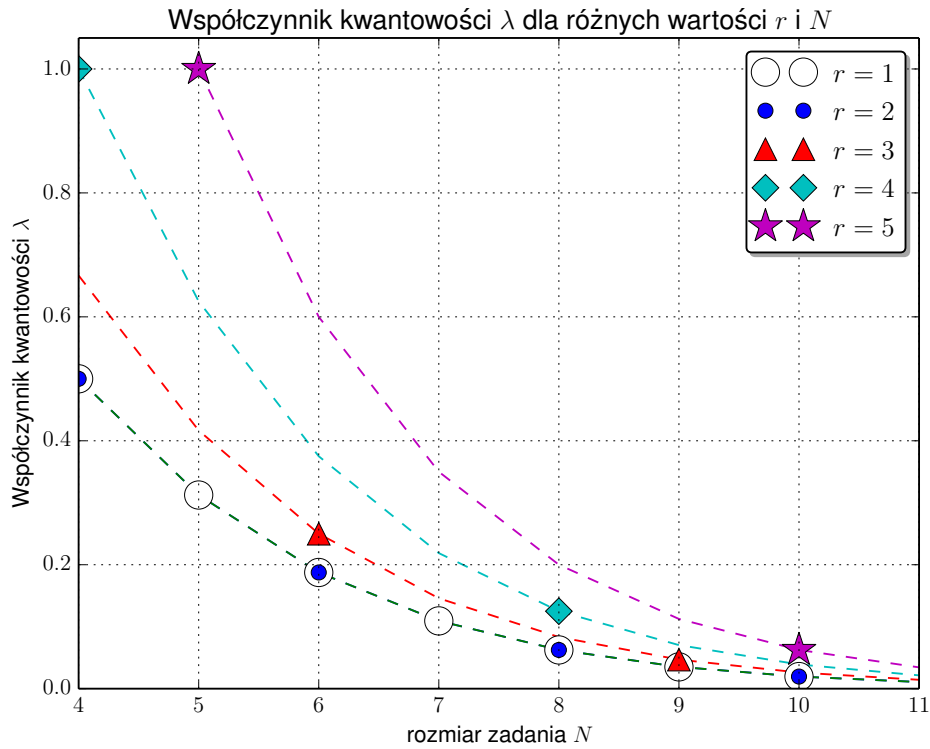
Definicja 5.4. Dla danego algorytmu definiuje się **współczynnik kwantowości operatorowej** $\lambda_O \in [0, 1]$ jako stosunek wymiaru bramek kwantowych w algorytmach danego rzędu do wymiaru przestrzeni operatorów w pełnym rejestrze kwantowym dla ustalonej wielkości zadania N . Liczbową wartość λ_O wyraża się więc jako:

$$\lambda_O = \frac{(2^r)^2 \frac{N}{r}}{(2^N)^2} = \frac{2^{2r} \frac{N}{r}}{2^{2N}} \quad (5.11)$$

gdzie r to rząd kwantowości algorytmu, a N to rozmiar zadania.

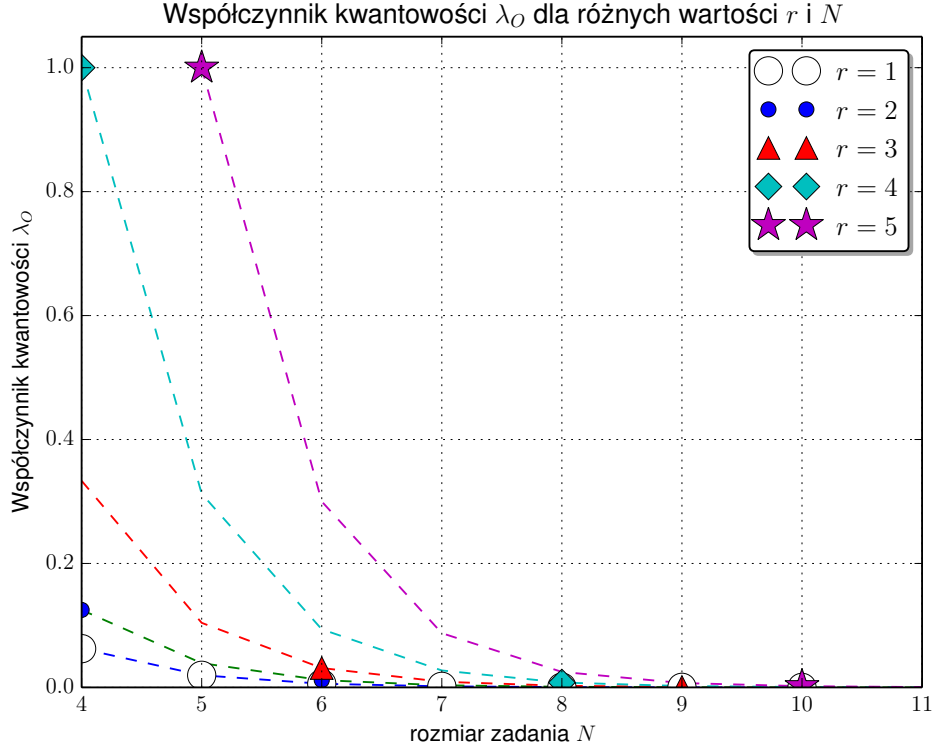
Wzór (5.11) można uzasadnić w następujący sposób: Bramki kwantowe dla r -kubitowych rejestrów kwantowych opisywane są przez macierze

unitarne rozmiaru $2^r \times 2^r$ (Rozdział 2), dlatego przestrzeń operatorów dla r -kubitowych rejestrów jest przestrzenią $(2^r)^2 = 2^{2r}$ -wymiarową (naturalnie istnieją ograniczenia w tej przestrzeni, wynikające z warunku unitarności operatora definiowanego przez daną macierz). Dla rozmiaru zadania N potrzebne jest $\frac{N}{r}$ rejestrów kwantowych o rozmiarze r , stąd wynika drugi czynnik w liczniku we wzorze (5.11). Mianownik we wzorze odpowiada natomiast wymiarowi bramki kwantowej, operującej na N -kubitowym rejestrze kwantowym.



RYS. 5.6: Wartość współczynnika kwantowości λ dla różnych rzędów kwantowości r i różnych rozmiarów zadań N

Na Rysunku 5.6 i 5.7 zostały przedstawione odpowiednio wartości współczynników λ i λ_O dla różnych rzędów kwantowości algorytmu r i różnych rozmiarów zadania N . Wykresy na tych rysunkach zostały wygenerowane za pomocą wzorów odpowiednio (5.7) i (5.11). Za pomocą symboli na wykresach oznaczono punkty (N, λ) , dla których spełnione jest założenie przyjęte na początku, $N \bmod r = 0$ (tzn. długość chromosomu



RYS. 5.7: Wartość współczynnika kwantowości operatorowej λ_O dla różnych rzędów kwantowości r i różnych rozmiarów zadań N

$N \in \mathbb{N}^+$ jest wielokrotnością rzędu r), linie przerywane prezentują natomiast to, w jaki sposób współczynnik λ maleje wraz ze wzrostem rozmiaru zadania N . Różne rodzaje symboli na wykresach odpowiadają różnym rzędom kwantowości r – białe okręgi (rząd $r = 1$), niebieskie punkty ($r = 2$), czerwone trójkąty ($r = 3$), cyjanowe romby ($r = 4$) oraz magentowe gwiazdy ($r = 5$). Należy zauważyć, że dla stałego rzędu kwantowości r , wraz ze wzrostem rozmiaru zadania maleją w tempie zbliżonym do wykładniczego wartości współczynników λ i λ_O (tzn. maleje stosunek wymiarów przestrzeni algorytmu rzędu r do wymiarów przestrzeni pełnych rejestrów kwantowych dla takiego samego rozmiaru zadania).

Szczególna własność występuje dla algorytmów QIGA rzędu I i rzędu II. Algorytmy tych rzędów mają te same współczynniki kwantowości λ , ale różne współczynniki kwantowości operatorowej λ_O . Takie same współczynniki kwantowości λ dla algorytmów rzędu I i II wynikają z faktu, że

wymiary przestrzeni stanów genotypów w tych algorytmach są takie same ($2N = 2^2 \frac{N}{2}$), pomimo że w algorytmie rzędu I występuje N zmiennych niezależnych, natomiast w algorytmie rzędu II występuje $(2^2 - 1) \frac{N}{2} = \frac{3}{2}N$ zmiennych niezależnych. Ta własność algorytmów rzędu $r = 2$ została wykorzystana do opracowania autorskiego algorytmu QIGA2 w dalszej części rozdziału.

Reprezentacja rozwiązań w algorytmach rzędu $r = 1$ oparta jest na przestrzeni $2N$ -wymiarowej (dla każdej zmiennej binarnej kodowany jest 2-punktowy rozkład prawdopodobieństwa, dwie zmienne kodowane są przez 4 amplitudy prawdopodobieństwa itd), natomiast w algorytmach rzędu $r = 2$ rozwiązania kodowane są przez $\frac{N}{r}$ 2-kubitowych rejestrów kwantowych (opisywanych przez 4 amplitudy prawdopodobieństwa)¹. Algorytmy rzędu I i II są zatem równoważne w sensie współczynników kwantowości λ , ponieważ:

$$\forall_{N \in \mathbb{N}^+} 2^1 \frac{N}{2^N} = 2^2 \frac{\frac{N}{2}}{2^N} \quad (5.12)$$

Współczynniki kwantowości operatorowej λ_O algorytmów rzędu I i II są natomiast różne, ponieważ przestrzeń operatorów kwantowych w algorytmie rzędu I generowana jest przez N macierzy unitarnych o wymiarach 2×2 (1-kubitowe bramki kwantowe), natomiast w algorytmach rzędu II jest ona generowana przez $\frac{N}{2}$ macierzy unitarnych o wymiarach 4×4 .

5.2.4 Przestrzenie działania algorytmów wyższych rzędów

Podobnie jak w przypadku przestrzeni stanów binarnego chromosomu kwantowego w algorytmie QIGA (opisanej w [Rozdziale 5.1.3](#)), przestrzenią działania algorytmu wyższego rzędu jest przestrzeń liniowa o nieskończonej liczbie elementów (ponieważ amplitudy prawdopodobieństw są liczbami rzeczywistymi).

Na [Rysunku 5.8](#) przedstawione zostały trzy wybrane chromosomy kwantowe w algorytmie rzędu $r = 2$, dla rozmiaru zadania $N = 6$. Chromosom kwantowy zbudowany jest w tym przypadku z trzech sąsiadujących ze sobą 2-kubitowych rejestrów kwantowych. Za pomocą odcieni

¹ Ta własność algorytmów rzędu $r = 1$ i $r = 2$ wynika po prostu z faktu, że $2 \cdot 2 = 2^2$, co nie zachodzi dla większych niż 2 liczb naturalnych.

szarości zaznaczono prawdopodobieństwo próbkowania danego obszaru w przestrzeni rozwiązań X , zgodnie z rozkładem zapisanym w danym chromosomie kwantowym algorytmu II rzędu.

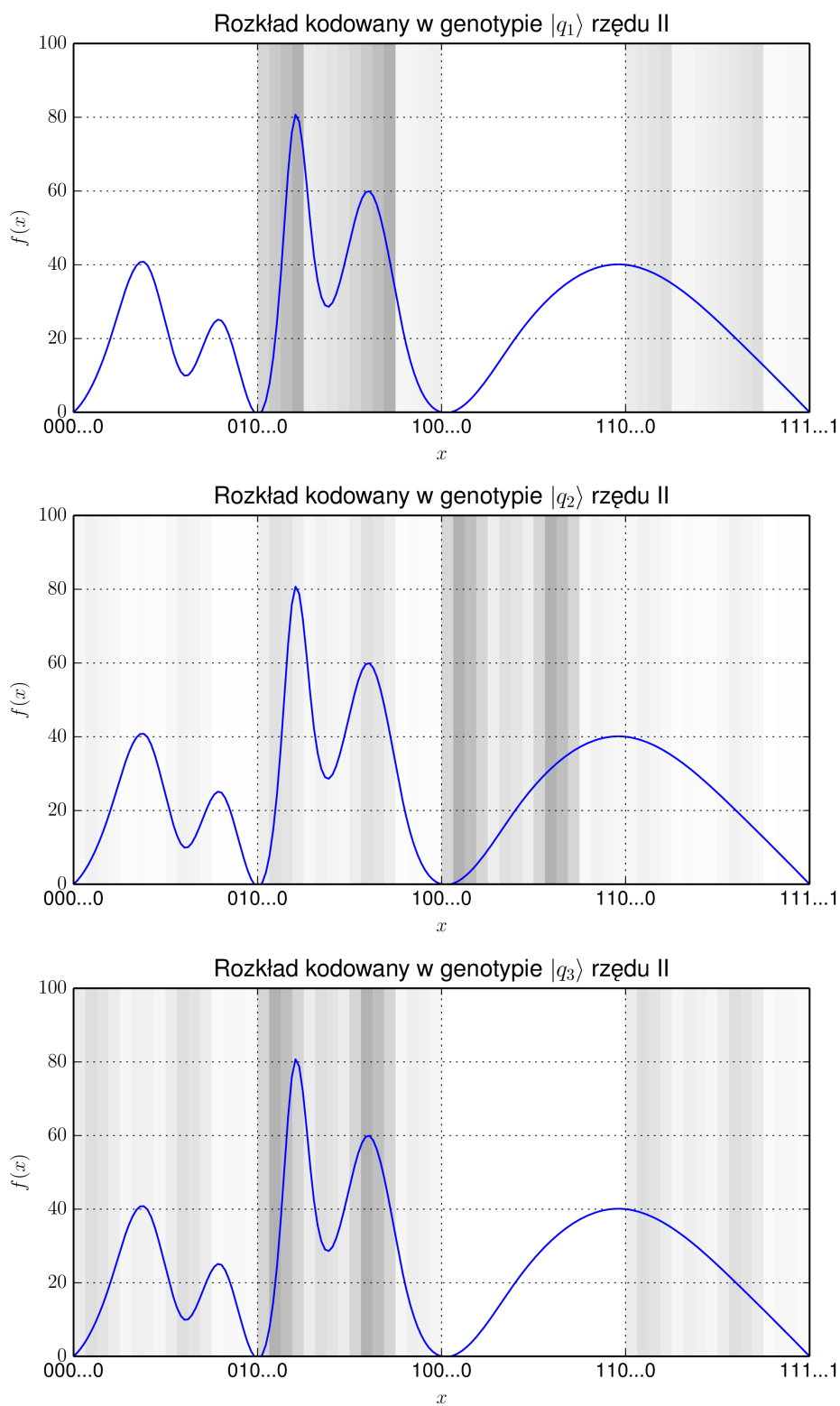
Genotypy kwantowe rzędu II, zaprezentowane na wykresach, to odpowiednio:

$$|q_1\rangle = \left[\begin{array}{c|c|c} 0.000 & 0.658 & 0.379 \\ 0.894 & 0.329 & 0.455 \\ 0.000 & 0.658 & 0.531 \\ 0.447 & 0.164 & 0.606 \end{array} \right] \quad (5.13)$$

$$|q_2\rangle = \left[\begin{array}{c|c|c} 0.213 & 0.658 & 0.392 \\ 0.426 & 0.329 & 0.627 \\ 0.853 & 0.658 & 0.548 \\ 0.213 & 0.164 & 0.392 \end{array} \right] \quad (5.14)$$

$$|q_3\rangle = \left[\begin{array}{c|c|c} 0.408 & 0.658 & 0.392 \\ 0.816 & 0.329 & 0.627 \\ 0.000 & 0.658 & 0.548 \\ 0.408 & 0.164 & 0.392 \end{array} \right] \quad (5.15)$$

Rozkład prawdopodobieństwa zapisany w genotypie kwantowym w algorytmie wyższego rzędu pozwala z większą dokładnością odwzorowywać ukształtowanie funkcji dopasowania $f(x)$ niż algorytm rzędu I, co zostało zilustrowane na wykresie dla trzech przykładowo wybranych genotypów kwantowych rzędu II.



RYS. 5.8: Przykłady rozkładów kodowanych w genotypach rzędu $r = 2$

5.2.5 Algorytmy wybranych rzędów dla rozmiaru zadania $N = 10$

Dla zilustrowania reprezentacji rozwiązań w algorytmach wybranych rzędów, przeanalizowano przypadek dla rozmiaru zadania $N = 10$ zmiennych binarnych i dla wybranych rzędów kwantowości ($r = 1, r = 2, r = 10$). Na **Rysunku 5.10** została przedstawiona ilustracja przestrzeni działania algorytmów tych rzędów.

W ujęciu klasycznym ($\lambda = 0$), możliwe rozwiązania zadania o rozmiarze N byłyby kodowane w postaci 10-elementowych ciągów binarnych i byłoby 2^{10} takich ciągów. Klasyczny algorytm SGA działa w tym przypadku w przestrzeni 2^{10} -elementowej przestrzeni dyskretnej: $X = \{0000000000, 0000000001, \dots, 1111111110, 1111111111\}$.

W kwantowo inspirowanym algorytmie genetycznym QIGA rzędu I ($r = 1, \lambda \gtrsim 0$) (**Rysunek 5.9a**), rozwiązania kodowane są w postaci 10 niezależnych kubitów:

$$|q_0\rangle = \begin{bmatrix} \alpha_{00} \\ \alpha_{10} \end{bmatrix}, |q_1\rangle = \begin{bmatrix} \alpha_{01} \\ \alpha_{11} \end{bmatrix}, \dots, |q_9\rangle = \begin{bmatrix} \alpha_{09} \\ \alpha_{19} \end{bmatrix}.$$

Przestrzeń kwantowych operatorów genetycznych tworzona jest zatem przez 10 macierzy unitarnych wymiarów $2^1 \times 2^1$ (**Rozdział 2.3, str. 40**).

W algorytmie QIGA rzędu II ($r = 2$) (którego reprezentacją rozwiązań i operatory przedstawiono na **Rysunku 5.9b**) rozwiązania kodowane są w postaci *niezależnych* 2-kubitowych rejestrów kwantowych $|\phi_0\rangle, |\phi_1\rangle, \dots$. Musi ich być $\frac{N}{r} = \frac{10}{2}$, aby mogły zakodować całą $4^5 = 2^{10} = 1024$ -elementową przestrzeń rozwiązań X .

$$|\phi_0\rangle = \begin{bmatrix} \alpha_{00} \\ \alpha_{10} \\ \alpha_{20} \\ \alpha_{30} \end{bmatrix}, |\phi_1\rangle = \begin{bmatrix} \alpha_{01} \\ \alpha_{11} \\ \alpha_{21} \\ \alpha_{31} \end{bmatrix}, \dots, |\phi_4\rangle = \begin{bmatrix} \alpha_{04} \\ \alpha_{14} \\ \alpha_{24} \\ \alpha_{34} \end{bmatrix},$$

Przestrzeń stanów każdego z sąsiadujących ze sobą rejestrów jest $2^2 = 4$ -wymiarową przestrzenią liniową $|\phi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$, rozpinaną przez bazę: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Amplitudy prawdopodobieństwa $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ kodują rozkład prawdopodobieństwa przyjmowania możliwych wartości przez parę sąsiadujących ze sobą binarnych genów.

Przestrzeń kwantowych operatorów genetycznych w tym przypadku tworzona jest przez 5 macierzy unitarnych wymiarów 4×4 .

W algorytmie QIGA rzędu $r = N = 10$ ($r = 10, \lambda = 1$) (którego reprezentację rozwiązań oraz operatory zilustrowano na [Rysunku 5.9c](#)) rozwiązania zakodowane są w pojedynczym rejestrze kwantowym $|\Psi\rangle$, obejmującym wszystkie 10 zmiennych binarnych. Rozmiar przestrzeni stanów takiego rejestru wynosi 2^{10} i w tym przypadku możliwe jest zakodowanie w rejestrze kwantowym dowolnego rozkładu prawdopodobieństwa próbkowania 2^{10} -elementowej przestrzeni X :

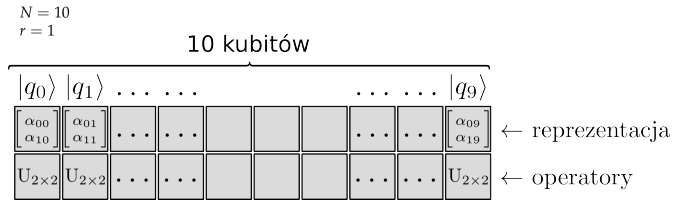
$$|\Psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_{2^N-1} = \alpha_{1023} \end{bmatrix}$$

Ponieważ możliwe jest również zapisanie rozkładów, w których wartość danego genu jest uzależniona od wartości dowolnego innego (lub wartości zespołu innych genów), reprezentacja rozwiązań w algorytmie „pełnego rzędu” ($r = N, \lambda = 1$) pozwala na zapisanie dowolnych relacji pomiędzy genami zlokalizowanymi na dowolnych pozycjach. Odpowiada to zjawisku kwantowego splątania pomiędzy genami. Ten sposób modelowania relacji pomiędzy genami pozwala przeciwdziałać problemowi zwodniczości i epistazy w algorytmie genetycznym, które oznaczają pozytywną nieliniowość i współdziałanie genów zlokalizowanych na różnych *loci* (opisane w [Rozdziale 1.3, str. 22](#)). Zatem reprezentacja rozwiązań w algorytmie QIGA rzędu r pozwala na zachowanie „wiedzy” o oddziałujących ze sobą podzbiorach zmiennych, należących do tych samych r -elementowych grup genów.

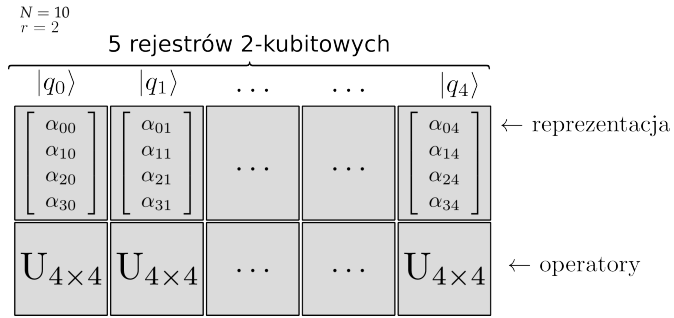
Uogólniając powyższe rozważania na wyższe rzędy kwantowości r , można zauważyć następującą własność:

Własność 5.5. Algorytm rzędu kwantowości r jest szczególnym przypadkiem algorytmu rzędu $2r$

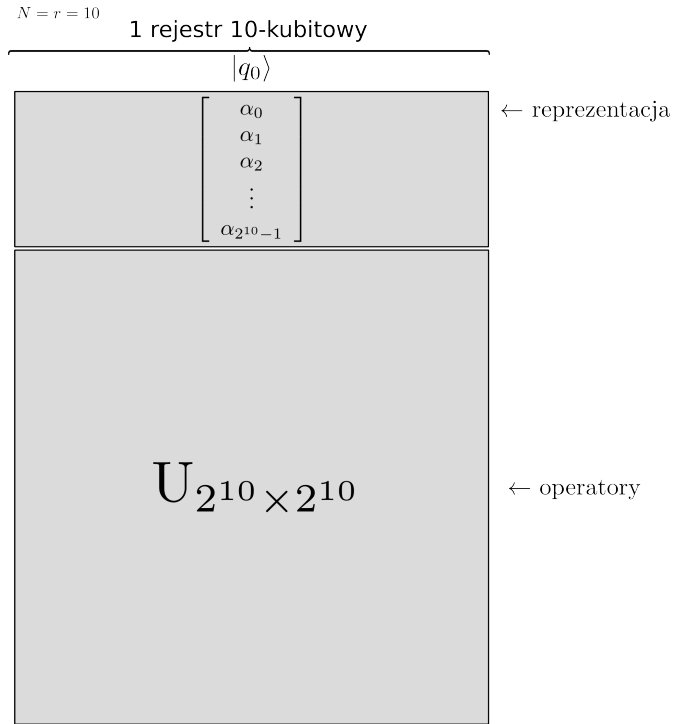
Wynika to z faktu, iż łącząc ze sobą po dwa sąsiadujące ze sobą kubity ($r = 1$) do wspólnych rejestrów kwantowych, można tworzyć algorytm wyższego rzędu ($r = 2$). Podobnie, łącząc ze sobą po dwa sąsiadujące rejestry kwantowe (dowolnego rozmiaru r), można tworzyć algorytm rzędu $2r$. W szczególności, algorytm QIGA rzędu I jest szczególnym przypadkiem algorytmu rzędu II.



(A) 10 niezależnych binarnych genów kwantowych (kubitów)

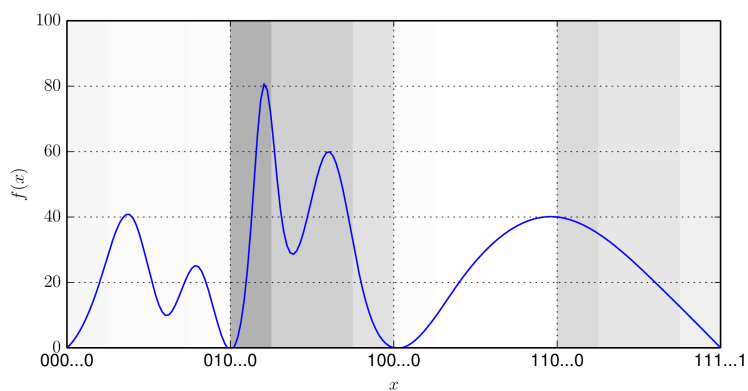


(B) 5 niezależnych 2-kubitowych rejestrów

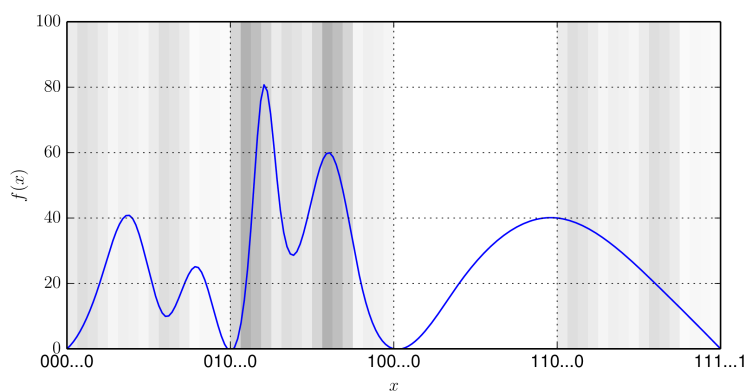


(C) Pojedynczy 10-kubitowy rejestr

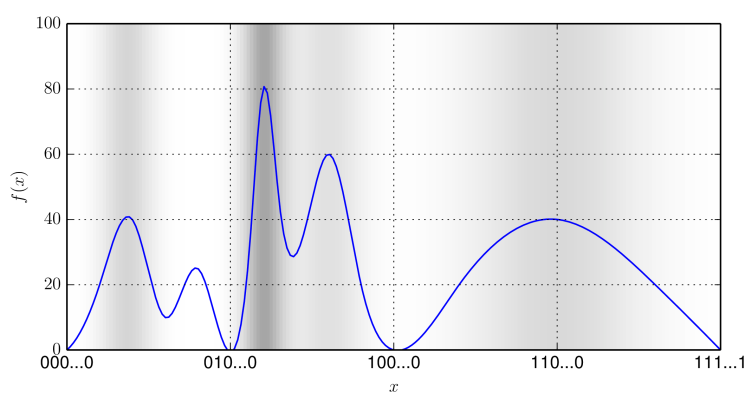
Rys. 5.9: Reprezentacja rozwiązań problemu o rozmiarze zadania $N = 10$ w algorytmach rzędów $r = 1$, $r = 2$, $r = 10$.



(A) Rząd kwantowości $r = 1$



(B) Rząd kwantowości $r = 2$



(C) Rząd kwantowości $r = N = 10$

RYS. 5.10: Przykładowe rozkłady prawdopodobieństw, kodowane w genotypach kwantowych rzędów $r = 1$, $r = 2$, $r = 10$.

5.3 AUTORSKI ALGORYTM QIGA RZĘDU II

W podrozdziale zaprezentowano szczegółowo autorski **kwantowo inspirowany algorytm genetyczny rzędu II** (ang. *Order-2 Quantum Inspired Genetic Algorithm* – **QIGA2**), opracowany w oparciu o wprowadzone wcześniej pojęcia teoretyczne. Algorytm został po raz pierwszy przedstawiony w pracy (Nowotniak, Kucharski, 2014).

Pseudokod algorytmu został zaprezentowany na **Listingu 5** i jego ogólny schemat nie różni się w żaden sposób od typowego kwantowo inspirowanego algorytmu ewolucyjnego QIEA, opisanego w **Rozdziale 3**. W każdej generacji algorytmu tworzona jest populacja klasyczna P (zbiór konkretnych elementów z przestrzeni rozwiązań X) poprzez obserwację stanów populacji kwantowej Q , tzn. $|P|$ -krotne próbkowanie przestrzeni X zgodnie z rozkładami prawdopodobieństw zapisanymi w Q . Dopasowanie $f(x)$ osobników wygenerowanej w ten sposób populacji P jest następnie obliczane w taki sam sposób jak w typowym algorytmie ewolucyjnym. Populacja kwantowa Q jest natomiast aktualizowana tak, by zwiększane było prawdopodobieństwo próbkowania najlepszego rozwiązania z poprzednich generacji, a aktualne najlepsze rozwiązanie z bieżącej generacji $P(t)$ zapisywane jest w zmiennej b .

Nowe elementy, odróżniające QIGA2 od dotychczas istniejących kwantowo inspirowanych algorytmów genetycznych rzędu I, to zmodyfikowany sposób reprezentacji rozwiązań oraz nowe operatory genetyczne, działające w przestrzeni o wyższym wymiarze i w sensie mechaniczno-kwantowym opisywane macierzami unitarnymi rozmiaru 4×4 . Elementy te zostały opisane w kolejnych podpunktach.

5.3.1 Reprezentacja rozwiązań

Podstawowa różnica pomiędzy dotychczasowymi algorytmami QIGA1 oraz QIGA2 polega na innym sposobie reprezentacji rozwiązań. W pierwotnym algorytmie QIGA1 poszczególne geny kwantowe modelowane są za pomocą kubitów, tj. dwupoziomowych układów kwantowych, mogących kodować dwupunktowe rozkłady prawdopodobieństwa. Odpowiada to możliwości przyjmowania przez pojedyncze geny wartości 0 lub 1 z praw-

Listing 5: Pseudokod autorskiego algorytmu QIGA2

```

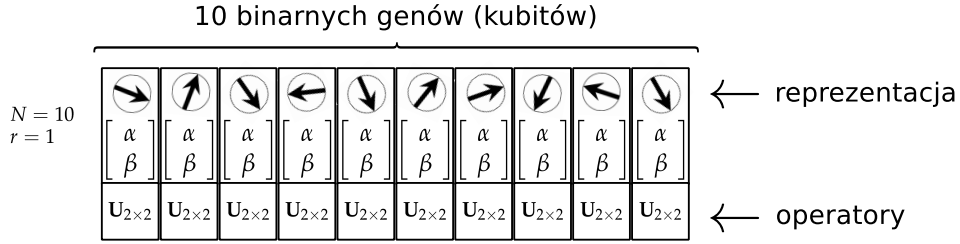
1 begin
2    $t \leftarrow 0$ 
3   Inicjalizuj populację kwantową  $Q(0)$ 
4   Utwórz populację  $P(0)$  poprzez obserwację stanów  $Q(0)$ 
5   Oceń osobniki w populacji  $P(0)$ 
6   Zapisz najlepszego osobnika z populacji  $P(0)$  w  $b$ 
7   while  $t \leq t_{\max}$  do
8      $t \leftarrow t + 1$ 
9     Utwórz populację  $P(t)$  poprzez obserwację stanów  $Q(t - 1)$ 
10    Oceń osobniki w populacji  $P(t)$ 
11    Aktualizuj stan populacji  $Q(t)$  za pomocą nowego
      operatora genetycznego
12    Zapisz najlepszego osobnika z populacji  $P(t)$  w  $b$ 
13  end
14 end

```

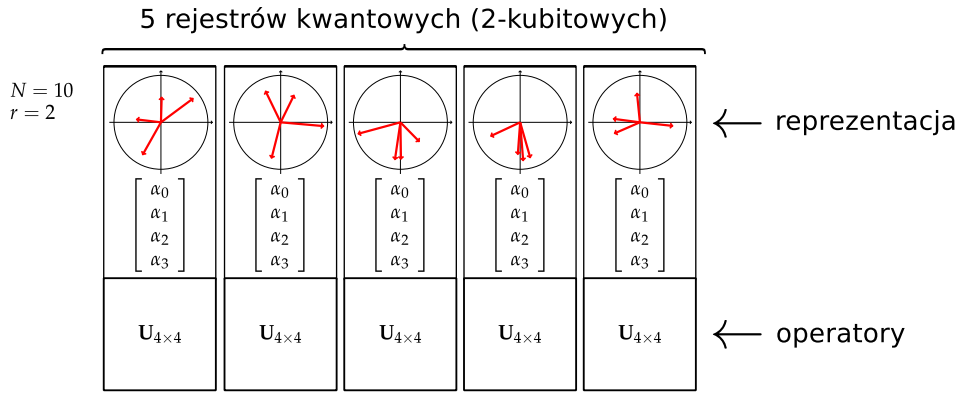
dopodobieństwem odpowiednio $|\alpha|^2$ oraz $|\beta|^2$. Zostało to zilustrowane na [Rysunku 5.11](#).

