POLITECHNIKA WARSZAWSKA

WYDZIAŁ ELEKTRONIKI I TECHNIK INFORMACYJNYCH

Rozprawa doktorska

mgr inż. Daniel Andrzej Piętak

Metoda oceny jakości wyników eksperymentów wzbudzeń kulombowskich z wykorzystaniem algorytmu genetycznego

Promotor prof. dr hab. inż. Piotr Bilski

WARSZAWA 2020

Pracę dedykuję mojej Mamie, która jako pierwsza zaszczepiła we mnie głód wiedzy i chęć jej odkrywania. Od zawsze jest dla mnie niedoścignionym wzorem interdyscyplinarnego patrzenia na świat.

Mamo,

bez Twojego wsparcia i pomocy nie skończyłbym tej pracy!

PODZIĘKOWANIA

Chciałbym wyrazić wdzięczność wszystkim, którzy przyczynili się do powstania niniejszej pracy.

Przede wszystkim bardzo serdecznie dziękuję mojemu Promotorowi, prof. dr. hab. inż. Piotrowi Bilskiemu, za okazane serce, poświęcony czas, życzliwość, wsparcie i zaangażowanie. Bardzo dziękuję za pomoc merytoryczną i trafne sugestie dotyczące redakcji mojej rozprawy.

Serdecznie dziękuję również śp. prof. dr. hab. inż. Jackowi Wojciechowskiemu, który był Promotorem niniejszej pracy na początkowym etapie moich studiów doktoranckich.

Szczególne podziękowania kieruję do dr. Pawła Jana Napiorkowskiego, szefa warszawskiej grupy wzbudzeń kulombowskich i zastępcy dyrektora Środowiskowego Laboratorium Ciężkich Jonów Uniwersytetu Warszawskiego. Na trzecim roku moich studiów informatycznych zainteresował mnie tematyką wzbudzeń kulombowskich. To on zainspirował mnie i zachęcił do rozwijania metod numerycznych w analizie danych eksperymentalnych dotyczących badań z zakresu fizyki jądrowej; najpierw w ramach dyplomowej pracy magisterskiej, a następnie pracy zawodowej. Towarzyszył mi i wspierał przy pracy nad niniejszą rozprawą doktorską. Jego upór, cierpliwość, ogromna wiedza oraz doświadczenie były i są nieocenione w pracy naukowo-badawczej.

Dziękuję moim Koleżankom i Kolegom ze Środowiskowego Laboratorium Ciężkich Jonów Uniwersytetu Warszawskiego, a w szczególności z warszawskiej grupy wzbudzeń kulombowskich: dr Katarzynie Hadyńskiej-Klęk, mgr Michalinie Komorowskiej, dr inż. Magdalenie Matejskiej-Mindzie, dr Mansi Saxena, dr Katarzynie Wrzosek-Lipskiej, dr Magdalenie Zielińskiej, dr. Jędrzejowi Iwanickiemu, mgr. Wojciechowi Piątkowi oraz dr. Julianowi Srebrnemu.

Serdecznie dziękuję mojej Siostrze, mgr inż. Katarzynie Ramsza, za jej trafne uwagi dotyczące opisu budowy programów JACOB, RepGen oraz ScanRep, jak również za cenne spostrzeżenia i pomoc w redakcji całej dysertacji.

Gorąco dziękuję całej mojej Rodzinie i Przyjaciołom, którzy nieustannie mnie wspierali. Ich zainteresowanie moimi postępami, podtrzymywanie na duchu i niesłabnąca wiara we mnie dodawały mi sił do dalszej pracy każdego dnia.

Szczególnie dziękuję łódzkiej grupie Odnowy w Duchu Świętym oraz Zgromadzeniu Sióstr Karmelitanek w Dysie za ich wsparcie modlitewne.

STRESZCZENIE

W niniejszej rozprawie doktorskiej przedstawiono Metodę AH – nową heurystyczną procedurę szacowania niepewności wartości parametrów, wyznaczanych w eksperymentach wzbudzeń kulombowskich. Analiza kształtu optymalizowanej funkcji celu wokół znalezionego rozwiązania optymalnego odbywa się jedynie w oparciu o próbkowanie przestrzeni zebrane podczas procesu przeszukiwania algorytmem genetycznym. Do jej wykonania nie są potrzebne żadne dodatkowe wywołania funkcji celu.

Stanowiący część Metody AH, autorski algorytm FLA (ang. *Front-Line Algorithm*) wskazuje te punkty z repozytorium próbkowania przestrzeni algorytmem genetycznym, które leżą najbliżej miejsca przecięcia powierzchni funkcji celu z płaszczyzną progową, dla której ma być przeprowadzona estymacja niepewności parametrów.

Metoda AH została opracowana do oceny jakości wyników pomiarów wzbudzeń kulombowskich (COULEX) – eksperymentalnej techniki badania elektromagnetycznej struktury jąder atomowych. Może być potencjalnie zastosowana również do innych problemów programowania nieliniowego, w których konieczna jest aproksymacja kształtu funkcji celu wokół znalezionego optimum.

Słowa kluczowe: wzbudzenia kulombowskie, struktura elektromagnetyczna jądra atomowego, algorytmy heurystyczne, programowanie nieliniowe, estymacja niepewności pomiaru.

ABSTRACT

In the PhD thesis the AH Method is presented, i. e. new heuristic procedure to estimate uncertainty of parameter values determination. The function shape analysis in direct vicinity of the found optimal solution is based only on the space sampling collected during the optimization process with genetic algorithm – no additional calls of the objective function are needed.

Constituting a part of AH Method, the original Front-Line Algorithm (FLA) indicates the points from genetic algorithm sample repository, which lie closest to the cut of the objective function surface with the threshold plane for which the parameter uncertainty estimation is to be conducted.

AH Method has been developed for results quality validation of Coulomb excitation (COULEX) measurements – experimental technique to study electromagnetic properties of nuclei. It can be also potentially adapted to many other non-linear programming problems, where the objective function shape approximation in direct vicinity of the found optimum is necessary.

Keywords: Coulomb excitation, electromagnetic properties of nucleus, heuristic algorithms, non-linear programming, measurement uncertainty estimation.

ENGLISH TITLE

Method of results quality validation of Coulomb excitation experiments with usage of genetic algorithm

<u>1.WSTĘP</u>	<u>10</u>
1.1.Tezy pracy	11
1.2.UKŁAD PRACY	11
2.WZBUDZENIA KULOMBOWSKIE	12
2.1.Pomiar i zbieranie danych	
2.1.1.Energia wiązki	13
2.1.2.Eksperyment	14
2.1.3.Rejestracja danych eksperymentalnych	15
2.2.Analiza danych eksperymentalnych	17
2.2.1.Charakterystyka danych podlegających analizie	
2.2.2.Cel analizy	
2.2.3.DOTYCHCZASOWA METODA ANALIZY	21
2.3.WNIOSKI	27
3.WYZNACZANIE OPTYMALNEGO ZESTAWU ELEMENTÓW MACIERZOW	YCH PRZY
UŻYCIU ALGORYTMU GENETYCZNEGO	29
3.1.ANALIZA PROBLEMU	
3.1.1.Definicja problemu	
3.1.2. Model problemu	
3.1.3.Próba klasycznego rozwiązania	40
3.2.BUDOWA ALGORYTMU GENETYCZNEGO.	41
3.2.1.Struktura algorytmu	41
3.2.2.Populacja początkowa	43
3.2.3. Operatory selekcji	45
3.2.4. Operatory krzyżowania	
3.2.5.Operatory mutacji	
3.2.6.Kryteria stopu	54
3.3.Implementacja algorytmu genetycznego	57
3.4.Program JACOB	58
3.5. Schemat analizy danych eksperymentalnych z wykorzystaniem algorytmu gene	TYCZNEGO60
4.ANALIZA PRÓBKOWANIA PRZESTRZENI OPTYMALIZACYJNEJ PRZEZ	
ALGORYTM GENETYCZNY	62
4.1.Opis przypadku testowego	64
4.2. Porównywane metody próbkowania przestrzeni – Raw Dataset (A)	65
4.3.Ograniczenie jakościowe punktów w repozytorium – Threshold Cut (B)	71
4.4.Ograniczenie gęstości próbkowania – Minimum Distance Rule (C)	76
4.5. Detekcja basenów przyciągania optimów – Clusterization (D)	81
4.5.1.Łączenie przylegających klastrów – modyfikacja algorytmu NBC	
4.6. Metody statystycznego badania rozkładu próbkowania – Statistics (E)	90
4.7.BADANIE ROZKŁADU PRÓBKOWANIA W OPARCIU O ODLEGŁOŚCI MIĘDZY PUNKTAMI W REPOZ Histograms (F)	'ytorium – 96

Spis treści

5.METODA AH – ESTYMACJA NIEPEWNOŚCI PARAMETRÓW Z WYKORZY	STANIEM
PRÓBKOWANIA PRZESTRZENI ALGORYTMEM GENETYCZNYM	101
5.1.GEOMETRYCZNA INTERPRETACJA NIEPEWNOŚCI PARAMETRÓW	
5.1.1.Statystyka najmniejszych kwadratów	
5.1.2. Niepewności wyznaczenia wartości parametrów	104
5.1.3.Liczba stopni swobody	107
5.2. Wyodrębnienie punktów najbliższych płaszczyźnie progowej – Front-Line Algo	rithm (G)110
5.3.Estymacja niepewności parametrów – Parameter Uncertainties (H)	
5.3.1.Dalsza estymacja niepewności	121
5.3.2. Wyniki estymacji niepewności dla przypadku testowego	
5.4.Program ScanRep	
5.5.Schemat Metody AH	126
6.ZASTOSOWANIE METODY AH DO ANALIZY DANYCH Z POMIARÓW WZI	BUDZEŃ
KULOMBOWSKICH	<u>128</u>
6.1.MODUŁ PRZETWARZANIA WSTĘPNEGO	
6.2. Moduł analizy próbkowania	
6.3.Moduł wyznaczenia niepewności	145
7.PODSUMOWANIE	150
7.1.Kierunki dalszych badań	151
7.2.Realizacja tez pracy	151
BIBLIOGRAFIA	152
DODATEK A – ZAIMPLEMENTOWANE PROGRAMY	160
	100
A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL	
A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL A.1.1.Klasa GeneticAlgorithm	
A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL A.1.1.KLASA GENETICALGORITHM A.1.2.KLASA POPULATION	
A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL A.1.1.KLASA GENETICALGORITHM A.1.2.KLASA POPULATION A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR	
A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL A.1.1.KLASA GENETICALGORITHM A.1.2.KLASA POPULATION A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST	
A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL. A.1.1.KLASA GENETICALGORITHM. A.1.2.KLASA POPULATION. A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR. A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST A.1.5.KLASA IGENETICOPERATOR.	
 A.1.Budowa biblioteki GeneticAlgorithm.dll. A.1.1.Klasa GeneticAlgorithm. A.1.2.Klasa Population. A.1.3.Klasy ICreature oraz ICreaturesGenerator. A.1.4.Klasy IObjectiveFunctionDeterminator, IObjetiveFunction oraz IFitnessT A.1.5.Klasa IGeneticOperator. A.1.6.Klasa IPhenotype. 	
A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL. A.1.1.KLASA GENETICALGORITHM. A.1.2.KLASA POPULATION. A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR. A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST A.1.5.KLASA IGENETICOPERATOR. A.1.6.KLASA IPHENOTYPE. A.2.BUDOWA PROGRAMU REPGEN.	
 A.1.Budowa biblioteki GeneticAlgorithm.dll A.1.1.Klasa GeneticAlgorithm A.1.2.Klasa Population A.1.3.Klasy ICreature oraz ICreaturesGenerator A.1.4.Klasy IObjectiveFunctionDeterminator, IObjetiveFunction oraz IFitnessT A.1.5.Klasa IGeneticOperator A.1.6.Klasa IPhenotype A.2.Budowa programu RepGen A.3.Budowa programu ScanRep 	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL. A.1.1.KLASA GENETICALGORITHM. A.1.2.KLASA POPULATION. A.1.2.KLASA POPULATION. A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR. A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST A.1.5.KLASA IGENETICOPERATOR. A.1.6.KLASA IGENETICOPERATOR. A.2.BUDOWA PROGRAMU REPGEN. A.3.1.KLASA IMENU. 	
 A.1.Budowa biblioteki GeneticAlgorithm.dll A.1.1.Klasa GeneticAlgorithm. A.1.2.Klasa Population. A.1.3.Klasy ICreature oraz ICreaturesGenerator. A.1.4.Klasy IObjectiveFunctionDeterminator, IObjetiveFunction oraz IFitnessT A.1.5.Klasa IGeneticOperator. A.1.6.Klasa IPhenotype. A.2.Budowa programu RepGen. A.3.Budowa programu ScanRep. A.3.1.Klasa IMenu. A.3.2.Klasy Repository oraz Cluster. 	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL. A.1.1.KLASA GENETICALGORITHM. A.1.2.KLASA POPULATION. A.1.2.KLASA POPULATION. A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR. A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST A.1.5.KLASA IGENETICOPERATOR. A.1.6.KLASA IGENETICOPERATOR. A.1.6.KLASA IPHENOTYPE. A.2.BUDOWA PROGRAMU REPGEN. A.3.1.KLASA IMENU. A.3.2.KLASY REPOSITORY ORAZ CLUSTER. A.3.3.KLASY POINT ORAZ POINTSET. 	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL. A.1.1.KLASA GENETICALGORITHM. A.1.2.KLASA POPULATION. A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR. A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST A.1.5.KLASA IGENETICOPERATOR. A.1.6.KLASA IGENETICOPERATOR. A.1.6.KLASA IPHENOTYPE. A.2.BUDOWA PROGRAMU REPGEN. A.3.1.KLASA IMENU. A.3.2.KLASY REPOSITORY ORAZ CLUSTER. A.3.4.KLASA IALGORITHM. 	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL. A.1.1.KLASA GENETICALGORITHM. A.1.2.KLASA POPULATION. A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR. A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST A.1.5.KLASA IGENETICOPERATOR. A.1.6.KLASA IPHENOTYPE. A.2.BUDOWA PROGRAMU REPGEN. A.3.BUDOWA PROGRAMU SCANREP. A.3.1.KLASA IMENU. A.3.2.KLASY REPOSITORY ORAZ CLUSTER. A.3.4.KLASA IALGORITHM. A.3.5.KLASA GENERATOR. 	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL. A.1.1.KLASA GENETICALGORITHM. A.1.2.KLASA POPULATION. A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR. A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST A.1.5.KLASA IGENETICOPERATOR. A.1.6.KLASA IPHENOTYPE. A.2.BUDOWA PROGRAMU REPGEN. A.3.BUDOWA PROGRAMU SCANREP. A.3.1.KLASA IMENU. A.3.2.KLASY REPOSITORY ORAZ CLUSTER. A.3.4.KLASA IALGORITHM. A.3.5.KLASA GENERATOR. A.4.PLIK Z REPOZYTORIUM PRÓBKOWANIA PRZESTRZENI. 	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL. A.1.1.KLASA GENETICALGORITHM. A.1.2.KLASA POPULATION. A.1.3.KLASY ICREATURE ORAZ ICREATURESGENERATOR. A.1.4.KLASY IOBJECTIVEFUNCTIONDETERMINATOR, IOBJETIVEFUNCTION ORAZ IFITNESST A.1.5.KLASA IGENETICOPERATOR. A.1.6.KLASA IPHENOTYPE. A.2.BUDOWA PROGRAMU REPGEN. A.3.BUDOWA PROGRAMU SCANREP. A.3.1.KLASA IMENU. A.3.2.KLASY REPOSITORY ORAZ CLUSTER. A.3.4.KLASA IALGORITHM. A.3.5.KLASA GENERATOR. A.4.PLIK Z REPOZYTORIUM PRÓBKOWANIA PRZESTRZENI. A.5.PLIK Z INFORMACJAMI DO GENEROWANIA REPOZYTORIUM. 	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL	
 A.1.BUDOWA BIBLIOTEKI GENETICALGORITHM.DLL	

1. Wstęp

Przedmiotem niniejszej rozprawy doktorskiej było zaprojektowanie oraz implementacja nowej metody oceny jakości wyników uzyskiwanych w eksperymentach fizyki jądrowej.

Pomiary wzbudzeń kulombowskich są techniką badania elektromagnetycznej struktury jąder atomowych. Bezpośrednim celem eksperymentów jest wyznaczenie, na podstawie zebranych danych eksperymentalnych, optymalnego zestawu elementów macierzowych przejść elektromagnetycznych między stanami energetycznymi badanego izotopu. Pośrednio charakteryzują one kształt jądra atomowego.

Do znajdowania optymalnych wartości powyższych parametrów zaimplementowany został algorytm genetyczny, który jako funkcji oceny używa testu χ^2 zdefiniowanego jako odstępstwo danych eksperymentalnych od wyliczeń teoretycznych (przy użyciu proponowanego zestawu elementów macierzowych, **ME**). Do obliczania wartości χ^2 (**ME**) standardowo stosuje się sprawdzone od strony interpretacji fizycznych i stabilności numerycznej oprogramowanie GOSIA autorstwa dr. hab. Tomasza Czosnyki [10].

Możliwość oszacowania niepewności wyznaczenia wartości elementów macierzowych jest koniecznym elementem analizy danych doświadczalnych.

Głównym celem pracy była prezentacja Metody AH, opracowanej do określania niepewności pomiaru wartości elementów macierzowych wyznaczonych przy użyciu zaimplementowanego algorytmu genetycznego, tj. do analizy kształtu funkcji celu wokół znalezionego optimum. Przy pomocy analizy histogramowej, teorii grafów oraz statystyki matematycznej wykazano, że można w tym celu wykorzystać próbkowanie przestrzeni rozwiązań przez algorytm genetyczny w procesie optymalizacji – nie są potrzebne żadne dodatkowe wywołania funkcji celu.

Aproksymacja kształtu funkcji celu wokół optimum jest wykonywana z wykorzystaniem autorskiego algorytmu FLA (ang. *Front-Line Algorithm*). Wskazuje on punkty z repozytorium próbkowania przestrzeni algorytmem genetycznym leżące najbliżej miejsca przecięcia powierzchni χ^2 z płaszczyzną progową, dla której przeprowadza się estymację niepewności pomiarowej.

Stworzone narzędzia dają możliwość weryfikacji wyników uzyskanych w przeprowadzonych już eksperymentach, jak również ułatwiają analizę danych z nowych pomiarów.

Opracowana heurystyczna metoda estymacji niepewności może być potencjalnie zastosowana do innych problemów programowania nieliniowego, gdzie konieczna jest aproksymacja kształtu funkcji celu wokół znalezionego optimum.

1.1. Tezy pracy

Tezy pracy są następujące:

- Możliwe jest oszacowanie niepewności wyznaczenia wartości elementów macierzowych w analizie danych z pomiarów wzbudzeń kulombowskich w oparciu jedynie o próbkowanie przestrzeni rozwiązań przez algorytm genetyczny.
- Dokładność wyznaczenia niepewności przedstawioną Metodą AH jest porównywalna z innymi metodami wykorzystywanymi do estymacji niepewności wartości parametrów, a czas niezbędny do wykonania obliczeń jest znacząco krótszy.

1.2. Układ pracy

Zawartość pracy przedstawia się następująco: W rozdziale 2. zaprezentowano metodę wzbudzeń kulombowskich, jako technikę badania kształtu jądra atomowego. Przedstawiono proces analizy danych eksperymentalnych, w szczególności dotychczasową metodę analizy danych.

W rozdziale 3. opisano zastosowaną technikę optymalizacji – wyznaczanie wartości elementów macierzowych przy użyciu algorytmu genetycznego. Zdefiniowano model problemu, przedstawiono szczegóły budowy i implementacji algorytmu genetycznego oraz program JACOB, realizujący zadanie optymalizacji.

W rozdziałach 4. i 5. zaprezentowano Metodę AH. W rozdziale 4. znajduje się opis modułu przetwarzania wstępnego oraz modułu analizy próbkowania przestrzeni optymalizacyjnej, a w rozdziale 5. przedstawiono estymację niepewności parametrów dla zadanej płaszczyzny progowej wraz ze schematem proponowanego algorytmu.

W rozdziale 6. opisano zastosowanie Metody AH do wyznaczenia niepewności w analizie danych z pomiaru wzbudzenia kulombowskiego ¹³²Ba.

W rozdziale 7. podsumowano zalety i wady zaproponowanego rozwiązania oraz zebrano wnioski i wskazówki dla przyszłych badań.

2. Wzbudzenia kulombowskie

Metoda wzbudzeń kulombowskich jest techniką eksperymentalnego badania elektromagnetycznej struktury jądra atomowego [2], [3]. Pozwala ona na wyznaczenie zredukowanych elementów macierzowych przejść elektromagnetycznych pomiędzy stanami jądrowymi. Dysponując zestawem zredukowanych elementów macierzowych, opisujących strukturę elektromagnetyczną badanego jądra, możliwe jest, poprzez zastosowanie techniki nieważonych energetycznie reguł sum, określenie kształtu jądra atomowego (rozkładu ładunku) indywidualnie dla każdego stanu wzbudzonego [18].

Warto podkreślić, że metoda wzbudzeń kulombowskich jest jedyną techniką eksperymentalną umożliwiającą niezależny od modeli jądrowych pomiar parametru trójosiowości jądra atomowego [16], [17] i stanowi tym samym najlepsze narzędzie do badań nad ewolucją i koegzystencją kształtu [32], [33], [49]. Pozwala ona również wyznaczyć nie tylko wartość bezwzględną, ale również znak elementów macierzowych. Znajomość znaku jest szczególnie istotna przy wzbudzeniach wielokrotnych – pozwala uwzględnić efekty interferencyjne wynikające z faz funkcji falowych poszczególnych stanów wzbudzonych [50].

Metoda wzbudzeń kulombowskich pozwala także na bezpośrednie wyznaczenie wartości elementów macierzowych dla różnych multipolowości, co stwarza możliwość weryfikacji współczynników zmieszania, zmierzonych metodami analizy rozkładów kątowych promieniowania gamma [25], [26], [44], [59].

Wreszcie, jedynie metodą wzbudzeń kulombowskich można eksperymentalnie wyznaczyć wartości diagonalnych elementów macierzowych [62]. Poznanie struktury elektromagnetycznej stanowi ważny element badań z zakresu fizyki jądra atomowego, prowadzących do stworzenia modelu sił jądrowych.

Wzbudzenia kulombowskie pozwalają, w sposób niezależny od przyjętego modelu oddziaływań silnych, wyznaczyć właściwości elektromagnetyczne jąder w stanach: podstawowym i wzbudzonych. Jest to metoda pozwalająca wyznaczyć parametry kształtu jąder zdeformowanych kwadrupolowo (przyjmujących kształt elipsoidy) [24], [37]. Umożliwia również badanie deformacji oktupolowej (kształt jąder atomowych podobny do gruszki) [15].

Główną trudność w modelowo-niezależnej analizie danych eksperymentalnych stanowi duża liczba zredukowanych elementów macierzowych, mających wpływ na proces wzbudzenia badanego jądra. Dla niektórych badanych obecnie przypadków, liczba ta może przekraczać 100. Proces zmiany obsadzeń stanów wzbudzonych w jądrze atomowym jest zjawiskiem przewidywanym przez mechanikę kwantową [27]. Pełny kwantowy opis tego zjawiska jest jednak bardzo skomplikowany i przez to nieefektywny w analizie danych z eksperymentów wzbudzeń kulombowskich. Stosowane jest przybliżenie półklasyczne, które polega na opisie ruchu rozpraszanych cząstek metodami mechaniki klasycznej, natomiast zjawiska zachodzące w uczestniczących w reakcji jądrach są badane metodami mechaniki kwantowej. Podejście to jest szczegółowo przedyskutowane w monografii Aldera i Winthera [1].

Metoda wzbudzeń kulombowskich jest z powodzeniem stosowana i rozwijana w Środowiskowym Laboratorium Ciężkich Jonów Uniwersytetu Warszawskiego (ŚLCJ UW) [53]. Laboratorium dysponuje cyklotronem U200P. Maksymalne dostępne wartości energii wiązki sięgają 10 MeV na nukleon. Pełen opis działania Warszawskiego Cyklotronu można znaleźć w [42].

Badania metodą wzbudzeń kulombowskich prowadzone są na stanowisku eksperymentalnym ze skonstruowaną specjalnie do tego celu aparaturą pomiarową. Dzięki niej można obserwować zachowanie cząstek jednego izotopu, rozpędzonych w akceleratorze i zogniskowanych w wiązkę skierowaną na umieszczoną wewnątrz aparatury tarczę wykonaną z innego izotopu [28].

2.1. Pomiar i zbieranie danych

2.1.1. Energia wiązki

Energię cząstek przyspieszanych w cyklotronie wyznacza się stosując wzór:

$$\frac{E}{A} = K \cdot \left(\frac{Q}{A}\right)^2 \tag{2.1}$$

gdzie:

E – energia kinetyczna jonu [MeV],

A – liczba masowa [amu],

Q – ładunek przyspieszanego jonu,

K – charakterystyczna stała cyklotronowa, która w przypadku cyklotronu w ŚLCJ UW zawiera się w przedziale: $120 \le K \le 160$ w zależności od promienia wyprowadzenia wiązki.

Wzbudzeniem kulombowskim jądra atomowego nazywamy proces, w wyniku którego jądra tarczy i pocisku oddziałują ze sobą jedynie za pośrednictwem pola elektromagnetycznego – krótkozasięgowe siły jądrowe można zaniedbać [45]. Proces ten zachodzi przy odpowiednio dużej separacji powierzchni zderzających się obiektów. Przyjęto, że jądro tarczy i jądro

pocisku oddziałują czysto elektromagnetycznie, gdy odległość najmniejszego zbliżenia d_{min} między środkami oddziałujących obiektów jest większa niż:

$$d = 1.25 \cdot \left(A_1^{1/3} + A_2^{1/3}\right) + 5.0 \left[fm\right]$$
(2.2)

gdzie:

 A_1 – liczba masowa jądra pocisku,

 $A_2\;-$ liczba masowa jądra tarczy.

Powyższe wyrażenie nosi nazwę *kryterium Cline'a* [7]. Jest to empiryczna zależność, między liczbami atomowymi jąder pocisku i tarczy, opisująca minimalną odległość (w femtometrach), dla której oddziaływania silne nie mają wpływu na obserwowane wzbudzenia.

Na tej podstawie można wyznaczyć wartość maksymalnej dopuszczalnej energii wiązki pocisku w zależności od kąta rozpraszania.

$$E_{max}(\theta_{CM}) = 0.72 \cdot \frac{Z_1 \cdot Z_2}{d} \cdot \frac{A_1 + A_2}{A_2} \cdot \left[1 + \frac{1}{\sin\left(\frac{\theta_{CM}}{2}\right)}\right] [MeV]$$
(2.3)

gdzie:

 Z_1 – liczba atomowa jądra pocisku,

 Z_2 – liczba atomowa jądra tarczy,

 $\boldsymbol{\theta}_{C\!M}$ – kąt rozproszenia cząstki w układzie środka masy.

Energia ta jest najmniejsza w przypadku rozpraszania wstecznego, w którym jądro pocisku na skutek oddziaływania kulombowskiego z jądrem tarczy zawraca, czyli zachowuje ten sam kierunek, lecz zmienia zwrot na przeciwny. Dla rozpraszania wstecznego wartość maksymalnej energii wiązki w funkcji parametrów jąder pocisku i tarczy wynosi:

$$E_{max} = E_{max} \left(180^{\circ} \right) = 1.44 \cdot \frac{Z_1 \cdot Z_2}{d} \cdot \frac{A_1 + A_2}{A_2} \left[MeV \right]$$
(2.4)

Tradycyjnie energię maksymalną dla rozpraszania wstecznego podaje się jako progową dla eksperymentów wzbudzeń kulombowskich.

2.1.2. Eksperyment

Rozpędzona w akceleratorze wiązka ciężkich jonów (nazywanych pociskami) jest ogniskowana na umieszczonych w komorze pomiarowej jądrach innego pierwiastka (tarczy). Badany izotop może pełnić rolę pocisku albo tarczy [19].

Przedstawione na Rys. 2.1 oraz Rys. 2.2 stanowiska eksperymentalne zbudowane są wokół komory rozproszeń kończącej tunel próżniowy prowadzący wiązkę z akceleratora. Przelatujące blisko jąder tarczy jądra pocisku oddziałują elektromagnetycznie. Następuje proces wzbudzenia stanów jądrowych oraz wybicie jądra z tarczy.

Wzbudzone wskutek oddziaływań kulombowskich między jądrami pocisku i tarczy, jądra atomowe ulegają deekscytacji. Emitują one kwanty promieniowania gamma o energiach równych różnicom między stanami wzbudzonymi w jądrze [50]. Typowy czas trwania tego procesu wynosi od kilku do kilkuset pikosekund i jest znacznie dłuższy niż czas procesu wzbudzenia, który jest rzędu 10⁻¹⁹ s.

Przykładowy schemat poziomów energetycznych stanów wzbudzonych i podstawowych pokazano na Rys. 2.4. Zawiera on również wartości energii kwantów emitowanych w czasie deekscytacji oraz dla poszczególnych stanów: wypadkowy moment pędu i parzystość. Szczegółowo jest to opisane w [48]. Dane potrzebne do zbudowania schematu poziomów można znaleźć w [51].

2.1.3. Rejestracja danych eksperymentalnych

Do rejestracji powstałych w trakcie eksperymentu kwantów promieniowania gamma używa się detektorów germanowych (vide Rys. 2.5). Natomiast rozproszone wskutek oddziaływań kulombowskich cząstki są rejestrowane przy pomocy detektorów cząstek (vide Rys. 2.6). Szczegółowy opis ich działania znajduje się w [14]. Detektory są rozmieszczone względem osi wiązki.

Sygnały z detektorów po pierwszej fazie wzmocnienia przez przedwzmacniacz umieszczony blisko detektora są przesyłane do wzmacniaczy szybkich (różniczkujących) oraz wzmacniaczy spektroskopowych (całkujących po długim czasie kształtowania). Celem pierwszych jest pozyskanie, po dyskretyzacji sygnału analogowego, informacji logicznej o fakcie i czasie rejestracji. Informacja ta jest potrzebna do zbudowania logiki warunku rejestracji zdarzeń (ang. *trigger condition*). Celem drugich jest uzyskanie informacji na temat energii zarejestrowanych cząstek lub kwantów gamma. Amplituda tak wzmocnionego sygnału jest przekształcana przetwornikami analogowo-cyfrowymi i zapisywana do pliku. Na Rys. 2.3 widoczne są elektroniczne moduły pomiarowe używane w układzie OSIRIS-II.

Moduł akwizycji danych zapisuje zdarzenia obserwowane za pomocą detektorów. Jako zdarzenie rejestruje się jednoczesne (w określonym oknie czasowym, np. 200 ns) zaobserwowanie cząstki oraz co najmniej jednego kwantu promieniowania gamma.



Rys. 2.1 Schemat oraz zdjęcie stanowiska pomiarowego OSIRIS-II [57]



3-



Rys. 2.2 Schemat oraz zdjęcie stanowiska pomiarowego CUDAC



Rys. 2.3 Elektronika OSIRIS-II



Rys. 2.4 Schemat poziomów w jądrze ¹⁰⁰Mo



Rys. 2.5 Detektor germanowy

Rys. 2.6 Komora rozproszeń (tylna połówka) [35]







Rys. 2.8 Widma promieniowania gamma w zależności od kąta rozproszenia [6]

W typowym eksperymencie wzbudzeń kulombowskich w ŚLCJ UW rejestruje się około 25·10⁶ zdarzeń. Zbiór zdarzeń jest przedmiotem dalszej analizy.

2.2. Analiza danych eksperymentalnych

Ze zbioru danych uzyskanych w opisanym pomiarze tworzone jest dyskretne widmo promieniowania gamma, którego przykład przedstawiono na Rys. 2.7. Jest ono dyskretną funkcją energii zaobserwowanych kwantów gamma w liczbę obserwacji. Dziedzina tej funkcji jest kalibrowana, a następnie przekształcana liniowo, aby odpowiadała energiom przejść elektromagnetycznych. Pierwotnie są to dane cyfrowe z detektorów. Rozdzielczość (dokładność dyskretyzacji) podyktowana jest rozdzielczością przetworników analogowo-cyfrowych. Widma promieniowania gamma można także tworzyć zależnie od kąta rozproszenia cząstek (vide Rys. 2.8). Pozwala to na uwzględnienie w analizie zróżnicowania siły oddziaływania związanej z różną kinematyką procesu rozproszenia dla różnych kątów rozproszenia pocisku.

Natężenie promieniowania gamma niesie informację o procesie wzbudzenia, który w mechanice kwantowej jest opisywany elementem macierzowym przejścia elektromagnetycznego pomiędzy stanami w jądrze. Wartość elementu jest związana z prawdopodobieństwem przejść między stanami wzbudzonymi w jądrze [1].

2.2.1. Charakterystyka danych podlegających analizie

Podstawą do dalszego procesu analizy są:

- zaobserwowane w eksperymencie liczby zliczeń w liniach widmowych promieniowania gamma;
- parametry geometrii układu pomiarowego;
- struktura stanów wzbudzonych badanego jądra;
- informacje z wcześniejszych eksperymentów.

Aby we właściwy sposób interpretować obserwowane kwanty gamma, potrzebna jest informacja o strukturze jądra atomowego, tzw. schemat poziomów. Zawiera on informację o energiach stanów wzbudzonych jądra, a także ich spinach i parzystościach (vide Rys. 2.4).

Dodatkową informację stanowią dane znane w literaturze, uzyskane z innych typów eksperymentów, na przykład pomiary czasów życia stanów jądrowych. Podobnie jak współczynniki zmieszania przejść elektromagnetycznych oraz stosunki rozgałęzień, czasy życia są wyrażane przez elementy macierzowe i stanowią niezależne więzy pozwalające ograniczyć przestrzeń potencjalnych rozwiązań.

2.2.2. Cel analizy

2.2.2.a. Parametry zadania optymalizacyjnego

Szukane elementy macierzowe są często określane przez specjalistów od analizy danych eksperymentalnych jako:

- wektor / zestaw / zbiór elementów macierzowych;
- albo wektor / zestaw / zbiór parametrów.

Ze względu na interdyscyplinarny charakter pracy określenia te będą stosowane jako synonimy. Należy podkreślić, że termin "wektor elementów macierzowych" nie odnosi się do

wiersza (lub kolumny) z konkretnej jednej macierzy. Jest to, zdefiniowany przez fizyków analizujących dane eksperymentalne, <u>zbiór uporządkowanych, niepowiązanych ze sobą parametrów</u>, określanych w fizyce kwantowej terminem "elementy macierzowe przejść elektromagnetycznych pomiędzy stanami wzbudzonymi w jądrze atomowym". Z punktu widzenia informatyki jest to zestaw parametrów (liczb rzeczywistych), których poszukiwanie (sub)optymalnych wartości stanowi zadanie optymalizacyjne.

2.2.2.b. Funkcja celu zadania optymalizacyjnego

Jako miarę dopasowania proponowanego wektora elementów macierzowych do zmierzonych danych doświadczalnych stosuje się zmodyfikowany test χ^2 . Pozwala on określić, w jakim stopniu wyliczenia teoretycznych modeli fizyki jądrowej dla proponowanych wartości elementów macierzowych odwzorowują dane zebrane w eksperymencie. Porównania dokonuje się poprzez sumę kwadratów odstępstw wielkości wyliczonych na podstawie założeń modelowych od danych doświadczalnych, unormowanych do dokładności pomiaru [9].

$$\chi^{2}(\boldsymbol{M}\boldsymbol{E}) = \frac{1}{N} \left| \sum_{I_{e}, I_{d}} w_{I_{e}, I_{d}} \sum_{k(I_{e}, I_{d})} \frac{1}{\sigma_{k}^{2}} \left(C_{I_{e}, I_{d}} Y_{k}^{c}(\boldsymbol{M}\boldsymbol{E}) - Y_{k}^{e} \right)^{2} + \sum_{j} \left(\frac{Y_{j}^{c}(I_{e}, I_{d}, \boldsymbol{M}\boldsymbol{E})}{Y_{n}^{c}(I_{e}, I_{d})} - u(I_{e}, I_{d}) \right)^{2} \cdot \frac{1}{u^{2}(I_{e}, I_{d})} + \sum_{i} w_{i} \sum_{n_{i}} \left(\frac{d_{n_{i}}^{c}(\boldsymbol{M}\boldsymbol{E}) - d_{n_{i}}^{e}}{\sigma_{n_{i}}^{2}} \right)^{2} \cdot \frac{1}{\sigma_{n_{i}}^{2}} \right|$$

$$(2.5)$$

gdzie:

ME – wektor elementów macierzowych (ang. Matrix Elements),

N – liczba danych doświadczalnych (wszystkich intensywności przejść γ, współczynników rozgałęzień, czasów życia, współczynników zmieszania oraz znanych elementów macierzowych),

$$I_e$$
 – liczba analizowanych eksperymentów,

$$I_d$$
 – liczba detektorów promieniowania γ ,

 ${}^{W_{I_{e},I_{d}}}$ – wagi dla poszczególnych zbiorów danych (osobno dla poszczególnych eksperymentów oraz detektorów promieniowania γ),

$$C_{I_e, I_d}$$
 – współczynniki normalizacyjne,

 Y^c , Y^e – obliczone oraz eksperymentalne intensywności linii γ ,

 Y_n^c – intensywność linii normalizacyjnej,

 $u(I_e, I_d)$ – tzw. "upper limit" - jest to stosunek intensywności najsłabszej linii możliwej do zaobserwowania w eksperymencie do intensywności linii normalizacyjnej. Wprowadza się go, aby uniknąć rozwiązań zawierających za duże elementy macierzowe, dla przejść, które nie są obserwowane w pomiarze,

W tradycyjnym teście χ^2 , określoną we wzorze (2.5) statystykę najmniejszych kwadratów normalizuje się względem liczby stopni swobody (czyli do liczby danych pomniejszoną o liczbę parametrów). W analizie danych z pomiarów wzbudzeń kulombowskich takie podejście jest niemożliwe, ponieważ poszczególne elementy macierzowe mają bardzo różny wpływ na proces wzbudzenia i deekscytacji. Liczba stopni swobody jest nieznana z uwagi na występowanie w opisie elementów macierzowych mających zaniedbywalny wpływ na wynik oraz wzajemną korelację elementów. Z tego powodu we wzorze (2.5) stosuje się normalizację względem liczby wszystkich danych doświadczalnych (zmierzonych intensywności przejść, czasów życia, stosunków rozgałęzień, współczynników zmieszania). W granicy dużej liczby danych oba sposoby normalizowania są równoważne [8], [46].

Wartość oczekiwana tradycyjnego testu χ^2 , znormalizowanego do liczby stopni swobody, wynosi 1. W analizie danych z pomiarów wzbudzeń kulombowskich uznana za optymalną (minimalną) wartość χ^2 zazwyczaj jest zbliżona do jedności – jako suma liczb nieujemnych jest liczbą nieujemną, a z praktyki eksperymentalnej wynika, że nie przekracza liczby 2. Na potrzeby niniejszej pracy zakłada się, że dla dowolnej analizy danych z pomiarów wzbudzeń kulombowskich:

$$\min\left(\chi^2(ME)\right) \in (0,3) \tag{2.6}$$

2.2.2.c. Cele analizy danych

Bezpośrednimi celami analizy danych z pomiarów wzbudzeń kulombowskich są:

• znalezienie takiego zestawu elementów macierzowych przejść elektromagnetycznych pomiędzy stanami wzbudzonymi w badanym jądrze, który z możliwie dużą dokładnością pozwala odtworzyć procesy wzbudzenia i deekscytacji obserwowane w eksperymencie. Jest to tożsame ze znalezieniem minimum opisanego powyżej testu χ^2 rozumianego jako funkcja, której argumentami są elementy macierzowe;

 oszacowanie niepewności pomiaru poprzez ocenę dokładności dokonanego wyznaczenia elementów macierzowych na podstawie zebranych danych doświadczalnych. Jest ono tożsame z określeniem kształtu powierzchni funkcji χ² w bezpośrednim otoczeniu znalezionego optymalnego wektora elementów macierzowych.

Wyznaczony wektor elementów macierzowych może poprzez Metodę Kwadrupolowych Niezmienników Rotacyjnych [18] służyć do wyznaczenia parametrów kształtu jądra atomowego, a ściślej jego ładunku w poszczególnych stanach wzbudzonych i podstawowym.

2.2.3. Dotychczasowa metoda analizy

2.2.3.a. Schemat analizy danych

Schemat poprzedniego procesu analizy danych doświadczalnych zawiera Rys. 2.9.



Rys. 2.9 Dotychczasowa metoda analizy danych eksperymentalnych

W doświadczeniu fizycznym bezpośrednio mierzy się intensywności przejść elektromagnetycznych między stanami wzbudzonymi jądra w zależności od kinematyki rozproszenia (kwanty promieniowania γ) [29], [30]. Schemat poziomów jest już z reguły znany. W analizie danych można porównywać także inne wielkości spektroskopowe znane z wcześniejszych eksperymentów: czasy życia, stosunki rozgałęzień, współczynniki zmieszania. Właściwe obliczenia poprzedza przygotowanie zbioru danych wejściowych. Ze względu na ich dużą ilość jest to zadanie skomplikowane. W szczególności od sposobu zdefiniowania geometrii układu eksperymentalnego znacząco zależy czas trwania obliczeń.

Następnie poszukuje się takiego zestawu elementów macierzowych **ME**, dla którego opisana w poprzednim punkcie wartość $\chi^2(\mathbf{ME})$ jest najmniejsza. Dla znalezionego optymalnego wektora elementów macierzowych szacuje się niepewności ich wyznaczenia. Stosując Kwadrupolowe Reguły Sum określa się rozkład ładunku elektromagnetycznego badanego jądra.

2.2.3.b. Program GOSIA

Podstawowym narzędziem do analizy danych z pomiarów wzbudzeń kulombowskich jest Program Kanałów Sprzężonych z minimalizacją Metodą Najmniejszych Kwadratów – GOSIA [9]. Napisany on został we współpracy polsko-amerykańskiej w Rochester w latach 80. ubiegłego wieku. Autorami są Tomasz Czosnyka, Douglas Cline i Ching-Yen Wu [10]. Przez kolejne dziesięciolecia, program był wielokrotnie udoskonalany i rozbudowywany.

Ma on rozbudowaną funkcjonalność i znajduje szerokie zastosowanie w całym procesie analizy danych z pomiarów wzbudzeń kulombowskich (np. we wspomnianym powyżej przygotowaniu danych wejściowych) [11]. Zagadnienia te jednak nie stanowią części niniejszej dysertacji i dlatego nie będą szerzej przedstawiane. Istotne z punktu widzenia niniejszej pracy są zawarte w programie GOSIA procedury: poszukiwania optymalnego wektora elementów macierzowych oraz estymacji niepewności pomiaru.

Dla zadanego wektora elementów macierzowych, program GOSIA rozwiązuje numerycznie równanie wzbudzenia, uzyskując obsadzenia poszczególnych stanów wzbudzonych. Mając populację stanów i powtórnie korzystając z zestawu elementów macierzowych oraz danych o położeniu i geometrii detektorów kwantów γ , oblicza intensywności przejść między stanami wzbudzonymi w badanym izotopie. Obliczone wartości są porównywane ze zmierzonymi za pomocą opisanego powyżej testu χ^2 .

Jeśli porównanie wielkości wyznaczonych na podstawie modeli teoretycznych z wartościami zmierzonymi jest niezadowalające, wówczas modyfikuje się wejściowy zestaw elementów macierzowych. Całą procedurę powtarza się aż do uzyskania zadowalającej zgodności obliczeń z pomiarami.

2.2.3.c. Szukanie optymalnego wektora elementów macierzowych

Opisana procedura stanowi klasyczny przykład zadania optymalizacyjnego (minimalizacji), w którym dla wejściowego zestawu parametrów (wektora elementów macierzowych **ME**) wyznaczana jest wartość funkcji celu χ^2 (**ME**).



Rys. 2.10 Schemat wyznaczania wartości funkcji celu

W programie GOSIA zaimplementowano gradientową metodę poszukiwania optymalnego wektora elementów macierzowych. Metody gradientowe mają charakter przeszukiwania lokalnego, wrażliwego na utknięcie w minimum lokalnym. Kluczowe jest wskazanie takiego początkowego wektora elementów macierzowych, który leży w basenie przyciągania globalnego minimum funkcji χ^2 .

Metoda wskazywania początkowego wektora elementów macierzowych opiera się w na:

- wynikach wcześniejszych eksperymentów wzbudzeń kulombowskich;
- przewidywaniach wynikających z modeli teoretycznych (model rotacyjny [4], model Dawydowa-Filipowa [12], model Dobaczewskiego-Rochozińskiego-Srebrnego [13]);
- wyznaczonych eksperymentalnie stosunkach intensywności oraz współczynnikach zmieszania;
- wiedzy fizyków prowadzących analizę danych eksperymentalnych;
- technice multi-startu wielokrotnym przeprowadzaniu minimalizacji metodą gradientową dla punktów startowych leżących w różnych miejscach przestrzeni rozwiązań;
- sztucznym zawężaniu przestrzeni rozwiązań dopuszczalnych poprzez próby wprowadzania a priori korelacji między poszczególnymi parametrami i przeprowadzaniu wstępnego procesu optymalizacji dla zmniejszonej liczby parametrów.

Wszystkie te zabiegi mają na celu zwiększenie prawdopodobieństwa trafienia w obszar przyciągania globalnego minimum. Są jednak obciążone następującymi wadami:

- duża czasochłonność ręcznie sterowanego procesu optymalizacji;
- brak pewności, że znalezione rozwiązanie jest globalnie optymalne;
- zakres stosowalności ograniczony jedynie do przypadków eksperymentalnych, w których jest: mała liczba szukanych elementów macierzowych albo duża liczba danych w literaturze dotyczących badanego izotopu.

2.2.3.d. Rozwój fizyki eksperymentalnej oraz narzędzi do analizy danych

Program GOSIA powstał ponad 30 lat temu. Przez wiele lat zaimplementowana w nim metoda gradientowa była wystarczająca. Jednakże stanowiska eksperymentalne fizyki jądrowej (w tym służące do pomiaru wzbudzeń kulombowskich) są wciąż rozbudowywane na całym świecie. Lawinowo wzrosła ilość danych podlegających analizie w poszczególnych eksperymentach.

Z jednej strony daje to możliwość eksperymentalnego zmierzenia wielu nowych wartości, które wcześniej były jedynie przewidywane teoretycznie¹. Dla techniki pomiarów wzbudzeń kulombowskich jest to np. badanie jąder izotopów silnie zdeformowanych, wyznaczanie parametrów wysokospinowych stanów wzbudzonych. Z drugiej stwarza to nowe wyzwania dla technik analizy danych. W szczególności liczba elementów macierzowych, które można wyznaczyć w oparciu o dane eksperymentalne również znacząco wzrosła.

Z tego powodu konieczne okazało się użycie adaptacyjnej metody poszukiwania optymalnych elementów macierzowych. W ramach pracy magisterskiej zaproponowałem implementację algorytmu genetycznego do analizy danych z pomiarów wzbudzeń kulombowskich [87]. Algorytm ten, wraz z graficznym interfejsem użytkownika, został znacząco rozbudowany i udoskonalony w ramach niniejszej pracy doktorskiej. Modyfikacje zostały szczegółowo opisane w następnym rozdziale.

Użycie algorytmu genetycznego znacząco usprawniło proces wyznaczania elementów macierzowych w oparciu o dane zebrane w trakcie eksperymentu. Procedura ta stanowi jednak jedynie część analizy danych z pomiarów wzbudzeń kulombowskich. Pomaga przy zautomatyzowanym przeszukiwaniu przestrzeni rozwiązań. Pozostałe elementy analizy (np. przygotowywanie danych wejściowych) są nadal realizowane z wykorzystaniem funkcjonalności programu GOSIA.

Wyznaczanie wartości χ^2 dla zadanego wektora elementów macierzowych zostało pozostawione programowi GOSIA – wywoływanemu jako zewnętrzny proces z poziomu aplikacji z zaimplementowanym algorytmem genetycznym. Decyzja ta została podjęta na podstawie sugestii fizyków zajmujących się analizą danych z pomiarów wzbudzeń kulombowskich. Przemawiają za nią następujące argumenty:

¹ Jednym z najbardziej znanych sukcesów fizyki doświadczalnej, będącym następstwem budowy nowych i rozbudowy istniejących stanowisk eksperymentalnych w ostatnich latach, jest eksperymentalne potwierdzenie istnienia bozonu Higgsa [40] oraz [41], którego istnienie fizyka teoretyczna przewidywała od dziesięcioleci.

- program GOSIA ma szerokie grono użytkowników, zajmujących się fizyką eksperymentalną na świecie, którzy nie widzą sensu w nauce nowego narzędzia do kompleksowej analizy danych. Potrzebują jedynie usprawnienia procesu szukania właściwego wektora elementów macierzowych;
- program GOSIA posiada bogaty zakres zastosowań w procesie analizy danych trudno byłoby go całkowicie zastąpić;
- niezawodność i brak błędów w kodzie programu GOSIA zostały potwierdzone przez 30 lat jego użytkowania. Nowa aplikacja zapewne potrzebowałaby wielu lat testów, żeby zyskać zbliżony poziom zaufania wśród użytkowników;
- program GOSIA jest napisany w języku FORTRAN. Szybkość działania języka niskiego poziomu oraz wysoka optymalizacja kodu są praktycznie nie do osiągnięcia w języku wyższego poziomu, takim jak język C++;
- przepisanie na C++ jedynie modułu obliczającego wartość χ² jest niemożliwe ze względów technicznych. Wyznaczenie wartości χ² jest zależne od wielu innych modułów i niemożliwym jest jego "wydzielenie".

Z tych samych powodów ostateczna wersja algorytmu genetycznego, przedstawiona w niniejszej pracy, korzysta z programu GOSIA jako "czarnej skrzynki" do wyznaczania wartości χ^2 (**ME**).

2.2.3.e. Procedura "szybkiej aproksymacji"²

Osobnego omówienia wymaga zaimplementowana w programie GOSIA metoda tzw. szybkiej aproksymacji wartości χ^2 w oparciu o zadany wektor elementów macierzowych. Ta opracowana przez Tomasza Czosnykę procedura jest nowatorska z punktu widzenia użytego w niej przybliżonego modelu opisu wzbudzenia, deekscytacji, rozpraszania cząstek oraz szacowania energii zarejestrowanych kwantów promieniowania γ .

Aproksymacje polegają w szczególności na:

- zastąpieniu dokładnej kalkulacji procesów wzbudzenia i deekscytacji ich szybkim, ale wystarczająco precyzyjnym przybliżeniem – w szczególności przez pomijanie tych elementów, które wnoszą mniejszy ilościowo wkład w numeryczne odtwarzanie procesów;
- pominięciu poprawki relatywistycznej (ang. *recoil-velocity correction*) w opisie rozpraszania cząstek i kwantów promieniowania γ;

² W środowisku użytkowników programu GOSIA określenie "szybka aproksymacja" jest używane jako nazwa własna dla określenia tej konkretnej procedury. Dlatego w tej pracy to określenie będzie również tak rozumiane.

 zastąpieniu całkowania po zależnych od eksperymentu kątach rozproszeń i zakresach energii bombardujących cząstek przez wyznaczenie wartości całki z wykorzystaniem wartości średnich.

Procedura szybkiej aproksymacji stwarza możliwość szybszej eksploracji przestrzeni rozwiązań i stosowania dokładnych obliczeń jedynie w otoczeniu znalezionych potencjalnych miejsc z globalnym minimum $\chi^2(ME)$. Użytkownicy programu GOSIA szacują, że jej zastosowanie przyspiesza obliczenia około 200-krotnie. Jednocześnie niedokładność wyznaczenia wartości $\chi^2(ME)$ jest szacowana na kilka do kilkunastu procent zależnie od danych doświadczalnych (analizowanego eksperymentu) oraz miejsca w przestrzeni parametrów, z którego pochodzi proponowany wektor elementów macierzowych.

