



You have downloaded a document from  
**RE-BUŚ**  
repository of the University of Silesia in Katowice

**Title:** Zastosowanie procesów epigenetycznych w algorytmach genetycznych

**Author:** Kornel Chromiński

**Citation style:** Chromiński Kornel. (2019). Zastosowanie procesów epigenetycznych w algorytmach genetycznych. Katowice : Uniwersytet Śląski

© Korzystanie z tego materiału jest możliwe zgodnie z właściwymi przepisami o dozwolonym użytku lub o innych wyjątkach przewidzianych w przepisach prawa, a korzystanie w szerszym zakresie wymaga uzyskania zgody uprawnionego.



UNIWERSYTET ŚLĄSKI  
W KATOWICACH



Biblioteka  
Uniwersytetu Śląskiego



Ministerstwo Nauki  
i Szkolnictwa Wyższego

Uniwersytet Śląski w Katowicach  
Wydział Informatyki i Nauki o Materiałach  
Instytut Informatyki

Rozprawa doktorska

Kornel Chromiński

**Zastosowanie procesów epigenetycznych  
w algorytmach genetycznych**

Promotor: dr hab. inż. Mariusz Boryczka

Promotor Pomocniczy: dr Magdalena Tkacz

Sosnowiec 2019

"Co my wiemy, to tylko kropelka. Czego nie wiemy, to cały ocean."

*Isaac Newton*

---

# Spis treści

<b>Spis treści</b>	<b>3</b>
<b>Wstęp</b>	<b>6</b>
<b>1 Algorytmy genetyczne w problemach optymalizacyjnych</b>	<b>10</b>
1.1 Problemy optymalizacyjne . . . . .	11
1.2 Biologiczne podstawy algorytmów ewolucyjnych . . . . .	13
1.3 Algorytmy ewolucyjne . . . . .	16
1.4 Algorytmy genetyczne . . . . .	18
1.4.1 Klasyczny algorytm genetyczny . . . . .	18
1.4.2 Elementy składowe algorytmów genetycznych . . . . .	20
1.4.3 Wybrane modyfikacje klasycznego algorytmu genetycznego . . . . .	29
1.4.4 Przykładowe zastosowania algorytmów genetycznych . . . . .	32
<b>2 Wybrane problemy optymalizacyjne</b>	<b>34</b>
2.1 Problem optymalizacji załadunku . . . . .	34
2.2 Problem grupowania danych . . . . .	36
2.3 Problem wykrywania wartości odstających . . . . .	38
2.4 Problem dopasowania sekwencji biologicznych . . . . .	39
<b>3 Modyfikacje algorytmu genetycznego imitujące procesy epigenetyczne</b>	<b>42</b>
3.1 Modyfikacja imitująca dziedziczenie za pomocą prionu . . . . .	44
3.1.1 Molekularne podstawy procesu dziedziczenia za pomocą prionu . . . . .	44
3.1.2 Opis modyfikacji imitującej dziedziczenie za pomocą prionu . . . . .	45
3.2 Modyfikacja imitująca metylację cytozyny . . . . .	47
3.2.1 Molekularne podstawy procesu metylacji cytozyny . . . . .	47
3.2.2 Opis modyfikacji imitującej metylację cytozyny . . . . .	47

---

3.3	Modyfikacja imitująca wyłączenie alleliczne . . . . .	49
3.3.1	Molekularne podstawy procesu wyłączenia allelicznego . . . . .	49
3.3.2	Opis modyfikacji . . . . .	50
<b>4</b>	<b>Eksperymenty</b>	<b>52</b>
4.1	Warunki eksperymentów . . . . .	52
4.2	Metodyka przeprowadzania eksperymentów . . . . .	53
4.3	Badanie algorytmu optymalizacji załadunku . . . . .	55
4.3.1	Dane wykorzystane w eksperymentach . . . . .	55
4.3.2	Podstawowe parametry algorytmu . . . . .	55
4.3.3	Operacja <i>Epi_1</i> . . . . .	56
4.3.4	Modyfikacja <i>Epi_2</i> . . . . .	60
4.3.5	Operacja <i>Epi_3</i> . . . . .	63
4.3.6	Podsumowanie wyników . . . . .	67
4.4	Badanie algorytmu grupowania danych . . . . .	68
4.4.1	Dane użyte w eksperymentach . . . . .	69
4.4.2	Parametry algorytmu . . . . .	69
4.4.3	Ocena skuteczności działania algorytmu genetycznego do grupowania danych . . . . .	70
4.4.4	Operacja <i>Epi_1</i> . . . . .	71
4.4.5	Modyfikacja <i>Epi_2</i> . . . . .	75
4.4.6	Operacja <i>Epi_3</i> . . . . .	78
4.4.7	Podsumowanie wyników . . . . .	82
4.5	Badania algorytmu wykrywania wartości odstających . . . . .	83
4.5.1	Dane wykorzystane w eksperymentach . . . . .	84
4.5.2	Parametry algorytmu . . . . .	84
4.5.3	Operacja <i>Epi_1</i> . . . . .	85
4.5.4	Modyfikacja <i>Epi_2</i> . . . . .	88
4.5.5	Operacja <i>Epi_3</i> . . . . .	92
4.5.6	Podsumowanie wyników . . . . .	96
4.6	Algorytm dopasowania sekwencji biologicznych . . . . .	97
4.6.1	Dane wykorzystane w eksperymentach . . . . .	98
4.6.2	Parametry algorytmu . . . . .	98
4.6.3	Operacja <i>Epi_1</i> . . . . .	99
4.6.4	Modyfikacja <i>Epi_3</i> . . . . .	102

---

4.6.5	Operacja <i>Epi_3</i> . . . . .	105
4.6.6	Podsumowanie wyników . . . . .	108
4.7	Podsumowanie eksperymentów . . . . .	110
	<b>Podsumowanie i wnioski</b>	<b>113</b>
	<b>Bibliografia</b>	<b>117</b>
	<b>Bibliografia</b>	<b>117</b>
	<b>Spis tabel</b>	<b>133</b>
	<b>Spis rysunków</b>	<b>137</b>

---

# Wstęp

Człowiek od wielu lat stara się naśladować przyrodę i wykorzystywać rozwiązania podpatrzone w naturze. Inspiracja procesami zachodzącymi w naturze znalazła również swoje odzwierciedlenie w informatyce. Na podstawie obserwacji przyrody opracowano algorytmy, które w swoim działaniu wykorzystują operacje naśladujące procesy zachodzące w naturze.

Algorytmy inspirowane naturą znalazły szczególne zastosowanie w przypadku problemów optymalizacyjnych z którymi nie do końca radzą sobie klasyczne algorytmy deterministyczne. Problemy optymalizacyjne to grupa problemów dla których jesteśmy w stanie zdefiniować pewną funkcję, dla której iteracyjnie dobieramy odpowiednio wartości, tak aby uzyskać jej optymalną wartość.

Szczególny typ algorytmów inspirowanych naturą – algorytmy genetyczne są przedmiotem badań w niniejszej pracy. Należą one do grupy algorytmów ewolucyjnych. Idea algorytmów genetycznych bazuje na teorii ewolucji i procesach zachodzących w komórkach każdego żywego organizmu. W ostatnich latach nastąpił znaczący postęp nauki w zakresie biologii i genetyki molekularnej, odkrytych zostało wiele nowych mechanizmów występujących w procesie dziedziczenia i odpowiedzialnych za ewolucję organizmów. Mechanizmy te nie są ściśle związane z bezpośrednimi modyfikacjami sekwencji DNA i zostały nazwane procesami epigenetycznymi. Procesy epigenetyczne mają znaczący wpływ na dziedziczenie oraz odpowiadają za wpływ czynników zewnętrznych na funkcjonowanie organizmów żywych. Klasyczne algorytmy genetyczne zostały opracowane w latach 60. XX wieku, w związku z tym nie jest możliwe by były w nich uwzględnione wszystkie znane obecnie procesy dotyczące dziedziczenia i zmienności gatunków.