W autorskim algorytmie QIGA2 rzędu II, reprezentacja rozwiązań opiera się na wykorzystaniu sąsiadujących ze sobą 2-kubitowych rejestrów kwantowych. W tym celu sąsiadujące ze sobą geny łączone są w pary. Odpowiadające tym parom 2-kubitowe rejestry kodują 4-punktowe rozkłady prawdopodobieństw. W pojedynczym rejestrze kwantowym zapisane są zatem cztery wartości prawdopodobieństw $|\alpha_0|^2, |\alpha_1|^2, |\alpha_2|^2, |\alpha_3|^2$. Są to prawdopodobieństwa przyjmowania przez daną parę genów binarnych wartości odpowiednio 00, 01, 10 i 11. Zostało to zilustrowane na [Rysunku 5.12](#). Wartości czterech prawdopodobieństw reprezentowane są na rysunku za pomocą czterech wektorów o różnych długościach (kąt nachylenia tych wektorów należy pominąć ze względu na ograniczenie do wartości rzeczywistych). Podobnie jak w przypadku algorytmów QIGA1, w zaproponowanym algorytmie rzędu II wykorzystywane są jedynie składniki rzeczywiste amplitud prawdopodobieństwa, pominięta jest część urojona

amplitud $\alpha_0, \dots, \alpha_3$. Na etapie inicjalizacji populacji bazowej $Q(0)$ wszystkim genom kwantowym przypisywana jest wartość $g_{ij} = [\frac{1}{2} \ \frac{1}{2} \ \frac{1}{2} \ \frac{1}{2}]^T$, co odpowiada próbkowaniu przez algorytm całej przestrzeni rozwiązań X z takim samym prawdopodobieństwem.



RYS. 5.11: Reprezentacja rozwiązań w algorytmach rzędu I oparta jest na niezależnych binarnych genach kwantowych.



RYS. 5.12: Reprezentacja rozwiązań w algorytmie QIGA2 oparta jest na sąsiadujących ze sobą 2-kubitowych rejestrach kwantowych.

5.3.2 Kwantowe operatory genetyczne rzędu II

Druga istotna zmiana dotyczy wykorzystywanych w algorytmie QIGA2 operatorów genetycznych. W pierwotnym algorytmie QIGA1 operatory genetyczne tworzone są przez kwantowe bramki unitarne rozmiaru 2×2 (dzięki ograniczeniu amplitud do zbioru $\mathbb{R}$ są to proste macierze obrotów unormowanego wektora stanu na płaszczyźnie).

Na [Listing 6](#) zaprezentowany został pseudokod operacji pomiaru stanu 2-kubitowego rejestru kwantowego $q_{ij} = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle = [\alpha_0 \ \alpha_1 \ \alpha_2 \ \alpha_3]^T$, kodującego parę klasycznych genów binarnych.

Listing 6: Pomiar stanu pary genów w algorytmie QIGA2

Input: $g_{ij} = [\alpha_0 \ \alpha_1 \ \alpha_2 \ \alpha_3]^T$ – 2-kubitowy rejestr
Result: p – para klasycznych genów binarnych $\{00, 01, 10, 11\}$

```

1 begin
2    $r \leftarrow$  losowa liczba z przedziału  $[0,1]$ 
3   if  $r < |\alpha_0|^2$  then
4      $p \leftarrow 00$ 
5   else if  $r < |\alpha_0|^2 + |\alpha_1|^2$  then
6      $p \leftarrow 01$ 
7   else if  $r < |\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2$  then
8      $p \leftarrow 10$ 
9   else
10     $p \leftarrow 11$ 
11  end
12 end

```

Funkcja pomiaru stanu zwraca ciągi 00, 01, 10 i 11 z prawdopodobieństwem odpowiednio $|\alpha_0|^2$, $|\alpha_1|^2$, $|\alpha_2|^2$ oraz $|\alpha_3|^2$.

Na [Listingu 7](#) zaprezentowano natomiast pseudokod zaproponowanego nowego operatora genetycznego w QIGA2 – aktualizacja stanu 2-kubitowego rejestru kwantowego. Indeks i głównej pętli operatora iteruje przez wszystkie osobniki w populacji kwantowej $q_0, \dots, q_{|Q|-1}$. Indeks j iteruje przez wszystkie kolejne *pary* genów $j \in \{0, 1, \dots, N/2\}$ danego osobnika q_i . Wewnątrz tych pętli obliczany jest nowy stan q' pary genów kwantowych numer j aktualnego osobnika q_i . Aktualizacja ta dokonywana jest w następujący sposób: Jeżeli indeks amplitudy $amp \in \{0, 1, 2, 3\}$ nie odpowiada j -tej parze bitów (zapisanej dziesiętnie) najlepszego dotychczas znalezionej osobnika b , to ta amplituda ulega zmniejszeniu (kontrakcja amplitudy) według zasady:

$$q_{ij}[amp]' = \mu \cdot q_{ij}[amp] \quad (5.16)$$

gdzie $\mu \in (0, 1)$ to **współczynnik kontrakcji amplitud** (parametr algorytmu). Amplituda związana z parą bitów, figurującą na pozycji j w najlepszym osobniku b , modyfikowana jest natomiast w taki sposób, by zachować

Listing 7: Aktualizacja stanu genów kwantowych w QIGA2

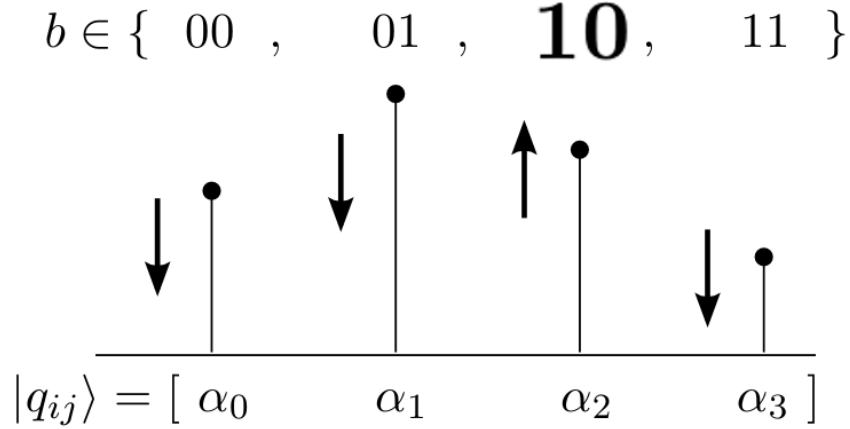
```

1 begin
2   for i in 0, ..., |Q| - 1 do
3     for j in 0, ..., N/2 do
4        $q' = [0 \ 0 \ 0 \ 0]^T$ 
5       bestamp  $\leftarrow$  j-ta para bitów w b, zapisana dziesiętnie
6       sum  $\leftarrow$  0
7       for amp in {0, 1, 2, 3} do
8         if amp  $\neq$  bestamp then
9            $q'[amp] \leftarrow \mu \cdot q_{ij}$ 
10          sum  $\leftarrow$  sum +  $(q'[amp])^2$ 
11        end
12      end
13       $q'[bestamp] \leftarrow \sqrt{1 - \text{sum}}$ 
14       $q_{ij} \leftarrow q'$ 
15    end
16  end
17 end

```

wany był warunek normalizacyjny wektora stanu (tzn. jednostkowa suma prawdopodobieństw $\sum_{amp=0}^3 |\alpha_{amp}|^2 = 1$). Na podstawie przeprowadzonych przez autora prób ustalono, że wysoka efektywność algorytmu uzyskiwana jest dla wartości współczynnika kontrakcji $\mu \approx 0.99$.

Działanie nowego operatora zilustrowane jest na [Rysunku 5.13](#). Pionowe słupki na osi reprezentują amplitudy prawdopodobieństw $|\alpha_0|^2, |\alpha_1|^2, |\alpha_2|^2, |\alpha_3|^2$. Jeżeli na pozycji $j \in \{0, 1, \dots, \frac{N}{2}\}$ osobnika b znajdowała się para bitów 10, to wszystkie amplitudy ulegają kontrakcji o współczynnik μ , za wyjątkiem amplitudy α_2 , która zostanie zwiększona. Jeżeli na pozycji j osobnika b znajdowała się para bitów 00, to wszystkie amplitudy ulegają kontrakcji o współczynnik μ , za wyjątkiem amplitudy α_0 , która zostanie zwiększona, itd. Zwiększeniu ulega zatem tylko amplituda o indeksie odpowiadającym j-tej parze bitów w b, zapisanej w postaci dziesiętnej.



Rys. 5.13: Działanie nowego operatora na rejestr kwantowy kodujący parę genów. Jeżeli w najlepszym znalezionym dotąd osobniku b j -ta para bitów to 10, to zwiększeniu ulega amplituda α_2 . itd.

Implementację algorytmu QIGA2 w języku C++ zaprezentowano w [Załączniku C](#). Za niewątpliwą zaletę algorytmu QIGA2 należy uznać jego prostotę zarówno w stosunku do pierwotnego algorytmu QIGA1, jak i w szczególności do jego wielu późniejszych zmodyfikowanych wariantów, których autorzy również podejmowali próby poprawienia efektywności algorytmu pierwotnego. Należy zwrócić uwagę, że w algorytmie QIGA2 zostało całkowicie wyeliminowane użycie tablic Lookup Table ([Tabela 3.2 i 3.3 na str. 63](#)), występujących w pierwotnym algorytmie QIGA1. Dzięki uproszczeniu algorytmu i w szczególności dzięki wyeliminowaniu tablic Lookup Table, również szybkość działania opracowanej implementacji algorytmu QIGA2 jest ok. **15-30%** większa niż algorytmu QIGA1 (porównanie implementacji zrealizowanych w tych samych językach programowania, z użyciem tych samych wersji kompilatorów i platform sprzętowych). Szczegółowe wyniki badań doświadczalnych zostały zaprezentowane w kolejnym rozdziale pracy.

5.4 PODSUMOWANIE I WNIOSKI

Zaprezentowaną w rozdziale analizę rozpoczęto od zbadania własności przestrzeni działania kwantowo inspirowanych algorytmów genetycznych w porównaniu z wybranymi algorytmami przeszukiwania, w tym z prostym algorytmem genetycznym, działającym w przestrzeni schematów. Pozwoliło to na poznanie związków pomiędzy binarnymi chromosomami kwantowymi i schematami oraz na stwierdzenie, że kwantowo inspirowany algorytm genetyczny ocenia w sposób stochastyczny wartość dopasowania schematów (wzór 5.3). Na tle przykładowej jednowymiarowej funkcji celu zilustrowane zostały wybrane rozkłady prawdopodobieństw, kodowane w genotypach kwantowych algorytmów różnych rzędów.

W wyniku tej analizy sformułowane zostały nowe pojęcia: rząd kwantowości r , współczynnik kwantowości λ oraz współczynnik kwantowości operatorowej λ_O . Zbadane zostały podstawowe związki pomiędzy tymi wielkościami w kwantowo inspirowanych algorytmach genetycznych. Zaproponowany w rozdziale współczynnik kwantowości $\lambda \in [0, 1]$ pozwala na scharakteryzowanie zarówno analizowanych algorytmów kwantowo inspirowanych (dla których $\lambda \gtrsim 0$), jak i „prawdziwych” algorytmów kwantowych ($\lambda = 1$), czyli algorytmów w których rząd kwantowości jest równy wielkości zadania $r = N$. Wraz ze wzrostem rzędu kwantowości algorytmu rośnie wymiar przestrzeni genotypów kwantowych w algorytmie oraz wymiar przestrzeni operatorów, a tym samym wraz ze wzrostem współczynnika kwantowości ($\lambda \rightarrow 1^-$) koszt symulacji elementu kwantowego w algorytmie przybiera tempo wykładnicze, co wynika z **Własności 2.9** na **str. 38**.

W oparciu o wprowadzone pojęcia teoretyczne, opracowany został kwantowo inspirowany algorytm genetyczny rzędu II, wykorzystujący 2-kubitowe rejestry kwantowe do kodowania rozkładów prawdopodobieństw przyjmowania możliwych wartości przez sąsiadujące ze sobą pary bitów w chromosomach. Zaproponowany został również nowy operator genetyczny, polegający na odpowiedniej modyfikacji stanów 2-kubitowych rejestrów w genotypach kwantowych.

Wynikiem przeprowadzonych w tym rozdziale badań teoretycznych jest wniosek, iż algorytmy kwantowo inspirowane wyższych rzędów pozwalają modelować zależności zachodzące między zbiorami genów w chromosomie kwantowym, co umożliwia poprawę efektywności algorytmów w szczególności dla zadań zwodniczych. Doświadczalny dowód tego stwierdzenia, wynikającego z przeprowadzonych w rozdziale rozważań teoretycznych, zaprezentowano w kolejnym rozdziale.

Wprowadzone nowe pojęcie rzędu kwantowości niesie ze sobą również interesującą interpretację fizyczną, wykraczającą poza obszar metod ewolucyjnych².

² Istnieją hipotezy (Penrose, Hameroff), według których niealgorytmiczny i niedeterministyczny sposób działania ludzkiego umysłu wynika ze zjawisk występujących na poziomie kwantowym w mózgu, które są nieodzwone do zrozumienia, wyjaśnienia i właściwego zinterpretowania sposobu działania umysłu i świadomości. Zgodnie z tą hipotezą można zatem przyjąć, że stan umysłu oraz sposób jego działania wymagałby modelowania z dokładnością na poziomie kwantowym. Górną granicę liczby informacji, niezbędnej do opisanie dowolnego układu fizycznego o danej masie i objętości, jest tzw. granica Bekensteina (Bekenstein, 1981). Granica Bekensteina podaje, że do pełnego opisu stanu materii w ludzkim mózgu byłoby potrzebne co najwyżej $2 \cdot 10^{42}$ bitów danych – jeśli cały mózg byłby traktowany jako pojedynczy układ kwantowy, mogący być w superpozycji lub splątaniu dowolnego z możliwych stanów całego mózgu (odpowiada to kwantowości „pełnego rzędu” w rozumieniu wprowadzonej teorii). Takie modelowanie całego mózgu, tj. obiektu makroskopowego, jako pojedynczego układu kwantowego (pełny rząd kwantowości) jest jednak wielce nieadekwatne z fizycznego punktu widzenia. Jako dolną granicę można natomiast przyjąć wymóg występowania wielu pojedynczych elementów kwantowych (np. niezależne 2-poziomowe układy kwantowe – według hipotez Penrose’a i Hameroff’a mogące występować w mikrotubulach w mózgu), dla zapewnienia elementu „prawdziwego” (kwantowego) niedeterminizmu jego działania (odpowiadałoby to rzędowi $r = 1$ w rozumieniu wprowadzonych pojęć). Kontynuując takie rozumowanie i uogólniając odpowiednio wprowadzone nowe pojęcia do zakresu szerszego niż tylko algorytmy genetyczne, można byłoby zadać pytanie, jakiego rzędu kwantowości jest hipotetyczny układ, mogący być jakościowym i funkcjonalnym odpowiednikiem ludzkiego mózgu – czy niezbędny jest sam element „prawdziwej” kwantowej losowości ($r = 1$, wiele niezależnych kubitów), pary kubitów ($r = 2$, zapewniające efekt kwantowego splątania) czy występowanie czynników kwantowych wyższych rzędów ($r > 2$).

Część III

BADANIA DOŚWIADCZALNE

Rozdział 6

ALGORYTMY QIGA DLA ZADAŃ OPTYMALIZACJI KOMBINATORYCZNEJ

Dla zbadania efektywności kwantowo inspirowanego algorytmu genetycznego rzędu II (QIGA2), przedstawionego w poprzednim rozdziale, przeprowadzono badania doświadczalne, polegające na porównaniu algorytmu z algorytmem QIGA1 w jego wersji pierwotnej jak i w wersji ulepszonej poprzez dobór wartości parametrów z wykorzystaniem techniki metaoptymalizacji.

6.1 PROCEDURA BADAWCZA

Badania przeprowadzone zostały według następującego schematu:

1. Przygotowano zestaw 20 zadań testowych optymalizacji kombinatorycznej, oparty na problemie plecakowym oraz problemie spełnialności (SAT).
2. Ze względu wysoki koszt obliczeniowy procesu metaoptymalizacji, zrealizowano zarówno klasyczną szeregową implementację algorytmu QIGA1 na jednostkach CPU (implementacja referencyjna), a następnie implementację masowo równoległą, wykorzystującą jednostki GPU oraz oryginalny sposób zrównoleglenia algorytmu, zaproponowany w (Nowotniak, Kucharski, 2011b).
3. Wykorzystując pojedyncze zadanie problemu plecakowego, przeprowadzono optymalizację wartości wybranych parametrów pierwotnego algorytmu QIGA1 (wartości kątów obrotów w przestrzeni stanów genów kwantowych), co pozwoliło na poprawę efektywności algorytmu.

4. Wykorzystując pełen zestaw zróżnicowanych 20 zadań testowych (problem plecakowy oraz problem spełnialności), wykazano przewagę algorytmu dostrojonego QIGA1* nad algorytmem pierwotnym QIGA1 oraz przewagę algorytmu autorskiego QIGA2 nad pozostałymi algorytmami.

METODYKA BADAŃ DOŚWIADCZALNYCH

W literaturze tematu zdecydowana większość publikacji, dotyczących algorytmów ewolucyjnych, prezentuje wyniki uzyskane na drodze badań doświadczalnych (eksperymenty obliczeniowe z różnymi wariantami algorytmów). Prace te opierają się na założeniu, że użyteczność nowej metody heurystycznej może polegać na ([Michalewicz, 2003](#), str. 399), ([Michalewicz, Fogel, 2006](#), str. 586):

1. tworzeniu wysokiej jakości rozwiązań **szybciej** niż inne metody;
2. wyznaczaniu rozwiązań o **lepszej jakości** niż inne metody.

Ocena skuteczności i efektywności algorytmu dokonywana jest więc zazwyczaj w dwóch zasadniczych kontekstach (odpowiadających bezpośrednio powyższym punktom):

1. Przy założeniu, że dostępne są ograniczone i z góry określone zasoby maszyny obliczeniowej (np. ograniczony dostępny czas obliczeń na procesorze CPU, podczas którego możliwe jest wykonanie ograniczonej liczby ewaluacji rozwiązań), zasadniczym pytaniem jest, jaki algorytm i o jakich parametrach pozwoli na znalezienie najlepszego rozwiązania.
2. Dla danego problemu o charakterze przeszukiwania lub optymalizacji należy w krótkim czasie (tj. po stosunkowo małej liczbie ewaluacji funkcji celu) znaleźć zadowalające rozwiązanie, które może być dalej poprawiane w kolejnych generacjach algorytmu. Tego typu sytuacja występuje np. w grach komputerowych, gdy potrzebne jest szybkie podjęcie decyzji (niekoniecznie najlepszej możliwej) przez komputer.

6.2 ZESTAWY ZADAŃ TESTOWYCH

Na potrzeby badań doświadczalnych wykorzystano dwa następujące, dobrze znane zadania optymalizacji kombinatorycznej – **problem plecakowy** (ang. *knapsack problem*), wykorzystywany w licznych dotychczasowych pracach dotyczących algorytmów QIGA, oraz **problem spełnialności** (ang. *satisfiability problem* – SAT). Zadania testowe SAT zostały zaczerpnięte z uznanego benchmarku SATLIB. Zaletą problemu SAT jest jego uniwersalność (NP-zupełność), tzn. każdy problem z klasy NP może być do niego zredukowany. Wykorzystanie uznanych benchmarków, obejmujących wiele zadań kombinatorycznych o różnych własnościach i rozmiarach, pozwala na wyciągnięcie ogólnych wniosków na temat efektywności badanych algorytmów. Opis poszczególnych zadań testowych, należących do benchmarku SATLIB, oraz ich własności można znaleźć w pracy (Hoos, Stützle, 2000).

Problem plecakowy

Parametry problemu plecakowego wygenerowane zostały w sposób losowy, dla różnych rozmiarów zadań (ilość elementów w plecaku) $N \in \{100, 250, \dots, 1000\}$. Wygenerowano silnie skorelowane ze sobą wartości i wagi elementów, co tworzy tzw. trudną wersję zadania, w której elementy o wysokiej wartości mają dużą wagę:

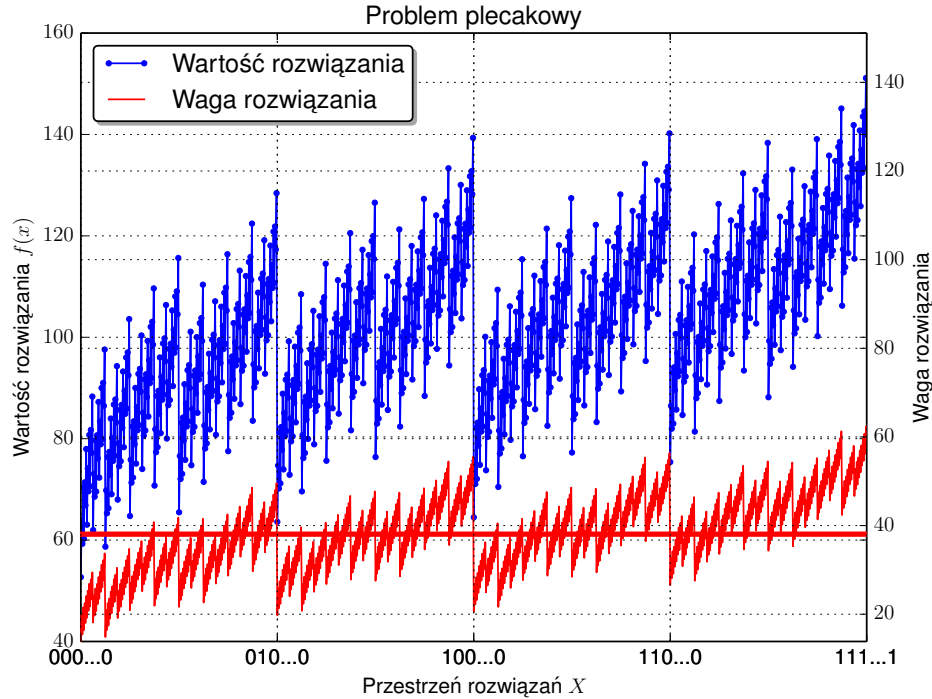
$$\begin{cases} w_i = \text{random}[1, 10) \\ p_i = w_i + 5 \end{cases} \quad (6.1)$$

gdzie w_i oznacza wagę i -tego elementu, a p_i oznacza wartość tego elementu. Ocena danego rozwiązania $f(x)$ wyznaczana jest za pomocą wzoru:

$$f(x) = \sum_{i=1}^N p_i x_i \quad (6.2)$$

gdzie x_i to i -ty bit w ciągu binarnym x . Dopuszczalna waga elementów (pojemność plecaka) ustalona została za pomocą wzoru:

$$C = \frac{1}{2} \sum_{i=1}^N w_i \quad (6.3)$$



RYS. 6.1: Wykres przestrzeni rozwiązań dla jednego z zadań testowych (problem plecakowy)

Celem zadania jest znalezienie rozwiązania x o maksymalnej wartości oceny $f(x)$, spełniającego jednak ograniczenie

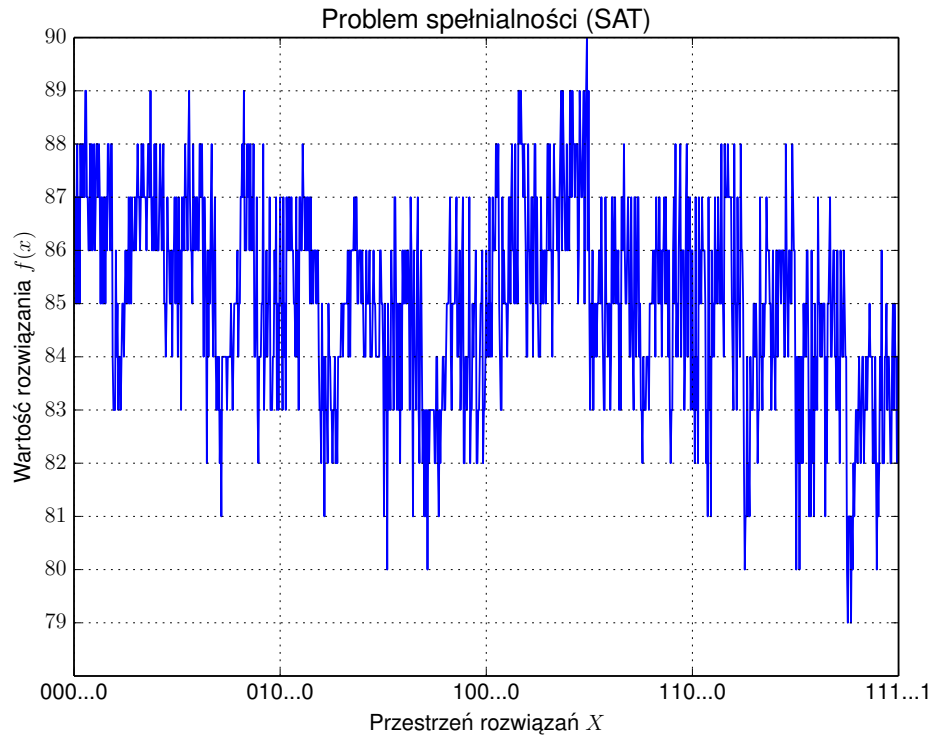
$$\sum_{i=1}^N w_i x_i \leq C \quad (6.4)$$

Jeśli dane rozwiązanie x nie spełnia ograniczenia (suma wag elementów przekracza dopuszczalną wagę C), stosowana jest następująca **procedura naprawcza**: Kolejne elementy $i \in \{1, \dots, N\}$ usuwane są z rozwiązania x aż do momentu, gdy ograniczenie zostaje spełnione, a następnie, jeśli jest to możliwe, kolejne elementy są dodawane do rozwiązania x dopóki ograniczenie jest spełnione.

Na **Rysunku 6.1** przedstawiony został wykres, prezentujący przybliżone ukształtowanie funkcji celu w przestrzeni rozwiązań dla jednego z zadań testowych problemu plecakowego, wykorzystanego w badaniach. Kolorem niebieskim na wykresie oznaczona została wartość danego rozwiązania $f(x)$ (oś pionowa po lewej stronie wykresu), kolorem czerwonym

oznaczona została waga danego rozwiązania (oś pionowa po lewej stronie wykresu), a pozioma czerwona linia oznacza dopuszczalną wagę rozwiązania (ograniczenie powyżej którego stosowana jest procedura naprawcza).

Problem spełnialności



Rys. 6.2: Wykres przestrzeni rozwiązań dla jednego z zadań testowych (zadanie spełnialności (SAT))

W przypadku problemu spełnialności wykorzystano zadania zaczerpnięte z benchmarku SATLIB o wymiarowościach $N \in \{48, \dots, 1000\}$. Rozmiar zadania N odpowiada liczbie literałów w danym zadaniu testowym, zaś wartość dopasowania $f(x)$ definiowana jest jako liczba spełnionych klauzul dla wartości literałów określonych przez kolejne bity w ciągu $x_1, x_2, \dots, x_N$. Przybliżone ukształtowanie funkcji celu jednego z wykorzystanych wariantów zadania spełnialności zostało zaprezentowane na [Rysunku 6.2](#). Rozmiar zilustrowanego na rysunku zadania to $N = 20$ zmiennych oraz 90 klauzul. Jak widać na wykresie, tylko jeden ciąg zmiennych binarnych

x^* spełnia wszystkie 90 klauzul, istnieje jednak kilka optimów lokalnych, spełniających suboptymalną liczbę klauzul.

Konieczność przybliżenia ukształtowania na wykresach wynika z mocy przestrzeni rozwiązań dla danego rozmiaru zadania N . Dla najmniejszego badanego rozmiaru $N = 48$ przestrzeń rozwiązań składa się z 2^{48} (ponad 10^{14}) możliwych ciągów x . Cechą charakterystyczną wykresów na [Rysunkach 6.1 i 6.2](#) są silne nieregularności, nieciągłość wartości $f(x)$ w przestrzeni ciągów binarnych X oraz liczne odizolowane optima lokalne. Te cechy zadań testowych wpływają na ich zwodniczość, co utrudnia rozwiązywanie tych zadań przez prosty algorytm genetyczny.

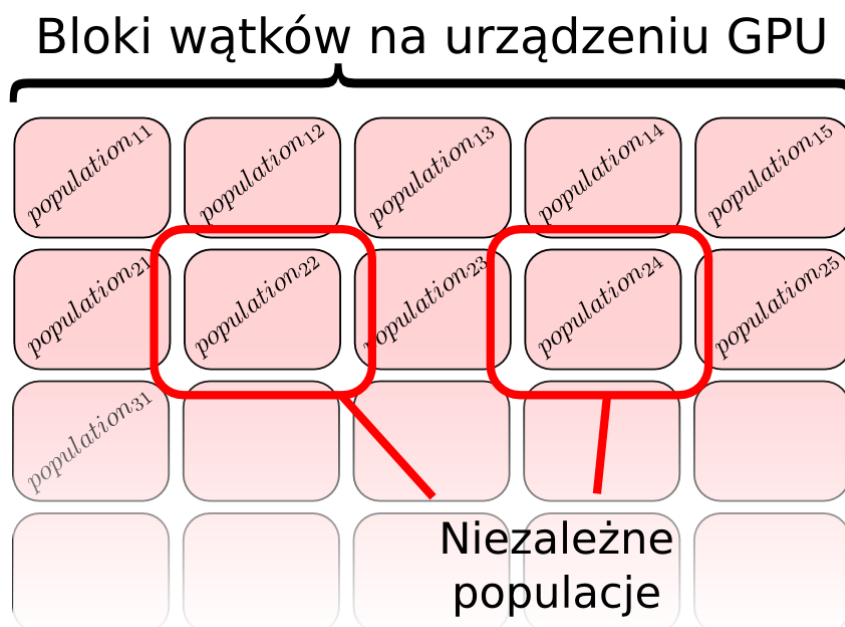
6.3 EFEKTYWNA IMPLEMENTACJA ALGORYTMU QIGA1 W TECHNOLOGII GPGPU

Ze względu na znaczny koszt obliczeniowy wielokrotnie powtarzanych i czasochłonnych eksperymentów numerycznych z algorytmami QIGA, niezbędne było opracowanie wysoce efektywnej implementacji algorytmu. W tym celu algorytm QIGA1 zaimplementowano na jednostkach GPU (ang. *Graphics Processing Unit*) z wykorzystaniem technologii NVidia CUDA™. Technologia ta pozwala na wykorzystanie procesorów kart graficznych do realizacji masowo równoległych obliczeń ogólnego przeznaczenia GPGPU (ang. *General Purpose Computation on Graphics Processors*). Podstawowe pojęcia w tej technologii to **wątek obliczeniowy CUDA** (ang. *CUDA compute thread*), **blok** (ang. *block of threads*) oraz **krata** (ang. *grid*) ([Nvidia, 2011](#)).

Zaproponowany został oryginalny sposób zrównoleglenia algorytmu ewolucyjnego oraz samej procedury badawczej ([Nowotniak, Kucharski, 2011b](#)). Zrównoleglenie zostało zrealizowane na dwóch poziomach, co zilustrowano na [Rysunku 6.3](#):

1. **Zrównoleglenie algorytmu** – Każdy wątek w bloku przetwarza innego osobnika lub inny gen kwantowy.
2. **Zrównoleglenie procedury badawczej** – W każdym bloku prowadzona jest ewolucja populacji kwantowej przez algorytm o innych wartościach parametrów.

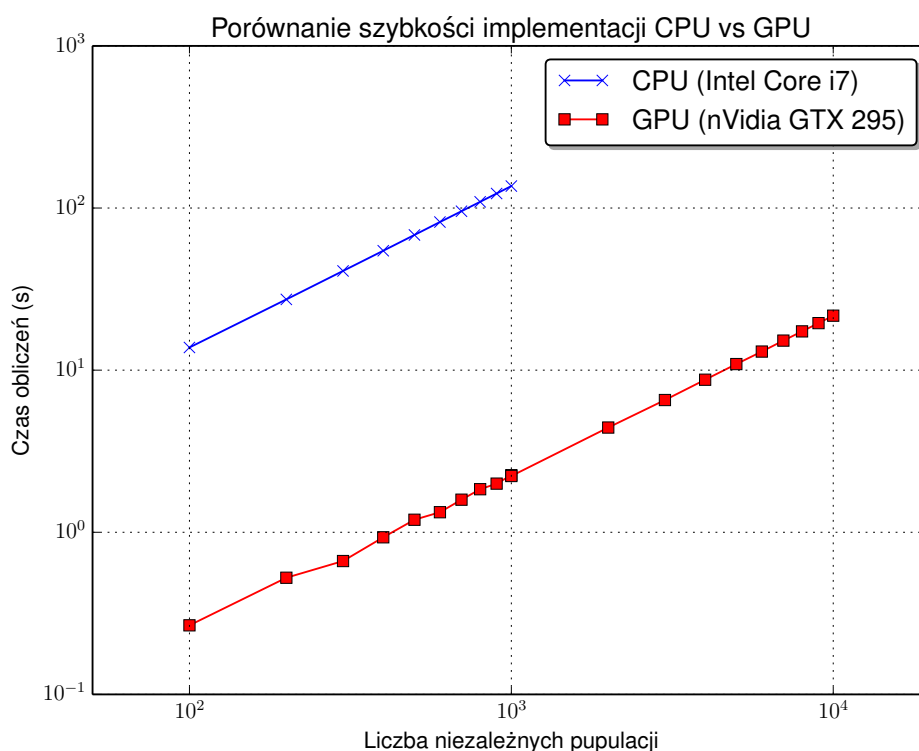
Realizacja zaproponowanego rozwiązania wymagała implementacji całego algorytmu QIGA1 jako jądra obliczeniowego CUDA (ang. *CUDA computational kernel*), działającego bezpośrednio na urządzeniu GPU. Wymagało to rozwiązania szeregu problemów, w tym wykonania odpowiedniego partycjonowania struktur danych w celu przenoszenia ich części do szybkiej pamięci dzielonej multiprocesorów GPU, oraz zapewnienia właściwej synchronizacji tysięcy wątków CUDA, działających jednocześnie na urządzeniu GPU. Szczegóły implementacyjne tego rozwiązania przedstawione zostały w pracach (Nowotniak, Kucharski, 2011b), (Nowotniak, Kucharski, 2012b).