2.2.3.f. Szacowanie niepewności pomiaru

Ostatnim elementem, który wymaga omówienia jest dotychczasowa metoda szacowania niepewności wyznaczenia elementów macierzowych. Zaimplementowana w programie GOSIA metoda ma charakter przybliżony. Prawdopodobna rzeczywista dokładność wyznaczenia elementów macierzowych może być nieco gorsza, nie zostało to jednak w żaden sposób oszacowane.

Obliczanie niepewności przebiega w dwóch etapach. Najpierw wyznacza się tzw. "błędy diagonalne" – niepewności uwzględniające wpływ zmian wybranego elementu macierzowego na wartość χ^2 . W drugim etapie wyznacza się niepewności elementów uwzględniając ich wzajemne korelacje. Ten rachunek wykonuje się jedynie dla elementów macierzowych o małych "błędach diagonalnych".

Metoda sprowadza się do przeprowadzenia przecięcia powierzchni χ^2 płaszczyzną prostopadłą do osi wartości funkcji celu na wysokości $+\Delta h$ względem znalezionej najmniejszej wartości χ^2 (zazwyczaj przyjmuje się $\Delta h=1.0$ [36]). Analiza kształtu otrzymanego konturu stanowi metodę estymacji niepewności. Wymaga to wielu kolejnych obliczeń wartości funkcji celu.

Zaproponowana w tej pracy metoda szacowania niepewności wyznaczonych elementów macierzowych polega na wykorzystaniu do analizy kształtu niniejszego konturu próbkowania przestrzeni algorytmem genetycznym wokół znalezionego optimum. Żadne dodatkowe wywołania funkcji celu nie są konieczne. Możliwe jest oszacowanie niepewności w zakresie dotychczas stosowanym w analizie danych z pomiarów wzbudzeń kulombowskich. Metoda stwarza również pewne możliwości rozszerzenia estymacji niepewności eksperymentalnej w przyszłości, w szczególności bardziej precyzyjnego wyznaczania korelacji między poszczególnymi parametrami.

Proponowaną metodę można potencjalnie zastosować do wielu innych zagadnień, dla których po wyznaczeniu (sub)optymalnego zestawu parametrów konieczna jest estymacja niepewności tego wyznaczenia.

2.3. Wnioski

Opisana w tym rozdziale technika eksperymentalnego badania elektromagnetycznej struktury jądra atomowego, to zadanie stawiające szereg wyzwań.

Z jednej strony wymagania narzucane stanowisku eksperymentalnemu i elektronicznemu systemowi akwizycji danych stwarzają konieczność pracy na sprzęcie możliwie najwyższej klasy oraz konieczność wykorzystania go "na granicy" jego dokładności. Na szczęście technika pomiarów wzbudzeń kulombowskich jest z sukcesami rozwijana od kilkudziesięciu lat [5], [21], [22], [23], [52], [58], [60], [64], [65]. Zajmujący się nią fizycy eksperymentalni dobrze radzą sobie z tymi wyzwaniami.

Z drugiej strony, ze względu na ciągły rozwój stanowisk eksperymentalnych na całym świecie, ilość danych pozyskiwanych w każdym eksperymencie rośnie [31], [34], [38], [39], [43], [47], [61], [63], [67]. Stwarza to nowe wyzwania dla procesu analizy danych eksperymentalnych – zarówno w zakresie złożoności obliczeniowej, jak i konieczności poszukiwania oraz wdrażania nowych metod numerycznych.

Jednym z podstawowych narzędzi do analizy danych z pomiarów wzbudzeń kulombowskich jest Program Kanałów Sprzężonych z minimalizacją Metodą Najmniejszych Kwadratów GOSIA. Ma on bogatą funkcjonalność i jest wykorzystywany na wielu etapach analizy danych eksperymentalnych. Mimo że powstał ponad 30 lat temu, wciąż dobrze spełnia stawiane przed nim oczekiwania.

Jedynym wyzwaniem, z którym program GOSIA radzi sobie coraz gorzej, jest wyznaczanie, w oparciu o zebrane dane eksperymentalne, wartości elementów macierzowych przejść elektromagnetycznych między stanami wzbudzonymi w badanym izotopie.

Ze względu na postęp technologiczny, możliwe staje się badanie bardziej złożonych przypadków fizycznych i (co za tym idzie) wyznaczanie większej liczby elementów macierzowych w jednym eksperymencie. Dotychczasowa technika przeszukiwania przestrzeni rozwiązań metodą gradientową jest z oczywistych względów narażona na utknięcie w minimum lokalnym. Z uwagi na rosnącą liczbę wymiarów (liczbę elementów

macierzowych), stosowane metody radzenia sobie z tym problemem (staranne dobierania punktu startowego, multi-start) nie są już wystarczające. Konieczne okazało się użycie inteligentnej metody wyznaczania rozwiązania globalnie optymalnego oraz opracowania powiązanego z nią nowego algorytmu estymacji niepewności wyznaczania wartości elementów macierzowych w oparciu o dane eksperymentalne.

W ŚLCJ UW³ rozpoczęto poszukiwania lepszej metody przeszukiwania przestrzeni wartości χ^2 . Podjęta przeze mnie współpraca z ŚLCJ UW zaowocowała stworzeniem aplikacji JACOB, która wykorzystywała do minimalizacji algorytm genetyczny z reprezentacją binarną. Opracowana nowa technika analizy danych z pomiarów wzbudzeń kulombowskich była prezentowana na europejskiej konferencji "EURISOL-EURONS Town Meeting" w 2007 roku [86]. Zastosowanie algorytmu genetycznego do analizy wzbudzeń kulombowskich zostało przedstawione na międzynarodowej konferencji "GOSIA Workshop 2008" [54], [55], [56], [66]. Stosowany wówczas do analizy danych eksperymentalnych algorytm genetyczny został znacząco rozbudowany (m. in. o reprezentację rzeczywistą, lepsze operatory genetyczne, moduł akwizycji punktów z przestrzeni χ^2). Szczegóły przedstawiono w rozdziale 3.

Celem pracy doktorskiej było opracowanie nowej metody estymacji niepewności wyznaczenia elementów macierzowych w oparciu o dane zebrane w eksperymencie fizycznym. W nowej metodzie do szacowania niepewności wykorzystywane jest jedynie próbkowanie przestrzeni rozwiązań algorytmem genetycznym, zebrane w postaci repozytorium punktów podczas procesu minimalizacji. Szeroka dyskusja nad możliwością takiego "powtórnego" wykorzystania tych danych oraz dogłębna prezentacja opracowanej metody estymacji niepewności wyznaczenia optimum przez algorytm genetyczny, stanowi treść kolejnych rozdziałów.

³ Miejsce to jest nieprzypadkowe. Tutaj aż do swojej przedwczesnej śmierci w 2006 r. pracował i kierował grupą badawczą zajmującą się Wzbudzeniami Kulombowskimi Tomasz Czosnyka – autor programu GOSIA.

3. Wyznaczanie optymalnego zestawu elementów macierzowych przy użyciu algorytmu genetycznego

Ewolucja jest naturalnym procesem optymalizacyjnym, którego celem jest poprawa zdolności przetrwania organizmu (bądź zbioru organizmów) w nieustannie zmieniającym się środowisku. Idea ewolucji stanowi jeden z paradygmatów biologii.

Po raz pierwszy koncepcję dziedziczenia cech nabytych (ang. *acquired traits*) postulował Jean-Baptiste Lamarck (1744 – 1829). Twierdził on, że podczas swojego życia osobniki adaptują się do środowiska poprzez zmianę zestawu cech fizycznych organizmu. Następnie przekazują tak wykształcone cechy potomstwu, które kontynuuje proces przystosowania. Metoda adaptacji miała polegać na wykorzystywaniu cech użytecznych oraz eliminacji tych już niepotrzebnych.

Charles Darwin uważany jest za ojca teorii ewolucji⁴. Sformułował on zarówno teorię doboru naturalnego (ang. *natural selection*) jak i zbiór zasad określających reguły dziedziczenia. Jego teorię można przedstawić następująco: w środowisku z ograniczoną ilością zasobów i stabilnymi populacjami gatunków każdy osobnik konkuruje z innymi o przetrwanie. Osobniki posiadające najlepsze cechy mają największe szanse na przeżycie i przekazanie swoich genów potomstwu. W ten sposób pożądane (lepsze) cechy są dziedziczone przez kolejne pokolenia i z czasem stają się dominujące w całej populacji. Darwin postulował również, że w procesie tworzenia potomka pewne losowe wydarzenia mogą również powodować zmiany w jego cechach. Tak pojmowany proces mutacji wpływa na szanse potomka na przetrwanie.

Ewolucję, jako proces doboru naturalnego dotyczący losowo wybranej populacji osobników, można postrzegać jako przeszukiwanie przestrzeni możliwych wartości, składających się na strukturę rozwiązania. W tym sensie jest ona algorytmem stochastycznego poszukiwania optymalnego osobnika (rozwiązania danego problemu) [69]. W takim procesie należy wyróżnić:

- formę kodowania cech (genotyp osobnika). Dla większości organizmów jest to łańcuch DNA, a w przypadku symulacji komputerowej jest to dowolna forma reprezentacji cech osobnika (np. łańcuchy bitów, liczby rzeczywiste, drzewa);
- zestaw cech rozwiązania wynikający z jego struktury (fenotyp osobnika). W środowisku naturalnym jest to forma fizyczna jaką przybiera osobnik, a w komputerowym rozwiązanie należące do dziedziny funkcji celu;

⁴ Niezależnie od Charlesa Darwina, podobną koncepcję zaproponował Alfred Wallace.

- metodę oceny zdolności osobnika do przetrwania. W świecie rzeczywistym weryfikację stanowi samo życie i możliwość przekazania genów potomstwu. W implementacji maszynowej konieczne jest zdefiniowanie funkcji oceny (miary dostosowania);
- metodę wyboru (inicjalizacji) populacji startowej;
- oddziałujące na populację operatory genetyczne: selekcja, krzyżowanie i mutacja.

Obliczenia ewolucyjne [68] są w istocie formą przeniesienia procesu ewolucji do modelu komputerowego. Kluczowymi elementami są: koncepcja doboru naturalnego, przetrwanie najlepiej przystosowanych osobników oraz reprodukcja. Poszczególne implementacje maszynowe daleko odbiegają od pierwotnej koncepcji naturalnej ewolucji gatunków. Stanowi ona jedynie inspirację, punkt wyjścia oraz źródłosłów dla nazw poszczególnych elementów formułowanych algorytmów.

John Henry Holland jest powszechnie uważany za prekursora obliczeń ewolucyjnych. W literaturze można znaleźć pojedyncze wcześniejsze publikacje na ten temat. Jednak to Holland spopularyzował ideę algorytmów genetycznych jako inspirowaną naturą metodę optymalizacyjną [70], [71]. Dzisiaj uznawana za kanoniczną postać algorytmu genetycznego charakteryzowała się:

- binarną reprezentacją osobników;
- selekcją proporcjonalną;
- krzyżowaniem jednopunktowym;
- mutacją o stałym niskim prawdopodobieństwie.

Taka implementacja algorytmu genetycznego może być uznana za uproszczoną, potencjalnie niedostrojoną do specyfiki problemu.

Przedstawiony w tym rozdziale algorytm genetyczny [76], [77] do analizy danych z pomiarów wzbudzeń kulombowskich został wyposażony w możliwość zarówno reprezentacji binarnej jak i rzeczywistej. Szeroki wachlarz sparametryzowanych operatorów genetycznych oraz możliwość dostosowywania charakterystyki funkcji oceny zależnie od aktualnie analizowanych danych eksperymentalnych stwarzają uniwersalne narzędzie automatyzujące proces poszukiwania w przestrzeni rozwiązań dopuszczalnych optymalnego zestawu elementów macierzowych.

W punkcie 3.1 zdefiniowano problem optymalizacyjny, przedstawiono model problemu oraz przeprowadzono próbę klasycznego rozwiązania. Punkty 3.2 oraz 3.3 zawierają szczegółowy opis zaimplementowanego algorytmu genetycznego. W punkcie 3.4 omówiono program

JACOB – aplikację implementującą algorytm genetyczny i udostępniającą graficzny interfejs użytkownika dla procesu optymalizacji. Punkt 3.5 przedstawia schemat analizy danych z pomiarów wzbudzeń kulombowskich z wykorzystaniem algorytmu genetycznego.

3.1. Analiza problemu

Niniejszy punkt opiera się na informacjach przedstawionych w rozdziale 2., szczególnie w punktach 2.2.2. oraz 2.2.3.

Tradycyjnie stosowaną w programie GOSIA metodą eksploracji przestrzeni rozwiązań dopuszczalnych była metoda gradientowa. Miała ona szereg zalet takich jak: prostota użycia, niska złożoność obliczeniowa, czy małe wymagania sprzętowe. Posiadała jednak również istotne wady: znajdowane rozwiązanie było jedynie lokalnie optymalne, co implikowało konieczność szczegółowego uzasadniania poprawności uzyskanego wyniku od strony fizyki teoretycznej; wąski zakres stosowalności (jedynie do prostych przypadków); duży stopień skomplikowania procesu optymalizacji (proponowanie "dobrego" punktu startowego, wielokrotne powtarzanie procedury optymalizacyjnej).

Zastosowane podejście ewolucyjne zasadniczo różni się od stosowanej metody gradientowej. Główne różnice zestawiono w Tabeli 3.1. Porównanie pokazuje, dlaczego algorytm genetyczny usprawnia proces wyszukiwania optymalnego zestawu elementów macierzowych w oparciu o dane zebrane w eksperymencie fizycznym. Procedura jest bardziej zautomatyzowana. Wykonanie stochastycznego przeglądu całej przestrzeni rozwiązań upraszcza uzasadnianie poprawności uzyskanego wyniku od strony fizyki teoretycznej. Zebrane w procesie optymalizacji próbkowanie przestrzeni rozwiązań może być następnie wykorzystane do analizy kształtu funkcji celu blisko znalezionych minimów w tym minimum globalnego – czyli do estymacji niepewności pomiaru.

	Podejście Ewolucyjne	Metoda Gradientowa
wybór miejsca startu	populacja (zbiór punktów) dla pierwszego	punkt startowy jest proponowany
	pokolenia jest losowana w obrębie całej	przez osobę prowadzącą analizę
	przestrzeni rozwiązań dopuszczalnych (można	danych
	również wykorzystać wiedzę ekspercką)	
informacja przetwarzana	na podstawie położenia całej populacji	na podstawie przetwarzanego punktu
w każdym kroku	rozwiązań oraz wartości ich funkcji oceny	wyznaczony jest wektor gradientu
procesu	tworzone jest kolejne pokolenie	i proponowany kolejny punkt

	Podejście Ewolucyjne	Metoda Gradientowa			
forma przetwarzanej informacji	stochastyczna. Wartości probabilistyczne oparte na wartościach funkcji oceny dla poszczególnych rozwiązań oraz częściowo	deterministyczna. Oparta na wektorze gradientu aktualnie przetwarzanego punktu			
zakres przetwarzanej informacji	w każdym pokoleniu (iteracji procesu) możliwe jest włączenie do następnego pokolenia rozwiązania z dowolnego miejsca przestrzeni	poszukiwanie optymalnego rozwią- zania jest ograniczone do hiper- doliny, w której leży punkt startowy			
zbieżność algorytmu	algorytm zagęszcza próbkowanie przestrzeni rozwiązań w miejscach o lepszych wartościach funkcji oceny. W granicy liczby wywołań automatycznie odnajduje wszystkie minima, w tym globalne	algorytm zatrzymuje się po znalezieniu lokalnego minimum			
charakter przeszukiwania przestrzeni rozwiązań	globalny	lokalny			
metoda przeszukiwania przestrzeni rozwiązań	 elastyczna: zależna od operatorów genetycznych; umożliwia zmianę operatorów oraz ich parametrów w trakcie pracy algorytmu; daje to szansę na dostrajanie algorytmu zależnie od potrzeb 	 stała: w pełni określona; oparta na wektorze gradientu aktualnie przetwarzanego punktu 			

Tabela 3.1 Porównanie podejścia ewolucyjnego i metody gradientowej.

3.1.1. Definicja problemu

Zadanie optymalizacyjne jest sformułowane następująco [81]: wyznaczyć taki wektor elementów macierzowych (punktów w przestrzeni rozwiązań), dla którego wyliczona przy użyciu programu GOSIA wartość χ^2 będzie najmniejsza. Z tego punktu widzenia program GOSIA pełni rolę "czarnej skrzynki" wyznaczającej wartości funkcji, której minimum należy znaleźć.

	wektor elementów	
minimalizacja	macierzowych ME	wyznaczanie wartości $\chi^2(\mathbf{ME})$
z wykorzystaniem		przy użyciu
algorytmu genetycznego	wartość $\chi^2(ME)$	programu GOSIA

Rys. 3.1 Schemat procedury minimalizacyjnej

Funkcja χ^2 jest multimodalna (posiada wiele minimów). Jednocześnie praktyka analizy danych eksperymentalnych wskazuje na brak ostrych krawędzi na jej powierzchni, ale "góry" pomiędzy minimami mogą być bardzo wysokie. Topografia funkcji χ^2 w przestrzeni rozwiązań dopuszczalnych jest pofałdowana. Znalezienie minimum globalnego tej funkcji metodami tradycyjnymi jest trudne. Trudność ta rośnie wraz ze wzrostem złożoności danych eksperymentalnych oraz liczbą wymiarów (elementów macierzowych) przestrzeni rozwiązań.

Stosowane dotychczas podejście wykorzystywało do poszukiwań metodę gradientową. Zadawany jako początkowy zestaw elementów macierzowych musiał być starannie wybierany przez fizyków jądrowych, doświadczonych w analizie danych z pomiarów wzbudzeń kulombowskich. Następnie proces minimalizacyjny musiał być wielokrotnie powtarzany z innych punktów startowych.

Element macierzowy jest liczbą rzeczywistą i zazwyczaj przyjmuje wartości z przedziału [-20,20]. Dokładny zakres zmienności każdego parametru zdefiniowany jest w pliku wejściowym programu GOSIA. Plik ten jest analizowany również przez aplikację implementującą algorytm genetyczny, aby określić dziedzinę funkcji celu.

Pożądana dokładność wartości parametru wynosi typowo 0.01 (do dwóch miejsc po przecinku). Dokładność względna ok. 0.025%. Gdy zakres zmienności parametru zmniejsza się, pożądana bezwzględna dokładność wyznaczanej wartości zazwyczaj rośnie. Wynika ona z jednej strony z dokładności danych eksperymentalnych, które obarczone są niepewnością pomiaru – wyznaczenie wartości elementów macierzowych z większą dokładnością jest przez to niemożliwe. Z drugiej strony elementy macierzowe służą późniejszemu wyznaczeniu parametrów kształtu jądra badanego izotopu. Określenie ich z mniejszą dokładnością implikowałoby małą precyzję ostatecznego wyniku eksperymentu.

Zależnie od eksperymentu, w analizie danych z pomiarów wzbudzeń kulombowskich liczba poszukiwanych elementów macierzowych wynosi od kilku do nawet kilkudziesięciu.

3.1.2. Model problemu

W celu skonstruowania modelu umożliwiającego implementację algorytmu [78] należy określić:

- reprezentację instancji problemu;
- funkcję celu;
- funkcję oceny.

Poniżej przedstawiono szczegóły każdego z wymienionych aspektów algorytmu ewolucyjnego.

3.1.2.a. Reprezentacja instancji problemu

W pierwszej wersji algorytmu genetycznego do analizy danych z pomiarów wzbudzeń kulombowskich, zaprezentowanej w mojej pracy magisterskiej, stosowana była reprezentacja binarna. Na potrzeby rozprawy doktorskiej implementacja została rozszerzona o możliwość stosowania reprezentacji zmiennopozycyjnej (rzeczywistej). W aktualnej wersji programu dostępne są obie reprezentacje, które charakteryzują się specyficznymi operatorami genetycznymi. W celu powtórnego wykorzystania próbkowania algorytmu genetycznego do estymacji niepewności parametrów, zalecane jest stosowanie reprezentacji rzeczywistej. Szczegółowo zostało to omówione w rozdziale 4.

W celu wyeliminowania ryzyka omyłkowego stosowania reprezentacji binarnej zamiast rzeczywistej, wybór pomiędzy reprezentacjami został pozostawiony jedynie na poziomie modyfikacji i rekompilacji kodu źródłowego.

Ponieważ poszukiwanym rozwiązaniem jest wektor elementów macierzowych, niezależnie od metody reprezentacji poszczególnych parametrów (łańcuch bitów albo liczba rzeczywista), cały genotyp stanowi ciąg genów (wektor elementów macierzowych).



Rys. 3.2 Budowa genotypu.

Reprezentacja binarna

W reprezentacji binarnej każdy parametr rozwiązania reprezentowany jest przez ciąg bitów będący rezultatem dyskretyzacji oryginalnej wartości rzeczywistej w oparciu o zakres zmienności parametru oraz liczbę bitów przewidzianych do jego reprezentacji. Zakres zmienności parametru dzielony jest na przedziały, których liczba jest równa liczbie różnych kombinacji -1. $(2^{n_b}, \text{gdzie } n_b$ to liczba bitów przeznaczonych do reprezentacji parametru.)⁵

Szerokość poszczególnych przedziałów (interwałów), na które podzielony jest zakres zmienności parametru wyraża się wzorem [80]:

⁵ W ogólności liczba bitów przeznaczonych do reprezentacji poszczególnych parametrów może być różna. W niniejszym projekcie pożądana dokładność rośnie, gdy zakres zmienności parametru maleje. Z tego powodu przyjęto jednakową liczbę bitów przeznaczonych do reprezentacji poszczególnych parametrów.

$$interval_{i} = \frac{range_{i}}{2^{n_{b}-1}}$$
(3.1)

gdzie:

interval – szerokość przedziału,

range – zakres zmienności parametru,

 n_b – liczba bitów przeznaczonych do reprezentacji parametru,

i – identyfikator parametru.

Na przykład dla parametru a, którego zakres zmienności wynosi [5.0, 8.5], a liczba bitów przeznaczonych do jego kodowania wynosi 3:

*interval*_a =
$$\frac{8.5 - 5.0}{2^3 - 1} = \frac{3.5}{7} = \frac{1}{2}$$

Łańcuchy bitowe kodujące poszczególne wartości parametru, przedstawione są w Tabeli 3.2.

wartość parametru	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5
kodujący ją łańcuch bitów	000	001	010	011	100	101	110	111

Tabela 3.2 Przykład kodowania wartości parametru łańcuchami binarnymi.

Zaimplementowana reprezentacja binarna korzysta z klasycznego kodowania binarnego zaproponowanego w kanonicznej postaci algorytmu genetycznego Johna Hollanda. Nie jest stosowane kodowanie Graya. W rezultacie odległość Hamminga⁶, dla niektórych sąsiednich rzeczywistych wartości parametru, jest wysoka.

W reprezentacji binarnej ważny jest taki dobór liczby bitów do reprezentowania poszczególnych parametrów, aby rozmiar szerokości przedziału był nie większy od pożądanej dokładności wartości parametru. W przeciwnym wypadku będzie ona niewystarczająca.

Zaletą stosowania reprezentacji binarnej jest ograniczenie a priori przestrzeni poszukiwań poprzez taki dobór liczby bitów do reprezentowania poszczególnych parametrów, aby rozmiar interwału był niewiele mniejszy od pożądanej dokładności. Wówczas do reprezentowania parametru przy określonym zakresie i dokładności potrzeba minimum:

$$n_b = \lceil \log_2 \frac{range_i}{accuracy_i} \rceil$$
(3.2)

gdzie *accuracy* jest pożądaną dokładnością i-tego parametru.

⁶ Odległość Hamminga jest to liczba miejsc, na których dwa łańcuchy bitowe się różnią. Dla łańcuchów $0111_2 = 7_{10}$ oraz $1000_2 = 8_{10}$ wynosi ona 4, mimo że w systemie dziesiętnym są to sąsiednie liczby całkowite.

Wadą reprezentacji binarnej jest konieczność przeliczania łańcucha bitów na wartość parametru przy określaniu wartości funkcji oceny. Dwustronna zależność pomiędzy ciągiem bitów $c = (b_{n_b-1}, \dots, b_j, \dots, b_0)$, a rzeczywistą wartością parametru x jest następująca:

$$x_{i} = min_{i} + \frac{range_{i}}{2^{n_{b}}} \cdot \sum_{j=0}^{n_{b}-1} \left(b_{j} \cdot 2^{j} \right)$$
(3.3)

$$c_i = \frac{\left(2^{n_b} - 1\right)\left(x_i - min_i\right)}{range_i} \tag{3.4}$$

gdzie min jest minimalną wartością parametru.

Reprezentacja rzeczywista

Każdy z parametrów ma wówczas reprezentację zmiennopozycyjną (ang. *floating-point number*), czyli zmienną przyjmującą wartości rzeczywiste.

Z punktu widzenia pożądanej dokładności wyniku na poziomie ok. 0.01 nie ma znaczenia czy wykorzystane zostaną zmienne pojedynczej czy podwójnej precyzji (w języku C++ odpowiednio: float oraz double). Ponieważ dla parametrów o wąskim zakresie zmienności pożądana dokładność może być wyższa (zależnie od jakości posiadanych danych eksperymentalnych), dla reprezentacji rzeczywistej zakłada się poszukiwanie rozwiązania co najmniej z precyzją 0.001 (do trzech miejsc po przecinku). Ze względu na związany z tym zwiększony koszt obliczeniowy, w procesie optymalizacji nie stosuje się zawężania dziedziny do pożądanej dokładności. Zamiast tego w procedurze estymacji niepewności stosuje się procedurę MDR, co przedstawione zostało w punkcie 4.4.

Jedynym ograniczeniem nakładanym na poszczególne parametry jest dopuszczalny zakres ich zmienności zdefiniowany w pliku wejściowym do programu GOSIA, tzw. "inpucie do GOSI"⁷ – analizowanym przez aplikację implementującą algorytm genetyczny. Zawarta jest w nim również informacja, które elementy macierzowe mają nie być brane pod uwagę w procesie optymalizacji jako parametry niezależne, lecz pozostawać w sztywnej korelacji⁸ z innym elementem macierzowym. Algorytm genetyczny uwzględnia te informacje w procesie

⁷ Program GOSIA, zależnie od wykorzystywanej funkcjonalności, może przetwarzać dane pochodzące nawet z 10 różnych plików wejściowych. Nazwa "input do GOSI" odnosi się do głównego pliku wejściowego, zawierającego podstawowe informacje oraz opcje wywołania programu. To określenie stosowane jest w całej pracy. Szczegóły przedstawiono w dodatku B.

⁸ Iloraz wartości takich dwóch parametrów ma wartość stałą.
optymalizacji. Przestrzeń rozwiązań dopuszczalnych może być dzięki temu czasowo ograniczona przez fizyka prowadzącego analizę danych eksperymentalnych. Dla użytkowników istotne jest pozostawienie tej funkcjonalności po stronie edytowania "inputu do GOSI".

3.1.2.b. Funkcja celu

Funkcją celu $f_{objective}$ (ang. objective function) w procesie optymalizacji nazywa się wyrażenie, którego globalne optimum chce się wyznaczyć. Dla procesu minimalizacji jest to wektor **a** należący do dziedziny *D* taki, że:

$$\forall \mathbf{x} \in D : f_{objective}(\mathbf{a}) \leq f_{objective}(\mathbf{x})$$
(3.5)

Celem analizy danych z pomiarów wzbudzeń kulombowskich jest znalezienie globalnego minimum, określonej w punkcie 2.2.2.b. funkcji χ^2 . Jak wynika ze wzoru (2.5), jej argumentami są elementy macierzowych przejść elektromagnetycznych między stanami energetycznymi badanego izotopu – pozostałe parametry są stałe dla danego eksperymentu. Należy znaleźć taki zestaw elementów macierzowych, dla którego wartość χ^2 będzie najmniejsza.

Z definicji funkcji χ^2 nie wynika żadne ograniczenie dotyczące wartości elementów macierzowych. Z tego powodu dziedziną problemu jest iloczyn kartezjański zbiorów liczb rzeczywistych. Przeciwdziedzinę stanowi zbiór nieujemnych liczb rzeczywistych. Funkcja χ^2 jest zatem opisana następująco:

$$\chi^2 : \mathbb{R}^m \to \mathbb{R}^+ \cup \{0\}$$
(3.6)

gdzie m jest liczbą elementów macierzowych, których wartości są poszukiwane w analizie danych z eksperymentu.

Dopiero w "inpucie do GOSI", fizyk prowadzący analizę danych eksperymentalnych definiuje zakresy zmienności poszczególnych parametrów oraz może wprowadzić sztywną korelację pomiędzy wybranymi parametrami, tym samym zmniejszając liczbę wymiarów przestrzeni rozwiązań dopuszczalnych.

3.1.2.c. Funkcja oceny

Funkcja oceny $f_{fitness}$ (ang. *fitness function*) w procesie optymalizacji służy do oceniania jakości poszczególnych rozwiązań. Określa ona, w jakim stopniu rozwiązanie spełnia kryteria postawione w zadaniu optymalizacyjnym oraz jak bardzo jest ono przydatne w poszukiwaniu najlepszego rozwiązania.

Najprostszą możliwością jest użycie jako funkcji oceny funkcji celu zadania optymalizacyjnego. Jednak najczęściej przy konstruowaniu funkcji oceny należy uwzględnić również:

- specyfikę przyjętej reprezentacji;
- ograniczenia przestrzeni rozwiązań dopuszczalnych, w tym sposobu zastosowania funkcji kary (ang. *penalty function*) dla rozwiązań naruszających ograniczenia dziedziny problemu;
- dla problemów, w których funkcja celu daje jedynie odpowiedź czy rozwiązanie jest dopuszczalne, funkcja oceny powinna określać stopień spełnienia kryteriów dopuszczalności (klasycznym przykładem jest tu problem spełnialności formuł logicznych, ang. *boolean satisfability*, SAT);
- możliwość zmiany charakterystyki funkcji celu (np. przeskalowanie, transformacja).

W ogólnym przypadku funkcja oceny ma postać:

$$f_{fitness} = \mathbf{Y} \circ \Psi \circ \Phi$$
$$f_{fitness} \colon P \to D \to \mathbb{R} \to \mathbb{R}^+$$
(3.7)

gdzie:

Y – funkcja adaptująca wartości funkcji celu z uwzględnieniem specyfiki procesu optymalizacji (np. przeskalowanie na wartości dodatnie, wprowadzenie kary za naruszenie ograniczeń, transformacja wartości funkcji celu, aby poprawić zbieżność),

 Ψ – funkcja celu,

 Φ – funkcja przekształcająca (dekodująca) postać osobnika przedstawioną zgodnie z przyjętą reprezentacją instancji problemu (genotypu) na wektor parametrów (fenotyp), dla którego wyznaczona zostanie wartość funkcji celu,

złożenie funkcji,

P – n-wymiarowa przestrzeń wartości genotypowych, przestrzeń poszukiwań zgodna z przyjętą reprezentacją instancji problemu,

D – m-wymiarowa przestrzeń wartości fenotypowych, przestrzeń poszukiwań zgodna z dziedziną funkcji celu.

W opisywanej implementacji algorytmu genetycznego poszczególne komponenty funkcji oceny zostały skonstruowane następująco:

 funkcja dekodująca Φ dla reprezentacji binarnej dokonuje najpierw przekształcenia genotypu z postaci wektora łańcuchów bitowych na wektor liczb rzeczywistych. Następnie (dla obu reprezentacji) wektor parametrów jest uzupełniany o wartości tych parametrów, które zostały wyłączone z procesu optymalizacji i zdefiniowane jako pozostające w ścisłej korelacji z innymi parametrami. W ten sposób n-elementowy genotyp jest rozszerzany do m-elementowego fenotypu;

• wartości funkcji celu Ψ są określane przy pomocy programu GOSIA. Jest on wywoływany jako oddzielny proces. Fenotyp, będący wektorem argumentów (zestawem elementów macierzowych), dla którego ma być wyznaczona wartość χ^2 , jest umieszczany w pliku pomocniczym. Wartość χ^2 (**ME**) jest pobierana z pliku wyjściowego programu GOSIA (tzw. "outputu GOSI"⁹).

W celu przyspieszenia procesu przeszukiwania przestrzeni rozwiązań dopuszczalnych, dla wyznaczania wartości $\chi^2(ME)$ mogą być stosowane aproksymacje opisane w punkcie 2.2.3.e. Ich użycie jest określane w "inpucie do GOSI" i jest stałe dla jednego przebiegu algorytmu genetycznego. Procedura wyznaczania wartości funkcji celu jest traktowana jako "czarna skrzynka", dlatego niezależnie od faktu użycia metod "szybkiej aproksymacji", funkcja Ψ jest traktowana przez implementację algorytmu genetycznego jako funkcja celu. Późniejsza interpretacja wyniku procesu optymalizacji jest pozostawiona fizykowi prowadzącemu całość analizy danych eksperymentalnych. Implementacja algorytmu genetycznego przewiduje również możliwość stosowania innych funkcji jako funkcji celu, np. funkcji F7 Schwefela [98]. Zostało to szerzej

opisane w dalszej części niniejszej rozprawy;

 funkcja adaptująca Y przewiduje możliwość stosowania transformacji wartości funkcji celu (np. logarytmicznej). Wpływa to na jej charakterystykę – zmniejsza wysokość "szczytów" pomiędzy poszczególnymi hiper-dolinami powierzchni funkcji celu i tym samym poprawia zbieżność algorytmu genetycznego. Użyteczność takiej transformacji jest zależna od danych z aktualnie analizowanego eksperymentu. Zalecanym rozwiązaniem jest stosowanie transformacji logarytmicznej. Zmiana możliwa jest jedynie przez modyfikację i rekompilację kodu źródłowego.

Podsumowując, stosowana w procesie optymalizacyjnym funkcja oceny wyznacza wektor wszystkich elementów macierzowych (fenotyp) w oparciu o wektor parametrów niezależnych (genotyp) oraz uwzględniając stosowaną reprezentację osobniczą. Wartość χ^2 (**ME**) wyznaczana jest przy użyciu programu GOSIA, a na ostateczną wartość funkcji oceny może mieć wpływ zastosowanie transformacji wartości funkcji celu (np. logarytmicznej).

⁹ Program GOSIA, zależnie od wykorzystywanej funkcjonalności, może zapisywać dane nawet do 8 różnych plików wyjściowych. Nazwa "output GOSI" odnosi się do głównego pliku wyjściowego, w którym umieszczana jest wartość $\chi^2(ME)$. To określenie stosowane jest w całej pracy. Szczegóły przedstawiono w dodatku B.

3.1.3. Próba klasycznego rozwiązania

W opisywanym problemie optymalizacyjnym nieznana jest a priori charakterystyka funkcji celu. Doświadczenie płynące z analizy danych z pomiarów wzbudzeń kulombowskich wskazuje, że jest ona multimodalna. Nie ma żadnych dodatkowych informacji umożliwiających zawężenie przestrzeni poszukiwań i tym samym przyspieszenie procesu znajdowania rozwiązania optymalnego, poza określonymi w "inpucie do GOSI" zakresami zmienności poszczególnych parametrów oraz pożądaną dokładnością wyniku.

Z tego powodu klasyczne rozwiązanie zadania sprowadza się do sprawdzenia wszystkich rozwiązań dopuszczalnych i wybraniu najlepszego. Tylko w ten sposób można zyskać pewność, że lepsze rozwiązanie nie istnieje. Taka metoda nazywana jest przeszukiwaniem wyczerpującym (ang. *exhaustive search*) lub wyliczeniowym (ang. *enumerative search*).

Poniżej przedstawiono estymację czasu potrzebnego do przeprowadzenia powyższej procedury na przykładzie reprezentacji binarnej. Załóżmy, że liczba elementów macierzowych, dla których poszukiwane są optymalne wartości wynosi m=8. Jednocześnie wszystkie są parametrami niezależnymi (żaden nie jest ściśle skorelowany z innymi, czyli n=m). Zakres zmienności dla każdego z nich wynosi [-20,20], a pożądana dokładność 0.01. Przyjmijmy, że czas potrzebny do wyznaczenia wartości χ^2 , dla jednego wektora parametrów wynosi: $t_{eval}=0.1s^{10}$.

Zgodnie ze wzorem (3.2) liczba bitów potrzebna do reprezentacji każdego z parametrów to:

$$n_b = \lceil \log_2 \frac{20 - (-20)}{0.01} \rceil = \lceil \log_2 4000 \rceil \approx 12$$

Liczba punktów przestrzeni rozwiązań dopuszczalnych P, wynosi:

card
$$(P) = 2^{n_b \cdot n} = 2^{12 \cdot 8} = 2^{96}$$

Zatem dokładne przeszukanie przestrzeni rozwiązań dopuszczalnych zajęłoby:

$$T = t_{eval} \cdot card(P) = 0.1s \cdot 2^{96} \approx 2.51 \cdot 10^{20} law$$

Dla porównania czas, który upłynął od Wielkiego Wybuchu to ok. $13.82 \cdot 10^9 lat$ [84].

Dokładne przeszukanie przestrzeni rozwiązań dopuszczalnych, dla problemu analizy danych z pomiarów wzbudzeń kulombowskich, jest niemożliwe. Z tego powodu konieczne jest zastosowanie algorytmu heurystycznego, który przeszukując fragment przestrzeni znajdzie rozwiązanie dostatecznie dobre – (sub)optymalne.

¹⁰ Oszacowano empirycznie.

3.2. Budowa algorytmu genetycznego

Algorytmy genetyczne należą do grupy heurystycznych¹¹. Heurystyczna metoda przeszukiwania przestrzeni rozwiązań dopuszczalnych polega na takim prowadzeniu procesu wnioskowania, aby z dotychczasowo odkrytej wiedzy oraz znajomości faktów i reguł obowiązujących w przestrzeni poszukiwań, wyprowadzać nowe fakty i powiększać wiedzę o zagadnieniu. Poszukiwania te prowadzi się najczęściej na odpowiednio dobranym modelu zjawiska.

Poszukiwania heurystyczne podejmuje się, gdy znalezienie rozwiązania dokładnego nie jest możliwe – z powodu zbyt długiego czasu obliczeń, zmian zachodzących w środowisku bądź niejednoznaczności funkcji oceny. Należy wtedy określić praktyczną, opartą na doświadczeniu, inteligentną metodę postępowania, która umożliwi w akceptowalnym czasie znaleźć dostatecznie dobre (przybliżone) rozwiązanie problemu.

Są dwa sposoby takiego uproszczenia problemu optymalizacyjnego, żeby możliwe było znalezienie jego rozwiązania. Pierwszy sposób zakłada uproszczenie (w szczególności zawężenie) modelu problemu, tak by możliwe było otrzymanie rozwiązania dokładnego modelu przybliżonego:

Problem → Model _{przybliżony} → Rozwiązanie _{dokładne} (Model _{przybliżony}) Drugi sposób znajduje przybliżone rozwiązanie dla dokładnego modelu:

Problem \rightarrow Model _{dokładny} \rightarrow Rozwiązanie _{przybliżone} (Model _{dokładny})

Drugi z tych sposobów jest zazwyczaj lepszy. Lepiej jest znaleźć rozwiązanie przybliżone modelu dokładnego niż odwrotnie. Takie podejście zostało przyjęte do rozwiązania problemu wyznaczania (sub)optymalnego zestawu elementów macierzowych w ramach opisywanej implementacji algorytmu genetycznego.

3.2.1. Struktura algorytmu

Podejście ewolucyjne zakłada równoległy start z wielu punktów przestrzeni poszukiwań. Zbiór rozwiązań jednocześnie przetwarzanych nazywany jest populacją. W kolejnych krokach algorytmu, tworzone są następne pokolenia populacji [85]. Do tworzenia poszczególnych osobników w pokoleniu używana jest informacja ze wszystkich rozwiązań z poprzedniego pokolenia. Pozwala to skutecznie unikać przedwczesnej zbieżności algorytmu – problemu lokalnych ekstremów.

¹¹ gr. *heuriskō* ($\dot{\eta}$ ουρισκω) – znajduję; *heuresis* ($\dot{\eta}$ ουρησις) – odkryć, znaleźć; *heurēka* ($\dot{\eta}$ ουρηκα) – znalazłem (słynny okrzyk Archimedesa). Heurystyka jest sztuką wykrywania nowych faktów oraz znajdowania związków występujących pomiędzy nimi, prowadzącą do odkrywania nowych prawd i stawiania hipotez. W informatyce heurystyką określa się metodę znajdowania rozwiązań przybliżonych, szczególnie gdy nie ma gwarancji znalezienia rozwiązania optymalnego lub nawet prawidłowego.

Umiejętne skonstruowanie algorytmu ewolucyjnego polega na odpowiednim opracowaniu operatorów służących do stworzenia populacji potomnej. Celem jest szybkie (przy sprawdzeniu możliwie małej liczby rozwiązań), ale i maksymalnie pewne, znalezienie globalnego optimum.

Schemat działania algorytmu genetycznego jest następujący:

```
    Wylosuj populację początkową.
    Oceń osobniki w populacji.
    Dopóki niespełnione jest kryterium stopu:

            wybierz nową populację w oparciu o aktualną (selekcja i krzyżowanie);
            zmień nową populację (mutacja);
            oceń osobniki w nowej populacji.
```

Rys. 3.3 Pseudokod algorytmu genetycznego

W terminologii przyjętej dla algorytmów genetycznych, każde proponowane rozwiązanie (punkt przestrzeni rozwiązań) nazywane jest osobnikiem. Działanie algorytmu rozpoczyna się od wygenerowania losowego zbioru osobników, nazywanego populacją początkową.

W każdej kolejnej iteracji (pokoleniu) tworzona jest, w oparciu o poprzednią, nowa populacja osobników. Zakłada się przy tym, że:

- osobniki lepiej ocenione mają większy wkład w tworzeniu potomków;
- potomek tworzony jest przez skrzyżowanie dwóch osobników z aktualnej populacji (rodziców);
- dobrze oceniony rodzic może (ale nie musi) zostać przeniesiony do następnej populacji.

Krzyżowanie polega na wymianie informacji pomiędzy dwoma osobnikami. Mutacja polega na losowej zmianie, z określonym prawdopodobieństwem, fragmentu genotypu osobnika. Wybór osobników na rodziców ma charakter losowy.

Skuteczność algorytmu genetycznego zależy od właściwego doboru operatorów genetycznych, a konkretnie metody selekcji, krzyżowania i mutacji. Jest to zadanie trudne i musi uwzględnić specyfikę problemu. Często niemożliwe jest skonstruowanie takich operatorów genetycznych, żeby działanie algorytmu dawało pożądany rezultat.

W poniższych punktach przedstawiono szczegółowy opis zastosowanych rozwiązań w omawianej implementacji algorytmu genetycznego.

3.2.2. Populacja początkowa

Algorytmy ewolucyjne są stochastyczną, opartą na populacji rozwiązań, metodą przeszukiwania przestrzeni. W każdym kroku algorytm przetwarza zbiór (populację) rozwiązań. Pierwszym zagadnieniem jest metoda tworzenia początkowego zbioru punktów z przestrzeni poszukiwań.

Należy wyróżnić trzy podstawowe metody generowania populacji początkowej:

- próbkowanie na siatce (ang. sampling on grid) punkty są umieszczane na siatce o z góry zdefiniowanej gęstości. Dzięki temu każdy fragment przestrzeni rozwiązań dopuszczalnych jest reprezentowany w populacji początkowej próbkowanie jest równomierne. Niestety, rozmiar populacji rośnie wykładniczo w stosunku do liczby punktów podziału poszczególnych wymiarów. Dla większej liczby wymiarów, aby uzyskać rozsądną gęstość próbkowania, trzeba byłoby przyjąć niepraktycznie dużą wielkość populacji początkowej. Z tego powodu próbkowanie na siatce jest rzadko stosowane. Może natomiast znaleźć zastosowanie przy estymacji niepewności parametru poprzez analizę kształtu funkcji celu w bezpośrednim otoczeniu optimum. Omówiono to w kolejnych rozdziałach;
- próbkowanie wokół określonego punktu przestrzeni współrzędne punktów są losowane z określonym rozkładem wokół zadanego miejsca przestrzeni rozwiązań dopuszczalnych. Zazwyczaj stosuje się rozkład Gaussa, dla którego za wartość oczekiwaną µ, dla poszczególnych wymiarów przyjmuje się współrzędne punktu "startowego", a wariancję traktuje się jako parametr umożliwiający sterowanie zagęszczeniem próbkowania populacji początkowej wokół tego punktu. Dzięki temu możliwe jest wygenerowanie populacji początkowej skupionej we fragmencie przestrzeni rozwiązań dopuszczalnych, o którym skądinąd wiadomo (lub przypuszcza się), że znajduje się w nim rozwiązanie optymalne;
- próbkowanie losowe z rozkładem równomiernym współrzędne punktów są losowane z rozkładem równomiernym dla całych zakresów zmienności poszczególnych parametrów. Dzięki temu w granicy liczby punktów próbkowanie jest równomierne dla całej przestrzeni rozwiązań dopuszczalnych. Jednocześnie niezależnie od przyjętej wielkości populacji początkowej, w granicy liczby startów algorytmu genetycznego, próbkowanie początkowe jest również równomierne. Takie rozwiązanie jest kompromisem pomiędzy ograniczonym złożonością obliczeniową rozmiarem populacji początkowej, a potrzebą reprezentowania całej przestrzeni rozwiązań dopuszczalnych.

W omawianej implementacji algorytmu genetycznego rozważono zastosowanie wszystkich trzech przedstawionych metod generowania populacji początkowej. Ze względu na dużą liczbę wymiarów (elementów macierzowych) w typowych analizach danych z pomiarów wzbudzeń kulombowskich, zrezygnowano z implementacji próbkowania na siatce.

Ponieważ algorytm genetyczny stanowi część szerszego zbioru narzędzi analitycznych pomiarów wzbudzeń kulombowskich, jako podstawową metodę generowania populacji początkowej zalecane jest próbkowanie losowe z rozkładem równomiernym. Przy takim podejściu możliwa jest szeroka eksploracja całej przestrzeni rozwiązań dopuszczalnych. Jest to ten element, którego brakowało w stosowanej uprzednio metodzie gradientowej zaimplementowanej w programie GOSIA.

W omawianej implementacji algorytmu genetycznego można również utworzyć populację początkową, próbkując przestrzeń rozwiązań dopuszczalnych wokół zadanego punktu. W tym przypadku jako punkt "startowy" przyjmowany jest, określany w "inpucie do GOSI", punkt startowy optymalizacji gradientowej. Dzięki takiemu rozwiązaniu, prowadzący analizę danych eksperymentalnych fizyk może wykorzystać wiedzę ekspercką o analizowanym przypadku fizycznym i rozpocząć optymalizację wokół zdefiniowanego punktu przestrzeni. Wariancja rozkładu próbkowania dla poszczególnych wymiarów jest parametrem określanym w programie implementującym algorytm genetyczny, znormalizowanym do zakresu zmienności danego elementu macierzowego.

Rozmiar populacji początkowej ma wpływ na złożoność obliczeniową oraz zdolność algorytmu do eksploracji przestrzeni poszukiwań. Duża liczba osobników w pokoleniu zwiększa różnorodność materiału genetycznego i przez to poprawia zdolność algorytmu do eksploracji przestrzeni poszukiwań. Oznacza to jednak również większą ilość obliczeń w każdej iteracji, co z kolei może skutkować potrzebą mniejszej liczby pokoleń do znalezienia rozwiązania optymalnego.

Mały rozmiar populacji powoduje reprezentowanie jedynie fragmentu przestrzeni poszukiwań. Skutkuje to możliwością szybkiego przetwarzania pojedynczego pokolenia, ale z drugiej strony zapewne zwiększy liczbę pokoleń do osiągnięcia zbieżności. W przypadku małego rozmiaru populacji zdolność do eksploracji przestrzeni może być poprawiona poprzez zwiększenie współczynnika mutacji. Typowa liczba osobników w populacji dla analizy danych z pomiarów wzbudzeń kulombowskich zawiera się pomiędzy 50 a 500 rozwiązań.

3.2.3. Operatory selekcji

Selekcja ma na celu wyeliminowanie słabszych osobników z populacji. Zgodnie z koncepcją Darwina, proces ten nie jest stricte deterministyczny. Jednocześnie prawdopodobieństwo przeżycia osobnika powinno być zależne pozytywnie od wartości funkcji przystosowania, tj. funkcji oceny. Osobniki, które przeżyją, biorą udział w reprodukcji. Mogą być bezpośrednio przeniesione do następnego pokolenia oraz są potencjalnymi rodzicami dla tworzonych potomków.

Z operatorami selekcji wiąże się pojęcie presji selekcyjnej (ang. *selective pressure*), czyli czasu, mierzonego liczbą pokoleń, potrzebnego by najlepszy osobnik zdominował całą populację przy założeniu, że operator mutacji nie działa. Operatory z wysoką presją selekcyjną zmniejszają zróżnicowanie materiału genetycznego gwałtowniej niż te z niską. Może to prowadzić do przedwczesnej zbieżności algorytmu i do utknięcia w lokalnym optimum. Wysoka presja selekcyjna ogranicza zdolność algorytmu do eksploracji przestrzeni poszukiwań, jednak zbyt niska może skutkować ślepym błądzeniem algorytmu, niepotrzebnym wzrostem złożoności obliczeniowej oraz w skrajnym przypadku całkowitą niemożnością znalezienia optimum.

Operator z najwyższą presją selekcyjną eliminuje z populacji wszystkie osobniki poza najlepszym. Następstwem tego jest utworzenie w następnym pokoleniu populacji składającej się jedynie z jego kopii. Operatorem z najniższą presją selekcyjną jest wybór losowy, dla którego każdy osobnik (niezależnie od wartości funkcji oceny) ma taką samą szansę na przekazanie materiału genetycznego do następnego pokolenia. Prawdopodobieństwo przeżycia osobnika jest w tym przypadku całkowicie niezależne od wartości funkcji przystosowania.

W literaturze opisano bardzo wiele operatorów selekcji. W przedstawionym algorytmie genetycznym zaimplementowano możliwość wyboru pomiędzy:

- selekcją elitarną (ang. *truncation selection*);
- selekcją ruletkową (ang. roulette selection);
- selekcją turniejową (ang. tournament selection).

W każdym przypadku osobniki, które przeżyją selekcję, są przenoszone do populacji tymczasowej (oraz pośrednio do następnego pokolenia). W ramach operatora krzyżowania, wybierani są spośród nich rodzice, których materiał genetyczny posłuży do uzupełnienia populacji tymczasowej o osobniki potomne, aby utrzymać jej stały rozmiar. Poniżej przedstawiono szczegóły poszczególnych operatorów selekcji.

3.2.3.a. Operator selekcji elitarnej

Selekcja elitarna zakłada, że osobniki, z wartością funkcji oceny niższą niż progowa, są eliminowane. Wartość progowa, nazywana też poziomem odcięcia (ang. *truncation level*) może być określana w oparciu o średnią wartość funkcji przystosowania dla każdego pokolenia lub o wartość funkcji przystosowania najlepszego osobnika w pokoleniu. Alternatywnie, próg odcięcia może być wyznaczany w oparciu o procent (lub liczbę) osobników, które mają być wyeliminowane.

W przedstawionym algorytmie genetycznym próg odcięcia wyznaczany jest w oparciu o określony przez użytkownika procent osobników, który ma być wyeliminowany. Parametr ten może być modyfikowany podczas procesu optymalizacji.

Przykładowo jeśli rozmiar populacji wynosi 200 osobników, a próg odcięcia określono na 60% to:

- 80 najlepszych osobników z aktualnego pokolenia zostanie bezpośrednio przeniesionych do następnego pokolenia;
- 120 osobników zostanie utworzonych w następnym pokoleniu w oparciu o materiał genetyczny przeniesionych osobników.