Właśnie te stosunkowo nowe odkrycia z zakresu genetyki molekularnej (jakimi są procesy epigenetyczne) zainspirowały autora niniejszej rozprawy do sprawdzenia możliwości ich zastosowania w algorytmach genetycznych. Punktem wyjścia dla wprowadzenia nowych operacji w algorytmie genetycznym (wzorowanych na procesach epigenetycznych) było to, że skoro mają duże znaczenie w naturze przy dostosowywaniu się organizmów żywych do warunków śro-

dowiskowych, to możliwe jest że ich odpowiednie odwzorowanie w algorytmach genetycznych umożliwi poprawię ich efektywności.

W niniejszej pracy zaprezentowane zostały trzy autorskie modyfikacje klasycznego algorytmu genetycznego, odwzorowujące mechanizmy zachodzące w procesach epigenetycznych:

- dziedziczenie za pomocą prionu,
- metylacja cytozyny,
- wyłączenie alleliczne.

## Teza rozprawy

*Zastosowanie mechanizmów wzorowanych na procesach epigenetyki pozwala na poprawę efektywności działania algorytmów genetycznych.*

## Cel rozprawy

Głównym celem rozprawy była opracowanie modyfikacji wzorowanych na procesach epigenetycznych, a następnie przetestowanie efektywności działania klasycznego algorytmu genetycznego wzbogaconego o te modyfikacje. Badania zaprezentowane w niniejszej rozprawie dotyczyły wpływu zaproponowanych modyfikacji na liczbę generacji algorytmu genetycznego, czyli na zmniejszeniu liczby cykli obliczeniowych związanych z operacjami tworzenia nowej populacji i pozostałych operacji występujących w algorytmach genetycznych. Kolejnym analizowanym parametrem był czas potrzebny na wykonanie wszystkich operacji w algorytmie genetycznym dla pojedynczej generacji w algorytmie.

Do osiągnięcia celu głównego autor sformułował następujące cele dodatkowe:

- analiza operacji wzorowanych na procesach epigenetycznych pod kątem ich zastosowania w algorytmach genetycznych,
- analiza istniejących modyfikacji algorytmów genetycznych,
- zaprojektowanie i zaimplementowanie zaproponowanych operacji bazujących na procesach epigenetycznych w algorytmach genetycznych,
- przeprowadzenie badań eksperymentalnych zaproponowanych operacji pod kątem wpływu na poprawę efektywności działania algorytmów genetycznych,



- zbadanie wpływu parametrów proponowanych operacji na efektywność działania algorytmów genetycznych,
- przeprowadzenie badań eksperymentalnych pod kątem oceny wpływu wprowadzenia zaproponowanych operacji na czas działania algorytmu genetycznego,
- porównanie działania algorytmów genetycznych z wprowadzonymi proponowanymi operacjami z algorytmami bazowymi (bez dodatkowych operacji epigenetycznych).

Część wyników badań zaprezentowanych w niniejszej rozprawie zostało już opublikowane w [40, 41].

## Układ rozprawy

Rozprawa składa się z pięciu rozdziałów. W Rozdziale pierwszym omówione zostały podstawowe informacje dotyczące algorytmów ewolucyjnych. Szczegółowo opisano jeden z rodzajów algorytmów ewolucyjnych jakim są algorytmy genetyczne, wraz z podstawami biologicznymi na których się inspirowano przy opracowywaniu algorytmów ewolucyjnych, a także spotykane w literaturze modyfikacje algorytmów genetycznych i ich przykładowe zastosowania. W Rozdziale pierwszym znalazły się również informacje o problemach optymalizacyjnych, jako grupie problemów, które najczęściej rozwiązuje się za pomocą algorytmów genetycznych.

Rozdział drugi zawiera omówienie wybranych problemów optymalizacyjnych, w odniesieniu do których określono algorytmy będące bazowymi do badań nad oceną efektywności proponowanych autorskich operacji w algorytmach genetycznych. Trzy algorytmy dla problemów optymalizacyjnych są algorytmami zaczerpniętymi z literatury, jeden jest autorską wersją algorytmu genetycznego umożliwiającego grupowanie danych.

W Rozdziale trzecim omówiono szczegółowo autorskie modyfikacje algorytmów genetycznych, inspirowane procesami epigenetycznymi. Na początku rozdziału zostało umieszczone krótkie wprowadzenie do zagadnienia epigenetyki oraz zaproponowanych modyfikacji. Dla każdej z zaproponowanych modyfikacji przedstawiono zarówno podstawy biologiczne, jak i opis sposobu implementacji.

W Rozdziale czwartym zaprezentowano wyniki eksperymentów dotyczących skuteczności zaproponowanych autorskich modyfikacji (zaprezentowanych w Rozdziale trzecim). Na początku rozdziału umieszczono szczegółowe informacje o warunkach w jakich przeprowadzone były eksperymenty. W kolejnych podrozdziałach przedstawiono wyniki badań otrzymane dla bazowych algorytmów genetycznych (omówionych w Rozdziale 2), z zastosowaniem poszczególnych

autorskich modyfikacji inspirowanych epigenetyką. Otrzymane wyniki dla algorytmów zmodyfikowanych o autorskie modyfikacje zostały porównane z wynikami otrzymanymi dla algorytmów bazowych (bez modyfikacji). Rozdział 5 zawiera zakończenie i wnioski z przeprowadzonych badań, oraz podsumowanie rozprawy i propozycje dalszych badań.

## Konwencja edytorska

W pracy zastosowano następującą konwencję edytorską:

- tekst zwykły pisany czcionką szeryfową o rozmiarze 12 pt,
- *tekst pisany kursywą* oznacza terminy(nazwy) w języku angielskim,
- akronimy nazw wprowadzone i używane w rozprawie wyróżnione są *czcionką pochyłą*,
- pojęcia pojawiające się po raz pierwszy w pracy wyróżniono podkreśleniem.

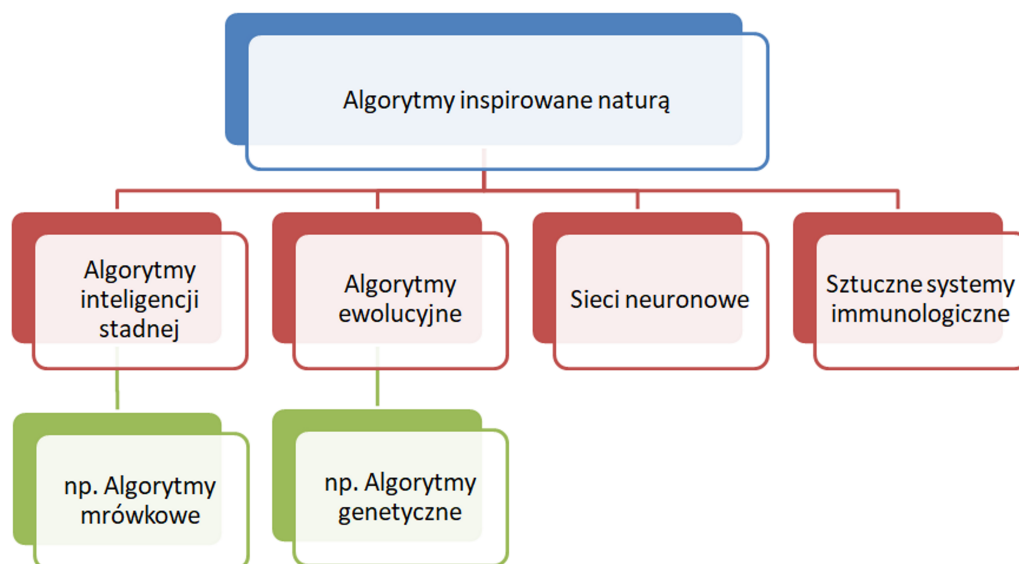
Rysunki i tabele zamieszczone w pracy, o ile nie podano źródła, są opracowaniem własnym.