Rys. 6.3: Autorski sposób zrównoleglenia obliczeń na urządzeniu GPU

Porównanie szybkości działania implementacji referencyjnej na pojedynczym rdzeniu CPU Intel Core i7 oraz autorskiej implementacji na jednym urządzeniu NVidia GTX 295 przedstawiono **Rysunku 6.4**. Oś odciętych na wykresie przedstawia liczbę niezależnych populacji, które przetwarzane są przez zaimplementowany algorytm ewolucyjny, natomiast na osi rzędnych przedstawiony jest czas w sekundach, w którym poszczególne implementacje algorytmu ukończą przetwarzanie danej liczby populacji. Na obydwu osiach wykresu zastosowana została skala logarytmiczna. Opraco-

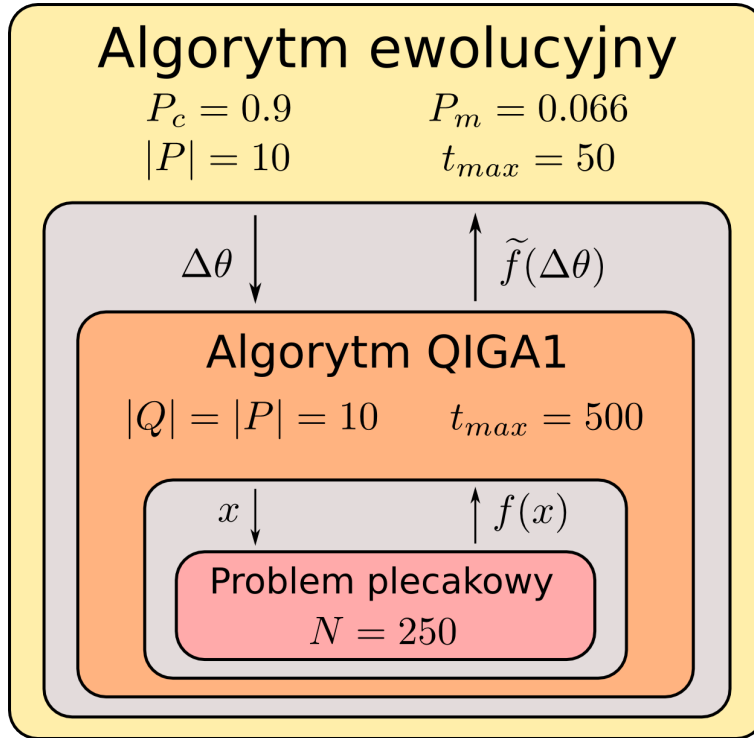
wana implementacja masowo równoległa uruchamiana była docelowo na ośmiu urządzeniach GPU (4 x Tesla T10 GPU, GTX 285, GTX 295 (karta podwójna) oraz Tesla C2070). Zaproponowany sposób rozproszenia obliczeń pozwolił na uzyskanie ponad 400-krotnego przyspieszenia algorytmu QIGA1 w stosunku do obliczeń na pojedynczym rdzeniu procesora CPU Intel Core i7 2.93 GHz.



RYS. 6.4: Porównanie szybkość działania implementacji (implementacja referencyjna na Intel i7 CPU vs pojedyncza karta GPU GTX 295)

6.4 METAOPTYMALIZACJA ALGORYTMU QIGA1

Opracowanie wysoce wydajnej implementacji algorytmu QIGA1 na jednostkach GPU pozwoliło w kolejnym kroku na dobór wartości wybranych parametrów algorytmu poprzez zastosowanie techniki metaoptymalizacji. Sposób realizacji procesu metaoptymalizacji oraz najważniejsze parametry algorytmów zostały przedstawione na **Rysunku 6.5** (por. **Rysunek 1.3** na str. 29).



RYS. 6.5: Zastosowane podejście do metaoptymalizacji

Metaoptymalizację algorytmu QIGA1 wykonano wykorzystując pojedyncze zadanie testowe problemu plecakowego o rozmiarze $N = 250$ zmiennych binarnych. Parametry zadania wygenerowane były w sposób losowy, z wykorzystaniem wzoru (6.1). Jako maksymalną liczbę wywołań funkcji oceny $f(x)$ w pojedynczym uruchomieniu każdego z algorytmów przyjęto wartość $\text{MaxFE} = 5000$. Wykorzystanie tylko jednego zadania w tej części eksperymentu było celowe, aby w kolejnych etapach badań móc ocenić, czy wartości parametrów, znajdowane tą metodą, poprawiają

efektywność algorytmu niezależnie od wyboru zadań i czy pozwala to na wyciągnięcie ogólnych wniosków z przeprowadzonych doświadczeń.

Wartości wszystkich parametrów pierwotnego algorytmu QIGA1 ustalone zostały na wartości oryginalne, zaczerpnięte z pracy (Han, Kim, 2000). Rozmiar populacji kwantowej i klasycznej wynosił $|P| = |Q| = 10$. Maksymalna liczba generacji wynikała z przyjętego wcześniej ograniczenia na maksymalną liczbę wywołań funkcji oceny: $t_{\max} = \text{MaxFE}/|P| = 5000/10 = 500$ generacji. Implementacja algorytmu na urządzeniu GPU uruchamiana była na kracie składającej się z 50 bloków, a w każdym bloku uruchamiane było 250 wątków CUDA, przetwarzających niezależnie kolejne geny w chromosomie (co odpowiada rozmiarowi zadania $N = 250$). Na każdym urządzeniu GPU było więc uruchamiane w sposób równoległy $50 \cdot 250 = 12500$ wątków CUDA.

x_i	b_i	$f(x) \geq f(b)$	$\Delta\theta_i$	Wartości podlegające metaoptymalizacji
0	0	Fałsz	0	
0	0	Prawda	0	
0	1	Fałsz	0	
0	1	Prawda	0.05π	
1	0	Fałsz	0.01π	
1	0	Prawda	0.025π	
1	1	Fałsz	0.005π	
1	1	Prawda	0.025π	

TABELA 6.1: Tablica *Lookup table* kątów obrotów w algorytmie QIGA1

Optymalizacji podlegały wartości kątów obrotów $\Delta\theta$, które są kluczowymi parametrami dla używanych w algorytmie QIGA1 kwantowych operatorów genetycznych (Rozdział 3.4.1, str. 60). Optymalizację przeprowadzono w przestrzeni $\mathbb{R}^5$, a oryginalne wartości kątów, zaczerpnięte z (Han, Kim, 2000), zostały przedstawione w Tabeli 6.1. W Tabeli 6.2 przedstawiono odpowiadającą tym kątom średnią wartość rozwiązania, znajdowanego podczas $t_{\max} = 500$ generacji algorytmu QIGA1.

$\Delta\theta$					Końcowa wartość funkcji dopasowania
0.05π	0.01π	0.025π	0.005π	0.025π	1408.25
0.157	0.031	0.079	0.016	0.079	
9°	1.8°	4.5°	0.9°	4.5°	

TABELA 6.2: Oryginalne wartości kątów obrotów podane w radianach (wiersz 1-2 tabeli) i w stopniach (wiersz 3), zaczerpnięte z (Han, Kim, 2000), oraz końcowa wartość funkcji dopasowania dla zadania testowego i algorytmu QIGA1 o danych parametrach.

Jako algorytm metaoptymalizacji zastosowano algorytm ewolucyjny z kodowaniem rzeczywistym, którego implementację zaczerpnięto z biblioteki PyEvolve (Perone, 2009) dla języka Python. W algorytmie na metapoziomie zastosowano krzyżowanie jednopunktowe z prawdopodobieństwem $P_c = 0.9$ oraz mutację Gaussowską z prawdopodobieństwem $P_m = 0.066$ oraz z odchyleniem standardowym $\sigma = 5^\circ$. Jako kryterium stopu na metapoziomie przyjęto ograniczenie maksymalnej liczby generacji $t_{\max} = 50$, rozmiar populacji ustalono na 10, tzn. każdy osobnik w algorytmie ewolucyjnym, realizującym metaoptymalizację, składał się z pięciu wartości rzeczywistych $\Delta\theta_i \in [0^\circ; 20^\circ]$.

Rozważono dwie postaci funkcji metadopasowania:

- $\tilde{f}_1(\Delta\theta) : [0^\circ, 20^\circ]^5 \mapsto \mathbb{R}^+$ – wartość oceny najlepszego osobnika w ostatniej generacji algorytmu QIGA1 (uśredniona dla 50 powtórzeń algorytmu). Celem była *maksymalizacja* tej wartości.
- $\tilde{f}_2(\Delta\theta) : [0^\circ, 20^\circ]^5 \mapsto \mathbb{R}^+$ – liczbę wywołań funkcji dopasowania (tj. liczba ewaluacji rozwiązań x , proporcjonalna do numeru generacji t), po których zostaje znalezione rozwiązanie o określonej wartości dopasowania (wartość uśredniona dla 50 powtórzeń algorytmu). Celem była *minimalizacja* tej wartości.

Definicja funkcji $\tilde{f}_1$ miała na celu znalezienie parametrów $\Delta\theta$, zapewniających maksymalną wartość rozwiązania w ostatniej iteracji algorytmu ($t = t_{\max}$), zaś definicja funkcji $\tilde{f}_2$ miała na celu znalezienie parametrów, które powodują jednocześnie szybką zbieżność algorytmu do rozwiązań o wysokich wartościach dopasowania.

Proces metaoptymalizacji wykonywany był najpierw dla funkcji meta-dopasowania $\tilde{f}_1(\Delta\theta)$, a następnie $\tilde{f}_2(\Delta\theta)$, dzięki czemu możliwe było znalezienie wartości parametrów $\Delta\theta$, które zapewniają jednocześnie wysoką końcową wartość oceny rozwiązań $f(x)$, jak i szybką zbieżność algorytmu QIGA1 do rozwiązań wysoko ocenianych. W Tabeli 6.4 przedstawiono 20 najlepszych znalezionych tą metodą zestawów parametrów oraz odpowiadające im końcowe wartości funkcji dopasowania $f(x)$ dla zadania testowego. W Tabeli 6.3 przedstawiono natomiast wartości średnie, mediany oraz odchylenia standardowe dla tych zestawów parametrów.

Ponieważ średnia wartość rozwiązania, znajdowanego przez pierwotny algorytm QIGA1, wynosi 1408.25 (Tabela 6.2), wartości parametrów $\Delta\theta$, przedstawione w Tabeli 6.4, wskazują na poprawę efektywności w stosunku do algorytmu pierwotnego.

Najlepsze znalezione wartości kątów obrotów to:

$$\Delta\theta = [0.000, 0.044, 0.223, 0.254, 0.151] \quad (6.5)$$

Średnią wartością najlepszego rozwiązania zadania testowego, znalezione-go w ostatniej generacji algorytmu o tych parametrach, jest $f(x) = 1462.65$. Algorytm o tych wartościach parametrów będzie oznaczany dalej **QIGA1***.

$\Delta\theta$					
0.108	0.042	0.193	0.273	0.160	Średnia
0.115	0.041	0.233	0.262	0.152	Mediana
0.075	0.006	0.096	0.042	0.094	Odchylenie std.

TABELA 6.3: Podsumowanie wyników z Tabeli 6.4

$\Delta\theta$					Wartość funkcji dopasowania
0.000	0.044	0.223	0.254	0.151	1462.65
0.128	0.042	0.262	0.262	0.200	1462.45
0.159	0.043	0.246	0.262	0.251	1462.28
0.056	0.040	0.235	0.262	0.244	1462.04
0.037	0.042	0.012	0.262	0.262	1461.55
0.193	0.049	0.256	0.262	0.181	1461.14
0.248	0.039	0.080	0.262	0.140	1460.18
0.067	0.041	0.247	0.262	0.063	1459.92
0.113	0.038	0.128	0.262	0.205	1459.48
0.000	0.038	0.349	0.334	0.349	1458.86
0.117	0.036	0.349	0.349	0.000	1458.79
0.022	0.054	0.076	0.262	0.132	1458.46
0.125	0.048	0.060	0.212	0.023	1458.40
0.063	0.038	0.239	0.326	0.320	1458.14
0.157	0.034	0.256	0.349	0.081	1456.82
0.065	0.051	0.035	0.242	0.135	1456.53
0.146	0.053	0.246	0.202	0.154	1456.44
0.037	0.034	0.141	0.262	0.000	1456.31
0.281	0.032	0.206	0.348	0.137	1456.09
0.156	0.055	0.232	0.231	0.178	1455.35

TABELA 6.4: Najlepsze znalezione wartości kątów $\Delta\theta$ dla algorytmu QIGA1 oraz odpowiadające im wartości oceny najlepszego rozwiązania znajdowanego przez algorytm o danych parametrach

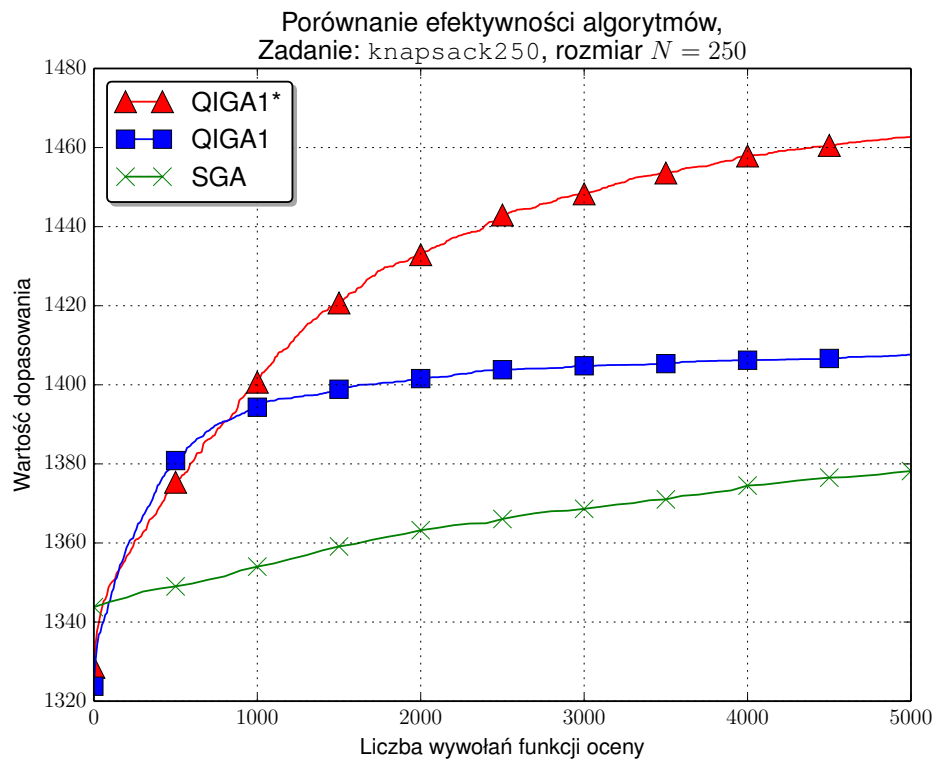
6.5 OCENA EFEKTYWNOŚCI ALGORYTMU QIGA1*

Kolejnym etapem badań było porównanie efektywności algorytmu pierwotnego (QIGA1), algorytmu dostrojonego (QIGA1*) oraz prostego algorytmu genetycznego (SGA). Wykres na [Rysunku 6.6](#) prezentuje porównanie efektywności tych algorytmów dla wybranego zadania testowego. Wykorzystanie liczby wywołań funkcji oceny zamiast numeru generacji na osi odciętych pozwala na porównanie różnych algorytmów na jednym wykresie, niezależnie od rozmiaru populacji w każdym z algorytmów.

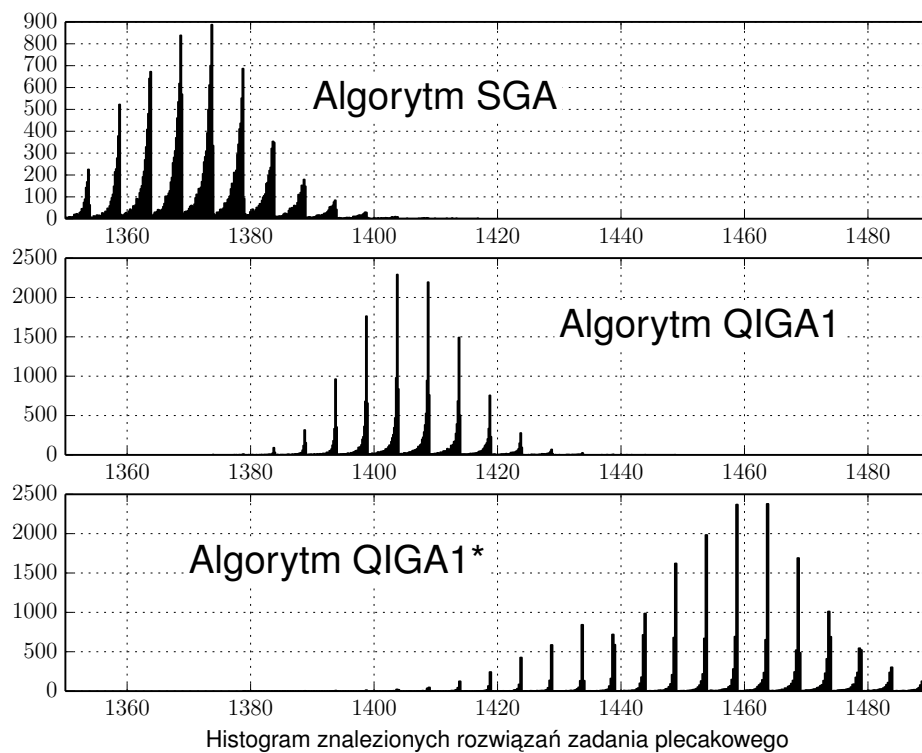
Algorytm SGA uruchamiany był z typowymi dla niego wartościami parametrów, zaczerpniętymi z ([Holland, 1975](#)) – selekcja proporcjonalna metodą koła ruletki, prawdopodobieństwo mutacji $P_m = 2\%$, prawdopodobieństwo krzyżowania $P_c = 90\%$, rozmiar populacji $|P| = 80$, kryterium stopu $\text{MaxFE} = 5000$ (co odpowiada $t_{\text{max}} = 5000/80 = 62$ generacjom). Implementacja algorytmu została zaczerpnięta z zewnętrznej biblioteki PyEvolve ([Perone, 2009](#)).

[Rysunek 6.7](#) przedstawia uzyskane wyniki również w postaci histogramów. Histogramy zostały wygenerowane na podstawie 30000-krotnego uruchomienia każdego z algorytmów SGA, QIGA1 oraz QIGA1*. Z porównania rozkładów na histogramach widać wyraźną przewagę (wartość średnia oraz mediana) dostrojonego algorytmu QIGA1* nad algorytmem pierwotnym oraz nad klasycznym algorytmem SGA dla rozważanego zadania testowego. Ponadto, na histogramach uwagę zwracać może charakterystyczny rozkład wartości. Jest on związany z dyskretną naturą problemu plecakowego oraz z zastosowaną procedurą naprawczą (analogicznie jak w ([Han, Kim, 2000](#)) oraz wielu późniejszych publikacjach wykorzystujących ten sam zestaw danych testowych). Widoczne na wykresie pionowe regularności występują co 5 jednostek na osi odciętych, co wynika ze sposobu generowania zestawów danych testowych w problemie plecakowym (wzór (6.1) na [str. 122](#)).

Wykazano tym samym, że algorytm QIGA1 o parametrach dobranych techniką metaoptymalizacji posiada lepszą efektywność niż pierwotny algorytm z ([Han, Kim, 2000](#)) (Teza II pracy).



RYS. 6.6: Porównanie efektywności algorytmów SGA, pierwotnego QIGA1, oraz algorytmu dostrojonego QIGA1* dla zadania testowego knapsack250, $N = 250$.



Rys. 6.7: Porównanie w postaci histogramów wyników uzyskiwanych przez poszczególne algorytmy dla zadania plecakowego

6.6 EFEKTYWNOŚĆ AUTORSKIEGO ALGORYTMU QIGA2

Dla pełnego zweryfikowania efektywności autorskiego algorytmu rzędu II (QIGA2) został użyty benchmark, składający się z szerokiego zestawu 20 uznawanych testowych zadań optymalizacji kombinatorycznej o różnych rozmiarach $N \in \{48, 90, \dots, 1000\}$. Cztery spośród tych zadań to losowo wygenerowane problemy plecakowe, pozostałe szesnaście zadań to zadania spełnialności SAT, kodujące różne problemy optymalizacji kombinatorycznej, zaczerpnięte z benchmarku SATLIB.

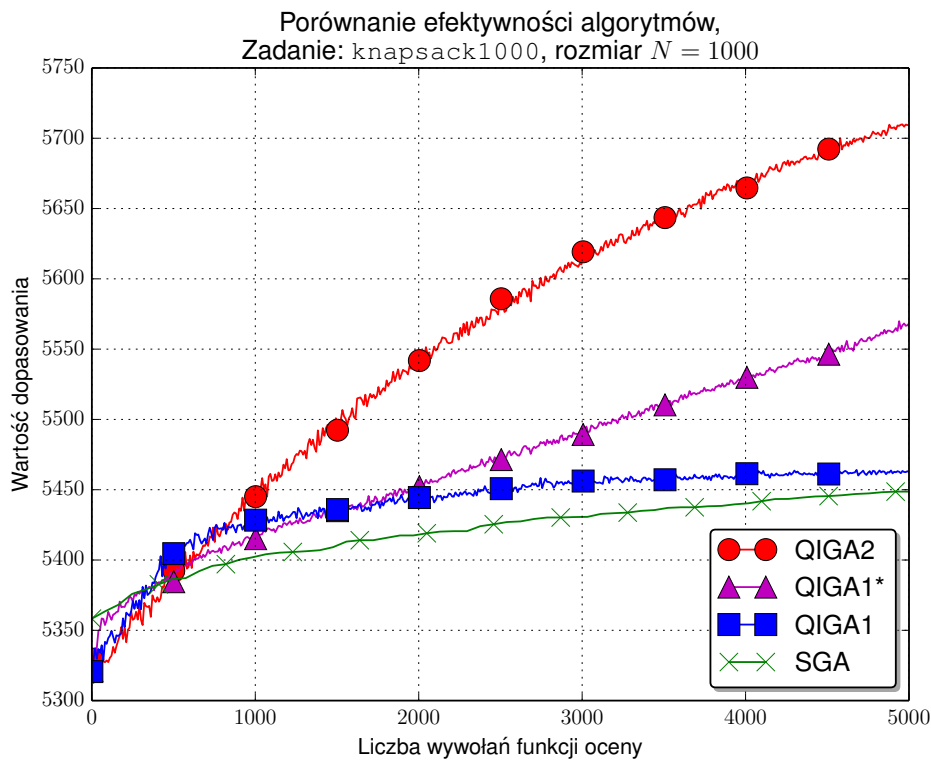
W porównaniu uwzględnione zostały algorytmy:

1. **algorytm SGA** – o parametrach typowych, takich jak w poprzednim podrozdziale ($P_c = 90\%$, $P_m = 2\%$, $|P| = 80$, selekcja proporcjonalna);
2. **pierwotny algorytm QIGA1** – ze wszystkimi oryginalnymi wartościami parametrów, zaczerpniętymi z (Han, Kim, 2000), rozmiar populacji kwantowej i klasycznej $|Q| = |P| = 10$;
3. **dostrojony algorytm QIGA1*** – ze zmodyfikowanymi wartościami kątów obrotów, znalezionymi w procesie metaoptymalizacji $\Delta\theta = [0.0, 0.044, 0.223, 0.254, 0.151]$;
4. **autorski algorytm QIGA2** – z wartością współczynnika kontrakcji amplitud $\mu = 0.9918$, rozmiar populacji kwantowej i klasycznej $|Q| = |P| = 10$ osobników.

Kryterium stopu dla każdego z algorytmów było uzyskanie liczby wywołań funkcji oceny $\text{MaxFE} = 5000$. Ponieważ w licznych dotychczasowych publikacjach wykazana została przewaga algorytmu QIGA1 nad innymi nowoczesnymi heurystycznymi metodami przeszukiwania (Rozdział 3.3), w badaniach pominięto porównanie z innymi algorytmami, uznając algorytm QIGA1 jako konkurencyjny w stosunku do innych nowoczesnych algorytmów.

Każdy algorytm uruchamiany był 50-krotnie dla każdego z zadań testowych. Jako miarę efektywności algorytmu przyjęto wartość dopasowania najlepszego znalezionej osobnika po spełnieniu kryterium stopu.

W Tabeli 6.5 przedstawiono wyniki uzyskane przez każdy z porównywanych algorytmów. Dla 17 spośród 20 zadań testowych (85%) zaproponowany algorytm QIGA2 uzyskiwał średnio lepsze rozwiązanie niż pierwotny oraz dostrojony algorytm QIGA1*. Tabela 6.6 przedstawia ranking porównywanych algorytmów, uszeregowanych według liczby zadań testowych, dla których dany algorytm uzyskiwał najlepszy wynik spośród wszystkich porównywanych algorytmów. Na Rysunku 6.8 przedstawiono szczegółowe porównanie efektywności algorytmów dla jednego z zadań testowych o rozmiarze $N = 1000$ zmiennych binarnych. Wykres ten przedstawia średnią wartość najlepszego rozwiązania, znalezionej przez każdy z algorytmów, w funkcji liczby wywołań funkcji oceny dopasowania osobnika. Wykresy, prezentujące porównania dla pozostałych zadań testowych, zostały zawarte w Załączniku D.



RYS. 6.8: Porównanie efektywności algorytmów dla wybranego zadania knapsack, rozmiar $N = 1000$

6.6. Efektywność autorskiego algorytmu QIGA2

Zadanie	N	SGA	QIGA1	QIGA1*	QIGA2
anomaly	48	251.4	252.55	254.65	255.25
sat	90	284.9	289.2	293.2	293.7
jnh	100	826.15	831.05	839.05	836.05
knapsack	100	577.709	578.812	592.819	596.476
sat	100	408.6	413.6	418.6	419.7
beijing	125	297.35	302.1	305.35	306.2
sat-uuf	225	886.75	898.25	921.65	921.5
random	250	1387.916	1406.528	1449.905	1467.407
sat1	250	981.45	995.15	1021.2	1023.1
sat2	250	982.95	994.6	1019.1	1020.6
sat3	250	984.2	994.3	1021.3	1019.7
beijing	252	709.85	731.0	724.4	745.75
parity	317	1141.65	1158.2	1179.35	1180.75
knapsack	400	2209.925	2222.160	2284.969	2334.494
knapsack	500	2803.266	2812.740	2869.774	2929.469
beijing	590	1263.8	1343.15	1284.0	1353.2
Iran	600	2310.9	2330.35	2386.8	2398.95
beijing	708	1510.65	1605.9	1523.15	1611.55
knapsack	1000	5451.656	5462.718	5568.234	5709.116
Iran	1000	3819.65	3848.4	3918.5	3937.3

TABELA 6.5: Porównanie efektywności algorytmów dla testowych zadań optymalizacji kombinatorycznej i różnych rozmiarów zadań N

lp	algorytm	najlepsze rozwiązania
1	QIGA2	17
2	QIGA1*	3
3	QIGA1	0
4	SGA	0

TABELA 6.6: Ranking porównawczy algorytmów

6.7 PODSUMOWANIE I WNIOSKI

W rozdziale zaprezentowana została skuteczna poprawa efektywności pierwotnego algorytmu QIGA1 w rozwiązywaniu zadań optymalizacji kombinatorycznej poprzez dobór wartości jego wybranych parametrów za pomocą techniki metaoptymalizacji. W celu realizacji procesu metaoptymalizacji zaproponowano autorski sposób zrównoleglenia algorytmu oraz samej procedury badawczej na urządzeniach GPU. Dzięki wykorzystaniu 8 jednostek GPU oraz zastosowaniu oryginalnej metody zrównoleglenia, uzyskano ok. 400-krotne przyspieszenie procedury badawczej w stosunku do implementacji na pojedynczym referencyjnym rdzeniu CPU. Wartości wybranych parametrów algorytmu QIGA1 zostały dobrane na podstawie metaoptymalizacji dla *pojedynczego zadania testowego* (problem plecakowy). Następnie wykazano doświadczalnie na *szerokim zestawie zadań testowych*, że dostrojony w ten sposób algorytm QIGA1* jest bardziej efektywny niż algorytm pierwotny QIGA1 oraz algorytm referencyjny SGA, a wykorzystanie implementacji w środowisku obliczeń równoległych oraz techniki metaoptymalizacji pozwala na dobór właściwych wartości parametrów algorytmu (Teza II pracy).

Zbadano następnie efektywność rozwiązywania zadań optymalizacji kombinatorycznej przez autorski algorytm QIGA2, opracowany w oparciu o wprowadzone w poprzednim rozdziale pojęcia kwantowo inspirowanych algorytmów wyższych rzędów. Na podstawie badań doświadczalnych, przeprowadzonych na zestawie 20 rozmaitych zwodniczych zadań testowych o różnej wymiarowości $N \in \{48, \dots, 1000\}$, wykazano, że autorski algorytm QIGA2 charakteryzuje się wyższą efektywnością niż pierwotny oraz dostrojony algorytm QIGA1*. Pokazuje to, iż wykorzystanie II rzędu kwantowości jest metodą poprawy efektywności kwantowo inspirowanego algorytmu genetycznego, co jest doświadczalnym dowodem Tezy I pracy.

Rozdział 7

ALGORYTMY QIEA DLA ZADAŃ OPTYMALIZACJI NUMERYCZNEJ

Optymalizacja numeryczna stanowi bardzo ważny obszar zastosowań dla nowoczesnych algorytmów ewolucyjnych, ponieważ wiele rzeczywistych problemów w dziedzinie nauk technicznych może zostać sformułowanych w postaci zadań o takim charakterze.

W literaturze tematu najwięcej badań, dotyczących kwantowo inspirowanych algorytmów ewolucyjnych dla zadań optymalizacji numerycznej, zostało dotychczas poświęconych dwóm wariantom algorytmów: **iQIEA** (da Cruz *et al.*, 2004) i **rQIEA** (Zhang, Rong, 2007), reprezentującym dwa odmienne podejścia do kodowania liczb rzeczywistych w algorytmie. Proponowano również kilkanaście innych wariantów algorytmów QIEA dla zadań optymalizacji numerycznej m.in. (Lin, 2011; Zongyao, Zhou, 2011; Lu, Juang, 2011; Li *et al.*, 2011b; Liu *et al.*, 2008; Qin *et al.*, 2007; Wang *et al.*, 2012c; Yang, Ding, 2007), jednak są one słabo zbadane i rzadko komentowane. Różnorodne modyfikacje i rozszerzenia (koewolucja, elementy memetyki i innego rodzaju podejścia hybrydowe) były w tych algorytmach wprowadzane bardzo arbitralnie, a badania nad danym wariantem algorytmu często nie były kontynuowane w jakiegokolwiek późniejszej pracy.

Przed przystąpieniem do badań doświadczalnych nad algorytmem iQIEA, rozważona została potencjalna możliwość wykorzystania wprowadzonych w **Rozdziale 5** elementów algorytmów wyższych rzędów poprzez uogólnienie pojęć do obszaru kwantowo inspirowanych algorytmów ewolucyjnych oraz zadań kodowanych liczbami rzeczywistymi. Podstawowym problemem, związanym z próbą takiego uogólnienia, jest trudność zapisu liczb rzeczywistych w genach kwantowych, opartych na kubitach i rejestrach kwantowych. Dotychczas istniejące „prawdziwe” algorytmy

kwantowe (**Rozdział 2**) to algorytmy kombinatoryczne (np. algorytm faktoryzacji liczb ([Shor, 1994](#)), algorytm przeszukiwania zbiorów ([Grover, 1996](#))), a podstawowe pojęcia Informatyki Kwantowej nie zapewniają dogodnej reprezentacji rozwiązań dla problemów kodowanych liczbami rzeczywistymi. Z tego powodu badania nad kwantowo inspirowanymi algorytmami ewolucyjnymi dla optymalizacji numerycznej przeprowadzono nad kwantowo inspirowanym algorytmem ewolucyjnym iQIEA ([da Cruz et al., 2006](#)), wykorzystującym tzw. kodowanie impulsowe (**Rozdział 3.4.2**) zamiast wykorzystania kubitów, które są właściwe dla kodowania zmiennych binarnych.