3.2.3.b. Operator selekcji ruletkowej

Selekcja ruletkowa zapewnia wybór rozwiązań do populacji tymczasowej proporcjonalnie do wartości ich funkcji przystosowania. Zależność ta może być poddana transformacji (np. logarytmicznej). Umożliwia to zmniejszenie różnic w wartości funkcji oceny u poszczególnych osobników.

Następnie tworzy się "koło ruletki", w którym każdy osobnik reprezentowany jest przez wycinek kołowy odpowiadający jego przekształconej z użyciem wybranej transformacji wartości funkcji oceny. Im słabszy osobnik tym większy kąt – większa szansa bycia wyeliminowanym. Tak zdefiniowane "koło ruletki" jest tożsame z dystrybuantą prawdopodobieństwa wyeliminowania.

Określana jest liczba osobników, które mają być wyeliminowane z populacji. Podobnie jak w przypadku selekcji elitarnej, wyznaczana jest ona w oparciu o określony przez użytkownika, procent osobników, które mają być wyeliminowane.

Selekcja odbywa się poprzez wirtualne "kręcenie kołem ruletki". Wskazany osobnik jest eliminowany. Selekcja ruletkowa wymaga określenia:

- procentu osobników do wyeliminowania;
- odwzorowania proporcjonalności do wyznaczenia dystrybuanty rodzaju transformacji wartości funkcji oceny. W praktyce w analizie danych z pomiarów wzbudzeń kulombowskich stosowana jest transformacja logarytmiczna (zmniejszająca różnice w wartości funkcji oceny), bądź liniowa (brak modyfikacji).

W selekcji ruletkowej presja selekcyjna jest wysoka, ale niższa niż w selekcji elitarnej.

3.2.3.c. Operator selekcji turniejowej

W selekcji turniejowej populacja jest losowo dzielona na równe grupy osobników (turnieje), o ustalonej przez użytkownika liczebności. Każdy osobnik przypisany jest do dokładnie jednego turnieju. Liczba osobników w ostatnim turnieju może być mniejsza, gdy rozmiar turnieju nie jest dzielnikiem wielkości populacji. W każdym turnieju eliminowane są wszystkie osobniki poza najlepszym. Zwycięzcy turniejów, czyli osobniki o najlepszej wartości funkcji przystosowania, są przenoszone do następnego pokolenia.

Selekcja turniejowa stwarza większą szansę na przeżycie osobnikom słabszym, gdyż wybór uczestników do turnieju jest losowy i dopiero wewnątrz turnieju premiowane są osobniki lepiej przystosowane. Losowe przypisywanie osobników do poszczególnych turniejów powoduje, że selekcja turniejowa ma niższą presję selekcyjną niż selekcja proporcjonalna (ruletkowa).

Presja selekcyjna selekcji turniejowej jest pozytywnie zależna od rozmiaru turnieju. Mała liczba osobników w turnieju stwarza większą szansę, że osobnik o gorszej wartości funkcji przystosowania wygra turniej i zostanie przeniesiony do następnego pokolenia. Duża liczba osobników w turnieju premiuje osobniki lepsze, co zwiększa presję selekcyjną.

Liczbę osobników, które przeżyją selekcję i zostaną przeniesione do następnego pokolenia, określa zależność:

$$nSurvivors = \left[\frac{populationSize}{tournamentSize}\right]$$
(3.8)

gdzie:

nSurvivors – liczba osobników, które przeżyją selekcję,

populationSize - rozmiar populacji,

tournamentSize – rozmiar turnieju.

Jest ona pośrednio definiowana przez użytkownika, gdyż zależy od rozmiaru turnieju, określanego przez użytkownika.

3.2.4. Operatory krzyżowania

Operatory krzyżowania wraz z operatorami mutacji składają się na proces reprodukcji, czyli tworzenie nowego pokolenia. Krzyżowanie polega na utworzeniu nowych osobników na podstawie informacji zebranej w genotypach osobników aktualnej populacji. Trwa ono do osiągnięcia założonego rozmiaru populacji.

Operatory krzyżowania można podzielić ze względu na liczbę rodziców potrzebnych do utworzenia potomka. Wyróżnia się operatory stosujące jednego, dwóch albo wielu rodziców. W omawianej implementacji stosowane są operatory wykorzystujące dwóch rodziców.

Drugi podział operatorów krzyżowania związany jest z przyjętą reprezentacją instancji problemu. Są one dostosowane albo do reprezentacji binarnej albo rzeczywistej. W niniejszej implementacji zastosowano jeden operator krzyżowania dla reprezentacji binarnej oraz dwa dla reprezentacji rzeczywistej:

- reprezentacja binarna: krzyżowanie jednopunktowe (ang. one-point crossover);
- reprezentacja rzeczywista: krzyżowanie heurystyczne-2 (ang. heuristic crossover2) [73];
- reprezentacja rzeczywista: krzyżowanie ziarniste (ang. discrete crossover) [73].

Przyjęto, że osobniki które przeżyły selekcję, są bezpośrednio przenoszone do następnego pokolenia, tym samym zmniejszając liczbę osobników do wyprodukowania w ramach krzyżowania. Spośród nich losowani są rodzice dla tworzonego potomstwa, które ma uzupełnić nowe pokolenie.

Wybór rodziców jest oparty na przedstawionej wcześniej metodzie "koła ruletki". Prawdopodobieństwo wybrania osobnika na rodzica jest proporcjonalne do wartości jego funkcji przystosowania. Zależność ta może być poddawana transformacji (np. logarytmicznej). Wpływa to na zmniejszenie różnic w wartości funkcji oceny pomiędzy osobnikami i w konsekwencji na prawdopodobieństwo wybrania na rodzica. W tworzonym "kole ruletki" każdy kandydat na rodzica jest reprezentowany przez wycinek kołowy odpowiadający jego przekształconej z użyciem wybranej transformacji wartości funkcji oceny. Samo odwzorowanie jest jednak odwrotne niż w selekcji ruletkowej. W ruletkowym wyborze na rodzica, im słabszy osobnik tym mniejszy kąt wycinka kołowego, czyli mniejsza szansa zostania rodzicem. Tak zdefiniowane "koło ruletki" jest tożsame z dystrybuantą prawdopodobieństwa zostania rodzicem.

W celu uniknięcia sytuacji, w której osobnik ze względu na wysoką wartość funkcji oceny będzie zbyt często wybierany na rodzica i tym samym zdominuje proces reprodukcji, zaproponowano mechanizm "zapobiegania kazirodztwu".

Po pierwsze, gdy osobnik zostaje wybrany na pierwszego rodzica, wybór drugiego odbywa się bez jego udziału – "koło ruletki" jest tworzone z wyłączeniem jego wartości funkcji oceny. Zapobiega to sytuacji, w której rodzic tworzy potomka sam ze sobą i w konsekwencji jest on kopią rodzica.

Po drugie, w jednym pokoleniu osobnik może być wybrany na rodzica jedynie określoną liczbę razy. Użytkownik określa maksymalną liczbę dzieci na rodzica – maksymalną liczbę krzyżowań, do których osobnik może być wybrany w danym pokoleniu. Istotne jest przy tym, aby to ograniczenie nie uniemożliwiało utworzenia następnego pokolenia.

$$2 \cdot \frac{nCasualties}{nSurvivors} \leq maxChildren$$

$$nCasualties + nSurvivors = populationSize$$
(3.9)

gdzie:

maxChildren – maksymalna liczba dzieci na rodzica,

nCasualties – liczba osobników wyeliminowanych przez selekcję (potomków do utworzenia).

Ograniczenie dolne na wartość parametru *maxChildren* wynika z proporcji liczby "wolnych miejsc" w nowym pokoleniu (liczby potomków do utworzenia) do liczby potencjalnych rodziców. Ponieważ do utworzenia potomka potrzebnych jest dwóch rodziców, iloraz ten trzeba podwoić.

Przyjęcie wartości *maxChildren* blisko dolnego ograniczenia skutkuje podobną liczbą potomstwa dla wszystkich osobników, którzy przeżyli selekcję. Aby lepsze osobniki miały statystycznie więcej potomków niż słabsze, należy przyjmować wartość *maxChildren* odpowiednio powyżej dolnego ograniczenia. Z drugiej strony jeśli wartość *maxChildren* będzie zbliżona do liczby wolnych miejsc w tworzonym pokoleniu, to nie będzie miała istotnego wpływu na "zapobieganie kazirodztwu". Osobnik z wysoką wartością funkcji oceny będzie mógł zostać wybrany na rodzica nawet wszystkich potomków – jego geny zdominują następne pokolenie. Aby tego uniknąć, wartość *maxChildren* powinna być ustawiona odpowiednio poniżej liczby wolnych miejsc w następnym pokoleniu. Podsumowując:

$$2 \cdot \frac{nCasualties}{nSurvivors} < maxChildren < nCasualties$$
(3.10)

Poniżej przestawiono szczegóły poszczególnych operatorów krzyżowania.

3.2.4.a. Operator krzyżowania jednopunktowego (reprezentacja binarna)

Krzyżowanie jednopunktowe (ang. *one-point crossover*) dla reprezentacji binarnej jest szczególnym przypadkiem krzyżowania wielopunktowego. Po wybraniu rodziców losowane jest miejsce cięcia genotypu. Konkatenacja fragmentu genotypu pierwszego i drugiego rodzica daje genotyp potomka.

W krzyżowaniu wielopunktowym często stosowana jest maska – losowy łańcuch binarny o długości genotypu. Wskazuje ona, które elementy genotypu potomek ma otrzymać od którego rodzica. Genotyp potomka jest sumą iloczynu maski z genotypem pierwszego rodzica oraz iloczynu przeciw-maski z genotypem drugiego rodzica.

Alternatywnie, jeśli wynikiem krzyżowania wielopunktowego dwóch rodziców miałyby być dwa różne osobniki potomne, można utworzyć po jednym klonie każdego z rodziców, a następnie wymienić między nimi te bity, dla których wartość odpowiadających im bitów maski wynosi jeden (taki operator jest symetryczny).

W opisywanym operatorze krzyżowania jednopunktowego wystarczy wylosować jedno miejsce cięcia, a następnie połączyć odpowiednią część genotypu pierwszego i drugiego rodzica. Miejsce cięcia jest liczbą całkowitą z przedziału $[0, n_b]$. Jeśli wylosowana zostanie któraś z wartości brzegowych, to potomek będzie klonem jednego z rodziców:

> if (cutPlace == 0) offspring = parent₁; if (cutPlace == n_b) offspring = parent₂;

gdzie:

cutPlace – miejsce cięcia, *parent* – rodzic, *offspring* – potomek.

W takim operatorze krzyżowania istotna jest kolejność rodziców – nie jest on symetryczny. To znaczy:

$$parent_1 \otimes_{binary} parent_2 \neq parent_2 \otimes_{binary} parent_1$$
 (3.11)

gdzie \otimes_{binary} jest opisywanym operatorem krzyżowania dla reprezentacji binarnej.

W istocie potomek dziedziczy bardziej znaczące bity po pierwszym rodzicu, a bity mniej znaczące po drugim rodzicu. Liczbę bitów dziedziczonych po każdym z rodziców określa miejsce cięcia. Rys. 3.4 przedstawia przykład działania opisanego operatora krzyżowania jednopunktowego:

krzyżowanie jednopunktowe wylosowane miejsce rozcięcia: 5

Rodzic 1: Rodzic 2: 00101100 11110001 10110110 10111001 1001010100101101 krzyżowanie 00101 | 1001100100101010101010101 krzyżowanie 00101 | 0011001010100101101

Potomek: 00101001 10010101 00101101

Rys. 3.4 Schemat operatora krzyżowania jednopunktowego

3.2.4.b. Operator krzyżowania heurystycznego-2 (reprezentacja rzeczywista)

Operator krzyżowania heurystycznego-2 (ang. *heuristic crossover2*) został opracowany w celu przeszukiwania przestrzeni rozwiązań dopuszczalnych w obiecujących kierunkach. Nie jest w nim istotna kolejność rodziców – operator jest symetryczny. Operator sprawdza, który z rodziców ma lepszą wartość funkcji oceny. Osobnik potomny proponowany jest na linii łączącej gorszego i lepszego rodzica w stronę potencjalnie lepszych wartości funkcji oceny. Przy czym:

$$dist(offspring, parent_{better}) \le dist(parent_1, parent_2)$$

$$(3.12)$$

gdzie $dist(\cdot)$ zwraca odległość euklidesową w przestrzeni rozwiązań dopuszczalnych.

Pseudokod działania opisywanego operatora przedstawia:

```
foreach dimension
if (f<sub>fitness</sub>(parent<sub>1</sub>) >= f<sub>fitness</sub>(parent<sub>2</sub>))
x_i(offspring) = \lambda_i(x_i(parent<sub>2</sub>) -x_i(parent<sub>1</sub>)) + x_i(parent<sub>2</sub>);
else x_i(offspring) = \lambda_i(x_i(parent<sub>1</sub>) -x_i(parent<sub>2</sub>)) + x_i(parent<sub>1</sub>);
while (x_i(offspring) < min<sub>i</sub>) x_i(offspring) += range<sub>i</sub>;
while (x_i(offspring) > max<sub>i</sub>) x_i(offspring) -= range<sub>i</sub>;
```

Rys. 3.5 Pseudokod operatora krzyżowania heurystycznego-2 gdzie:

- λ liczba rzeczywista z przedziału [0,1] losowana z rozkładem równomiernym dla każdego z parametrów,
- $x(\cdot)$ zwraca wartość określonego parametru osobnika.

Można dostrzec tu pewną analogię do podejścia gradientowego, z tą różnicą, że zamiast wektora gradientu istotne jest wzajemne położenie rodziców w przestrzeni rozwiązań dopuszczalnych, a odległość w jakiej proponowany jest potomek zależy jedynie od odległości między rodzicami pomnożonej przez losową liczbę z przedziału [0,1].

3.2.4.c. Operator krzyżowania ziarnistego (reprezentacja rzeczywista)

Operator krzyżowania ziarnistego (ang. *discrete crossover*) polega na rekombinacji poszczególnych parametrów. Nie jest w nim istotna kolejność rodziców – operator jest symetryczny. Osobnik potomny otrzymuje wartości poszczególnych parametrów od jednego albo od drugiego rodzica z równym prawdopodobieństwem.

Można tu dostrzec pewną analogię do krzyżowania w reprezentacji binarnej, z tą różnicą, że od wybranego rodzica dziedziczona jest wartość całego parametru a nie poszczególne bity. Schemat działania opisywanego operatora przedstawia:

```
foreach dimension

if (\lambda \le 0.5) x_i(\text{offspring}) = x_i(\text{parent}_1);

else x_i(\text{offspring}) = x_i(\text{parent}_2);
```

Rys. 3.6 Pseudokod operatora krzyżowania ziarnistego

3.2.5. Operatory mutacji

Operatory mutacji wraz z operatorami krzyżowania składają się na proces reprodukcji, czyli tworzenie nowego pokolenia. Celem mutacji jest wprowadzenie nowego materiału genetycznego do tworzonego pokolenia, czyli zwiększenie różnorodności genetycznej populacji. Dzięki mutacji, w każdej iteracji, z pewnym niezerowym prawdopodobieństwem, możliwe jest utworzenie osobnika z dowolnego miejsca przestrzeni rozwiązań dopuszczalnych. Zapobiega to ograniczeniu działania algorytmu do podprzestrzeni określonej populacją początkową i przedwczesnej zbieżności do lokalnego optimum.

Mutacja polega na losowej, określonej zadanym prawdopodobieństwem, zmianie genotypu osobnika w tworzonym pokoleniu. Prawdopodobieństwo zajścia mutacji, nazywane również współczynnikiem mutacji, jest liczbą rzeczywistą z przedziału [0, 1]. Potencjalnie zmieniony może być każdy z genów każdego z osobników.

W omawianej implementacji algorytmu genetycznego, mutacja dotyczy zarówno przeniesionych z poprzedniego pokolenia rodziców, jak i utworzonych na ich podstawie potomków. Wartość współczynnika mutacji jest stała w danym pokoleniu, ale może być zmienna pomiędzy pokoleniami.

Koncepcja automatycznej zmiany wartości współczynnika mutacji w funkcji liczby pokoleń zakłada stopniowe zmniejszanie prawdopodobieństwa zajścia mutacji. Dzięki temu na początku procesu optymalizacji próbkowana jest cała przestrzeń rozwiązań dopuszczalnych, a pod koniec osobniki z otoczenia globalnego optimum nie są niszczone operatorem mutacji.

W zaproponowanej implementacji dostępne są trzy metody wyznaczania współczynnika mutacji (ang. *mutation ratio*) w odniesieniu do bazowej wartości określanej przez użytkownika (ang. *base probability*) w funkcji numeru pokolenia (ang. *number of generation*):

- mutacja stała współczynnik mutacji pozostaje niezmienny w całym procesie optymalizacji:
 mutationRatio(*nGenerations*) = *baseProbability* (3.13)
 - mutacja hiperboliczna współczynnik mutacji maleje hiperbolicznie wraz z postępem procesu optymalizacji:

$$mutationRatio(nGenerations) = \frac{1}{nGenerations} \cdot baseProbability$$
(3.14)

 mutacja logarytmiczna – współczynnik mutacji maleje hiperbolicznie wraz z logarytmem z postępu procesu optymalizacji:

$$mutationRatio(nGenerations) = \frac{1}{\ln(nGenerations)} \cdot baseProbability$$
(3.15)

Jak przestawiłem w pracy magisterskiej, najlepsze rezultaty osiąga metoda mutacji logarytmicznej. Z testów wynika, że zarówno stopniowe zmniejszanie współczynnika mutacji jak i tempo zmiany jest w niej optymalne. Przy operatorze hiperbolicznym, mutacja jest zbyt szybko wygaszana, co znacząco zwiększa ryzyko utknięcia w lokalnym optimum.

Zaletą metody ze stałą wartością współczynnika mutacji jest możliwość zastosowania jej przy ręcznym sterowaniu procesem optymalizacji. Gdy program pracuje w trybie jednego kroku optymalizacyjnego, użytkownik może, obserwując przebieg algorytmu, łatwo określać wartość współczynnika mutacji dla każdego pokolenia.

W ramach konfiguracji operatora mutacji użytkownik określa:

- bazowe prawdopodobieństwo, z jakim każdy osobnik z nowego pokolenia będzie poddany mutacji;
- metodę wyznaczania współczynnika mutacji w funkcji numeru pokolenia;
- argument operatora mutacji, który precyzuje sposób przeprowadzenia mutacji dla każdego parametru osobnika wskazanego do mutacji. Na jego podstawie obliczana jest

tablica o długości równej liczbie wymiarów, w której każdy element jest iloczynem zadanego argumentu i zakresu zmienności poszczególnych parametrów (wymiarów). Dla parametrów o większym zakresie zmienności wartość jest większa, a przy mniejszym zakresie zmienności mniejsza.

Działanie operatora mutacji różni się w zależności od przyjętej reprezentacji instancji problemu.

Operator mutacji dla reprezentacji binarnej przeprowadza mutację poprzez możliwość zmiany na wartość przeciwną (ang. *flip*) każdego bitu każdego parametru osobnika, który został wskazany do mutacji. Prawdopodobieństwo tej zmiany w obrębie każdego parametru określa odpowiedni element tablicy znormalizowanych argumentów operatora mutacji.

Operator mutacji dla reprezentacji rzeczywistej zmienia wartości każdego parametru, u osobnika wskazanego do mutacji, z rozkładem Gaussa. Do wektora parametrów dodawany jest wektor losowany zgodnie z rozkładem normalnym. Tablica znormalizowanych argumentów operatora mutacji traktowana jest jako wartości odchylenia standardowego dla poszczególnych parametrów.

3.2.6. Kryteria stopu

Kryteria stopu definiują moment, w którym ma kończyć się proces optymalizacyjny. Określają kiedy operatory selekcji, krzyżowania i mutacji zaprzestają tworzenia nowych pokoleń. Wówczas rezultat przeszukiwania przestrzeni rozwiązań dopuszczalnych jest prezentowany użytkownikowi. Dla algorytmu genetycznego możliwych jest wiele kryteriów stopu. Dzielą się one na dwie grupy: ograniczające czas pracy algorytmu oraz określające kryteria zbieżności algorytmu.

Kryteria stopu algorytmu genetycznego związane z czasem pracy to przede wszystkim:

- określona liczba pokoleń;
- określona liczba wywołań funkcji celu;
- określona liczba przebiegów algorytmu (wielokrotny start z losową populacją początkową).

Spośród kryteriów stopu algorytmu genetycznego związanych ze zbieżnością należy wyróżnić:

- brak poprawy najlepszego znalezionego rozwiązania (osobnika) przez określoną liczbę pokoleń;
- brak poprawy średniej wartości funkcji oceny przez określoną liczbę pokoleń;
- zbyt mała zmiana w materiale genetycznym przez określoną liczbę pokoleń;

 znalezienie wystarczająco dobrego rozwiązania – osobnika o wartości funkcji oceny niższej (lepszej) niż zadana wartość.

W zaprezentowanym w pracy magisterskiej algorytmie genetycznym do analizy danych z pomiarów wzbudzeń kulombowskich, zaimplementowano szereg kryteriów stopu związanych zarówno z czasem pracy algorytmu, jak i jego zbieżnością. Po kilku latach analiza użytkowania pierwszej wersji oprogramowania wykazała, że kryteria związane z konwergencją nie są wykorzystywane.

Z tego powodu w implementacji algorytmu genetycznego prezentowanej w tej rozprawie podstawowym kryterium stopu jest wykonanie jednego kroku minimalizacyjnego – jednego pokolenia. Użytkownik może zażądać wykonania określonej liczby iteracji, jednak w każdej chwili może nakazać wstrzymanie procesu po aktualnie przetwarzanym pokoleniu. Z punktu widzenia omawianego algorytmu genetycznego jedynym kryterium stopu jest obliczenie jednego pokolenia na podstawie aktualnie zdefiniowanych przez użytkownika parametrów algorytmu oraz raportowanie wyników.

Użytkownik może określić dla każdego pokolenia:

- operator selekcji elitarnej oraz procent osobników, które mają być wyeliminowane;
- operator selekcji ruletkowej oraz procent osobników, które mają być wyeliminowane;
- operator selekcji turniejowej oraz rozmiar grupy turniejowej;
- maksymalną liczbę dzieci, które mogą zostać wygenerowane przez rodzica w operatorze krzyżowania;
- operator mutacji z niezmiennym współczynnikiem mutacji oraz bazowe prawdopodobieństwo mutacji i argument operatora mutacji;
- operator mutacji ze współczynnikiem mutacji malejącym hiperbolicznie wraz z numerem pokolenia oraz bazowe prawdopodobieństwo mutacji i argument operatora mutacji;
- operator mutacji ze współczynnikiem mutacji malejącym hiperbolicznie wraz z logarytmem naturalnym z numeru pokolenia oraz bazowe prawdopodobieństwo mutacji i argument operatora mutacji.

Po każdym kroku minimalizacyjnym (pokoleniu), za pośrednictwem graficznego interfejsu użytkownika, program prezentuje wszystkie informacje na temat przebiegu procesu optymalizacyjnego. Użytkownik otrzymuje następujące informacje:

- najniższa wartość χ^2 w pokoleniu najlepsza wartość funkcji celu;
- wektor elementów macierzowych, dla których uzyskano najniższą wartość funkcji χ² w pokoleniu;
- średnia wartość χ^2 w pokoleniu średnia wartość funkcji celu;
- najniższa wartość χ^2 w dotychczasowym przebiegu algorytmu genetycznego;
- wektor elementów macierzowych o najniższej wartości χ² w dotychczasowym przebiegu algorytmu genetycznego;
- czas przetwarzania pokolenia;
- czas i liczbę wywołań funkcji celu wyznaczeń wartości χ² dla zadanych wektorów elementów macierzowych przez program GOSIA;
- czas wykonania poszczególnych operatorów genetycznych: selekcji, krzyżowania i mutacji;
- numer pokolenia.

Na ich podstawie użytkownik podejmuje decyzje o dalszym procesie optymalizacji.

Na początku procesu optymalizacyjnego użytkownik wskazuje:

- lokalizację programu GOSIA, w tym "inputu do GOSI";
- rozmiar populacji;
- sposób generowania populacji początkowej (próbkowanie przestrzeni z rozkładem równomiernym albo próbkowanie wokół zadanego punktu przestrzeni z rozkładem Gaussa).

Użytkownik może zapisywać następujące dane do plików:

- najlepszy osobnik w każdym pokoleniu do pliku "best.txt";
- cała populacja po każdym operatorze genetycznym (po selekcji, krzyżowaniu i mutacji) w każdym pokoleniu – do pliku "log.txt";
- najlepszy osobnik w całym procesie optymalizacyjnym do pliku "solution.txt"
- próbkowanie przestrzeni rozwiązań dopuszczalnych przez algorytm genetyczny w całym procesie optymalizacyjnym – do pliku "repository.txt".

Następujące ustawienia algorytmu genetycznego wymagają zmiany i rekompilacji kodu źródłowego (użytkownik nie może ich modyfikować w trakcie procesu optymalizacyjnego):

 reprezentacja instancji problemu – domyślnie reprezentacja rzeczywista z krzyżowaniem heurystycznym-2 (dla reprezentacji binarnej również liczba bitów przeznaczonych do reprezentacji każdego z parametrów);

- funkcja celu domyślnie χ^2 (**ME**) obliczana za pomocą programu GOSIA;
- transformacja wartości funkcji celu domyślnie transformacja logarytmiczna.

Przedstawiona struktura przetwarzania i prezentacji danych z przebiegu algorytmu genetycznego została wypracowana po analizie użytkowania pierwszej wersji oprogramowania w porozumieniu z fizykami. Ma ona na celu maksymalne usprawnienie działania algorytmu genetycznego jako części składowej analizy danych z pomiarów wzbudzeń kulombowskich [79], [88], [89].

3.3. Implementacja algorytmu genetycznego

Omawiany algorytm genetyczny został zaimplementowany w natywnym języku C++ [72], [74], [82] jako biblioteka łączona dynamicznie (ang. *Dynamic-Link Library*, DLL). Jest to, charakterystyczna dla systemów Microsoft Windows, biblioteka współdzielona (ang. *shared library*) o następującej funkcjonalności:

- możliwość przechowywania implementacji różnych funkcji (podprogramów) programu głównego oraz jego zasobów;
- możliwość jednoczesnego importowania przez wiele programów;
- dynamiczne wczytywanie do pamięci operacyjnej (gdy jest faktycznie potrzebna);
- możliwość wykorzystywania w programach napisanych w różnych językach przeznaczonych na platformę Windows.

To właśnie ta ostatnia cecha zdecydowała o implementacji algorytmu genetycznego jako biblioteki DLL. Celem było połączenie szybkości działania języka średniego poziomu z wygodą projektowania graficznego interfejsu użytkownika z wykorzystaniem platformy .NET. Z tego powodu sam algorytm został napisany w natywnym języku C++ (bez automatycznego zarządzania pamięcią), a następnie zapakowany w bibliotekę DLL.

Ponieważ biblioteka współdzielona wymaga by klasy i zasoby publiczne były typami z automatycznym zarządzaniem pamięcią – biblioteka została wyposażona w napisaną w języku Visual C++ publiczną klasę referencyjną GeneticAlgorithm, która pełni rolę fasady całej biblioteki [75]. Zawartość dostępna jest jedynie za jej pośrednictwem, jako interfejsu dostępowego. Jednocześnie biblioteka DLL została nazwana od klasy fasadowej GeneticAlgorithm.dll.

Wszystkie ustawienia algorytmu genetycznego, które wymagają zmiany i rekompilacji kodu źródłowego ograniczają się do modyfikacji referencyjnej klasy fasadowej GeneticAlgorithm. Dodatek A.1. zawiera szczegółowe omówienie wszystkich klas biblioteki GeneticAlgorithm.dll.

3.4. Program JACOB

Program JACOB został napisany w języku C# [83], przy pomocy środowiska Microsoft Visual Studio 2010 i platformy .NET. Stanowi on graficzny interfejs użytkownika dla opisanej w punkcie 3.3 biblioteki GeneticAlgorithm.dll. W szczególności udostępnia pełną funkcjonalność klasy fasadowej GeneticAlgorithm, przedstawionej w punkcie A.1.1.

Program JACOB umożliwia znajdowanie optymalnego zestawu elementów macierzowych przejść elektromagnetycznych pomiędzy stanami jądrowymi dla danych eksperymentalnych z pomiarów wzbudzeń kulombowskich, z wykorzystaniem przedstawionego w tym rozdziale algorytmu genetycznego. Obliczenia związane z pracą algorytmu genetycznego realizowane są jako oddzielny wątek. Dzięki temu użytkownik może w każdej chwili nakazać przerwanie pracy algorytmu po aktualnie przetwarzanym pokoleniu. Ponadto po każdym kroku minimaliza-cyjnym aktualizowana jest informacja o czasach wykonania poszczególnych części algorytmu.

Okno startowe programu JACOB przedstawia rysunek Rys. 3.7. Umożliwia ono:

- wskazanie lokalizacji: "inputu do GOSI" oraz samego programu GOSIA;
- określenie rozmiaru populacji;
- wybór sposobu rozmieszczenia populacji początkowej w przestrzeni rozwiązań dopuszczalnych.

Okno główne programu JACOB przedstawia Rys. 3.8. Składa się ono z czterech części:

- w lewej górnej części użytkownik decyduje ile kroków minimalizacyjnych (pokoleń) algorytm ma wykonać. Ponadto, za pomocą przycisku Simulation/Cancel, uruchamia i zatrzymuje pracę algorytmu genetycznego;
- w prawej górnej części użytkownik zmienia ustawienia operatorów algorytmu genetycznego;
- w lewej dolnej części przedstawione są informacje o pomiarze czasu trwania poszczególnych części algorytmu genetycznego;
- w prawej dolnej części użytkownik może zapisać do pliku repozytorium próbkowania przestrzeni rozwiązań dopuszczalnych przez algorytm genetyczny. Ponadto użytkownik może włączyć/wyłączyć zapis do plików: całej populacji po każdym operatorze genetycznym oraz najlepszego osobnika w każdym pokoleniu.

Welcome in Jacob v2.0										
 Specify the input and Gosia location. Set population size. (optionally) Choose the starting population distribution around the point from input. 										
Input file location:										
D:\Jacob\2d (-40,15)(-40,15).inp										
Gosia location:										
D:Vacob\Gosia\gosia.exe										
Population size: 100 🗢 Generate population round the point in input										
with normal distribution										
OK Sigma: 0,05 🛟										

Rys. 3.7 Okno startowe programu JACOB

Jacob v2.0								
- Simulation		Algorithm settin	igs					
Population size: 100 Best chiSq:	1,834	Selection:	truncation 🔽	Casualties [%]:	60 💲			
Average chiSq: 4,165 Best ME vector: One generation Number of generations 40 Simulation	-25,877 -25,878	Crossing over: Mutation:	Parent by roulette 🗸	Max children: Probability: Sigma:	5 🗘 0,10 🗘 0,20 🗘			
Time measurement		Logs and Repo	sitory					
Time of generation:0,000 sTime of Gosia call:0,000 s				Save rep	ository			
Number of Gosia calls: 62		🗹 Log file						
Time of selection: 0,000 s		D:Vacob\log.txt						
Time of crossing over: 0,000 s								
Time of mutation: 0,000 s		☑ Store the best creature from each generation						
Time of others: 0,000 s		D:Vacob\best.txt	;					
Generation number: 16		Memory usage:	21304 K		.:			

Rys. 3.8 Okno główne programu JACOB

3.5. Schemat analizy danych eksperymentalnych z wykorzystaniem algorytmu genetycznego

Opisywany w tej rozprawie schemat analizy danych eksperymentalnych przedstawia Rys. 3.9.

W porównaniu ze stosowanym wcześniej schematem (vide Rys. 2.9), należy wyszczególnić następujące różnice:

- program GOSIA jest używany jedynie do wyznaczania wartości χ²(ME), w oparciu o zebrane dane eksperymentalne, dla wektorów elementów macierzowych proponowanych przez program JACOB;
- poszukiwanie optymalnego zestawu elementów macierzowych zostało przeniesione z programu GOSIA do programu JACOB;



Rys. 3.9 Schemat analizy danych eksperymentalnych z wykorzystaniem algorytmu genetycznego¹²

¹² Autorem programów JACOB oraz ScanRep jest autor niniejszej rozprawy doktorskiej.

- minimalizacja metodą gradientową została zastąpiona przeszukiwaniem przestrzeni rozwiązań dopuszczalnych za pomocą algorytmu genetycznego;
- program JACOB, realizując proces optymalizacji, wywołuje program GOSIA jako oddzielny proces, gdy potrzebne jest wyznaczenie wartości funkcji celu dla punktu z przestrzeni rozwiązań dopuszczalnych;
- rezultatem pracy programu JACOB jest znalezienie globalnie optymalnego zestawu elementów macierzowych przy użyciu algorytmu genetycznego;
- efektem ubocznym pracy algorytmu genetycznego jest powstanie zbioru punktów z przestrzeni rozwiązań dopuszczalnych, dla których wyznaczono wartości funkcji celu χ²(ME);
- oszacowanie niepewności wyznaczenia optymalnych wartości elementów macierzowych zostało przeniesione z programu GOSIA do programu ScanRep;
- program ScanRep, wykorzystując zebrane repozytorium próbkowania funkcji χ², szacuje przedziały niepewności wyznaczenia optymalnych wartości elementów macierzowych, analizując kształt funkcji w bezpośrednim otoczeniu znalezionego optimum.

Proponowana nowa metoda szacowania niepewności, w oparciu o analizę próbkowania przestrzeni rozwiązań dopuszczalnych przez algorytm genetyczny, została przedstawiona w kolejnych rozdziałach.

4. Analiza próbkowania przestrzeni optymalizacyjnej przez algorytm genetyczny

Przeprowadzenie analizy sposobu próbkowania przestrzeni optymalizacyjnej przez algorytm genetyczny jest kluczowe dla poprawnego oszacowania niepewności wyznaczenia optymalnych wartości elementów macierzowych Metodą AH – nową autorską metodą opracowaną do oceny jakości wyników pomiarów wzbudzeń kulombowskich, prezentowaną w niniejszej rozprawie. Ocena jakości zebranego w procesie optymalizacji repozytorium punktów ma na celu sprawdzenie, czy znajduje się w nim wystarczająco precyzyjna informacja o kształcie funkcji celu wokół znalezionego rozwiązania optymalnego. Wymaga to określenia kryteriów oraz metod, na podstawie których będzie można wnioskować o jakości repozytorium punktów oraz stwierdzić, czy na jego podstawie możliwe jest oszacowanie niepewności parametrów.

Ze względu na złożoność tego zagadnienia, jego opis został przedstawiony w oddzielnym rozdziale. Tym samym prezentacja Metody AH została podzielona na dwa rozdziały. W rozdziale 4. omówiono:

- moduł przetwarzania wstępnego, mający na celu wyodrębnienie z repozytorium punktów znajdujących się w basenie przyciągania znalezionego rozwiązania optymalnego (etapy od A do D);
- moduł analizy rozkładu próbkowania przestrzeni wokół optimum, którego budowa opiera się na statystyce matematycznej i teorii grafów (etapy E oraz F).

W rozdziale 5. opisano:

• moduł wyznaczania niepewności parametrów (etapy G oraz H).

Rozdział 5. zawiera również teoretyczne uzasadnienie stosowania Metody AH do analizy danych z pomiarów wzbudzeń kulombowskich, opis programu ScanRep oraz podsumowanie działania Metody AH.



Schemat blokowy poszczególnych modułów Metody AH przedstawia Rys. 4.1.

Rys. 4.1 Metoda AH: moduły przetwarzania informacji

Poszczególne etapy Metody AH oznaczone są kolejnymi literami alfabetu od A do H. Ze względu na międzynarodowy charakter jej użytkowników, nazwy etapów zostały podane w języku angielskim. Polskie tłumaczenie zamieszczono razem z opisem etapu w odpowiednich punktach rozdziałów 4. oraz 5.

Działanie Metody AH zostało opisane na podstawie przypadku testowego o znanym a priori rozwiązaniu. Ponadto, aby umożliwić wizualizację¹³ poszczególnych etapów, przyjęto że przykładowy problem optymalizacyjny będzie się składał z dwuelementowego wektora parametrów $\mathbf{x} = (x_1, x_2)$ oraz wartości funkcji $y = f(\mathbf{x})$. Rysunki przedstawiono w wariancie dwuwymiarowym (2D) oraz trójwymiarowym (3D). Rysunki 2D będą zawierać współrzędne (x_1, x_2) , a rysunki 3D współrzędne (x_1, x_2, y) .



Rys. 4.2 Funkcja przypadku testowego

¹³ Wizualizacje zostały wykonane przy użyciu programów: Gnuplot [101] oraz Matlab [95] i [96].

4.1. Opis przypadku testowego

W ramach przedstawianego przypadku testowego przyjęto, że:

• dziedziną D, problemu jest zbiór punktów $\mathbf{x} = (x_1, x_2)$ takich że:

$$\mathbf{x} \in D \Leftrightarrow \forall_{i=1,2} \ x_i \in [-40.0, \ 15.0] \tag{4.1}$$

• funkcją celu jest funkcja testowa F7 [98] dana wzorem ogólnym:

$$F7(\mathbf{x}) = \sum_{i} (-1) \cdot x_{i} \cdot \sin \sqrt{|x_{i}|}$$
(4.2)

• wartość funkcji oceny f, dla punktu $\mathbf{x} \in D$ jest równa wartości funkcji $F7(\mathbf{x})$ zwiększonej o 50, aby funkcja oceny przyjmowała wartości dodatnie dla całej dziedziny:

$$f(\mathbf{x}) = F7(\mathbf{x}) + 50 = -x_1 \cdot \sin \sqrt{|x_1|} - x_2 \cdot \sin \sqrt{|x_2|} + 50$$
(4.3)

 zadaniem optymalizacyjnym jest znalezienie minimum funkcji (4.3), czyli takiego x_{min}∈D, że:

$$\forall \mathbf{x} \in D : f\left(\mathbf{x}_{min}\right) \leq f\left(\mathbf{x}\right) \tag{4.4}$$

- niepewności wartości parametrów są wyznaczone dla płaszczyzny progowej¹⁴ $y_{PU} = 10.0$;
- wyniki będą podawane z dokładnością do trzech miejsc po przecinku.

Wykres funkcji zdefiniowanej w (4.3) przedstawiono na Rys. 4.2. Jej kształt jest podobny do kształtu funkcji $\chi^2(ME)$ określonej wzorem (2.5). Funkcja jest multimodalna – jej dziedzina została tak określona, aby funkcja miała cztery minima. Jej topografia jest pofałdowana, brak jest ostrych krawędzi na jej powierzchni. Z tych powodów została ona wybrana na funkcję testową.

Globalne minimum funkcji (4.3), dla określonej powyżej dziedziny, wynosi:

$$f(\mathbf{x}_{min}) \approx 1.8340796038... \approx 1.834$$

$$\mathbf{x}_{min} = (x_1, x_2) = (-25.877, -25.877)$$
 (4.5)

Funkcja (4.3) w najbliższym otoczeniu minimum jest płaska. Dla zaokrąglonej do trzech miejsc po przecinku wartości minimalnej można wskazać następujący przedział zmienności parametrów:

$$\forall \mathbf{x} : \|\mathbf{x} - \mathbf{x}_{\min}\|_{2} \le 0.058 \quad \Rightarrow \quad f(\mathbf{x}) \approx 1.834 \tag{4.6}$$

gdzie $\|\cdot\|_2$ jest normą euklidesową. Wskazany w (4.5) punkt x_{min} został wyznaczony przed zaokrągleniem wartości funkcji.

¹⁴ Zagadnienie wskazywania właściwej płaszczyzny progowej dla estymacji niepewności wartości parametrów w analizie danych z pomiarów wzbudzeń kulombowskich, zostało omówione w punkcie 5.1.

Niepewność wyznaczenia wartości parametrów dla płaszczyzny progowej $y_{PU} = 10.0$ wynosi:

$$\begin{aligned} \forall_{i=1,2} & x_i \in [-33.502, -17.928] \\ czyli & x_i = -25.877^{+7.949}_{-7.625} \end{aligned}$$
(4.7)

Powyższa notacja uwzględnia niesymetryczną niepewność wyznaczenia wartości elementu macierzowego, ponieważ w analizie danych z pomiarów wzbudzeń kulombowskich ma ona właśnie taki charakter.

Przy prezentowaniu oszacowanej niepewności stosuje się notację z dokładnością do dwóch cyfr znaczących. W Metodzie AH stosowana jest dokładność do trzech miejsc po przecinku, co ułatwia zrozumienie niuansów jej działania. W estymacji niepewności wyznaczenia wartości elementów macierzowych wystarczy zaokrąglić wynik do dwóch cyfr znaczących.

4.2. Porównywane metody próbkowania przestrzeni – Raw Dataset (A)

Pierwszym etapem Metody AH (ang. *raw dataset* – surowy zbiór danych) jest zgromadzenie repozytorium punktów będących efektem próbkowania przestrzeni optymalizacyjnej. Jest ono zapisywane na dysku w postaci pliku .rep o składni określonej w dodatku A.4. Repozytorium może pochodzić z dowolnego źródła, np. być ubocznym skutkiem działania opisanego w rozdziale 3. algorytmu genetycznego.

Z punktu widzenia Metody AH, algorytm genetyczny jest formą sztucznej inteligencji, próbkującą przestrzeń optymalizacyjną w specyficzny, właściwy sobie sposób. Jego cechą charakterystyczną jest zagęszczanie próbkowania w obszarach wokół lokalnych optimów, a w szczególności w basenie przyciągania optimum globalnego. Jest to korzystne, gdyż użycie Metody AH wymaga posiadania możliwie precyzyjnej informacji o kształcie funkcji wokół optimum. Zagęszczanie próbkowania wokół optimum zmniejsza liczbę punktów, z którego musi składać się repozytorium, aby uzyskać wystarczającą jakość próbkowania.

Jednocześnie, aby próbkowanie przestrzeni mogło być wykorzystane w Metodzie AH, musi zawierać informację o całym otoczeniu znalezionego optimum. Kontrprzykładem jest tu próbkowanie przestrzeni charakterystyczne dla algorytmu gradientowego, który znajduje optimum, próbkując w kolejnych iteracjach przestrzeń ciągiem punktów, które można połączyć krzywą. Poza tą ścieżką poszukiwań (ang. *seeking track*), próbkowanie przestrzeni przez algorytm gradientowy nie zawiera informacji o basenie przyciągania optimum. Z drugiej strony

najwięcej danych na temat otoczenia optimum zawiera próbkowanie równomierne. Niestety dla przestrzeni o dużej liczbie wymiarów jest ono najczęściej zbyt kosztowne obliczeniowo.

Należy zakładać, że nie każda implementacja algorytmu genetycznego będzie próbkowała otoczenie optimum w pożądany przez Metodę AH sposób. Jednocześnie, ze względu na stochastyczny charakter podejścia ewolucyjnego, jeśli konkretna wersja algorytmu daje dobre efekty, to należy się spodziewać, że dla podobnych zadań optymalizacyjnych tworzone przez nią repozytoria będą podobnej jakości. Jest to istotny wniosek, gdyż funkcja χ^2 (**ME**) ma podobną charakterystykę niezależnie od analizowanych danych eksperymentalnych.

Średnią liczbę punktów w repozytorium utworzonym z wykorzystaniem algorytmu genetycznego o stałym współczynniku mutacji można wyznaczyć ze wzoru:

 $GApoints = populationSize + nGenerations \cdot (nCasualties + nSurvivors \cdot mutationRatio)$ (4.8) gdzie zgodnie z przyjętymi w rozdziale 3. oznaczeniami:

GApoints – oczekiwana średnia liczba punktów,

populationSize – rozmiar populacji,

nGenerations – liczba pokoleń,

nCasualties – liczba osobników, wyeliminowanych przez selekcję,

nSurvivors – liczba osobników, które przeżyją selekcję,

mutationRatio – współczynnik mutacji.

W ramach przypadku testowego, porównywane są trzy typy próbkowania przestrzeni:

- próbkowanie równomierne, w którym punkty rozmieszczone są na siatce (ang. grid)
 co 1.0. Dla zdefiniowanej powyżej dziedziny daje to 56²=3136 punktów
 w repozytorium. Zostało ono oznaczone grid3k;
- próbkowanie losowe (ang. *random*), w którym współrzędne punktów losowane są z rozkładem równomiernym oddzielnie dla każdego wymiaru. Jest to przykład zastosowania Metody Monte Carlo¹⁵. Przyjęto, że repozytorium będzie zawierać 3120 punktów. Zostało ono oznaczone rand3k;

¹⁵ Metoda Monte Carlo została opracowana i pierwszy raz zastosowana przez Stanisława Ulama, wybitnego polskiego matematyka pierwszej połowy XX wieku. Jest to metoda numeryczna polegająca na skonstruowaniu gry losowej opartej na modelu, która prowadzi do rozwiązania zadania [91]. Losowość polega na losowaniu według rozkładów znanych skądinąd. Na podstawie wyznaczonej charakterystyki badanego procesu znajdowane jest ostateczne rozwiązanie. W przypadku zadań optymalizacyjnych sprowadza się to do losowego próbkowania przestrzeni rozwiązań. Jako ostatecznie najlepsze wybierane jest najlepsze rozwiązanie ze zbioru wylosowanych.

próbkowanie algorytmem genetycznym, opisanym w rozdziale 3., o następującej konfiguracji: reprezentacja rzeczywista, 40 osobników w populacji, 120 pokoleń, selekcja elitarna z progiem odcięcia 60%, krzyżowanie heurystyczne-2 z parametrem *maxChildren*=5, mutacja gaussowska ze stałym współczynnikiem mutacji 10% oraz argumentem operatora mutacji *sigma*=0.20¹⁶. Zgodnie ze wzorem (4.8) repozytorium zawiera średnio 3112 punktów. Zostało ono oznaczone ga3k.

Repozytoria grid3k oraz rand3k zostały utworzone przy użyciu programu RepGen (vide dodatek A.2.), zaś repozytorium ga3k z wykorzystaniem programu JACOB. Dla każdego punktu wyznaczono wartość $f(\mathbf{x})$ zgodnie ze wzorem (4.3). Każde z repozytoriów rand3k oraz ga3k zostało utworzone 10 razy. Do dalszej analizy wybrano najlepsze z każdej grupy. Jako kryterium porównawcze zastosowano precyzję wskazania minimum oraz liczbę punktów w basenie jego przyciągania.

Etap	rep. 1	rep. 2	rep. 3	rep.4	rep. 5	rep. 6	rep. 7	rep. 8	rep. 9	rep. 10	Średnio	Najlepsze	Najgorsze	grid3k
Α	3120	3120	3120	3120	3120	3120	3120	3120	3120	3120	3120.0	3120	3120	3136
В	986	974	958	951	1008	1008	953	981	1021	999	983.9	1021	951	964
С	720	694	700	694	724	704	699	700	735	711	708.1	735	694	964
D	526	513	524	504	533	517	509	507	521	512	516.6	533	504	708
Mini mum	1.853	1.840	1.923	1.919	1.892	1.907	1.885	1.878	1.864	1.877	1.884	1.840	1.923	1.838

Tabela 4.1 Porównanie dziesięciu repozytoriów rand3k.

Etap	rep. 1	rep. 2	rep. 3	rep.4	rep. 5	rep. 6	rep. 7	rep. 8	rep. 9	rep. 10	Średnio	Najlepsze	Najgorsze		grid3k
Α	3096	3104	3141	3102	3094	3097	3126	3101	3125	3122	3110.8	3141	3094	ſ	3136
В	2934	2947	2934	2946	2948	2930	2951	2951	2970	2960	2947.1	2970	2930		964
с	380	363	365	340	376	385	360	369	351	380	366.9	385	340		964
D	362	350	351	326	357	374	349	360	337	368	353.4	374	326		708
Minim um	1.834	1.834	1.834	1.834	1.834	1.834	1.834	1.834	1.834	1.834	1.834	1.834	1.834		1.838

Tabela 4.2 Porównanie dziesięciu repozytoriów ga3k.

16 Wstępne testy pokazały, że takie ustawienia dają repozytoria o najlepszej jakości. Użycie reprezentacji binarnej albo innych operatorów krzyżowania lub mutacji znacząco obniżają jakość repozytorium. Pozostałe ustawienia mają mniejszy wpływ.







Rys. 4.3 oraz Rys. 4.4 przedstawiają wizualizacje etapu A dla repozytorium grid3k odpowiednio w wariancie 2D oraz 3D. Analogiczne wizualizacje dla repozytoriów rand3k oraz ga3k zawierają Rys. 4.5 – Rys. 4.8.

W Tabeli 4.1 przedstawiono liczby punktów we wszystkich dziesięciu repozytoriach rand3k po kolejnych etapach przetwarzania wstępnego Metody AH. Tabela 4.2 zawiera dane dotyczące dziesięciu repozytoriów ga3k. W obu tabelach wyróżniono repozytorium wybrane do dalszej analizy. Zawarto tam również średnie, najlepsze i najgorsze wartości dla poszczególnych etapów. Dla porównania skrajna prawa kolumna zawiera analogiczne informacje o repozytorium grid3k.

4.3. Ograniczenie jakościowe punktów w repozytorium – Threshold Cut (B)

Drugim etapem Metody AH (ang. *threshold cut* – cięcie progowe) jest usunięcie z repozytorium punktów, których wartość $f(\mathbf{x})$ jest zbyt wysoka. Nie wnoszą one żadnej informacji o kształcie funkcji wokół optimum, gdyż znajdują się w zupełnie innych miejscach przestrzeni, zatem są zbędne. Ich usunięcie usprawnia proces grupowania w etapie D. Ponadto ograniczenie liczby punktów w repozytorium skraca czas obliczeń w następnych etapach, a pozostaje bez wpływu na dokładność wyników.

Kluczowe jest właściwe dobranie progu cięcia, to jest wskazanie wartości y_{cut} dla której:

 $\forall \mathbf{x} : f(\mathbf{x}) \ge y_{cut} \implies \mathbf{x} \text{ jest usuwany}$ Należy wybrać taką wartość y_{cut} aby: (4.9)

- oddzielić poszczególne baseny przyciągania optimów. Wartość y_{cut} musi być dostatecznie niska, aby skupiska punktów powstałe w wyniku cięcia progowego oddzielone były pustymi obszarami, zgodnie z otoczeniami poszczególnych minimów;
- ograniczyć czas obliczeń. Wartość y_{cut} powinna być dostatecznie niska, aby wyeliminować z repozytorium te punkty, które nie dostarczają informacji o kształcie funkcji wokół minimum globalnego, a jedynie wydłużają czas obliczeń poprzez sztucznie zawyżanie rozmiaru repozytorium;
- nie zmniejszyć dokładności oszacowania niepewności parametrów. Wartość y_{cut} musi być dostatecznie wysoka, aby nie miała wpływu na dokładność oszacowania niepewności parametrów (vide opis etapu G oraz H).

Dla zastosowania Metody AH do szacowania niepewności elementów macierzowych w analizie danych z pomiarów wzbudzeń kulombowskich można przyjmować, że:

$$y_{cut} \ge 10 \tag{4.10}$$

Wynika to z następujących faktów i obserwacji:

- wartość punktu przestrzeni wskazanego przez algorytm optymalizacyjny jako minimum, zgodnie ze wzorem (2.6) nie przekracza liczby 3;
- płaszczyzna, dla której wyznaczane są niepewności parametrów, znajduje się na $y_{PU} \leq 4$;
- wyznaczana w etapie H wartość *furthest* $_{EFL} \leq 10$.

Jeśli *furthest*_{EFL} $\approx y_{cut}$, to należy tak zwiększyć wartość y_{cut} , aby *furthest*_{EFL} $< y_{cut}$. Różnica między tymi parametrami nie musi być bardzo duża, ale powinna być zauważalna.