# Algorytmy genetyczne w problemach optymalizacyjnych

W informatyce odrębną rodzinę algorytmów stanowią algorytmy inspirowane procesami zachodzącymi w naturze. Algorytmy inspirowane naturą naśladują w swym działaniu procesy zachodzące w naturze. W algorytmach tych odwzorowano zarówno zachowanie całych grup organizmów, jak na przykład w algorytmach inteligencji stadnej [172], jak i teorię ewolucji i mechanizmy dziedziczenia na poziomie DNA i RNA – są to algorytmy ewolucyjne. Inną grupą są algorytmy symulujące procesy zachodzące w mózgu organizmów żywych – te zwane są sztucznymi sieciami neuronowymi [162]. Jeszcze inna grupa algorytmów inspirowanych naturą bazuje na procesach zachodzących w systemie immunologicznym organizmów żywych [76]. Algorytmów inspirowanych naturą jest bardzo wiele i pochodzą z różnych aspektów działania świata ożywionego i służą do rozwiązywania wielu problemów.

Najpopularniejszą klasyfikację algorytmów inspirowanych naturą zaprezentowano na Rysunku 1.1. W związku z tym, że przedmiotem badań zaprezentowanych w niniejszej pracy były algorytmy genetyczne, w dalszej części pracy skupiono się jedynie na tej grupie algorytmów.

Algorytmy genetyczne [50, 69] należą do grupy algorytmów ewolucyjnych. Algorytmy ewolucyjne to takie algorytmy, które inspirowane są mechanizmami zachodzącymi wśród organizmów żywych. Naśladują głównie procesy występujące w trakcie ewolucji organizmów żywych. Algorytmy ewolucyjne znalazły szczególne zastosowanie w rozwiązywaniu problemów optymalizacyjnych, czyli takich w których poszukuje się najlepszego rozwiązania spośród możliwych rozwiązań. Za najlepsze rozwiązanie, uznaje się takie, które najlepiej rozwiązuje problem optymalizacyjny. Jakość rozwiązania ocenia się za pomocą odpowiednio skonstruowanej (w zależności od problemu) funkcji celu, omówionej dokładniej w podrozdziale 1.4.



*Rysunek 1.1: Przykładowa klasyfikacja algorytmów inspirowanych naturą*

## 1.1 Problemy optymalizacyjne

Problemy dla których opracowuje się algorytmy można podzielić według różnych kryteriów. Jednym z możliwych podziałów jest podział ze względu na złożoność problemu:

- **problemy łatwo rozwiązywalne** – mające rozsądne algorytmy deterministyczne do ich rozwiązywania (rozsądne, czyli o akceptowalnej złożoności obliczeniowej),
- **problemy trudno rozwiązywalne** – nie mające rozsądnych algorytmów deterministycznych,
- **problemy nierozstrzygalne** – nie posiadające algorytmów.

Jednym z typów problemów trudno rozwiązywalnych, są problemy optymalizacyjne, wybrane algorytmy do rozwiązywania problemów optymalizacyjnych zostały wykorzystane w eksperymentach, których wyniki przedstawione zostaną w Rozdziale 4. W rozwiązywaniu problemów optymalizacyjnych zwykle dobrze sprawdza się zastosowanie algorytmów inspirowanych naturą.

Dla problemów optymalizacyjnych w większości przypadków, nie ma dobrych algorytmów deterministycznych, służących ich rozwiązaniu. Poprzez pojęcie „dobrego algorytmu” rozumie się takie algorytmy, które posiadają akceptowalną złożoność obliczeniową. Z problemami optymalizacyjnymi ma się do czynienia bardzo często, stąd też niezwykle ważne jest opracowywanie odpowiednich algorytmów je rozwiązujących. Do przykładowych zadań optymalizacyjnych można zaliczyć:

- optymalizację produkcji,

- optymalizację załadunku,
- znajdowanie minimów i maksimów funkcji,
- wyznaczanie optymalnej trasy,
- wykrywanie nieprawidłowości w danych – na przykład wykrywanie danych odstających,
- dopasowanie danych – na przykład grupowanie,
- wszystkie pozostałe problemy, dla których jesteśmy w stanie określić funkcję celu (omówioną w dalszej części rozdziału).

Problemy optymalizacyjne to grupa problemów, dla których poszukuje się rozwiązania  $x_0$  należącego do grupy wszystkich rozwiązań  $X$ , dla którego wartość **funkcji celu**  $f(x)$  przyjmuje optymalną wartość [29, 42]. Formalną definicję problemu optymalizacyjnego przedstawiono w równaniu 1.1, rozwiązanie problemu optymalizacyjnego polega zaś na znalezieniu takiego rozwiązania  $x_0$  należącego do zbioru rozwiązań  $X$ , że w zależności od problemu:

$$\begin{aligned} f(x_0) &= \max(f(x)), \text{ dla wszystkich } x \text{ ze zbioru } X \\ f(x_0) &= \min(f(x)), \text{ dla wszystkich } x \text{ ze zbioru } X \end{aligned} \tag{1.1}$$

gdzie:

$X$  – dowolny zbiór skończony (przestrzeń rozwiązań),

$f : X \rightarrow \mathbb{R}$  – rzeczywista funkcja na  $X$  (funkcja celu).

Inaczej mówiąc, problemy optymalizacyjne to problemy, w których poszukuje się najlepszego możliwego rozwiązania, spośród grupy wszystkich możliwych rozwiązań.

Pierwszym etapem rozwiązywania problemu optymalizacyjnego jest skonstruowanie odpowiedniego modelu. Model ma na celu opisanie w sposób matematyczny celu, zmiennych i ograniczeń. W rozwiązywaniu problemów optymalizacyjnych dąży się do maksymalizacji, bądź minimalizacji ustalonej funkcji celu. Funkcja celu opisuje na ile dobre jest dane rozwiązanie problemu optymalizacyjnego. Zmienne rozumiemy jako elementy, których wartości chcemy dopasować w taki sposób, aby osiągnąć optymalną wartość funkcji celu. Drugim etapem opracowania rozwiązania problemu optymalizacyjnego, jest ustalenie, z jakiego typu problemem optymalizacyjnym ma się do czynienia.

Problemy optymalizacyjne można podzielić na kilka typów, w zależności od zastosowanych kryteriów [70, 158], co zostało zaprezentowane w Tabeli 1.1. W Tabeli 1.2 przedstawiono przykładowe problemy optymalizacyjne dla podziału zaprezentowanego w Tabeli 1.1.

**Tabela 1.1:** Podział problemów optymalizacyjnych według różnych kryteriów

optymalizacja	Typ danych	optymalizacja ciągła	Dane do modelu mają postać ciągłą.
		optymalizacja dyskretna	Dane do modelu mają postać dyskretną.
	Ograniczenia	optymalizacja nieograniczona	Brak dodatkowych ograniczeń wartości zmiennych.
		optymalizacja ograniczona	Pojawiają się dodatkowe ograniczenia w wartości zmiennych. Ograniczenia powodują, że nawet w przypadku prawidłowych wartości zmiennych, otrzymana kombinacja kilku wartości nie jest dopuszczalnym rozwiązaniem problemu.
	Liczba funkcji celów	brak celów	W przypadku braku funkcji celu ma się do czynienia z takim dopasowaniem wartości zmiennych, aby spełnione były ograniczenia modelu optymalizacyjnego, bez sprecyzowanej funkcji celu.
		jeden cel	Problem optymalizacyjny posiada jedną funkcję celu, dąży się do jej minimalizacji bądź maksymalizacji.
		wiele celów	W przypadku wielu funkcji celu, można równocześnie dążyć do minimalizacji lub maksymalizacji poszczególnych funkcji celu. W takim przypadku bardzo często dokonuje się podziału modelu tak, aby zredukować go do pojedynczych funkcji celu, a dopiero otrzymane wyniki odpowiednio waży się, aby uzyskać optymalny wynik całościowy.
	Znajomość danych	optymalizacja deterministyczna	W przypadku modelu deterministycznego, zakłada się, że wszystkie dane dla modelu są dokładnie znane.
		optymalizacja stochastyczna	W przypadku modeli stochastycznych mamy do czynienia z niepewnymi danymi. Przykładem takich danych mogą być dane prognozowe, dotyczące przyszłego stanu. Dane prognozowe są w takim przypadku jedynie swego rodzaju przypuszczeniem, a ich ostateczna wartość nie jest znana.