7.1 PROCEDURA BADAWCZA

Jako najbardziej reprezentatywny dla kwantowo inspirowanych algorytmów ewolucyjnych wybrano algorytm iQIEA, opisany szczegółowo w **Rozdziale 3.4.2** na **str. 64**, a następnie poddano go procesowi metaoptymalizacji. Badania w dalszej części rozdziału zostały zorganizowane według następującego schematu:

1. Opracowano własną efektywną implementację algorytmu iQIEA oraz procedury badawczej, zrealizowanej w środowisku obliczeń równoległych, co pozwoliło odtworzyć wyniki podawane w literaturze ([da Cruz et al., 2004](#))
2. Wykonano metaoptymalizację algorytmu na pełnym zestawie 28 zadań testowych, charakteryzujących się dużym kosztem obliczeniowym, zaczerpniętych z benchmarku CEC2013, o wymiarowościach $N \in \{2, 10, 30, 50\}$. Zidentyfikowane zostały w ten sposób przedziały wartości wybranych parametrów (ξ, δ) , które zapewniają najwyższą efektywność algorytmu (**Teza II** pracy). Wykazano następnie przewagę algorytmu iQIEA nad referencyjnymi nowoczesnymi heurystykami przeszukiwania.
3. Dla porównania efektywności algorytmów wykorzystano klasyczny algorytm genetyczny ([Holland, 1975](#)), metodę roju cząstek ([Kennedy, Eberhart, 1995](#); [Angeline, 1998](#)), sympleksową metodę spadku ([Nel-](#)

der, Mead, 1965), oraz algorytm CMAES (Hansen, Ostermeier, 1996). Implementacje tych algorytmów zaczerpnięte zostały z biblioteki PyBrain (Schaul *et al.*, 2010) dla języka Python. Ponieważ w pracy (da Cruz *et al.*, 2006) zostało wykazane, że algorytm iQIEA charakteryzuje się lepszą efektywnością w zadaniach optymalizacji numerycznej niż heurystyczne algorytmy przeszukiwania m.in. Stochastic Genetic Algorithm (Tu, Lu, 2004), Fast Evolutionary Programming (Yao *et al.*, 1997), Fast Evolutionary Strategy (Yao *et al.*, 1999), porównanie z tymi algorytmami nie zostało powielone w przeprowadzonych przez autora eksperymentach.

7.2 EFEKTYWNA IMPLEMENTACJA ALGORYTMU iQIEA NA KLASTRZE

W celu przeprowadzenia badań nad wpływem wartości parametrów algorytmu iQIEA oraz ustalenia wartości, zapewniających najwyższą jego efektywność, niezbędne było opracowanie efektywnej implementacji.

Podobnie jak w **Rozdziale 6**, autor podjął początkowo próbę implementacji algorytmu iQIEA na jednostkach GPU w technologii NVidia CUDATM. Ograniczenia architektury dostępnych podczas badań kart graficznych zadecydowały jednak o niewielkiej korzyści z tego typu implementacji. W odróżnieniu od zaprezentowanego w przypadku optymalizacji kombinatorycznej przyspieszenia rzędu kilkudziesięciu razy na pojedynczym urządzeniu GPU, w przypadku optymalizacji numerycznej możliwe do uzyskania przyspieszenie było na poziomie poniżej 10x. Wynikało to przede wszystkim z dużych rozmiarów struktur danych (macierze obrotów w wielowymiarowych przestrzeniach), wykorzystywanych przez zastosowany w badaniach benchmark CEC2013. Rozmiar tych struktur uniemożliwiał umieszczenie ich w całości w szybkiej pamięci współdzielonej multiprocessora na urządzeniu GPU. Nie było również możliwości dokonania takiego partycjonowania tych struktur, aby zapewnić wysoką wydajność algorytmu poprzez kopiowanie części danych do szybkiej pamięci z pamięci globalnej. Ponadto, wykorzystany benchmark CEC2013 wykorzystuje obliczenia na zmiennych podwójnej precyzji, natomiast na dostępnych autorowi kartach graficznych obliczenia o tej precyzji były kilkukrotnie wolniejsze od ob-

liczeń pojedynczej precyzji. Możliwe do uzyskania na urządzeniu GPU przyśpieszenie było porównywalnego do tego, które można uzyskać, wykorzystując wszystkie rdzenie nowoczesnego procesora CPU (np. Intel Core i7).

Ze względu na te trudności, zamiast wykorzystania jednostek GPU, zastosowano inną technologię obliczeń równoległych: obliczenia gridowe (Nowotniak, 2012). Wykorzystano w tym celu część mocy obliczeniowych uczelnianego klastra BlueOcean, znajdującego się w Centrum Komputerowym Politechniki Łódzkiej, oraz polską infrastrukturę gridową PL-GRID (w ramach testowego grantu obliczeniowego). Implementacja na CPU nie wymagała kłopotliwej w tym przypadku i podatnej na błędy ponownej implementacji kodu w technologii CUDA.

Uczelniany klaster obliczeniowy BlueOcean oparty jest na systemie kolejowania (ang. *job schedulling*) PBS Torque oraz sieciowej przestrzeni składowania, wykorzystującej rozproszony system plików MooseFS (Baojun *et al.*, 2014). Klaster BlueOcean składa się z ponad 40 węzłów obliczeniowych o architekturze x86_64, spośród których każdy posiada 8, 12 lub 32 fizycznych rdzenie obliczeniowych CPU (na węzłach wyłączona jest celowo technologia HyperThreading). Procesory CPU na węzłach klastra BlueOcean to jednostki Intel(R) Xeon(R) E5620 @ 2.40 GHz, a każdy z węzłów wyposażony jest w co najmniej 32 GB pamięci operacyjnej RAM. Najwięcej jest węzłów obliczeniowych posiadających 8 rdzeni i na tych węzłach procesy obliczeniowych były uruchamiane najczęściej (węzły o większej liczbie rdzeni są zazwyczaj dedykowane procesom obliczeniowym prowadzonych przez innych użytkowników klastra i były one zajęte przez te procesy). Obliczenia na klastrze uruchamiane były w trybie wsadowym za pomocą autorskich skryptów powłoki systemowej Bash, oraz autorskiego oprogramowania qopt, stworzonego na potrzeby rozprawy, wykorzystującego interfejs Cython do komunikacji pomiędzy językiem Python a językiem niższego poziomu C, zapewniającym wysoką wydajność obliczania rzeczywistych funkcji celu.

7.3 METAOPTYMALIZACJA ALGORYTMU iQIEA

Dzięki wykorzystaniu środowiska obliczeń równoległych możliwe było przeprowadzenie metaoptymalizacji algorytmu iQIEA (da Cruz *et al.*, 2004) oraz porównanie jego efektywności z algorytmami referencyjnymi na pełnym zestawie funkcji z benchmarku CEC2013 (Liang *et al.*, 2013).

Celem tej części badań była analiza wpływu wartości dwóch parametrów (ξ, δ) oraz znalezienie przybliżonego przedziału ich optymalnych wartości. Algorytm iQIEA został zaimplementowany w języku Python. W celu przyspieszenia obliczeń, optymalizowane testowe funkcje celu zostały zaimplementowane w języku niższego poziomu – C. Jako generator liczb losowych wykorzystany został algorytm Mersenne Twister (Matsumoto, Nishimura, 1998), zapewniający bardzo dobre własności statystyczne, oraz charakteryzujący się bardzo dużą wartością okresu, co jest istotne dla prawidłowego przeprowadzenia eksperymentów numerycznych nad algorytmami stochastycznymi (Luke, 2009; Shilane *et al.*, 2008).

W odróżnieniu od częściowych wyników badań przedstawionych w pracach (Nowotniak, Kucharski, 2010b; Nowotniak, 2012), badania przedstawione w tym rozdziale zostały rozszerzone i przeprowadzone w sposób w pełni systematyczny, i obejmują one 28 trudnych zadań testowych o różnej wymiarowości, zaczerpniętych z uznawanego benchmarku CEC2013 (Liang *et al.*, 2013). Sposób porównania efektywności algorytmów (metody tworzenia rankingów) również został oparty na tym benchmarku.

Benchmark CEC20123 składa się z 28 zadań testowych, opartych na wielomodalnych, silnie nieliniowych funkcjach rzeczywistych $f : \mathbb{R}^N \mapsto \mathbb{R}$, obejmujących cztery wymiarowości $N \in \{2, 10, 30, 50\}$.

$$F = \{f_1, f_2, \dots, f_{28}\} \quad (7.1)$$

Wybrane funkcje z zestawu zostały zilustrowane na **Rysunku 7.1** dla wymiarowości $N = 2$. Zadanie optymalizacji w benchmarku CEC2013 polega na znalezieniu minimum tych funkcji. Funkcje charakteryzują się dużą złożonością i zgodnie z benchmarkiem algorytm iQIEA testowany był we wszystkich czterech wymiarowościach. Jako warunek stopu dla każdego z porównywanych algorytmów przyjęto maksymalną liczbę wywołań

funkcji oceny rozwiązań $\text{MaxFE} \in \mathbb{N}^+$, zgodnie z założeniami benchmarku:

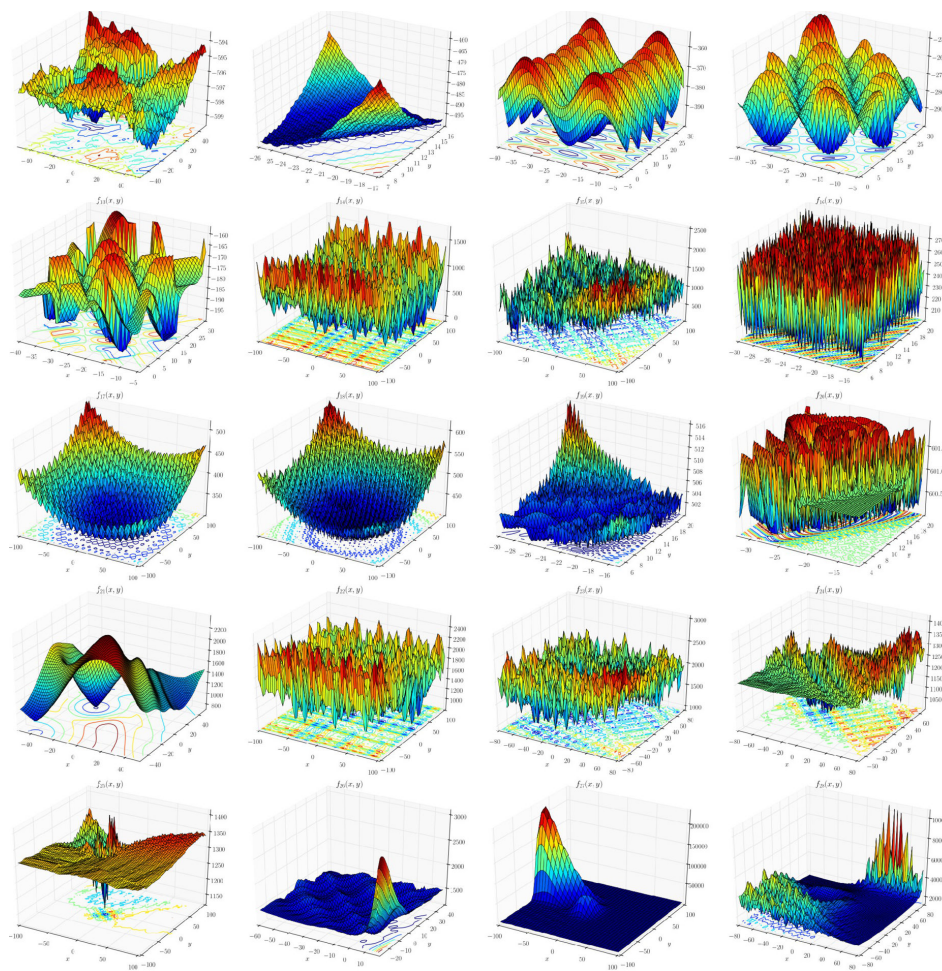
$$\text{MaxFE} = N \cdot 10000 \quad (7.2)$$

Autorzy benchmarku CEC2013 udostępniają jego implementację w języku C oraz w języku środowiska Matlab. Do realizacji badań wykorzystana została szybsza wersja implementacji w języku C, która została następnie zintegrowana z opracowanym przez autora rozprawy systemem w języku Python za pomocą interfejsu Cython (Behnel *et al.*, 2011; Seljebotn, 2009).

W przeprowadzonym eksperymencie numerycznym wykonano metaoptymalizację algorytmu iQIEA (da Cruz *et al.*, 2004) oraz porównanie jego efektywności z czterema znanymi algorytmami referencyjnymi:

1. **Metoda Neldera-Meada** (Nelder, Mead, 1965) – tzw. sympleksowa metoda spadku z typowymi wartościami współczynnika odbicia $\alpha = 1$, ekspansji $\gamma = 2$, kontrakcji $\rho = 1/2$ i redukcji $\sigma = 1/2$.
2. **Algorytm Genetyczny** (Holland, 1975) – zastosowany został wariant z kodowaniem rzeczywistym, rozmiar populacji $|P| = 40$, mutacja Gaussowska z prawdopodobieństwem $P_m = 0.1$, krzyżowanie jednopunktowe z prawdopodobieństwem $P_c = 0.8$, selekcja metodą proporcjonalną.
3. **PSO** (Eberbach, 2005) – metoda roju cząstek, rozmiar populacji 40, współczynnik inercji $w = 0.9$, współczynniki decydujące o kolektywnym zachowaniu cząstek $\phi_1 = 2$, $\phi_2 = 2$.
4. **CMAES** (Hansen, Ostermeier, 1996; Hansen, Kern, 2004) – strategia ewolucyjna z adaptacyjną macierzą kowariancji, rozmiar populacji w zależności od wymiarowości N , $\lambda = 4 + \lfloor 3 \log N \rfloor$, $\mu = \lambda/2$, $\sigma = 0.5$, pozostałe parametry dobierane na zasadzie adaptacyjnej podczas działania algorytmu, zgodnie z założeniami metody.

Implementacje algorytmów referencyjnych zaczerpnięte zostały z biblioteki PyBrain (Schaul *et al.*, 2010) dla języka Python. Wartości parametrów algorytmów referencyjnych nie były dostrajane w żaden sposób, ale zostały one ustalone na wartości domyślne z biblioteki PyBrain i odpowiadają



RYS. 7.1: Wykresy wybranych funkcji testowych $f : \mathbb{R}^N \mapsto \mathbb{R}$ z benchmarku CEC2013 dla wymiarowości $N = 2$ (Liang *et al.*, 2013)

one typowym wartościami, podawanym w literaturze. Ze względu na elementy losowości, występujące w porównywanych algorytmach, algorytmy uruchamiane były 20-krotnie dla każdej z funkcji testowych. Jest to procedura typowa dla porównywania efektywności algorytmów o charakterze stochastycznym (De Jong, 1975; Luke, 2009).

Na potrzeby eksperymentu przyjęto następujące stałe i typowe wartości podstawowych parametrów iQIEA (da Cruz *et al.*, 2004, 2006, 2010):

- Rozmiar populacji kwantowej $|Q| = 10$;
- Rozmiar populacji klasycznej $|P| = 15$;
- Stosunek liczby osobników w populacji P, decydujący o przeskalowaniu rozkładów w populacji Q o współczynnik δ , $\varphi = 1/5$;
- Liczba generacji, co ile wykonywane jest przeskalowanie rozkładów w genach kwantowych o współczynnik kontrakcji δ , $kstep = 20$;

Optymalizacji poddano wartości dwóch kluczowych parametrów rzeczywistych algorytmu $(\xi, \delta) \in [0, 1]^2$. Parametr ξ to parametr typowy dla klasycznych algorytmów ewolucyjnych, natomiast parametr δ to parametr o charakterze kwantowym, decydujący o skalowaniu rozkładów prawdopodobieństw zapisanych w genach kwantowych w populacji Q. Wymiarowość przestrzeni parametrów, podlegających optymalizacji, to $\dim(\{\xi, \delta\}) = \tilde{N} = 2$.

Jako algorytm metaoptymalizacji wybrano metodę **Local Unimodal Sampling** (Pedersen, Chipperfield, 2008; Pedersen, 2010). Zgodnie z (Pedersen, 2010, str. 14), w każdej iteracji algorytmu LUS przestrzeń parametrów (ξ, δ) badana jest poprzez sprawdzenie wartości metadopasowania $\tilde{f}(\vec{p}) \in \mathbb{R}$ w losowo wybieranym punkcie $\vec{p}_{new} \in [0, 1]^2$ według wzoru:

$$\vec{p}_{new} = \vec{p} + \vec{a} \quad (7.3)$$

gdzie: $\vec{p} \in [0, 1]^2$ to najlepszy dotychczas znaleziony punkt w przestrzeni parametrów (ξ, δ) , zaś $\vec{a} \in \mathbb{R}^2$ jest parą losowych liczb z przedziału $[-d, d]$. Na etapie inicjalizacji algorytmu LUS wartość $d \in \mathbb{R}$ przyjmuje wartość obejmującą cały zakres parametrów, tzn. $d = 1$. Jeżeli punkt $\vec{p}_{new}$ jest spoza obszaru $[0, 1]^2$, to składnik $\vec{a}$ we wzorze (7.3) losowany jest ponownie.

W każdej kolejnej iteracji algorytmu LUS zasięg próbkowania $d \in \mathbb{R}$ jest zmniejszany poprzez jego przeskalowanie o współczynnik kontrakcji $q \in (0, 1)$. W literaturze rekomendowane jest przyjęcie wartości współczynnika q zgodnie ze wzorem:

$$q = \sqrt[N/\beta]{1/2} = 2^{-\beta/\tilde{N}} = 2^{-\beta/2} \quad (7.4)$$

gdzie $\tilde{N}$ to wymiar przestrzeni parametrów poddawanych metaoptymalizacji. Ponieważ optymalizacji podlegają dwa parametry ξ i δ , to $\tilde{N} = 2$. W literaturze (Pedersen, Chipperfield, 2008) zalecana jest wartość $\beta = \frac{1}{3}$, co daje wartość współczynnika $q = 2^{-1/3/2} \approx 0.89$. Liczne próby uruchamiania algorytmu z różnymi wartościami q , pokazały, że algorytm uzyskuje szybszą zbieżność dla nieco mniejszej wartości $q = 0.8$ i taka wartość została przyjęta w zasadniczej części eksperymentu. Jako kryterium stopu algorytmu metaoptymalizacji LUS przyjęto maksymalną liczbę iteracji algorytmu $t = t_{\max}$. Wartość $t_{\max}$ była różna w zależności od wymiarowości N , co wynika ze wzrostu czasu potrzebnego do obliczenia wartości funkcji celu f wraz ze wzrostem wymiarowości.

Metody oceny efektywności algorytmów – Rankingi

Porównanie efektywności algorytmów zostało wykonane za pomocą dwóch metod (dwa sposoby tworzenia rankingów), zgodnie z założeniami CEC2013 (Liang *et al.*, 2013): liczba funkcji, dla których dany algorytm jest najlepszy; oraz średnie miejsce algorytmu w rankingach dla poszczególnych zadań testowych.

Ranking 1 – ocena danego algorytmu oparta jest na liczbie zadań testowych $\{1, 2, \dots, 28\}$, dla których ten algorytm osiągnął najlepszy wynik spośród wszystkich testowanych algorytmów. Algorytm uznawany jest za najlepszy dla danego zadania testowego, jeśli spośród wszystkich algorytmów znalazł on rozwiązanie, którego wartość jest najbardziej zbliżona do wartości optymalnej danej funkcji $f(x^*)$. Wartość ta jest uśredniana dla 20-krotnego uruchomienia algorytmu. Dany algorytm jest tym lepszy, im większa jest jego wartość oceny w tym rankingu.

Ranking 2 – Dla każdej spośród 28 funkcji tworzony jest niezależnie ranking wszystkich algorytmów (na zasadzie analogicznej jak w Rankingu 1), a ranking zbiorczy tworzony jest na podstawie uśrednionego miejsca poszczególnych algorytmów we wszystkich 28 rankingach. Ocena algorytmu w tym rankingu jest więc liczbą rzeczywistą, nie mniejszą niż 1 (algorytm najlepszy ze wszystkich porównywanych dla wszystkich zadań testowych) i nie większą niż liczba porównywanych algorytmów (algorytm najgorszy dla wszystkich 28 zadań testowych). Na przykład, jeśli dany algorytm jest liderem w Rankingu 2, może to oznaczać, że uzyskuje dobry wynik dla wszystkich funkcji np. jest drugi w kolejności, mimo że dla żadnej funkcji nie był najlepszy ze wszystkich. Wartość oceny 1.5 w Rankingu 2 oznacza, że średnia pozycja algorytmu dla wszystkich 28 zadań testowych to 1.5, więc dla połowy zadań testowych algorytm mógł być najlepszy ze wszystkich, a dla drugiej połowy był drugi w kolejności itd. Dany algorytm jest tym lepszy, im mniejsza jest jego wartość oceny w tym rankingu.

Do realizacji procesu metaoptymalizacji funkcję metadopasowania $\tilde{f}$ określono w następujący sposób:

$\tilde{f} : [0, 1]^2 \mapsto [1, 5]$ – wartość oceny algorytmu iQIEA w Rankingu 2 (średnia pozycja algorytmu w 28 rankingach dla poszczególnych zadań testowych) w porównaniu z algorytmami referencyjnymi.

Celem metaoptymalizacji była minimalizacja wartości metadopasowania $\tilde{f}$ w przestrzeni parametrów $\vec{p} = [0, 1]^2$. Metadopasowanie $\tilde{f}$ oparte zostało na wartości oceny algorytmu według Rankingu 2, więc jest ono wartością rzeczywistą z przedziału $[1, 5]$. Ponieważ wartości ocen w Rankingu 1 to liczby naturalne ze zbioru $\{1, 2, \dots, 28\}$, wykorzystanie tej miary powodowałoby, że dla różnych wartości parametrów (ξ, δ) algorytm otrzymywałby tę samą wartość oceny. Utrudniałoby to porównanie efektywności i wpływałoby niekorzystnie na przebieg procesu metaoptymalizacji.

CZAS OBLICZEŃ

W Tabeli 7.1 przedstawiono czas obliczeń algorytmu iQIEA podczas badań zasadniczych dla danej wymiarowości. W kolumnie t_1 przedstawiony jest czas pojedynczego uruchomienia algorytmu iQIEA o danym zestawie parametrów (ξ, δ) dla danej funkcji testowej f , dla danej wymiarowości N oraz przy założeniu, że warunek stopu MaxFE uzależniony był od wymiarowości zgodnie ze wzorem (7.2). Wartość t_1 została uśredniona dla $|F| = 28$ funkcji, ponieważ rzeczywisty czas obliczenia wartości poszczególnych funkcji z benchmarku różni się nieznacznie od siebie.

Ze względu na wzrost czasu obliczeń wraz ze wzrostem wymiarowości, przestrzeń parametrów (ξ, δ) została zbadana dla wymiarowości $N = 2$ w kilkukrotnie większej liczbie punktów ($\mathcal{P} = 800$ punktów, obliczenia rozproszone na 8 rdzeni CPU). Dla wymiarowości 10, 30 oraz 50 obliczenia zostały rozproszone na 24 rdzenie CPU klastra (trzy typowe węzły obliczeniowe klastra, z procesorami Intel(R) Xeon(R) CPU E5620 @ 2.40GHz).

Wartości w kolumnie t_2 przedstawiają sumaryczny czas obliczeń i są one iloczynem czasu pojedynczego uruchomienia algorytmu iQIEA, liczby funkcji, liczby powtórzeń, oraz liczby punktów w przestrzeni (ξ, δ) , zbadanych dla danej wymiarowości, zgodnie ze wzorem:

$$t_2 = t_1 \cdot R \cdot |F| \cdot \mathcal{P} \quad (7.5)$$

gdzie:

t_1 – czas pojedynczego uruchomienia algorytmu iQIEA

$R = 20$ – liczba powtórzeń

$|F| = 28$ – liczba funkcji (zadań testowych) w CEC2013

$\mathcal{P}$ – liczba punktów (ξ, δ) zbadanych dla danej wymiarowości

Dzięki rozproszeniu obliczeń na wiele rdzeni CPU na węzłach klastra, rzeczywisty czas obliczeń t_3 dla danej wymiarowości wyraża się wzorem:

$$t_3 = \frac{t_2}{n_{\text{cpu}}} \quad (7.6)$$

gdzie n_{cpu} to liczba niezależnych rdzeni CPU, na których obliczenia prowadzone były równolegle. Wartości t_2 i t_3 w Tabeli 7.1 wynikają więc ze wzorów (7.5) oraz (7.6). Jak widać w ostatniej kolumnie tabeli, łączny

czas obliczeń dla wszystkich wymiarowości przy ciągłym wykorzystaniu kilku węzłów klastra obliczeniowego wynosił ponad tydzień.

Wymiarowość N	Punkty $\mathcal{P}$	ncpu	t_1	t_2	t_3
2	800	8	1 s	5 dni	15 godzin
10	200	24	14 s	18 dni	18 godzin
30	100	24	107 s	69 dni	2.9 dnia
50	100	24	209 s	135 dni	5.6 dnia

TABELA 7.1: Czas obliczeń dla poszczególnych wymiarowości N :

t_1 – czas pojedynczego uruchomienia algorytmu iQIEA

t_2 – sumaryczny czas obliczeń

t_3 – rzeczywisty czas obliczeń na klastrze

7.4 WYNIKI – PORÓWNANIE EFEKTYWNOŚCI ALGORYTMÓW

W trakcie zasadniczego¹ procesu metaoptymalizacji przestrzeni parametrów $(\xi, \delta) = [0, 1]^2$ została spróbkowana następującą liczbę razy:

1. Wymiarowość $N = 2$ – 800 punktów
2. Wymiarowość $N = 10$ – 200 punktów
3. Wymiarowość $N = 30$ – 100 punktów
4. Wymiarowość $N = 50$ – 100 punktów

Wielokrotnie większa liczba punktów w przestrzeni parametrów (ξ, δ) , sprawdzonych dla wymiarowości $N = 2$ (800 punktów), wynika ze wzrostu czasu obliczeń, potrzebnego do przeprowadzenia badania algorytmu iQIEA (obliczenie wartości metadopasowania $\tilde{f}$) wraz ze wzrostem wymiarowości N .

¹Podczas realizacji badań przeprowadzono również liczne próby z uruchamianiem algorytmu iQIEA z wieloma innymi wartościami jego parametrów, dobieranymi ręcznie, w celu weryfikacji wyników oraz potwierdzenia przewagi algorytmu o parametrach znalezionych na drodze metaoptymalizacji. Efektywność algorytmu iQIEA z parametrami dobieranymi ręcznie była każdorazowo gorsza niż algorytmu z wartościami parametrów znalezionymi za pomocą zaprezentowanej procedury badawczej.

Punkty w przestrzeni parametrów $(\xi, \delta) = [0, 1]^2$, zapewniające najlepszą efektywność algorytmu iQIEA dla poszczególnych wymiarowości, zostały przedstawione w Tabeli 7.3 oraz zilustrowane na wykresach (A) – (D) na Rysunku 7.3.

Kolejnym etapem eksperymentu było porównanie efektywności algorytmu iQIEA z dobranymi w ten sposób parametrami z algorytmami referencyjnymi, zgodnie z założeniami benchmarku CEC2013 (Liang *et al.*, 2013). Wyniki porównania zostały przedstawione w Tabeli 7.2.

Wyniki rankingów, zamieszczonych w Tabeli 7.2, pokazują, że algorytm iQIEA z parametrami znalezionymi w procesie metaoptymalizacji jest najlepszy spośród wszystkich testowanych algorytmów zarówno w Rankingu 1 oraz 2. Dla każdej z wymiarowości $N \in \{2, 10, 30, 50\}$ algorytmy od najlepszego do najgorszego według Rankingu 2 to kolejno iQIEA, GA, CMAES, PSO i NelderMead. W Rankingu 1 dwa najlepsze algorytmy to kolejno iQIEA i CMAES (niezależnie od wymiarowości), kolejność algorytmów sytuujących się na pozycjach 3-5 jest natomiast różna w zależności od wymiarowości N . Obydwa rankingi pokazują, że algorytm iQIEA z dostrojonymi wartościami parametrów jest najlepszy zarówno pod względem liczby zadań testowych rozwiązywanych najlepiej (Ranking 1), jak i pod względem średniej pozycji dla wszystkich zadań testowych (Ranking 2). Przewaga algorytmu iQIEA została dodatkowo zilustrowana na wykresie słupkowym na Rysunku 7.2.

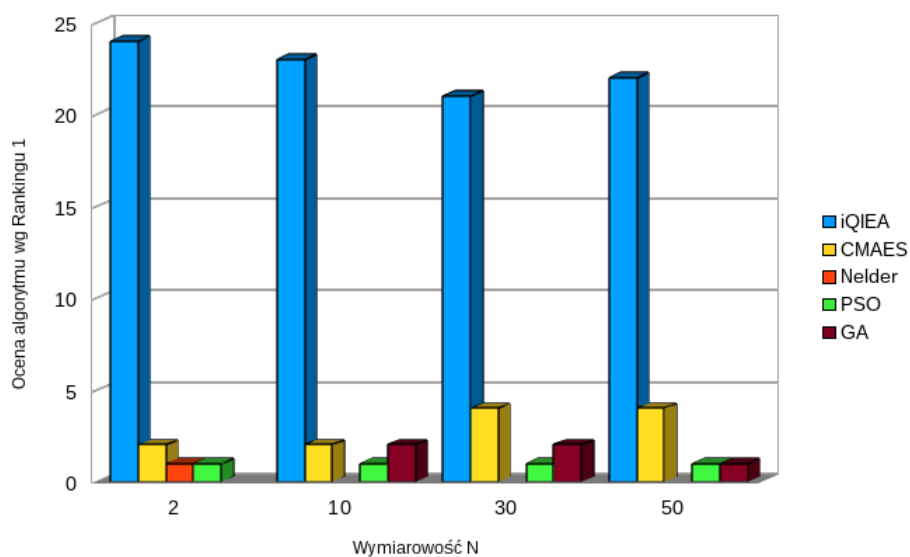
Ja widać na wykresach na Rysunku 7.3, wartości zapewniające najlepszą efektywność algorytmu iQIEA skupiają się w określonym obszarze przestrzeni (ξ, δ) . Dla wymiarowości $N = 2$, algorytm osiąga najwyższą efektywność dla $(\xi, \delta) = (0.591414, 0.77766)$. Dla wymiarowości $N = 10$, algorytm osiąga najwyższą efektywność dla $(\xi, \delta) = (0.650384, 0.95879)$. Dla wymiarowości $N = 30$, algorytm osiąga najwyższą efektywność dla $(\xi, \delta) = (0.199516, 0.88135)$. Dla wymiarowości $N = 50$, algorytm osiąga najwyższą efektywność dla $(\xi, \delta) = (0.255039, 0.979016)$.

Dla każdej wymiarowości $N \in \{10, 30, 50\}$ algorytm iQIEA uzyskuje najlepszą efektywność dla wartości współczynnika kontrakcji $\delta \gtrsim 0.9$. Jedynie dla niskiej wymiarowości $N = 2$, współczynnik kontrakcji ma wartość $\delta \geq 0.78$. Współczynnik krzyżowania ξ , zapewniający najlepszą efek-

Wymiarowość N	Ranking 1		Ranking 2	
	Algorytm	Ocena	Algorytm	Ocena
N = 2	iQIEA	24	iQIEA	1.285
	CMAES	2	GA	3.071
	NelderMead	1	CMAES	3.321
	PSO	1	PSO	3.321
	GA	0	NelderMead	4
N = 10	iQIEA	23	iQIEA	1.214
	CMAES	2	GA	2.5
	GA	2	CMAES	2.821
	PSO	1	PSO	4.035
	NelderMead	0	NelderMead	4.428
N = 30	iQIEA	21	iQIEA	1.357
	CMAES	4	GA	2.428
	GA	2	CMAES	2.571
	PSO	1	PSO	4.285
	NelderMead	0	NelderMead	4.357
N = 50	iQIEA	22	iQIEA	1.357
	CMAES	4	GA	2.392
	PSO	1	CMAES	2.5
	GA	1	PSO	4.321
	NelderMead	0	NelderMead	4.428

TABELA 7.2: Porównanie dostrojonego algorytmu iQIEA z algorytmami referencyjnymi

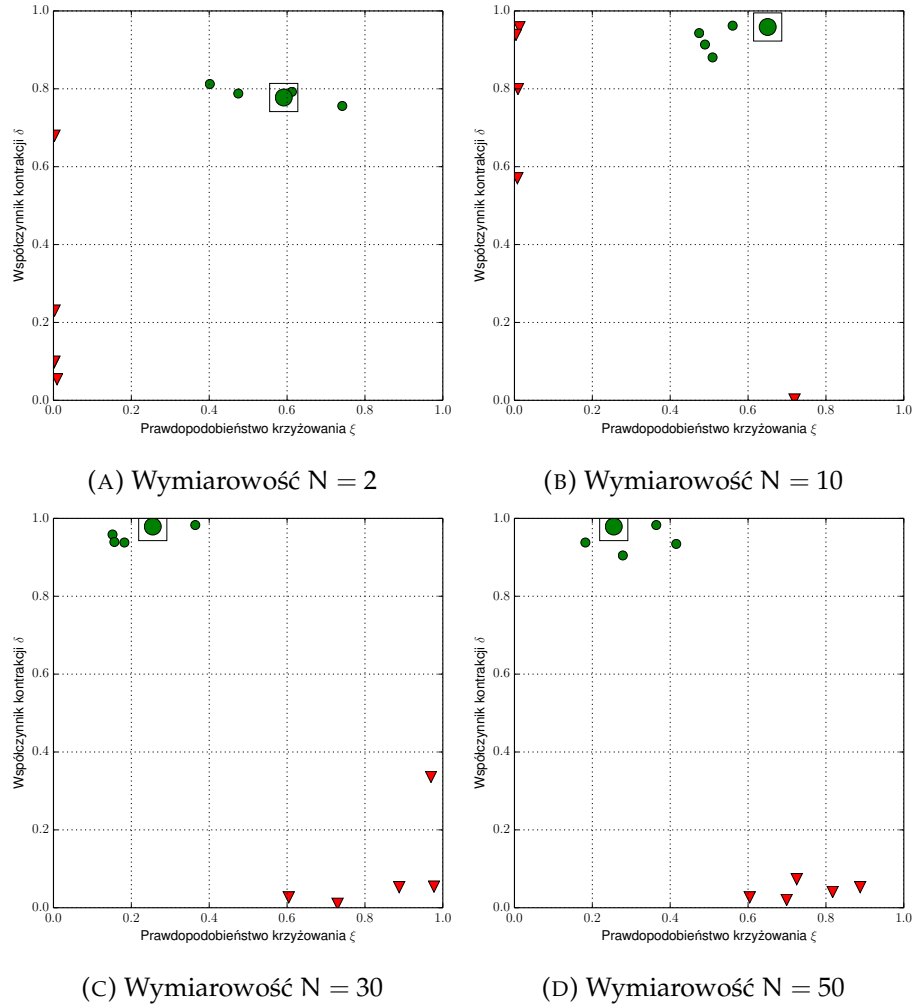
tywność dla trzech spośród czterech wymiarowości $N \in \{10, 30, 50\}$, sytuuje się w przedziale: $(0.35, 0.65)$.