Dla omawianego przypadku testowego, na podstawie analizy kształtu funkcji określonej wzorem (4.3), przyjęto $y_{cut} = 28$.

Na podstawie analizy drugiego wiersza Tabel 4.1 i 4.2, można określić na ile dany typ próbkowania przestrzeni koncentruje się wokół optimów. Dla repozytoriów typu grid3k i rand3k jedynie ok. 30% punktów pozostaje w repozytorium po etapie B Metody AH. Dla repozytoriów typu ga3k ponad 90% punktów leży w basenie przyciągania minimów.

Umożliwia to ilościowe oszacowanie, jaka część repozytoriów jest bezużyteczna z punktu widzenia późniejszego szacowania niepewności parametrów. Dla próbkowania przestrzeni z wykorzystaniem algorytmu genetycznego, jedynie kilka procent punktów jest odrzucanych, podczas gdy przy próbkowaniu równomiernym albo Metodą Monte Carlo odrzucone jest wielokrotnie więcej.

Dla przypadku testowego, w którym wektor parametrów ma tylko dwa elementy, nawet gorsza metoda próbkowania przestrzeni może zostać zaakceptowana, gdyż nie powoduje nieakceptowalnie wysokiego wzrostu czasu obliczeń. Jednak w przypadku szacowania niepewności wyznaczonych wartości elementów macierzowych w analizie danych z pomiarów wzbudzeń kulombowskich, gdzie przestrzeń optymalizacyjna ma dużo więcej wymiarów, wyklucza to zastosowanie próbkowania równomiernego bądź Metodą Monte Carlo.

Rys. 4.9 – Rys. 4.14 przedstawiają wizualizacje etapu B repozytoriów grid3k, rand3k oraz ga3k w wariantach 2D i 3D.






4.4. Ograniczenie gęstości próbkowania – Minimum Distance Rule (C)

Trzecim etapem Metody AH (ang. *Minimum Distance Rule*, MDR – zasada minimalnej odległości) jest ograniczenie gęstości próbkowania przestrzeni w repozytorium. Wstępne testy pokazały, że w repozytoriach powstałych z wykorzystaniem algorytmu genetycznego losowo pojawiają się miejsca ze znacząco zagęszczonym próbkowaniem. Takie skupiska punktów na małym obszarze nie wnoszą dodatkowej informacji o kształcie całego otoczenia optimum. Ponadto ich obecność uniemożliwia sprawdzenie czy próbkowanie jest wystarczające w całym basenie przyciągania optimum (etap E oraz F) oraz znacząco utrudnia proces grupowania (etap D).

Aby temu przeciwdziałać, sformułowano zasadę minimalnej odległości, która definiuje maksymalną gęstość próbkowania. Opiera się ona na założeniu, że w repozytorium nie mogą znajdować się dwa punkty, pomiędzy którymi odległość¹⁷ jest mniejsza niż określona wartość, podawana jako argument procedury MDR. Stwarza to możliwość przesiania miejsc ze zbyt dużą gęstością punktów, bez wpływu na pozostałą część repozytorium.

Przy wyborze punktów, które mają pozostać w repozytorium, a które zostać usunięte, brane są pod uwagę ich wartości $f(\mathbf{x})$. Pozostawiane są lepsze punkty, a więc te z niższą wartością $f(\mathbf{x})$. Algorytm MDR najpierw sortuje zbiór punktów, rosnąco ze względu na wartość $f(\mathbf{x})$. Otrzymana kolejka jest przeglądana w celu określenia, które punkty mają pozostać w repozytorium. Usuwane są te, których odległość od któregokolwiek z już zatwierdzonych punktów jest mniejsza niż zadany argument.

Algorytm rozrzedza przypadkowe gęste skupiska punktów. Jednocześnie nie ma wpływu na inne części basenu przyciągania optimum. Kluczowy jest właściwy dobór wartości argumentu procedury.

Schemat działania algorytmu MDR jest następujący:

1.Umieść punkty w kolejce rosnącej ze względu na ich wartość f(${f x}$).										
2.Dopóki są punkty do sprawdzenia:										
° weź z kolejki pierwszy niesprawdzony punkt;										
 sprawdź czy punkt jest w dostatecznej odległości od 										
wszystkich już zatwierdzonych punktów:										
 (TAK) zatwierdź punkt, 										
■ (NIE) usuń punkt.										



17 Za odległość między dwoma punktami w repozytorium przyjmuje się odległość euklidesową ich wektorów parametrów \mathbf{x} . Przy wyznaczaniu odległości nie bierze się pod uwagę wartości $f(\mathbf{x})$.

Dla zastosowania Metody AH do szacowania niepewności elementów macierzowych w analizie danych z pomiarów wzbudzeń kulombowskich, właściwa wartość argumentu procedury MDR wyznaczana jest empirycznie – na podstawie analizy statystycznej i histogramowej punktów pozostałych w repozytorium (vide opis etapu E oraz F). W trakcie tego procesu można określić jakość repozytorium, a więc stwierdzić czy wyznaczenie niepewności wartości parametrów może być oszacowane z wystarczającą dokładnością przy wykorzystaniu konkretnego zbioru punktów.

Przyjęta wartość argumentu procedury MDR (minimalna odległość między punktami) ma pośredni wpływ na dokładność oszacowania niepewności parametrów.

Dla omawianego przypadku testowego ustalono ją na 0.5. Taka wartość daje satysfakcjonujące wyniki etapów E oraz F, dostarczając informacji o jakości repozytorium oraz umożliwia porównanie wszystkich trzech typów próbkowania.

Przyjęta wartość argumentu procedury MDR jest niższa niż odległość między punktami w próbkowaniu równomiernym, dlatego nie ma ona wpływu na repozytorium grid3k. W repozytoriach rand3k oraz ga3k procedura MDR rozrzedza próbkowanie. Szczególnie zauważalne jest to w odniesieniu do licznego skupiska punktów znajdujących się tuż przy minimum globalnym w repozytorium ga3k.

Rys. 4.16 – Rys. 4.21 przedstawiają wizualizacje etapu C repozytoriów grid3k, rand3k oraz ga3k w wariantach 2D i 3D.

Jak wynika z wiersza trzeciego Tabel 4.1 i 4.2, w efekcie przeprowadzenia etapu C, repozytorium ga3k zostało zmniejszone ośmiokrotnie. Jego rozmiar stał się około dwukrotnie mniejszy od referencyjnych repozytoriów grid3k oraz rand3k. Wynika to częściowo z faktu, że repozytoria grid3k oraz rand3k zawierają więcej punktów znajdujących się w basenach przyciągania innych (lokalnych) minimów. Zgodnie z wierszem czwartym tabel punkty te zostaną wykluczone z repozytoriów podczas etapu D. Ponadto na Rys. 4.21 widoczny jest spadek gęstości rozmieszczenia punktów w repozytorium ga3k dla obszarów znajdujących się na obrzeżach basenu przyciągania minimum globalnego. Wyniki etapu F potwierdzą tę obserwację.

77







4.5. Detekcja basenów przyciągania optimów – Clusterization (D)

Czwartym etapem Metody AH (ang. *clusterization* – klasteryzacja, grupowanie) jest detekcja grup punktów przynależnych do poszczególnych basenów przyciągania optimów. Dla oszacowania niepewności wartości parametrów istotna jest informacja o kształcie funkcji wokół minimum globalnego, dlatego do dalszej analizy zebrane w repozytorium próbkowanie przestrzeni ograniczane jest jedynie do punktów znajdujących się w jego otoczeniu.

Wykrycie znaczących skupisk punktów należących do basenów przyciągania innych minimów (optimów lokalnych) może być istotne w początkowej fazie analizy danych z pomiarów wzbudzeń kulombowskich. Jeśli znalezione lokalne minimum jest podobnej głębokości co wyznaczone optimum globalne, może to świadczyć o istnieniu alternatywnego rozwiązania numerycznego problemu optymalizacyjnego. Jest to istotne z punktu widzenia zastosowanych modeli i przewidywań teoretycznych oraz stanowi przesłankę dla lepszego sformułowania (dookreślenia) funkcji celu w ramach konfiguracji programu GOSIA.

Podział punktów znajdujących się w repozytorium na klastry ze względu na ich przynależność do basenów przyciągania optimów stanowi zadanie klasyfikacyjne. Głównym celem etapu B było rozdzielenie skupisk punktów przynależnych do otoczeń poszczególnych minimów pustymi obszarami. Istnienie takich "przerw" między grupami punktów umożliwia użycie jednego z istniejących algorytmów klasteryzacji do wyodrębnienia w repozytorium klastrów punktów związanych z poszczególnymi obszarami przestrzeni.

Problem klasteryzacji jest szeroko omówiony w literaturze [90], [92], [93], [94], [99], [100]. Do realizacji procedury grupowania w ramach etapu D Metody AH został zaimplementowany algorytm NBC (ang. *Neighborhood-Based Clustering algorithm* – grupowanie oparte na otoczeniu/sąsiedztwie) autorstwa Shuigeng Zhou, Yue Zhao, Jihong Guan i Joshua Huang.

Według autorów algorytm NBC jest "algorytmem grupowania bez nadzoru, który potrafi wykryć klastry o dowolnym kształcie i gęstościach". Jest to istotne, gdyż dla zastosowania Metody AH do analizy danych z pomiarów wzbudzeń kulombowskich nie jest znany a priori kształt basenu przyciągania optimum. Również gęstość próbkowania przestrzeni nie jest precyzyjnie określona.

Poniżej przedstawiono szkie działania algorytmu NBC wraz z intuicyjnym opisem działania. Pełen opis znajduje się w [102]. Algorytm NBC przyjmuje na wejściu:

- repozytorium punktów, które mają zostać pogrupowane;
- wartość argumentu k , który określa liczbę elementów zbioru k-najbliższych sąsiadów (ang. *k-nearest neighbors set*) dla każdego punktu.

Grupowanie wejściowego repozytorium odbywa się jedynie w oparciu o wartość argumentu k.

Rezultatem działania algorytmu NBC jest przydzielenie każdemu punktowi w repozytorium etykiety określającej do którego klastra należy. Alternatywnie punkt może otrzymać etykietę identyfikującą go jako szum (ang. *noise*).

Najpierw, dla każdego punktu *p* w repozytorium, wyznaczana jest wartość współczynnika *NDF* (ang. *Neighborhood-based Density Factor* – współczynnik gęstości oparty na otoczeniu/sąsiedztwie):

$$NDF(p) = \frac{|RkNB(p)|}{|kNB(p)|}$$
(4.11)

gdzie:

kNB(p) – (ang. *k-neighborhood* – k-te otoczenie/sąsiedztwo) zbiór punktów, których odległość od punktu *p* jest nie większa niż k-tego najbliższego sąsiada;

RkNB(p) – (ang. reverse k-neighborhood – odwrotne k-te otoczenie/sąsiedztwo) zbiór punktów, dla których punkt p należy do ich kNB;

- liczba elementów zbioru, wielkość zbioru.

Wielkość zbioru kNB(p) może być większa niż wartość k. Dzieje się tak, ponieważ więcej punktów może znajdować się w tej samej odległości co k-ty sąsiad punktu p. W takim przypadku wszystkie zaliczane są do zbioru kNB(p). Wartość współczynnika NDF(p) stanowi miarę gęstości otoczenia punktu p. Jest to gęstość określona w sensie lokalnym (relatywnym), a nie globalnym (absolutnym). Umożliwia to, według Autorów, grupowanie zbiorów o różnych gęstościach.

Współczynnik NDF dzieli punkty na dwie kategorie. Jeżeli:

- $NDF(p) \ge 1$ to punkt p jest punktem lokalnie gestym (ang. *local dense point*)¹⁸;
- NDF(p) < 1 to punkt p jest punktem lokalnie rzadkim (ang. *local sparse point*).

¹⁸ Autorzy algorytmu wydzielają z tej kategorii punkty lokalnie równe (ang. *local even point*), dla których $NDF(p) \approx 1$, ale w dalszej części opisu algorytmu są one traktowane identycznie jak punkty lokalnie gęste. Z tego powodu w niniejszym opisie połączono te kategorie.

Schemat działania algorytmu NBC jest następujący:

```
    Dla zadanego k, wyznacz wartości NDF(p) każdego punktu w repozytorium.
    Dopóki w repozytorium są niezaklasyfikowane lokalnie gęste punkty:

            a) losowo wybierz niezaklasyfikowany lokalnie gęsty punkt p
            i rozpocznij tworzenie nowego klastra c;
            b) zaklasyfikuj punkt p do klastra c;
            c) zaklasyfikuj wszystkie niezaklasyfikowane punkty z kNB(p)
            do klastra c;
            d) rekurencyjnie powtórz kroki b) i c) dla każdego lokalnie gęstego punktu p' należącego do kNB(p).

    Pozostałe niezaklasyfikowane punkty w repozytorium uznaj za szum.
```

Rys. 4.22 Pseudokod algorytmu NBC

Dla właściwego rezultatu działania algorytmu NBC, kluczowe jest umiejętne dobranie wartości argumentu k do klasyfikowanego zbioru punktów. Autorzy sugerują, że dla $k \approx 10$ algorytm działa w pożądany sposób. Wartość argumentu k ma wpływ na minimalną liczność grupy punktów, które mogą zostać uznane za klaster. Minimalna wielkość klastra wynosi k+1.

Ważnym wnioskiem z obserwacji sposobu działania algorytmu NBC jest zależność wyniku od kolejności przetwarzania punktów. Istotna jest kolejność, w jakiej algorytm wybiera niezaklasyfikowane lokalnie gęste punkty, co ma wpływ na przydział lokalnie rzadkich punktów do klastrów. Jeśli taki punkt r należy do k-tego otoczenia dwóch lokalnie gęstych punktów p_1 i p_2 , zaklasyfikowanych do różnych klastrów c_1 i c_2 , to punkt r zostanie zaklasyfikowany do klastra c_1 albo c_2 zależnie od tego który z punktów p_1 czy p_2 będzie przetwarzany jako pierwszy:

Właściwość ta (pominięta w opisie oryginalnego algorytmu) została wykorzystana w tej implementacji.

Aby uniknąć dwuznaczności, repozytoria przetwarzane w ramach etapu D są posortowane rosnąco ze względu na wartość $f(\mathbf{x})$, zaś w kroku 2a algorytmu NBC punkty wybierane są według ich kolejności w repozytorium, a nie losowo. Dzięki temu potencjalnie sporne lokalnie rzadkie punkty są przydzielane do klastrów zawierających punkty z niższymi wartościami $f(\mathbf{x})$. Zwiększa to liczbę punktów zaklasyfikowanych do basenu przyciągania globalnego minimum. Wstępne testy algorytmu NBC ujawniły istotne trudności w jego zastosowaniu do detekcji grup punktów przynależnych do poszczególnych basenów przyciągania optimów. Okazało się, że algorytm daje pożądane rezultaty jedynie dla zbiorów danych, w których gęstość próbkowania jest podobna w obrębie poszczególnych klastrów. Może się ona natomiast różnić pomiędzy klastrami. Dla repozytoriów rand3k oraz ga3k rezultat działania algorytmu NBC jest daleki od oczekiwanego.

Z tego powodu konieczne okazało się opracowanie procedury MDR, która rozrzedzając przypadkowe gęste skupiska punktów, znacząco zmniejsza różnicę w gęstości próbkowania w obrębie danego klastra. Przeprowadzenie procedury MDR przed algorytmem NBC przynosi znaczącą poprawę.

W podpunkcie 4.5.1. przedstawiam modyfikację algorytmu NBC, która zwiększa jego skuteczność w detekcji grup punktów przynależnych do poszczególnych basenów przyciągania optimów. Różnica jest szczególnie widoczna dla małych wartości argumentu k.

Rys. 4.23 – Rys. 4.28 przedstawiają wizualizacje etapu D repozytoriów grid3k, rand3k oraz ga3k w wariantach 2D i 3D.

Dla zawierającego próbkowanie równomierne repozytorium grid3k (stała gęstość próbkowania w obrębie klastra), wystarczającą wartością argumentu k jest 5. Dla repozytoriów rand3k oraz ga3k należało przyjąć k=40. W prezentowanych wynikach etapu D nie wykorzystano opisanej w podpunkcie 4.5.1. modyfikacji algorytmu NBC. Są to rezultaty oryginalnej wersji algorytmu NBC po przeprowadzeniu w ramach etapu C procedury MDR.

Etap D kończy moduł przetwarzania wstępnego Metody AH. Klaster zawierający minimum globalne jest interpretowany jako zbiór punktów należących do basenu przyciągania optimum. W dalszych etapach Metoda AH przetwarza informację zawartą jedynie w tym klastrze. W punktach 4.6. i 4.7. opisany jest moduł analizy próbkowania przestrzeni, który dostarcza ilościowego oszacowania czy wyodrębnione próbkowanie otoczenia minimum jest wystarczające do oszacowania niepewności wartości parametrów.

4.5.1. Łączenie przylegających klastrów – modyfikacja algorytmu NBC

Rys. 4.29 przedstawia wizualizację 2D rezultatu działania algorytmu NBC na repozytorium rand3k dla k=7. Algorytm mylnie podzielił grupy punktów przynależnych do poszczególnych basenów przyciągania optimów na mniejsze klastry. Stało się tak, ponieważ gęstość próbkowania w obrębie otoczenia optimów nie jest stała.









Rys. 4.29 Repozytorium rand3k. NBC(k=7). Bez łączenia przyległych klastrów. 2D

Dla zastosowania algorytmu NBC do analizy danych z pomiarów wzbudzeń kulombowskich, w przypadku pogrupowania punktów z repozytorium w dużą liczbę klastrów zachodzi uzasadnione podejrzenie, że algorytm mylnie podzielił baseny przyciągania optimów na mniejsze klastry. W takim przypadku, aby otrzymać prawidłową klasyfikację repozytorium, można postąpić dwojako:

 zwiększyć wartość argumentu k. Dzięki temu zwiększa się zarówno minimalną liczność klastra, jak i wielkość otoczenia punktu w rozumieniu algorytmu NBC. Skutkuje to powiększeniem obszaru, w którym lokalnie gęsty punkt szuka innych lokalnie gęstych punktów, aby rekurencyjnie dołączać je do tworzonego klastra;



Rys. 4.30 Repozytorium rand3k. NBC(k=7). Po połączeniu przyległych klastrów. 2D

 wykorzystać specyfikę przetwarzanych repozytoriów i przeprowadzić a posteriori procedurę łączenia przylegających klastrów (ang. *merge clusters* – połącz klastry). Taka procedura zaprezentowana jest poniżej.

W przetwarzanych w ramach etapu D Metody AH repozytoriach, poszczególne baseny przyciągania optimów przedzielone są pustymi obszarami, a punkty posortowane są rosnąco ze względu na wartość $f(\mathbf{x})$. W takim przypadku można, przeglądając kolejno punkty w repozytorium, przeprowadzić próbę scalenia klastrów w obrębie otoczeń poszczególnych minimów, w oparciu o zbiory kNB.

Schemat działania algorytmu łączenia przyległych klastrów jest następujący:

```
    Weź kolejny punkt p z repozytorium, który nie został uznany za szum.
    Jeśli w kNB(p) są punkty uznane za szum, to dołącz je do klastra z
punktem p.
    Jeśli w kNB(p) są punkty z innych klastrów, to dołącz te klastry do
klastra z punktem p.
    Powtarzaj kroki od 1 do 3, aż przetworzone zostanie całe repozytorium.
```

Rys. 4.31 Pseudokod algorytmu łączenia przyległych klastrów

Algorytm łączenia przyległych klastrów przynosi pożądane efekty, gdy dla wszystkich punktów z basenu przyciągania danego optimum wszystkie punkty należące do ich zbiorów kNB należą do tego samego basenu. Założenie to jest spełnione, gdy puste obszary między otoczeniami minimów są wystarczająco szerokie, a wartość argumentu k algorytmu NBC (określająca wielkość otoczenia punktu) jest wystarczająco mała.

Z tych powodów prawidłowe przeprowadzenie etapu D wymaga oceny czy otrzymane rezultaty są akceptowalne z punktu widzenia fizyki teoretycznej. Procedura łączenia przyległych klastrów pozostawiona została do ewentualnego zastosowania w ramach etapu D Metody AH.

Rys. 4.30 przedstawia wizualizację 2D efektu działania procedury łączenia przyległych klastrów na repozytorium rand3k dla k=7. Porównanie Rys. 4.29 z Rys. 4.30 pokazuje działanie procedury.

4.6. Metody statystycznego badania rozkładu próbkowania – Statistics (E)

Piątym etapem Metody AH (ang. *statistics* – dane statystyczne, statystyka matematyczna) jest badanie rozkładu próbkowania przestrzeni za pomocą metod statystycznych. Spośród punktów zebranych początkowo w repozytorium, na skutek przeprowadzenia czterech pierwszych etapów (modułu przetwarzania wstępnego), wyodrębnione zostały te, które leżą w basenie przyciągania globalnego minimum. Pozostałe punkty są usuwane z repozytorium. Z tego powodu analiza statystyczna punktów pozostałych w repozytorium jest jednocześnie analizą statystyczną danego typu próbkowania przestrzeni wokół optimum.

Etapy E oraz F, składające się na moduł analizy próbkowania, służą do oceny jakości repozytorium. Dostarczają ilościowej oceny czy wyodrębnione próbkowanie otoczenia minimum jest wystarczające do estymacji niepewności wartości parametrów [103]. W module analizy próbkowania badaniu poddawane są jedynie wektory parametrów x poszczególnych punktów. Dla skrócenia zapisu używany jest termin punkty.

W ramach etapu E punkty z repozytorium traktowane są jako próbki n-wymiarowej zmiennej losowej $(X_1, X_2, ..., X_n)$. Dziedzina problemu stanowi przestrzeń próbkową. Charakterystyka zmiennych losowych $X_1, X_2, ..., X_n$ stanowi przedmiot analizy.

Próbkowanie równomierne oraz losowe z rozkładem równomiernym (repozytoria grid3k oraz rand3k), zgodnie z zasadami statystyki matematycznej [97], dostarczają informacji zarówno o rozkładzie samej próby, jak i kształcie basenu przyciągania optimum. Wynika to z faktu, że wartości parametrów cechy elementów populacji są z prawdopodobieństwem 1 zbieżne do ich charakterystyk teoretycznych (Prawa Wielkich Liczb).

Ze względu na stochastyczny charakter procesu ewolucyjnego należy zakładać, że również próbkowanie przestrzeni algorytmem genetycznym (repozytorium ga3k) ma tę własność. Zarówno wstępne testy, jak i wyniki etapu E potwierdzają tę hipotezę.

W ramach etapu E prezentowana informacja na temat repozytorium zawiera:

- ogólną informację o klastrze (ang. general info) rozumianym jako zbiór punktów, które pozostały w repozytorium po etapie D. Składa się na nią: liczba punktów oraz wartość f(x) i wektor x najlepszego punktu;
- zakresy próbkowania (ang. *sampling ranges*) przedziały zmienności wartości elementów wektora parametrów x dla zbioru punktów: wartość minimalna, wartość maksymalna, wielkość przedziału;
- statystyki: wartość przeciętna, odchylenie standardowe, znormalizowane odchylenie standardowe, skośność, kurtoza;
- macierz kowariancji oraz macierz korelacji.

Do wyznaczania wartości poszczególnych funkcji próbki (statystyk) przyjęto następujące wzory na parametry empiryczne. Wymienione estymatory są zgodne i nieobciążone:

• wartość przeciętna dla i-tego wymiaru informuje, gdzie koncentruje się próbkowanie:

$$\bar{x}_{i} = \frac{1}{N} \sum_{k=1}^{N} x_{ik}$$
(4.13)

 odchylenie standardowe dla i-tego wymiaru (pierwiastek z wariancji empirycznej) informuje o rozrzucie próbkowania wokół wartości średniej:

$$s_{i} = \sqrt{\frac{1}{N-1} \sum_{k=1}^{N} \left(x_{i\,k} - \bar{x}_{i} \right)^{2}} \tag{4.14}$$

 znormalizowane względem zakresu próbkowania odchylenie standardowe dla i-tego wymiaru umożliwia porównanie jego bezwzględnej wartości między wymiarami:

$$s_i^{norm} = \frac{s_i}{samplingRange_i}$$
(4.15)

• współczynnik asymetrii dla i-tego wymiaru (ang. skewness - skośność):

$$SKE_{i} = \frac{N}{(N-1)(N-2)} \cdot \frac{\sum_{k=1}^{N} (x_{ik} - \bar{x}_{i})^{3}}{s_{i}^{3}}$$
(4.16)

Skośność jest miarą asymetrii rozkładu. Przyjmuje ona wartość 0, jeśli próbki rozkładają się równomiernie po obu stronach wartości średniej. Gdy współczynnik skośności ma wartość większą od 0, mówi się o rozkładzie prawoskośnym (dodatnioskośnym), wtedy wartości większości próbek są mniejsze niż średnia – szczyt rozkładu jest przesunięty w lewo, a po prawej stronie jest widoczny tzw. ogon. Analogicznie, gdy współczynnik skośności ma wartość mniejszą od 0, mówi się o rozkładzie lewoskośnym (ujemnoskośnym), wtedy wartości większości próbek są większe niż średnia – szczyt rozkładu jest przesunięty w prawo, a tzw. ogon jest po lewej stronie.

współczynnik spłaszczenia dla i-tego wymiaru (gr. kurtos (κυρτος) – wydęty, łac. kurtosis – kurtoza):

$$Kurt_{i} = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \cdot \frac{\sum_{k=1}^{N} (x_{ik} - \bar{x}_{i})^{4}}{s_{i}^{4}} - 3 \cdot \frac{(N-1)^{2}}{(N-2)(N-3)}$$
(4.17)

Kurtoza jest miarą koncentracji wyników wokół wartości średniej, czyli spłaszczenia rozkładu w odniesieniu do rozkładu normalnego (Gaussa). Rozkłady prawdopodobieństwa można podzielić ze względu na wartość kurtozy na rozkłady:

- mezokurtyczne (rozkład normalny, Kurt = 0). Wartość kurtozy wynosi 0. Spłaszczenie rozkładu jest podobne do spłaszczenia rozkładu normalnego,
- leptokurtyczne (rozkład wysmukły, *Kurt* > 0). Wartość kurtozy jest dodatnia.
 Próbki są bardziej skoncentrowane niż przy rozkładzie normalnym,
- platokurtyczne (rozkład spłaszczony, *Kurt <* 0). Wartość kurtozy jest ujemna.
 Próbki są mniej skoncentrowane niż przy rozkładzie normalnym.

Porównanie różnych wartości skośności oraz kurtozy w odniesieniu do rozkładu normalnego przedstawia Rys. 4.32;



Rys. 4.32 Porównanie różnych wartości skośności i kurtozy w odniesieniu do rozkładu normalnego

• kowariancja dla wymiarów i-tego i j-tego (ang. covariance):

$$s_{i,j} = \frac{1}{N-1} \sum_{k=1}^{N} \left(x_{ik} - \bar{x}_i \right) \left(x_{jk} - \bar{x}_j \right)$$
(4.18)

• korelacja dla wymiarów i-tego i j-tego (ang. correlation):

$$r_{i,j} = \frac{s_{i,j}}{s_i \cdot s_j}$$
(4.19)

Współczynnik korelacji jest miarą względnej współzależności liniowej między cechami. Jego wartość zmienia się w granicach $-1 \le r \le 1$. Im |r| jest bliższy 1, tym korelacja między dwoma wymiarami w repozytorium jest mocniejsza.

Powyższe parametry empiryczne dostarczają informacji o rozkładzie próbkowania oddzielnie dla każdego wymiaru (ang. *dimension*) oraz o zależnościach liniowych par wymiarów (macierze kowariancji i korelacji). Na tej podstawie można wnioskować o jakości repozytorium.

Rys. 4.33 - Rys. 4.35 przedstawiają wyniki etapu E dla repozytoriów grid3k, rand3k oraz ga3k w postaci wydruku z programu ScanRep.

Porównanie przedstawionych wyników dowodzi podobieństwa próbkowania przestrzeni wokół optimum przez wszystkie trzy typy próbkowania analizowane w ramach przypadku testowego. Zakresy próbkowania oraz wartości przeciętne wskazują, że punkty rozmieszczone

są po obu stronach minimum dla każdego wymiaru. Trochę niższa wartość odchylenia standardowego oraz wyższa kurtozy dla zbiorów ga3k w porównaniu z pozostałymi repozytoriami ujawnia większe skupienie punktów wokół minimum. Jednocześnie ujemy znak kurtozy dowodzi, że punkty są nadal mniej skupione niż w rozkładzie normalnym.

Basen przyciągania optimum jest lekko wydłużony w stronę wyższych wartości parametrów. Najwyraźniej pokazało to próbkowanie przestrzeni algorytmem genetycznym, o czym świadczą dodatnie wartości skośności w repozytorium ga3k (szczególnie wymiar 2). Wyniki dla zbioru rand3k są zbliżone do zera, przez co mylnie sugerują symetrię, a wartości skośności dla próbkowania równomiernego są jedynie lekko dodatnie. Macierze korelacji dla wszystkich trzech zbiorów danych testowych prawidłowo wykazały brak znaczącej korelacji między wymiarami. Nieznacznie dodatnia wartość korelacji w repozytorium ga3k potwierdza kształt basenu przyciągania optimum.

Wyniki etapu E potwierdzają, że jakość repozytorium ga3k jest porównywalna z repozytoriami referencyjnymi. Wykorzystanie próbkowania przestrzeni algorytmem genetycznym do wyznaczenia niepewności parametrów dla przypadku testowego powinno dać podobne efekty, co próbkowanie równomierne oraz Metoda Monte Carlo.

```
#GENERAL_INFO#
Number of points:
                       708
Best point value:
                        1.838
Best point parameters: -26.000 -26.000
   #SAMPLING_RANGES#
Dim
     Min
               Max
                           Ranae
               -9.000
      -40.000
                             31.000
 1
 2
       -40.000
                  -9.000
                             31.000
   #MOMENTS#
Dimension 1
                      -24.999
 Expected value:
                       7.549 (0.244 - normalized to cluster range)
 Standard deviation:
                       0.083 - RIGHT side longer
 Skewness:
                       -0.882 - LESS concentrated then normal
 Kurtosis:
Dimension 2
 Expected value:
                      -24.999
 Standard deviation:
                       7.549 (0.244 - normalized to cluster range)
 Skewness:
                        0.083 - RIGHT side longer
 Kurtosis:
                       -0.882 - LESS concentrated then normal
   #COVARIANCE_MATRIX#
   56.980
           -0.769
           56.980
   -0.769
   #CORRELATION_MATRIX#
   1.000
           -0.014
   -0.014
            1.000
```

Rys. 4.33 Repozytorium grid3k. Metoda AH: etap E. Wydruk z programu ScanRep

```
#GENERAL_INFO#
Number of points:
                        513
                         1.840
Best point value:
Best point parameters: -25.723 -25.747
    #SAMPLING_RANGES#
Dim
     Min
                Мах
                            Range
       -39.798
                   -8.650
                              31.148
  1
  2
       -39.631
                   -8.687
                              30.944
    #MOMENTS#
Dimension 1
                       -24.671
  Expected value:
  Standard deviation:
                         7.499 (0.241 - normalized to cluster range)
                         0.040 - RIGHT side longer
  Skewness:
                        -0.878 - LESS concentrated then normal
  Kurtosis:
Dimension 2
  Expected value:
                       -24.423
 Standard deviation:
                        7.633 (0.247 - normalized to cluster range)
  Skewness:
                         0.006 - RIGHT side longer
  Kurtosis:
                        -0.890 - LESS concentrated then normal
   #COVARIANCE_MATRIX#
          -0.363
   56.241
   -0.363
            58.257
    #CORRELATION_MATRIX#
    1.000
           -0.006
   -0.006
             1.000
```

Rys. 4.34 Repozytorium rand3k. Metoda AH: etap E. Wydruk z programu ScanRep

```
#GENERAL_INFO#
Number of points:
                        374
Best point value:
                          1.834
Best point parameters:
                        -25.877 -25.877
   #SAMPLING_RANGES#
Dim
     Min
                            Range
                Мах
 1
       -39.739
                  -10.750
                              28.989
 2
       -39.921
                  -9.728
                              30.193
   #MOMENTS#
Dimension 1
 Expected value:
                       -25.817
                         6.440 (0.222 - normalized to cluster range)
 Standard deviation:
 Skewness:
                         0.034 - RIGHT side longer
                        -0.593 - LESS concentrated then normal
 Kurtosis:
Dimension 2
                       -25.764
 Expected value:
 Standard deviation:
                        6.996 (0.232 - normalized to cluster range)
 Skewness:
                         0.190 - RIGHT side longer
                        -0.606 - LESS concentrated then normal
 Kurtosis:
   #COVARIANCE MATRIX#
  41.473
            2.748
   2.748
            48.946
    #CORRELATION_MATRIX#
    1.000
             0.061
   0.061
             1.000
```

Rys. 4.35 Repozytorium ga3k. Metoda AH: etap E. Wydruk z programu ScanRep

4.7. Badanie rozkładu próbkowania w oparciu o odległości między punktami w repozytorium – Histograms (F)

Szóstym etapem Metody AH (ang. *histograms* – histogramy) jest badanie rozkładu próbkowania przestrzeni optymalizacyjnej za pomocą analizy histogramicznej. W oparciu o odległości między punktami wyciągane są wnioski na temat jakości repozytorium. Etap F jest uzupełnieniem analizy statystycznej odbywającej się w etapie E. Wspólnie stanowią one moduł analizy próbkowania, dostarczając ilościowej oceny czy wyodrębnione próbkowanie otoczenia minimum jest wystarczające do oszacowania niepewności wartości parametrów.

W ramach etapu F prezentowana informacja na temat repozytorium zawiera:

- histogram próbkowania przestrzeni optymalizacyjnej (ang. Space Sampling Histogram, SSH) oddzielnie dla każdego z parametrów. Wartość argumentu określa liczbę przedziałów (ang. bin number), na które dzielony jest każdy wymiar. Rezultatem procedury SSH jest liczba punktów znajdujących się w poszczególnych przedziałach każdego wymiaru. Na tej podstawie można wnioskować, które części przestrzeni są gęsto próbkowane oraz stwierdzić czy w kluczowych obszarach (np. basen przyciągania globalnego optimum) nie występują przerwy w próbkowaniu;
- histogram odległości od najlepszego punktu w repozytorium (ang. *Distance from Best Histogram*, DBH), czyli histogram odległości punktów w repozytorium od znalezionego minimum. Wartość argumentu określa szerokość przedziałów (ang. *bin width*), na które dzielona jest wartość odległości punktów. Rezultatem procedury DBH jest liczba punktów znajdujących się w poszczególnych współśrodkowych przylegających do siebie hiper-pierścieniach¹⁹ kołowych, oddalających się od znalezionego minimum. Na tej podstawie można wnioskować czy w kolejnych przedziałach odległości liczba punktów rośnie proporcjonalnie do obszaru przestrzeni, którą zajmuje określony pierścień²⁰;
- histogram stopni wierzchołków (ang. Vertex Degree Histogram, VDH), w którym repozytorium interpretowane jest jako graf euklidesowy z zadaną jako argument maksymalną długością krawędzi (ang. max edge length). Drugi argument procedury VDH określa szerokość przedziałów, na które dzielona jest wartość stopni wierzchołków.

¹⁹ W tej pracy przyjmuje się, że hiper-pierścień kołowy (ang. *hyper-ring*) to część przestrzeni zawierająca punkty, których odległość od środka pierścienia mieści się w zadanym przedziale.

²⁰ Gdy wektor parametrów jest dwuelementowy, powierzchnia kolejnych pierścieni rośnie liniowo.

Rezultatem procedury są liczności zbiorów punktów, w których otoczeniach znajduje się określona liczba sąsiadów, czyli wielkości zbiorów wierzchołków o tym samym stopniu. Na tej podstawie można wnioskować czy otoczenia punktów (zdefiniowane przez wartość argumentu *maxEdgeLength*) zawierają podobną liczbę sąsiadów, co świadczy o równomierności próbkowania przestrzeni.

Dane do histogramów dostarczane są w postaci wydruków z programu ScanRep. Na ich podstawie, przy użyciu programu Gnuplot, tworzone są wizualizacje graficzne histogramów DBH oraz VDH. Histogram SSH prezentowany jest w postaci tekstowej, gdyż ułatwia to porównanie rozkładu próbkowania przestrzeni pomiędzy poszczególnymi wymiarami.

Rys. 4.36 - Rys. 4.44 przedstawiają wyniki etapu F repozytoriów grid3k, rand3k oraz ga3k w postaci histogramów SSH, DBH i VDH.

Dimension 1 / Dimension 2			Dimer	nsion 1	Dimension 2			Dimension 1			Dimension 2		
From	Counts	From	Counts	From	Counts	From	Counts		From	Counts	/	From	Counts
-40	3	/ -40	3	-40	5	/ -40	5		-40	3	/	-40	6
-39	10	-39	10	-39	7	-39	7		-39	5	/	-39	4
-38	<u>1</u> 4	-38	14	-38	9	-38	10		-38	7	/	-38	6
-37	17	-37	17	-37	12	-37	15		-37	11	/	-37	9
-36	20	-36	20	-36	19	-36	18		-36	9	/	-36	12
-35	22	-35	22	-35	16	-35	12		-35	9	/	-35	16
-34	24	-34	24	-34	17	-34	19		-34	15	/	-34	14
-33	25	-33	25	-33	11	-33	16		-33	14	/	-33	11
-32	27	-32	27	-32	23	-32	15		-32	9	/	- <i>32</i>	12
-31	28	-31	28	-31	17	-31	20		-31	14	/	-31	18
-30	29	-30	29	-30	24	-30	18		-30	20	/	-30	13
-29	30	-29	30	-29	22	-29	18		-29	20	/	-29	23
-28	31	-28	31	-28	21	-28	20		-28	22	/	-28	26
-27	32	-27	32	-27	27	-27	23		-27	24	/	-27	16
-26	32	-26	32	-26	22	-26	28		-26	28	/	-26	21
-25	32	/ -25	32	-25	23	/ -25	21		-25	27	/	-25	21
-24	31	-24	31	-24	23	-24	20		-24	16	/	-24	24
-23	30	-23	30	-23	25	-23	26		-23	12	/	-23	16
-22	30	-22	30	-22	21	-22	25		-22	14	/	-22	12
-21	29	-21	29	-21	16	-21	21		-21	21	/	-21	14
-20	27	/ -20	27	-20	20	/ -20	21		-20	17	/	-20	8
-19	26	-19	26	-19	20	-19	17		-19	7	/	-19	12
-18	24	-18	24	-18	18	-18	14		-18	12	/	-18	13
-17	23	-17	23	-17	19	-17	24		-17	12	/	-17	7
-16	22	/ -16	22	-16	16	-16	18		-16	7	/	-16	9
-15	20	/ -15	20	-15	14	/ -15	11		-15	8	/	-15	8
-14	18	-14	18	-14	12	-14	12		-14	2	/	-14	7
-13	15	-13	15	-13	11	-13	12		-13	4	/	-13	4
-12	13	-12	13	-12	10	-12	11		-12	4	/	-12	5
-11	11	-11	11	-11	6	/ -11	7		-11	1	/	-11	6
-10	8	/ -10	8	-10	5	/ -10	5		-10	0	/	-10	1
-9	5	/ -9	5	-9	2	/ -9	4		-9	0	/	-9	0
-8	0	/ -8	0	-8	0	/ -8	0		-8	0	/	-8	0

Rys. 4.36 Repozytorium grid3k.

Rys. 4.37 Repozytorium rand3k.

Rys. 4.38 Repozytorium ga3k.

Metoda AH: etap F. Histogram SSH | Metoda AH: etap F. Histogram SSH | Metoda AH: etap F. Histogram SSH

Analiza histogramów SSH dowodzi, że basen przyciągania optimum globalnego jest gęsto próbkowany. Jednocześnie brak jest przerw w próbkowaniu otoczenia minimum dla wszystkich trzech zbiorów testowych.

Porównanie histogramów DBH pokazuje, że w repozytoriach grid3k i rand3k liczba punktów w kolejnych pierścieniach wokół optimum rośnie liniowo od minimum do odległości 13. Dla repozytorium ga3k liniowy wzrost obserwowany jest jedynie blisko minimum. Wskazuje to, że w tym zbiorze danych na obrzeżach basenu przyciągania optimum zmniejsza się gęstość próbkowania przestrzeni.

Histogramy VDH potwierdzają, że najbardziej równomierne próbkowanie przestrzeni zawiera repozytorium grid3k, ponieważ 512 z 708 punktów w repozytorium ma dwunastu sąsiadów²¹. Dla wartości *maxEdgeLength*=2.0 otoczenia większości punktów w zbiorze rand3k zawierają od sześciu do jedenastu sąsiadów. Najbardziej zróżnicowane z punktu widzenia liczby sąsiadów są otoczenia punktów w repozytorium ga3k.

Z połączonych informacji przedstawionych w etapach E oraz F wynika, że wszystkie trzy typy próbkowania przestrzeni nadają się do wyznaczenia niepewności parametrów w rozważanym przypadku testowym. Jednocześnie ilościowe oszacowanie jakości repozytorium wskazuje, że dla zbiorów grid3k i rand3k oszacowane przedziały niepewności będą najbardziej wiarygodne, jeśli szukany kontur będzie w odległości nie większej niż 13 od optimum. Dla repozytorium ga3k potencjalna niedokładność estymacji niepewności może być nieznacznie większa.

Podstawowym zadaniem modułu analizy próbkowania jest kompleksowa ocena jakości repozytorium i możliwości oszacowania niepewności wartości parametrów za jego pomocą. Jednak informacja przez niego dostarczona może być również wykorzystana do optymalnego doboru wartości argumentu procedury MDR (etap C) oraz właściwej klasyfikacji punktów należących do basenu przyciągania optimum (etap D).

Jeśli jakość przetwarzanego przez Metodę AH repozytorium zostanie pozytywnie zweryfikowana przez moduł analizy próbkowania, to zbiór punktów jest następnie wykorzystywany do wyznaczenia niepewności wartości parametrów. Opis modułu wyznaczania niepewności wraz z teoretycznym uzasadnieniem poprawności stosowania oraz prezentacja programu ScanRep implementującego Metodę AH są zawarte w rozdziale 5.

²¹ Na Rys. 4.42 nie zmieniono skali na osi pionowej, aby ułatwić porównanie między trzema histogramami VDH.



Rys. 4.39 Repozytorium grid3k. Metoda AH: etap F. Histogram DBH



Rys. 4.40 Repozytorium rand3k. Metoda AH: etap F. Histogram DBH



Rys. 4.41 Repozytorium ga3k. Metoda AH: etap F. Histogram DBH



Rys. 4.42 Repozytorium grid3k. Metoda AH: etap F. Histogram VDH



Rys. 4.43 Repozytorium rand3k. Metoda AH: etap F. Histogram VDH



Rys. 4.44 Repozytorium ga3k. Metoda AH: etap F. Histogram VDH

5. Metoda AH – estymacja niepewności parametrów z wykorzystaniem próbkowania przestrzeni algorytmem genetycznym

Estymacja niepewności parametrów dla zadanej płaszczyzny progowej Metodą AH odbywa się w module wyznaczenia niepewności. Wobec przetwarzanego w ramach etapów G i H repozytorium zakłada się, że zostało ono pozytywnie zweryfikowane przez moduł analizy próbkowania. Rozpatrywany zbiór punktów znajduje się w basenie przyciągania optimum. Jednocześnie próbkowanie zawiera wystarczająco precyzyjną informację o kształcie funkcji wokół całego otoczenia minimum.

W punkcie 5.1. przedstawiono teoretyczne uzasadnienie stosowania Metody AH do szacowania niepewności wyznaczenia wartości elementów macierzowych za pomocą analizy danych z pomiarów wzbudzeń kulombowskich. Napisałem je przede wszystkim w oparciu o:

- czwarte wydanie książki prof. Siegmunda Brandta "Data Analysis. Statistical and Computational Methods for Scientists and Engineers" (punkty 5.10, 9.5, 9.7 i 9.8) [108];
- manual do programu GOSIA autorstwa dra hab. Tomasza Czosnyki (punkty 6.7 i 6.8) [8];
- artykuł prof. Jacka Dobaczewskiego "Error Estimates of Theoretical Models: a Guide" [109].

Korzystałem również z następującej literatury: [104] – [107], [110] – [127]. W tym punkcie zawarłem też krótkie rozważania na temat wskazywania właściwej płaszczyzny progowej do wyznaczenia niepewności dla poszczególnych danych eksperymentalnych.

W punkcie 5.2. opisano (składający się na etap G) autorski algorytm FLA [129]. Spośród repozytorium próbkowania basenu przyciągania optimum wyodrębnia punkty leżące najbliżej płaszczyzny progowej, dla której wyznaczane są niepewności parametrów. Ten zbiór w największym stopniu zawiera informację o kształcie szukanego konturu oraz umożliwia wnioskowanie na temat dokładności oszacowania niepewności parametrów w oparciu o dane repozytorium próbkowania przestrzeni.

W punkcie 5.3. omówiono etap H Metody AH, który dostarcza kompleksowej informacji na temat przedziałów niepewności wartości poszczególnych parametrów oraz o dokładności oszacowania samej niepewności pomiarowej.

W punkcie 5.4. zaprezentowano program ScanRep, który stanowi implementację Metody AH.

Punkt 5.5. zawiera schemat Metody AH oraz podsumowanie jej działania.

5.1. Geometryczna interpretacja niepewności parametrów

Na podstawie danych obserwowanych bezpośrednio w pomiarze wzbudzeń kulombowskich (rozproszone cząstki wraz z kwantami promieniowania gamma), geometrii układu pomiarowego oraz zakładanej struktury stanów elektromagnetycznych badanego jądra wyznacza się wielkości mierzalne:

- intensywności przejść pomiędzy stanami wzbudzonymi;
- czasy życia stanów wzbudzonych;
- stosunki rozgałęzień;
- współczynniki zmieszania.

Wyznaczanie wartości elementów macierzowych za pomocą analizy danych z pomiarów wzbudzeń kulombowskich odbywa się z wykorzystaniem metody najmniejszych kwadratów. Jest to przypadek pośredniego wyznaczania szukanych wartości parametrów. Związek pomiędzy wielkościami mierzonymi, a nieznanym wektorem parametrów jest nieliniowy i opiera się na modelach fizyki teoretycznej.

Każdy pomiar wielkości mierzonych obarczony jest błędem. Zakłada się, że błędy pomiarowe podlegają rozkładowi normalnemu z zerową wartością oczekiwaną. Hipoteza ta jest uzasadniona dzięki centralnemu twierdzeniu granicznemu.

5.1.1. Statystyka najmniejszych kwadratów

Związek pomiędzy wielkościami mierzonymi i wektorem parametrów określa się, przyjmując za wartość prawdziwą wielkości mierzonej, jej wartość teoretyczną wyznaczoną przy pomocy przyjętego modelu fizyki teoretycznej:

$$\boldsymbol{\eta}_j = \boldsymbol{h}_j(\boldsymbol{x}) \quad dla \quad j = 1 \dots N_{data} \tag{5.1}$$

gdzie:

 $h_i(\cdot)$ – zależność funkcyjna wynikająca z fizyki teoretycznej,

x – wektor parametrów,

 η_i – wartość prawdziwa (teoretyczna) wielkości mierzonej.

Związek pomiędzy zmierzoną wartością j-tej wielkości mierzonej, a jej wartością prawdziwą określa się następująco:

$$y_j = \eta_j + \epsilon_j$$
, $E(\epsilon_j) = 0$, $E(\epsilon_j^2) = \sigma_j^2 = \frac{1}{g_j}$ (5.2)

gdzie:

 y_i – wielkość mierzona²²,

 ϵ_i – nieznany błąd pomiarowy podlegający rozkładowi Gaussa,

 σ_i^2 – wariancja błędu pomiarowego,

 g_i – waga wielkości mierzonej.

Ponieważ y_j są zmiennymi niezależnymi, można przedstawić wariancje σ_j^2 w postaci diagonalnej macierzy kowariancji zmiennych y_j lub ϵ_j :

$$C_{y} = C_{\epsilon} = \begin{bmatrix} \sigma_{1}^{2} & 0 \\ \sigma_{2}^{2} & \\ & \ddots & \\ 0 & \sigma_{N_{data}}^{2} \end{bmatrix}$$
(5.3)

gdzie C_{v} , C_{ϵ} jest macierzą kowariancji wielkości mierzonych.

Macierz odwrotna (tj. macierz wag wielkości mierzonych) ma następującą postać:

$$G_{y} = G_{\epsilon} = C_{y}^{-1} = C_{\epsilon}^{-1} = \begin{bmatrix} g_{1} & & 0 \\ & g_{2} & \\ & & \ddots & \\ 0 & & & g_{N_{data}} \end{bmatrix}$$
(5.4)

Zgodnie z metodą największej wiarygodności, wektor parametrów x najlepiej odpowiada danym doświadczalnym, gdy statystyka najmniejszych kwadratów przyjmuje wartość minimalną:

$$M(\mathbf{x}) = \sum_{j=1}^{N_{data}} \frac{\epsilon_j^2}{\sigma_j^2} = \sum_{j=1}^{N_{data}} \frac{(y_j - \eta_j)^2}{\sigma_j^2} = \sum_{j=1}^{N_{data}} \frac{(y_j - h_j(\mathbf{x}))^2}{\sigma_j^2} = min$$
(5.5)

lub dla wektora wag:

$$M(\mathbf{x}) = \sum_{j=1}^{N_{data}} g_{j} \cdot \epsilon_{j}^{2} = \sum_{j=1}^{N_{data}} g_{j} \cdot (y_{j} - \eta_{j})^{2} = \sum_{j=1}^{N_{data}} g_{j} \cdot (y_{j} - h_{j}(\mathbf{x}))^{2} = min$$
(5.6)

gdzie $M(\mathbf{x})$ jest statystyką najmniejszych kwadratów.

Zapis macierzowy powyższych zależności ma postać:

²² Dla uproszczenia zapisu zmienne losowe nie będą wyróżniane specjalną czcionką. Z kontekstu zawsze będzie wynikać, które zmienne są losowymi.

$$M(\mathbf{x}) = \mathbf{\epsilon}^{T} G_{\mathbf{\epsilon}} \mathbf{\epsilon} = \min$$

$$\mathbf{\epsilon} = \mathbf{y} - \mathbf{\eta} = \mathbf{y} - \mathbf{h}(\mathbf{x})$$

$$G_{\mathbf{\epsilon}} = G_{y}$$
(5.7)

a więc ostatecznie:

$$M(\mathbf{x}) = (\mathbf{y} - \mathbf{h}(\mathbf{x}))^{T} G_{y} (\mathbf{y} - \mathbf{h}(\mathbf{x})) = min$$
(5.8)

Zależność (5.8) definiuje zadanie optymalizacyjne²³. W procesie optymalizacji poszukiwany jest taki wektor parametrów, który z największą dokładnością odwzorowuje dane z eksperymentu. Wartość $M(\mathbf{x})$ stanowi miarę odstępstwa danych eksperymentalnych od wyliczeń teoretycznych.

Porównanie wzorów (5.5) i (5.6) ze wzorem (2.5) pokazuje, że statystyka najmniejszych kwadratów stosowana w analizie danych z pomiarów wzbudzeń kulombowskich została sformułowana w taki sposób, aby uwzględniać specyfikę analizy danych z tych eksperymentów. W szczególności:

- umożliwia normalizację wyliczeń teoretycznych do danych doświadczalnych;
- uwzględnia ograniczony zakres danych możliwych do zmierzenia w konkretnym eksperymencie;
- umożliwia określenie dodatkowych wag dla poszczególnych podzbiorów danych zdefiniowanych przez osobę prowadzącą analizę danych;
- bierze pod uwagę wielkości zmierzone we wcześniejszych eksperymentach;
- normalizuje wartość statystyki względem liczby wielkości mierzonych.

Mimo skomplikowanej postaci zasada działania jest jednak taka sama. Statystyka $\chi^2(ME)$ jest funkcją celu dla zadania optymalizacyjnego.