## 1.2 Biologiczne podstawy algorytmów ewolucyjnych

Idea powstania algorytmów ewolucyjnych pojawiła się przy obserwacji naturalnych procesów zachodzących w przyrodzie. Podstawą stała się teoria ewolucji organizmów ży-

**Tabela 1.2:** Przykłady problemów optymalizacyjnych w zależności od typu problemu optymalizacyjnego

Typ optymalizacji	Przykład
optymalizacja ciągła	znajdowanie minimum funkcji matematycznej
optymalizacja dyskretna	problem komiwojażera
optymalizacja nieograniczona	grupowanie danych
optymalizacja ograniczona	problem optymalnego załadunku
brak celów	przetwarzanie i rekonstrukcja obrazów
jeden cel	wykrywanie wartości odstających
wiele celów	prognozowanie produkcji
optymalizacja deterministyczna	dopasowanie sekwencji biologicznych
optymalizacja stochastyczna	prognozowanie kursów giełdowych

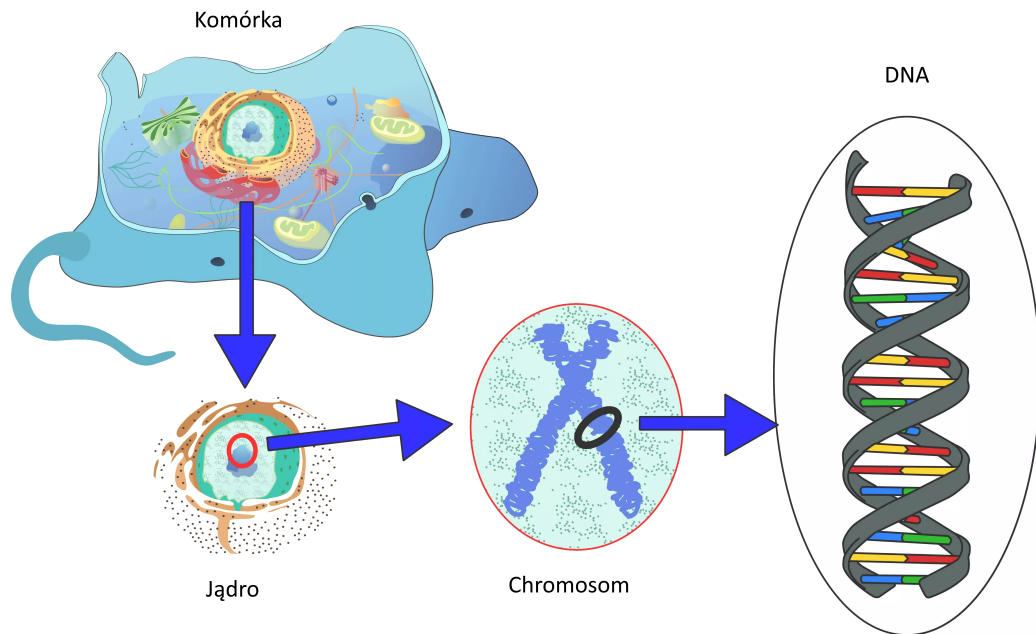
wych [10, 24, 108], sformułowana w roku 1838 przez Karola Darwina. Głównymi założeniami teorii ewolucji są:

- w wyniki ewolucji przetrwają tylko najlepiej przystosowane organizmy,
- dzięki zmianom zachodzącym w genotypie osobników, są one coraz lepiej przystosowane do życia,
- osobnikom najsłabiej przystosowanym do warunków środowiskowych grozi całkowite wyginięcie.

Korzystając z założeń tej teorii opracowano algorytmy ewolucyjne (w tym algorytmów genetycznych), bazujące na idei dziedziczenia i selekcji naturalnej występującej w przyrodzie. Selekcja naturalna to zdolność organizmu do przeżycia, dostosowania się do zmian środowiskowych i jego szanse na wydanie na świat potomstwa.

Podstawową jednostką każdego organizmu żywego jest komórka. W przyrodzie istnieją organizmy zbudowane tylko z jednej komórki (jak na przykład bakterie), oraz organizmy wielokomórkowe jak rośliny i zwierzęta. Organizmy wielokomórkowe składają się z wielu często wyspecjalizowanych w pełnieniu szczególnych funkcji komórek. Wszystkie funkcje komórek zależą od białek, a białka zbudowane są z łańcuchów aminokwasów [30, 124]. Struktura i funkcja danego białka zależy od kolejności aminokwasów w łańcuchu DNA i mechanizmów potranslacyjnej modyfikacji białek. Informacja, o tym czy dane białko powstanie, oraz jaką ma mieć budowę zapisane jest w sekwencji DNA, DNA które u większości organizmów znajduje się w jądrze komórkowym (w komórce). DNA zbudowane jest z podwójnej helisy, zawierającej dwie nici DNA połączone wiązaniami chemicznymi.

Pojedyncza nić DNA zbudowana jest z czterech (komplementarnych) aminokwasów (tyminy – T, adeniny – A, cytozyny – C i guaniny – G), cukrów i fosforanów. Komplementarność zasad oznacza, że w przypadku, gdy na jednej nici DNA występuje dany aminokwas to jest on



**Rysunek 1.2:** Budowa i miejsce występowania nici DNA [168]

połączony wiązaniem chemicznym z odpowiadającym mu aminokwasem na drugiej nici DNA (A–T, C–G). W jądrze komórkowym nić DNA upakowana jest w postaci chromosomów (ma to na celu kondensację objętości nici DNA, u człowieka na przykład długość nici DNA wynosi około 2 metrów). Każdy organizm posiada specyficzną liczbę i strukturę chromosomów. Nić DNA „upakowana” w formie chromosomu zawiera informacje genotypowe osobnika. Genotyp osobnika to z kolei zestaw genów. Aby dany gen mógł się uaktywnić, musi zostać odczytany z nici DNA i w procesie translacji zamieniony na odpowiednie białko. Geny determinują cechy fenotypowe osobnika. Gen jest uważany za podstawową jednostkę dziedziczenia u organizmów żywych. Cechy fenotypowe to ogół cech osobnika w skład których wchodzi wygląd osobnika, cechy osobowościowe (np. kolor oczu, sierści, wzrost, liczba kończyn).

Każdy żywy organizm posiada możliwość powielenie swojego kodu genetycznego. Powielenie kodu odbywa się podczas podziału komórkowego. W wyniku podziału komórkowego z jednej komórki powstają dwie komórki potomne, zawierające identyczny genotyp jak komórka rodzicielska. Genotyp osobnika może być również przekazywany do organizmów potomnych, z tym że organizm potomny otrzymuje po połowie genotypu od każdego rodzica. Pojedynczy gen zawiera pojedynczą informację fenotypową, w procesie translacji kod genetyczny tłumaczony jest na sekwencję aminokwasów (w konsekwencji na białko).