RYS. 7.2: Porównanie efektywności algorytmów według Rankingu 1 dla poszczególnych wymiarowości

N = 2		N = 10		N = 30		N = 50	
Ranking 1	Ranking 2	Ranking 1	Ranking 2	Ranking 1	Ranking 2	Ranking 1	Ranking 2
(ξ, δ)	(ξ, δ)	(ξ, δ)	(ξ, δ)	(ξ, δ)	(ξ, δ)	(ξ, δ)	(ξ, δ)
(0.748,0.638)	(0.591,0.777)	(0.560,0.962)	(0.650,0.958)	(0.465,0.986)	(0.255,0.979)	(0.182,0.937)	(0.255,0.979)
(0.882,0.643)	(0.741,0.755)	(0.448,0.711)	(0.508,0.880)	(0.537,0.967)	(0.151,0.958)	(0.255,0.979)	(0.182,0.937)
(0.487,0.674)	(0.401,0.812)	(0.474,0.787)	(0.560,0.962)	(0.151,0.958)	(0.156,0.939)	(0.278,0.904)	(0.278,0.904)
(0.517,0.693)	(0.474,0.787)	(0.508,0.880)	(0.474,0.943)	(0.199,0.881)	(0.182,0.937)	(0.415,0.934)	(0.364,0.982)
(0.521,0.722)	(0.612,0.792)	(0.519,0.761)	(0.489,0.913)	(0.255,0.979)	(0.364,0.982)	(0.416,0.946)	(0.415,0.934)
(0.522,0.721)	(0.703,0.794)	(0.517,0.582)	(0.537,0.967)	(0.271,0.768)	(0.416,0.946)	(0.156,0.939)	(0.417,0.947)
(0.522,0.627)	(0.732,0.773)	(0.526,0.538)	(0.581,0.921)	(0.311,0.898)	(0.465,0.986)	(0.311,0.898)	(0.133,0.958)
(0.527,0.541)	(0.792,0.785)	(0.528,0.846)	(0.278,0.904)	(0.401,0.812)	(0.497,0.986)	(0.364,0.982)	(0.156,0.939)
(0.538,0.746)	(0.955,0.789)	(0.549,0.726)	(0.283,0.773)	(0.415,0.934)	(0.537,0.967)	(0.465,0.986)	(0.466,0.987)
(0.538,0.717)	(0.271,0.768)	(0.561,0.502)	(0.287,0.940)	(0.416,0.946)	(0.560,0.962)	(0.133,0.957)	(0.118,0.932)
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
(0.043,0.231)	(0.025,0.383)	(0.152,0.363)	(0.070,0.078)	(0.948,0.449)	(0.609,0.319)	(0.715,0.072)	(0.264,0.026)
(0.059,0.141)	(0.003,0.938)	(0.161,0.481)	(0.135,0.160)	(0.200,0.018)	(0.700,0.354)	(0.163,0.014)	(0.603,0.293)
(0.034,0.004)	(0.034,0.004)	(0.604,0.026)	(0.152,0.363)	(0.361,0.034)	(0.814,0.068)	(0.256,0.027)	(0.917,0.366)
(0.056,0.066)	(0.036,0.021)	(0.003,0.938)	(0.072,0.287)	(0.609,0.319)	(0.865,0.303)	(0.264,0.026)	(0.865,0.303)
(0.036,0.021)	(0.028,0.006)	(0.008,0.799)	(0.093,0.111)	(0.873,0.190)	(0.873,0.190)	(0.194,0.035)	(0.969,0.335)
(0.028,0.006)	(0.002,0.679)	(0.012,0.957)	(0.719,0.002)	(0.969,0.335)	(0.887,0.052)	(0.725,0.073)	(0.817,0.040)
(0.001,0.097)	(0.009,0.054)	(0.072,0.287)	(0.012,0.957)	(0.887,0.052)	(0.969,0.335)	(0.817,0.040)	(0.887,0.052)
(0.001,0.099)	(0.002,0.230)	(0.093,0.111)	(0.003,0.938)	(0.977,0.054)	(0.977,0.054)	(0.604,0.026)	(0.725,0.073)
(0.002,0.230)	(0.001,0.097)	(0.719,0.002)	(0.008,0.799)	(0.604,0.026)	(0.604,0.026)	(0.887,0.052)	(0.604,0.026)
(0.009,0.054)	(0.001,0.099)	(0.007,0.570)	(0.007,0.570)	(0.729,0.010)	(0.729,0.010)	(0.699,0.020)	(0.699,0.020)

TABELA 7.3: Wartości parametrów (ξ, δ) , znalezione w procesie metaoptymalizacji, zapewniające odpowiednio najlepszą (kolor zielony) i najgorszą (kolor czerwony) efektywność algorytmu iQIEA.



Rys. 7.3: Wyniki procesu metaoptymalizacji algorytmu iQIEA na pełnym benchmarku CEC2013. Zielone okrągłe punkty prezentują wartości parametrów (ξ, δ) , zapewniające najlepszą efektywność algorytmu iQIEA. Czerwone trójkąty prezentują zbadane wartości parametrów, dla których algorytm charakteryzuje się najgorszą efektywnością

7.5 PODSUMOWANIE I WNIOSKI

Najważniejszym efektem badań doświadczalnych, zaprezentowanych w rozdziale, jest określenie zawężonego przedziału parametrów (ξ, δ) , które zapewniają wysoką efektywność algorytmu iQIEA dla różnorodnego zestawu zadań optymalizacji numerycznej o różnych wymiarowościach. Pokazuje to, że zastosowanie metaoptymalizacji oraz implementacja w środowisku obliczeń równoległych jest skuteczną metodą doboru wartości parametrów algorytmu i pozwala ona określić przybliżone przedziały ich właściwych wartości. Stanowi to dowód **Tezy II** rozprawy. Wykonane eksperymenty numeryczne pokazały, że właściwym przedziałem wartości współczynnika kontrakcji jest $0.9 \lesssim \delta < 1$, natomiast właściwym przedziałem wartości współczynnika krosowania jest $0.2 \lesssim \xi \lesssim 0.7$. Wyniki pokazują ponadto, że wpływ wartości współczynnika δ na efektywność algorytmu jest znacznie większy niż współczynnika ξ . Silny wpływ wartości obydwu parametrów ξ, δ jest szczególnie widoczny w przypadku wymiarowości optymalizowanych funkcji wyższych niż $N = 2$.

PODSUMOWANIE I WNIOSKI

Głównym celem niniejszej rozprawy była analiza własności nowej klasy kwantowo inspirowanych algorytmów ewolucyjnych i genetycznych. W rozprawie przedstawiono wyniki badań, obejmujące zarówno teoretyczną analizę własności algorytmów, jak i badania doświadczalne w postaci eksperymentów numerycznych. Zasadnicze problemy, które wymagały rozwiązania, polegały na braku usystematyzowania możliwości wprowadzania elementów inspirowanych Informatyką Kwantową do algorytmu ewolucyjnego, nielicznych dotychczasowych próbach analiz teoretycznych własności algorytmów oraz braku wiedzy na temat możliwości i skutków zwiększenia wpływu elementów kwantowych na działanie algorytmów ewolucyjnych. Najważniejsze oryginalne elementy zaproponowanych rozwiązań obejmują:

1. wprowadzenie pojęć kwantowo inspirowanych algorytmów wyższych rzędów (rząd i współczynniki kwantowości) (Rozdział 5.2);
2. kwantowo inspirowany algorytm genetyczny rzędu II (Rozdział 5.3);
3. analizę podstawowych własności algorytmów (Rozdział 4);
4. metaoptymalizację wartości parametrów algorytmów (Rozdział 6.4 i Rozdział 7.3);
5. efektywne implementacje w środowisku obliczeń równoległych (Rozdział 6.3 i Rozdział 7.2).

Do najważniejszych elementów pracy należy wprowadzenie podstawowych pojęć kwantowo inspirowanych algorytmów wyższych rzędów oraz opracowanie autorskiego kwantowo inspirowanego algorytmu genetycznego rzędu II.

Było to możliwe dzięki przeprowadzeniu teoretycznych analiz własności algorytmów poprzez adaptację kilku wybranych, znanych metod do obszaru algorytmów kwantowo inspirowanych. Poddano badaniom

zarówno zagadnienie zbieżności ogólnego kwantowo inspirowanego algorytmu ewolucyjnego, wykorzystując twierdzenie Banacha o punkcie stałym odwzorowania zwężającego (Rozdział 4.1), jak i dopasowanie binarnych chromosomów kwantowych do schematów w kwantowo inspirowanych algorytmach genetycznych, w których reprezentacja oparta jest na kubitach (Rozdział 4.2). Wyniki tych analiz zostały zaprezentowane w pracach (Nowotniak, Kucharski, 2012a), (Nowotniak, Kucharski, 2010a). Przeprowadzona analiza pozwoliła na uogólnienie pojęcia dopasowania chromosomu kwantowego do schematu w kwantowo inspirowanym algorytmie genetycznym. Zostały również wyznaczone podstawowe miary statystyczne liczby chromosomów pasujących do wybranego schematu w populacji kwantowej (Rozdział 4.2.3 i Rozdział 4.2.4).

Szczególną uwagę zwrócono na usystematyzowanie sposobów działania wybranych algorytmów przeszukiwania w odniesieniu do przestrzeni ich działania. Zidentyfikowano w ten sposób związek pomiędzy binarnymi chromosomami kwantowymi i odpowiadającymi im schematami oraz pokazano, iż kwantowo inspirowany algorytm genetyczny może w przybliżony sposób oceniać dopasowanie schematów poprzez wielokrotną obserwację i ocenę binarnych chromosomów kwantowych (Rozdział 5.1.3).

Wnioski płynące z tych analiz doprowadziły do sformułowania pojęć kwantowo inspirowanych algorytmów wyższych rzędów – rzędu i współczynników kwantowości kwantowo inspirowanego algorytmu genetycznego. Pokazano, że łączenie wielu genów w chromosomie do wspólnych rejestrów kwantowych w algorytmach wyższych rzędów pozwala na modelowanie związków pomiędzy wartościami genów, oraz że reprezentacja rozwiązań w algorytmach wyższych rzędów pozwala na lepsze odwzorowanie ukształtowania funkcji celu w genotypie kwantowym. Pozwoliło to postawić hipotezę o możliwości wykorzystania II rzędu kwantowości kwantowo inspirowanego algorytmu genetycznego do poprawy efektywności algorytmu w szczególności dla zadań zwodniczych (Rozdział 5.2).

W rezultacie przeprowadzonych analiz opracowano autorski kwantowo inspirowany algorytm genetyczny rzędu II, w którym pary genów połączone są do wspólnych rejestrów kwantowych (Rozdział 5.3), (Nowotniak, Kucharski, 2014). W opracowanym algorytmie zaproponowano

połączenie sąsiadujących ze sobą genów do wspólnych 2-kubitowych rejestrów kwantowych. Zaproponowano również nowy operator genetyczny, odpowiedni dla nowego rodzaju reprezentacji rozwiązań w algorytmie (Rozdział 5.3).

Efektywność autorskiego algorytmu rzędu II została zweryfikowana doświadczalnie poprzez porównanie z algorytmami referencyjnymi: prostym algorytmem genetycznym, pierwotnym kwantowo inspirowanym algorytmem genetycznym oraz algorytmem pierwotnym o parametrach dobranych w procesie metaoptymalizacji. Wyniki przeprowadzonych eksperymentów numerycznych pokazały, iż autorski kwantowo inspirowany algorytm genetyczny rzędu II posiada w większości przypadków lepszą efektywność niż pozostałe algorytmy, co stanowi dowód I tezy rozprawy (Rozdział 6).

Implementacje algorytmów i procedur badawczych zostały wykonane w środowiskach obliczeń równoległych i masowo równoległych (CUDATM, PL-GRID) (Nowotniak, Kucharski, 2011a,b, 2012b). W przypadku badań doświadczalnych nad kwantowo inspirowanym algorytmem genetycznym pozwoliło to na uzyskanie ponad 400-krotnego przyspieszenia obliczeń w stosunku do implementacji na pojedynczym rdzeniu CPU.

Dzięki opracowaniu wysoce efektywnych implementacji algorytmów, możliwe było zastosowanie techniki metaoptymalizacji do doboru wartości wybranych parametrów algorytmów. Metaoptymalizacji poddano wartości parametrów zarówno pierwotnego kwantowo inspirowanego algorytmu genetycznego, jak i reprezentatywnego dla kwantowo inspirowanych algorytmów ewolucyjnych algorytmu iQIEA. Pozwoliło to na dobranie właściwych wartości parametrów algorytmów zarówno dla zadań optymalizacji kombinatorycznej (Rozdział 6) jak i numerycznej (Rozdział 7). Zaprezentowany skuteczny dobór właściwych wartości parametrów kwantowo inspirowanego algorytmu genetycznego i ewolucyjnego za pomocą techniki metaoptymalizacji stanowi doświadczalny dowód II tezy rozprawy (Nowotniak, Kucharski, 2010b; Nowotniak, 2012).

Pozwala to stwierdzić, że został osiągnięty postawiony na wstępie rozprawy cel badań. Wyniki przeprowadzonych badań były podstawą

do opublikowania 14 publikacji w recenzowanych czasopismach naukowych i materiałach konferencyjnych, w tym jeden artykuł (Nowotniak, Kucharski, 2012b) w czasopiśmie z tzw. listy filadelfijskiej. Wyniki tych publikacji były cytowane w czasopismach o zasięgu międzynarodowym (Manju, Nigam, 2012; Perroni, 2013; Bansal, 2013). Efektywność charakterystycznego dla nowej klasy dotychczasowego kwantowo inspirowanego algorytmu genetycznego i ewolucyjnego zbadano również na przykładzie kilku wybranych przykładów aplikacyjnych. Wyniki tych badań zaprezentowano w pracach (Jeżewski, Łaski, Nowotniak, 2010), (Jopek, Nowotniak, Postolski, Babout, Janaszewski, 2009).

Możliwe jest wskazanie kilku obiecujących kierunków dalszych badań. Jak dowiedziono w rozprawie, zaproponowany przez autora kwantowo inspirowany algorytm genetyczny rzędu II charakteryzuje się lepszą efektywnością niż algorytm pierwotny o parametrach dobranych w procesie metaoptymalizacji. Parametry algorytmu autorskiego zostały dobrane na podstawie kilku prób, ale dzięki skutecznemu wykorzystaniu II rzędu kwantowości w algorytmie autorskim, algorytm ten ma lepszą efektywność niż algorytmy pierwotny o parametrach dobranych poprzez metaoptymalizację. Interesującym kierunkiem dalszych prac jest zastosowanie techniki metaoptymalizacji do dalszej poprawy efektywności autorskiego algorytmu rzędu II.

Innym szczególnie interesującym kierunkiem dalszych badań jest możliwość zastosowania wprowadzonych pojęć algorytmów wyższych rzędów (rząd i współczynnik kwantowości) do algorytmów innych niż algorytmy ewolucyjne, w celu zbadania zwiększenia wpływu elementów kwantowych na działanie innych metod inteligencji obliczeniowej.

ZAŁĄCZNIKI

Załącznik A

POSTULATY MECHANIKI KWANTOWEJ

Możliwości kwantowego modelu obliczeniowego są determinowane przez kwantowomechaniczny opis fizycznej rzeczywistości. W tej sekcji zostały przedstawione postulaty mechaniki kwantowej, zgodnie z którą działałby komputer kwantowy. Należy przy tym zaznaczyć, że Informatyka Kwantowa, koncentrując się na aspekcie obliczeniowym, wykorzystuje jedynie ograniczony zakres teorii mechaniki kwantowej. W ogólnym przypadku ewolucję czasową układu kwantowego opisuje równanie Schrödingera z czasem, będące równaniem różniczkowym zwyczajnym, jednakże w Informatyce Kwantowej zmiany stanów opisuje się w postaci dyskretnych przekształceń, odpowiadających kolejnym etapom obliczeń kwantowych.

Postulat A.1. Przestrzenią stanów izolowanego układu fizycznego jest przestrzeń Hilberta $\mathcal{H}$, a stan układu opisywany jest przez jednostkowy wektor $|\psi\rangle \in \mathcal{H}$ z przestrzeni $\mathcal{H}$, nazywany *wektorem stanu*.

Postulat A.2. Przestrzenią stanów złożonego układu fizycznego jest iloczyn tensorowy przestrzeni stanów układów składowych. W przypadku n układów, znajdujących się w stanach $\phi_1, \phi_2, \dots, \phi_n$, stan układu jako całości opisywany jest przez iloczyn tensorowy $|\phi_1\rangle \otimes |\phi_2\rangle \otimes \dots \otimes |\phi_n\rangle$.

Postulat A.3. Ewolucja czasowa stanu $|\psi\rangle$ izolowanego układu fizycznego opisywana jest przez równanie Schrödingera z czasem:

$$i\hbar \frac{\delta}{\delta t} |\psi\rangle = \hat{H} |\psi\rangle \quad (\text{A.1})$$

gdzie $\hat{H}$ jest operatorem samosprężonym, nazywanym *hamiltonianem* układu, w przestrzeni $\mathcal{H}$.

Ponieważ obliczenia kwantowe opisuje się za pomocą serii dyskretnych przekształceń, zamiast równania różniczkowego (A.1), dynamikę układu kwantowego można opisać prościej w następujący sposób:

Jeśli izolowany układ fizyczny w chwili czasu t_1 znajdował się w stanie $|\psi\rangle$, to w chwili czasu $t_2 \geq t_1$ stan układu $|\psi'\rangle$ może być opisany za pomocą operatora unitarnego $U = U(t_2 - t_1)$ jako:

$$|\psi'\rangle = U |\psi\rangle \quad (\text{A.2})$$

Postulat A.4. Pomiar rzutowy¹ (ang. *projective measurement*) stanu określony jest za pomocą pewnego operatora hermitowskiego M . Ponieważ macierze hermitowskie są macierzami normalnymi ($MM^\dagger = M^\dagger M$), operator M posiada rozkład spektralny

$$M = \sum_m m P_m \quad (\text{A.3})$$

gdzie P_m jest operatorem rzutowym (ang. *projector*) na przestrzeń własną M , a wartości własne m odpowiadają możliwym do otrzymania wynikom pomiarów.

Prawdopodobieństwo otrzymania rezultatu m pomiaru stanu układu określone jest wyrażeniem:

$$p(m) = \langle \phi | P_m | \phi \rangle \quad (\text{A.4})$$

Wraz z uzyskaniem rezultatu pomiaru m stan układu zostaje ustalony na

$$|\phi'\rangle = \frac{P_m |\phi\rangle}{\sqrt{p(m)}} \quad (\text{A.5})$$

Stan bezpośrednio po pomiarze staje się więc jednym z wektorów własnych użytego operatora pomiarowego M .

¹możliwe jest także bardziej ogólne sformułowanie operacji pomiaru (ang. *general measurement*), ale jest ono rzadko wykorzystywane w Informatyce Kwantowej

Załącznik B

TWIERDZENIE BANACHA O PUNKCIE STAŁYM

W załączniku podane zostały definicje podstawowych pojęć wraz z twierdzeniem Banacha, wykorzystywanych w rozdziale 4.1 do analizy zbieżności kwantowo inspirowanego algorytmu ewolucyjnego.

Definicja B.1. Niech S będzie pewnym niepustym zbiorem. *Metryką* (odległością) w zbiorze S nazywamy odwzorowanie $\delta : S \times S \mapsto \mathbb{R}$, spełniające następujące warunki dla wszystkich $a, b, c \in S$:

- $\delta(a, b) = 0 \iff a = b$
- $\delta(a, b) = \delta(b, a)$ (δ jest relacją symetryczną)
- $\delta(a, b) \leq \delta(a, c) + \delta(c, b)$ (tzw. warunek trójkąta)

Definicja B.2. Jeśli δ jest metryką w zbiorze S , to parę $\langle S, \delta \rangle$ nazywamy *przestrzenią metryczną*.

Definicja B.3. Przestrzeń metryczna $\langle S, \delta \rangle$ jest *przestrzenią zupełną*, jeżeli każdy ciąg Cauchy’ego (zbieżny ciąg elementów tej przestrzeni) posiada granicę, należącą do tej przestrzeni.

Definicja B.4. Niech $\langle S, \delta \rangle$ będzie przestrzenią metryczną oraz niech $G : S \mapsto S$ będzie odwzorowaniem tej przestrzeni w nią samą. Mówimy, że odwzorowanie G jest *związujące* (jest *kontrakcją*), jeżeli zachodzi:

$$\exists \epsilon \in (0, 1) \forall x, y \in S \delta(G(x), G(y)) \leq \epsilon * \delta(x, y) \quad (\text{B.1})$$

Inaczej mówiąc, odległość (w sensie metryki δ) dowolnych dwóch elementów przestrzeni S , przekształconych przez G , jest mniejsza lub równa odległości tych elementów pomnożonej przez pewną stałą $\epsilon \in (0, 1)$.

Twierdzenie B.1 (Banacha o punkcie stałym). Niech $\langle S, \delta \rangle$ będzie przestrzenią metryczną zupełną oraz niech $G : S \mapsto S$ będzie odwzorowaniem związanym (kontrakcją). Wówczas:

-
- odwzorowanie G ma jednoznacznie określony punkt stały $x^* \in S$
 - dla dowolnego $x_0 \in S$ ciąg $(x_0, G(x_0), G(G(x_0)), \dots)$ jest zbieżny do x^*

$$\lim_{i \rightarrow \infty} G^i(x_0) = x^*$$

Załącznik C

WYBRANE SZCZEGÓŁY IMPLEMENTACYJNE ALGORYTMÓW

Na potrzeby realizacji badań doświadczalnych, których wyniki zostały zaprezentowane w rozprawie, opracowano starannie zaplanowane autorskie oprogramowanie, wykorzystujące języki programowania C, Python, CUDA C oraz skrypty powłoki systemowej bash. Poszczególne języki programowania stosowane były do implementacji elementów, do których były najbardziej adekwatne. Benchmarki i funkcje testowe, których obliczanie wartości jest operacją dominującą, zaimplementowane były w kompilowanym języku niskiego poziomu, odpowiednio C lub CUDA C (w przypadku implementacji w technologii GPGPU). Dotyczyło to również implementacji samych algorytmów ewolucyjnych, zrealizowanych w tych językach. Procedury badawcze oraz algorytmy przetwarzania i analizy wyników zaimplementowane zostały natomiast w interpretowanym języku obiektowym wysokiego poziomu (Python) i zintegrowane z pozostałymi modułami za pomocą interfejsu Cython ([Behnel et al., 2011](#); [Seljebotn, 2009](#)).

Do elementów opracowanego oprogramowania należą:

IMPLEMENTACJE ALGORYTMÓW o wysokiej wydajności;

MODEL OBIEKTOWY zapewniający podstawowe klasy abstrakcyjne, umożliwiające dostęp z poziomu językach Python do efektywnych implementacji algorytmów w językach niższego poziomu;

SYSTEM LOGOWANIA , który w automatyczny sposób zapisuje stan całego algorytmu w trakcie kolejnych generacji podczas ewolucji;

BIBLIOTEKA FUNKCJI POMOCNICZYCH , zawierająca podstawowe procedury, wykorzystywane w obszarze algorytmów ewolucyjnych,

m.in. szeroki zestaw operatorów genetycznych, metody selekcji, metody generowania liczb losowych o określonych rozkładach etc.;

WARSTWA ROZPRASZANIA OBLICZEŃ , pozwalająca na łatwe wykonywanie eksperymentów obliczeniowych z wykorzystaniem wielu rdzeni lokalnego komputera, wielu jednostek obliczeniowych a także obliczenia w infrastrukturze PL-GRID;

CUDA implementacje algorytmów z wykorzystaniem obliczeń masowo równoległych na karcie graficznej, jeśli karta o takich właściwościach jest dostępna na lokalnym komputerze;

ZESTAWY FUNKCJI TESTOWYCH – m.in. gotowe benchmarki CEC2005, CEC2011, BBOB, a także opracowane przez autora zestawy danych testowych dla typowych problemów kombinatorycznych (problem plecakowy, problem komiwojażera);

NARZĘDZIE WIZUALIZACJI WYNIKÓW – skrypty służące m.in. do tworzenia wykresów porównujących efektywności algorytmów, analizy propagacji bloków budujących, badania histogramu rozkładu wyników, ukształtowania funkcji celu;

SKRYPTY POMOCNICZE do przetwarzania danych zapisanych w wyjściowych plikach tekstowych.

Opracowane autorskie oprogramowanie umożliwia łatwe wykonywanie i analizę wyników eksperymentów numerycznych oraz łatwą wymienialność poszczególnych elementów implementacyjnych z poziomu językach Python.

IMPLEMENTACJA WYPROWADZONYCH MIAR STATYSTYCZ- NYCH $E[L]$ I $V[L]$

```
1 def M(qchromo, schema):
2     result = 1.
3     for i in xrange(len(schema)):
4         if schema[i] == '0':
5             result *= np.square(qchromo[0,i])
6         elif schema[i] == '1':
7             result *= np.square(qchromo[1,i])
8     return result
```

```
1 def E(Q, schema):
2     sum1 = 0.
3     for w in xrange(len(Q)+1):
4         sum2 = 0.
5         for c in xrange(2**len(Q)):
6             bstr = dec2bin(c, len(Q))
7             if bstr.count('1') == w:
8                 elem = 1.
9                 for j in xrange(len(bstr)):
10                    if bstr[j] == '0':
11                        elem *= 1 - M(Q[j], schema)
12                    else:
13                        elem *= M(Q[j], schema)
14                sum2 += elem
15            sum1 += 1. * w * sum2
16    return sum1
```

```

1 def V(Q,schema):
2     sum1 = 0.
3     for w in xrange(len(Q)+1):
4         sum2 = 0.
5         for c in xrange(2**len(Q)):
6             bstr = dec2bin(c, len(Q))
7             if bstr.count('1') == w:
8                 elem = 1.
9                 for j in xrange(len(bstr)):
10                    if bstr[j] == '0':
11                        elem *= 1 - M(Q[j], schema)
12                    else:
13                        elem *= M(Q[j], schema)
14                    sum2 += elem
15            sum1 += 1. * w*w * sum2
16    return sum1 - E(Q,schema)**2

```

IMPLEMENTACJA ALGORYTMU QIGA2 W JĘZYKU C

```

1
2 #include "qiga2.h"
3
4 #define Qij (Q + i * (4 * (chromlen/2)) + (4 * j))
5
6 /*
7  * Quantum genes initialization stage
8  */
9 void QIGA2::initialize() {
10     int i, j;
11     for (i = 0; i < popsize; i++) {
12         for (j = 0; j < chromlen/2; j++) {
13             Qij[0] = .5f; // equal superposition of states
14             Qij[1] = .5f;
15             Qij[2] = .5f;
16             Qij[3] = .5f;
17         }
18     }

```

```

19 }
20
21 /*
22  * Observation of classical population stage;
23  * Sampling the search space with respect to the quantum
24  * population probability distributions
25  */
26 void QIGA2::observe() {
27     int i, j;
28     for (i = 0; i < popsize; i++) {
29         for (j = 0; j < chromlen/2; j++) {
30             float r = 1.0f * rand() / (RAND_MAX + 1.0);
31             if (r < Qij[0]*Qij[0]) {
32                 memcpy(&(P[i][2*j]), "00", 2);
33             }
34             else if (r < Qij[0]*Qij[0] + Qij[1]*Qij[1]) {
35                 memcpy(&(P[i][2*j]), "01", 2);
36             }
37             else if (r < Qij[0]*Qij[0] + Qij[1]*Qij[1] + Qij
38                 [2]*Qij[2]) {
39                 memcpy(&(P[i][2*j]), "10", 2);
40             }
41             else { // r < P(00) + P(01) + P(10) + P(11) = 1
42                 memcpy(&(P[i][2*j]), "11", 2);
43             }
44         }
45     }
46
47     // for knapsack problem only
48     void QIGA2::repair() {
49         for (int i = 0; i < popsize; i++) {
50             problem->repairer(P[i], chromlen);
51         }
52     }
53
54
55     /*

```

```

56  * Individuals evaluation stage
57  */
58  void QIGA2::evaluate() {
59      int i, j;
60      for (i = 0; i < popsize; i++) {
61          fvals[i] = problem->evaluator(P[i], chromlen);
62      }
63  }
64
65
66  /*
67   * Update stage -- quantum genetic operators;
68   * rotations in qubit state spaces
69   */
70  void QIGA2::update() {
71      for (int i = 0; i < popsize; i++) {
72          int fxGTfb = fvals[i] >= bestval; //  $f(x) \geq f(b)$ 
73          for (int j = 0; j < chromlen / 2; j++) {
74
75              //  $Q'$ 
76              float Qprim[4];
77
78              // update  $Q$ 
79              char buf[3] = { '\\0' };
80              memcpy(buf, best + (j * 2), 2);
81              int bestamp = strtol(buf, 0, 2);
82              float sum = 0.f;
83              for (int amp = 0; amp < 4; amp++) {
84                  if (amp != bestamp) {
85                      Qprim[amp] = this->mi * Qij[amp];
86                      sum += Qprim[amp] * Qprim[amp];
87                  }
88              }
89              Qprim[bestamp] = sqrtf(1.f - sum);
90
91              //  $Q \leftarrow Q'$ 
92              Qij[0] = Qprim[0];
93              Qij[1] = Qprim[1];

```

```

94         Qij[2] = Qprim[2];
95         Qij[3] = Qprim[3];
96     }
97 }
98 }
99
100 void QIGA2::storebest() {
101     int i;
102     float val = -1;
103     char *b;
104     for (i = 0; i < popsize; i++) {
105         if (fvals[i] > val) {
106             val = fvals[i];
107             b = P[i];
108         }
109     }
110     if (val > bestval) {
111         bestval = val;
112         memcpy(best, b, chromlen);
113     }
114 }
115
116 void QIGA2::QIGA2() {
117     int t = 0;
118     bestval = -1;
119     initialize();
120     observe();
121     repair();
122     evaluate();
123     storebest();
124     while (t < tmax) {
125         observe();
126         repair();
127         evaluate();
128         update();
129         storebest();
130         t++;
131     }

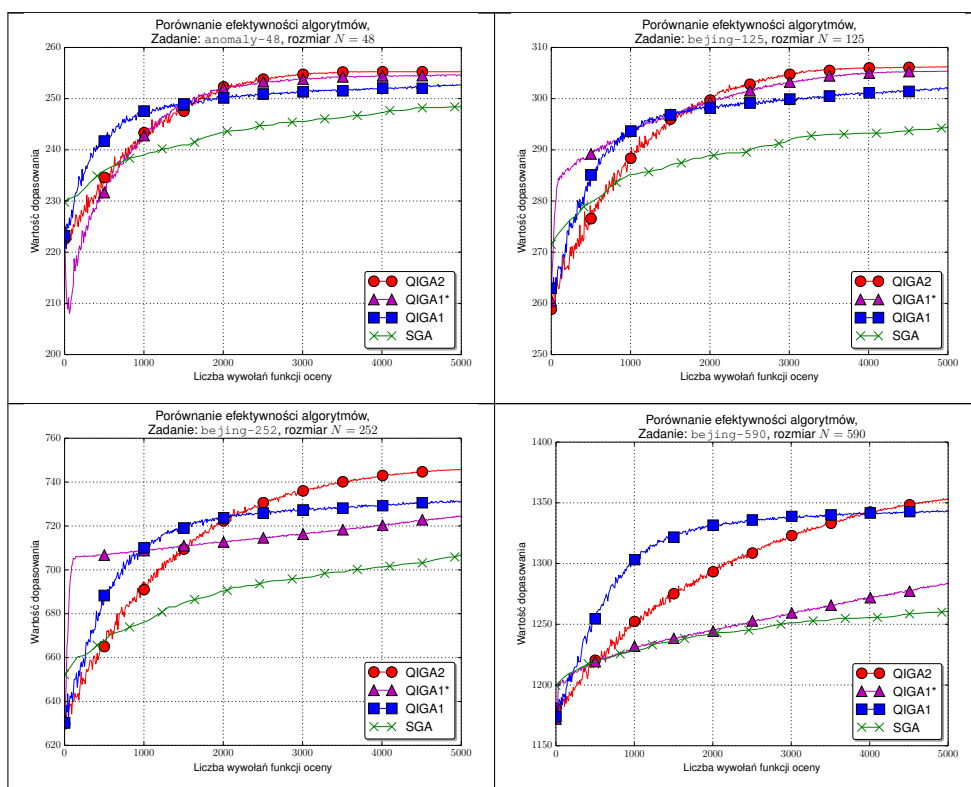
```