5.1.2. Niepewności wyznaczenia wartości parametrów

Określenie niepewności wyznaczenia wektora parametrów metodą najmniejszych kwadratów, gdy związek między wielkościami mierzonymi, a wektorem parametrów jest liniowy, sprowadza się do wyznaczenia macierzy kowariancji dla punktu x_{min} na podstawie zależności:

$$C_{\boldsymbol{x}_{\min}} = G_{\boldsymbol{x}_{\min}}^{-1} = \left(A_{linear}^{T}G_{y}A_{linear}\right)^{-1}$$
(5.9)

gdzie A_{linear} jest macierzą współczynników zależności liniowej.

²³ W ogólności macierz wag wielkości mierzonych G_y , (oraz ich macierz kowariancji C_y) nie musi być diagonalna, jednak wtedy jej wyznaczenie w procesie analizy danych eksperymentalnych jest dużo trudniejsze.

Jeśli w bezpośrednim otoczeniu x_{min} związek pomiędzy wielkościami mierzonymi, a wektorem parametrów jest zbliżony do liniowego, można z dobrym rezultatem stosować powyższą zależność jako metodę przybliżoną. Jest to tożsame z założeniem możliwości aproksymowania kształtu funkcji celu w basenie przyciągania optimum globalnego za pomocą wielomianu drugiego stopnia.

W analizie danych z pomiarów wzbudzeń kulombowskich takie podejście jest niemożliwe. Zastosowanie modelowania wynikającego z fizyki teoretycznej do odwzorowania związku pomiędzy wielkościami mierzonymi, a wektorem parametrów skutkuje bardziej złożonym charakterem zależności między nimi. Ponadto zarówno z punktu widzenia fizyki teoretycznej, jak i praktyki doświadczalnej konieczne jest wprowadzenie do procesu analizy danych tzw. uciążliwych parametrów (ang. *nuisance parameters*), które są istotne z punktu widzenia poprawności formalnej, jednak nie mają bezpośredniego (lub istotnego) wpływu na wartość statystyki $\chi^2(ME)$. To uniemożliwia analityczne wyznaczenie macierzy kowariancji $C_{x_{min}}$.

Statystyka najmniejszych kwadratów jest implementacją metody największej wiarygodności przy założeniu, że błędy pomiarowe ϵ wielkości mierzonych *y* podlegają rozkładowi normalnemu²⁴. Tak jest w przypadku pomiarów wzbudzeń kulombowskich.

Znalezione za pomocą minimalizacji wartości statystyki $M(\mathbf{x})$ rozwiązanie \mathbf{x}_{min} ma następujące cechy:

- jest asymptotycznie nieobciążone (dla dużej liczby próbek);
- stanowi estymator o najmniejszej wariancji.

Ponadto wielkość $M(\mathbf{x})$ podlega rozkładowi χ^2 o następującej liczbie stopni swobody:

 $v_{effective} = N_{data} - n_{effective}$ (5.10)

gdzie:

 $v_{effective}$ – faktyczna (efektywna) liczba stopni swobody,

 $n_{effective}$ – liczba parametrów pomniejszona o parametry z minimalnym lub zerowym wpływem na wartość $M(\mathbf{x})$ (*nuisance parameters*) oraz po uwzględnieniu korelacji między pozostałymi parametrami. Wartość $n_{effective}$ nie musi być liczbą naturalną, a jedynie nieujemną.

²⁴ Twierdzenie Gaussa-Markowa częściowo rozszerza możliwość wnioskowania na podstawie statystyki najmniejszych kwadratów nawet w przypadku, gdy błędy pomiarowe nie podlegają rozkładowi normalnemu.

Ze względu na tę ostatnią właściwość statystyka najmniejszych kwadratów wykorzystywana w analizie danych z pomiarów wzbudzeń kulombowskich została oznaczona symbolem $\chi^2(ME)$.

Niepewność wyznaczenia wartości parametrów, dla zadanego poziomu ufności (ang. *confidence level*), zarówno w przypadku liniowym, jak i nieliniowym metody najmniejszych kwadratów, można określić na podstawie analizy konturu, na który składają się takie punkty x, że:

$$M(\mathbf{x}) = M(\mathbf{x}_{\min}) + q \quad , \quad q = \chi_W^2(\mathbf{v}_{effective}) \tag{5.11}$$

gdzie $\chi^2_W(v_{effective})$ jest kwantylem rozkładu χ^2 o liczbie stopni swobody $v_{effective}$ dla poziomu ufności W.

W przestrzeni parametrów interpretacją geometryczną wzoru (5.11) jest kontur powstały z przecięcia powierzchni funkcji $M(\mathbf{x})$ płaszczyzną poziomą na wysokości $M(\mathbf{x}_{min})+q$. Jako niepewności wyznaczenia wartości parametrów dla zadanego poziomu ufności przyjmuje się skrajne wartości dla poszczególnych wymiarów należące do konturu. Jest to tożsame ze współrzędnymi wierzchołków hiper-prostokąta opisanego na tym konturze. W Metodzie AH powyższą płaszczyznę poziomą nazywa się płaszczyzną progową do wyznaczenia niepewności parametrów i oznacza y_{PU} .

W przypadku liniowej zależności między wielkościami mierzonymi, a wektorem parametrów oraz przy standardowym poziomie ufności, kontur pokrywa się z elipsoidą kowariancji (ang. *covariance ellipsoid*), którą można określić przy pomocy wzoru (5.9). Dla innych poziomów ufności, kontur stanowi inne elipsoidy ufności (ang. *confidence ellipsoid*) leżące wewnątrz albo na zewnątrz elipsoidy kowariancji.

Gdy związek ten jest nieliniowy, szukany kontur znajduje się na zewnątrz elipsoidy o danym poziomie ufności. Oznacza to, że zastosowanie metody estymacji niepewności opartej na macierzy kowariancji skutkowałoby niedoszacowaniem niepewności wartości parametru. Ponadto wyznaczone przy pomocy wzoru (5.11) granice ufności (ang. *confidence boundaries*) są zazwyczaj asymetryczne względem punktu x_{min} . Jedyne co można powiedzieć o kształcie konturu definiującego obszar ufności (ang. *confidence region*), to że jest on wypukły. Wynika to z charakteru statystyki najmniejszych kwadratów.

Wartość statystyki $M(\mathbf{x})$ może zostać znormalizowana do liczby stopni swobody. Dla takiej zredukowanej statystyki najmniejszych kwadratów niepewność wyznaczenia wartości parametru można określić przy pomocy:



Rys. 5.1 Zbieżność kwantylu znormalizowanej do liczby stopni swobody statystyki najmniejszych kwadratów w funkcji liczby stopni swobody dla standardowego poziomu ufności

$$M_{reduced}(\mathbf{x}) = \frac{M(\mathbf{x})}{\mathbf{v}_{effective}} = \frac{M(\mathbf{x}_{min}) + q}{\mathbf{v}_{effective}} = M_{reduced}(\mathbf{x}_{min}) + q_{reduced}$$
(5.12)

gdzie $M_{reduced}(\cdot)$ jest znormalizowaną do liczby stopni swobody statystyką najmniejszych kwadratów. Wartość $q_{reduced}$ w funkcji liczby stopni swobody jest szybko zbieżna do jedności dla każdego poziomu ufności 0 < W < 1. Rys. 5.1 przedstawia jej zbieżność dla standardowego poziomu ufności.

Ze względu na dużą liczbę stopni swobody, w analizie danych z pomiarów wzbudzeń kulombowskich przyjmuje się:

$$M_{reduced}(\boldsymbol{x}) = M_{reduced}(\boldsymbol{x}_{min}) + 1$$
(5.13)

5.1.3. Liczba stopni swobody

Precyzyjne określenie faktycznej liczby stopni swobody $v_{effective}$ w analizie danych z pomiarów wzbudzeń kulombowskich jest trudne. Wpływa na to zarówno obecność uciążliwych parametrów (ang. *nuisance parameters*), które mają zaniedbywalnie mały wpływ na wartość statystyki najmniejszych kwadratów, jak i istnienie korelacji między parametrami, których wartości nie da się a priori oszacować.

Z tego powodu stosowana statystyka najmniejszych kwadratów jest normalizowana do liczby wielkości mierzonych N_{data} zamiast do liczby stopni swobody $v_{effective}$ (vide wzór 5.10), co stanowi dobre przybliżenie dopóki $n_{effective} \ll N_{data}$. Istotne jest jednak przeanalizowanie, jaki ma ono wpływ na szacowanie niepewności wartości wektora parametrów.

Z racji tego iż $v_{effective} < N_{data}$ to stała używana do normalizacji statystyki najmniejszych kwadratów w analizie danych z pomiarów wzbudzeń kulombowskich jest zbyt duża, a wartości tak zredukowanej statystyki zbyt małe. Wyznaczenie niepewności wartości parametrów dla $y_{PU} = y_{min} + 1.0$ skutkuje jej przeszacowaniem, co oznacza podniesienie poziomu ufności. Aby oszacować niepewności dla standardowego poziomu ufności należałoby przyjąć wartość y_{PU} trochę mniejszą od jedności względem minimum. Z drugiej strony prawdziwa wartość $q_{reduced}$, dla skończonej liczby stopni swobody i standardowego poziomu ufności, jest nieznacznie większa od jedności. Omawiane zjawiska są przeciwstawne i częściowo się znoszą.

Wartość kwantylu $q'_{reduced}$, dla niedokładnie znormalizowanej statystyki najmniejszych kwadratów i określonego poziomu ufności, można wyznaczyć następująco:

$$q'_{reduced} = \frac{\mathbf{v}_{effective}}{N_{data}} \cdot q_{reduced}$$
(5.13)

Gdy wartość ilorazu jest bliska jedności, to wartość $q'_{reduced}$ jest bliższa jedności niż wartość kwantylu rozkładu χ^2 dla liczby stopni swobody równej ilorazowi.

Rys. 5.2 przedstawia wartość kwantylu rozkładu χ^2 w funkcji liczby stopni swobody (zbliżonych do jedności) dla poziomów ufności: standardowego oraz 95%. Rys. 5.3 pokazuje jak zmienia się poziom ufności w funkcji liczby stopni swobody (zbliżonych do jedności) dla kilku przykładowych wartości kwantylu rozkładu χ^2 . Z analizy rysunków wynika, że potencjalna niedokładność wyznaczenia niepewności za pomocą jej interpretacji geometrycznej jest nieznaczna.

Przedstawiony problem stanowi obszar potencjalnej niedokładności wyznaczenia niepewności parametrów w analizie danych z pomiarów wzbudzeń kulombowskich za pomocą jej interpretacji geometrycznej. Niedokładność można zmniejszyć szacując wartość $n_{effective}$ w danym eksperymencie i odejmując tę estymatę od N_{data} przy normalizacji statystyki najmniejszych kwadratów. Jest to jednak zagadnienie z pogranicza doświadczalnej fizyki jądrowej oraz statystyki matematycznej i nie będzie tu szerzej dyskutowane.


Rys. 5.2 Wartość kwantylu rozkładu χ^2 *w funkcji liczby stopni swobody dla różnych poziomów ufności*



Rys. 5.3 Poziom ufności w funkcji liczby stopni swobody dla różnych wartości kwantylu rozkładu χ^2

109

5.2. Wyodrębnienie punktów najbliższych płaszczyźnie progowej – Front-Line Algorithm (G)

Siódmym etapem Metody AH (ang. *Front-Line Algorithm*, FLA – algorytm linii frontu) jest wyodrębnienie punktów znajdujących się najbliżej płaszczyzny progowej, dla której mają być wyznaczone niepewności parametrów. Ten zbiór punktów zawiera najwięcej informacji o kształcie konturu powstałego z przecięcia powierzchni funkcji $\chi^2(ME)$ z płaszczyzną poziomą określoną przez wartość argumentu y_{PU} . Wyodrębnione punkty znajdują się w części przestrzeni bezpośrednio otaczającej szukany kontur.

Etapy G oraz H składają się na moduł wyznaczania niepewności. Ich zadaniem jest estymacja przedziałów niepewności wartości parametrów oraz dostarczenie informacji na temat dokładności wyznaczenia niepewności w oparciu o dane repozytorium próbkowania przestrzeni.

W ramach algorytmu FLA stosowane są następujące pojęcia:

- FrontLine szukany kontur (ang. *frontline* linia frontu);
- p_{BEST} punkt z najmniejszą wartością $f(\mathbf{x})$;
- Allies, A zbiór punktów należących do basenu przyciągania optimum, dla których f(x) ≤ y_{PU} (ang. ally – sojusznik);
- Enemies, $E zbiór punktów należących do basenu przyciągania optimum, dla których <math>f(\mathbf{x}) > y_{PU}$ (ang. *enemy* przeciwnik);
- Whole, Ω zbiór wszystkich punktów należących do basenu przyciągania optimum (ang. *whole* całość);
 A ∪ E = Ω ; A ∩ E = Ø
- AllySoldiers, AS zbiór punktów należących do zbioru A, z których każdy jest najbliższym takim punktem dla przynajmniej jednego punktu ze zbioru E.
 Zbiór ten jest uzupełniany o punkty skrajne ze zbioru A (wartość minimalna albo maksymalna dla któregoś z parametrów);

 $a_{s} \in AS \Leftrightarrow a_{s} \text{ jest skrajny } wA \lor \exists e \in E \forall a \in A \text{ dist}(a_{s}, e) \leq \text{dist}(a, e)$

EnemySoldiers, ES – zbiór punktów należących do zbioru E, z których każdy jest najbliższym takim punktem dla przynajmniej jednego punktu ze zbioru A;
 e_s∈ES ⇔ ∃a∈A ∀e∈E dist(e_s, a)≤dist(e, a)

• Soldiers, *S* – suma zbiorów *AS* i *ES* (ang. *soldier* – żołnierz);

 $AS \subset A \quad ; \quad AS \cup ES = S \\ ES \subset E \quad ; \quad AS \cap ES = \emptyset$

AllySoldierFriends, ASF – zbiór punktów należących do zbioru A i nienależących do zbioru AS, z których każdy jest najbliższym takim punktem dla przynajmniej jednego punktu ze zbioru AS;

 $a_{SF} \in ASF \Leftrightarrow \exists a_{S} \in AS \quad \forall a \in A \setminus AS \quad dist(a_{SF}, a_{S}) \leq dist(a, a_{S})$

EnemySoldierFriends, ESF – zbiór punktów należących do zbioru E i nienależących do zbioru ES, z których każdy jest najbliższym takim punktem dla przynajmniej jednego punktu ze zbioru ES;

$$e_{SF} \in ESF \Leftrightarrow \exists e_{S} \in ES \quad \forall e \in E \setminus ES \quad dist(e_{SF}, e_{S}) \leq dist(e, e_{S})$$

• SoldierFriends, SF – suma zbiorów ASF i ESF (ang. friend – przyjaciel);

- AllyFrontLiners, AFL suma zbiorów AS i ASF; $AS \cup ASF = AFL$; $AS \cap ASF = \emptyset$
- EnemyFrontLiners, EFL suma zbiorów ES i ESF; $ES \cup ESF = EFL$; $ES \cap ESF = \emptyset$
- FrontLiners, FL suma zbiorów AFL i EFL (ang. frontliner frontowiec); $AFL \cup EFL = FL$; $AFL \cap EFL = \emptyset$
- AllyCivilians, AC zbiór punktów należących do zbioru A i nienależących do zbioru AFL;
 AC = A \ AFL
- EnemyCivilians, EC zbiór punktów należących do zbioru E i nienależących do zbioru EFL;

$$EC = E \setminus EFL$$

- Civilians, C suma zbiorów AC i EC (ang. *civilian* cywil); $AC \cup EC = C$; $AC \cap EC = \emptyset$
- furthestAllyFrontLiner, $furthest_{AFL}$ najmniejsza wartość $f(\mathbf{x})$ punktu ze zbioru AFL; $furthest_{AFL} = \min_{\mathbf{x} \in AFL} (f(\mathbf{x}))$

- spreadOfAllyFrontLiners, spread_{AFL} różnica pomiędzy furthest_{AFL} i y_{PU} ; spread_{AFL} = furthest_{AFL} – y_{PU}
- furthestEnemyFrontLiner, $furthest_{EFL}$ największa wartość $f(\mathbf{x})$ punktu ze zbioru EFL; $furthest_{EFL} = \max_{\mathbf{x} \in EFL} (f(\mathbf{x}))$
- spreadOfEnemyFrontLiners, spread_{EFL} różnica pomiędzy furthest_{EFL} i y_{PU} ; spread_{EFL} = furthest_{EFL} – y_{PU}
- spreadOfFrontLiners, spread_{FL} różnica pomiędzy spread_{EFL} i spread_{AFL}; spread_{FL} = spread_{EFL} – spread_{AFL}

Na Rys. 5.4 przedstawiono zależności między powyższymi pojęciami.

nies	EnemyCivilians		
Ener	EnemyFrontLiners { EnemySoldierFriends EnemySoldiers	sprea	
	frontline	- 	l e
ies	AllyFrontLiners AllySoldiers AllySoldierFriends	AFL	F J furthestAFL
ALI	AllyCivilians		

Rys. 5.4 Podział próbkowania basenu przyciągania optimum w ramach algorytmu FLA

Algorytm FLA przyjmuje na wejściu:

- repozytorium punktów stanowiących próbkowanie przestrzeni optymalizacyjnej, należących do basenu przyciągania optimum;
- wartość argumentu y_{PU} , który określa płaszczyznę progową oraz pośrednio położenie szukanego konturu będącego przecięciem powierzchni funkcji z samą płaszczyzną poziomą.

Rezultatem działania algorytmu FLA jest:

- przydzielenie każdemu punktowi z repozytorium etykiety, określającej do których zdefiniowanych powyżej zbiorów należy;
- wyznaczenie wartości opisanych powyżej współczynników.

Schemat działania algorytmu FLA jest następujący:

1.	Utwórz wektor etykiet o długości równej liczbie punktów w repozytorium.
2.	Na podstawie zadanej wartości argumentu y $_{\text{PU}}$ oraz f($\boldsymbol{x})$ poszczególnych
	punktów, podziel punkty na zbiory A oraz E. Punkt z najmniejszą
	wartością f (\mathbf{x}) oznacz etykietą BEST.
3.	Na podstawie odległości pomiędzy punktami ze zbiorów A oraz E znajdź
	i oznacz punkty należące do zbiorów AS oraz ES. Do zbioru AS dodaj
	(jeśli potrzebne) punkty skrajne ze zbioru A.
4.	Punkty, które należą do zbioru A i nie należą do zbioru AS, przydziel do
	zbioru AC. Punkty, które należą do zbioru E i nie należą do zbioru ES,
	przydziel do zbioru EC.
5.	Na podstawie odległości między punktami ze zbiorów AC oraz AS znajdź i oznacz
	punkty należące do zbioru ASF. Ze zbioru AC usuń punkty zaliczone do zbioru ASF.
6.	Na podstawie odległości między punktami ze zbiorów EC oraz ES znajdź i oznacz
	punkty należące do zbioru ESF. Ze zbioru EC usuń punkty zaliczone do zbioru ESF.
7.	Zbiory: Ω, S, SF, FL, C uznaj za sumy odpowiednich zbiorów.
8.	Wyznacz wartości współczynników: furthest_{AFL}, furthest_{EFL}, spread_{AFL},
	$spread_{EFL}$ oraz $spread_{FL}$.

Rys. 5.5 Pseudokod algorytmu FLA

Jak wynika z Rys. 5.5, działanie algorytmu FLA w dużej części opiera się na odległościach między punktami w repozytorium. Z tego powodu, aby przyspieszyć działanie algorytmu zaleca się wyznaczenie a priori macierzy odległości.

Punkty należące do zbioru *FL* znajdują się w bezpośredniej bliskości szukanego konturu i jako takie zawierają najpełniejszą informację o jego kształcie. Wartość współczynnika *spread*_{FL} opisuje ich stopień skupienia wokół konturu.

Rys. 5.6 – Rys. 5.17 przedstawiają wizualizacje etapu G repozytoriów grid3k, rand3k oraz ga3k w czterech wariantach:

- 3D, 2D wyróżniono wszystkie punkty ze zbioru FL (S oraz SF);
- 3D noSF, 2D noSF wyróżniono jedynie punkty ze zbioru S (bez zbioru SF).

Z analizy rysunków wynika jak istotne jest utworzenie zbioru FL jako sumy zbiorów S oraz SF. Dodanie punktów zbioru SF znacząco redukuje obszary, w których potencjalnie brakuje punktów ze zbioru FL wokół szukanego konturu. Efekt ten jest widoczny dla repozytoriów z nierównym próbkowaniem: rand3k oraz ga3k.













5.3. Estymacja niepewności parametrów – Parameter Uncertainties (H)

Ostatnim etapem Metody AH (ang. *parameter uncertainties* – niepewności parametrów) jest estymacja niepewności wyznaczonych wartości poszczególnych parametrów dla zadanej jako argument wartości y_{PU} oraz dostarczenie informacji o dokładności jej wyznaczenia w oparciu o dane repozytorium próbkowania przestrzeni.

W ramach etapu H przyjmuje się:

- niepewności wyznaczenia wektora parametrów przedziały określone dla poszczególnych wymiarów przez wartości min-max wśród punktów, których $f(\mathbf{x}) \leq y_{PU}$, to jest punktów skrajnych w zbiorze AS;
- informację o dokładności wyznaczenia niepewności wartość współczynnika spread_{FL}, który został zdefiniowany w etapie G oraz odległości najdalszych punktów ze zbiorów *EFL* i *AFL* od szukanego konturu.

Niedokładność wyznaczenia niepewności objawia się jej niedoszacowaniem. Punkty z wartościami min-max dla danego parametru znajdują się bliżej punktu p_{BEST} niż szukany kontur.

Współczynnik *spread*_{FL} stanowi dolne oszacowanie dokładności. Im jego wartość jest mniejsza, tym precyzja estymacji niepewności większa. Twierdzenie odwrotne nie jest uzasadnione. Wyznaczenie niepewności parametrów może być precyzyjne, mimo że wartość współczynnika *spread*_{FL} jest wysoka.

Pośredni wpływ na dokładność wyznaczenia niepewności ma określona w etapie C wartość minimalnej odległości między punktami. Niedokładność wyznaczenia niepewności może sięgnąć podwojonej wartości argumentu procedury MDR.

Niedokładność wyznaczenia niepewności Metodą AH jest zjawiskiem przeciwstawnym do jej przeszacowywania ze względu na przesadną redukcję statystyki najmniejszych kwadratów w analizie danych z pomiarów wzbudzeń kulombowskich. Razem z omawianym w podpunkcie 5.1.3 niedoszacowywaniem wartości kwantylu rozkładu χ^2 dla skończonej liczby stopni swobody, te trzy zjawiska w dużym stopniu się znoszą.

5.3.1. Dalsza estymacja niepewności

Aby zmniejszyć potencjalną niedokładność wyznaczenia niepewności, zaproponowałem procedurę jej dalszej estymacji (ang. *Further Uncertainty Estimation*, FUE) w oparciu o informację na temat kształtu funkcji zawartą w repozytorium próbkowania basenu przyciągania optimum.

Procedura FUE polega na wskazaniu, dla dolnej (oraz górnej) granicy przedziału niepewności każdego parametru, takiego punktu p_{approx} nienależącego do repozytorium, którego aproksymowana wartość $f_{approx}(\mathbf{x}) \leq y_{PU}$. Poszukiwania prowadzone są iteracyjnie w kierunku zewnętrznym wzdłuż półprostej mającej początek w punkcie p_{BEST} i przechodzącej przez punkt skrajny ze zbioru AS, którego dana wartość parametru została przyjęta jako minimum (albo maksimum) przedziału niepewności. Aproksymacja wartości $f_{approx}(\mathbf{x})$ opiera się na średniej ważonej wartości $f(\mathbf{x})$ sąsiednich punktów należących do repozytorium. Wagą jest odległość. Jako liczbę sąsiadów przyjmuje się podwojoną długość wektora parametrów.

Procedura FUE przyjmuje jako argumenty:

- punkty p_{BEST} oraz p_{AS} definiujące półprostą, wzdłuż której prowadzone są poszukiwania;
- identyfikator wymiaru;
- wartość $step_{min}$ określającą minimalną zmianę granicy przedziału niepewności.

Ponadto procedura korzysta z:

- repozytorium próbkowania przestrzeni;
- wartości argumentu y_{PU} .

Rezultatem działania procedury FUE jest:

nienależący do przetwarzanego repozytorium punkt p_{approx}, leżący na półprostej zdefiniowanej przez punkty p_{BEST} i p_{AS}, którego aproksymowana wartość f_{approx}(x) ≤ y_{PU} jest większa niż wartość punktu p_{AS};

albo

informacja, że nie da się poszerzyć granicy przedziału niepewności dla zadanej wartości step_{min} w oparciu o dane repozytorium próbkowania basenu przyciągania optimum. W takim przypadku wartość określonego parametru punktu p_{AS} stanowi najlepsze oszacowanie konkretnej wartości minimalnej (albo maksymalnej) przedziału niepewności.

Schemat procedury FUE przedstawia Rys. 5.18.

```
x<sub>BEST</sub> = p<sub>BEST</sub>.x; x<sub>AS</sub> = p<sub>AS</sub>.x; x<sub>CURRENT</sub> = x<sub>AS</sub>; y<sub>CURRENT</sub> = p<sub>AS</sub>.y;
step = unitVector (x_{BEST}, x_{AS});
if (|step<sub>i</sub>| < step<sub>min</sub>) step *= step<sub>min</sub> / |step<sub>i</sub>|;
while (1) {
    x_{\text{NEXT}} = x_{\text{CURRENT}} + \text{step}; y_{\text{NEXT}} = f_{\text{APPROX}}(x_{\text{NEXT}});
    if (y_{\text{NEXT}} > y_{\text{PU}} | | y_{\text{NEXT}} < y_{\text{CURRENT}}) {
        if (step<sub>i</sub> = step<sub>min</sub>) break; //ends while loop
        else {
            step /= 3;
            if (|step<sub>i</sub>| < step<sub>min</sub>) step *= step<sub>min</sub> / |step<sub>i</sub>|;
        }
    }
   else {
        y_{\text{CURRENT}} = y_{\text{NEXT}};
        x_{\text{CURRENT}} = x_{\text{NEXT}};
    }
}
if (x_{\text{CURRENT}} \mid = x_{\text{AS}}) p_{\text{APPROX}} = \text{Point}(x_{\text{CURRENT}}, y_{\text{CURRENT}});
else p<sub>APPROX</sub> = NULL;
```

Rys. 5.18 Pseudokod algorytmu FUE

Poszukiwanie rozpoczyna się w punkcie p_{AS} . Wektor *step* jest inicjalizowany wektorem jednostkowym określonym dla wektorów parametrów punktów p_{BEST} i p_{AS} . W każdym kroku do aktualnego punktu dodawany jest wektor *step*. Jeśli aproksymowana wartość $f_{approx}(\mathbf{x})$ nie przekracza wartości y_{PU} , to poszukiwania są kontynuowane. W przeciwnym przypadku procedura cofa się do poprzedniego punktu, a długość kroku *step* jest zmniejszana trzykrotnie. W każdym przypadku procedura FUE wykona przynajmniej jeden krok, w którym *step_i=step_{min}*.

5.3.2. Wyniki estymacji niepewności dla przypadku testowego

W ramach etapu H prezentowana informacja o niepewności wyznaczenia wektora parametrów zawiera:

- informację o płaszczyźnie progowej y_{PU}, dla której wyznaczono niepewność oraz wartość f(x) punktu p_{BEST} (ang. threshold plane info);
- przedziały niepewności wyznaczenia poszczególnych parametrów określone jedynie na podstawie repozytorium próbkowania basenu przyciągania optimum

(ang. *uncertainties directly from repository*). W nawiasach podawana jest odległość granicy przedziału niepewności od optymalnej wartości parametru;

- przedziały niepewności wyznaczenia poszczególnych parametrów określone na podstawie procedury FUE (ang. *uncertainties after further estimations*). W nawiasach podawana jest odległość granicy przedziału niepewności od optymalnej wartości parametru;
- informację o dokładności wyznaczenia niepewności (ang. uncertainties estimation accuracy).

Rys. 5.19 – Rys. 5.21 przedstawiają wyniki etapu H dla repozytoriów grid3k, rand3k oraz ga3k w postaci wydruku z programu ScanRep. Tabela 5.1 zawiera przypomnienie rozwiązania przypadku testowego, zdefiniowanego w punkcie 4.1. Układ kolumn tabeli ułatwia jego porównanie z rezultatami osiągniętymi dla trzech metod próbkowania przestrzeni.

Porównanie przedstawionych wyników dowodzi poprawnego oszacowania niepewności w przypadku testowym dla wszystkich trzech typów próbkowania przestrzeni. Tak jak przewidywano, niedokładności wyznaczenia przedziału niepewności określonej jedynie na podstawie repozytorium próbkowania basenu przyciągania optimum objawiają się jej niedoszacowaniem. Różnica jest co najwyżej zmianą o jeden drugiej cyfry znaczącej. Główny wpływ na niedokładność ma przyjęta wartość argumentu procedury MDR. W etapie C określono minimalną odległość między punktami na 0.5.

Zastosowanie procedury dalszej estymacji niepewności zmniejsza niedokładności najczęściej do trzeciej cyfry znaczącej. Wyjątek stanowi dolna granica przedziału niepewności pierwszego parametru dla repozytorium rand3k, gdzie niedokładność nadal objawia się zmianą o jeden drugiej cyfry znaczącej. Ze względu na powszechnie stosowane zaokrąglanie niepewności do dwóch cyfr znaczących, niedokładność wyznaczenia przedziału niepewności określonego z zastosowaniem procedury FUE jest najczęściej mniejsza niż błąd zaokrąglenia, a więc jest nieistotna.

Ciekawym zjawiskiem jest przeszacowywanie przedziału niepewności w niektórych przypadkach przez procedurę FUE. Ma na to wpływ zarówno lokalny spadek gęstości próbkowania, jak i liniowy charakter aproksymacji wartości punktu $p_{\rm approx}$.

Analiza rezultatów Metody AH dla repozytoriów grid3k oraz rand3k ujawnia nieprawidłowe wskazywanie optymalnych wartości parametrów. Ma to istotny wpływ na wyznaczane niepewności względne. Jedynie dla próbkowania przestrzeni algorytmem genetycznym, niepewności względne są prawidłowo oszacowane. Pokazuje to, że zastosowanie algorytmu genetycznego do estymacji niepewności daje wyniki nie tylko porównywalne, ale w niektórych aspektach

nawet lepsze niż w innych metodach próbkowania przestrzeni. Ponadto powtórne wykorzystanie próbkowania przestrzeni pochodzącego z procesu optymalizacji algorytmem genetycznym, niweluje konieczność kolejnych kosztownych czasowo wywołań funkcji celu.

```
#THRESHOLD_PLANE_INFO#
Threshold Plane : 10.000 Best point value: 1.838
#UNCERTAINTIES_DIRECTLY_FROM_REPOSITORY#
No | Value | Uncertainty [min, max] | Range | Uncertainty[%]
1 | -26.000 | -33.000( -7.000), -18.000( +8.000) | 15.000 | -26.9%, +30.8%
2 | -26.000 | -33.000( -7.000), -18.000( +8.000) | 15.000 | -26.9%, +30.8%
#UNCERTAINTIES_AFTER_FURTHER_ESTIMATIONS#
No | Value | Uncertainty [min, max] | Range | Uncertainty[%]
1 | -26.000 | -33.428( -7.428), -17.802( +8.198) | 15.625 | -28.6%, +31.5%
2 | -26.000 | -33.428( -7.428), -17.802( +8.198) | 15.625 | -28.6%, +31.5%
#UNCERTAINTIES_ESTIMATION_ACCURACY#
Spread of FrontLiners : 7.330
furthestEFL (spreadEFL): 14.003 (+4.003)
furthestAFL (spreadAFL): 6.673 (-3.327)
```

Rys. 5.19 Repozytorium grid3k. Metoda AH: etap H. Wydruk z programu ScanRep

```
#THRESHOLD PLANE INFO#
Threshold Plane : 10.000
                           Best point value: 1.840
   #UNCERTAINTIES_DIRECTLY_FROM_REPOSITORY#
No | Value | Uncertainty [min, max]
                                          | Range | Uncertainty[%]
#UNCERTAINTIES_AFTER_FURTHER_ESTIMATIONS#
No | Value |
                 Uncertainty [min, max]
                                          / Range | Uncertainty[%]
1 | -25.723 | -32.440( -6.717), -18.005( +7.718) | 14.435 | -26.1%, +30.0%
2 | -25.747 | -33.067( -7.320), -18.047( +7.700) | 15.020 | -28.4%, +29.9%
   #UNCERTAINTIES_ESTIMATION_ACCURACY#
Spread of FrontLiners : 9.215
furthestEFL (spreadEFL): 14.733 (+4.733)
furthestAFL (spreadAFL):
                     5.518 (-4.482)
```

Rys. 5.20 Repozytorium rand3k. Metoda AH: etap H. Wydruk z programu ScanRep

#THRESHOLD_PLANE_INFO# Threshold Plane : 10.000 Best point value: 1.834 *#UNCERTAINTIES_DIRECTLY_FROM_REPOSITORY#* No | Value | Uncertainty [min, max] | Range | Uncertainty[%] 1 | -25.877 | -33.003(-7.126), -18.464(+7.413) | 14.539 | -27.5%, +28.6% 2 | -25.877 | -32.993(-7.116), -18.640(+7.237) | 14.353 | -27.5%, +28.0% #UNCERTAINTIES_AFTER_FURTHER_ESTIMATIONS# | Range | Uncertainty[%] No | Value | Uncertainty [min, max] 1 | -25.877 | -33.212(-7.335), -18.137(+7.740) | 15.074 | -28.3%, +29.9% 2 | -25.877 | -33.479(-7.602), -17.663(+8.214) | 15.816 | -29.4%, +31.7% #UNCERTAINTIES_ESTIMATION_ACCURACY# Spread of FrontLiners : 9.186 furthestEFL (spreadEFL): 14.714 (+4.714) furthestAFL (spreadAFL): 5.528 (-4.472)

Rys. 5.21 Repozytorium ga3k. Metoda AH: etap H. Wydruk z programu ScanRep

Nr	Wartość	Niepewność [min, max]	Przedział	Niepewność [%]
1	-25.877	-33.502 (-7.625) , -17.928 (+7.949)	15.574	-29.5% , +30.7%
2	-25.877	-33.502 (-7.625) , -17.928 (+7.949)	15.574	-29.5% , +30.7%

Tabela 5.1 Rozwiązanie dokładne przypadku testowego

Number of points: 385 Number of dimensions: 2 Best point value: 1.834 Best point parameters: -25.877 -25.877 *Number of clusters:* 1 **REPOSITORY menu:** 1 - Append repository, 2 - Start distances calculation in separate thread (needs 0.28 MB RAM), 3 - Cut to phenotype (phenotype = domain defined in input), 4 - ThresholdCut and MDR, 5 - Find clusters (calculatedDistances improve speed), 6 - Merge clusters, 7 - Work with cluster, Print: 8 - Clusters' info, 9 - ALL points (sorted by clusterID), 10 - Distances array, 11 - PROJECTION, 12 - Ranges in Phenotype, 13 - SSH (Space Sampling HISTOGRAM), 14 - Simplified Parameter Uncertainties, 0 - Exit.

Rys. 5.22 Menu przetwarzania repozytorium w programie ScanRep

```
Number of points:
                        374
Best point value:
                         1.834
Best point parameters: -25.877 -25.877
CLUSTER menu:
1 - TheresholdCut,
2 - Start distances calculation. (separate thread)
3 - print WHOLE cluster,
4 - STATISTICS (generalInfo, samplingRanges, moments,
                 covarianceMatrix, correlationMatrix),
5 - SSH (Space Sampling HISTOGRAM),
6 - DBH (Distances from Best HISTOGRAM),
7 - VDH (Vertex Degree HISTOGRAM),
8 - Front-Line Algorithm,
9 - Parameter Uncertainties,
0 - Exit.
```

Rys. 5.23 Menu przetwarzania klastra w programie ScanRep

5.4. Program ScanRep

Program ScanRep zawiera implementację Metody AH. Umożliwia on przeprowadzenie na wskazanym repozytorium wszystkich jej etapów oraz procedur pomocniczych. Na każdym etapie można zapisać rezultaty do pliku albo wypisać je na konsolę.

Program ScanRep został napisany w języku C++, przy pomocy środowiska Microsoft Visual Studio 2010. Działa on w trybie konsolowym (tekstowym), jednak w przyszłości możliwa jest implementacja graficznego interfejsu użytkownika. Program wymaga podania właściwego "inputu do GOSI" wraz z repozytorium. Użytkownik może przetwarzać całe repozytorium (vide Rys. 5.22) albo wyodrębniony z niego klaster punktów (vide Rys. 5.23). Dodatek A.3. zawiera szczegółowe omówienie wszystkich klas programu ScanRep.

5.5. Schemat Metody AH

Niniejszy punkt zawiera podsumowanie opisu Metody AH zawartego w rozdziałach 4. oraz 5. Rys. 5.24 przedstawia schemat działania Metody AH wraz z krótkim opisem poszczególnych etapów.

Metoda AH składa się z ośmiu etapów oznaczonych kolejnymi literami alfabetu od A do H. Ze względu na międzynarodowy charakter jej użytkowników, nazwy etapów zostały określone w języku angielskim. Polskie tłumaczenie zamieszczono razem z opisem danego etapu.

Przetwarzanie wejściowego repozytorium punktów podzielone jest na trzy moduły. Pierwsze cztery etapy składają się na moduł przetwarzania wstępnego, w którym wyodrębniane jest próbkowanie basenu przyciągania optimum. Kolejne dwa etapy tworzą moduł analizy rozkładu próbkowania przestrzeni wokół optimum. Ostatnie dwa etapy stanowią moduł wyznaczania niepewności.

W ramach modułu przetwarzania wstępnego konieczne jest podanie wartości trzech argumentów Metody AH. W etapie B użytkownik określa wartość y_{CUT} , w etapie C wartość argumentu procedury MDR, a w etapie D parametr algorytmu klasteryzacji NBC, tj. liczbę elementów zbioru k-najbliższych sąsiadów. W opisie etapów B, C oraz D zawarto wskazówki odnośnie wyboru wartości tych argumentów. Jak pokazano w rozdziale 6., w analizie danych z pomiarów wzbudzeń kulombowskich, szczególnie w przypadku wartości argumentu procedury MDR, najczęściej konieczne jest porównanie rezultatów dla kilku różnych wartości, aby wybrać najlepszą.

W module wyznaczania niepewności użytkownik musi określić wartość y_{PU} , definiującą płaszczyznę poziomą, dla której ma być wyznaczona niepewność. Wskazówki dotyczące wybrania właściwej wartości y_{PU} zamieszczono w punkcie 5.1.

	Etap		Nazwa Etapu	Opis Etapu		
		A Raw Dataset		wejściowe repozytorium próbkowania przestrzeni optymalizacyjnej		
	nie wstępne	В	Threshold Cut	wszystkie punkty z wartością powyżej zadanego progu są usuwane		
Η	Przetwarzar	С	Minimum Distance Rule	wszystkie punkty będące bliżej niż określona minimalna odległość od punktów z lepszą wartością są odsiewane		
d a A		D Clusterization		wykrycie punktów znajdujących się w basenie przyciągania globalnego optimum – zastosowanie algorytmu NBC		
M e t o	óbkowania	Е	Statistics	informacja o rozkładzie statystycznym punktów w basenie przyciągania globalnego optimum (zakresy, momenty, kowariancja, korelacja)		
I	Analiza pro	F	Histograms: Space Sampling Histogram, Vertex Degree Histogram, Distances from Best Histogram	informacja o rozkładzie punktów w basenie przyciągania globalnego optimum – na podstawie odległości między punktami		
	niepewności	G G_noSF	Front-Line Algorithm	wykrycie punktów rozmieszczonych najbliżej płaszczyzny wyznaczenia niepewności		
	Wyznaczenie	Н	Parameter Uncertainties	wyznaczenie niepewności parametrów dla zadanej płaszczyzny progowej		

Rys. 5.24 Schemat Metody AH

6. Zastosowanie Metody AH do analizy danych z pomiarów wzbudzeń kulombowskich

Zastosowanie Metody AH do przypadku rzeczywistego zostało pokazane na przykładzie analizy danych z pomiaru wzbudzenia kulombowskiego baru ¹³²Ba [130], [131].

Opisywany eksperyment odbył się w dniach 22–28 czerwca 2015 roku w Środowiskowym Laboratorium Ciężkich Jonów Uniwersytetu Warszawskiego na stanowisku pomiarowym EAGLE [20] zgodnie z techniką przedstawioną w rozdziale 2. Rozpędzone w cyklotronie U-200P do energii 90 MeV jądra siarki ³²S wzbudzały kulombowsko znajdujące się w tarczy o grubości ~ $631 \mu g/cm^2$ jądra wzbogaconego chlorku baru BaCl₂ (w tym ~ $410 \mu g/cm^2$ baru). Piętnaście jednolitych detektorów germanowych HPGe rejestrowało wyemitowane kwanty promieniowania gamma, a 48 PIN-diod każda o powierzchni czynnej $0.5 \times 0.5 cm^2$ rejestrowało rozproszone wstecznie ($110^\circ \div 170^\circ$) jądra pocisku lub tarczy. Układ akwizycyjny rejestrował dane eksperymentalne zgodnie z techniką koincydencji cząstkagamma. Jako zdarzenie zapisywano jednoczesne (w oknie czasowym o długości 200 ns) zaobserwowanie rozproszonej cząstki oraz co najmniej jednego kwantu promieniowania gamma. Zebrane dane (około 10^7 zdarzeń) zostały poddane korekcie dopplerowskiej i przygotowane do dalszej analizy przez zespół fizyków.



Rys. 6.1 Schemat poziomów w jądrze ¹³²Ba

Rys. 6.1 przedstawia zakładany schemat poziomów energetycznych stanów wzbudzonych i podstawowego ¹³²Ba. Na podstawie danych obserwowanych w opisanym eksperymencie, geometrii układu pomiarowego oraz niniejszego schematu utworzono zbiór plików wejściowych do programu GOSIA.

Zdefiniowana w "inpucie do GOSI" (vide Dodatek B.1) przestrzeń optymalizacyjna jest ośmiowymiarowa. Zakresy zmienności poszczególnych parametrów wraz z ich oznaczeniem

fizycznym zawarto w Tabeli 6.1. Znalezienie ośmioelementowego wektora elementów macierzowych ME, dla którego wartość $\chi^2(ME)$ jest najmniejsza, stanowi problem optymalizacyjny.

Zadanie minimalizacji funkcji $\chi^2(ME)$ wykonano przy użyciu opisanego w rozdziale 3. algorytmu genetycznego. Zastosowane ustawienia algorytmu zebrano w Tabeli 6.2. Są one podobne do konfiguracji stosowanej w omówionym przypadku testowym, co do której wykazano w rozdziałach 4. i 5., że powstałe przy jej użyciu repozytoria próbkowania przestrzeni można następnie wykorzystać do oszacowania niepewności wartości parametrów. Ze względu na czterokrotny wzrost liczby wymiarów zwiększono jedynie liczbę osobników w populacji oraz liczbę pokoleń.

Parametr	Minimum	Maksimum	Oznaczenie fizyczne
ME_1	0.01	2.00	$\langle \ 2^+_1 \parallel E2 \parallel 0^+_1 \ angle$
ME ₂	0.01	2.00	$\langle \ 2^+_2 \parallel E2 \parallel 0^+_1 angle$
ME ₃	-2.00	2.00	$\langle \ 2^+_1 \parallel E2 \parallel 2^+_1 \ angle$
ME ₄	0.01	2.00	$\langle \hspace{.1cm} 4^{+}_{1} \hspace{.1cm} \ \hspace{.1cm} \textit{E2} \hspace{.1cm} \ \hspace{.1cm} 2^{+}_{1} \hspace{.1cm} angle$
ME ₅	-2.00	2.00	$\langle \ 2^+_2 \parallel E2 \parallel 2^+_1 angle$
ME ₆	-2.00	2.00	$\langle \hspace{.1cm} 4_1^+ \hspace{.05cm} \parallel E2 \hspace{.05cm} \parallel \hspace{.05cm} 2_2^+ \hspace{.05cm} angle$
ME ₇	-2.00	2.00	$\langle \ 2^+_2 \parallel E2 \parallel 2^+_2 angle$
ME ₈	-2.00	2.00	$\langle 2_2^+ \parallel Ml \parallel 2_1^+ \rangle$

Tabela 6.1 Zakresy zmienności parametrów

reprezentacja	rzeczywista		
rozmiar populacji	300		
liczba pokoleń	1000		
operator selekcji	selekcja elitarna z progiem odcięcia 60%		
operator krzyżowania	krzyżowanie heurystyczne-2 z parametrem <i>maxChildren</i> =5		
operator mutacji	mutacja gaussowska ze stałym współczynnikiem mutacji 10% oraz argumentem operatora mutacji <i>sigma</i> =0.20		

Tabela 6.2 Konfiguracja algorytmu genetycznego

Aby zapewnić stochastyczny skan całej przestrzeni optymalizacyjnej utworzono 40 repozytoriów. Oznaczono je jako rep*i*, gdzie *i* jest numerem przebiegu algorytmu genetycznego. Konieczne do tego było ok. $7.7 \cdot 10^6$ wywołań funkcji celu, a łączny czas obliczeń wyniósł ok. 128 godzin.

Średnia liczba punktów w repozytorium dla tej konfiguracji algorytmu genetycznego wyniosła 192 327. Czas potrzebny do wyznaczenia wartości $\chi^2(ME)$, dla jednego wektora parametrów, oszacowano empirycznie na ok. 0.06 *s*. Stąd czas potrzebny do realizacji jednego przebiegu algorytmu genetycznego (utworzenia jednego repozytorium) to ok. 192 $327 \cdot 0.06 \, s \approx 3.2 \, godziny$.

Utworzenie repozytorium z próbkowaniem równomiernym o podobnej jakości wymagałoby rozmieszczenia punktów na siatce co 0.1. Dla omawianej ośmiowymiarowej przestrzeni optymalizacyjnej wymagałoby to $21^3 \cdot 41^5 \approx 1.07 \cdot 10^{12}$ wywołań funkcji celu, co zajęłoby ok. $1.8 \cdot 10^7 godzin \approx 2000 \, lat$. Potwierdza to zamieszczony w punkcie 4.3. wniosek, że zastosowanie algorytmu genetycznego do tworzenia repozytoriów radykalnie zmniejsza liczbę wywołań funkcji celu oraz skraca czas obliczeń na skutek zagęszczania próbkowania przestrzeni w basenach przyciągania optimów.

Porównanie punktów z najniższą wartością $\chi^2(ME)$ z każdego z otrzymanych 40 repozytoriów pokazało, że koncentrują się one (w stosunku 25:15) w basenach przyciągania dwóch optimów. Tabela 6.3 zawiera informacje dotyczące tych dwóch alternatywnych rozwiązań. Dla każdego przedstawiono najlepsze znalezione wartości elementów macierzowych albo przedziały zmienności wartości tych parametrów, jeśli różniły się one w poszczególnych repozytoriach. Na tej podstawie można ocenić, w jakim stopniu dla różnych wymiarów w najbliższym otoczeniu minimum funkcja celu jest płaska. Może to wskazywać na potencjalną mniejszą czułość danych eksperymentalnych w tych obszarach przestrzeni. Wartości $\chi^2(ME)$ dla obu znalezionych rozwiązań są zbliżone. Z tego powodu należy je uznać za minima równorzędne numerycznie (pomijając ich interpretację z punktu widzenia fizyki teoretycznej).

W kolejnych punktach niniejszego rozdziału opisano zastosowanie Metody AH do oszacowania niepewności obu przedstawionych wektorów elementów macierzowych. W punkcie 6.1. umieszczono konkluzje z przebiegu działania modułu przetwarzania wstępnego. W punkcie 6.2. zawarto rezultaty modułu analizy próbkowania. W punkcie 6.3. zaprezentowano wyniki modułu wyznaczenia niepewności.

Aby porównać działanie Metody AH dla zbiorów o różnej liczności, rezultaty dla pojedynczych repozytoriów zostały zestawione z wynikami otrzymanymi dla repozytoriów

złączonych: po dziesięć oraz wszystkie razem. Konkatenacje oznaczono odpowiednio: ten*i* oraz all, gdzie *i* jest numerem zbioru złączonego.

Umieszczenie wszystkich rezultatów z poszczególnych etapów Metody AH dla wszystkich repozytoriów pojedynczych oraz złączonych zajęłoby ok. 100 stron. Ze względu na ograniczoną objętość rozprawy, zamieszczono wyniki najbardziej istotne z punktu widzenia prezentacji samej Metody AH, w szczególności związane z większą liczbą wymiarów niż w przypadku testowym oraz z porównaniem repozytoriów o różnej liczności.

	Rozwiązanie 1	Rozwiązanie 2
	(otrzymane przedziały)	(najlepsze wartości albo otrzymane przedziały)
$\chi^2(ME)$	[0.953 , 0.954]	0.971
ME_1	[0.922 , 0.925]	0.937
ME ₂	[0.266 , 0.267]	0.261
ME ₃	[0.075 , 0.201]	[0.048 , 0.069]
ME ₄	[1.231 , 1.250]	[1.159 , 1.162]
ME ₅	[1.018 , 1.035]	[-0.862 , $-0.860]$
ME ₆	[0.001 , 0.026]	[0.759 , 0.761]
ME ₇	[-1.519 , -1.359]	[-1.999 , -2.000]
ME ₈	[0.645 , 0.651]	[-0.687, -0.688]

Tabela 6.3 Wyznaczone optymalne wartości parametrów dla poszczególnych repozytoriów

6.1. Moduł przetwarzania wstępnego

W ramach modułu przetwarzania wstępnego konieczne jest określenie wartości trzech argumentów Metody AH. Wartości zastosowane do analizy omawianego przypadku rzeczywistego podano w Tabeli 6.4.

Etap B	Etap C	Etap D	
Threshold Cut	Minimum Distance Rule	Clusterization	
	0.01		
	0.02	L 100	
$y_{cut} = 10$	0.05	$\kappa = 100$	
	0.10		

Tabela 6.4 Przyjęte wartości argumentów Metody AH

Zgodnie ze wskazówkami zawartymi w opisie etapu B (punkt 4.3), z repozytoriów usuwano punkty, których wartość $\chi^2(ME)$ była większa niż $y_{cut} = 10$. Najbardziej złożonym zagadnieniem okazało się właściwe określenie minimalnej odległości między punktami w repozytorium. Z tego powodu w module przetwarzania wstępnego porównano rezultaty dla czterech różnych wartości argumentu procedury MDR (punkt 4.4). Jako wielkość zbioru knajbliższych sąsiadów w algorytmie grupowania NBC (punkt 4.5) przyjęto k = 100. Z jednej strony wartość ta stanowi dolne ograniczenie rozmiaru klastra jaki może zostać wyodrębniony, z drugiej większa wartość argumentu k skutkowałaby niepotrzebnym wydłużeniem czasu pracy algorytmu grupowania. Jak pokazały testy, powyższa wartość jest w większości przypadków wystarczająca do podziału repozytoriów na właściwą liczbę klastrów. Powstałe w nielicznych przypadkach dodatkowe grupy szczątkowe są małoliczne i nie mają istotnego wpływu na późniejszą dokładność wyznaczenia niepewności.

Celem modułu przetwarzania wstępnego jest wyodrębnienie z repozytorium punktów znajdujących się w basenie przyciągania wyznaczonego rozwiązania optymalnego. W rozważanym przypadku rzeczywistym znaleziono dwa komplementarne minima. Aby określić niepewności dla obu wektorów elementów macierzowych, konieczne jest wyodrębnienie grup punktów skupionych wokół każdego z nich.