Przekazywanie genów organizmom potomnych odbywa się za pomocą mechanizmów dziedziczenia sformułowanych w prawach Mendla [24]. Na podstawie obserwacji stwierdzono, że niektóre geny, kodujące tę samą informację genetyczną, mają większe szanse na przekazanie i uaktywnienie się u osobnika potomnego (na przykład kolor oczu, mamy większe szanse na odziedziczenie



brązowego koloru oczu niż niebieskiego, tak zwane geny dominujące). Kolejną ważną cechą jest to, że geny odpowiadające za kodowanie różnych cech fenotypowych dziedziczą się niezależnie.

W procesie dziedziczenia bardzo ważną rolę spełniają również mutacje, które odpowiadają za różnicowanie gatunków. Mutacja polega na samoistnej zmianie fragmentu kodu genetycznego osobnika i może mieć pozytywny wpływ na danego osobnika (poprawić jego dostosowanie do środowiska w jakim żyje) lub negatywny zwiększając podatność na wpływy środowiska, powodując choroby genetyczne, a w skrajnych przypadkach powodując śmierć.

W ostatnich latach odkryto dodatkowe mechanizmy wpływające na proces dziedziczenia. Mechanizmy te zwane procesami epigenetycznymi nie są powiązane z kodem genetycznym organizmu. Procesy epigenetyczne nie podlegają prawom Mendla, mają jednak olbrzymie znaczenie przy różnicowaniu się osobników i dostosowaniu się do nowych warunków środowiskowych. Procesy epigenetyczne zostały odkryte już po stworzeniu algorytmów ewolucyjnych w związku z tym nie zostały odwzorowane w tychże algorytmach. Dla autora niniejszej pracy procesy epigenetyczne stały się inspiracją dla zaproponowanych modyfikacji algorytmów genetycznych.

### 1.3 Algorytmy ewolucyjne

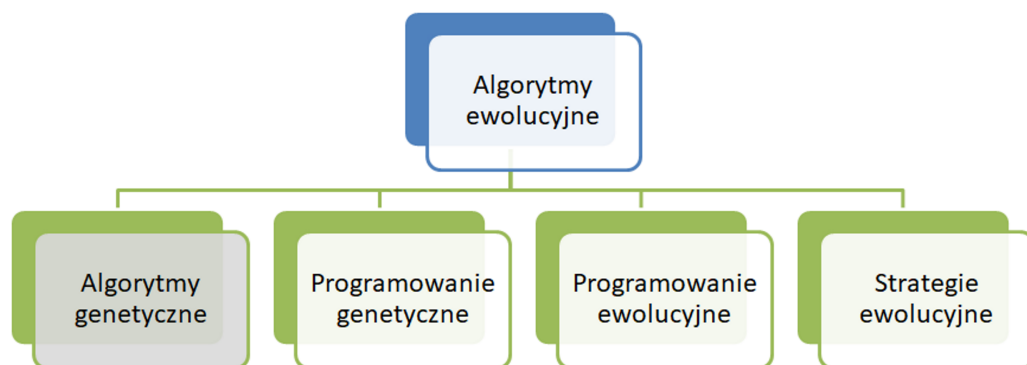
Jak już wspomniano algorytmy ewolucyjne [17, 18, 20, 117, 131, 151] to grupa algorytmów wzorowanych na procesie ewolucji występującej wśród organizmów żywych, są one grupą algorytmów umożliwiających poszukiwanie najlepszego rozwiązania problemu. Aby nie przeszukiwać pełnej przestrzeni rozwiązań, operuje się na podprzestrzeni możliwych rozwiązań wykonując na tych rozwiązaniach operacje wzorowane na mechanizmach występujących w ewolucji organizmów żywych (rozwiązanie problemu traktowane jest jako osobnik, a zbiór rozwiązań jako populacja). Wzorując się na procesach zachodzących w ewolucji, rozwiązanie lepsze, czyli takie które bliższe jest wartości oczekiwanej funkcji celu, ma większe szanse na przetrwanie, niż rozwiązania gorsze. W przypadku organizmów żywych lepiej przystosowany osobnik ma większe szanse na przeżycie i przekazanie genotypu kolejnemu pokoleniu, niż osobnik słabiej przystosowany (selekcja naturalna). Efektem takiego działania ma być optymalne rozwiązanie problemu, który rozwiązywany jest przy użyciu algorytmu ewolucyjnego.

Cechy charakterystyczne algorytmów ewolucyjnych to:

- dane do przetwarzania są w zakodowanej postaci, zależnej od problemu,
- poszukiwanie rozwiązania odbywa się na podstawie pewnej puli możliwych rozwiązań,

- wykorzystuje się pewną funkcję przystosowania, dążąc w zależności od problemu do jej maksymalizacji, bądź minimalizacji,
- stosuje się probabilistyczne metody wyboru,

Podstawowy podział algorytmów ewolucyjnych przedstawiono na Rysunku 1.3 [20] (kolorem szarym zaznaczono algorytmy genetyczne, które są przedmiotem badań, niniejszej pracy.



*Rysunek 1.3: Podział algorytmów ewolucyjnych*

Najczęściej spotykanymi rodzajami algorytmów ewolucyjnych są:

- Algoritmy genetyczne – grupa algorytmów, które pod względem swojego działania naśladują procesy zachodzące podczas ewolucji organizmów żywych. Służą głównie do rozwiązywania problemów optymalizacyjnych, omówione zostały dokładnie w kolejnych podrozdziałach.
- Programowanie genetyczne – algorytmy o zasadzie działania podobnej do algorytmów genetycznych, również wykorzystywane są operacje bazujące na teorii ewolucji. W wyniku programowania genetycznego uzyskuje się jednak rozwiązanie nie w postaci ciągu bitów (jak w przypadku algorytmów genetycznych), a program komputerowy. Dodatkowo wykorzystywane są struktury drzewa, a właśnie na ich węzłach i gałęziach przeprowadzone są operacje genetyczne.
- Programowanie ewolucyjne – są to algorytmy stosowane do rozwiązywania problemów optymalizacji ciągłej. W programowaniu ewolucyjnym jedynymi operacjami są operacje selekcji i mutacji; oznacza to, że nowy osobnik powstaje jedynie na drodze mutacji, które zachodzą u rodziców.

- Strategie ewolucyjne – wykorzystywane są głównie do rozwiązywania problemów optymalizacyjnych, gdzie ma się czynienia z danymi wielowymiarowymi w postaci liczb rzeczywistych. Kodowanie osobników odbywa się więc za pomocą liczb rzeczywistych.

Ponieważ celem pracy była modyfikacja algorytmów genetycznych, w kolejnym podrozdziale zostały one opisane w sposób bardziej szczegółowy.

## 1.4 Algorytmy genetyczne

W podrozdziale tym opisano szerzej ideę algorytmu genetycznego, a także omówiono zasadę działania klasycznego algorytmu genetycznego (KAG) wraz z operacjami jakie w nim są stosowane. W dalszej części rozdziału zaprezentowano literaturowy przegląd znanych modyfikacji algorytmu genetycznego (zarówno te w postaci nowych operacji zaimplementowanych w algorytmach genetycznych, jak i te polegające na modyfikacji istniejących operacji). W ostatniej części rozdziału zaprezentowano literaturowe przykłady zastosowania algorytmów genetycznych.

Zestawienie pojęć związanych z algorytmami genetycznymi wraz z ich wytłumaczeniem i odniesieniem nauk o życiu zawarto w Tabeli 1.3.

### 1.4.1 Klasyczny algorytm genetyczny

Algorytmy genetyczne znalazły zastosowanie w rozwiązywaniu problemów optymalizacyjnych w przypadku w którym nie do końca sprawdzały się podejścia deterministyczne [25, 63, 65, 116, 150, 174]. Idea algorytmu genetycznego, jako algorytmu inspirowanego teorią ewolucji, została zaprezentowana przez Johna Hollanda w latach 60. XX wieku, a następnie rozwinięta przez Davida E. Goldberga w latach 80. XX wieku.