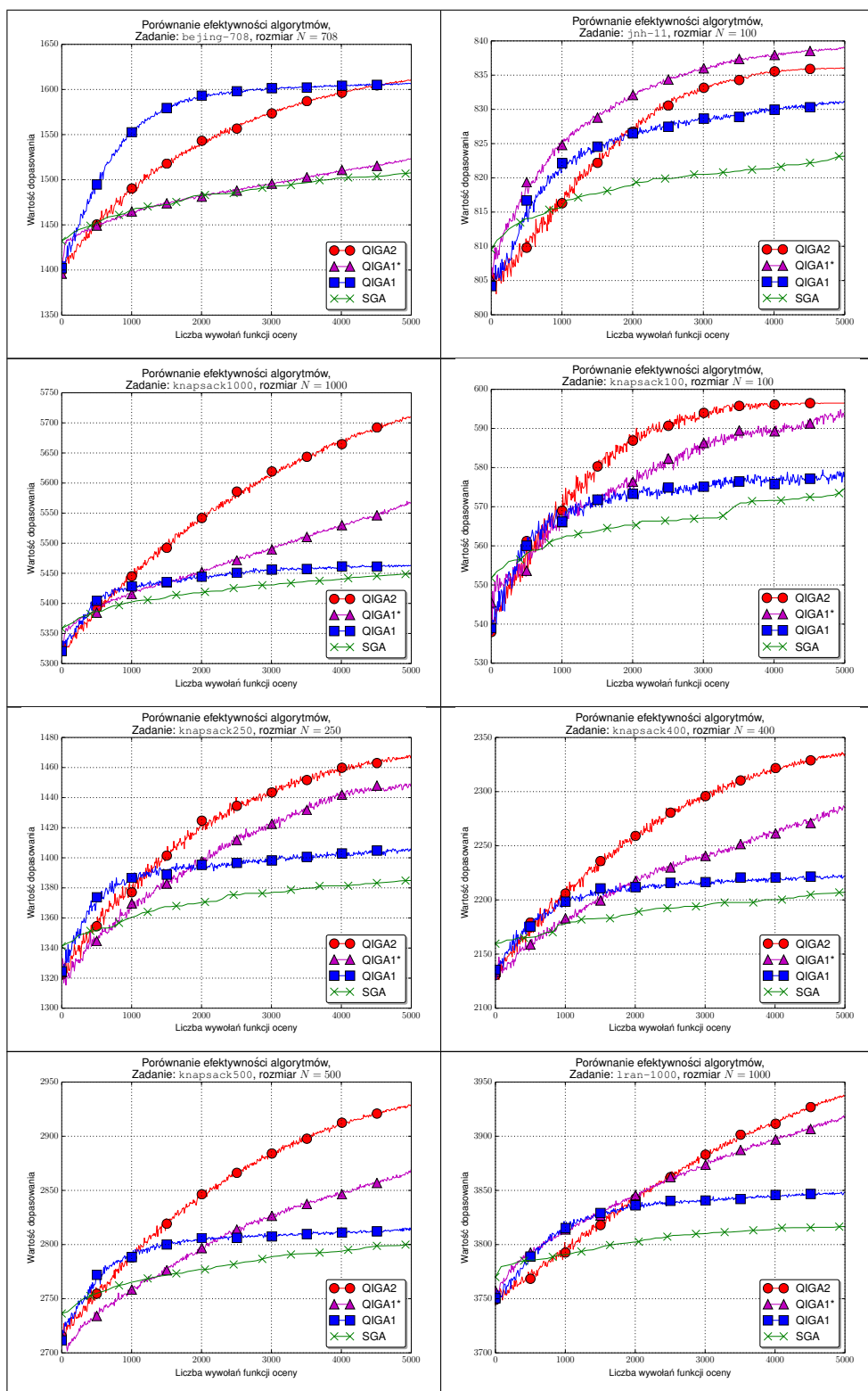
```
132 |
133 |     printf("best solution: ");
134 |     fwrite(best, 1, chromlen, stdout);
135 |     printf("\nfitness: %f\n", bestval);
136 | }
```

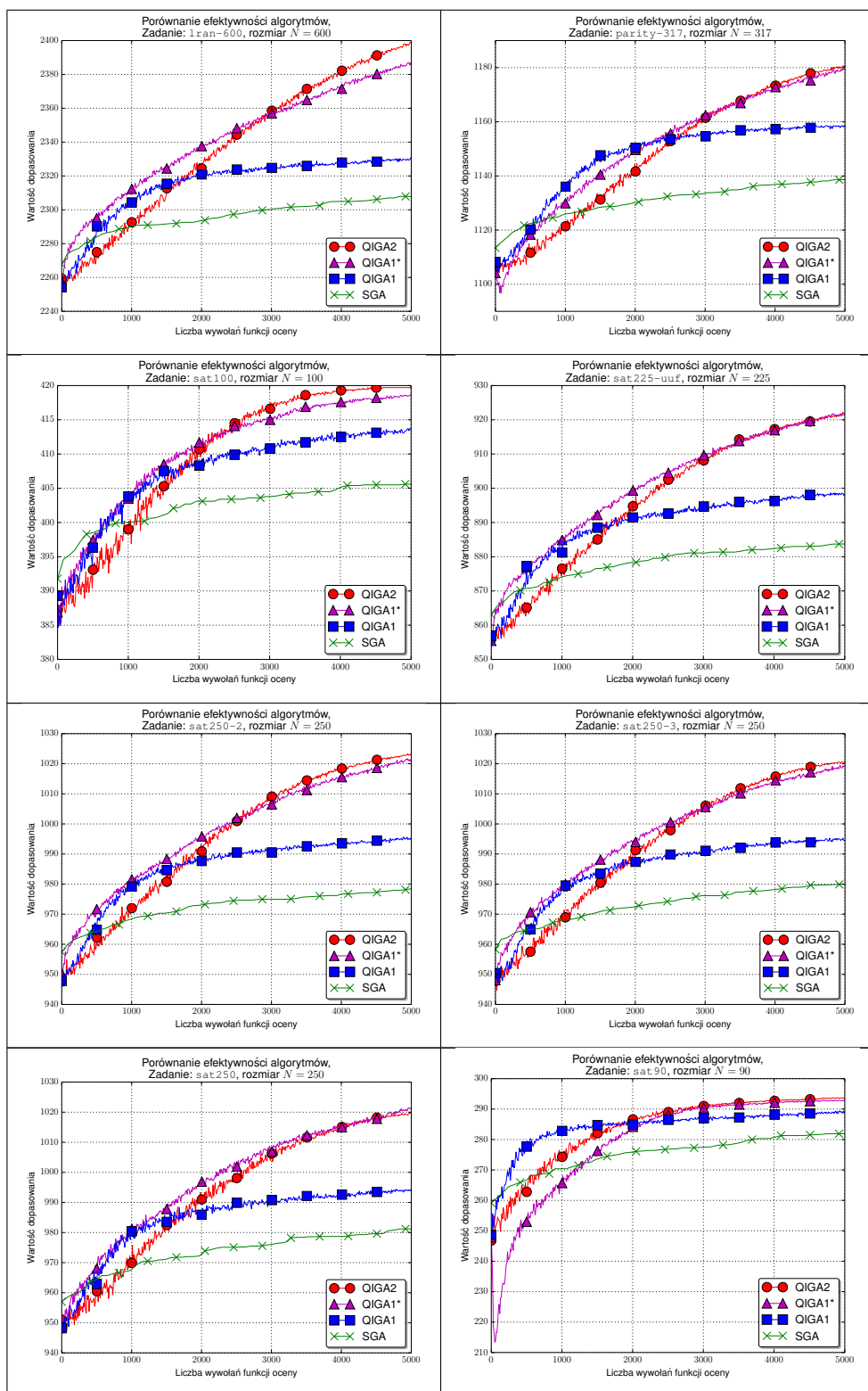
Załącznik D

WYNIKI SZCZEGÓŁOWE BADAŃ DOŚWIADCZALNYCH

TABELA D.1: Porównanie efektywności algorytmów SGA, QIGA1, QIGA1* i QIGA2 dla 20 zadań optymalizacji kombinatorycznej







BIBLIOGRAFIA

- 1) Al-Othman, A., Al-Fares, F., EL-Nagger, K., Power System Security Constrained Economic Dispatch Using Real Coded Quantum Inspired Evolution Algorithm. *International Journal of Electrical, Computer, and Systems Engineering*, 1:4, 2007. (Cyt. na stronie: 57)
- 2) Alfares, F., Esat, I., Real-Coded Quantum Inspired Evolution Algorithm Applied to Engineering Optimization Problems. In *ISOLA '06: Proceedings of the Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*, pages 169–176, Washington, DC, USA, nov. 2006. IEEE Computer Society. ISBN 978-0-7695-3071-0. doi: 10.1109/ISOla.2006.12. (Cyt. na stronie: 57)
- 3) Ali, H., Ashrafinia, S., Liu, J., Lee, D., Wireless Mesh Network Planning Using Quantum Inspired Evolutionary Algorithm. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5, sept. 2011. doi: 10.1109/VETECF.2011.6093270. (Cyt. na stronie: 57)
- 4) Allauddin, R., Boehmer, S., Behrman, E., Gaddam, K., Steck, J., Quantum Simultaneous Recurrent Networks for Content Addressable Memory. *Quantum Inspired Intelligent Systems*, pages 57–76, 2008. (Cyt. na stronie: 10)
- 5) Angeline, P. J., Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In *Evolutionary Programming VII*, pages 601–610. Springer, 1998. (Cyt. na stronie: 141)
- 6) Arabas, J., *Wykłady z algorytmów ewolucyjnych*. Wydawnictwa Naukowo-Techniczne, 2004. (Cyt. na stronie: 19)
- 7) Araujo, M., Nedjah, N., De Macedo Mourelle, L., Designing hardware for finite synchronous state machines using quantum inspired evolution. *International Journal of Innovative Computing and Applications*, 1(4):252–259, 2008a. (Cyt. na stronie: 57)
- 8) Araujo, M., Nedjah, N., Macedo Mourelle, L., Optimised State Assignment for FSMs Using Quantum Inspired Evolutionary Algorithm. In *Proceedings of the 8th international conference on Evolvable Systems: From Biology to Hardware*, page 341. Springer, 2008b. (Cyt. na stronie: 57)
- 9) Araujo, M. P., Nedjah, N., Macedo Mourelle, L., Logic Synthesis for FSMs Using Quantum Inspired Evolution. In *IDEAL '08: Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning*, pages 32–39, Berlin, Heidelberg, 2008c. Springer-Verlag. ISBN 978-3-540-88905-2. doi: http://dx.doi.org/10.1007/978-3-540-88906-9_5. (Cyt. na stronie: 58)

- 10) Araújo, R. d. A., de Oliveira, A. L. I., Soares, S. C. B., Hybrid evolutionary quantum inspired method to adjust time phase distortions in financial time series. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1153–1154, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-639-7. doi: <http://doi.acm.org/10.1145/1774088.1774332>. (Cyt. na stronie: 57)
- 11) Arpaia, P., Meccariello, G., Rapone, M., Zanesco, A., Quantum-Inspired Evolutionary Classification of Driving Sequences in Vehicle Emission Factor Measurement. 2007. (Cyt. na stronie: 58)
- 12) Arpaia, P., Maisto, D., Manna, C., A Quantum-inspired Evolutionary Algorithm with a competitive variation operator for Multiple-Fault Diagnosis. *Applied Soft Computing*, 11(8):4655 – 4666, 2011. ISSN 1568-4946. doi: 10.1016/j.asoc.2011.07.017. (Cyt. na stronach: 57 i 58)
- 13) Ashish Mani, C. P., An Adaptive Quantum Evolutionary Algorithm for Engineering Optimization Problems. *International Journal of Computer Applications*, 1(22):43–48, February 2010. Published By Foundation of Computer Science. (Cyt. na stronach: 56 i 59)
- 14) Bäck, T., *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996. ISBN 0-19-509971-0. (Cyt. na stronie: 26)
- 15) Bansal, S., Algorithm Engineering for solution of some hard Combinatorial Optimization Problems. 2013. (Cyt. na stronie: 161)
- 16) Bao, X., Xu, X., Li, C., Yuan, Z., Lu, C., Pan, J., Quantum teleportation between remote atomic-ensemble quantum memories. *Proceedings of the National Academy of Sciences*, 109(50):20347–20351, 2012. (Cyt. na stronie: 9)
- 17) Baojun, Z., Ruifang, P., Fujun, Y., Analyzing and Improving Load Balancing Algorithm of MooseFS. *International Journal of Grid & Distributed Computing*, 7(4), 2014. (Cyt. na stronie: 143)
- 18) Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., Smith, K., Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39, 2011. (Cyt. na stronach: 145 i 167)
- 19) Bekenstein, J. D., Universal upper bound on the entropy-to-energy ratio for bounded systems. *Physical Review D*, 23(2):287, 1981. (Cyt. na stronie: 118)
- 20) Bell, J. S., others, . On the einstein-podolsky-rosen paradox. *Physics*, 1(3):195–200, 1964. (Cyt. na stronie: 39)

- 21) Bhattacharyya, S., Dutta, P., Chakraborty, S., Chakraborty, R., Dey, S., Determination of optimal threshold of a gray-level image using a quantum inspired genetic algorithm with interference based on a random map model. In *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on*, pages 1–4, dec. 2010. doi: 10.1109/ICCIC.2010.5705806. (Cyt. na stronie: 57)
- 22) Bin, L., Junan, Y., Zhenquan, Z., GAQPR and its application in discovering frequent structures in time series. In *Proceedings of the 2003 IEEE International Conference on Neural Networks & Signal Processing*, volume 1, pages 399–403. Citeseer, 2003. (Cyt. na stronie: 53, 56 i 57)
- 23) Birattari, M., *Tuning metaheuristics: a machine learning perspective*. Springer Verlag, 2009. (Cyt. na stronie: 30)
- 24) Boschi, D., Branca, S., De Martini, F., Hardy, L., Popescu, S., Experimental realization of teleporting an unknown pure quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters*, 80(6):1121, 1998. (Cyt. na stronie: 39)
- 25) Brassard, G., Crépeau, C., Jozsa, R., Langlois, D., A quantum bit commitment scheme provably unbreakable by both parties. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 362–371. IEEE, 1993. (Cyt. na stronie: 39)
- 26) Broyden, C., Quasi-Newton methods and their application to function minimisation. *Mathematics of Computation*, pages 368–381, 1967. (Cyt. na stronie: 21)
- 27) Buras, A., Ellis, J., Gaillard, M., Nanopoulos, D., Aspects of the grand unification of strong, weak and electromagnetic interactions. *Nuclear Physics B*, 135(1):66–92, 1978. (Cyt. na stronie: 8)
- 28) Cao, M., Shang, F., Training of Process Neural Networks Based on Improved Quantum Genetic Algorithm. In *WCSE '09: Proceedings of the 2009 WRI World Congress on Software Engineering*, pages 160–165, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3570-8. doi: <http://dx.doi.org/10.1109/WCSE.2009.127>. (Cyt. na stronie: 56)
- 29) Chaiyaratana, N., Zalzala, A., Recent developments in evolutionary and genetic algorithms: theory and applications. In *IEE conference publication*, pages 270–277. Institution of Electrical Engineers, 1997. (Cyt. na stronie: 27)
- 30) Charan Kumari, A., Srinivas, K., Gupta, M. P., Software Requirements Optimization Using Multi-Objective Quantum-Inspired Hybrid Differential Evolution. In Schütze, O., Coello Coello, C. A., Tantar, A.-A., Tantar, E., Bouvry, P., Del Moral, P., Legrand, P., editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, volume 175 of *Advances in Intelligent Systems and Computing*, pages 107–120. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-31519-0. (Cyt. na stronie: 58)

- 31) Chen, J.-C., Application of Quantum-Inspired Evolutionary Algorithm to Reduce PAPR of an OFDM Signal Using Partial Transmit Sequences Technique. *Broadcasting, IEEE Transactions on*, 56(1):110 –113, march 2010. ISSN 0018-9316. doi: 10.1109/TBC.2009.2036949. (Cyt. na stronie: 58)
- 32) Chen, M., Ding, L., Analysis on the Convergence of Quantum-inspired Evolutionary Algorithms. *IJACT : International Journal of Advancements in Computing Technology*, 3(4): 202–212, 2011. doi: 10.4156/ijact.vol3.issue4.22. (Cyt. na stronach: 53, 54, 55 i 76)
- 33) Chen, R., Huang, Y., Lin, M., Solving Unbounded Knapsack Problem Based on Quantum Genetic Algorithms. *Intelligent Information and Database Systems*, pages 339–349, 2010. (Cyt. na stronie: 58)
- 34) Cho, S.-Y., Lee, J.-W., Lee, J.-J., Type-2 Fuzzy PD Controller Tuning using Quantum-inspired Evolutionary algorithm. In *Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on*, pages 292 –296, 31 2011-june 3 2011. doi: 10.1109/DEST.2011.5936641. (Cyt. na stronie: 58)
- 35) Chung, C., Yu, H., Wong, K. P., An Advanced Quantum-Inspired Evolutionary Algorithm for Unit Commitment. *Power Systems, IEEE Transactions on*, 26(2):847 –854, may 2011. ISSN 0885-8950. doi: 10.1109/TPWRS.2010.2059716. (Cyt. na stronach: 56 i 57)
- 36) Coelho, L., Nedjah, N., Mourelle, L., Gaussian quantum-behaved particle swarm optimization applied to fuzzy pid controller design. *Quantum Inspired Intelligent Systems*, pages 1–15, 2008. (Cyt. na stronach: 11 i 57)
- 37) Colorni, A., Dorigo, M., Maniezzo, V., others, . Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life*, volume 142, pages 134–142, 1991. (Cyt. na stronach: 4 i 19)
- 38) Compton, A., A quantum theory of the scattering of X-rays by light elements. *Physical Review*, 21(5):483, 1923. (Cyt. na stronie: 8)
- 39) da Cruz, A., Vellasco, M., Pacheco, M., Quantum-Inspired Evolutionary Algorithm for Numerical Optimization. pages 2630 –2637, 0-0 2006. doi: 10.1109/CEC.2006.1688637. (Cyt. na stronach: 5, 45, 53, 56, 60, 64, 66, 67, 68, 141, 142 i 147)
- 40) da Cruz, A., Vellasco, M., Pacheco, M., Quantum-Inspired Evolutionary Algorithms applied to numerical optimization problems. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1 –6, july 2010. doi: 10.1109/CEC.2010.5586193. (Cyt. na stronach: 56 i 147)
- 41) da Cruz, A. V. A., Barbosa, C. R. H., Pacheco, M. A. C., Vellasco, M. B. R. . Quantum-Inspired Evolutionary Algorithms and Its Application to Numerical Optimization Problems. In Pal, N. R., Kasabov, N., Mudi, R. K., Pal, S., Parui, S. K., editors, *ICONIP*, volume 3316 of *Lecture Notes in Computer Science*, pages 212–217. Springer, 2004. ISBN 3-540-23931-6. (Cyt. na stronach: 52, 53, 54, 56, 64, 66, 140, 141, 144, 145 i 147)

- 42) da Cruz, A. V. A., Pacheco, M. A. C., Vellasco, M. B. R., Barbosa, C. R. H. . Cultural Operators for a Quantum-Inspired Evolutionary Algorithm Applied to Numerical Optimization Problems. In Mira, J., Álvarez, J. R., editors, *IWINAC (2)*, volume 3562 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2005. ISBN 3-540-26319-5. (Cyt. na stronach: 56 i 58)
- 43) Dai, H., Li, C., Improved Quantum Interference Crossover-Based Genetic Algorithm and its Application. In *Intelligent Networks and Intelligent Systems, 2008. ICINIS'08. First International Workshop on*, pages 35–38, 2008. (Cyt. na stronach: 56 i 58)
- 44) Davidor, Y., Epistasis variance: Suitability of a representation to genetic algorithms. *Complex Systems*, 4(4):369–383, 1990. (Cyt. na stronie: 23)
- 45) De Jong, K., Genetic algorithms are NOT function optimizers. *Foundations of Genetic Algorithms*, 2:5–17, 1993. (Cyt. na stronie: 25)
- 46) De Jong, K., *Evolutionary computation: a unified approach*. 2006. (Cyt. na stronie: 75)
- 47) De Jong, K. A., *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1975. (Cyt. na stronie: 147)
- 48) de Pinho, A., Vellasco, M., da Cruz, A., A new model for credit approval problems: A quantum-inspired neuro-evolutionary algorithm with binary-real representation. pages 445–450, dec. 2009. doi: 10.1109/NABIC.2009.5393327. (Cyt. na stronie: 56)
- 49) Defoin-Platel, M., Schliebs, S., Kasabov, N., A versatile quantum-inspired evolutionary algorithm. In *Proceedings of the International Conference on Evolutionary Computation (Singapore, September 25–28, 2007)*. CEC, volume 7, 2007. (Cyt. na stronie: 53)
- 50) Dias, D., Singulani, A., Pacheco, M., de Souza, P., Pires, M., Neto, O., Self-assembly quantum dots growth prediction by quantum-inspired linear genetic programming. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2075–2082, june 2011. doi: 10.1109/CEC.2011.5949871. (Cyt. na stronie: 59)
- 51) Dias, D. M., Pacheco, M. A. C., Toward a quantum-inspired linear genetic programming model. In *CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, pages 1691–1698. Institute of Electrical and Electronics Engineers Inc., The, 2009. ISBN 978-1-4244-2958-5. (Cyt. na stronie: 57)
- 52) Dias, D. M., Pacheco, M. A. C., Quantum-Inspired Linear Genetic Programming as a Knowledge Management System. *The Computer Journal*, 2012. doi: 10.1093/comjnl/bxs108. (Cyt. na stronie: 58)
- 53) Dorigo, M., Maniezzo, V., Colorni, A., Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1): 29–41, 1996. (Cyt. na stronie: 19)

- 54) dos Santos Nicolau, A., Schirru, R., de Lima, A. M. M., Nuclear reactor reload using Quantum Inspired Algorithm. *Progress in Nuclear Energy*, 55(0):40 – 48, 2012. ISSN 0149-1970. doi: 10.1016/j.pnucene.2011.11.001. (Cyt. na stronie: 57)
- 55) Draa, A., Meshoul, S., Talbi, H., Batouche, M., A Quantum-Inspired Differential Evolution Algorithm for Solving the N-Queens Problem. *Int. Arab J. Inf. Technol.*, 7(1):21–27, 2010. (Cyt. na stronie: 59)
- 56) Eberbach, E., Toward a theory of evolutionary computation. *Biosystems*, 82(1):1 – 19, 2005. ISSN 0303-2647. doi: DOI:10.1016/j.biosystems.2005.05.006. (Cyt. na stronie: 145)
- 57) Eberhart, R., Shi, Y., Kennedy, J., *Swarm intelligence*. Morgan Kaufmann, 2001. (Cyt. na stronie: 9)
- 58) Eiben, A. E., Rudolph, G., Theory of evolutionary algorithms: a bird's eye view. *Theor. Comput. Sci.*, 229(1-2):3–9, 1999. ISSN 0304-3975. doi: [http://dx.doi.org/10.1016/S0304-3975\(99\)00089-4](http://dx.doi.org/10.1016/S0304-3975(99)00089-4). (Cyt. na stronie: 25)
- 59) Einstein, A., *Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt*. JA Barth, 1905. (Cyt. na stronie: 8)
- 60) Einstein, A., Podolsky, B., Rosen, N., Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777, 1935. (Cyt. na stronie: 39)
- 61) Eremeev, A., Modeling and analysis of genetic algorithm with tournament selection. In *Artificial Evolution*, pages 84–95. Springer, 2000. (Cyt. na stronie: 25)
- 62) Fan, K., Brabazon, A., O'Sullivan, C., O'Neill, M., Option pricing model calibration using a real-valued quantum-inspired evolutionary algorithm. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1983–1990, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-697-4. doi: <http://doi.acm.org/10.1145/1276958.1277351>. (Cyt. na stronie: 56 i 57)
- 63) Fan, K., Brabazon, A., O'Sullivan, C., O'Neill, M., Quantum-Inspired Evolutionary Algorithms for Financial Data Analysis. In *EvoWorkshops*, pages 133–143, 2008a. (Cyt. na stronie: 56)
- 64) Fan, K., O'Sullivan, C., Brabazon, A., O'Neill, M., McGarraghy, S., Calibration of the VGSSD Option Pricing Model using a Quantum-inspired Evolutionary Algorithm, 2008b. (Cyt. na stronie: 56)
- 65) Fang, W., Sun, J., Xu, W., A New Mutated Quantum-Behaved Particle Swarm Optimizer for Digital IIR Filter Design. 2009. (Cyt. na stronie: 58)
- 66) Feng, X., Wang, Y., Ge, H., Zhou, C., Liang, Y., Quantum-inspired evolutionary algorithm for travelling salesman problem. In LIU, G., TAN, V., HAN, X., editors, *Computational Methods*, pages 1363–1367. Springer Netherlands, 2006. ISBN 978-1-4020-3953-9. (Cyt. na stronie: 58)

- 67) Feynman, R., Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488, 1982. (Cyt. na stronie: 8)
- 68) Feynman, R. P., *Wykłady o obliczeniach*. Prószyński i S-ka, Warszawa, 1996. ISBN 978-83-7469-606-7. (Cyt. na stronie: 40)
- 69) Fletcher, R., Reeves, C. M., Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964. (Cyt. na stronie: 21)
- 70) Fogel, L. J., Owens, A. J., Walsh, M. J., Artificial intelligence through simulated evolution. 1966. (Cyt. na stronie: 21)
- 71) Fortuna, Z., Macukow, B., Wasowski, J., Metody numeryczne. WNT. Warszawa. Technical report, ISBN 83-204-3075-5, 2005. (Cyt. na stronie: 21)
- 72) Fu, X., Ding, M., Zhou, C., Sun, Y., Multi-threshold image segmentation with improved quantum-inspired genetic algorithm. In *Proceedings of SPIE*, volume 7495, page 749518, 2009. (Cyt. na stronie: 56)
- 73) Gawron, P., Klamka, J., Miszczak, J., Winiarczyk, R., Extending scientific computing system with structural quantum programming capabilities. *BULLETIN OF THE POLISH ACADEMY OF SCIENCES TECHNICAL SCIENCES*, 58(1):1, 2010. (Cyt. na stronach: 35 i 39)
- 74) Gharipour, A., Jazi, A., Sameti, M., Forecast combination with optimized SVM based on quantum-inspired hybrid evolutionary method for complex systems prediction. In *Computational Intelligence for Financial Engineering and Economics (CIFER)*, 2011 IEEE Symposium on, pages 1–6, april 2011. doi: 10.1109/CIFER.2011.5953562. (Cyt. na stronie: 57)
- 75) Giraldi, G., Portugal, R., Thess, R., Genetic algorithms and quantum computation. *Arxiv preprint cs/0403003*, 2004. (Cyt. na stronach: 12, 53 i 54)
- 76) Gödel, K., Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. *Monatshefte für Mathematik*, 38(1):173–198, 1931. (Cyt. na stronie: 10)
- 77) Goldberg, D., *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, 1989. (Cyt. na stronach: 20, 21, 25 i 85)
- 78) Goldberg, D., *Algorytmy genetyczne i ich zastosowania*. Wydawnictwa Naukowo-Techniczne, 1998. (Cyt. na stronach: 19, 22, 24, 26 i 27)
- 79) Gong, C.-Q., Zhang, B., Li, Y., Resources Scheduling of TT&C Network Based on Quantum Genetic Algorithm. pages 1–4, sept. 2009. doi: 10.1109/WICOM.2009.5302019. (Cyt. na stronie: 57)

- 80) Grefenstette, J. J., Optimization of control parameters for genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(1):122–128, 1986. (Cyt. na stronie: 28)
- 81) Grover, L. K., A fast quantum mechanical algorithm for database search. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, New York, NY, USA, 1996. ACM Press. (Cyt. na stronach: 9, 32 i 141)
- 82) Grygiel, W., Hohol, M., Rogera Penrose’a kwantowanie umysłu. *Filozofia nauki*, 3:67, 2009. (Cyt. na stronie: 10)
- 83) Gu, J., Cao, C., Jiao, B., Gu, X., An improved quantum genetic algorithm for stochastic flexible scheduling problem with breakdown. In *GEC '09: Proceedings of the first ACM/SIGE-VO Summit on Genetic and Evolutionary Computation*, pages 163–170, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-326-6. doi: <http://doi.acm.org/10.1145/1543834.1543858>. (Cyt. na stronie: 57)
- 84) Gu, J., Gu, M., Cao, C., Gu, X., A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. *Computers and Operations Research*, 2009b. (Cyt. na stronach: 56 i 59)
- 85) Gu, J., Gu, X., Gu, M., A novel parallel quantum genetic algorithm for stochastic job shop scheduling. *Journal of Mathematical Analysis and Applications*, 355(1):63–81, 2009c. (Cyt. na stronie: 57)
- 86) Guo, J., Juan Sun, L., Chuan Wang, R., Gen Yu, Z., An Improved Quantum Genetic Algorithm. In *Genetic and Evolutionary Computing, 2009. WGECC '09. 3rd International Conference on*, pages 14–18, Oct. 2009a. (Cyt. na stronie: 58)
- 87) Guo, R., Li, B., Zou, Y., Zhuang, Z., Hybrid quantum probabilistic coding genetic algorithm for large scale hardware-software co-synthesis of embedded systems. In *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, pages 3454–3458, 2007. (Cyt. na stronie: 58)
- 88) Guo, X., Wang, T., Chen, Z., Wang, L., Zhao, W., Fast FPGA placement algorithm using Quantum Genetic Algorithm with Simulated Annealing. In *IEEE 8th International Conference on ASIC, 2009. ASICON'09*, pages 730–733, 2009b. (Cyt. na stronie: 58)
- 89) Gwiazda, T., Algorytmy genetyczne Kompendium t. 1. PWN/Mikom, Warszawa, 2009. (Cyt. na stronie: 19)
- 90) Hadad, B. S., Eick, C. F., Supporting polyploidy in genetic algorithms using dominance vectors. In *Evolutionary Programming VI*, pages 223–234. Springer, 1997. (Cyt. na stronie: 22)
- 91) Haijun, Y., Minqiang, L., Jisong, K., Exact schema theorem based on the space of schema. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 1, pages 349–354 vol.1, oct. 2003. doi: 10.1109/ICSMC.2003.1243840. (Cyt. na stronie: 26)

- 92) Hamed, A., Nuzly, H., Kasabov, N., Michlovský, Z., Shamsuddin, S., String Pattern Recognition Using Evolving Spiking Neural Networks and Quantum Inspired Particle Swarm Optimization. In *Proceedings of the 16th International Conference on Neural Information Processing: Part II*, pages 611–619. Springer, 2009. (Cyt. na stronie: 58)
- 93) Hameroff, S., Consciousness, neurobiology and quantum mechanics: The case for a connection. *The emerging physics of consciousness*, pages 193–253, 2006. (Cyt. na stronie: 10)
- 94) Han, K., Kim, J., Genetic quantum algorithm and its application to combinatorial optimization problem. In *Proceedings of the 2000 Congress on Evolutionary computation*, volume 2, pages 1354–1360. Citeseer, 2000. (Cyt. na stronach: 5, 13, 45, 49, 52, 53, 60, 61, 90, 129, 130, 133 i 136)
- 95) Han, K., Kim, J., Analysis of Quantum-Inspired Evolutionary Algorithm. In *Proceedings of the 2001 International Conference on Artificial Intelligence*, pages 727–730, 2001. (Cyt. na stronach: 52 i 53)
- 96) Han, K., Kim, J., Introduction of Quantum-Inspired Evolutionary Algorithm. In *Proc of the FIRA Robot World Congress. Seoul, Korea*, pages 101–106, 2002a. (Cyt. na stronie: 52)
- 97) Han, K., Kim, J., Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE transactions on evolutionary computation*, 6(6):580–593, 2002b. (Cyt. na stronie: 52)
- 98) Han, K., Kim, J., On setting the parameters of quantum-inspired evolutionary algorithm for practical applications. In *Proceedings of IEEE Congress on Evolutionary Computing*, pages 178–184. Citeseer, 2003a. (Cyt. na stronie: 53)
- 99) Han, K., Kim, J., Quantum-inspired evolutionary algorithms with a new termination criterion, H_c Gate, and two-phase scheme. *IEEE transactions on evolutionary computation*, 8(2):156–169, 2004. (Cyt. na stronach: 53 i 56)
- 100) Han, K., Kim, J., On the analysis of the quantum-inspired evolutionary algorithm with a single individual. In *IEEE Congress on Evolutionary Computation*, pages 16–21, 2006. (Cyt. na stronach: 53 i 55)
- 101) Han, K., Park, K., Lee, C., Kim, J., Parallel quantum-inspired genetic algorithm for combinatorial optimization problem. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 2, pages 1422–1429. Citeseer, 2001. (Cyt. na stronie: 52)
- 102) Han, K.-H., *Quantum-inspired Evolutionary Algorithm*. PhD thesis, Korea Advanced Institute of Science and Technology (KAIST), 2003. (Cyt. na stronie: 53)
- 103) Han, K.-H., Kim, J.-H., On Setting the Parameters of QEA for Practical Applications: Some Guidelines Based on Empirical Evidence, 2003b. (Cyt. na stronie: 53)