Podczas przetwarzania repozytoriów repi, z których każde zawiera próbkowanie przestrzeni optymalizacyjnej z jednego przebiegu algorytmu genetycznego, zawsze wyodrębniano klaster należący do basenu przyciągania jednego z dwóch rozwiązań równorzędnych. W repozytoriach zbiorczych wyodrębniano grupy punktów należące do basenów przyciągania dwóch rozwiązań komplementarnych. Jest to logiczne, gdyż składające się na konkatenacje repozytoria zawierają klastry z próbkowaniem w pobliżu obu minimów.

Tabela 6.5 zawiera liczby punktów we wszystkich repozytoriach repi po kolejnych etapach modułu przetwarzania wstępnego dla przyjętych wartości argumentów Metody AH. Przedstawia ona ponadto informacje statystyczne na temat przeprowadzonego próbkowania.

Tabela 6.6 zawiera analogiczne dane dla repozytoriów zbiorczych wzbogaconych o informację na temat liczności obu głównych klastrów (zawierających próbkowanie dla dwóch rozwiązań komplementarnych). W zestawieniu pominięto rezultaty klasteryzacji dwóch najliczniejszych zbiorów. Ich wyznaczenie kosztowałoby ponad miesiąc obliczeń, a nie są one istotne dla treści przedstawionych w dalszych punktach tego rozdziału.

rep	Raw Dataset	yCut=10	MDR 0.01	NBC k=100	MDR 0.02	NBC k=100	MDR 0.05	NBC k=100	MDR 0.1	NBC k=100
1	192 208	128 658	48 823	45 884	39 224	36 805	20 737	19 519	9 104	8 691
2	192 223	119 405	44 194	41 784	33 561	31 896	17 574	16 649	9 155	8 661
3	192 343	139 599	92 913	88 107	72 748	68 999	34 981	33 280	12 412	11 878
4	192 254	141 551	64 453	60 552	41 352	38 567	19 832	18 579	7 479	6 943
5	192 386	153 103	51 446	47 861	29 382	27 077	11 994	11 061	4 627	4 257
6	192 332	146 012	66 853	63 344	47 092	44 843	18 501	17 679	7 276	6 904
7	192 433	144 611	64 914	61 440	48 595	45 943	17 843	16 846	4 922	4 634
8	192 408	133 440	53 939	51 289	36 729	34 864	17 558	16 676	7 581	7 180
9	192 205	147 831	46 073	43 328	29 845	28 015	14 619	13 661	7 954	7 495
10	192 508	129 559	60 516	57 706	47 171	45 043	24 695	23 438	10 253	9 732
11	192 365	146 366	108 790	103 212	72 370	68 719	32 395	30 673	13 927	13 254
12	192 459	144 741	39 520	36 985	24 367	22 778	14 915	14 078	8 961	8 517
13	192 391	146 700	82 501	77 749	53 980	50 879	23 896	22 405	9 618	9 022
14	192 195	145 675	69 382	65 291	42 160	39 649	18 095	16 915	8 250	7 727
15	192 237	147 425	74 955	70 445	50 676	47 525	23 185	21 783	11 993	11 399
16	192 380	136 425	67 078	63 531	50 254	47 568	17 746	16 663	5 980	5 630
17	192 304	141 332	94 341	88 778	68 799	64 884	32 864	30 997	12 681	12 099
18	192 168	144 342	55 906	52 556	34 856	32 558	17 993	16 788	7 788	7 300
19	192 131	151 022	52 773	49 309	38 393	35 642	17 168	15 796	7 938	7 393
20	192 426	129 850	61 603	58 273	42 568	40 322	20 824	19 937	10 304	9 916
21	192 257	127 225	37 900	35 949	27 130	25 762	15 128	14 389	8 865	8 420
22	192 436	140 364	76 568	72 442	52 068	49 593	20 001	19 185	5 896	5 657
23	192 440	117 477	52 108	46 861	44 816	40 046	28 297	24 549	12 272	11 650
24	192 201	152 018	55 572	51 671	32 140	29 961	15 687	14 562	7 015	6 549
25	192 394	145 860	68 363	64 892	34 069	32 226	12 561	11 803	5 971	5 603
26	192 262	146 239	76 710	73 009	64 722	61 674	37 044	35 553	12 847	12 403
27	192 384	123 228	41 781	39 561	33 187	31 316	20 122	19 242	11 626	11 155
28	192 481	143 211	88 712	84 023	54 055	50 978	25 958	24 528	12 916	12 260
29	192 450	146 554	34 361	32 178	20 858	19 231	12 533	11 635	6 774	6 261
30	192 056	126 557	48 876	45 731	40 218	37 731	25 805	24 426	12 017	11 567
31	192 380	146 014	71 674	67 778	49 729	46 996	23 081	22 036	9 496	9 116
32	192 306	133 080	64 909	61 856	43 036	41 136	16 782	16 059	7 177	6 855
33	192 320	150 764	51 086	47 681	33 496	31 144	13 281	12 124	5 728	5 244
34	192 343	149 639	42 609	40 078	24 983	23 228	9 952	9 285	4 531	4 267
35	192 380	150 036	65 715	61 443	35 030	32 676	14 503	13 416	8 238	7 657
36	192 216	131 437	43 938	41 020	27 052	25 303	14 682	13 819	8 756	8 319
37	192 240	130 005	32 009	30 209	20 830	19 707	10 659	10 011	4 997	4 640
38	192 166	149 085	80 678	75 904	56 930	53 588	24 427	23 209	7 962	7 621
39	192 415	129 347	48 603	46 281	34 609	32 981	19 367	18 333	10 748	10 211
40	192 584	148 989	35 336	33 586	25 350	24 049	12 379	11 719	5 034	4 726
avg	192 327	140 119	60 462	56 989	41 461	39 048	19 742	18 583	8 677	8 220
avg	100.00%	72.85%	31.44%	29.63%	21.56%	20.30%	10.26%	9.66%	4.51%	4.27%
<u>%້</u>	-	100.00%	43.15%	40.67%	29.59%	27.87%	14.09%	13.26%	6.19%	5.87%

Tabela 6.5 Porównanie repozytoriów pojedynczych

ten	Raw Dataset	yCut=10	MDR 0.01	NBC* k=100	MDR 0.02	NBC* k=100	MDR 0.05	NBC* k=100	MDR 0.1	NBC* k=100
1	1923300	1383769	581 058	358 318 194 492	413 687	248 165 145 319	185 442	105 236 71 070	72 858	38 683 30 906
2	1923056	1433878	677 986	522 357 121 421	452 060	343 512 85 990	198 228	154 579 34 197	83 462	66 280 14 165
3	1923361	1368733	561 819	294 329 238 121	385 792	184 862 180 073	198 048	93 497 94 336	85 364	39 784 42 499
4	1923350	1418396	512 506	309 079 176 861	331 354	198 697 116 150	145 447	82 665 55 510	64 335	33 758 27 902
						* podano r	ozmiary ob	u komplen	entarnych	klastrów
avg	1923267	1401194	583 342	371 021 182 724	395 723	243 809 131 883	181 791	108 994 63 778	76 505	44 626 28 868
avg	100.00%	72.85%	30.33%	19.29% 9.50%	20.58%	12.68% 6.86%	9.45%	5.67% 3.32%	3.98%	2.32% 1.50%
%	-	100.00%	41.63%	26.48% 13.04%	28.24%	17.40% 9.41%	12.97%	7.78% 4.55%	5.46%	3.18% 2.06%

all	7693067	5604776	2216308	-	1478547	-	650 211	387 957 230 638	257 626	151 502 97 046
all	100.00%	72.85%	28.81%	-	19.22%	-	8.45%	5.04% 3.00%	3.35%	1.97% 1.26%
%	-	100.00%	39.54%	-	26.38%	-	11.60%	6.92% 4.12%	4.60%	2.70% 1.73%

Tabela 6.6 Porównanie repozytoriów zbiorczych

Analiza rezultatów modułu przetwarzania wstępnego potwierdza, że próbkowanie przestrzeni algorytmem genetycznym w znacznym stopniu koncentruje się wokół optimum. Główny wpływ na wielkość repozytorium po etapie D ma przyjęta minimalna odległość między punktami. Na poszczególnych etapach i dla danej wartości argumentu procedury MDR, rozmiary przetwarzanych zbiorów są zbliżone do siebie. Potwierdza to stochastyczną powtarzalność wyników Metody AH dla repozytoriów próbkowania przestrzeni algorytmem genetycznym.

Ze zbiorów złączonych usuwanych jest procentowo trochę więcej punktów, niż ze składających się na nie repozytoriów pojedynczych. Wpływa na to większe nasycenie basenów przyciągania optimów w odniesieniu do przyjętej minimalnej odległości między punktami. Ta hipoteza znajduje potwierdzenie w wynikach modułu analizy statystycznej.

W kolejnych punktach porównane zostały rezultaty modułu analizy statystycznej oraz modułu wyznaczenia niepewności: dla zbiorów o różnej liczności oraz dla różnej minimalnej odległości między punktami. Repozytoria pojedyncze zawierające największą liczbę punktów w basenie

przyciągania określonego rozwiązania równorzędnego (rep11 oraz rep22) zestawiono z próbkowaniem basenów przyciągania odpowiednich minimów wyodrębnionych z repozytoriów zbiorczych: ten*i* oraz all. Minimalna odległość między punktami w zbiorze ten*i* jest taka sama jak w repozytoriach pojedynczych, natomiast w zbiorze all jest o rząd wielkości większa. Jak wynika z przedstawionych liczności zbiorów, rozmiar każdego z dwóch komplementarnych klastrów w zbiorze all przy argumencie procedury MDR wynoszącym 0.10 jest zbliżony do liczby punktów w basenie przyciągania optimum w rep11 oraz rep22 dla minimalnej odległości między punktami wynoszącej 0.01. Przyjęte w celu uproszczenia nazewnictwa oznaczenia przedstawia Tabela 6.7.

Oznaczenie	Rozmiar	Opis zawartości
REP11	103 212	repl1, $y_{cut} = 10$, minimalna odległość 0.01 , $k = 100$, klaster należący do basenu przyciągania rozwiązania 1
TEN_S1	522 357	ten2, $y_{cut} = 10$, minimalna odległość 0.01 , $k = 100$, klaster należący do basenu przyciągania rozwiązania 1
ALL_S1	151 502	all, $y_{cut} = 10$, minimalna odległość 0.10 , $k = 100$, klaster należący do basenu przyciągania rozwiązania 1
REP22	72 442	rep22, $y_{cut} = 10$, minimalna odległość 0.01, $k = 100$, klaster należący do basenu przyciągania rozwiązania 2
TEN_S2	238 121	ten3, $y_{cut} = 10$, minimalna odległość 0.01, $k = 100$, klaster należący do basenu przyciągania rozwiązania 2
ALL_S2	97 046	all, $y_{cut} = 10$, minimalna odległość 0.10 , $k = 100$, klaster należący do basenu przyciągania rozwiązania 2

Tabela 6.7 Przyjęte oznaczenia przetwarzanych zbiorów punktów

6.2. Moduł analizy próbkowania

Celem modułu analizy próbkowania jest dostarczenie ilościowego oszacowania jakości repozytorium z punktu widzenia wykorzystania zebranego w nim próbkowania przestrzeni do estymacji przedziałów niepewności znalezionego optymalnego wektora parametrów. Ocena odbywa się na podstawie analizy statystycznej oraz histogramowej zbioru punktów.

W omawianym przypadku rzeczywistym rezultaty modułu analizy próbkowania należy rozważać oddzielnie dla basenu przyciągania każdego rozwiązania równorzędnego. Z tego powodu są one prezentowane jako trójki: REP11, TEN_S1 oraz ALL_S1, a następnie: REP22, TEN_S2 oraz ALL_S2.

Rys. 6.2 – Rys. 6.7 przedstawiają wyniki analizy statystycznej przetwarzanych zbiorów punktów. Ze względu na liczbę wymiarów oraz przejrzystość prezentacji, w wydrukach z programu ScanRep pominięto macierze kowariancji, dzięki czemu każdy wydruk mieści się na jednej stronie.

Porównanie przedstawionych wyników ujawnia istotne różnice w próbkowaniu przestrzeni wokół optimum w obrębie rozważanych trójek repozytoriów. Jednocześnie zmiany te są analogiczne dla obu trójek:

- dla wszystkich sześciu zbiorów, w każdym wymiarze, punkty rozmieszczone są po obu stronach minimum, ale zakresy próbkowania dla repozytoriów pojedynczych są węższe niż dla złączonych;
- wartości kurtozy dla zbiorów pojedynczych oraz złączonych po dziesięć są znacząco wyższe niż dla repozytoriów ALL_S1 oraz ALL_S2. Pokazuje to działanie procedury MDR. Przyjęcie o rząd wielkości wyższej minimalnej odległości między punktami skutkuje rozrzedzeniem próbkowania przestrzeni w najbliższym otoczeniu minimum. Jednocześnie nie ma to wpływu na odleglejsze części basenu przyciągania optimum.

Rys. 6.8 – Rys. 6.13 zawierają histogramy DBH omawianych repozytoriów. Analiza histogramowa została przeprowadzona w oparciu o wszystkie trzy typy histogramów opisane w punkcie 4.7., a wnioski przedstawiono poniżej. Histogramy SSH nie zostały zamieszczone w pracy ze względu na ich dużą objętość dla ośmiowymiarowej przestrzeni optymalizacyjnej. Ze względu na różną minimalną odległość między punktami w przetwarzanych repozytoriach i będącą tego konsekwencją utrudnioną czytelność, histogramy VDH również zostały jedynie omówione.

Analiza histogramów SSH dowodzi, że baseny przyciągania obu minimów równorzędnych są gęsto próbkowane. Jednocześnie brak jest przerw w próbkowaniu otoczeń optimów dla wszystkich sześciu zbiorów punktów.

Porównanie histogramów DBH pokazuje, że w repozytoriach pojedynczych liczba punktów w kolejnych hiper-pierścieniach kołowych wokół optimów jest nawet o rząd wielkości mniejsza niż w złączonych. W pojedynczych zbiorach danych na obrzeżach basenów przyciągania optimów istotnie zmniejsza się gęstość próbkowania przestrzeni. Może to skutkować zwiększeniem niedokładności wyznaczenia niepewności w oparciu o te repozytoria.

Histogramy VDH potwierdzają, że najbardziej równomierne próbkowania przestrzeni zawierają repozytoria ALL_S1 oraz Al1_S2. Jednocześnie punkty w repozytoriach TEN_S1 oraz TEN_S2 są rozmieszczone bardziej równomiernie niż w repozytoriach pojedynczych.

#GENERAL_INFO#					
Number of points:	103212				
Best point value:	0.95	3			
Best point parameters	: 0.92	3 0.2	266 0.2	01 1.	250
	1.01	8 0.0	001 -1.3	59 0.	651
#SAMPLING_RANGES#					
Dim Min Max	Rai	nge			
1 0.739 1	.2//	0.538			
2 0.18/ 0	.399	0.212			
3 -1.806 1	.999	3.805			
4 0.626 1 5 0.784 1	. 381	0.955			
5 -0.784 I	.420	2.204			
7 -2.000 1	.000 997	3 997			
8 0.468 1	.211	0.743			
0 01100 1		017 15			
#MOMENTS#					
Dimension 1					
Expected value:	0.927				
Standard deviation:	0.025	(0.046	- normali	zed to c	luster range)
Skewness:	3.498	- RIGHT	r side lon	ger	
Kurtosis:	25.708	- MORE	concentra	ted then	norma1
Dimension 2					
Expected value:	0.266	~~ ~ ~ ~			-
Standard deviation:	0.009	(0.040	- normali	zed to c	luster range)
Skewness:	2.570	- RIGHT	side lon	ger	7
Kurtosis:	25.771	- MORE	concentra	ted then	normal
Dimension 3	0 503				
Expected Value:	0.593	(0 106	normali	and the c	luston nanga)
Stanuaru uevration.	-0.404		- normari		iuster range)
Kurtosis:	-0.445	- LEFI	concentra	ei tad than	normal
Dimension 4	2.025	MORL	concentra	Lea Lilen	normar
Expected value:	1,181				
Standard deviation:	0.096	(0.101	- normali	zed to c	luster range)
Skewness:	-1.036	- LEFT	side lona	er	
Kurtosis:	1.034	- MORE	concentra	ted then	norma1
Dimension 5					
Expected value:	0.845				
Standard deviation:	0.212	(0.096	- normali	zed to c	luster range)
Skewness:	-2.808	- LEFT	side long	er	_
Kurtosis:	<i>11.054</i>	- MORE	concentra	ted then	norma1
Dimension 6					
Expected value:	0.425	(0 1 2 0	7 .		7
Standard deviation:	0.429	(0.139	- normaii	zea το c	luster range)
SKewness:	0.830	- KIGHI	side ion	ger tod thom	normal
RUILOSIS. Dimension 7	-0.101	- LESS	concentra	Leu Liien	noi illa i
Expected value	-0 414				
Standard deviation	0.728	(0.182	- normali	zed to c	luster ranne)
Skewness:	0.784	- RIGHT	r side lon	aer	luster runge)
Kurtosis:	-0.191	- LESS	concentra	ted then	norma1
Dimension 8					
Expected value:	0.698				
Standard deviation:	0.057	(0.076	- normali	zed to c	luster range)
Skewness:	2.508	- RIGHT	r side lon	ger	_
Kurtosis:	9.514	- MORE	concentra	ted then	norma1
#CODDE:	T \ <i>(H</i>				
#CORRELATION_MATR	1X#	0 124	0 400	0 000	0 225 0 450
1.000 0.369	-U.216	-0.124	-0.402	0.002	0.225 0.450
0.309 I.000 _0.216 _0.144	-U.144 1 000	-0.140 _0 010	-0.103	0.04/	-0.043 0.562
-0.210 $-0.144-0.124$ -0.146	-0 048	1 000	0.030	0.440	-0.676 -0.029
-0.402 - 0.185	0.098	0.683	1,000	-0.546	-0.693 -0.848
0.002 0.047	0.440	-0.901	-0.546	1.000	0.825 0.576
0.225 -0.045	0.564	-0.676	-0.693	0.825	1.000 0.624
0.450 0.562	-0.029	-0.688	-0.848	0.576	0.624 1.000

#GENERAL_INFO#					
Number of points:	522357				
Best point value:	0.95	3			
Best point parameters	: 0.92	3 0.2	267 0.1	104 1.	235
	1.03	3 0.0)21 -1.4	<i>195 0.</i>	648
#SAMPLING_RANGES#					
Dim Min Max	Rai	ige			
1 0.698 1	. 529	0.831			
2 0.187 0	.444	0.257			
3 -1.970 2	.000	5.9/0 1 725			
4 0.003 1 5 $-1.303 1$.030 017	2 250			
6 -1.005 1	. 947	3 915			
7 -2 000 2	000	4 000			
8 -0.267 1	. 451	1.718			
		0			
#MOMENTS#					
Dimension 1					
Expected value:	0.927				
Standard deviation:	0.028	(0.033	- normalı	ized to c	luster range)
Skewness:	2.982	- RIGHT	r side lor	nger	
Kurtosis:	26.418	- MORE	concentra	ated ther	n normal
Dimension 2					
Expected value:	0.266		-		7
Standard deviation:	0.010	(0.040	- normali	zed to c	cluster range)
Skewness:	3.194	- RIGHT	side lor	iger	7
Kurtosis:	28.221	- MORE	concentra	ited ther	n normal
Dimension 3	0 647				
Standard doviation:	0.047	(0 122	- normala	izad ta	Justar ranga)
Stanuaru uevration.	-0 148		- normari		Tuster Tange)
Kurtosis:	-0.140	- LEFI	concentra	ntad thar	normal
Dimension 4	0.105	MORL	concentra	ileu thei	1 1101 1110 1
Expected value:	1,233				
Standard deviation:	0.114	(0.092	- normali	ized to c	luster range)
skewness:	-0.374	- 1 FFT	side lond	ier	
Kurtosis:	1.217	- MORE	concentra	ted ther	normal
Dimension 5		-			
Expected value:	0.873				
Standard deviation:	0.265	(0.082	- normali	ized to c	luster range)
Skewness:	-3.009	- LEFT	side long	<i>jer</i>	
Kurtosis:	12.596	- MORE	concentra	ated ther	n normal
Dimension 6					
Expected value:	0.241	(0.111	-		7
Standard deviation:	0.448	(0.114)	- normali	zed to c	luster range)
SKewness:	1.248	- RIGHI	side lor	nger Had thar	normo7
RUPLOSIS:	1.700	- MORE	Concentra	ilea ther	ΠΟΓΙΙΙΑΙ
Expected value:	-0 546				
Standard deviation:	0.779	(0 195	- normala	ized to a	Juster range)
skewness:	0.879	- RTGHT	r side lor	naer	
Kurtosis:	0.357	- MORE	concentra	ited ther	normal
Dimension 8					
Expected value:	0.688				
Standard deviation:	0.067	(0.039	- normali	ized to c	luster range)
Skewness:	2.025	- RIGHT	r side lor	nger	_
Kurtosis:	12.266	- MORE	concentra	ited ther	normal
#22555	T)///				
#CORRELATION_MATR	1X#	0 017	0 214	0 004	0 100 0 075
1.000 0.281	-0.033 ·	-U.UL3	-0.214	-0.024	0.182 0.275
0.281 1.000	0.043 -	-0.091 0 201	-0.15/	0.104	
	1.000 0 201	0.291 1 000	0.005	-0.200	-0.306 -0.454
-0.013 $-0.091-0.214$ -0.157	0.291	0 428	1 NNN	-0.030 -0 120	-0.500 -0.454 -0.554 -0.802
-0.024 0 104	0.206	-0.858	-0.420	1_000	0.608 0.443
0.182 -0.036	0.634	-0,306	-0.554	0.608	1.000 0.520
0.275 0.552	0.029	-0.454	-0.803	0.443	0.520 1.000

#GENERAL INFO#					
Number of points:	151502				
Rest noint value:	0 95	3			
Best noint narameters	0.92	, 3 0.26	56 0.082	1,233	
	1.030	0.01	18 -1.503	0.646	
	1,050	0.01	1.505	01010	
#SAMPLITNG_RANGES#					
Dim Min Max	Rai	nae			
1 0.692 1	.618	0.926			
2 0.184 0	560	0 376			
3 -1999 2	000	3 999			
4 0 539 1	888	1 349			
5 -1 371 1	975	3 346			
6 -1 996 2	000	3 996			
7 -2 000 2	000	4 000			
8 _0.073 1	657	1 730			
0 0.075 1	.057	1.750			
#MOMENTS#					
Dimension 1					
Expected value	0 947				
Standard deviation:	0.073	(0 079 -	- normalized	d to cluster	ranne)
skownoss.	0.874	- RTCHT	side longe	r	runge)
Kurtosis:	2 378	- MORE (concentrate	d then normal	
Dimonsion 2	2.570	- MORE C			
Expected values	0 272				
Expected value.	0.272	(0 077	normaliza	d to cluston	
Stanuaru ueviation:	0.029	(0.077 -		LO CIUSLEI	range)
SKewness:	1.841	- KIGHI	side longel	r d than nama7	
KUPTOSIS:	9.076	- MORE C	concentrated	a then normal	
Dimension 3	0 400				
Expected value:	0.488	(0. 221			
Standard deviation:	0.925	(0.231 -	- normaiized	a to cluster	range)
Skewness:	-0.421	- LEFI S	side longer		
KURTOSIS	-0.635	- LESS C	concentrated	d then normal	
Dimension 4					
Expected value:	1.195	(0.450		, -	
Standard deviation:	0.215	(0.159 -	- normalized	d to cluster	range)
Skewness:	-0.192	- LEFT S	side longer		
Kurtosis:	-0.494	- LESS (concentrated	d then normal	
Dimension 5					
Expected value:	0.609				
Standard deviation:	0.500	(0.149 -	- normalized	d to cluster	range)
Skewness:	-0.687	- LEFT S	side longer		
Kurtosis:	0.452	- MORE C	concentrated	d then normal	
Dimension 6					
Expected value:	0.295				
Standard deviation:	0.765	(0.191 -	- normalized	d to cluster	range)
Skewness:	0.084	- RIGHT	side longe	r	
Kurtosis:	-0.654	- LESS C	concentrated	d then normal	
Dimension 7					
Expected value:	-0.295				
Standard deviation:	0.983	(0.246 -	- normalized	d to cluster	range)
Skewness:	0.391	- RIGHT	side longe	r	-
Kurtosis:	-0.692	- LESS C	concentrated	d then normal	
Dimension 8					
Expected value:	0.754				
Standard deviation:	0.138	(0.080 -	- normalized	d to cluster	range)
Skewness:	0.281	- RIGHT	side longel	r	2
Kurtosis:	2.220	- MORE C	concentrated	d then normal	
#CORRELATION_MATR1	TX#				
1.000 0.252	-0.007	0.082	-0.046 -0	0.122 0.16	0 0.185
0.252 1.000	0.102 .	-0.115	-0.004 0	0.177 -0.17	5 0.643
-0.007 0.102	1.000	0.555	0.358 -0	0.109 0.33	1 -0.232
0.082 -0.115	0.555	1.000	0.330 -0	0.858 -0.13	0 -0.322
-0.046 -0.004	0.358	0.330	1.000 -0	0.164 -0.37	1 -0.621
-0.122 0.177	-0.109 -	-0.858	-0.164	1.000 0.31	0 0.202
0.160 -0.175	0.331 -	-0.130	-0.371 (0.310 1.00	0 0.186
0.185 0.643	-0.232 ·	-0.322	-0.621 (0.202 0.18	6 1.000

#GENERAL TNEO#					
Number of points:	72442				
Ract point value	0 07	1			
Best point value.	. 0.97	7 0 '		n 1101	
Best point parameters	: 0.93	0.2	261 0.060) 1.161	
	-0.862	2 0.7	/61 -2.000) -0.687	
#SAMPLING_RANGES#					
Dim Min Max	Rai	nge			
1 0.770 1	.224	0.454			
2 0.188 0	.371	0.183			
3 -1.132 1	. 989	3.121			
4 0.816 1	. 363	0.547			
5 -1.156 -0	. 410	0.746			
6 0.274 2	000	1.726			
7 -2 000 1	818	3 818			
8 _1 111 _0	166	0 645			
0 1.111 0	. 400	0.045			
#MOMENTS#					
#MOMENTS#					
	0.044				
Expected value:	0.944	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	- ·		
Standard deviation:	0.017	(0.037)	- normaiize	ea to cluster	range)
Skewness:	1.395	- RIGHT	r side longe	er .	_
Kurtosis:	15.168	- MORE	concentrate	ed then norma	7
Dimension 2					
Expected value:	0.260				
Standard deviation:	0.005	(0.025	- normalize	ed to cluster	range)
skewness:	2.587	- RIGHT	⊤ side longe	er	-
Kurtosis:	50.123	- MORE	concentrate	ed then norma	7
Dimension 3					
Expected value:	0.611				
Standard deviation:	0 351	(0 112	- normalize	nd to cluster	range)
skawnass'	_0 227	_ / FET	side londer		runge)
Vurtosis:	0.227		concentrate	od than narma	7
Dimonsion 1	0.570	- MORE	concentrate	eu litett tiotilla	1
Dimension 4	1 1 1 0				
Expected value:	1.148	(0.000	7		
Standard deviation:	0.048	(0.088)	- normaiize	ea to cluster	range)
Skewness:	-1.285	- LEFT	side longer		_
Kurtosis:	1.792	- MORE	concentrate	ed then norma	7
Dimension 5					
Expected value:	-0.787				
Standard deviation:	0.057	(0.076	- normalize	ed to cluster	range)
Skewness:	-0.473	- LEFT	side longer	-	-
Kurtosis:	1.642	- MORE	concentrate	ed then norma	7
Dimension 6					
Expected value:	1.082				
Standard deviation:	0.275	(0.159)	- normalize	ed to cluster	range)
skewness'	1 049	- RTGHT	r side longe	r	. ungej
Kurtosis:	0 547	- MORE	concentrate	nd than norma	7
Dimonsion 7	0.547	MORE	concentrate	eu chen norma	'
Expected values	0 016				
Expected value.	-0.010	(0, 170)		ad to alwatar	
Standard deviation:	0.682	(0.179)		ea to cluster	range)
Skewness:	0.189	- RIGHI	i stae longe	er	7
KURTOSIS:	-0.445	- LESS	concentrate	ea then norma	1
Dimension 8					
Expected value:	-0.703				
Standard deviation:	0.020	(0.032	- normalize	ed to cluster	range)
Skewness:	-0.599	- LEFT	side longer	n 	_
Kurtosis:	26.344	- MORE	concentrate	ed then norma	7
#CORRELATION_MATR.	TX#				
1.000 0.164	0.061	-0.414	-0.162	0.401 0.3	59 0.052
0.164 1.000	-0.146	-0.135	-0.139	0.022 -0.1	72 -0.555
0.061 -0.146	1.000	0.073	0.378	0.563 0.7	12 -0.122
-0.414 -0.135	0.073	1.000	0.076 -	-0.705 -0.5	45 0.021
-0,162 -0.139	0.378	0.076	1.000 -	-0.050 0.4	34 -0.594
0.401 0.022	0.563	-0.705	-0.050	1.000 0.8	37 0.071
0.359 - 0.172	0.712	-0.545	0.434	0.837 1 0	00 - 0.140
0,052 -0.555	-0.122	0.021	-0.594	0.071 -0.1	40 1.000

#GENERAL_INFO#					
Number of points:	238121				
Best point value:	0.97	1			
Best point parameters	: 0.93	7 0.2	261 0.048	8 1.15	9
	-0.860) 0.7	760 -1.99	9 -0.688	8
#SAMPLING_RANGES#					
Dim Min Max	Rai	ige			
1 0.741 1	.339	0.598			
2 0.184 0	.402	0.218			
3 -1.998 2	.000	3.998			
4 0.510 1 5 1.206 1	.007	1.1//			
5 -1.200 1	.090	2.302			
7 -2 000 2	.000	A 000			
8 -1.235 -0	385	0.850			
0 11255 0		0.050			
#MOMENTS#					
Dimension 1					
Expected value:	0.943				
Standard deviation:	0.028	(0.046	- normalize	ed to clus	ster range)
Skewness:	1.101	- RIGHT	🗆 side longe	er	
Kurtosis:	11.107	- MORE	concentrate	ed then no	orma1
Dimension 2					
Expected value:	0.261	~~ ~ ~ ~		, -	
Standard deviation:	0.009	(0.040	- normalize	ed to clus	ster range)
Skewness:	1.265	- RIGHT	side longe	er	7
Kurtosis:	12.067	- MORE	concentrate	ed then no	ormal
Dimension 3	0 257				
Standard doviation:	0.337	(0 150	- normaliz	nd to clu	stor range)
Stanuaru uevratron.	_0.000		- normarize		ster range)
Kurtosis:	-0.024	- LEFI	concentrate	ad than n	ormal
Dimension 4	0.150	MORL	concentrate	eu chen no)
Expected value:	1,132				
Standard deviation:	0.100	(0.085	- normalize	d to clus	ster range)
Skewness:	-1.505	- IFFT	side lonael	r	, ange,
Kurtosis:	5.852	- MORE	concentrate	ed then no	ormal
Dimension 5		-			
Expected value:	-0.727				
Standard deviation:	0.207	(0.090	- normalize	ed to clus	ster range)
Skewness:	2.894	- RIGHT	🗆 side longe	er	
Kurtosis:	9.762	- MORE	concentrate	ed then no	orma1
Dimension 6					
Expected value:	0.934	(0.107		7	
Standard deviation:	0.445	(0.127	- normalize	ed to clus	ster range)
Skewness:	-0.519	- LEFT	side longel	r ad than m	- 1 m - 7
RUILOSIS: Dimension 7	2.338	- MUKE	concentrate	eu chen no) i iid i
Expected value	_0 \$72				
Standard deviation:	1 000	(0 250	- normaliza	ad to clus	star ranga)
skewness:	0.836	– RTGH1	side lona	r	ster range)
Kurtosis:	-0.183	- LESS	concentrate	ed then no	orma1
Dimension 8					
Expected value:	-0.717				
Standard deviation:	0.046	(0.055	- normalize	ed to clus	ster range)
Skewness:	-2.198	- LEFT	side longer	r	
Kurtosis:	7.967	- MORE	concentrate	ed then no	ormal
#coc	T \ <i>(</i> //				
#CORRELATION_MATR	1X#	0 202	0 022	0 170	0.200 0.010
1.000 0.130	-0.018	-0.203	-0.023	0.178	0.286 - 0.048
0.130 1.000	-U.1/3 ·	-U.U23 0 E22	0.015	-U.130 ·	-0.290 -0.561
-0.010 -0.173	0 532	1 000	0.119	-0 631	0.422 $0.030-0.272$ -0.122
-0.203 -0.023 -0.023 -0.015	0.333	1 220	1 000	-0.031 .	0.272 -0.120
0.178 -0.136	0.169	-0.631	-0.531	1.000	0.376 0.490
0.286 -0.290	0.422	-0.272	0.509	0.376	1.000 -0.218
-0.048 -0.561	0.030	-0.126	-0.733	0.490	-0.218 1.000

#GENERAL INFO#					
Number of points:	97046				
Best point value:	0.97	1			
Best point parameters	: 0.93	7 0.2	261 0.0	62 1.	161
	-0.86	1 0.7	760 -2.0	00 -0.	687
#SAMPLING_RANGES#					
Dim Min Max	Ra	nge			
1 0.722 1	.469	0.747			
2 0.184 0	.479	0.295			
3 -1.999 2	.000	3.999			
4 0.510 1	.773	1.263			
5 -1.301 1	.660	2.961			
6 -1.997 2	.000	3.997			
7 -2.000 2	.000	4.000			
8 -1.545 0	.065	1.610			
//					
#MOMENTS#					
Dimension 1	0 057				
Expected Value:	0.957	(0.000			7
Standard deviation:	0.073	(0.098	- normaii	zea το c	Tuster range)
SKEWNESS:	0.838	- KIGH	i stae ton	ger tod thom	n o 5m o 7
KULLOSIS:	2.055	- MORE	concentra	Lea Lhen	ΠΟΓΊΙΙΑ Ι
Expected values	0 267				
Expected value:	0.207	(0 002	- normali	zad to c	Justar range)
Stanuaru uevratron.	1 561		- normarr Feido lon	zeu lo l	Tuster Tange)
Skewness. Vurtosis:	1.504		concentra	yei tod thon	normal
Dimension 3	4.550	- MORE	concentra	Leu Liien	normar
Expected value	0 357				
Standard deviation	0.943	(0 236	- normali	zed to c	luster range)
skewness:	-0.294	- 1 FFT	side lona	er	ruseer runge,
Kurtosis:	-0.801	- 1 ESS	concentra	ted then	normal
Dimension 4	0.002				
Expected value:	1.144				
Standard deviation:	0.182	(0.144	- normali	zed to c	luster range)
Skewness:	-0.375	- LEFT	side long	er	5 -
Kurtosis:	0.057	- MORE	concentra	ted then	normal
Dimension 5					
Expected value:	-0.429				
Standard deviation:	0.395	(0.134	- normali	zed to c	luster range)
Skewness:	1.190	- RIGH	⊤side lon	ger	_
Kurtosis:	1.792	- MORE	concentra	ted then	norma1
Dimension 6					
Expected value:	0.588				-
Standard deviation:	0.773	(0.194	- normali	zed to c	luster range)
Skewness:	-0.400	- LEFT	side long	er	7
Kurtosis	-0.142	- LESS	concentra	ted then	normal
Dimension /	0 241				
Expected Value:	-0.341	(0 200		mad the m	Tuston nonco)
Standard deviation:	1.073	(0.208	- normann Feide len	zea το c	Tuster range)
SKEWNESS:	0.337	- KIGH	i side idii	yer tod thom	norma ⁷
Ruitosis.	-0.937	- LESS	concentra	Leu Liien	ποτιπατ
Expected value	_0 781				
Standard deviation:	0.116	(0 072	- normali	zed to c	Justar ranga)
skewness	-1 222	- I FFT	side lond	zeu lo c	ruster range)
Kurtosis:	3 038	- MORE	concentra	ted then	normal
	5.050	HORL	20		
#CORRELATION MATE	IX#				
1.000 0.174	-0.118	-0.050	-0.038	-0.053	0.195 -0.109
0.174 1.000	-0.080	-0.115	0.373	0.003	-0.242 -0.849
-0.118 -0.080	1.000	0.679	-0.012	-0.110	0.149 0.076
-0.050 -0.115	0.679	1.000	0.184	-0.711	-0.213 0.002
-0.038 0.373	-0.012	0.184	1.000	-0.482	0.194 -0.635
-0.053 0.003	-0.110	-0.711	-0.482	1.000	0.288 0.259
0.195 -0.242	0.149	-0.213	0.194	0.288	1.000 0.069
-0.109 -0.849	0.076	0.002	-0.635	0.259	0.069 1.000



Rys. 6.8 Repozytorium REP11. Metoda AH: etap F. Histogram DBH



Rys. 6.9 Repozytorium TEN_S1. Metoda AH: etap F. Histogram DBH



Rys. 6.10 Repozytorium ALL_S1. Metoda AH: etap F. Histogram DBH



Rys. 6.11 Repozytorium REP22. Metoda AH: etap F. Histogram DBH



Rys. 6.12 Repozytorium TEN_S2. Metoda AH: etap F. Histogram DBH



Rys. 6.13 Repozytorium ALL_S2. Metoda AH: etap F. Histogram DBH
Z połączonych informacji przedstawionych w ramach modułu analizy próbkowania wynika, że dla rozpatrywanego ośmiowymiarowego przypadku rzeczywistego wykorzystanie do wyznaczenia przedziałów niepewności repozytorium zawierającego próbkowanie przestrzeni optymalizacyjnej z jednego przebiegu algorytmu genetycznego (REP11 oraz REP22) może prowadzić do ich znacznego niedoszacowania. Potencjalna niedokładność estymacji niepewności może być tym większa, im mniejszy wpływ ma dany parametr na wartość $\chi^2(ME)$. Dla szerokich przedziałów niepewności zmiana może być bardzo istotna.

Wyniki modułu analizy próbkowania nie sugerują istotnych różnic pomiędzy repozytoriami zbiorczymi o różnej minimalnej odległości między punktami (TEN versus ALL) z punktu widzenia niedokładności wyznaczenia niepewności. Tę potencjalnie zaskakującą hipotezę potwierdzają rezultaty modułu wyznaczenia niepewności.

6.3. Moduł wyznaczenia niepewności

Celem modułu wyznaczenia niepewności jest możliwie dokładne oszacowanie przedziałów niepewności wartości poszczególnych parametrów dla każdego z rozważanych dwóch alternatywnych wektorów elementów macierzowych. Podobnie jak w poprzednim punkcie rezultaty zostały umieszczone oddzielnie dla basenu przyciągania każdego z rozwiązań równorzędnych, tj. w postaci dwóch trójek.

Rys. 6.14 – Rys. 6.19 przedstawiają wyniki etapu H Metody AH dla przetwarzanych zbiorów punktów w omawianym przypadku rzeczywistym.

Porównanie wyników ujawnia istotne różnice w oszacowaniu niepewności w obrębie rozważanych trójek repozytoriów próbkowania przestrzeni. Zmiany te są analogiczne dla obu trójek. Jednocześnie rozbieżności potwierdzają wnioski sformułowane w punkcie 6.2. w oparciu o rezultaty modułu analizy próbkowania.

Dla rozpatrywanego ośmiowymiarowego przypadku rzeczywistego, użycie repozytoriów zawierających próbkowanie przestrzeni optymalizacyjnej tylko z jednego przebiegu algorytmu genetycznego (REP11 oraz REP22) prowadzi do niedoszacowania przedziałów niepewności. Jest to szczególnie zauważalne, gdy przedział niepewności wartości danego parametru jest szeroki.

#THRESHOLD_PLANE_INFO#
Threshold Plane : 1.953 Best point value: 0.953
#UNCERTAINTIES_DIRECTLY_FROM_REPOSITORY#
No Value Uncertainty [min, max] Range Uncertainty[%]
1 0.923 +0.847(-0.076), +1.027(+0.104) 0.180 -8.2%, +11.3%
2 0.266 +0.238(-0.028), +0.298(+0.032) 0.060 -10.5%, +12.0%
3 0.201 -0.954(-1.155), +1.995(+1.794) 2.949 -574.6%,+892.5%
4 1.250 +0.775(-0.475), +1.554(+0.304) 0.779 -38.0%, +24.3%
5 1.018 +0.268(-0.750), +1.196(+0.178) 0.928 -73.7%, +17.5%
6 / 0.001 / -0.757(-0.758), +2.000(+1.999) / 2.757 /-75800.0%,+199900.0%
7 -1.359 -1.996(-0.637), +1.997(+3.356) 3.993 -46.9%,+246.9%
8 0.651 +0.563(-0.088), +0.866(+0.215) 0.303 -13.5%, +33.0%
#UNCERTAINTIES_AFTER_FURTHER_ESTIMATIONS#
No Value Uncertainty [min, max] Range Uncertainty[%]
1 0.923 +0.847(-0.076), +1.027(+0.104) 0.180 -8.2%, +11.3%
2 0.266 +0.238(-0.028), +0.298(+0.032) 0.060 -10.5%, +12.0%
3 / 0.201 / -0.959(-1.160), +1.995(+1.794) / 2.954 /-577.0%,+892.5%
4 1.250 +0.775(-0.475), +1.554(+0.304) 0.779 -38.0%, +24.3%
5 / 1.018 / +0.268(-0.750), +1.196(+0.178) / 0.928 / -73.7%, +17.5%
6 / 0.001 / -0.758(-0.759), +2.000(+1.999) / 2.758 /-75900.0%,+199900.0%
7 -1.359 -2.000(-0.641), +1.997(+3.356) 3.997 -47.2%,+246.9%
8 0.651 +0.563(-0.088), +0.866(+0.215) 0.303 -13.5%, +33.0%
#UNCERTAINTIES_ESTIMATION_ACCURACY#
Spread of FrontLiners : 9.031
furthestEFL (spreadEFL): 9.998 (+8.045)
furthestAFL (spreadAFL): 0.967 (-0.986)

Rys. 6.14 Repozytorium REP11. Metoda AH: etap H. Wydruk z programu ScanRep

#THRESHOLD_PLANE_INFO#
Threshold Plane : 1.953 Best point value: 0.953
#UNCERTAINTIES_DIRECTLY_FROM_REPOSITORY#
NO Value Uncertainty [min, max] Range Uncertainty[%]
1 0.923 +0.839(-0.084), +1.040(+0.117) 0.201 -9.1%, +12.7%
2 0.267 +0.231(-0.036), +0.314(+0.047) 0.083 -13.5%, +17.6%
3 0.104 -1.464(-1.568), +2.000(+1.896) 3.464 -1507.7%,+1823.1%
4 1.235 +0.801(-0.434), +1.688(+0.453) 0.887 -35.1%, +36.7%
5 / 1.033 / +0.178(-0.855), +1.496(+0.463) / 1.318 / -82.8%, +44.8%
6 0.021 -1.179(-1.200), +1.982(+1.961) 3.161 -5714.3%,+9338.1%
7 -1.495 -2.000(-0.505), +1.999(+3.494) 3.999 -33.8%,+233.7%
8 0.648 +0.415(-0.233), +0.881(+0.233) 0.466 -36.0%, +36.0%
#UNCERTAINTIES_AFTER_FURTHER_ESTIMATIONS#
No Value Uncertainty [min, max] Range Uncertainty[%]
1 0.923 +0.839(-0.084), +1.040(+0.117) 0.201 -9.1%, +12.7%
2 0.267 +0.231(-0.036), +0.314(+0.047) 0.083 -13.5%, +17.6%
3 0.104 -1.464(-1.568), +2.000(+1.896) 3.464 -1507.7%,+1823.1%
4 1.235 +0.801(-0.434), +1.688(+0.453) 0.887 -35.1%, +36.7%
5 / 1.033 / +0.178(-0.855), +1.496(+0.463) / 1.318 / -82.8%, +44.8%
6 / 0.021 / -1.179(-1.200), +1.982(+1.961) / 3.161 /-5714.3%,+9338.1%
7 -1.495 -2.000(-0.505), +1.999(+3.494) 3.999 -33.8%,+233.7%
8 0.648 +0.415(-0.233), +0.881(+0.233) 0.466 -36.0%, +36.0%
#UNCERTAINTIES_ESTIMATION_ACCURACY#
Spread of FrontLiners : 9.040
furthestEFL (spreadEFL): 10.000 (+8.047)
furthestAFL (spreadAFL): 0.960 (-0.993)

Rys. 6.15 Repozytorium TEN1_S1. Metoda AH: etap H. Wydruk z programu ScanRep

#THRESHOLD_PLANE_INFO#	
Threshold Plane : 1.953 Best point value: 0.953	
#UNCERTAINTIES_DIRECTLY_FROM_REPOSITORY#	
No Value Uncertainty [min, max] Range Uncertainty[%]	
1 0.923 +0.822(-0.101), +1.061(+0.138) 0.239 -10.9%, +15.0%	
2 0.266 +0.231(-0.035), +0.322(+0.056) 0.091 -13.2%, +21.1%	
3 0.082 -1.862(-1.944), +2.000(+1.918) 3.862 -2370.7%,+2339.0%	6
4 1.233 +0.679(-0.554), +1.695(+0.462) 1.016 -44.9%, +37.5%	
5 1.030 +0.191(-0.839), +1.598(+0.568) 1.407 -81.5%, +55.1%	
6 0.018 -1.358(-1.376), +2.000(+1.982) 3.358 -7644.4%,+11011	1%
7 -1.503 -2.000(-0.497), +1.998(+3.501) 3.998 -33.1%,+232.9%	
8 0.646 +0.367(-0.279), +0.878(+0.232) 0.511 -43.2%, +35.9%	
#UNCERTAINTIES_AFTER_FURTHER_ESTIMATIONS#	
No Value Uncertainty [min, max] Range Uncertainty[%]	
1 0.923 +0.822(-0.101), +1.061(+0.138) 0.239 -10.9%, +15.0%	
2 / 0.266 / +0.231(-0.035), +0.322(+0.056) / 0.091 / -13.2%, +21.1%	
3 0.082 -1.862(-1.944), +2.000(+1.918) 3.862 -2370.7%,+2339.0%	6
4 1.233 +0.679(-0.554), +1.695(+0.462) 1.016 -44.9%, +37.5%	
5 / 1.030 / +0.191(-0.839), +1.598(+0.568) / 1.407 / -81.5%, +55.1%	
6 0.018 -1.358(-1.376), +2.000(+1.982) 3.358 -7644.4%,+11011	1%
7 -1.503 -2.000(-0.497), +2.000(+3.503) 4.000 -33.1%,+233.1%	
8 0.646 +0.367(-0.279), +0.878(+0.232) 0.511 -43.2%, +35.9%	
#UNCERTAINTIES_ESTIMATION_ACCURACY#	
Spread of FrontLiners : 9.047	
furthestEFL (spreadEFL): 10.000 (+8.047)	
furthestAFL (spreadAFL): 0.953 (-1.000)	

Rys. 6.16 Repozytorium ALL_S1. Metoda AH: etap H. Wydruk z programu ScanRep

#THRESHOLD_	PLANE_INFO#	
Threshold Plane	: 1.971	Best point value: 0.971
#UNCERTAINT	TES_DIRECTLY_FROM	M_REPOSITORY#
No Value	Uncertainty	y [min, max] Range Uncertainty[%]
1 0.937	+0.849(-0.088),	, +1.078(+0.141) 0.229 -9.4%, +15.0%
2 0.261	+0.239(-0.022),	, +0.289(+0.028) 0.050 -8.4%, +10.7%
3 0.060	-1.039(-1.099),	, +1.983(+1.923) 3.022 -1831.7%,+3205.0%
4 1.161	+0.900(-0.261),	, +1.304(+0.143) 0.404 -22.5%, +12.3%
5 -0.862	-1.016(-0.154),	, -0.517(+0.345) 0.499 -17.9%, +40.0%
6 0.761	+0.424(-0.337),	, +1.973(+1.212) 1.549 -44.3%,+159.3%
7 / -2.000 /	-2.000(+0.000),	, +1.691(+3.691) 3.691 +0.0%,+184.6%
8 -0.687	-0.813(-0.126),	, -0.582(+0.105) 0.231 -18.3%, +15.3%
#UNCERTAINT	IES_AFTER_FURTHE	R_ESTIMATIONS#
No Value	Uncertainty	y [min, max] Range Uncertainty[%]
1 0.937	+0.849(-0.088),	, +1.078(+0.141) 0.229 -9.4%, +15.0%
2 0.261	+0.239(-0.022),	, +0.289(+0.028) 0.050 -8.4%, +10.7%
3 0.060	-1.041(-1.101),	, +1.988(+1.928) 3.029 -1835.5%,+3213.4%
4 1.161	+0.900(-0.261),	, +1.308(+0.147) 0.408 -22.5%, +12.6%
5 -0.862	-1.016(-0.154).	0.517(+0.345) 0.499 -17.9%, +40.0%
6 0.761	+0.421(-0.340).	. +1.973(+1.212) 1.552 -44.6%.+159.3%
7 / -2.000 /	-2.000(+0.000).	. +1.692(+3.692) 3.692 +0.0%.+184.6%
8 -0.687	-0.814(-0.127)	-0.576(+0.111) 0.238 -18.5%, +16.2%
- , ,	,	,, ,, ,,
#UNCERTAINT	IES_ESTIMATION_AC	CCURACY#
Spread of Front	Liners : 9.001	1
furthestEFL (sp	readEFL): 9.975	5 (+8.004)
furthestAFL (sp	oreadAFL): 0.974	4 (-0.997)
	-	

Rys. 6.17 Repozytorium REP22. Metoda AH: etap H. Wydruk z programu ScanRep

#THRESHOLD_I	PLANE_INFO#	
Threshold Plane	: 1.971	Best point value: 0.971
#UNCERTAINT	IES_DIRECTLY_FROM	M_REPOSITORY#
No Value	Uncertainty	y [min, max] Range Uncertainty[%]
1 0.937	+0.836(-0.101),	, +1.078(+0.141) 0.242 -10.8%, +15.0%
2 0.261	+0.230(-0.031),	, +0.301(+0.040) 0.071 -11.9%, +15.3%
3 0.048	-1.828(-1.876),	, +1.999(+1.951) 3.827 -3908.3%,+4064.6%
4 1.159	+0.726(-0.433),	, +1.394(+0.235) 0.668 -37.4%, +20.3%
5 -0.860	-1.049(-0.189),	, -0.149(+0.711) 0.900 -22.0%, +82.7%
6 0.760	-0.489(-1.249),	, +2.000(+1.240) 2.489 -164.3%,+163.2%
7 -1.999	-2.000(-0.001),	, +1.988(+3.987) 3.988 -0.1%,+199.4%
8 -0.688	-0.895(-0.207),	, -0.573(+0.115) 0.322 -30.1%, +16.7%
#UNCERTAINT	IES_AFTER_FURTHER	R_ESTIMATIONS#
No Value	Uncertainty	y [min, max] Range Uncertainty[%]
1 0.937	+0.836(-0.101),	, +1.078(+0.141) 0.242 -10.8%, +15.0%
2 0.261	+0.230(-0.031),	, +0.301(+0.040) 0.071 -11.9%, +15.3%
3 0.048	-1.828(-1.876),	, +2.000(+1.952) 3.828 -3908.3%,+4066.7%
4 1.159	+0.726(-0.433),	, +1.394(+0.235) 0.668 -37.4%, +20.3%
5 / -0.860 /	-1.049(-0.189),	, -0.149(+0.711) 0.900 -22.0%, +82.7%
6 0.760	-0.489(-1.249),	, +2.000(+1.240) 2.489 -164.3%,+163.2%
7 -1.999	-2.000(-0.001),	, +1.988(+3.987) 3.988 -0.1%,+199.4%
8 -0.688	-0.895(-0.207),	, -0.573(+0.115) 0.322 -30.1%, +16.7%
#UNCERTAINT	IES_ESTIMATION_AC	CCURACY#
Spread of Front	Liners : 9.024	4
furthestEFL (spi	readEFL): 9.999	9 (+8.028)
furthestAFL (spi	readAFL): 0.975	5 (-0.996)

Rys. 6.18 Repozytorium TEN1_S2. Metoda AH: etap H. Wydruk z programu ScanRep

#THRESHOLD PLANE TNEO#
"Threshold plane · 1 071 Post point value · 0 071
The short France . 1.571 Best point value. 0.571
#UNCERTAINTIES DIRECTLY FROM REPOSITORY#
"Oreconstruction of the second
1 + 0.027 + .0.836(-0.101) + 1.000(-0.153) + 0.254 + .10.84 + .16.34
1 + 0.551 + 10.050(-0.101), +1.050(+0.155) + 0.254 + 10.000, +10.57
2 + 0.201 + 10.227 (-0.034), + 0.005 (-0.044) + 0.006 (-1.5.0%, +10.5%)
5 + 0.002 + -1.300 + 2.040, $+1.337 + 1.337 + 3.303 + 3.303 + 3.303 + 3.24 + 2.8$
4 + 1.101 + 10.720(-0.435), +1.430(+0.277) + 0.712 + -57.5%, +23.5%
5 - 0.001 - 1.134 - 0.235, $-0.134 + 0.707 - 1.000 - 54.0%$, $+02.1%$
$0 \mid 0.760 \mid -0.353(-1.315), +1.999(+1.239) \mid 2.334 \mid -1/3.0\%, +105.0\%$
/ -2.000 / -2.000 +0.000), +1.988 +3.988 / 3.988 / +0.0%,+199.4%
8 -0.687 -0.895(-0.208), -0.515(+0.172) 0.380 -30.3%, +25.0%
<i>и</i>
#UNCERTAINTIES_AFTER_FURTHER_ESTIMATIONS#
No Value Uncertainty [min, max] Range Uncertainty[%]
1 0.937 +0.836(-0.101), +1.090(+0.153) 0.254 -10.8%, +16.3%
2 0.261 +0.227(-0.034), +0.305(+0.044) 0.078 -13.0%, +16.9%
3 0.062 -1.987(-2.049), +2.000(+1.938) 3.987 -3305.4%,+3125.8%
4 1.161 +0.726(-0.435), +1.438(+0.277) 0.712 -37.5%, +23.9%
5 -0.861 -1.154(-0.293), -0.154(+0.707) 1.000 -34.0%, +82.1%
6 0.760 -0.555(-1.315), +2.000(+1.240) 2.555 -173.0%,+163.2%
7 -2.000 -2.000(+0.000), +1.988(+3.988) 3.988 +0.0%,+199.4%
8 -0.687 -0.895 (-0.208)0.515 (+0.172) 0.380 -30.3%, +25.0%
#UNCERTAINTIES_ESTIMATION_ACCURACY#
Spread of FrontLiners : 9.026
furthestEFL (spreadEFL): 9.998 (+8.027)
furthestAFL (spreadAFL): 0.972 (-0.999)

Rys. 6.19 Repozytorium ALL_S2. Metoda AH: etap H. Wydruk z programu ScanRep

Zestawienie rezultatów dla repozytoriów zbiorczych o różnej minimalnej odległości między punktami (TEN vs. ALL) potwierdza brak istotnych rozbieżności w estymacji niepewności. Szersze przedziały niepewności otrzymano w oparciu o repozytoria ALL_S1 oraz ALL_S2. Różnice są co najwyżej rzędu drugiej cyfry znaczącej.