W algorytmach genetycznych terminologia zaczerpnięta została z nauk biologicznych i zaadaptowana na potrzeby algorytmów genetycznych (patrz Tabela 1.3).

Pseudokod klasycznego algorytmu genetycznego (KAG) został przedstawiony jako Algorytm 1 [18].

Pierwszym niezbędnym krokiem w algorytmach genetycznych jest wybór odpowiedniego sposobu kodowania osobnika – ma to istotny wpływ na to, czy za pomocą algorytmu uzyska się oczekiwany wynik.

Kolejnym krokiem przy konstruowaniu zadania dla algorytmu genetycznego jest odpowiednie skonstruowanie funkcji oceny osobników (funkcja przystosowania). Funkcja przystoso-

**Tabela 1.3:** Pojęcia związane z algorytmami genetycznymi i odniesienie do pojęć stosowanych w naukach o życiu

		Natura	Algorytm genetyczny
określenia	osobnik	pojedynczy osobnik danego gatunku	pojedyncze rozwiązanie algorytmu genetycznego
	populacja	grupa osobników jednego gatunku, zamieszkująca dany obszar	podzbiór możliwych rozwiązań problemu rozwiązywanego za pomocą algorytmu genetycznego
	genotyp	informacja genetyczna zawarta w nici DNA	sposób kodowania pojedynczego osobnika
	fenotyp	ogół cech zewnętrznych osobnika	zbiór cech podlegających ocenie za pomocą funkcji przystosowania
	kariotyp	kompletny zestaw chromosomów	brak odpowiednika
	gen	pojedyncza informacja genetyczna	pojedyncza wartość w genotypie osobnika
	allel	jedna z wersji genu	wartości genów
	chromosom	forma organizacji materiału genetycznego	zapis rozwiązania problemu u pojedynczego osobnika
operacje	selekcja	dobór naturalny, zwiększenie szans na przeżycie osobnikom lepiej przystosowanym	ocena populacji pod względem wartości funkcji przystosowania
	krzyżowanie	proces powstania nowego osobnika na drodze wymiany komórek pomiędzy osobnikami	złączenie genotypów rodziców, w celu utworzenia nowych osobników
	mutacja	nagłe i skokowe zmiany w materiale genetycznym komórki	losowa zmiana wartości genu w genotypie

---

**Algorytm 1:** Klasyczny algorytm genetyczny

---

```

1 Dane wejściowe: populacja osobników;
2 Wynik: Osobnik z najlepszą wartością funkcji przystosowania;
3 begin
4   t = 0;
5   Utwórz populację początkową –  $Pop(0)$ ;
6   Dokonaj oceny osobników – oblicz wartość funkcji przystosowania dla każdego
   osobnika z populacji  $P_0$ ;
7   do
8     wybierz osobników do nowej populacji  $Pop(t)$  – selekcja;
9     wykonaj operację krzyżowania;
10    wykonaj operację mutacji osobników;
11    dokonaj oceny osobników;
12    zastąp starą populację nową;
13    t = t + 1;
14  while osiągnięto warunek stopu;
15 end

```

---

wania jest wartością, która pokazuje jak dobre jest dane rozwiązanie. Dla wszystkich osobników w danej populacji oblicza się wartość funkcji przystosowania, a jej wartość ma wpływ na prawdopodobieństwo udziału osobnika w reprodukcji – tworzeniu nowej populacji. Operacje występujące w KAG (przedstawione w Algorytmie 1) zostały omówione dokładniej w kolejnym

Podrozdziale 1.4.2.

## 1.4.2 Elementy składowe algorytmów genetycznych

W podrozdziale tym zostaną omówione dokładniej podstawowe kroki algorytmu genetycznego i występujące w nim operacje, przedstawione w Algorytmie 1 [4, 18, 116].

### Reprezentacja osobników

Jak już wspomniano pierwszym problemem jaki pojawia się przy rozwiązywaniu problemu za pomocą algorytmu genetycznego jest dobranie odpowiedniego sposobu kodowania osobników [84, 142], inaczej mówiąc sposobu przedstawienia osobnika, tak aby w dobry sposób reprezentował możliwe rozwiązanie. Odpowiednie kodowanie ma też bezpośredni wpływ na kolejne kroki algorytmu genetycznego. W kodowaniu osobników jako genotyp rozumie się zapis pełnego pojedynczego rozwiązania, a jako wartość genu pojedynczą wartość w genotypie. W przypadku KAG najczęściej stosowane sposoby kodowania osobników zostały zebrane w Tabeli 1.4.

*Tabela 1.4: Sposoby kodowania osobników w KAG*

typ kodowania	opis
kodowanie binarne	Każdy osobnik kodowany jest za pomocą ciągu zer i jedynek. Dodatkową zaletą takiego podejścia jest łatwość przeprowadzania na tak zakodowanych osobnikach operacji genetycznych. Dla części problemów kodowanie takie nie jest optymalne, gdyż nie odzwierciedla wszystkich możliwych kombinacji rozwiązań.
kodowanie za pomocą liczb rzeczywistych	Geny osobnika kodowane są za pomocą liczb rzeczywistych. Stosuje się je przede wszystkim gdy zachodzi potrzeba zakodowania osobników za pomocą wartości ciągłych.
kodowanie za pomocą liczb całkowitych	Stosowane jest w przypadku, gdy ze względu na możliwe wartości genu nie można zastosować kodowania binarnego (które jest ograniczone tylko do dwóch wartości).
kodowanie permutacyjne	Wykorzystywane jest najczęściej w przypadku gdy poszukiwana jest kolejność ustalenia elementów. Klasycznym problemem kodowanym za pomocą kodowania permutacyjnego jest problem komiwojażera.

Przedstawione sposoby kodowania są standardowymi sposobami kodowania osobników w KAG. Istnieją także dodatkowe sposoby kodowania osobników, które są dostosowane do problemu jaki ma być rozwiązany za pomocą algorytmu genetycznego. Przykładem takiego specyficznego kodowania jest kodowanie osobników w algorytmie do dopasowania sekwencji biolo-

gicznych (GAMSA), wykorzystanym w eksperymentach zaprezentowanych w pracy i omówiony w Rozdziale 2.

### **Funkcja celu i funkcja przystosowania**

Funkcja celu wyraża wielkość którą chcemy za pomocą algorytmu genetycznego minimalizować lub maksymalizować. W naturalnym środowisku funkcja celu oznacza podatność osobnika na choroby, szanse na osiągnięcie dojrzałości i wydanie na świat potomstwa. W przypadku algorytmów genetycznych odpowiada wprost za to czy dany ciąg kodowy (dany osobnik) pojawi się w kolejnej generacji w algorytmie genetycznym.

Równie ważnym problemem, w przypadku algorytmów genetycznych, jest odpowiednie skonstruowanie funkcji przystosowania [90].

Funkcja przystosowania to miara jakości danego osobnika w algorytmie genetycznym. Na podstawie funkcji przystosowania danego osobnika w populacji ocenia się na ile dobre jest rozwiązanie, które reprezentuje dany osobnik. W algorytmach genetycznych chodzi o to, żeby znaleźć osobnika z najlepszą wartością funkcji przystosowania. Funkcję przystosowania konstruuje się zawsze w oparciu o problem, jaki chce się rozwiązać za pomocą algorytmu genetycznego. Odpowiednie skonstruowanie funkcji przystosowania wpływa bezpośrednio na to czy za pomocą danego algorytmu genetycznego uzyskany zostanie najlepszy wynik. Dzięki modyfikacjom funkcji celu (min, max) dążymy do osiągnięcia wartości określonej przez funkcję przystosowania jako optymalną – najlepsze rozwiązanie = najlepsza funkcja przystosowania. Przykładem takiego celu może być minimalizacja kosztów produkcji, lub maksymalizacja zysków w firmie. Funkcja celu powinna posiadać takie cechy jak:

- powinna być jasno zdefiniowana,
- powinna być zaimplementowana w sposób optymalny, aby nie wpływała znacząco na złożoność obliczeniową algorytmu,
- powinna umożliwiać ilościowy pomiar na ile dobre jest dane rozwiązanie problemu,
- powinna generować intuicyjne wyniki: najlepsze rozwiązania powinny mieć minimalne/maksymalne wartości funkcji.