- 104) Hansen, N., Kern, S., Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 282–291. Springer, 2004. (Cyt. na stronach: 21 i 145)
- 105) Hansen, N., Ostermeier, A., Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 312–317. IEEE, 1996. (Cyt. na stronach: 4, 21, 142 i 145)
- 106) Hao, L., Shiyong, L., Quantum particle swarm evolutionary algorithm with application to system identification. In *Measurement, Information and Control (MIC), 2012 International Conference on*, volume 2, pages 1032–1036, may 2012. doi: 10.1109/MIC.2012.6273477. (Cyt. na stronie: 57)
- 107) Hao, T., Bao-hua, Z., Shi-xian, W., Chaos quantum genetic algorithm based on Tent map. volume 4, pages V4–403–V4–406, apr. 2010. doi: 10.1109/ICCET.2010.5485510. (Cyt. na stronie: 56)
- 108) Hawking, S., *Public Lectures: Does God Play Dice?*, 2007. (Cyt. na stronie: 8)
- 109) Higgs, P., Broken symmetries and the masses of gauge bosons. *Physical Review Letters*, 13(16):508, 1964. (Cyt. na stronie: 8)
- 110) Hirvensalo, M., *Algorytmy kwantowe*. WSiP, Warszawa, 2004. ISBN 83-02-09155-3. (Cyt. na stronach: 9, 31, 36, 39, 41 i 42)
- 111) Holland, J., *Adaptation in natural and artificial system: an introduction with application to biology, control and artificial intelligence*. Ann Arbor, University of Michigan Press, 1975. (Cyt. na stronach: 4, 9, 21, 25, 26, 77, 86, 87, 133, 141 i 145)
- 112) Hooke, R., Jeeves, T. A., “Direct Search” Solution of Numerical and Statistical Problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961. (Cyt. na stronie: 21)
- 113) Hoos, H., Stützle, T., SATLIB: An Online Resource for Research on SAT. *Sat2000: highlights of satisfiability research in the year 2000*, page 283, 2000. (Cyt. na stronie: 122)
- 114) Hossain, M. A., Hossain, M. K., Hashem, M. M. A., A generalized hybrid real-coded quantum evolutionary algorithm based on particle swarm theory with arithmetic crossover, 2005. (Cyt. na stronach: 58 i 59)
- 115) Hu, J., Goodman, E., Robust and efficient genetic algorithms with hierarchical niching and a sustainable evolutionary computation model. In *Genetic and Evolutionary Computation–GECCO 2004*, pages 1220–1232. Springer, 2004. (Cyt. na stronie: 22)
- 116) Hu, W., *Cryptanalysis of TEA Using Quantum-Inspired Genetic Algorithms*. 2010. (Cyt. na stronie: 58)

- 117) Hui, C., Jiashu, Z., Chao, Z., Chaos updating rotated gates quantum-inspired genetic algorithm. In *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on*, volume 2, 2004. (Cyt. na stronie: 56)
- 118) Huo, H., Xie, Q., Shen, X., Stojkovic, V., A Probabilistic Coding Based Quantum Genetic Algorithm for Multiple Sequence Alignment. In *Computational systems bioinformatics/Life Sciences Society. Computational Systems Bioinformatics Conference*, volume 7, page 15, 2008. (Cyt. na stronie: 56)
- 119) Ibrahim, A., Mohamed, A., Shareef, H., Ghoshal, S., An effective power quality monitor placement method utilizing quantum-inspired particle swarm optimization. In *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, pages 1–6, july 2011. doi: 10.1109/ICEEI.2011.6021845. (Cyt. na stronie: 57)
- 120) Imabeppu, T., Nakayama, S., Ono, S., A study on a quantum-inspired evolutionary algorithm based on pair swap. *Artificial Life and Robotics*, 12(1):148–152, 2008. (Cyt. na stronach: 56 i 58)
- 121) Imabeppu, T., Ono, S., Morishige, R., Kurose, M., Nakayama, S., A Comparative Study between Migration and Pair-Swap on Quantum-Inspired Evolutionary Algorithm. *Transactions of the Japanese Society for Artificial Intelligence*, 24:250–262, 2009. (Cyt. na stronach: 56 i 58)
- 122) Izadinia, H., Ebadzadeh, M., Quantum-Inspired Evolution Strategy. pages 724–727, dec. 2009. doi: 10.1109/SoCPaR.2009.146. (Cyt. na stronie: 56)
- 123) Izadinia, H., Ebadzadeh, M., Adaptive Quantum-inspired Evolution Strategy. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8, june 2012. doi: 10.1109/CEC.2012.6256433. (Cyt. na stronach: 55 i 59)
- 124) Jang, J., Han, K., Kim, J., Quantum-inspired evolutionary algorithm-based face verification. *Lecture Notes in Computer Science*, pages 2147–2156, 2003. (Cyt. na stronie: 57)
- 125) Jang, J.-S., Han, K.-H., Kim, J.-H., Face detection using quantum-inspired evolutionary algorithm. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 2100–2106 o.2, June 2004. (Cyt. na stronie: 57)
- 126) JEONG, Y., PARK, J., SHIN, J., LEE, K., A Thermal Unit Commitment Approach Using an Improved Quantum Evolutionary Algorithm. *Electric Power Components and Systems*, 37(7):770–786, 2009. (Cyt. na stronie: 56)
- 127) Jeżewski, S., Łaski, M., Nowotniak, R., Comparison of algorithms for simultaneous localization and mapping problem for mobile robot. *Sci. Bull. Ac. Sci. and Technology, Automatics*, 14:439–452, 2010. ISSN 1429-3447. (Cyt. na stronie: 161)

- 128) Jiao, L., Li, Y., Gong, M., Zhang, X., Quantum-inspired immune clonal algorithm for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(5): 1234–1253, 2008. (Cyt. na stronie: 58)
- 129) Jopek, L., Nowotniak, R., Postolski, M., Babout, L., Janaszewski, M., Zastosowanie kwantowych algorytmów genetycznych do selekcji cech. *Sci. Bull. Ac. Sci. and Technology, Automatics*, 13(3/2):1219–1231, 2009. ISSN 1429-3447. (Cyt. na stronach: 54, 57 i 161)
- 130) Kennedy, J., Eberhart, R., Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995. (Cyt. na stronach: 4, 21 i 141)
- 131) Khan, M., Akter, S., Multiple-case outlier detection in least-squares regression model using quantum-inspired evolutionary algorithm. In *Computers and Information Technology, 2009. ICCIT '09. 12th International Conference on*, pages 7 –12, dec. 2009. doi: 10.1109/ICCIT.2009.5407176. (Cyt. na stronie: 58)
- 132) Khorsand, A.-R., Akbarzadeh-T, M.-R., Quantum gate optimization in a meta-level genetic quantum algorithm. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 4, pages 3055 – 3062 Vol. 4, oct. 2005. doi: 10.1109/ICSMC.2005.1571615. (Cyt. na stronie: 54)
- 133) Khosraviani, M., Pour-Mozafari, S., Ebadzadeh, M., Convergence analysis of quantum-inspired genetic algorithms with the population of a single individual. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1115–1116. ACM, 2008. (Cyt. na stronach: 53, 54, 55 i 76)
- 134) Kim, K., Hwang, J., Han, K., Kim, J., Park, K., A quantum-inspired evolutionary computing algorithm for disk allocation method. *IEICE Transactions on Information and Systems*, 86:645–649, 2003. (Cyt. na stronach: 52 i 57)
- 135) Kim, Y., Kim, J., Han, K., Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 knapsack problems. In *2006 IEEE Congress on Evolutionary Computation*. Citeseer, 2006. (Cyt. na stronie: 58)
- 136) Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., others, . Optimization by simulated annealing. *science*, 220(4598):671–680, 1983. (Cyt. na stronie: 21)
- 137) Koppaka, S., Hota, A. R., Superior Exploration-Exploitation Balance with Quantum-Inspired Hadamard Walks. *Arxiv preprint cs/1004.0514v1*, 2010. (Cyt. na stronach: 58 i 59)
- 138) Kotowski, S., *Analiza algorytmów genetycznych jako układów dynamicznych*. PhD thesis, 2008. (Cyt. na stronie: 25)
- 139) Koza, J. R., Evolution and co-evolution of computer programs to control independently-acting agents. *Meyer and Wilson*, 1568:366–375, 1991. (Cyt. na stronie: 22)

- 140) Koza, J. R., Genetic programming: on the programming of computers by means of natural selection (complex adaptive systems), 1992. (Cyt. na stronie: 21)
- 141) Kumari, A., Srinivas, K., Gupta, M., Software requirements selection using Quantum-inspired Elitist Multi-objective Evolutionary algorithm. In *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*, pages 782–787, march 2012. (Cyt. na stronie: 59)
- 142) Kurzweil, R., *How to Create a Mind: The Secret of Human Thought Revealed*. Viking Adult, 2012. (Cyt. na stronie: 9)
- 143) Kusiak, J., Danielewska-Tulecka, A., Oprocha, P., *Optymalizacja: wybrane metody z przykładami zastosowań*. Wydawnictwo Naukowe PWN, 2009. (Cyt. na stronach: 21 i 86)
- 144) Kusztełak, G., Rudnicki, M., Wiak, S., Propagation of Building Blocks in SGA and MPGA. *Lecture notes in computer science*, pages 438–443, 2004. (Cyt. na stronach: 26 i 82)
- 145) Laplace, P.-S., *A Philosophical Essay on Probabilities*. 1814. (Cyt. na stronie: 8)
- 146) Layeb, A., Saidouni, D., Quantum Genetic Algorithm for Binary Decision Diagram Ordering Problem. *IJCSNS*, 7(9):130, 2007. (Cyt. na stronie: 56)
- 147) Layeb, A., Saidouni, D.-E., A New Quantum Evolutionary Local Search Algorithm for MAX 3-SAT Problem. In *HAIS '08: Proceedings of the 3rd international workshop on Hybrid Artificial Intelligence Systems*, pages 172–179, Berlin, Heidelberg, 2008a. Springer-Verlag. ISBN 978-3-540-87655-7. doi: http://dx.doi.org/10.1007/978-3-540-87656-4_22. (Cyt. na stronie: 58)
- 148) Layeb, A., Saidouni, D. E., A quantum genetic algorithm with hill climbing algorithm for max 3-SAT problems. *Journal of Theoretical and Applied Information Technology*, 4(11), 2008b. (Cyt. na stronie: 58)
- 149) Layeb, A., Meshoul, S., Batouhe, M., Multiple Sequence Alignment by Quantum Genetic Algorithm. In *The 7th International Workshop on Parallel and Distributed Scientific and Engineering Computing of the 20th International Parallel and Distributed Processing Symposium, Greece*, pages 1–8, 2006. (Cyt. na stronie: 56)
- 150) Li, B., Wang, L., A hybrid quantum-inspired genetic algorithm for multi-objective scheduling. *Lecture Notes in Computer Science*, 4113:511, 2006. (Cyt. na stronie: 58)
- 151) Li, B., Zhou, Z., Zou, W., Li, D., Quantum Memetic Evolutionary Algorithm-Based Low-Complexity Signal Detection for Underwater Acoustic Sensor Networks. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(5):626–640, sept. 2012. ISSN 1094-6977. doi: 10.1109/TSMCC.2011.2176486. (Cyt. na stronie: 58)

- 152) Li, F., Zhou, M., Li, H., A novel neural network optimized by Quantum Genetic Algorithm for signal detection in MIMO-OFDM systems. In *Computational Intelligence in Control and Automation (CICA), 2011 IEEE Symposium on*, pages 170 –177, april 2011a. doi: 10.1109/CICA.2011.5945763. (Cyt. na stronie: 57)
- 153) Li, P., Li, S., Quantum-inspired evolutionary algorithm for continuous space optimization based on Bloch coordinates of qubits. *Neurocomputing*, 72(1-3):581–591, 2008. (Cyt. na stronie: 56)
- 154) Li, P., Song, K., Shang, F., Double chains quantum genetic algorithm with application to neuro-fuzzy controller design. *Advances in Engineering Software*, 42(10):875 – 886, 2011b. ISSN 0965-9978. doi: DOI:10.1016/j.advengsoft.2011.06.006. (Cyt. na stronach: 53, 55, 56, 57 i 140)
- 155) Li, Y., Zhang, Y., Zhao, R., Jiao, L., The immune quantum-inspired evolutionary algorithm. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, 2004. (Cyt. na stronie: 58)
- 156) Li, Y., Zhao, J., Jiao, L., Wu, Q., Quantum-inspired evolutionary multicast algorithm. In *SMC'09: Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, pages 1496–1501, Piscataway, NJ, USA, 2009a. IEEE Press. ISBN 978-1-4244-2793-2. (Cyt. na stronie: 57)
- 157) Li, Y., Wu, N., Liu, R., Improved quantum-inspired immune clonal clustering algorithm applied to SAR image segmentation. In *Radar (Radar), 2011 IEEE CIE International Conference on*, volume 2, pages 1566 –1569, oct. 2011c. doi: 10.1109/CIE-Radar.2011.6159862. (Cyt. na stronie: 11)
- 158) Li, Z., Rudolph, G., Li, K., Convergence performance comparison of quantum-inspired multi-objective evolutionary algorithms. *Computers and Mathematics with Applications*, 57 (11-12):1843–1854, 2009b. (Cyt. na stronach: 53 i 58)
- 159) Liang, J., Qu, B., Suganthan, P., Hernández-Díaz, A. G., Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, 201212, 2013. (Cyt. na stronach: 144, 146, 148 i 152)
- 160) Liao, R., Wang, X., Qin, Z., A Novel Quantum-Inspired Genetic Algorithm with Expanded Solution Space. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2010 2nd International Conference on*, volume 2, pages 192 –195, aug. 2010. doi: 10.1109/IHMSC.2010.148. (Cyt. na stronie: 56)
- 161) Lin, D., Waller, S., A quantum-inspired genetic algorithm for dynamic continuous network design problem. *Transportation Letters: The International Journal of Transportation Research*, 1(1):81–93, 2009. (Cyt. na stronie: 56)

- 162) Lin, G., A novel real-coded quantum-inspired genetic algorithm and its application in data reconciliation. In *Control and Decision Conference (CCDC), 2011 Chinese*, pages 1450–1453, may 2011. doi: 10.1109/CCDC.2011.5968420. (Cyt. na stronie: 140)
- 163) Liu, C., Zhang, G., Zhu, Y., Fang, C., Liu, H., A quantum-inspired evolutionary algorithm based on P systems for radar emitter signals. In *Proc. Fourth International Conference on Bio-Inspired Computing: Theories and Applications*, pages 24–28, 2009. (Cyt. na stronie: 57)
- 164) Liu, H., Zhang, G., Liu, C., Fang, C., A novel Memetic Algorithm based on real-observation Quantum-inspired evolutionary algorithms. In *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, volume 1, pages 486–490, nov. 2008. doi: 10.1109/ISKE.2008.4730980. (Cyt. na stronach: 53 i 140)
- 165) Liu, S., You, X., Self-organizing Quantum Evolutionary Algorithm Based on Quantum Dynamic Mechanism. In *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence*, pages 69–77. Springer, 2009. (Cyt. na stronach: 58 i 59)
- 166) Liu, W., Chen, H., Yan, Q., Liu, Z., Xu, J., Zheng, Y., A novel quantum-inspired evolutionary algorithm based on variable angle-distance rotation. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–7, july 2010. doi: 10.1109/CEC.2010.5586281. (Cyt. na stronach: 56 i 58)
- 167) Lu, T.-C., Juang, J.-C., Quantum-inspired space search algorithm (QSSA) for global numerical optimization. *Applied Mathematics and Computation*, 218(6):2516 – 2532, 2011. ISSN 0096-3003. doi: 10.1016/j.amc.2011.07.067. (Cyt. na stronie: 140)
- 168) Lucas, J., Minds, machines and Gödel. *Philosophy*, pages 112–127, 1961. (Cyt. na stronie: 10)
- 169) Luitel, B., Venayagamoorthy, G., Quantum inspired PSO for the optimization of simultaneous recurrent neural networks as MIMO learning systems. *Neural Networks*, 2010. (Cyt. na stronach: 11 i 58)
- 170) Luke, S., *Essentials of Metaheuristics*. 2009. available at <http://cs.gmu.edu/~sean/book/metaheuristics/>. (Cyt. na stronach: 28, 29, 144 i 147)
- 171) Luo, Z., Wang, P., Li, Y., Zhang, W., Tang, W., Xiang, M., Quantum-inspired evolutionary tuning of SVM parameters. *Progress in Natural Science*, 18(4):475–480, 2008. (Cyt. na stronie: 58)
- 172) Lv, Y., Li, D., Improved Quantum Genetic Algorithm and Its Application in Nutritional Diet Optimization. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, volume 1, 2008. (Cyt. na stronach: 56 i 58)

- 173) Lv, Y., Liu, N., Application of quantum genetic algorithm on finding minimal reduct. In *IEEE International Conference on Granular Computing, 2007. GRC 2007*, pages 728–728, 2007. (Cyt. na stronie: 58)
- 174) Mahdabi, P., Jalili, S., Abadi, M., A multi-start quantum-inspired evolutionary algorithm for solving combinatorial optimization problems. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 613–614. ACM, 2008. (Cyt. na stronie: 56)
- 175) Mahdabi, P., Abadi, M., Jalili, S., A novel quantum-inspired evolutionary algorithm for solving combinatorial optimization problems. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1807–1808. ACM, 2009. (Cyt. na stronach: 53 i 58)
- 176) Malossini, A., Blanzieri, E., Calarco, T., QGA, A Quantum Genetic Algorithm. 2004. (Cyt. na stronie: 12)
- 177) Malossini, A., Blanzieri, E., Calarco, T., Quantum Genetic Optimization. *IEEE Transactions on Evolutionary Computation*, 12(2):231–241, 2008. (Cyt. na stronach: 55 i 58)
- 178) Mani, A., Patvardhan, C., Solving Ceramic Grinding Optimization Problem by Adaptive Quantum Evolutionary Algorithm. *Intelligent Systems, Modelling and Simulation, International Conference on*, 0:43–48, 2010. doi: <http://doi.ieeecomputersociety.org/10.1109/ISMS.2010.19>. (Cyt. na stronie: 59)
- 179) Manju, A., Nigam, M., Applications of quantum inspired computational intelligence: a survey. *Artificial Intelligence Review*, pages 1–78, 2012. ISSN 0269-2821. 10.1007/s10462-012-9330-6. (Cyt. na stronach: 10, 11, 13, 53, 55, 77, 84 i 161)
- 180) Matsumoto, M., Nishimura, T., Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998. (Cyt. na stronie: 144)
- 181) Melo, L. M., Costa, G. A. O. P., Feitosa, R. Q., da Cruz, A., Vargas, A., Quantum-inspired Evolutionary Algorithm and Differential Evolution Used in The Adaptation of Segmentation Parameters. 2008. (Cyt. na stronie: 57)
- 182) Menneer, T., Narayanan, A., Quantum-inspired neural networks. *Department of Computer Science, University of Exeter, Exeter, United Kingdom, Technical Report R*, 329:1995, 1995. (Cyt. na stronach: 10, 52 i 53)
- 183) Mercer, R. E., Sampson, J., Adaptive search using a reproductive meta-plan. *Kybernetes*, 7(3):215–228, 1978. (Cyt. na stronie: 28)
- 184) Metropolis, N., The beginning of the Monte Carlo method. *Los Alamos Science*, 15(584): 125–130, 1987. (Cyt. na stronie: 90)

- 185) Miao, H., Wang, H., Deng, Z., Quantum Genetic Algorithm and its Application in Power System Reactive Power Optimization. In *Computational Intelligence and Security, 2009. CIS '09. International Conference on*, volume 1, pages 107–111, Dec. 2009. doi: 10.1109/CIS.2009.133. (Cyt. na stronie: 57)
- 186) Michalewicz, Z., *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. Wydawnictwa Naukowo-Techniczne WNT, Warszawa, 2003. (Cyt. na stronach: 19, 21, 22, 25, 71, 73, 75, 90 i 121)
- 187) Michalewicz, Z., Fogel, D., *Jak to rozwiązać czyli nowoczesna heurystyka*. Wydawnictwa Naukowo-Techniczne, 2006. (Cyt. na stronach: 19 i 121)
- 188) Michalewicz, Z., Nazhiyath, G., Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In *Evolutionary Computation, 1995., IEEE International Conference on*, volume 2, pages 647–651. IEEE, 1995. (Cyt. na stronie: 22)
- 189) Ming, L., Wang, Y.-P., ming Cheung, Y., A new schema theorem for uniform crossover based on ternary representation. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004*, pages 235 – 239, dec. 2004. doi: 10.1109/ISSNIP.2004.1417468. (Cyt. na stronie: 26)
- 190) Minsky, M., *Computation: Finite and Infinite Machines*, 1967. (Cyt. na stronie: 10)
- 191) Minsky, M., *Society of mind*. Simon & Schuster, 1988. (Cyt. na stronie: 10)
- 192) Moore, M., Narayanan, A., Quantum-inspired computing. *Dept. Comput. Sci., Univ. Exeter, Exeter, UK*, 1995. (Cyt. na stronie: 52)
- 193) Moravec, H., *Mind children: The future of human and robot intelligence*, 1988. (Cyt. na stronie: 10)
- 194) Moscato, P., others, . On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826:1989, 1989. (Cyt. na stronie: 22)
- 195) Muramoto, N., Matsui, N., Isokawa, T., Searching ability of qubit-inspired genetic algorithm. In *SICE Annual Conference (SICE), 2012 Proceedings of*, pages 443 –446, aug. 2012. (Cyt. na stronie: 58)
- 196) Narayan, A., Patvardhan, C., A novel quantum evolutionary algorithm for quadratic knapsack problem. In *SMC'09: Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, pages 1388–1392, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2793-2. (Cyt. na stronach: 56 i 58)

- 197) Narayanan, A., Moore, M., Quantum-inspired genetic algorithms. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 61–66, 1996. (Cyt. na stronach: 13, 52 i 53)
- 198) Naudts, B., Verschoren, A., others, . Epistasis and deceptivity. *Simon Stevin-Bulletin of the Belgian Mathematical Society*, 6:147–154, 1999. (Cyt. na stronie: 24)
- 199) Nedjah, N., dos Santos Coelho, L., de Macedo Mourelle, L., *Quantum inspired intelligent systems*. Springer Verlag, 2008. (Cyt. na stronie: 11)
- 200) Nelder, J. A., Mead, R., A simplex method for function minimization. *Computer journal*, 7(4):308–313, 1965. (Cyt. na stronach: 4, 21, 141 i 145)
- 201) Neto, J. X. V., de Andrade Bernert, D. L., dos Santos Coelho, L., Improved quantum-inspired evolutionary algorithm with diversity information applied to economic dispatch problem with prohibited operating zones. *Energy Conversion and Management*, In Press, Corrected Proof:–, 2010. ISSN 0196-8904. doi: DOI:10.1016/j.enconman.2010.05.023. (Cyt. na stronie: 57)
- 202) Neto, J. X. V., de Andrade Bernert, D. L., dos Santos Coelho, L., Improved quantum-inspired evolutionary algorithm with diversity information applied to economic dispatch problem with prohibited operating zones. *Energy Conversion and Management*, 52(1):8 – 14, 2011. ISSN 0196-8904. doi: 10.1016/j.enconman.2010.05.023. (Cyt. na stronie: 57)
- 203) Neubauer, A., Intrinsic System Model of the Genetic Algorithm with α -Selection. *Parallel Problem Solving from Nature–PPSN X*, pages 940–949, 2008. (Cyt. na stronie: 25)
- 204) Nielsen, M. A., Chuang, I. L., *Quantum Computation and Quantum Information*. Cambridge University Press, October 2000. ISBN 521635039. (Cyt. na stronach: 9, 31, 33 i 42)
- 205) Niu, Q., Zhou, F., Zhou, T., Quantum Genetic Algorithm for Hybrid Flow Shop Scheduling Problems to Minimize Total Completion Time. In Li, K., Fei, M., Jia, L., Irwin, G., editors, *Life System Modeling and Intelligent Computing*, volume 6329 of *Lecture Notes in Computer Science*, pages 21–29. Springer Berlin / Heidelberg, 2010. (Cyt. na stronie: 57)
- 206) Novaes, S., Standard model: An introduction. *arXiv preprint hep-ph/0001283*, 2000. (Cyt. na stronie: 8)
- 207) Nowotniak, R., Wykorzystanie metod ewolucyjnych w projektowaniu algorytmów kwantowych. Master's thesis, Instytut Informatyki, Politechnika Łódzka, 2008. (Cyt. na stronach: 12, 35 i 39)
- 208) Nowotniak, R., Survey of Quantum-Inspired Evolutionary Algorithms. In *Materiały konferencyjne Forum Innowacji Młodych Badaczy*, 2010. ISBN 2082-4831. (Cyt. na stronach: 12, 13 i 15)

- 209) Nowotniak, R., Meta-optimization of Quantum-Inspired Evolutionary Algorithms in The Polish Grid Infrastructure. In *Materiały konferencyjne II Wyjazdowej Sesji Naukowej Doktorantów PŁ*, 2012. ISBN 978-83-7283-490-4. (Cyt. na stronach: 55, 143, 144 i 160)
- 210) Nowotniak, R., Kucharski, J., Building blocks propagation in quantum-inspired genetic algorithm. *Sci. Bull. Ac. Sci. and Technology, Automatics*, 14, 2010a. ISSN 1429-3447. (Cyt. na stronach: 15, 54, 56, 77, 84 i 159)
- 211) Nowotniak, R., Kucharski, J., Meta-optimization of Quantum-Inspired Evolutionary Algorithm. In *XVII International Conference on Information Technology Systems, 2010 Proceedings of*, 2010b. ISBN 978-83-7283-378-5. (Cyt. na stronach: 15, 16, 54, 64, 144 i 160)
- 212) Nowotniak, R., Kucharski, J., GPU-based tuning of quantum-inspired genetic algorithm for a combinatorial optimization problem. In *XIV International Conference SYSTEM MODELING and CONTROL, 2012 Proceedings of*, 2011a. ISBN 978-83-927875-1-8. (Cyt. na stronach: 15 i 160)
- 213) Nowotniak, R., Kucharski, J., GPU-based massively parallel implementation of meta-heuristic algorithms. *Sci. Bull. Ac. Sci. and Technology, Automatics*, 15(3):595–611, 2011b. ISSN 1429-3447. (Cyt. na stronach: 15, 120, 125, 126 i 160)
- 214) Nowotniak, R., Kucharski, J., Convergence Analysis of Quantum-Inspired Evolutionary Algorithms Based on The Banach Fixed-Point Theorem. In *Materiały konferencyjne Forum Innowacji Młodych Badaczy*, 2012a. ISBN 2082-4831. (Cyt. na stronach: 15, 55, 56, 71 i 159)
- 215) Nowotniak, R., Kucharski, J., GPU-based tuning of quantum-inspired genetic algorithm for a combinatorial optimization problem. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 60(2):323–330, March 2012b. ISSN 2300-1917. (Cyt. na stronach: 5, 55, 126, 160 i 161)
- 216) Nowotniak, R., Kucharski, J., Higher-Order Quantum-Inspired Genetic Algorithms. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, Annals of Computer Science and Information Systems*, pages 465–470. IEEE, 2014. ISBN 978-83-60810-58-3. (Cyt. na stronach: 5, 15, 55, 56, 85, 111 i 159)
- 217) Nvidia, C., NVIDIA CUDA programming guide, 2011. (Cyt. na stronie: 125)
- 218) Obuchowicz, A., The evolutionary search with soft selection and deterioration of the objective function. In *Proc. 6th Int. Conf. Intel. Inform. Sys. IIS'97*, pages 288–295, 1997. (Cyt. na stronie: 22)
- 219) Ong, Y.-S., Lim, M.-H., Zhu, N., Wong, K.-W., Classification of adaptive memetic algorithms: a comparative study. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(1):141–152, 2006. (Cyt. na stronie: 22)

- 220) Pareek, U., Naeem, M., Lee, D., Quantum Inspired Evolutionary algorithm for joint user selection and power allocation for uplink cognitive MIMO systems. In *Computational Intelligence in Scheduling (SCIS), 2011 IEEE Symposium on*, pages 33 –38, april 2011. doi: 10.1109/SCIS.2011.5976551. (Cyt. na stronie: 57)
- 221) Park, I.-W., Lee, B.-J., Kim, Y.-H., Han, J.-H., Kim, J.-H., Multi-objective quantum-inspired evolutionary algorithm-based optimal control of two-link inverted pendulum. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1 –7, july 2010. doi: 10.1109/CEC.2010.5586053. (Cyt. na stronie: 58)
- 222) Paszynska, A., An extension of vose's markov chain model for genetic algorithms. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, page 1554. ACM, 2005. (Cyt. na stronie: 25)
- 223) Patvardhan, C., Narayan, A., Srivastav, A., Enhanced Quantum Evolutionary Algorithms for Difficult Knapsack Problems. *Lecture Notes in Computer Science*, 4815:252, 2007. (Cyt. na stronie: 58)
- 224) Pedersen, M., *Tuning & Simplifying Heuristical Optimization*. PhD thesis, University of Southampton, School of Engineering Sciences, 2010. (Cyt. na stronach: 28, 30 i 147)
- 225) Pedersen, M. E. H., Chipperfield, A. J., Local unimodal sampling. *HL0801 Hvass Laboratories*, 2008. (Cyt. na stronach: 4, 30, 147 i 148)
- 226) Penrose, R., The Emperor's New Mind: Concerning Computers. *Minds, and the Laws of Physics*, 1989. (Cyt. na stronie: 10)
- 227) Penrose, R., Shadows of the mind: an approach to the missing science of consciousness, 1994. (Cyt. na stronie: 10)
- 228) Penrose, R., *The road to reality: A complete guide to the laws of the universe*. Vintage Books, 2006. (Cyt. na stronach: 9 i 39)
- 229) Penrose, R., Hameroff, S., Consciousness in the Universe: Neuroscience, Quantum Space-Time Geometry and Orch OR Theory. *Journal of Cosmology*, 14, 2011. (Cyt. na stronie: 10)
- 230) Penrose, R., Cartwright, N., Hawking, S., Shimony, A., The Large, the Small, and the Human Mind. 1997. (Cyt. na stronie: 10)
- 231) Perone, C. S., Pyevolve: a Python open-source framework for genetic algorithms. *SIGEVolution*, 4(1):12–20, 2009. doi: 10.1145/1656395.1656397. (Cyt. na stronach: 130 i 133)
- 232) Perroni, P. F., Point spread function estimation of solar surface images with a cooperative particle swarm optimization on GPUS. 2013. (Cyt. na stronie: 161)

- 233) Petko, J. S., Werner, D. H., An autoploidy-based genetic algorithm for enhanced evolution of linear polyfractal arrays. *Antennas and Propagation, IEEE Transactions on*, 55 (3):583–593, 2007. (Cyt. na stronie: 22)
- 234) Petrowski, A., A clearing procedure as a niching method for genetic algorithms. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 798–803. IEEE, 1996. (Cyt. na stronie: 22)
- 235) Pla, J., Tan, K., Dehollain, J., Lim, W., Morton, J., Jamieson, D., Dzurak, A., Morello, A., A single-atom electron spin qubit in silicon. *Nature*, 489(7417):541–545, 2012. (Cyt. na stronie: 9)
- 236) Planck, M., On the law of distribution of energy in the normal spectrum. *Annalen der Physik*, 4(553):1, 1901. (Cyt. na stronie: 8)
- 237) Platel, M., Schliebs, S., Kasabov, N., Quantum-Inspired Evolutionary Algorithm: A Multimodel EDA. *Evolutionary Computation, IEEE Transactions on*, 13(6):1218–1232, Dec. 2009. ISSN 1089-778X. doi: 10.1109/TEVC.2008.2003010. (Cyt. na stronie: 53)
- 238) Poli, R., Schema theorems without expectations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, page 806, 1999. (Cyt. na stronie: 26)
- 239) Poli, R., Why the schema theorem is correct also in the presence of stochastic effects. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 487–492 vol.1, 2000. doi: 10.1109/CEC.2000.870336. (Cyt. na stronie: 26)
- 240) Poli, R., Exact schema theory for genetic programming and variable-length genetic algorithms with one-point crossover. *Genetic Programming and Evolvable Machines*, 2(2): 123–163, 2001. (Cyt. na stronie: 26)
- 241) Pollack, J. B., Blair, A. D., Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32(3):225–240, 1998. (Cyt. na stronie: 22)
- 242) Popa, R., Nicolau, V., Epure, S., A new Quantum Inspired Genetic Algorithm for Evolvable Hardware. In *Electrical and Electronics Engineering (ISEEE), 2010 3rd International Symposium on*, pages 64–69, sept. 2010. doi: 10.1109/ISEEE.2010.5628539. (Cyt. na stronie: 58)
- 243) Qin, C., Zheng, J., Lai, J., A Multiagent Quantum Evolutionary Algorithm for Global Numerical Optimization. *LECTURE NOTES IN COMPUTER SCIENCE*, 4689:380, 2007. (Cyt. na stronie: 140)
- 244) Qin, C., Liu, Y., Zheng, J., A real-coded quantum-inspired evolutionary algorithm for global numerical optimization. In *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*, pages 1160–1164, Sept. 2008. doi: 10.1109/ICCIS.2008.4670779. (Cyt. na stronie: 56)