Zastosowanie procedury dalszej estymacji niepewności nie przyniosło istotnych zmian dla repozytoriów zbiorczych. Różnica jest co najwyżej rzędu trzeciej cyfry znaczącej. Wpływ procedury FUE na szacowanie niepewności jest więc w tym przypadku zaniedbywalny. Oznacza to również, że próbkowanie przestrzeni można uznać za wystarczająco gęste.

W oparciu o całość informacji przedstawionych w tym rozdziale, za najbardziej wiarygodne należy uznać przedziały niepewności wartości parametrów określone w oparciu o repozytoria ALL_S1 oraz ALL_S2. Jednocześnie potencjalna niedokładność wyznaczenia niepewności ma charakter zaniedbywalny ze względu na powszechnie stosowane zaokrąglanie granic niepewności do dwóch cyfr znaczących.

W ramach prezentowanego przypadku rzeczywistego porównano działanie Metody AH dla trzech typów zbiorów punktów. Wyniki uzyskane dla repozytoriów pojedynczych okazały się niesatysfakcjonujące z punktu widzenia dokładności wyznaczenia niepewności.

Zestawienie rezultatów dla dwóch typów repozytoriów zbiorczych prowadzi do interesujących wniosków. Zastosowanie w repozytoriach ALL_S1 oraz ALL_S2 o rząd wyższej wartości argumentu procedury MDR okazało się mniej istotne niż fakt, że zbiory TEN_S1 oraz TEN_S2 stanowią konkatenacje jedynie dziesięciu (zamiast czterdziestu) repozytoriów pojedynczych. Repozytoria ALL_S1 oraz ALL_S2, ze względu na dziesięciokrotnie większą minimalną odległość między punktami, są kilkukrotnie mniejsze, ale określone na ich podstawie przedziały niepewności wartości parametrów są co najmniej równie dobre. Jednocześnie różnice nie są duże, co jest istotne z punktu widzenia przyszłego stosowania Metody AH do analizy danych z pomiarów wzbudzeń kulombowskich.

Kilkadziesiąt uruchomień opisanego algorytmu genetycznego zapewnia stochastyczny skan całej przestrzeni optymalizacyjnej, a zebrane próbkowanie stanowi dobrą podstawę do oszacowania niepewności z wystarczającą dokładnością. Konieczna do tego liczba wywołań funkcji celu jest akceptowalna oraz wyraźnie niższa niż dla metod klasycznych.

7. Podsumowanie

Przedmiotem niniejszej pracy było zaprojektowanie i implementacja nowej metody oceny jakości wyników z eksperymentów fizyki jądrowej. Wykorzystywane dotychczas do analizy danych z pomiarów wzbudzeń kulombowskich oprogramowanie powstało ponad 30 lat temu. Zastosowana w nim technika poszukiwania minimum wielowymiarowej funkcji mierzącej odstępstwo danych eksperymentalnych od wyliczeń teoretycznych, oparta była na metodzie gradientowej.

W tej rozprawie do realizacji powyższego zadania zastosowano heurystyczną metodę optymalizacji globalnej. Użyty algorytm genetyczny z reprezentacją rzeczywistą zapewnia z jednej strony stochastyczny skan całej przestrzeni rozwiązań dopuszczalnych, a z drugiej pozwala znaleźć globalne minimum w akceptowalnym czasie.

Zasadniczym elementem pracy było opracowanie heurystycznej metody estymacji przedziałów niepewności wartości wyznaczonego optymalnego wektora parametrów. Zaproponowana Metoda AH realizuje to zadanie wykorzystując próbkowanie przestrzeni rozwiązań przez algorytm genetyczny w procesie optymalizacji. Nie są potrzebne żadne dodatkowe wywołania funkcji celu.

Metoda AH składa się z trzech modułów. W ramach modułu przetwarzania wstępnego z przetwarzanego repozytorium próbkowania przestrzeni wyodrębniany jest klaster punktów należących do basenu przyciągania optimum globalnego. Moduł analizy próbkowania ma na celu ocenę jakości tego zbioru pod kątem wykorzystania go do estymacji niepewności. Moduł wyznaczania niepewności odpowiada za oszacowanie przedziałów niepewności wartości poszczególnych parametrów.

Opis oraz działanie Metody AH zostały przedstawione przy użyciu przypadku testowego o znanym a priori rozwiązaniu. Jednocześnie porównano wykorzystanie próbkowania przestrzeni algorytmem genetycznym z innymi typami próbkowania: równomiernym oraz losowym, które są zazwyczaj wykorzystywane do analizy kształtu funkcji celu wokół znalezionego optimum. Wyniki potwierdziły możliwość użycia do tego celu repozytoriów próbkowania przestrzeni algorytmem genetycznym.

Zastosowanie zaimplementowanego algorytmu genetycznego oraz Metody AH do przypadku rzeczywistego zostało omówione na przykładzie analizy danych z pomiaru wzbudzenia kulombowskiego ¹³²Ba. Osiągnięte rezultaty potwierdziły przydatność opracowanych rozwiązań.

7.1. Kierunki dalszych badań

Wyniki pracy są obiecujące. Stworzone narzędzia dają możliwość weryfikacji wyników uzyskanych w przeprowadzonych już eksperymentach, jak również ułatwią analizę nowych. Poniżej przedstawiono propozycje dalszego rozwijania projektu.

Zaimplementowany w ramach etapu D Metody AH algorytm NBC (do grupowania repozytoriów punktów) daje pożądane rezultaty jedynie dla zbiorów danych, w których gęstość próbkowania jest podobna w obrębie poszczególnych klastrów. Aby usprawnić proces grupowania, można opracować system oparty na innych algorytmach uczenia nienadzorowanego, np. samoorganizujących się mapach [132] – [136] (ang. *Self-Organizing Maps*, SOM). Potencjalnym problemem jest walidacja poprawności działania takiej sieci neuronowej. Zazwyczaj stosowane metody oceny jakości grupowania mogą nie stanowić miarodajnej oceny, ze względu na specyfikę problemu. Z tego powodu w pierwszym okresie stosowania należałoby porównywać wyniki obu rozwiązań.

Jako płaszczyznę progową do wyznaczenia niepewności parametrów, na tym etapie projektu, przyjmuje się wartość $y_{PU} = y_{min} + 1.0$. Możliwym kierunkiem rozwoju jest opracowanie metody wyznaczania faktycznej liczby stopni swobody w danym eksperymencie, a następnie normalizowanie do tej estymaty stosowanej statystyki najmniejszych kwadratów oraz wyznaczanie wartości y_{PU} w oparciu o wartość kwantylu rozkładu χ^2 dla tej liczby stopni swobody.

Zawierający implementację Metody AH, program ScanRep działa w trybie konsolowym. Planowane jest utworzenie graficznego interfejsu użytkownika z wykorzystaniem platformy Qt [137].

7.2. Realizacja tez pracy

Zaproponowany system spełnił pokładane w nim nadzieje. Niedokładności wyznaczenia przedziałów niepewności w omówionych przypadkach testowym oraz rzeczywistym są poniżej stosowanego powszechnie zaokrąglenia do dwóch cyfr znaczących. Tym samym można stwierdzić, że tezy pracy:

- Możliwe jest oszacowanie niepewności wyznaczenia wartości elementów macierzowych w analizie danych z pomiarów wzbudzeń kulombowskich w oparciu jedynie o próbkowanie przestrzeni rozwiązań przez algorytm genetyczny.
- Dokładność wyznaczenia niepewności przedstawioną Metodą AH jest porównywalna z innymi metodami wykorzystywanymi do estymacji niepewności wartości parametrów, a czas niezbędny do wykonania obliczeń jest znacząco krótszy.

zostały udowodnione.

Bibliografia

Rozdział 2 – Wzbudzenia Kulombowskie

- [1] K. Alder, A. Winther: *Electromagnetic Excitation. Theory of Coulomb Excitation with Heavy Ions*, North-Holland Publishers Company, Amsterdam-Oxford, New York, 1975
- [2] K. Alder, A. Winther: *Coulomb Excitation: A Collection of Reprints*, Academic Publishing, New York, 1966
- [3] K. Alder, A. Bohr, T. Huus, B. Mottelson, A. Winther: *Study of Nuclear Structure by Electromagnetic Excitation with Accelerated Ions*, Review of Modern Physics, 28, 432, 1956; Erratum Review of Modern Physics, 30, 353, 1958
- [4] A. Bohr, B. R. Mottelson: *Struktura jądra atomowego. Deformacje jądrowe.* Tom 2, Wydawnictwo Naukowe PWN, Warszawa, 1984
- [5] P. Boutachkov, M. Górska, N. A. Kondratjev, S. Pietri, F. Naqvi, J. Gerl, S. Weick, W. Prokopowicz, H. Schaffner, C. Nociforo, H. Geissel, I. Kojouharov, M. A. G. Alvarez, I. Mukha, K. Hadyńska, P. Napiorkowski, **D. Piętak**: *GSI Scientific Report*, 236, 2008
- [6] E. Clément, A. Görgen, W. Korten, E. Bouchez, A. Chatillon, J.-P. Delaroche, M. Girod, H. Goutte, A. Hürstel, Y. Le Coz, A. Obertelli, S. Péru, Ch. Theisen, J. N. Wilson, M. Zielińska, C. Andreoiu, F. Becker, P. A. Butler, J. M. Casandjian, W. N. Catford, T. Czosnyka, G. de France, J. Gerl, R.-D. Herzberg, J. Iwanicki, D. G. Jenkins, G. D. Jones, P. J. Napiorkowski, G. Sletten, C. Timis: *Shape coexistence in neutron-deficient krypton isotopes*, Physical Review C Nuclear Physics, 75, 054313, 2007
- [7] D. Cline: *Nuclear structure inferred from heavy ion Coulomb excitation*, Bulletin of the American Physical Society, 14, 726, 1969
- [8] D. Cline, T. Czosnyka, A. B. Hayes, P. J. Napiorkowski, N. Warr, C. Y. Wu: GOSIA User Manual for Simulation and Analysis of Coulomb Excitation Experiments, http://www.pas.rochester.edu/~cline/Gosia/Gosia_Manual_20120510.pdf, instrukcja użytkownika programu GOSIA, 2012
- [9] T. Czosnyka, D. Cline, C.Y. Wu: *Coulomb excitation data analysis code GOSIA*, Rochester, NY 14627, USA, 1983, http://www.slcj.uw.edu.pl/~gosia
- [10] T. Czosnyka, D. Cline, C.Y. Wu, Bulletin of the American Physical Society, 28, 745, 1983
- [11] T. Czosnyka, D. Cline, L. Hasselgren, C. Y. Wu, R. M. Diamond, H. Kluge, C. Roulet, E. K. Hulet, R. W. Lougheed, C. Baktash: *E2 properties of the ground state band in ²⁴⁸Cm*, Nuclear Physics A, 458, 123, 1986
- [12] A. S. Davydov, G. F. Filippov: Rotational states in even atomic nuclei, Nuclear Physics, 8, 237, 1958
- [13] J. Dobaczewski, S. G. Rohoziński, J. Srebrny: *Nuclei from the barium region: Non-axial or gamma-soft?*, Zeitschrift für Physik A Atoms and Nuclei, 282, 203, 1977
- [14] J. B. A. England: *Metody doświadczalne fizyki jądrowej*, Wydawnictwo Naukowe PWN, Warszawa, 1980
- [15] L. P. Gaffney, P. A. Butler, M. Scheck, A. B. Hayes, F. Wenander, M. Albers, B. Bastin, C. Bauer, A. Blazhev, S. Bönig, N. Bree, J. Cederkäll, T. Chupp, D. Cline, T. E. Cocolios, T. Davinson, H. De Witte, J. Diriken, T. Grahn, A. Herzan, M. Huyse, D. G. Jenkins, D. T. Joss, N. Kesteloot, J. Konki, M. Kowalczyk, T. Kröll, E. Kwan, R. Lutter, K. Moschner, P. Napiorkowski, J. Pakarinen, P. Pfeiffer, P. Van Duppen, M. J. Vermeulen, M. Von Schmid, D. Voulot, N. Warr, K. Wimmer, K. Wrzosek-Lipska, C. Y. Wu, M. Zielińska: *Studies of pearshaped nuclei using accelerated radioactive beams*, Nature, 497, 199, 2013

- K. Hadyńska-Klek, P. J. Napiorkowski, A. Maj, F. Azaiez, M. Kicińska-Habior, J. J. Valiente-Dobón, [16] G. de Angelis, T. Abraham, G. Anil Kumar, B. Q. Arnés, D. Bazzacco, M. Bellato, D. Bortolato, P. Bednarczyk, G. Benzoni, L. Berti, B. Birkenbach, B. Bruyneel, S. Brambilla, F. Camera, J. Chavas, M. Ciemała, P. Cocconi, P. Coleman-Smith, A. Colombo, A. Corsi, F. C. L. Crespi, D. M. Cullen, A. Czermak, P. Désesquelles, B. Dulny, J. Eberth, E. Farnea, B. Fornal, S. Franchoo, A. Gadea, A. Giaz, A. Gottardo, X. Grave, J. Grębosz, M. Gulmini, T. Habermann, R. Isocrate, J. Iwanicki, G. Jaworski, A. Jungclaus, N. Karkour, M. Kmiecik, D. Karpiński, M. Kisieliński, N. Kondratyev, A. Korichi, M. Komorowska, M. Kowalczyk, W. Korten, M. Krzysiek, G. Lehaut, S. Leoni, A. Lopez-Martens, S. Lunardi, G. Maron, K. Mazurek, R. Menegazzo, D. Mengoni, E. Merchán, W. Męczyński, C. Michelagnoli, J. Mierzejewski, B. Million, P. Molini, S. Myalski, D. R. Napoli, R. Nicolini, M. Niikura, A. Obertelli, S. F. Özmen, M. Palacz, A. Pullia, G. Rampazzo, F. Recchia, N. Redon, P. Reiter, D. Rosso, K. Rusek, E. Sahin, M.-D. Salsac, P.-A. Söderström, J. Srebrny, I. Stefan, O. Stézowski, J. Styczeń, Ch. Theisen, N. Toniolo, C. A. Ur, V. Vandone, R. Wadsworth, B. Wasilewska, A. Wiens, K. Wrzosek-Lipska, M. Zielińska, M. Ziebliński: Toward the superdeformation in ⁴²Ca, Acta Physica Polonica B, 44, 617, 2013
- [17] K. Hadyńska-Klęk, P. Napiorkowski, A. Maj, F. Azaiez, J. J. Valiente-Dobón, G. de Angelis, G. Anil Kumar, D. Bazzacco, P. Bednarczyk, M. Bellato, G. Benzoni, L. Berti, D. Bortolato, B. Bruyneel, F. Camera, M. Ciemała, P. Cocconi, A. Colombo, A. Corsi, F. Crespi, A. Czermak, B. Dulny, E. Farnea, B. Fornal, S. Franchoo, A. Gadea, A. Giaz, A. Gottardo, X. Grave, J. Grębosz, M. Gulmini, H. Hess, R. Isocrate, G. Jaworski, M. Kicińska-Habior, M. Kmiecik, N. Kondratyev, A. Korichi, W. Korten, G. Lehaut, S. Lenzi, S. Leoni, S. Lunardi, G. Maron, R. Menegazzo, D. Mengoni, E. Merchán, W. Męczyński, C. Michelagnoli, P. Molini, D. Napoli, R. Nicolini, M. Niikura, M. Palacz, G. Rampazzo, F. Recchia, N. Redon, P. Reiter, D. Rosso, E. Sahin, J. Srebrny, I. Stefan, O. Stézowski, J. Styczeń, N. Toniolo, C. Ur, V. Vandone, B. Wadsworth, A. Wiens, K. Wrzosek-Lipska, M. Zielińska, M. Zielbliński: *Coulomb excitation of the presumably super-deformed band in*⁴²Ca. Preliminary results from the first AGATA Demonstrator experiment, Acta Physica Polonica B, 42, 817, 2011
- [18] K. Kumar: Intrinsic Quadrupole Moments and Shapes of Nuclear Ground States and Excited States, Physical Review Letters, 28, 249, 1972
- [19] T. Mayer-Kuckuk: Fizyka jądrowa, Wydawnictwo Naukowe PWN, Warszawa, 1983
- [20] J. Mierzejewski, J. Srebrny, H. Mierzejewski, J. Andrzejewski, W. Czarnacki, Ch. Droste, E. Grodner, A. Jakubowski, M. Kisieliński, M. Komorowska, A. Kordyasz, M. Kowalczyk, J. Kownacki, A. A. Pasternak, J. Perkowski, A. Stolarz, M. Zielińska, R. Anczkiewicz: *EAGLE the central European Array for Gamma Levels Evaluation at the Heavy Ion Laboratory of the University of Warsaw*, Nuclear Instruments and Methods in Physics Research A, 659, 84, 2011, w: http://slcj.uw.edu.pl/pl/eagle/, dostęp: maj 2020
- [21] F. Naqvi, P. Boutachkov, M. Górska, J. Gerl, F. Farinon, E. T. Gregor, K. Hadyńska, A. Jhingan, R. Janik, I. Kojouharov, N. A. Kondratyev, M. A. G. Alvarez, I. Mukha, P. Napiorkowski, C. Nociforo, **D. Piętak**, W. Prokopowicz, S. Pietri, A. Prochazka, H. Schaffner, P. Strmen, H. Weick, H. J. Wollersheim: *Development of Slowed Down Beams at the Fragment Separator for Fair*, Acta Physica Polonica B, 42, 725, 2011
- [22] F. Naqvi, P. Boutachkov, M. Górska, J. Gerl, F. Farinon, K. Hadyńska, R. Janik, I. Kojouharov, N. A. Kondratyev, M. A. G. Alvarez, I. Mukha, P. Napiorkowski, C. Nociforo, **D. Piętak**, W. Prokopowicz, S. Pietri, A. Prochazka, H. Schaffner, P. Strmen, H. Weick: *Extremes of the Nuclear Landscape*, Poster on Zakopane Conference on Nuclear Physics, Instytut Fizyki Jądrowej Polskiej Akademii Nauk oraz Instytut Fizyki Uniwersytetu Warszawskiego, Zakopane, 2010
- [23] F. Naqvi, P. Boutachkov, M. Górska, J. Gerl, F. Farinon, K. Hadyńska, R. Janik, I. Kojouharov, N. A. Kondratyev, M. A. G. Alvarez, I. Mukha, P. Napiorkowski, C. Nociforo, D. Piętak, W. Prokopowicz, S. Pietri, A. Prochazka, H. Schaffner, P. Strmen, H. Weick: *Development of Slowed Down Beams at GSI/FAIR*, Poster on Zakopane Conference on Nuclear Physics, Instytut Fizyki Jądrowej Polskiej Akademii Nauk oraz Instytut Fizyki Uniwersytetu Warszawskiego, Zakopane, 2010

- [24] J. Srebrny, D. Cline: *Model independent determination of quadrupole deformation parameters* from Coulomb excitation measurements, International Journal of Modern Physics E, 20, 422, 2011
- [25] J. Srebrny, T. Czosnyka, W. Karczmarczyk, P. Napiorkowski, Ch. Droste, H.-J. Wollersheim, H. Emling, H. Grein, R. Kulessa, D. Cline, C. Fahlander: *E1, E2, E3 and M1 information from heavy ion Coulomb excitation*, Nuclear Physics A, 557, 663, 1993
- [26] J. Srebrny, T. Czosnyka, Ch. Droste, S. G. Rohoziński, L. Próchniak, K. Zając, K. Pomorski, D. Cline, C. Y. Wu, A. Bäcklin, L. Hasselgren, R. M. Diamond, D. Habs, H. J. Körner, F. S. Stephens, C. Baktah, R. P. Kostecki: *Experimental and theoretical investigations of quadrupole collective degrees of freedom in ¹⁰⁴Ru*, Nuclear Physics A, 25, 766, 2006
- [27] A. Strzałkowski: *Wstęp do fizyki jądra atomowego*, Wydawnictwo Naukowe PWN, Warszawa, 1979
- [28] Z. Wilhelmi: Fizyka reakcji jądrowych, Wydawnictwo Naukowe PWN, Warszawa, 1976
- [29] A. Winther, J. de Boer: *A Computer Program for Multiple Coulomb Excitation*, Technical Report, California Institute of Technology, 1965
- [30] A. Winther, J. de Boer: *Coulomb excitation*, ed. K. Alder and A. Winther Academic, New York, 303, 1966
- [31] K. Wrzosek-Lipska, K. Hadyńska-Klęk, J. Iwanicki, P. J. Napiorkowski, L. Pieńkowski, D. A. Piętak, J. Srebrny, M. Zielińska: *Coulomb excitation of ¹⁰⁰Mo*, wystąpienie na Zakopane Conference on Nuclear Physics, Instytut Fizyki Jądrowej Polskiej Akademii Nauk oraz Instytut Fizyki Uniwersytetu Warszawskiego, Zakopane 2010
- [32] K. Wrzosek-Lipska, L. Próchniak, M. Zielińska, K. Hadyńska-Klęk, J. Iwanicki, M. Kisieliński, M. Kowalczyk, P. J. Napiorkowski, D. Piętak, J. Srebrny, T. Czosnyka: Electromagnetic properties of ¹⁰⁰Mo experimental results and theoretical description of quadrupole degrees of freedom, Physical Review C, 86, 064305, 2012
- [33] K. Wrzosek-Lipska, M. Zielińska, K. Hadyńska-Klęk, Y. Hatsukawa, J. Iwanicki, J. Katakura, M. Kisieliński, M. Koizumi, M. Kowalczyk, H. Kusakari, M. Matsuda, T. Morikawa, P. J. Napiorkowski, A. Osa, M. Osihima, D. Piętak, L. Próchniak, T. Shizuma, J. Srebrny, M. Sugawara, Y. Toh: *Shape evolution in heaviest stable even-even molybdenum isotopes studied via Coulomb excitation*, International Journal of Modern Physics E, 20, 2, 2011
- [34] K. Wrzosek-Lipska, M. Zielińska, K. Hadyńska-Klęk, J. Iwanicki, M. Kisieliński, M. Kowalczyk, P. J. Napiorkowski, **D. Piętak**, J. Srebrny: *Quadrupole moment of the 2*⁺ state in ¹⁰⁰Mo, Acta Physica Polonica B, 42, 803, 2011
- [35] K. Wrzosek, M. Zielińska, T. Czosnyka, J. Choiński, K. Hadyńska, J. Iwanicki, M. Kisieliński, M. Kowalczyk, J. Kownacki, P. J. Napiorkowski, **D. Piętak**, J. Srebrny, K. Zając: New γparticle detection set-up for Coulomb excitation experiments – towards determination of triaxiality of ¹⁰⁰Mo, Acta Physica Polonica B, 39, 513, 2008
- [36] M. Zielińska, L. P. Gaffney, K. Wrzosek-Lipska, E. Clément, T. Grahn, N. Kesteloot, P. Napiorkowski, J. Pakarinen, P. Van Duppen, N. Warr: *Analysis methods of safe Coulomb-excitation experiments with radioactive ion beams using the GOSIA code*, The European Physical Journal A, 52, 99, 2016
- [37] M. Zielińska, A. Görgen, E. Clément, J.-P. Delaroche, M. Girod, W. Korten, A. Bürger, W. Catford, C. Dossat, J. Iwanicki, J. Libert, J. Ljungvall, P. J. Napiorkowski, A. Obertelli, D. Piętak, R. Rodríguez-Guzmán, G. Sletten, J. Srebrny, Ch. Theisen, K. Wrzosek: On the Shape of ⁴⁴Ar: Onset of deformation in neutron-rich nuclei near ⁴⁸Ca, Physical Review C, 80, 014317, 2009
- [38] M. Zielińska, A. Görgen, A. Bürger, W. Catford, E. Clément, C. Dossat, J. Iwanicki, W. Korten, J. Ljungvall, P. J. Napiorkowski, **D. Piętak**, G. Sletten, J. Srebrny, Ch. Theisen, K. Wrzosek: *Coulomb excitation of Neutron-Rich* ⁴⁴Ar at SPIRAL, Acta Physica Polonica B, 39, 519-524, 2008

- [39] M. Ziębliński, M. Jastrząb, Neha Dokania, V. Nanal, S. Brambilla, P. Bednarczyk, M. Ciemała, E. Dutkiewicz, M. Kmiecik, M. Krzysiek, J. Lekki, A. Maj, Z. Szklarz, B. Wasilewska, M. Dudeło, K. Hadyńska-Klęk, P. Napiorkowski, B. Genolini, Ch. Schmitt, W. Catford, M. Nakhostin, N. Yavuzkanat, O. Dorvaux, R. G. Pillay, M. S. Pose, S. Mishra, S. Mathimalar, V. Singh, N. Katyan, D. R. Chakrabarty, V. M. Datar, Suresh Kumar, G. Mishra, S. Mukhopadhyay, D. Pandit, S. Erturk: *Testing of the PARIS LaBr₃–NaI Phoswich detector with high energy gammarays*, Acta Physica Polonica B, 44, 651, 2013
- [40] CMS collaboration: Combined results of searches for the standard model Higgs boson in pp collisions at $\sqrt{s} = 7$ TeV, Physics Letters B, 710, 26, 2012
- [41] ATLAS collaboration, CMS collaboration: Combined measurement of the Higgs boson mass in pp collisions at $\sqrt{s} = 7$ and 8 TeV with the ATLAS and CMS experiments, Physical Review Letters, 114, 191803, 2015
- [42] J. Choiński: *Struktura elektromagnetyczna i dynamika wiązek Warszawskiego Cyklotronu U-*200P, Rozprawa doktorska, Uniwersytet Warszawski, 2003
- [43] K. Hadyńska-Klęk: Badanie struktury kolektywnej w izotopach wapnia metodą wzbudzeń kulombowskich, Rozprawa doktorska, Środowiskowe Laboratorium Ciężkich Jonów oraz Instytut Fizyki Doświadczalnej, Uniwersytet Warszawski, 2013
- [44] J. Iwanicki: Badanie struktury elektromagnetycznej jąder atomowych metodą analizy wzbudzenia kulombowskiego, Rozprawa doktorska, Uniwersytet Warszawski, 2001
- [45] P. J. Napiorkowski: *Eksperymentalne wyznaczenie elementów macierzowych przejść elektromagnetycznych w jądrze*¹²⁸Xe, Praca dyplomowa, Uniwersytet Warszawski, 1992
- [46] P. J. Napiorkowski: *Elektromagnetyczne własności stabilnych izotopów hafnu*, Rozprawa doktorska, Środowiskowe Laboratorium Ciężkich Jonów, Uniwersytet Warszawski, 2004
- [47] K. Wrzosek: Badanie struktury elektromagnetycznej jądra ¹⁰⁰Mo metodą wzbudzeń kulombowskich. Praca dyplomowa, Uniwersytet Warszawski, 2005
- [48] K. Wrzosek-Lipska: Badanie struktury elektromagnetycznej niskospinowych stanów wzbudzonych jądra ¹⁰⁰Mo metodą wzbudzeń kulombowskich, Rozprawa doktorska, Środowiskowe Laboratorium Ciężkich Jonów oraz Instytut Fizyki Doświadczalnej, Uniwersytet Warszawski, 2010
- [49] M. Zielińska: *Struktura elektromagnetyczna jąder atomowych izotopów molibdenu badana metodą wzbudzenia kulombowskiego*, Rozprawa doktorska, Uniwersytet Warszawski, 2005
- [50] P. J. Napiorkowski: *Informacja własna*, Środowiskowe Laboratorium Ciężkich Jonów, Uniwersytet Warszawski, 2020
- [51] Strona internetowa National Nuclear Data Center: https://www.nndc.bnl.gov, dostęp: maj 2020
- [52] Strona internetowa GSI Darmstadt: https://www.gsi.de, dostęp: maj 2020
- [53] P. J. Napiorkowski, K. Hadyńska-Klęk, J. Iwanicki, **D. A. Piętak**, J. Srebrny, K. Wrzosek-Lipska, M. Zielińska (Warszawska Grupa Wzbudzeń Kulombowskich): *Wzbudzenia kulombowskie w Warszawskim Cyklotronie*, http://www.slcj.uw.edu.pl/pl/cudac, dostęp: maj 2020
- [54] M. Zielińska, P. J. Napiorkowski: *GOSIA Workshop 8-10 April 2008*, Środowiskowe Laboratorium Ciężkich Jonów, Uniwersytet Warszawski, 2008, w: http://www.old.slcj.uw.edu.pl/en/154.html, dostęp: maj 2020
- [55] M. Zielińska, P. J. Napiorkowski: 2nd GOSIA Workshop 9-11 April 2013, Środowiskowe Laboratorium Ciężkich Jonów, Uniwersytet Warszawski, 2013, w: http://www.old.slcj.uw.edu.pl/en/118.html, dostęp: maj 2020
- [56] M. Zielińska, P. J. Napiorkowski: 3rd GOSIA Workshop at the Heavy Ion Laboratory 9-11 April 2018, Środowiskowe Laboratorium Ciężkich Jonów, Uniwersytet Warszawski, 2018, w: http://www.slcj. uw.edu.pl/en/gosia_workshop, dostęp: maj 2020

- [57] J. Kownacki: *Spektroskopia gamma przy użyciu układu OSIRIS*-II, w: http://slcj.uw.edu.pl/pl/osirisii/, dostęp: maj 2020
- [58] K. Hadyńska-Klęk, D. Balabanski, M. Ciemała, E. Clément, A. Dijon, A. K. Gourishetty, M. Kicińska-Habior, M. Kisieliński, S. Kisyov, M. Kmiecik, M. Kowalczyk, S. Lalkovski, A. Maj, W. Męczyński, M. Moszyński, P. J. Napiorkowski, D. A. Piętak, J. Srebrny, Ł. Świderski, D. Wolski, K. Wrzosek-Lipska, M. Zielińska, M. Ziębliński: *In-beam investigations of the properties of LaBr₃ detectors*, HIL Annual Report, 49, 2009
- [59] K. Hadyńska-Klęk, K. Wrzosek-Lipska, J. Iwanicki, M. Kisieliński, A. Kordyasz, M. Kowalczyk,
 J. Kownacki, S. Lalkovski, P. J. Napiorkowski, D. A. Piętak, J. Srebrny, M. Zielińska:
 Coulomb excitation of ¹⁰⁰Mo determination of the E3 strength, HIL Annual Report, 39, 2009
- [60] W. Piątek, A. Rubin, J. Górecki, **D. A. Piętak**, P. J. Napiorkowski: *Parallel determination of the objective function value based on a distributed application structure*, HIL Annual Report, 55, 2010
- [61] K. Wrzosek, K. Hadyńska, J. Iwanicki, M. Kisieliński, A. Kordyasz, M. Kowalczyk, J. Kownacki, S. Lalkovski, P. Napiorkowski, L. Pieńkowski, **D. Piętak**, J. Srebrny, A. Trzcińska, M. Zielińska: *Coulomb excitation of ¹⁰⁰Mo – towards determination of triaxiality*, HIL Annual Report, 28, 2007
- [62] K. Wrzosek-Lipska, K. Hadyńska-Klęk, J. Iwanicki, M. Kisieliński, A. Kordyasz, M. Kowalczyk, J. Kownacki, S. Lalkovski, P. J. Napiorkowski, D. A. Piętak, L. Pieńkowski, J. Srebrny, M. Zielińska: *Quadrupole moment of the state 2*₁⁺ *in ¹⁰⁰Mo*, HIL Annual Report, 41, 2009
- [63] K. Wrzosek-Lipska, L. Próchniak, M. Zielińska, K. Hadyńska-Klęk, J. Iwanicki, M. Kisieliński, M. Kowalczyk, P. J. Napiorkowski, **D. Piętak**, J. Srebrny: *Shape evolution in ¹⁰⁰Mo studied via Coulomb excitation – comparison with the GBH model predictions*, HIL Annual Report, 39, 2010
- [64] M. Zielińska, A. Görgen, A. Bürger, W. Catford, E. Clément, C. Dossat, J. Iwanicki, W. Korten, J. Ljungvall, P. J. Napiorkowski, **D. Piętak**, G. Sletten, J. Srebrny, Ch. Theisen, K. Wrzosek-Lipska: *Coulomb excitation of ¹⁰⁹Ag*, HIL Annual Report, 57, 2009
- [65] **D. A. Piętak**, P. J. Napiorkowski, A. Pająk, J. Srebrny: *Implementation of a genetic algorithm to the Coulex data analysis*, HIL Annual Report, 29, 2007
- [66] **D. A. Piętak**: *Implementation of a genetic algorithm to the COULEX data analysis*, wystąpienie na GOSIA Workshop, Warszawa, 2008
- [67] **D. A. Piętak**, P. J. Napiorkowski, K. Hadyńska, K. Wrzosek, J. Srebrny, M. Zielińska: *Graphical User Interface for Coulex analysis tool*, Poster on EURISOL-EURONS Town Meeting, Rhodes, 2008

Rozdział 3 – Algorytm genetyczny

- [68] J. Arabas: Wykłady z algorytmów ewolucyjnych, WNT, 2004
- [69] A. P. Engelbrecht: Computational Intelligence, John Wiley & Sons Ltd., 2002
- [70] D. E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, MA, 1989
- [71] D. E. Goldberg: Algorytmy genetyczne i ich zastosowania, WNT, 1995
- [72] J. Grębosz: Symfonia C++. Programowanie w języku C++ orientowane obiektowo, Oficyna Kallimach, 1999
- [73] T. D. Gwiazda: Algorytmy genetyczne kompendium. Tom I, Wydawnictwo Naukowe PWN, 2007
- [74] S. B. Lippman, J. Lajoie: *Podstawy języka C++*, WNT, 2001
- [75] S. J. Metsker: C# Wzorce projektowe, Helion, 2005
- [76] Z. Michalewicz: Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1999
- [77] Z. Michalewicz: Algorytmy genetyczne + struktury danych = programy ewolucyjne, WNT, 2003
- [78] Z. Michalewicz, D. B. Fogel: Jak to rozwiązać, czyli nowoczesna heurystyka, WNT, 2006

- [79] P. J. Napiorkowski, **D. A. Piętak**: *Obliczanie kształtów jąder atomowych z wykorzystaniem algorytmów heurystycznych i przetwarzania rozproszonego*, seminarium, Uniwersytet Kardynała Stefana Wyszyńskiego, Warszawa, 2011
- [80] B. Sklar: Digital Communications: Fundamentals and Applications, Prentice-Hall, 2001
- [81] A. Stachurski, A. Wierzbicki: *Podstawy optymalizacji*, Oficyna wydawnicza Politechniki Warszawskiej, 2001
- [82] B. Stroustrup: *Język C++*, WNT, 1995
- [83] A. Troelsen: Język C# i platforma .Net, Wydanie drugie, Wydawnictwo Naukowe PWN, Warszawa, 2006
- [84] Planck Collaboration: *Plank 2015 results. XIII. Cosmological parameters*, Astronomy & Astrophysics. 594: A13. arXiv:1502.01589, 2016, w: https://arxiv.org/pdf/1502.01589.pdf, dostęp: maj 2020
- [85] K. Piętak (Ramsza): *Wirusowe algorytmy genetyczne inspirowane obliczeniami molekularnymi*, Praca dyplomowa, Politechnika Warszawska, 1998
- [86] **D. A. Piętak**, P. J. Napiorkowski, A. Pająk, J. Srebrny: *An implementation of a genetic algorithm to the Coulex data analysis*, Poster on EURISOL-EURONS Town Meeting, Helsinki, 2007
- [87] **D. A. Piętak**: *Implementacja algorytmu genetycznego do analizy danych z pomiarów wzbudzeń kulombowskich*, Praca dyplomowa, Politechnika Warszawska, 2008
- [88] D. A. Piętak, P. J. Napiorkowski, Z. Walczak, J. Wojciechowski: Application of Genetic Algorithm with Real Representation to COULEX Data Analysis, Proceedings of the Conference on Evolutionary Computation and Global Optimization, Oficyna Wydawnicza Politechniki Warszawskiej, 2009
- [89] D. A. Piętak, P. J. Napiorkowski, K. Hadyńska-Klęk, K. Wrzosek-Lipska, J. Srebrny, M. Zielińska: An application of genetic algorithm to the COULEX data analysis, Poster on Zakopane Conference on Nuclear Physics, Instytut Fizyki Jądrowej Polskiej Akademii Nauk oraz Instytut Fizyki Uniwersytetu Warszawskiego, Zakopane, 2010

Rozdział 4 – Analiza próbkowania

- [90] M. Ankerst, M. Breunig, H.-P. Kriegel, J. Sander: *OPTICS: Ordering points to identify the clustering structure*, Proceedings of 1999 ACM-SIGMOD International Conference on Management of Data (SIGMOD'99), pages 49-60, Philadelphia, 1999
- [91] A. Björck, G. Dahlquist: *Metody numeryczne*, Wydawnictwo Naukowe PWN, 1987 2006
- [92] M. Ester, H. Kriegel, J. Sander, X. Xu: *A density-based algorithm for discovering clusters in large spatial databases with noise*, Proceedings of 1998 International Conference on Knowledge Discovery and Data Mining (KDD'96), pages 226-231, Portland Oregon, 1996
- [93] J. Han, M. Kamber: Data mining: concepts and techniques, Morgan Kaufmann Publishers, 2000
- [94] A. K. Jain, M. N. Murty, P. J. Flyn: *Data clustering: A review*, ACM Computing Surveys, 31(3): 264-323, 1999
- [95] The MathWorks: Matlab User's Manual, MathWorks Inc., Natick, Mass., USA, 1992
- [96] C. B. Moler: Numerical computing with MATLAB, SIAM, Philadelphia, 2004
- [97] A. Plucińska, E. Pluciński: *Probabilistyka. Rachunek prawdopodobieństwa, statystyka matematyczna, procesy stochastyczne*, WNT, 2006
- [98] H.-P. Schwefel: Numerical optimization of computer models, Chichester: Wiley & Sons, 1981
- [99] R. Weber, H.-J. Schek, S. Blott: *A quantitative analysis and performance study for similaritysearch methods in high-dimensional spaces*, Proceedings of the Conference on Very Large DataBases (VLDB'98), pages 194-205, New York City, 1998

- [100] R. Weber, S. Blott: An approximation based data structure for similarity search, Technical Report 24, ESPRIT project HERMES (no. 9141), 1997
- [101] T. Williams, C. Kelley: *Gnuplot 5.3. An Interactive Plotting Program*, 2018, w: http://www.gnuplot.info/docs 5.5/gnuplot.pdf, dostęp: maj 2020
- [102] S. Zhou, Y. Zhao, J. Guan, J. Huang: *NBC: A Neighborhood-Based Clustering Algorithm*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2005
- [103] **D. A. Piętak**: *Statistical distribution of the genetic algorithm sampling with Schwefel's F7 objective function*, Proceedings of the IEEE International Conference on Signals and Electronic Systems (ICSES), Gliwice, 2010

Rozdział 5 – Estymacja Niepewności

- [104] J. E. Amaro, R. Navarro Pérez, E. Ruiz Arriola: *Error analysis of nuclear matrix elements*, Few-Body Systems, 2013
- [105] P. R. Bevington, D. K. Robinson: *Data Reduction and Error Analysis for the Physical Sciences*, McGraw-Hill, 2003
- [106] G. E. P. Box, N. R. Draper: *Empirical Model Building and Response Surfaces*, John Wiley & Sons, New York, 1987
- [107] G. E. P. Box: Science and statistics, Journal of American Statistical Association, 71:791, 1976
- [108] S. Brandt: Data Analysis. Statistical and Computational Methods for Scientists and Engineers, Fourth Edition, Springer, New York, 2014
- [109] J. Dobaczewski, W. Nazarewicz, P.-G. Reinhard: *Error Estimates of Theoretical Models: a Guide*, Journal of Physics G: Nuclear and Particle Physics, 41, 074001, 2014
- [110] W. T. Eadie, D. Drijard, F. E. James, M. Roos, B. Sadoulet: *Metody statystyczne w fizyce doświadczalnej*, Wydawnictwo Naukowe PWN, Warszawa, 1989
- [111] D. Higdon: *Statistical approaches for combining model runs with experimental data*, Presentation at ISNET Meeting, Glasgow, 2013
- [112] International Organization for Standardization (ISO): Guide to the expression of Uncertainty in Measurement (GUM) – Supplement 1: Numerical methods for the propagation of distributions, Report, 2004
- [113] J. M. Jaworski, R. Z. Morawski, J. S. Olędzki: Wstęp do metrologii i techniki eksperymentu, WNT, 1992
- [114] Joint Committee for Guides in Metrology (JCGM): JCGM 100:2008 Evaluation of measurement data Guide to the expression of Uncertainty in Measurement, Report, 2008
- [115] Joint Committee for Guides in Metrology (JCGM): JCGM 101:2008 Evaluation of measurement data Supplement 1 to the "Guide to the expression of Uncertainty in Measurement" Propagation of distributions using a Monte Carlo method, Report, 2008
- [116] Joint Committee for Guides in Metrology (JCGM): JCGM 102:2011 Evaluation of measurement data Supplement 2 to the "Guide to the expression of Uncertainty in Measurement" Extension to any number of output quantities, Report, 2011
- [117] Joint Committee for Guides in Metrology (JCGM): JCGM 103 Evaluation of measurement data – Supplement 3 to the "Guide to the expression of Uncertainty in Measurement" – Developing and using measurement models, Report, 2018
- [118] Joint Committee for Guides in Metrology (JCGM): JCGM 104:2009 Evaluation of measurement data An introduction to the "Guide to the expression of Uncertainty in Measurement" and related documents, Report, 2009

- [119] Joint Committee for Guides in Metrology (JCGM): JCGM 106:2012 Evaluation of measurement data – The role of measurement uncertainty in conformity assessment, Report, 2012
- [120] Joint Committee for Guides in Metrology (JCGM): JCGM 200:2008 International vocabulary of metrology basic and general concepts and associated terms, Report, 2008
- [121] A. Kubiaczyk: Określanie Niepewności Pomiarów (poradnik do Laboratorium Fizyki), Wydział Fizyki, Politechnika Warszawska, 2019
- [122] S. L. Meyer: Data Analysis for Scientists and Engineers, John Wiley, 1975
- [123] R. Z. Morawski, A. Miękina: Monte-Carlo evaluation of measurement uncertainty using a new generator of pseudo-random numbers, PAK vol. 59, nr 5, 2013
- [124] E. M. Pugh, G. H. Winslow: The Analysis of Physical Measurements, Addison-Wesley, 1966
- [125] A. Saltelli, S. Funtowicz: When all models are wrong, Issues in Science and Technology, Fall 2013:79, 2013
- [126] A. Tarantola: *Inverse problem theory and methods for model parameter estimation*, SIAM, Philadelphia, 2005
- [127] J. R. Taylor: Wstep do Analizy Błędu Pomiarowego, Wydawnictwo Naukowe PWN, Warszawa, 1995
- [128] B. A. Wichmann, I. D. Hill: *Generating good pseudo-random numbers*, Computational Statistics and Data Analysis 51, pages 1614-1622, 2006
- [129] D. A. Piętak, J. Wojciechowski, P. J. Napiorkowski: A Front-Line algorithm for error estimation in datasets with nonuniform sampling distribution, Proceedings of the 20th European Conference on Circuit Theory and Design (ECCTD), 6043319, pp. 210-213, Linköping, Szwecja, 2011

Rozdział 6 – Zastosowanie metody

- [130] S. Dutt, M. Saxena, R. Kumar, A. Jhingan, A. Agarwal, A. Banerjee, R. K. Bhowmik, Ch. Joshi, J. Kaur, A. Kumar, M. Matejska-Minda, V. Mishra, I. Rizvi, A. Stolarz, H. J. Wollersheim, P. J. Napiorkowski: *Re-measurement of Reduced Transition Probabilities in* ¹³²Ba, Acta Physica Polonica Series B, 49, 3, 535, 2018
- [131] S. Dutt, M. Saxena, P. J. Napiorkowski, R. Kumar, T. Abrahahm, J. M. Allmond, A. Gwalik, K. Hadyńska-Klęk, M. Hlebowicz, J. Iwanicki, M. Kisieliński, M. Komorowska, M. Kowalczyk, T. Marchlewski, M. Matejska-Minda, F. Oleszczuk, M. Palacz, W. Piątek, L. Próchniak, I. A. Rizvi, J. Samorajczyk, J. Srebrny, A. Tucholski, W. Wróblewski, K. Wrzosek-Lipska: *Multi-step Coulomb excitation of ¹³²Ba at HIL, Warsaw*, Indian Journal of Pure & Applied Physics, 57, 599, 2019

Podsumowanie

- [132] S. Osowski: Sieci neuronowe do przetwarzania informacji, Oficyna Wydawnicza Politechniki Warszawskiej, 2000
- [133] P. Bilski: *Hierarchical diagnostics of analog systems based on the ambiguity groups detection*, Measurement, 119(1): 1-10, 2018
- [134] P. Kossakowski, P. Bilski: Analysis of the Self-Organizing Map-Based Investment Strategy, International Journal of Computing, 16(1), 10-17, 2017
- [135] A. Baldominos, Y. Saez, P. Isasi: Hybridizing Evolutionary Computation and Deep Neural Networks: An Approach to Handwriting Recognition Using Committees and Transfer Learning, Complexity, article ID 2952304, 2019
- [136] Y. LeCun, Y. Bengio, G. Hinton: Deep Learning, Nature, vol. 521, 2015
- [137] J. Blanchette, M. Summerfield: C++ GUI Programming with Qt 4, Trolltech Press,

Dodatek A – Zaimplementowane programy

A.1. Budowa biblioteki GeneticAlgorithm.dll

Niniejszy punkt stanowi rozwinięcie punktu 3.3. Zawiera szczegółowe omówienie wszystkich klas biblioteki GeneticAlgorithm.dll. Stanowią one jedną przestrzeń nazw namespace GA. Można wśród nich wyróżnić następujące grupy:

- referencyjna klasa fasadowa biblioteki GeneticAlgorithm;
- klasa reprezentująca całą populację Population;
- klasy związane z reprezentacją instancji problemu ICreature, ICreatureGenerator oraz klasy potomne;
- klasy związane z funkcją celu IObjectiveFunctionDeterminator,
 IObjectiveFunction, IFitnessTransform oraz klasy potomne;
- klasy związane z operatorami genetycznymi IGeneticOperator oraz klasy potomne;
- klasy związane z fenotypem osobnika (dziedziną problemu) IPhenotype oraz klasy potomne i pomocnicze.

Jak widać z powyższego zestawienia w bibliotece GeneticAlgorithm.dll wykorzystywany jest obiektowy paradygmat dziedziczenia i polimorfizmu. Ponadto biblioteka GeneticAlgorithm.dll korzysta z implementacji generatorów liczb pseudolosowych zebranych w klasie REP::Generator, która została omówiona w punkcie A.3.5.

A.1.1. Klasa GeneticAlgorithm

Klasa GeneticAlgorithm stanowi fasadę całej biblioteki DLL. Jest ona typu referencyjnego, co pozwala utworzyć jej instancję w dowolnym języku programowania w systemie Windows. Dzięki temu algorytm genetyczny jest oddzielony od graficznego interfejsu użytkownika nie tylko zgodnie z zasadą warstwa kodu – warstwa wizualizacji, ale dodatkowo algorytm jest zaimplementowany w natywnym języku C++, a interfejs użytkownika w języku C#. Klasa GeneticAlgorithm zarządza i raportuje całym procesem optymalizacji algorytmem genetycznym. Diagram klasy przedstawia Rys. A.1.

Aby utworzyć obiekt klasy GeneticAlgorithm należy wskazać lokalizację programu GOSIA i "inputu do GOSI" oraz podać rozmiar populacji początkowej. Populacja początkowa będzie wynikiem losowego próbkowania przestrzeni rozwiązań dopuszczalnych z rozkładem

równomiernym. Jeśli populacja początkowa ma być umieszczona wokół zadanego w "inpucie do GOSI" punktu startowego z rozkładem Gaussa, to należy dodatkowo podać wartość parametru sigma. Na jego podstawie, zgodnie ze wzorem (A.1), zostanie utworzona tablica sigmaTable o rozmiarze równym liczbie parametrów (niezależnych elementów macierzowych). W ten sposób odchylenia standardowe próbkowania przestrzeni będą znormalizowane do zakresów zmienności poszczególnych parametrów.