- 245) Radha, B., Rughooputh, H., Optimal network reconfiguration of electrical distribution systems using real coded quantum inspired evolutionary algorithm. In *Networking, Sensing and Control (ICNSC), 2010 International Conference on*, pages 38 –43, april 2010. doi: 10.1109/ICNSC.2010.5461550. (Cyt. na stronie: 57)
- 246) Ramdane, C., Meshoul, S., Batouche, M., Kholadi, M.-K., A quantum evolutionary algorithm for data clustering. *International Journal of Data Mining, Modelling and Management*, 2:369–387(19), 2010. doi: doi:10.1504/IJDM.2010.035564. (Cyt. na stronie: 57)
- 247) Rechenberg, I., Evolutionsstrategie–Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. 1973. (Cyt. na stronie: 21)
- 248) Reeves, C., Rowe, J., *Genetic algorithms: principles and perspectives: a guide to GA theory*. Kluwer Academic Pub, 2003. (Cyt. na stronie: 25)
- 249) Riebe, M., Häffner, H., Roos, C., Hänsel, W., Benhelm, J., Lancaster, G., Körber, T., Becher, C., Schmidt-Kaler, F., James, D., others, . Deterministic quantum teleportation with atoms. *Nature*, 429(6993):734–737, 2004. (Cyt. na stronie: 39)
- 250) Rudolph, G., Convergence analysis of canonical genetic algorithms. *IEEE transactions on Neural Networks*, 5(1):96–101, 1994. (Cyt. na stronie: 25)
- 251) Russell, S., Norvig, P., Davis, E., Russell, S., Russell, S., *Artificial intelligence: a modern approach*. Prentice hall Upper Saddle River, NJ, 2010. (Cyt. na stronach: 9 i 19)
- 252) Rutkowski, L., *Metody i techniki sztucznej inteligencji: inteligencja obliczeniowa*. Wydawnictwo Naukowe PWN, 2006. (Cyt. na stronach: 19, 22 i 26)
- 253) Ryu, S.-J., Lee, K.-B., Kim, J.-H., Improved version of a multiobjective quantum-inspired evolutionary algorithm with preference-based selection. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1 –7, june 2012. doi: 10.1109/CEC.2012.6256555. (Cyt. na stronie: 58)
- 254) Salam, A., Gauge unification of fundamental forces. *Reviews of Modern Physics*, 52: 525–538, 1980. (Cyt. na stronie: 8)
- 255) Samuel, A. L., Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 11(6):601–617, 1967. (Cyt. na stronie: 19)
- 256) Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., Sehnke, F., Rückstieß, T., Schmidhuber, J., PyBrain. *The Journal of Machine Learning Research*, 11:743–746, 2010. (Cyt. na stronach: 142 i 145)
- 257) Schmitt, L. M., Theory of genetic algorithms. *Theoretical Computer Science*, 259(1-2):1 – 61, 2001. ISSN 0304-3975. doi: DOI:10.1016/S0304-3975(00)00406-0. (Cyt. na stronie: 27)

- 258) Schmitt, L. M., Theory of Genetic Algorithms II: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. *Theoretical Computer Science*, 310(1-3):181 – 231, 2004. ISSN 0304-3975. doi: DOI:10.1016/S0304-3975(03)00393-1. (Cyt. na stronie: 25)
- 259) Seljebotn, D. S., Fast numerical computations with Cython. In *Proceedings of the 8th Python in Science Conference*, volume 37, 2009. (Cyt. na stronach: 145 i 167)
- 260) Shannon, S., *Trends in quantum computing research*. Nova Science Pub Inc, 2006. (Cyt. na stronie: 12)
- 261) Shen, S., Liu, Y., Probability evolutionary algorithm for functional and combinatorial optimization. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 7893–7897, 2008. (Cyt. na stronach: 56 i 58)
- 262) Shilane, D., Martikainen, J., Dudoit, S., Ovaska, S., A general framework for statistical performance comparison of evolutionary computation algorithms. *Information Sciences*, 178(14):2870–2879, 2008. (Cyt. na stronie: 144)
- 263) Shill, P., Hossain, M., Amin, M., Murase, K., An adaptive fuzzy logic controller based on real coded quantum-inspired evolutionary algorithm. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 614 –621, june 2011. doi: 10.1109/FUZZY.2011.6007374. (Cyt. na stronie: 57)
- 264) Shor, P. W., Algorithms for Quantum Computation: Discrete Log and Factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. Institute of Electrical and Electronic Engineers Computer Society Press, 1994. (Cyt. na stronach: 9 i 141)
- 265) Shu, W., Quantum-Inspired Genetic Algorithm Based on Simulated Annealing for Combinatorial Optimization Problem. *International Journal of Distributed Sensor Networks*, 5(1):64–65, 2009. (Cyt. na stronach: 58 i 59)
- 266) Silveira, L., Tanscheit, R., Vellasco, M., Quantum-inspired genetic algorithms applied to ordering combinatorial optimization problems. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1 –7, june 2012. doi: 10.1109/CEC.2012.6256511. (Cyt. na stronach: 56 i 58)
- 267) Skolicki, Z., De Jong, K., Improving evolutionary algorithms with multi-representation island models. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 420–429. Springer, 2004. (Cyt. na stronie: 22)
- 268) Socała, J., Kosiński, W., Kotowski, S., O asymptotycznym zachowaniu prostego algorytmu genetycznego. *Matematyka Stosowana: matematyka dla społeczeństwa Selected full texts*, 6:70–86, 2005. (Cyt. na stronie: 25)

- 269) Socala, J., Kosiński, W., Zastosowanie metody funkcji dolnej do badania zbieżności algorytmów genetycznych. 2007. (Cyt. na stronie: 25)
- 270) Sofge, D., Toward a framework for quantum evolutionary computation. In *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pages 1–6. IEEE, 2006. (Cyt. na stronie: 12)
- 271) Sofge, D., Prospective algorithms for quantum evolutionary computation. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, 2008. (Cyt. na stronach: 12 i 54)
- 272) Song, Q., Liu, X., Hybrid Quantum Evolutionary Algorithm and Its Application in Multiuser Detection of Electronic Communication System. In Jin, D., Lin, S., editors, *Advances in Mechanical and Electronic Engineering*, volume 178 of *Lecture Notes in Electrical Engineering*, pages 287–293. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-31528-2. (Cyt. na stronie: 57)
- 273) Stephens, C., Waelbroeck, H., Schemata evolution and building blocks. *Evolutionary computation*, 7(2):109–124, 1999. (Cyt. na stronie: 26)
- 274) Streichert, F., Stein, G., Ulmer, H., Zell, A., A clustering based niching method for evolutionary algorithms. In *Genetic and Evolutionary Computation—GECCO 2003*, pages 644–645. Springer, 2003. (Cyt. na stronie: 22)
- 275) Su, H., Yang, Y., Quantum-Inspired Differential Evolution for Binary Optimization. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, volume 1, 2008. (Cyt. na stronie: 59)
- 276) Tadeusiewicz, R., *Sieci neuronowe*. Akademicka Oficyna Wydawnicza RM, 1993. (Cyt. na stronie: 19)
- 277) Talbi, H., Batouche, M., Draa, A., A quantum-inspired genetic algorithm for multi-source affine image registration. *Lecture Notes in Computer Science*, pages 147–154, 2004a. (Cyt. na stronie: 57)
- 278) Talbi, H., Draa, A., Batouche, M., Abdelkader, U., Constantine, A., A new quantum-inspired genetic algorithm for solving the travelling salesman problem. In *2004 IEEE International Conference on Industrial Technology, 2004. IEEE ICIT'04*, volume 3, 2004b. (Cyt. na stronie: 58)
- 279) Talbi, H., Batouche, M., Draa, A., A Quantum-Inspired Evolutionary Algorithm for Multiobjective Image Segmentation. *International Journal of Mathematical, Physical and Engineering Sciences*, 1(2):109–114, 2007. (Cyt. na stronach: 57 i 58)
- 280) Tao, F., Zhang, L., Zhang, Z., Nee, A., A quantum multi-agent evolutionary algorithm for selection of partners in a virtual enterprise. *CIRP Annals - Manufacturing Technology*,

- In Press, Corrected Proof:–, 2010. ISSN 0007-8506. doi: DOI:10.1016/j.cirp.2010.03.120. (Cyt. na stronie: 56)
- 281) Tayarani, M., Akbarzadeh, T. M., A cellular structure and diversity preserving operator in Quantum Evolutionary Algorithms. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2665–2670, june 2008. doi: 10.1109/CEC.2008.4631156. (Cyt. na stronie: 56)
- 282) Teng, H., Zhao, B., Yang, B., An Improved Mutative Scale Chaos Optimization Quantum Genetic Algorithm. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, volume 6, 2008. (Cyt. na stronie: 56)
- 283) Teng, H., Zhao, B., Cao, A., Chaos Quantum Genetic Algorithm Based on Henon Map. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, volume 1, pages 922–925, 11-12 2010. doi: 10.1109/ICICTA.2010.810. (Cyt. na stronie: 56)
- 284) Toosi, T., Mashhadi, H., A Modified Genetic Algorithm Comparable to Quantum GA. 2005. (Cyt. na stronie: 58)
- 285) Tu, Z., Lu, Y., A robust stochastic genetic algorithm (StGA) for global numerical optimization. *Evolutionary Computation, IEEE Transactions on*, 8(5):456–470, 2004. (Cyt. na stronie: 142)
- 286) Udrescu, M., Prodan, L., Vlăduțiu, M., Implementing quantum genetic algorithms: a solution based on Grover's algorithm. In *CF '06: Proceedings of the 3rd conference on Computing frontiers*, pages 71–82, New York, NY, USA, 2006. ACM. ISBN 1-59593-302-6. doi: <http://doi.acm.org/10.1145/1128022.1128034>. (Cyt. na stronie: 12)
- 287) Vaidman, L., Teleportation of quantum states. *Physical Review A*, 49(2):1473, 1994. (Cyt. na stronie: 39)
- 288) Vlachogiannis, J., Lee, K., Quantum-Inspired Evolutionary Algorithm for Real and Reactive Power Dispatch. *IEEE Transactions on Power Systems*, 23(4):1627–1636, 2008. (Cyt. na stronie: 57)
- 289) Vlachogiannis, J., Østergaard, J., Reactive power and voltage control based on general quantum genetic algorithms. *Expert Systems with Applications*, 36(3P2):6118–6126, 2009. (Cyt. na stronie: 57)
- 290) Vose, M., *The simple genetic algorithm: foundations and theory*. The MIT Press, 1999. (Cyt. na stronach: 26 i 87)
- 291) Vose, M. D., Wright, A. H., The simple genetic algorithm and the walsh transform: Part I, theory. *Evol. Comput.*, 6(3):253–273, 1998a. ISSN 1063-6560. doi: <http://dx.doi.org/10.1162/evco.1998.6.3.253>. (Cyt. na stronach: 25, 26 i 87)

- 292) Vose, M. D., Wright, A. H., The simple genetic algorithm and the walsh transform: Part II, the inverse. *Evol. Comput.*, 6(3):275–289, 1998b. ISSN 1063-6560. doi: <http://dx.doi.org/10.1162/evco.1998.6.3.275>. (Cyt. na stronach: 26 i 87)
- 293) Wang, H., Huang, Y., Ding, H., Application of support vector machine and quantum genetic algorithm in infrared target recognition. volume 7820, page 78201O. SPIE, 2010. doi: 10.1117/12.866715. (Cyt. na stronie: 58)
- 294) Wang, J., Zhou, R., A Novel Quantum Genetic Algorithm for PID Controller. In Huang, D.-S., Zhao, Z., Bevilacqua, V., Figueroa, J., editors, *Advanced Intelligent Computing Theories and Applications*, volume 6215 of *Lecture Notes in Computer Science*, pages 72–77. Springer Berlin / Heidelberg, 2010. (Cyt. na stronach: 56 i 58)
- 295) Wang, L., Li, L., An effective hybrid quantum-inspired evolutionary algorithm for parameter estimation of chaotic systems. *Expert Systems With Applications*, 2009. (Cyt. na stronie: 58)
- 296) Wang, L., Tang, F., Wu, H., Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation. *Applied Mathematics and Computation*, 171(2):1141–1156, 2005a. (Cyt. na stronach: 56 i 57)
- 297) Wang, L., Wu, H., Tang, F., Zheng, D., A hybrid quantum-inspired genetic algorithm for flow shop scheduling. *Lecture Notes in Computer Science*, 3645:636, 2005b. (Cyt. na stronie: 57)
- 298) Wang, L., Wu, H., Zheng, D., A quantum-inspired genetic algorithm for scheduling problems. *Lecture Notes in Computer Science*, 3612:417, 2005c. (Cyt. na stronie: 57)
- 299) Wang, L., Wang, X., Fei, M., A novel quantum-inspired pseudorandom proportional evolutionary algorithm for the multidimensional knapsack problem. *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pages 545–552, 2009. (Cyt. na stronie: 56)
- 300) Wang, R., Guo, N., Xiang, F., Mao, J., An improved quantum genetic algorithm with mutation and its application to 0-1 knapsack problem. In *Measurement, Information and Control (MIC), 2012 International Conference on*, volume 1, pages 484 –488, may 2012a. doi: 10.1109/MIC.2012.6273347. (Cyt. na stronie: 58)
- 301) Wang, Y., Shi, Y., The application of quantum-inspired evolutionary algorithm in analog evolvable hardware. In *Environmental Science and Information Application Technology (ESIAT), 2010 International Conference on*, volume 2, pages 330 –334, july 2010. doi: 10.1109/ESIAT.2010.5567359. (Cyt. na stronie: 58)
- 302) Wang, Y., Feng, X., Huang, Y., Pu, D., Zhou, W., Liang, Y., Zhou, C., A novel quantum swarm evolutionary algorithm and its applications. *Neurocomputing*, 70(4-6):633–640, 2007. (Cyt. na stronie: 58)

- 303) Wang, Y., Zhou, J., Mo, L., Ouyang, S., Zhang, Y., A clonal real-coded quantum-inspired evolutionary algorithm with Cauchy mutation for short-term hydrothermal generation scheduling. *International Journal of Electrical Power and Energy Systems*, 43(1):1228 – 1240, 2012b. ISSN 0142-0615. doi: 10.1016/j.ijepes.2012.06.036. (Cyt. na stronie: 58)
- 304) Wang, Y., Zhou, J., Mo, L., Zhang, R., Zhang, Y., Short-term hydrothermal generation scheduling using differential real-coded quantum-inspired evolutionary algorithm. *Energy*, 44(1):657 – 671, 2012c. ISSN 0360-5442. doi: 10.1016/j.energy.2012.05.026. <ce:title>Integration and Energy System Engineering, European Symposium on Computer-Aided Process Engineering 2011</ce:title>. (Cyt. na stronie: 140)
- 305) Wang-xin, X., Xue, Z., Xin-ping, Y., QGA based Bandwidth Adaptation Scheme for Wireless/Mobile Networks. pages 1323 –1326, june 2006. doi: 10.1109/ITST.2006.288891. (Cyt. na stronie: 57)
- 306) Wei, W., Li, B., Zou, Y., Zhang, W., Zhuang, Z., A multi-objective HW-SW co-synthesis algorithm based on quantum-inspired evolutionary algorithm. *International Journal of Computational Intelligence and Applications*, 7(2):129–148, 2008. (Cyt. na stronie: 58)
- 307) Weinberg, R., Computer simulation of a living cell, University of Michigan. *Ann Arbor, MI*, 1970. (Cyt. na stronie: 28)
- 308) Weise, T., Global Optimization Algorithms–Theory and Application. URL: <http://www.it-weise.de>, Abrufdatum, 1, 2008. (Cyt. na stronie: 25)
- 309) Whitley, D., A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 06/01 1994. M3: 10.1007/BF00175354. (Cyt. na stronie: 22)
- 310) Whitley, D., Rana, S., Heckendorn, R. B., The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–48, 1999. (Cyt. na stronie: 22)
- 311) Wolpert, D., Macready, W., No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, Apr 1997. ISSN 1089-778X. doi: 10.1109/4235.585893. (Cyt. na stronie: 25)
- 312) Wu, X., Li, S., A quantum inspired algorithm for the job shop scheduling problem. In *Computing, Control and Industrial Engineering (CCIE), 2011 IEEE 2nd International Conference on*, volume 2, pages 212 –215, aug. 2011. doi: 10.1109/CCIENG.2011.6008104. (Cyt. na stronie: 59)
- 313) Xiao, J., Liu, B., Quantum swarm evolutionary algorithm with time-varying acceleration coefficients for partner selection in virtual enterprise, 2009. (Cyt. na stronie: 59)

- 314) Xiao, J., Yan, Y., Lin, Y., Yuan, L., Zhang, J., A Quantum-inspired Genetic Algorithm for Data Clustering. In *Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong*, pages 1513–1518. Citeseer, 2008. (Cyt. na stronie: 56)
- 315) Xiao, J., Yan, Y., Zhang, J., Tang, Y., A quantum-inspired genetic algorithm for k-means clustering. *Expert Systems with Applications*, 2009. (Cyt. na stronach: 56, 57 i 58)
- 316) Xing, H., Ji, Y., Bai, L., Liu, X., Qu, Z., Wang, X., An adaptive-evolution-based quantum-inspired evolutionary algorithm for QoS multicasting in IP/DWDM networks. *Computer Communications*, 32(6):1086–1094, 2009a. (Cyt. na stronie: 59)
- 317) Xing, H., Ji, Y., Bai, L., Sun, Y., An improved quantum-inspired evolutionary algorithm for coding resource optimization based network coding multicast scheme. *AEU - International Journal of Electronics and Communications*, In Press, Corrected Proof:–, 2009b. ISSN 1434-8411. doi: DOI:10.1016/j.aeue.2009.11.012. (Cyt. na stronie: 59)
- 318) Xing, H., Liu, X., Jin, X., Bai, L., Ji, Y., A multi-granularity evolution based Quantum Genetic Algorithm for QoS multicast routing problem in WDM networks. *Computer Communications*, 32(2):386–393, 2009c. (Cyt. na stronach: 56 i 59)
- 319) Xing, H., Ji, Y., Bai, L., Sun, Y., An improved quantum-inspired evolutionary algorithm for coding resource optimization based network coding multicast scheme. *AEU - International Journal of Electronics and Communications*, 64(12):1105 – 1113, 2010. ISSN 1434-8411. doi: 10.1016/j.aeue.2009.11.012. (Cyt. na stronie: 57)
- 320) Xiong, H., Xiong, K., Tang, Q., A Novel Variable-Boundary-Coded Quantum Genetic Algorithm for Function Optimization. In *DASC '09: Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 279–285, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3929-4. doi: <http://dx.doi.org/10.1109/DASC.2009.10>. (Cyt. na stronie: 56)
- 321) Xu, N., Zhu, J., Lu, D., Zhou, X., Peng, X., Du, J., Quantum factorization of 143 on a dipolar-coupling nmr system. *arXiv preprint arXiv:1111.3726*, 2011. (Cyt. na stronie: 9)
- 322) Xue, H., Li, X., Ma, H.-X., Random fuzzy chance-constrained programming based on adaptive chaos quantum honey bee algorithm and robustness analysis. *International Journal of Automation and Computing*, 7(1):115–122, 02/01 2010. M3: 10.1007/s11633-010-0115-6. (Cyt. na stronie: 59)
- 323) Yager, R. R., Zadeh, L. A., *An introduction to fuzzy logic applications in intelligent systems*. Kluwer Academic Publishers, 1992. (Cyt. na stronie: 19)
- 324) Yan, L., Chen, H., Ji, W., Lu, Y., Li, J., Optimal VSM Model and Multi-Object Quantum-Inspired Genetic Algorithm for Web Information Retrieval. pages 1 –4, jan. 2009. doi: 10.1109/CNMT.2009.5374788. (Cyt. na stronie: 58)

- 325) Yang, H., Li, M., Form invariance of schema and exact schema theorem. *Science in China Series F: Information Sciences*, 46(6):475–484, 2003. (Cyt. na stronie: 26)
- 326) Yang, Q., Ding, S., Methodology and Case Study of Hybrid Quantum-Inspired Evolutionary Algorithm for Numerical Optimization. In *Natural Computation, 2007. ICNC 2007. Third International Conference on*, volume 5, 2007. (Cyt. na stronie: 140)
- 327) Yang, S., Jiao, L., The quantum evolutionary programming. In *Computational Intelligence and Multimedia Applications, 2003. ICCIMA 2003. Proceedings. Fifth International Conference on*, pages 362–367, Sept. 2003. (Cyt. na stronie: 59)
- 328) Yang, X., *Nature-inspired metaheuristic algorithms*. Luniver Pr, 2008. book describing a variety of modern metaheuristic algorithms (GA, ACO, Bee, SA, HS, Firefly, TS). (Cyt. na stronie: 4 i 21)
- 329) Yao, X., Liu, Y., Lin, G., Quantum gate optimization in a meta-level genetic quantum algorithm. In *Proceedings on Evolutionary Programming VI*, pages 151–161, 1997. (Cyt. na stronie: 142)
- 330) Yao, X., Liu, Y., Lin, G., Evolutionary programming made faster. *Evolutionary Computation, IEEE Transactions on*, 3(2):82–102, 1999. (Cyt. na stronie: 142)
- 331) Yasin, Z., Rahman, T., Zakaria, Z., Multiobjective quantum-inspired evolutionary programming for optimal load shedding. In *Control and System Graduate Research Colloquium (ICSGRC), 2012 IEEE*, pages 160–165, july 2012. doi: 10.1109/ICSGRC.2012.6287154. (Cyt. na stronie: 57)
- 332) Yi, S., Chen, M., Zeng, Z., Convergence analysis on a class of quantum-inspired evolutionary algorithms. In *Natural Computation (ICNC), 2011 Seventh International Conference on*, volume 2, pages 1072–1076, july 2011. doi: 10.1109/ICNC.2011.6022161. (Cyt. na stronie: 76)
- 333) Yin, G., Li, J., Dong, H., Wei, J., Quantum Genetic Algorithm Using a Mixed Update Strategy. In Zhu, R., Ma, Y., editors, *Information Engineering and Applications*, volume 154 of *Lecture Notes in Electrical Engineering*, pages 412–419. Springer London, 2012. ISBN 978-1-4471-2386-6. (Cyt. na stronie: 56)
- 334) Yin, X., Khoo, L., An exact schema theorem for adaptive genetic algorithm and its application to machine cell formation. *Expert Systems with Applications*, 38(7):8538–8552, 2011. (Cyt. na stronie: 27)
- 335) Yu, H.-Y., Fan, J.-L., Three-Level Image Segmentation Based on Maximum Fuzzy Partition Entropy of 2-D Histogram and Quantum Genetic Algorithm. In *ICIC '08: Proceedings of the 4th international conference on Intelligent Computing*, pages 484–493, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-85983-3. doi: http://dx.doi.org/10.1007/978-3-540-85984-0_58. (Cyt. na stronie: 57)

- 336) Yu, Y., Cunhua, L., Shangce, G., Zheng, T., Quantum interference crossover-based clonal selection algorithm and its application to traveling salesman problem. *IEICE TRANSACTIONS on Information and Systems*, 92(1):78–85, 2009. (Cyt. na stronie: 58)
- 337) Zadeh, L. A., Fuzzy sets. *Information and control*, 8(3):338–353, 1965. (Cyt. na stronie: 19)
- 338) Zak, M., *From Quantum Computing to Intelligence*. Nova Science Publishers, Inc., 2011. (Cyt. na stronie: 11)
- 339) Zhang, G., Quantum-inspired evolutionary algorithms: a survey and empirical study. *Journal of Heuristics*, pages 1–49, 2010a. ISSN 1381-1231. 10.1007/s10732-010-9136-0. (Cyt. na stronach: 10, 12, 13, 53, 54 i 60)
- 340) Zhang, G., Time-Frequency Atom Decomposition with Quantum-Inspired Evolutionary Algorithms. *Circuits, Systems, and Signal Processing*, pages 1–25, 2010b. (Cyt. na stronie: 58)
- 341) Zhang, G., Rong, H., Real-Observation Quantum-Inspired Evolutionary Algorithm for a Class of Numerical Optimization Problems. In Shi, Y., van Albada, G., Dongarra, J., Sloot, P., editors, *Computational Science ICCS 2007*, volume 4490 of *Lecture Notes in Computer Science*, pages 989–996. Springer Berlin / Heidelberg, 2007. (Cyt. na stronach: 53, 54 i 140)
- 342) Zhang, G., Jin, W., Li, N., An improved quantum genetic algorithm and its application. *Lecture notes in computer science*, pages 449–452, 2003. (Cyt. na stronie: 56)
- 343) Zhang, G., Li, N., Jin, W., Hu, L., Novel quantum genetic algorithm and its applications. *Frontiers of Electrical and Electronic Engineering in China*, 1(1):31–36, 2006. (Cyt. na stronach: 56 i 57)
- 344) Zhang, G., Gheorghe, M., Wu, C., A Quantum-Inspired Evolutionary Algorithm Based on P systems for Knapsack Problem. *Fundamenta informaticae*, 87(1):93–116, 2008a. (Cyt. na stronie: 58)
- 345) Zhang, H., Zhang, G., Rong, H., Cheng, J., Comparisons of quantum rotation gates in quantum-inspired evolutionary algorithms. In *Natural Computation (ICNC), 2010 Sixth International Conference on*, volume 5, pages 2306–2310, aug. 2010. doi: 10.1109/ICNC.2010.5584179. (Cyt. na stronach: 53 i 56)
- 346) Zhang, J.-L., Zhao, Y.-W., Peng, D.-J., Wang, W.-L., A Hybrid Quantum-Inspired Evolutionary Algorithm for Capacitated Vehicle Routing Problem. In *ICIC '08: Proceedings of the 4th international conference on Intelligent Computing*, pages 31–38, Berlin, Heidelberg, 2008b. Springer-Verlag. ISBN 978-3-540-87440-9. doi: http://dx.doi.org/10.1007/978-3-540-87442-3_5. (Cyt. na stronie: 56)

- 347) Zhang, W., Shi, Z., Luo, Z., Prediction of urban passenger transport based-on wavelet SVM with quantum-inspired evolutionary algorithm. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1509–1514, June 2008c. doi: 10.1109/IJCNN.2008.4633996. (Cyt. na stronie: 58)
- 348) Zhang, X. J., Li, S. Y., Shen, Y., Song, S. M., Evaluation of several quantum genetic algorithms in medical image registration applications. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, volume 2, pages 710–713, may 2012. doi: 10.1109/CSAE.2012.6272866. (Cyt. na stronie: 57)
- 349) Zhao, S., Xu, G., Tao, T., Liang, L., Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks. *Computers and Mathematics with Applications*, 57(11-12):2009–2015, 2009a. (Cyt. na stronie: 56)
- 350) Zhao, Z., Peng, Z., Zheng, S., Shang, J., Cognitive radio spectrum allocation using evolutionary algorithms. *Trans. Wireless. Comm.*, 8(9):4421–4425, 2009b. ISSN 1536-1276. doi: <http://dx.doi.org/10.1109/TWC.2009.080939>. (Cyt. na stronie: 58)
- 351) Zheng, T., Yamashiro, M., A novel quantum differential evolutionary algorithm for non-permutation flow shop scheduling problems. In *Electrical Engineering Computing Science and Automatic Control (CCE), 2010 7th International Conference on*, pages 357–362, sept. 2010a. doi: 10.1109/ICEEE.2010.5608564. (Cyt. na stronie: 59)
- 352) Zheng, T., Yamashiro, M., Solving flow shop scheduling problems by quantum differential evolutionary algorithm. *The International Journal of Advanced Manufacturing Technology*, pages 1–20, 2010b. (Cyt. na stronie: 59)
- 353) Zheng, Y., Liu, J., A novel quantum-inspired genetic algorithm for a weekly university scheduling optimization. In *Information Science and Technology (ICIST), 2011 International Conference on*, pages 373–376, march 2011. doi: 10.1109/ICIST.2011.5765270. (Cyt. na stronie: 58)
- 354) Zhou, R., Cao, J., Quantum novel genetic algorithm based on parallel subpopulation computing and its application. *Artificial Intelligence Review*, 41(3):359–371, 2014. (Cyt. na stronie: 55)
- 355) Zhou, W. G., Zhou, C. G., Liu, G. X., Lv, H. Y., Liang, Y. C., An Improved Quantum-Inspired Evolutionary Algorithm for Clustering Gene Expression Data, 2006. (Cyt. na stronie: 56 i 57)
- 356) Zongyao, H., Zhou, L., A New Real-Coded Quantum-Inspired Evolutionary Algorithm. *IJACT : International Journal of Advancements in Computing Technology*, 3(7):108–106, 2011. doi: DOI:10.4156/ijact.vol3.issue7.14. (Cyt. na stronie: 56 i 140)

SPIS TABEL

1	Tabela oznaczeń algorytmów	4
2	Tabela symboli	5
3.1	Najważniejsze publikacje w kolejności chronologicznej . . .	53
3.2	Tabela <i>Lookup Table</i> dla algorytmu QIGA1 (kąty obrotów) . .	63
3.3	Tabela <i>Lookup Table</i> dla algorytmu QIGA1 (kierunki obrotów)	63
4.1	Dopasowanie chromosomów kwantowych do schematu . .	79
5.1	Przestrzenie działania algorytmów i ich własności	95
6.1	Tablica <i>Lookup table</i> kątów obrotów QIGA1	129
6.2	Wartości kątów obrotów $\Delta\theta$ w algorytmie QIGA1	130
6.3	Najlepsze wartości kątów $\Delta\theta$ w QIGA1 – statystyka	131
6.4	Wartości kątów $\Delta\theta$ znalezione przez metaoptymalizację . . .	132
6.5	Porównanie SGA, QIGA1, QIGA1* i QIGA2 dla 20 zadań . .	138
6.6	Ranking porównawczy algorytmów	138
7.1	Czas obliczeń dla algorytmu iQIEA na klastrze	151
7.2	Rankingi algorytmów NelderMead,GA,PSO,CMAES,iQIEA	153
7.3	Najlepsze i najgorsze wartości parametrów (ξ, δ) iQIEA . . .	155

SPIS RYSUNKÓW

1	Informatyka Kwantowa i Sztuczna Inteligencja	11
1.1	Klasyfikacja metod Sztucznej Inteligencji	19
1.2	Ogólny schemat algorytmu ewolucyjnego	24
1.3	Ogólny schemat metaoptymalizacji	29
2.1	Usytuowanie klasy algorytmów QIEA	32
2.2	Geometryczna reprezentacja stanu kubitu	34
2.3	Ilustracja stanu 4-kubitowego rejestru	38
2.4	Działanie bramki kwantowej NOT	42
2.5	Działanie bramki kwantowej obrotu $ROT(\Delta\theta)$	43
2.6	Działanie bramki kwantowej Hadamarda	44
3.1	Ilustracja stanu binarnego genu kwantowego	50
3.2	Ilustracja stanu binarnej populacji kwantowej	51
3.3	Liczba publikacji nt. algorytmów QIEA w kolejnych latach	59
3.4	Ilustracja genu kwantowego w algorytmie iQIEA	65
3.5	Ogólny schemat algorytmu iQIEA	66
4.1	Dopasowanie chromosomu kwantowego do schematu	78
4.2	Porównanie propagacji schematu w algorytmach	83
5.1	Przestrzeń ciągów binarnych	86
5.2	Przestrzeń schematów	88
5.3	Przestrzeń zbioru niezależnych genów kwantowych	89
5.4	Przestrzeń rejestru kwantowego	94
5.5	Geny połączone do wspólnych rejestrów kwantowych	97
5.6	Wykres współczynnika λ dla różnych rzędów	101
5.7	Wykres współczynnika λ_O dla różnych rzędów	102
5.8	Rozkłady kodowane w genotypach rzędu $r = 2$	105
5.9	Reprezentacja rozwiązań dla różnych rzędów kwantowości	109
5.10	Przestrzeń algorytmu dla różnych rzędów kwantowości	110

5.11	Reprezentacja rozwiązań w algorytmie QIGA1	113
5.12	Reprezentacja rozwiązań w algorytmie QIGA2	113
5.13	Działanie nowego operatora w algorytmie QIGA2	116
6.1	Przestrzeń rozwiązań zadania plecakowego	123
6.2	Przestrzeń rozwiązań zadania spełnialności SAT	124
6.3	Autorski sposób zrównoleglenia obliczeń na urządzeniu GPU	126
6.4	Porównanie szybkości implementacji CPU vs GPU	127
6.5	Zastosowane podejście do metaoptymalizacji	128
6.6	Porównanie efektywności SGA,QIGA1 i QIGA1*	134
6.7	Porównanie algorytmów za pomocą histogramów	135
6.8	Porównanie efektywności dla zadania knapsack250	137
7.1	Wykresy wybranych funkcji testowych z CEC2013	146
7.2	Porównanie algorytmów na wykresie słupkowym	154
7.3	Wyniki metaoptymalizacji iQIEA w przestrzeni (ξ, δ)	156

SKOROWIDZ

- allel, 24
- amplitudy prawdopodobieństwa, 35
- baza, 35
 - obliczeniowa, 35
 - standardowa, 35
- bramka kwantowa, 42
 - Hadamarda HAD, 45
 - negacji NOT, 43
 - obrotu ROT($\Delta\theta$), 44
- chromosom, 24
 - binarny kwantowy, 49
- epistaza, 25
- fenotyp, 24
- funkcja
 - błędu, 25
 - celu, 22
 - dopasowania, 25
 - kosztu, 25
 - metaprzystosowania, 30
- gen, 24
 - kwantowy, 49
 - kwantowy binarny, 49
- genotyp, 24
 - kwantowy, 50
- iloczyn tensorowy, 38
- Informatyka Kwantowa, 33
- kontrakcja, 173
- kubit, 35
- locus, 24
- metaoptymalizacja, 29
- metryka, 173
- optymalizacja, 22
 - kombinatoryczna, 22
 - numeryczna, 22
- populacja, 24
 - kwantowa, 50
- przestrzeń
 - metryczna, 173
 - rozwiązań, 22
 - zupełna, 173
- rejestr kwantowy, 37
- rozpiętość schematu, 28
- rząd
 - kwantowości, 102
 - względny, 102
- schematu, 28
- stan splątany, 41
- Sztuczna Inteligencja, 21
- twierdzenie
 - Banacha, 75, 173
 - Hollanda o schematach, 28
- współczynnik kwantowości, 103
 - operatorowej, 105
- zwodniczość, 25