 $sigmaTable_i = sigma \cdot range_i$

(A.1)

Podstawową metodą klasy GeneticAlgorithm jest funkcja runOneGeneration(). Umożliwia ona przeprowadzenie jednego kroku optymalizacyjnego przy określonych parametrach. Do zmiany parametrów algorytmu genetycznego służą metody:

GeneticAlgorithm Class	8
🖃 Fields	
population : Population* repository : PointSet*	
Methods	
 reactions creationAndInitialization(String^ input, String^ gosia) : void GeneticAlgorithm(String^ gosia, String^ input, int popSize) GeneticAlgorithm(String^ gosia, String^ input, int popSize, double sigma) getAverageChiSq() : double getBestChiSq() : double getEverBestChiSq() : double getEverBestME(int i) : double getIverBestME(int i) : double getIverBestME(int i) : double getTimeOfChiSqDeterminations() : int getTimeOfChiSqDeterminations() : double getTimeOfCrossingOver() : double getTimeOfGeneration() : double getTimeOfGeneration() : double getTimeOfGeneration() : double getTimeOfSelection() : double resetBestFile() : void resetBestFile() : void 	
vunOneGeneration() : void	
saveBestInRepository(String^ solutionFileName) : void	
 saveRepository(String^ repositoryFileName) : void setBestFile(String^ bestFileName) : void setLogFile(String^ logFileName) : void 	
useMutator_Constant(double sigma, double baseProbability) : void	
 useMutator_Hyperbolic(double sigma, double baseProbability) : void useMutator_Logarithmic(double sigma, double baseProbability) : void useParentByRoulette(int maxChildrenNumber) : void useSelectionByRoulette(int perCentToBeEliminated) : void useSelectionByTournament(int tournamentSize) : void 	
useSelectionByTruncation(int perCentToBeEliminated) : void	

Rys. A.1 Diagram klasy GeneticAlgorithm

- useSelectionByRoulette(), useSelectionByTournament(),
 useSelectionByTruncation() dla operatorów selekcji;
- useParentByRoulette() dla operatora krzyżowania;
- useMutator_Constant(), useMutator_Hyperbolic(),
 useMutator_Logarithmic() dla operatorów mutacji.

Ich działanie zostało opisane w odpowiednich częściach punktu 3.2.

Jak wskazano w punkcie 3.2.6 pozostałe ustawienia algorytmu genetycznego (reprezentacja instancji problemu, funkcja celu, transformacja wartości funkcji celu) wymagają zmiany i rekompilacji kodu źródłowego biblioteki. Są one określone w konstruktorze klasy oraz w prywatnej metodzie creationAndInitialization().

Do włączania/wyłączania zapisu do pliku całej populacji po każdym operatorze genetycznym służą metody: setLogFile() oraz resetLogFile(). Do włączania/wyłączania zapisu do pliku najlepszego osobnika w każdym pokoleniu służą metody: setBestFile() oraz resetBestFile(). Zapis do pliku całego zebranego próbkowania przestrzeni rozwiązań dopuszczalnych oraz najlepszego znalezionego rozwiązania jest możliwy poprzez metody saveRepository() oraz saveBestInRepository(). Metody udostępniające informacje o przebiegu procesu optymalizacyjnego zebrane zostały w Tabeli A.1.

Nazwa metody	Informacje zwracane przez metodę
getBestChiSq()	najniższa wartość χ^2 w pokoleniu
getBestME(i)	wartość i-tego elementu macierzowego dla wektora o najniższej
	wartości χ^2 w pokoleniu
getAverageChiSq()	średnia wartość χ^2 w pokoleniu
getEverBestChiSq()	najniższa wartość χ^2 w całym przebiegu algorytmu genetycznego
getEverBestME(i)	wartość i-tego elementu macierzowego dla wektora o najniższej
	wartości χ^2 w całym przebiegu algorytmu genetycznego
getTimeOfGeneration()	czas obliczania pokolenia
getTimeOfChiSqDeterminations()	czas i liczba wywołań funkcji celu
<pre>getNumberOfChiSqDeterminations()</pre>	
getTimeOfSelection()	czas poszczególnych operatorów genetycznych
getTimeOfCrossingOver()	
getTimeOfMutation()	
getNumberOfGeneration()	numer pokolenia

Tabela A.1 Metody raportujące przebieg algorytmu genetycznego w klasie GeneticAlgorithm.

A.1.2. Klasa Population

Klasa Population reprezentuje całą populację wraz z przeprowadzaną symulacją algorytmem genetycznym. Przechowuje ona zbiór osobników składających się na aktualne pokolenie – jako wskaźnik na obiekt std::vector. Elementami kontenera są wskaźniki na klasę bazową osobników ICreature. Diagram klasy Polulation przedstawia Rys. A.2.



Rys. A.2 Diagram klasy Population

Aby utworzyć obiekt klasy Population należy podać wskaźniki na następujące obiekty, które będą wykorzystywane w tej instancji klasy:

- generator osobników ICreaturesGenerator;
- funkcję celu IObjectiveFunctionDeterminator;
- transformację wartości funkcji celu IFitnessTransform;

oraz rozmiar populacji.

Klasa Population za pomocą prywatnej metody generateCreatures() uzupełnia wektor osobników do określonego rozmiaru populacji. Publiczna metoda setPopSize() umożliwia zmianę rozmiaru populacji. Gdy rozmiar jest zwiększany – tworzone są dodatkowe osobniki, a gdy zmniejszany – osobniki są losowo usuwane z wektora. Ze względu na założenie projektowe, że rozmiar populacji jest stały w obrębie jednego procesu optymalizacyjnego, funkcjonalność zmiany rozmiaru populacji po utworzeniu instancji klasy Population nie jest udostępniana przez klasę fasadową biblioteki GeneticAlgorithm. Podstawową metodą klasy Population jest funkcja runOneGeneration(). Przyjmuje ona jako argumenty wskazania na klasy bazowe operatorów: selekcji (ISelection), krzyżowania (ICrossingOver) i mutacji (IMutation) oraz przeprowadza jeden krok minimalizacyjny przy ich użyciu. Opcjonalnie metoda jako czwarty argument przyjmuje wskazanie na strumień wyjściowy, do którego po każdym operatorze genetycznym zapisywana jest cała populacja.

Klasa Population udostępnia również metody raportujące aktualny stan populacji:

- getGenerationNumber() zwraca numer aktualnego pokolenia;
- getBest()-zwraca wskazanie na najlepszego osobnika w pokoleniu;
- getAverageFitness() zwraca średnią wartość funkcji oceny w pokoleniu;
- print () wypisuje do wskazanego strumienia aktualną zawartość wektora osobników.

A.1.3. Klasy ICreature oraz ICreaturesGenerator

Klasy bazowe ICreature i ICreaturesGenerator oraz ich klasy potomne odpowiadają za tworzenie i przechowywanie informacji o każdym osobniku zgodnie z przyjętą reprezentacją instancji problemu. Diagram klasy bazowej ICreature oraz jej klas potomnych przedstawia Rys. A.3. Diagram klasy bazowej ICreaturesGenerator oraz jej klas potomnych przedstawia Rys. A.4.

Klasa bazowa ICreature zawiera między innymi metody wirtualne, dlatego niemożliwe jest utworzenie jej obiektów. Stanowi natomiast rodzaj interfejsu²⁵ dla klas implementujących poszczególne reprezentacje instancji problemu. Wymaga od klas potomnych przekazania, jako argument konstruktora, wskaźnika na klasę IPhenotype, która przechowuje m. in. informacje o zakresach zmienności poszczególnych parametrów. Ponadto klasy potomne implementują różne operatory krzyżowania, dlatego w omawianym projekcie są trzy:

- BitsetCreature implementacja reprezentacji binarnej;
- RealCreatureHX implementacja reprezentacji rzeczywistej z krzyżowaniem heurystycznym-2;
- RealCreatureDC implementacja reprezentacji rzeczywistej z krzyżowaniem ziarnistym.

²⁵ Język C++ nie definiuje pojęcia interfejsu. Aby uniknąć dwuznaczności z definicją stosowaną w innych językach programowania (np. Java), w dalszej części niniejszej pracy pojęcie to będzie używane w znaczeniu powszechnie przyjętym.



Rys. A.3 Diagram klasy bazowej ICreature i jej klas potomnych

Klasa bazowa ICreature implementuje następujące metody związane z genotypem i wartością funkcji oceny:

- n () zwraca rozmiar tablicy zawierającej genotyp, czyli liczbę parametrów;
- getFitness() zwraca wartość funkcji oceny. Jeśli nie została ona wyznaczona, bądź jej wartość jest nieprawidłowa, to zrzucany jest jako wyjątek, obiekt klasy FitnessException;
- getGenotype() zwraca wskaźnik na tablicę wartości double będących genotypem osobnika;
- isWrongFitness() zwraca wartość true, jeśli wartość funkcji oceny jest nieprawidłowa lub nie została wyznaczona;
- setFitness() ustawia wartość funkcji oceny (metoda jest wykorzystywana przez klasy pochodne interfejsu IObjectiveFunctionDeterminator).

Ponadto deklaruje następujące metody wirtualne:

- setRandomGenotype() chroniona metoda do ustawiania losowego genotypu zgodnie z rozkładem równomiernym;
- setGaussGenotype() chroniona metoda do ustawiania losowego genotypu zgodnie z rozkładem Gaussa;
- cross() metoda przeprowadzająca krzyżowanie. Przyjmuje jako argument wskaźnik na drugiego osobnika. Zwraca wskaźnik na osobnik potomny;
- mutate() metoda przeprowadzająca mutację osobnika. Ustawia chronione pole wrongFitness na wartość true, gdyż wartość funkcji oceny staje się nieprawidłowa;
- print () metoda wypisująca osobnika do podanego jako argument strumienia wyjściowego.



Rys. A.4 Diagram klasy bazowej ICreatureGenerator i jej klas potomnych oraz klasy FitnessException

Klasa potomna BitsetCreature implementuje reprezentację binarną. Wymaga podania w konstruktorze informacji o liczbie bitów przeznaczonych do reprezentowania poszczególnych parametrów. Przechowuje reprezentację binarną (jako tablicę obiektów klasy std::bitset) równolegle z wymuszoną przez klasę bazową ICreature reprezentacją rzeczywistą. Dba, by w obu reprezentacjach był przechowywany ten sam genotyp. Argument operatora mutacji interpretuje jako prawdopodobieństwo mutacji (ang. *flip*) każdego z bitów dla każdego z parametrów.

Klasy potomne RealCreatureHX oraz RealCreatureDC implementują reprezentację rzeczywistą. Różnią się między sobą jedynie implementacją metody wirtualnej cross(), czyli operatorem krzyżowania. Argument operatora mutacji interpretują jako wartości odchyleń standardowych dla poszczególnych wymiarów zgodnie ze wzorem (3.16).

Klasa bazowa ICreaturesGenerator stanowi interfejs dla klas tworzących osobniki odpowiednich klas potomnych klasy ICreature. Wymaga od klas potomnych przekazania, jako argument konstruktora, wskaźnika na klasę IPhenotype, który następnie przekazywany jest każdemu tworzonemu osobnikowi. Ponadto deklaruje metodę wirtualną generateCreature(), której implementacje w klasach potomnych tworzą osobniki odpowiednich typów i zwracają je jako wskaźniki na typ bazowy ICreature.

A.1.4. Klasy IObjectiveFunctionDeterminator, IObjetiveFunction oraz IFitnessTransform

Klasy bazowe IObjectiveFunction i IObjectiveFunctionDeterminator oraz ich klasy potomne odpowiadają za wyznaczanie wartości funkcji celu. Klasa bazowa IFitnessTransform oraz jej klasy potomne realizują transformację wartości funkcji celu. Diagramy poszczególnych klas bazowych wraz z ich klasami potomnymi przedstawiają rysunki Rys. A.5, Rys. A.6 oraz Rys. A.7.

Klasa bazowa IObjectiveFunction jest interfejsem dla klas implementujących funkcjonalność wyznaczania wartości określonych funkcji celu dla zadanego punktu z przestrzeni rozwiązań dopuszczalnych. Zawiera deklarację jednej metody wirtualnej determine Fitness(), która przyjmuje dwa argumenty: wektor parametrów x (np. zbiór elementów macierzowych) oraz ich liczbę n. Zwraca liczbę rzeczywistą y podwójnej precyzji (double) będącą wartością funkcji celu dla zadanego punktu z przestrzeni rozwiązań dopuszczalnych.



Rys. A.5 Diagram klasy bazowej IObjectiveFunction i jej klas potomnych



Rys. A.6 Diagram klasy bazowej IObjectiveFunctionDeterminator i jej klas potomnych

IFitnessTransform Class			
🖂 🖃 Methods			
 i getFitnessTransforr i getInversionOfFitne i giv transform(double ch 	n(vector <icreature sssTransform(vector niSq) : double</icreature 	*>* creatures) : double* <icreature*>* creatures) : double*</icreature*>	*
public			public
LinearTransform Class	۲	LogarithmicTransform Class	8
➡ IFitnessTransform	7	IFitnessTransform	7

Rys. A.7 Diagram klasy bazowej IFitnessTransformijej klas potomnych

Nazwa klasy	Wartość zwracana przez metodę determineFitness ()
GosiaCaller	wektor x jest interpretowany jako zbiór elementów macierzowych ME. Zwracana jest wartość: $y = \chi^2 (ME)$
F7	funkcja F7 Schwefela. Przeskalowana i podniesiona o zdefiniowane przy inicjalizacji obiektu stałe scale oraz shift. Zwracana wartość wyraża się wzorem: $y = \sum_{n} \left[-(scale \cdot x) \cdot \sin(\sqrt{ scale \cdot x }) \right] + shift $ (3.17)
FConstant	funkcja stała. Zwraca wartość stałej, constValue, zdefiniowanej przy inicjalizacji obiektu. y=constValue
FNumbering	funkcja numerująca. Zwraca wartość licznika wywołań, callNumber, funkcji determineFitness() w instancji obiektu. y=callNumber++
FDistFrom RefPoint	zwraca odległość euklidesową zadanego wektora x od punktu p przestrzeni rozwiązań dopuszczalnych wskazanego przy inicjalizacji obiektu. $y=dist(x, p)$

Tabela A.2 Dostępne implementacje funkcji celu.

Podstawowa klasa implementującą interfejs IObjectiveFunction jest klasa GosiaCaller. Wymaga ona podania w konstruktorze: wskazania obiekt na GosiaPhenotype (przechowujący informacje pobrane z "inputu do GOSI") oraz lokalizacji programu GOSIA. Następnie uruchamia, jako oddzielny proces, program GOSIA i z jego "outputu" odczytuje wartość χ^2 dla podanego wektora elementów macierzowych. Jeśli w "outpucie programu GOSIA" nie ma informacji o wartości $\chi^2(ME)$, to metoda determineFitness() zwraca wartość stałej NO_CHI_SQ_IN_OUTPUT. Jeśli podczas uruchamiania programu GOSIA wystąpi błąd, to zrzucony zostanie jako wyjątek obiekt klasy GosiaException.

Wszystkie klasy implementujące interfejs IObjectiveFunction wraz z wartościami zwracanymi przez ich implementacje metody wirtualnej determineFitness() zostały przedstawione w Tabeli A.2.

Funkcja F7, ze względu na podobieństwo kształtu do funkcji χ^2 jest wykorzystywana w projekcie jako funkcja testowa. Zostało to szerzej omówione i przedstawione w kolejnych rozdziałach. Pozostałe implementacje interfejsu IObjectiveFunction były wykorzystywane na różnych etapach testów poprawności działania programu.

Klasa bazowa IObjectiveFunctionDeterminator odpowiada za wyznaczanie wartości funkcji celu dla całego pokolenia. Zawiera deklarację metody wirtualnej determine Fitnesses(), która jako argument otrzymuje wektor osobników, dla których należy wyznaczyć i ustawić (za pomocą metody ICreature::setFitness()) wartości funkcji celu. Ponadto klasa IObjectiveFunctionDeterminator implementuje następujące metody udostępniające informacje o przebiegu procesu wyznaczania wartości funkcji celu:

- getTotalDeterminationTime() zwraca ilość czasu jaka była potrzebna do wyznaczenia wartości funkcji celu;
- getNumberOfDeterminations() zwraca liczbę wyznaczeń wartości funkcji celu;
- resetMeasurement() resetuje pomiar.

Podstawową klasą potomną klasy IObjectiveFunctionDeterminator jest klasa LocalObjectiveFunctionDeterminator. Wymaga ona podania w konstruktorze wskazania na implementację interfejsu IObjectiveFunction, która ma być używana jako funkcja celu. Jako argument konstruktora można również podać wskaźnik na obiekt REP::PointSet²⁶, który będzie gromadzić próbkowanie przestrzeni rozwiązań dopuszczalnych przez omawiany algorytm genetyczny.

Jeżeli dla jakiegoś osobnika wartość zwrócona przez metodę IObjectiveFuncton:: determineFitness() jest mniejsza od zera, a więc spoza zakresu wartości funkcji celu, klasa LocalObjectiveFunctionDeterminator interpretuje to jako błąd i wywołuje powtórnie niniejszą metodę. Maksymalna liczba iteracyjnych prób wyznaczenia wartości funkcji celu dla osobnika zdefiniowana jest jako stała NUMBER_OF_FITNESS_DETERMINATION _TRIALS. Jeśli nadal nie udało się wyznaczyć wartości funkcji celu dla jakiegoś punktu

²⁶ Klasa REP:: PointSet jest częścią programu ScanRep. Jej opis znajduje się w dodatku A.3.3.

z przestrzeni rozwiązań dopuszczalnych, to metoda ICreature:: isWrongFitness() odpowiadającego mu osobnika będzie zwracać wartość TRUE, a próba wywołania jego metody ICreature:getFitness() zrzuci jako wyjątek obiekt klasy FitnessException.

Drugą klasą potomną klasy IObjectiveFunctionDeterminator jest klasa RemoteObjectiveFunctionDeterminator. Stanowi ona punkt możliwej przyszłej implementacji równoległego wyznaczania wartości funkcji celu dla całego pokolenia.

Klasa bazowa IFitnessTransform zawiera deklarację chronionej metody wirtualnej transform(), która jako argument przyjmuje pierwotną wartość funkcji celu, a zwraca jej wartość przekształconą zgodnie z przyjętą transformacją. Ponadto klasa definiuje dwie metody publiczne:

- getFitnessTransform() dla zadanego wektora osobników, zwraca tablicę ich przekształconych wartości funkcji celu;
- getInversionOfFitnessTransform() dla zadanego wektora osobników, zwraca tablicę odwrotności ich przekształconych wartości funkcji celu.

Klasa bazowa IFitnessTransform posiada dwie klasy potomne:

- LinearTransform brak transformacji. Implementacja metody transform()
 zwraca pierwotną wartość funkcji celu;
- LogarithmicTransform transformacja logarytmiczna. Implementacja metody transform() zwraca logarytm naturalny z powiększonej o jeden wartości funkcji celu.

Instancja wybranej klasy potomnej klasy IFitnessTransform przechowywana jest przez obiekt Population i wykorzystywana przez klasy implementujące operatory genetyczne.

A.1.5. Klasa IGeneticOperator

Klasa bazowa IGeneticOperator oraz jej klasy potomne odpowiadają za realizację operacji genetycznych na zadanym wektorze osobników. Diagramy wszystkich klas związanych z operatorami genetycznymi przedstawiają rysunki Rys. A.8 i Rys. A.9.

Klasa IGeneticOperator implementuje jedynie metody do rejestracji czasu działania poszczególnych operatorów genetycznych. Poprzez chronioną metodę setTime(), umożliwia klasom potomnym zapis pomiaru czasu operatora dla danego pokolenia, a publiczna metoda getTime() zwraca czas dotychczasowych obliczeń. Poszczególne klasy potomne mogą przeciążać wirtualną publiczną metodę reset(), udostępniając reset działania operatora genetycznego oraz zmianę jego parametrów (podawanych jako argumenty niniejszej metody).

Po klasie bazowej IGeneticOperator dziedziczą trzy klasy określające operatory genetyczne:

- ISelection klasa bazowa dla operatorów selekcji;
- ICrossingOver klasa bazowa dla operatorów krzyżowania;
- IMutation klasa bazowa dla operatorów mutacji.

Dodatkowo klasy implementujące operatory genetyczne mogą skorzystać z klasy ICumulativeDistribution, która udostępnia wyznaczanie dystrybuanty. Wykorzystywana jest ona do konstrukcji "koła ruletki" przez operatory: selekcji ruletkowej (klasa SelectionByRoulette) oraz krzyżowania (klasa CrossingOverByRoulette).



Rys. A.8 Diagram klasy bazowej IGeneticOperator i jej klas potomnych oraz klasy ICumulativeDistribution





Rys. A.9 Diagram klas wszystkich operatorów genetycznych oraz ich klas bazowych

Podstawową metodą klasy ISelection jest funkcja selection(). Przyjmuje ona jako argumenty: wskazanie na wektor osobników, dla których ma przeprowadzić selekcję, oraz wskazanie na obiekt IFitnessTransform. Przeprowadza ona selekcję w oparciu o wirtualną metodę chronioną getCasualties(), która poprzez poszczególne klasy potomne dostarcza właściwej implementacji dla wyznaczania ofiar operatora selekcji. W omawianym projekcie są trzy klasy potomne klasy ISelection, implementujące opisane w punkcie 3.2.3 operatory selekcji:

- SelectionByTruncation operator selekcji elitarnej. Wymaga podania, jako argument konstruktora, procentu osobników, którzy mają być eliminowani przez operator;
- SelectionByRoulette operator selekcji ruletkowej. Wymaga podania, jako argument konstruktora, procentu osobników, którzy mają być eliminowani przez operator;
- SelectionByTournament operator selekcji turniejowej. Wymaga podania, jako argument konstruktora, rozmiaru turnieju.

Podstawową metodą klasy ICrossingOver jest funkcja crossing_over(). Przyjmuje ona jako argumenty: wskazanie na wektor osobników, który ma zostać uzupełniony o osobniki potomne z wykorzystaniem pozostałych osobników jako rodziców, docelowy rozmiar populacji oraz wskazanie na obiekt IFitnessTransform. Przeprowadza ona krzyżowanie w oparciu o wirtualną metodę chronioną selectParent(), która poprzez poszczególne klasy potomne dostarcza właściwej implementacji dla wybierania osobników na rodziców. W omawianym projekcie jest tylko jedna klasa potomna interfejsu ICrossingOver, realizująca wszystkie opisane w punkcie 3.2.4 operatory krzyżowania:

 CrossingOverByRoulette – operator krzyżowania wykorzystujący "koło ruletki" do wyboru rodzica. Wymaga podania, jako argument konstruktora, maksymalnej liczby dzieci na rodzica.

Sam proces krzyżowania dwóch osobników jest zależny od przyjętej reprezentacji instancji problemu, dlatego realizowany jest w odpowiednich klasach przechowujących informacje o osobniku (ICreature).

Klasa IMutation wymaga od klas potomnych przekazania, jako argument konstruktora, wskaźnika na tablicę odchyleń standardowych znormalizowanych do zakresów zmienności poszczególnych parametrów, bazowego prawdopodobieństwa mutacji oraz liczby parametrów (niezależnych elementów macierzowych). Podstawową metodą klasy IMutation jest funkcja mutation(). Przyjmuje ona jako argumenty: wskazanie na wektor osobników, dla których

ma przeprowadzić mutację oraz numer pokolenia. Przeprowadza ona mutację w oparciu o wirtualną metodę chronioną getMutationRatio(), która poprzez poszczególne klasy potomne dostarcza właściwej implementacji dla wyznaczania współczynnika mutacji w funkcji numeru pokolenia. W omawianym projekcie są trzy klasy potomne klasy IMutation, implementujące opisane w punkcie 3.2.5 operatory mutacji:

- Mutation_Constant operator mutacji, w którym współczynnik mutacji pozostaje niezmienny podczas całego procesu optymalizacji;
- Mutation_Hyperbolic operator mutacji, w którym współczynnik mutacji maleje hiperbolicznie wraz z numerem pokolenia;
- Mutation_Logarithmic operator mutacji, w którym współczynnik mutacji maleje hiperbolicznie wraz z logarytmem z numeru pokolenia.

A.1.6. Klasa IPhenotype

Klasa bazowa IPhenotype oraz jej klasy potomne i pomocnicze odpowiadają za zebranie i przechowywanie informacji o fenotypie (dziedzinie funkcji celu). Dostarczają również informacji potrzebnych do zdefiniowania przestrzeni rozwiązań dopuszczalnych (genotypu) oraz umożliwiają przekształcenie genotypu osobnika w jego fenotyp. Diagram klas za to odpowiedzialnych przedstawia Rys. A.10.

Klasa bazowa IPhenotype dostarcza informacji o przestrzeni rozwiązań dopuszczalnych. Zawiera deklaracje następujących metod wirtualnych:

- numberOfParameters() zwraca liczbę parametrów (elementów macierzowych);
- min(i) zwraca wartość minimalną i-tego parametru;
- max(i) zwraca wartość maksymalną i-tego parametru;
- getProposedStartingPoint() zwraca punkt z przestrzeni rozwiązań dopuszczalnych wokół którego zostanie rozlosowana z rozkładem Gaussa populacja początkowa, w przypadku wyboru tej opcji przez użytkownika. Współrzędne punktu są pobierane z "inputu do GOSI" lub ze wskazanego strumienia (np. z pliku .me opisanego w dodatku B.2).

Ponadto definiuje metody:

- range(i) zwraca zakres zmienności i-tego parametru, tj. różnicę max(i) min(i);
- print () wypisuje do wskazanego strumienia informacje o parametrach.



Rys. A.10 Diagram klasy bazowej IPhenotype, jej klasy potomnej oraz klas pomocniczych

W omawianym projekcie jedyną klasą pochodną klasy IPhenotype jest klasa GosiaPhenotype. Przechowuje ona informacje pozyskane z "inputu do GOSI", którego lokalizacja została wskazana jako argument konstruktora. Jeśli instancja klasy GosiaPhenotype nie jest w stanie prawidłowo odczytać informacji z "inputu do GOSI", zrzucany jest, jako wyjątek, obiekt klasy GosiaException. Sytuacja taka wystąpi jeśli:

- plik z "inputem do GOSI" nie został znaleziony we wskazanej lokalizacji;
- w pliku nie występuje znacznik "ME" brak informacji o zakresach zmienności poszczególnych elementów macierzowych;
- w pliku nie występuje znacznik "OP, FILE" opcja "12, 3, 1" brak informacji o nazwie pliku, w którym ma być umieszczany zestaw elementów macierzowych, dla których program GOSIA wyznaczy wartość χ²(ME);
- w pliku nie występuje znacznik "OP, FILE" opcja "22, 3, 1" brak informacji o nazwie pliku, z którego ma być odczytywana wyznaczona wartość χ²(ME).

Poza implementacjami metod wirtualnych klasy IPhenotype, klasa GosiaPhenotype zawiera definicje następujących metod:

- numberOfAllME() zwraca liczbę wszystkich elementów macierzowych (zarówno niezależnych jak i pozostających z nimi w ścisłej korelacji);
- determineAllME() zwraca wektor wszystkich elementów macierzowych w oparciu o wektor niezależnych elementów macierzowych (przekształca genotyp w fenotyp danego osobnika);
- getInputFileLocalization() zwraca lokalizację "inputu do GOSI", który został podany jako argument konstruktora przy tworzeniu obiektu;
- getMEfileName() zwraca nazwę pliku, w którym ma być umieszczany zestaw elementów macierzowych, dla którego program GOSIA wyznaczy wartość $\chi^2(ME)$.

Klasa GosiaPhenotype przechowuje pozyskane z "inputu do GOSI" informacje o wszystkich elementach macierzowych jako wskaźnik na obiekt std::vector. Elementami kontenera są wskaźniki na obiekty struktury MEinfo. Nazwy pól niniejszej struktury są zgodne z dokumentacją techniczną programu GOSIA. Stanowią tym samym wraz z całą klasą GosiaPhenotype punkt styku sposobu przechowywania i reprezentowania informacji w programie GOSIA i w programie JACOB.

A.2. Budowa programu RepGen

Program RepGen (skrót od ang. *Repository Generator*) służy do tworzenia repozytoriów o zadanych kryteriach. Dla generowanego zbioru punktów, użytkownik wybiera rodzaj funkcji celu oraz sposób próbkowania przestrzeni.

Select Input	Repository Generator
Input file:	Change Input
D:/2d (-40,15)(-40,15).inp Browse	Number of points: 1000
.gin file (Generator Information)	Generator mode: Random
Browse	Objective function: E7
Gosia localization:	Scale: 1,0 🗘
Browse	Shift: 50,00 🗘
ОК	Generate



Rys. A.12 Okno główne programu RepGen

3
Save Close

Rys. A.13 Okno zapisu repozytorium programu RepGen



Rys. A.14 Schemat działania programu RepGen

Jak pokazano na Rys. A.11, program RepGen wymaga podania "inputu do GOSI", na podstawie którego ma być utworzone repozytorium. Pobierana jest z niego informacja o liczbie i zakresach zmienności poszczególnych parametrów (znacznik ME). Jeśli funkcją celu ma być funkcja χ^2 (**ME**), której wartości wyznacza program GOSIA, to należy wskazać jego lokalizację. Ponadto niektóre sposoby generowania repozytorium wymagają podania dodatkowych informacji poprzez wskazanie lokalizacji pliku .gin. Jego budowę przedstawiono w punkcie A.5.

Wybór kryteriów, na podstawie których ma być wygenerowane repozytorium, odbywa się za pomocą okna głównego programu, przedstawionego na Rys. A.12. Użytkownik określa:

- liczbę punktów, z których ma się składać repozytorium;
- sposób próbkowania przestrzeni. Do wyboru spośród próbkowania:

- losowego, ang. *random* metoda Monte Carlo,
- ° na siatce równomiernej, ang. grid punkty są rozmieszczone w równych odstępach,
- na siatce dowolnej, ang. *mesh value* współrzędne punktów dla poszczególnych wymiarów są określone we wskazanym pliku .gin,
- ° Gaussa wokół punktu określonego w "inpucie do GOSI",
- Gaussa wokół punktu określonego we wskazanym pliku .gin;
- funkcję celu. Do wyboru spośród funkcji zebranych w Tabeli A.2.

Jak widać na Rys. A.13, rezultatem działania programu RepGen jest wygenerowane repozytorium. Użytkownik może je zapisać do pliku. Format pliku przedstawiono w punkcie A.4.

Program RepGen został zaimplementowany w języku C++, przy pomocy środowiska Microsoft Visual Studio 2010 oraz biblioteki Qt 4.8.7. Jego schemat działania zawiera Rys. A.14. Program jest w istocie graficznym interfejsem użytkownika dla wykorzystywanych powtórnie klas powstałych przy okazji pisania programów JACOB oraz ScanRep. Został utworzony w celu szybkiego generowania repozytoriów z różnymi rozkładami próbkowania przestrzeni. Usprawniło to proces porównywania sposobu próbkowania przestrzeni przez algorytm genetyczny z innymi metodami.

Program RepGen wykorzystuje następujące klasy programu JACOB:

- klasę bazową IObjectiveFunction oraz jej klasy potomne: GosiaCaller, F7, FConstant, FNumbering, FDistFromRefPoint omówione w punkcie A.1.4;
- klasę IPhenotype, jej klasę potomną GosiaPhenotype oraz klasy pomocnicze: MEinfo i GosiaException – omówione w punkcie A.1.6.

Ponadto program RepGen korzysta z następujących klas programu ScanRep:

- PointSet, PointSetException oraz Point omówione w punkcie A.3.3;
- Generator omówiona w punkcie A.3.5.

A.3. Budowa programu ScanRep

Niniejszy punkt stanowi rozwinięcie punktu 5.4. Zawiera szczegółowe omówienie wszystkich klas programu ScanRep. Stanowią one jedną przestrzeń nazw namespace REP. Można wśród nich wyróżnić następujące grupy:

• klasy realizujące tekstowy interfejs użytkownika – IMenu oraz klasy potomne;

- klasy zarządzające całym repozytorium oraz wyodrębnionymi klastrami Repository oraz Cluster;
- klasy związane z reprezentacją i przechowywaniem próbkowania przestrzeni PointSet, PointSetException oraz Point;
- klasy zawierające implementacje algorytmów IAlgorithm oraz klasy potomne;
- klasa przechowująca implementacje generatorów liczb pseudolosowych Generator.

Jak widać z powyższego zestawienia w programie ScanRep wykorzystywany jest paradygmat dziedziczenia i polimorfizmu.



A.3.1. Klasa IMenu

Rys. A.15 Diagram klasy bazowej IMenu i jej klas potomnych

Klasa bazowa IMenu oraz jej klasy potomne odpowiadają za realizację tekstowego interfejsu użytkownika. Diagram klasy bazowej IMenu oraz jej klas potomnych przedstawia Rys. A.15.

Klasa bazowa IMenu deklaruje metodę wirtualną run (). Jej definicje w klasach potomnych odpowiadają za wyświetlenie oraz pracę danego menu. Ponadto definiuje ona metodę
destinationStream(), która zwraca wskaźnik do strumienia do pliku wskazanego przez użytkownika, do którego mają być zapisane rezultaty pracy. Jeśli informacja ma być wypisana na konsolę to metoda zwraca wartość NULL.

Klasa potomna RepositoryMenu implementuje menu przetwarzania repozytorium. Wymaga podania w konstruktorze lokalizacji: właściwego "inputu do GOSI" oraz pliku .rep z repozytorium. Na ich podstawie tworzy i przechowuje obiekty: GA::IPhenotype oraz Repository. Prywatne metody klasy RepositoryMenu odpowiadają za pobranie od użytkownika określonych argumentów i wywołanie odpowiednich metod klasy Repository.

Klasa potomna ClusterMenu implementuje menu przetwarzania klastra. Wymaga ona podania w konstruktorze wskazań na obiekty GA::IPhenotype oraz Cluster. Jej prywatne metody odpowiadają za pobranie od użytkownika określonych argumentów i wywołanie odpowiednich metod klasy Cluster.



A.3.2. Klasy Repository oraz Cluster

Rys. A.16 Diagram klasy Repository oraz klasy Cluster

Klasy Repository oraz Cluster odpowiadają za zarządzanie odpowiednio całym repozytorium oraz wyodrębnionym klastrem. Stanowią one fasady wykorzystywane przez klasy realizujące interfejs użytkownika. Dostarczają funkcjonalność niezbędną do przetwarzania całego repozytorium oraz klastrów. Wewnętrzna reprezentacja zbioru punktów w obiektach typu Repository oraz Cluster oparta jest na instancjach klasy PointSet. Diagramy klas Repository oraz Cluster przedstawia Rys. A.16.

Klasa Repository wymaga podania w konstruktorze wskazania na właściwy obiekt GA::IPhenotype oraz lokalizacji pliku .rep z repozytorium. Na ich podstawie tworzy swoją instancję klasy PointSet. Poza wskaźnikami na obiekty GA::IPhenotype oraz PointSet przechowuje ona również zbiór klastrów wyodrębnionych podczas grupowania – jako wskaźnik na obiekt std::vector. Elementami kontenera są wskaźniki na obiekty typu Cluster. Nad prawidłową zawartością wektora klastrów czuwa prywatna metoda updateClusterVector().

Klasa Cluster wymaga podania w konstruktorze wskazania na właściwy obiekt GA::IPhenotype. Tworzona jest pusta instancja klasy PointSet. Dodawanie punktów do klastra odbywa się poprzez metodę addPoint().

A.3.3. Klasy Point oraz PointSet



Rys. A.17 Diagram klasy Point

Klasa Point odpowiada za reprezentację danych. Obiekty tej klasy przechowują informację o pojedynczym punkcie. Klasa PointSet odpowiada za przechowywanie próbkowania przestrzeni. Zbiór punktów zebrany jest w kontenerze std::vector, którego elementami są wskaźniki na obiekty typu Point. Jeśli podczas przetwarzania zbioru punktów wystąpi błąd, to zrzucany jest jako wyjątek obiekt klasy PointSetException. Diagramy klas Point, PointSet oraz PointSetException przedstawiają Rys. A.17 oraz Rys. A.18.

PointSet	8
C1422	7
🖃 Fields	
 distance : float** generator : Generator* nDimensions : const int ph : IPhenotype* points : vector <point*>*</point*> 	
Methods	
 add(Point* p, bool suppressSortingByValue) : bool append(ifstream* stream) : void at(int i) : Point* calculateParameterRanges() : void closestPoint(Point* p) : Point* cutToPhenotype(bool deleteCasualties) : void generate(const int* grid, bool includeBoundaryPoints, IObjectiveFunction* objFunc, void(* generate(int toGenerate, const double* mi, const double** R, bool cutToPhenotype, IObje generate(int toGenerate, const double* mi, double sigmaFactor, IObjectiveFunction* objFunc generate(int toGenerate, const double* mi, double sigmaFactor, IObjectiveFunction* objFunc generate(int toGenerate, const double* mi, double sigmaFactor, IObjectiveFunction* objFunc generate(vector<double>** meshValue, IObjectiveFunction* objFunc, void(*)(int percent.</double> getBestPoint() : const Point* getDistance(int i, int j, float moreThen) : float getDistancesToPoint(int dim) : double intervalInPhenotype(int dim) : double intervalInPhenotype(int dim) : double maxInPhenotype(int dim) : double minInPhenotype(int dim) : double minInPhenotype(int dim) : double minInPointset(int dim) : float pointSet(IPhenotype* ph) PointSet(IPhenotype* ph, string repFileName, bool oldSyntax) PointSet(PointSet* ps, bool deleteCasualties, bool copyDistancesArray, double threshold, dou printDistancesThread(bool forceDistancesRecalculation) : void shift(double shiftValue) : void)(int percentage) ctiveFunction* obj nc, void(*)(int p) : void age) progressInfo
stopCalculateDistancesThread() : void	
PointSetException (Class)



Rys. A.18 Diagram klasy PointSet oraz klasy PointSetException

Klasa PointSet wymaga podania w konstruktorze wskazania na obiekt IPhenotype. Jeśli zbiór punktów ma być wczytany z pliku to należy podać jego lokalizację. Klasa PointSet posiada ponadto jeden konstruktor kopiujący wykorzystywany do realizacji etapów B i C Metody AH.

Metody klasy PointSet można podzielić na trzy grupy:

- kreatory powiększające zbiór punktów: add(), append(), generator();
- manipulatory-przetwarzające zbiór punktów: cutToPhenotype(), shift(), sortByValue();
- informatory dostarczające informacji o zbiorze punktów: pozostałe metody.

Klasa PointSet umożliwia obliczenie i zapisanie, w postaci macierzy trójkątnej, odległości pomiędzy każdą parą punktów. Usprawnia to przetwarzanie zbioru punktów, w szczególności radykalnie przyspiesza działanie algorytmu klasteryzacji oraz FLA. Macierz odległości jest usuwana w przypadku wywołania metody z grupy kreatorów lub manipulatorów, z wyjątkiem metody shift().

W klasie PointSet przechowywane w kontenerze punkty są posortowane rosnąco ze względu na wartość $f(\mathbf{x})$. Wywołanie metody z grupy kreatorów powoduje automatyczne wywołanie metody sortByValue(). Metoda add() udostępnia możliwość wstrzymania sortowania. Jest to przydatne w przypadku sekwencyjnego dodawania większej liczby punktów. Po dodaniu ostatniego punktu należy pamiętać o wywołaniu metody sortByValue().

A.3.4. Klasa IAlgorithm

Klasa bazowa IAlgorithm oraz jej klasy potomne odpowiadają za realizację etapów: D, E, F, G oraz H Metody AH. Diagram klasy bazowej IAlgorithm oraz jej klas potomnych przedstawia Rys. A.19.

Klasa bazowa IAlgorithm wymaga od klas potomnych przekazania, jako argument konstruktora, wskaźnika na obiekt PointSet, na którym będzie pracował dany algorytm. Ponadto definiuje chronioną metodę findKthSmallestInArray(), która zwraca wartość k-tego najmniejszego elementu tablicy otrzymanej jako argument. Metoda ta jest wykorzystywana przez klasy AlgNBC oraz AlgFrontLine.

Klasa AlgNBC implementuje opisany w punkcie 4.5 algorytm klasteryzacji. Na podstawie posiadanego obiektu PointSet, dla każdego punktu, tworzy instancję struktury NBCdata. W tej kolekcji przechowywane są dane potrzebne do pracy algorytmu NBC. Grupowanie odbywa się poprzez wywołanie metody findClusters() z argumentem k. Metoda mergeClusters() zawiera implementację opisanej w podpunkcie 4.5.1. procedury łączenia przylegających klastrów stanowiącej autorską modyfikację algorytmu NBC.



Rys. A.19 Diagram klasy bazowej IAlgorithmijej klas potomnych

Generator Class	 ?
Fields	
 a : const4intVector b : const4intVector c : const4intVector d : const4intVector t : int[4] 	
Methods	
 gaussian(): double gaussian(const double* mi, const double** R, int n): double* gaussian(double mi, double sigma): double gaussian(double mi, double sigma, double min, double max): double Generator() notTooSmall(double x): double readGINfile(const int n, string generatorInfoFileName): GeneratorInf rectangular(): double rectangular(double min, double max): double rectangular(int min, int max): int round(double x): int 	; :o*
Nested Types	
···	
GeneratorInfo Struct	
🖃 Fields	



1

Rys. A.20 Diagram klasy Generator oraz struktury GeneratorInfo

Klasa AlgStatistics implementuje opisane w punkcie 4.6 metody statystycznego badania rozkładu próbkowania. Umożliwia pobranie surowych danych w postaci dwuwymiarowej tablicy albo ich zapis do wskazanego strumienia.

Klasa AlgHistograms implementuje opisane w punkcie 4.7 metody badania rozkładu próbkowania w oparciu o odległości między punktami. Umożliwia pobranie surowych danych w postaci wskazania na obiekt std::map albo ich zapis do wskazanego strumienia.

Klasa AlgFrontLine implementuje opisany w punktach 5.2 oraz 5.3 moduł wyznaczania niepewności parametrów. Umożliwia pobranie surowych wyników działania algorytmu FLA

(wektor etykiet) oraz dwuwymiarowej tablicy zawierającej wyznaczone niepewności poszczególnych parametrów. Powyższe dane mogą być zapisane do wskazanego strumienia. Klasa AlgFrontLine korzysta z zagnieżdżonej klasy Vector, której obiekty przyspieszają obliczenia oraz poprawiają czytelność kodu metod approximatePU() oraz approximateValue().

A.3.5. Klasa Generator

Klasa Generator zawiera implementacje generatorów liczb pseudolosowych napisane zgodnie z zalecaniami zawartymi w [112] oraz [115]. Podstawowy generator, losujący z rozkładem równomiernym liczbę z przedziału [0,1], stanowi implementację poprawionego generatora Wichmanna-Hilla [128] z cyklem o długości 2¹²¹. Diagram klasy Generator oraz pomocniczej struktury GeneratorInfo przedstawia Rys. A.20. Klasa Generator korzysta z zagnieżdżonej klasy const4intVector, której obiekty poprawiają czytelność kodu inicjalizującego ziarna generatora liczb pseudolosowych. Ponadto umieszczono w niej definicje kilku pobocznych metod.

Klasa Generator definiuje następujące metody:

- rectangular() generator liczb pseudolosowych z rozkładem równomiernym;
- gaussian() generator liczb pseudolosowych z rozkładem Gaussa;
- notTooSmall() zaokrągla liczbę z przedziału [-0.0001, 0.0001] do zera;
- round () zaokrągla liczbę rzeczywistą do całkowitej;
- readGINfile() zwraca wskaźnik do utworzonej instancji struktury GeneratorInfo, która zawiera informacje do generowania zbioru punktów przeczytane ze wskazanego pliku.

A.4. Plik z repozytorium próbkowania przestrzeni

Zbiór punktów z badanej przestrzeni przechowywany jest w:

- pamięci operacyjnej jako obiekt klasy REP::PointSet;
- w pamięci masowej jako plik tekstowy o rozszerzeniu . rep.

Przestrzeń definiowana jest w:

- przestrzeni operacyjnej przez obiekt klasy GA::IPhenotype, na który wskazuje odpowiednie pole powiązanej z nią instancji klasy REP::PointSet;
- pamięci masowej przez plik tekstowy o rozszerzeniu .inp, zawierający przynajmniej znacznik "ME" oraz znacznik "OP, FILE" opcja "12, 3, 1" z "inputu do GOSI".

Aby utworzyć obiekt REP::PointSet, niezbędna jest właściwa instancja klasy GA::IPhenotype. Oznacza to, że do wczytania pliku z repozytorium (.rep), należy wskazać odpowiedni plik z definicją przestrzeni (.inp).

Podstawowym założeniem związanym ze strukturą pliku z repozytorium jest, aby w jednej linijce znajdowała się informacja o jednym punkcie przestrzeni²⁷. W pliku mogą też występować puste linie oraz linie zaczynające się od znaku # – są one ignorowane klasę REP::PointSet. Linijka tekstu zawierająca informację o jednym punkcie ma następującą budowę:

clusterID y x1 x2 … xn

gdzie:

clusterID – jest liczbą całkowitą nieujemną wskazującą na przynależność punktu do klastra (tzw. szum ma clusterID = 0). Jeśli nie przeprowadzono klasteryzacji to clusterID = -1, y – jest liczbą rzeczywistą nieujemną stanowiącą wartość funkcji celu. Jeśli nie wyznaczono wartości $f(\mathbf{x})$ to y = -1,

 $x_1 \dots x_n$ – są liczbami rzeczywistymi, wartościami poszczególnych parametrów.

Poszczególne liczby oddzielone są białymi znakami (spacja, tabulacja).

Jeśli w linijce znajduje się więcej parametrów niż wartość zwrócona przez metodę GA::IPhenotype::numberOfParameters(), to nadmiarowe parametry są ignorowane. Jeśli w linijce brakuje części informacji o punkcie, to konstruktor klasy REP::PointSet zrzuci jako wyjątek obiekt klasy REP::PointSetException, a wczytywanie zostanie anulowane.

Przykładową zawartość pliku z repozytorium próbkowania przestrzeni określonej w punkcie 4.1. przedstawia Rys. A.21. Zbiór zawiera 9 punktów: pięć należy do klastra nr 1, trzy do klastra nr 2, a dwa punkty to tzw. szum. Poszczególne grupy punktów oddzielone są dwiema pustymi liniami.

1 1.000 -39.931 -9.003	<i># Vector of Expectations:</i>
1 2.000 -29.368 4.481	23
1 3.000 -7.824 -13.607	
1 4.000 -20.734 9.278	<i># Covariance Matrix:</i>
	2.0 1.9
	1.9 2.0
2 5.000 -30.424 7.242	
2 6.000 -0.922 -11.756	<i># Grid vector:</i>
2 7.000 -23.280 -39.176	68
	# Mesh values:
0 8.000 -34.973 -19.955	-25.0 -15.0 0.0 5.0
0 9.000 -31.898 -30.876	-20.0 -10.0 10.0
0 8.000 -34.973 -19.955 0 9.000 -31.898 -30.876	-25.0 -15.0 0.0 5.0 -20.0 -10.0 10.0



Rys. A.22 Przykładowa zawartość pliku .gin

²⁷ Ułatwia to ewentualne otwarcie repozytorium przez programy takie jak: Gnuplot, Matlab, Excel...

A.5. Plik z informacjami do generowania repozytorium

W pliku .gin zawarte są informacje wymagane przez niektóre sposoby generowania zbioru punktów w programie RepGen. Przy próbkowaniu przestrzeni:

- wielowymiarowym rozkładem Gaussa (ang. multivariate (joint) Gaussian distribution)

 należy podać bloki: "Vector of Expectations" oraz "Covariance Matrix".

 Zawierają one wektor wartości oczekiwanych oraz macierz kowariancji;
- na siatce równomiernej należy podać blok "Grid vector". Zawiera on liczbę interwałów dla każdego wymiaru, na które mają być podzielone zakresy zmienności. Wygenerowany zbiór punktów jest iloczynem kartezjańskim wartości interwałowych;
- na siatce dowolnej należy podać blok "Mesh values". Zawiera on dla każdego wymiaru zbiór wartości jakie ma przyjmować dany parametr. Wygenerowany zbiór punktów jest iloczynem kartezjańskim tych zbiorów. Jeśli któraś wartość w zbiorze jest spoza zakresu zmienności danego parametru, to zostanie ona zignorowana.

Przykładową zawartość pliku .gin przedstawia Rys. A.22. Zawiera on wszystkie cztery bloki z danymi dla dwuwymiarowego przypadku testowego omawianego w rozdziale 4. Każdy blok rozpoczyna się od linijki z sygnaturą: znak # oraz nazwa bloku. Blok kończy się pustą linią.

Liczba wymiarów musi być zgodna z liczbą parametrów w obiekcie GA::IPhenotype (wczytana z "inputu do GOSI"). Jeśli jest większa, to metoda REP::Generator:: readGINfile(), tworząc obiekt REP::GeneratorInfo, zignoruje nadmiarowe dane. Jeśli jest mniejsza to określony blok nie zostanie wczytany.

<i>OP,FILE</i>					
22,3,1					
132ba-v5-hil-iuac_8D.out					
12,3,1					
132ba mini-hil-iuac.me					
()					
ME					
2,0,0,0,0					
1 2 0.927 .01 2	2.0				
1 4 0.27 .01 2	2.0				
2 2 .1 -2.0 2	2.0				
2 3 .1 .01 2	2.0				
2 4 .1 -2.0 2	2.0				
3 4 .1 -2.0 2	2.0				
4 4 .1 -2.0 2	2.0				
70000					
2 4 0.1 -2.0 2	2.0				
00000					
()					

0.963035
0.257394
0.689220
1.38342
0.925758
-0.457178
-0.701038
0.639200

Rys. B.2 Plik z wartościami elementów macierzowych

()		
*** CHISQ=	0.157068E+01	* * *
()		

Rys. B.3 Fragment pliku wyjściowego programu GOSIA

Dodatek B – Komunikacja z programem GOSIA

Program GOSIA jest wywoływany jako oddzielny proces przez program JACOB oraz RepGen. Odpowiada za to klasa GA::GosiaCaller.

B.1. Plik wejściowy programu GOSIA

Zamieszczony na Rys. B.1 fragment pliku wejściowego programu GOSIA, tzw. "input do GOSI", był używany w analizie danych eksperymentalnych omawianej w rozdziale 6. Zawiera on znaczniki (pisane wielkimi literami) i parametry konfiguracyjne programu. Szczegółowy opis jego budowy znajduje się w [9].

Omawiana w punkcie A.1.6 klasa GosiaPhenotype pobiera z niego:

- liczbę i zakresy zmienności elementów macierzowych (znacznik "ME");
- nazwę pliku, w którym ma być umieszczany zestaw elementów macierzowych, dla którego program GOSIA wyznaczy wartość χ²(ME) (znacznik "OP, FILE" opcja "12, 3, 1");
- nazwę pliku, z którego ma być odczytywana wyznaczona wartość χ²(ME) (znacznik "OP, FILE" opcja "22, 3, 1").

Pozostałe informacje (kilkaset linii) dotyczą m. in. struktury plików wejściowych oraz geometrii układu eksperymentalnego. Z punktu widzenia programów JACOB, RepGen oraz ScanRep są nieistotne. Ze względu na objętość niniejszej dysertacji zostały pominięte.

B.2. Plik z wartościami elementów macierzowych

Zestaw elementów macierzowych, dla których program GOSIA ma wyznaczyć wartość χ^2 (**ME**), jest umieszczany jako wektor kolumnowy w pliku tekstowym (vide Rys. B.2).

B.3. Wynik programu GOSIA

Rezultat pracy programu GOSIA jest umieszczany w dwóch miejscach:

- na konsolę wypisywana jest lista przetworzonych plików;
- do pliku wyjściowego zapisywane są wszystkie informacje dotyczące pracy programu GOSIA.

Plik wyjściowy, tzw. "output GOSI" (vide Rys. B.3), jest analizowany przez metodę GA::GosiaCaller::gosiaOutputFileParsing(). Pobierana jest z niego wartość χ^2 (ME). Pozostałe informacje (kilkaset linii) dotyczą pracy programu GOSIA – z punktu widzenia programu JACOB są nieistotne. Ze względu na objętość pracy zostały pominięte.