W praktyce najczęściej funkcją przystosowania jest w niewielkim stopniu zmodyfikowana funkcja celu. Wartość funkcji przystosowania osobnika wpływa bezpośrednio na jego szansę wzięcia udziału w operacji krzyżowania.

## Operacja selekcji osobników

Proces selekcji osobników [64, 82, 140, 152, 173] pozwala na wybór osobników o najlepszej wartości funkcji przystosowania: wezmą oni udział w procesie tworzenia nowej populacji osobników. Celem selekcji osobników jest ograniczenie możliwości tworzenia nowej populacji przez osobniki najslabiej przystosowane (z najgorszymi wartościami funkcji przystosowania): za pomocą selekcji osobników dąży się do tego, by każda nowo utworzona populacja składała się z osobników lepiej przystosowanych (o lepszej wartości funkcji przystosowania) [27]. Spośród spotykanych metod selekcji osobników najczęściej wykorzystuje się następujące:

- **selekcja proporcjonalna** – zwana również selekcją metodą ruletki [57, 100] polega na ustaleniu prawdopodobieństwa wylosowania każdego z osobników z populacji, gdzie wartość prawdopodobieństwa wylosowania zależy od wartości funkcji przystosowania. Osobnik lepiej przystosowany (z lepszą wartością funkcji przystosowania) będzie miał większe prawdopodobieństwo wylosowania, niż osobnik gorzej przystosowany (wzór na wyznaczenie prawdopodobieństwa wylosowania przedstawiony jest równaniem 1.2).

$$P(i) = \frac{F(i)}{\sum_{i=1}^k F(i)} \quad (1.2)$$

gdzie:

$P(i)$  – prawdopodobieństwo selekcji  $i$ -tego osobnika,

$F(i)$  – wartość funkcji przystosowania  $i$ -tego osobnika,

$k$  – liczba osobników w populacji.

Można to sobie wyobrazić jako koło ruletki: każdy osobnik „zajmuje” pewną część, ale wielkość zajętego wycinku koła zależy od wartości funkcji przystosowania – im lepiej przystosowany osobnik, tym większy wycinek koła zajmuje.

W następnym kroku następuje "zakręcenie" kołem i osobnik na którego wskaże koło bierze udział w operacji krzyżowania. Każdemu osobnikowi, na podstawie obliczonej dla niego wartości prawdopodobieństwa selekcji (równanie 1.2), oblicza się wycinek koła jaki zajmuje w kole ruletki, na podstawie wzoru  $v(i) = P(i) \cdot 100$ . Wadą metody jest możliwość wystąpienia sytuacji, gdy prawie całą powierzchnię koła ruletki zajmuje jeden osobnik.

- **selekcja rankingowa** – w metodzie tej wszystkie osobniki z populacji sortowane są malejąco według wartości funkcji przystosowania (od najlepiej przystosowanego osobnika do najgorzej) [179]. Do operacji krzyżowania "przechodzi" określona liczba osobników z najlepszymi wartościami funkcji przystosowania. Dodatkowo, miejsce w rankingu wskazuje

rangę danego osobnika co przekłada się bezpośrednio na liczbę jego wystąpień w procesie krzyżowania. Ta metoda pozwala na uniknięcie niepożądanego zajęcia prawie całego koła przez jednego osobnika występującego w selekcji proporcjonalnej.

- **selekcja turniejowa** – przeprowadzana jest dwuetapowo: w pierwszym etapie dokonuje się losowego wyboru grupy osobników z populacji (liczebność takiej grupy, jest najczęściej jednym z parametrów algorytmu genetycznego [118]), a w drugim etapie przeprowadza się "turniej" pomiędzy wybranymi osobnikami. Turniej polega na porównywaniu wartości funkcji przystosowania osobników. Zwycięzcą turnieju zostaje osobnik o najlepszej wartości funkcji przystosowania i ten osobnik ma prawo reprodukcji i pojawia się w nowej populacji.

Poza wymienionymi powyżej najczęściej spotykanymi metodami selekcji osobników, w literaturze można spotkać następujące metody selekcji:

- selekcja losowa – losowy wybór osobników,
- selekcja stabilnego stanu *Steady-State Selection* [81],
- selekcja elitarna (ang. *Elitism Selection*) [46],
- selekcja koła ruletki wielowskaźnikowej (SUS, ang. *Stochastic Universal Sampling*) [128],
- selekcja Boltzmanna [97].

## Operacja krzyżowania osobników

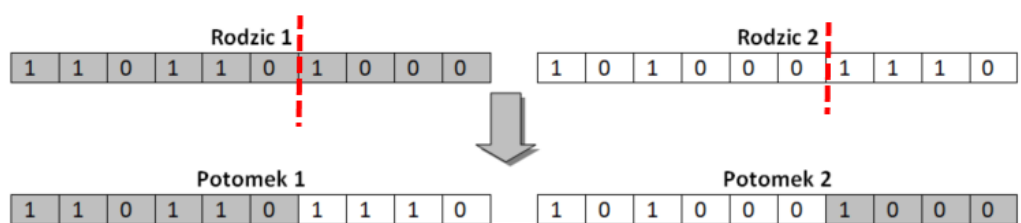
Krzyżowanie osobników [2, 49, 87, 105, 157] ma na celu wymianę genotypu pomiędzy wybranymi osobnikami. Efektem procesu krzyżowania osobników jest nowa populacja osobników o nowym genotypie, która w późniejszym kroku zastępuje poprzednią populację osobników. Operacja krzyżowania odwzorowuje proces rozmnażania u organizmów żywych, gdzie rodzice przekazują co najmniej część swojego genotypu swoim dzieciom. Operacja krzyżowania zachodzi z pewnym prawdopodobieństwem, dlatego część osobników z populacji rodziców przechodzi do nowego pokolenia, bez udziału w operacji krzyżowania.

To, które z osobników podlegają operacji krzyżowania determinowane jest w czasie operacji selekcji, omówionej w podrozdziale 1.4.2.

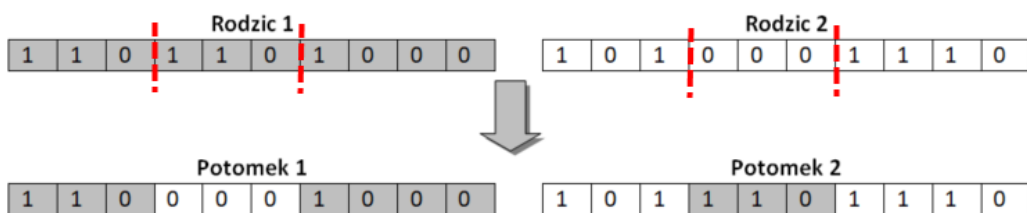
Najczęściej spotykanymi metodami w operacji krzyżowania są [123, 156, 167]:



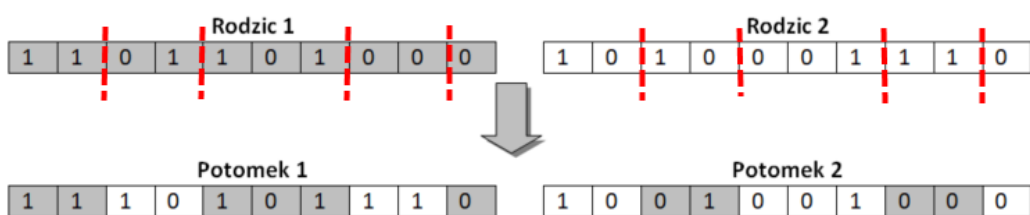
- **krzyżowanie jednopunktowe** (proste, ang. *one-point crossover*). Jest to metoda krzyżowania polegająca na wylosowaniu miejsca w którym wystąpi podział genotypu osobników biorących udział w krzyżowaniu. „Rodzic” w wylosowanym miejscu zostają „przecięc”, a ich potomkowie powstają w taki sposób, że pierwszy potomek otrzymuje fragment genotypu pierwszego rodzica od początku do wylosowanego miejsca podziału, a resztę genotypu od drugiego rodzica. Drugi potomek otrzymuje początek genotypu do wylosowanego miejsca podziału od drugiego rodzica, a resztę genotypu od pierwszego rodzica (por. Rysunek 1.4a);
- **krzyżowanie dwupunktowe** (ang. *two-point crossover*). Metoda ta jest modyfikacją krzyżowania jednopunktowego. W przypadku tej metody losowane są dwa miejsca podziału "genotypu". Genotyp pierwszego potomka składa się z fragmentu genotypu rodzica pierwszego od początku do pierwszego wylosowanego miejsca podziału, oraz od drugiego miejsca podziału do końca genotypu, reszta genotypu pochodzi od rodzica drugiego. Genotyp drugiego potomka tworzy się w sposób analogiczny, z tym, że fragmenty genotypu od początku do pierwszego miejsca podziału i od drugiego miejsca podziału do końca genotypu pochodzą od drugiego rodzica, a fragment genotypu pomiędzy wylosowanymi miejscami podziału od rodzica pierwszego (por. Rysunek 1.4b);
- **krzyżowanie  $k$ -punktowe** (ang. *k-point crossover*). Metoda ta jest modyfikacją krzyżowania dwupunktowego, z tym, że zamiast losowania dwóch punktów podziału genotypu osobnika, jest losowana większa ich liczba,  $k > 2$ ,  $k \in \mathbb{N}$ ;
- **krzyżowanie równomierne** (ang. *uniform crossover*). W metodzie tej losowany jest pewien wzorzec wskazujący, które geny są dziedziczone od którego rodzica i na tej podstawie następuje tworzenie nowego osobnika potomnego (Rysunek 1.4c). Wzorcem jest najczęściej łańcuch binarny w którym wartości 1 wskazują na gen rodzica pierwszego, a wartości 0 na gen rodzica drugiego;
- **krzyżowanie arytmetyczne** (ang. *arithmetic crossover*). W metodzie tej, w przypadku kodowania osobników za pomocą liczb rzeczywistych następuje uśrednienie wartości genów pochodzących od rodziców. W przypadku kodowania bitowego metoda ta polega na wykonaniu operacji logicznych na rodzicach w celu utworzenia nowego potomka (rysunek 1.4d).



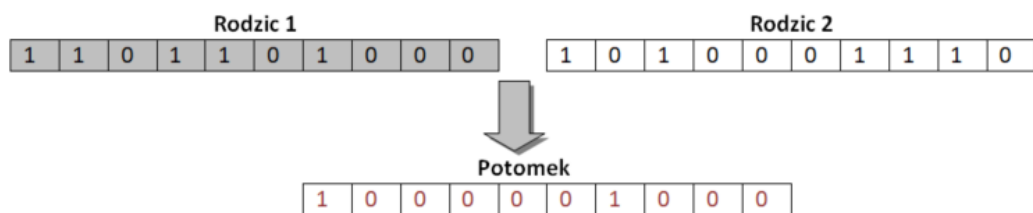
(a) Krzyżowanie 1 punktowe



(b) Krzyżowanie 2 punktowe



(c) Krzyżowanie równomierne



(d) Krzyżowanie arytmetyczne (iloczyn logiczny)

**Rysunek 1.4:** Schemat najczęstszych operacji krzyżowania występujących w algorytmach genetycznych

Wymienione metody krzyżowania są najczęściej spotykanymi metodami w literaturze. Użycie innych metod w dużej mierze jest uzależnione od sposobu kodowania osobników w algorytmie genetycznym, mogą to być przykładowo:

- krzyżowanie heurystyczne (ang. *heuristic crossover*) [5, 185],
- krzyżowanie średnicą (ang. *circle-ring crossover*) [32, 88],
- krzyżowanie rozmyte (ang. *fuzzy crossover*) [170],
- krzyżowanie tasujące (ang. *shuffle crossover*) [33],
- krzyżowanie zastępujące (ang. *reduced surrogate crossover*) [130],
- krzyżowanie mieszające (ang. *blend crossover*) [3, 74],
- krzyżowanie homologiczne (ang. *homologous crossover*) [58, 96],

- krzyżowanie diagonalne (ang. *diagonal crossover*) [51],

oraz wiele innych [69].

Czytelnika zainteresowanego metodami przeprowadzania operacji krzyżowania poleca się lekturę [69].

## Operacja mutacji

Operacja mutacji [13, 156] odwzorowuje proces mutacji genotypu zachodzący u organizmów żywych. Operacja mutacji polega na losowej zmianie niewielkiego fragmentu genotypu osobnika. Operacja ta (tak jak w naturze) występuje z niewielkim prawdopodobieństwem. Ogólnym celem operacji mutacji jest zwiększenie różnorodności osobników. Operacja mutacji powoduje zmianę w genotypie osobnika – nie pochodzącą od rodziców i może przynieść różne niespodziewane rezultaty: jej wynikiem może być zarówno poprawienie wartości funkcji przystosowania osobnika, jak i jej pogorszenie. Najczęściej spotykanymi typami mutacji występującymi w algorytmach genetycznych [67, 121] są:

- **mutacja punktowa** (ang. *non-uniform mutation*) – polega na losowej zamianie pojedynczego genu w genotypie osobnika. W przypadku kodowania bitowego osobników następuje zamiana 0 na 1, lub 1 na 0,
- **mutacja w wyniku przesunięcia** (ang. *bit string mutation*) – miejsce mutacji jest losowane, następnie usuwany jest gen na wylosowanej pozycji, a wszystkie następnie geny przesuwa się o jedno miejsce w lewo, koniec genotypu osobnika uzupełnia się wartością 0,
- **mutacja graniczna** (ang. *boundary mutation*) – wartość genu ulegającego mutacji zamieniana jest na wartość brzegową spośród możliwych wartości genu. Mutacja ta jest szczególnie przydatna, w przypadku gdy spodziewa się że rozwiązanie jest bliskie wartościom brzegowym,
- **mutacja jednolita** (ang. *uniform mutation*) – podobna do mutacji punktowej, jednak losowanie nowej wartości genu odbywa się ze zbioru o rozkładzie normalnym o zadanym zakresie.
- **mutacja gaussowska** (ang. *gaussian mutation*) – w mutacji gaussowskiej, wartość genu poddanego procesowi mutacji jest zamieniana na wartość losową z rozkładu Gaussa,
- **mutacja zamieniająca** (ang. *swap mutation*) – polega na wylosowaniu dwóch genów w genotypie osobnika, a następnie dokonaniu zamiany miejsc tych genów,

- **mutacja odwracająca** (ang. *inversion mutation*) – polega na wylosowaniu dwóch miejsc w genotypie osobnika, a następnie pomiędzy wylosowanymi punktami dokonuje się odwrócenia kolejności genów w genotypie osobnika.

Dobór odpowiedniego typu mutacji zależy od problemu jaki jest do rozwiązania za pomocą algorytmu genetycznego, oraz od sposobu kodowania osobników w populacji.

### Warunki stopu algorytmu genetycznego

Niezwykle ważnym problemem w konstruowaniu algorytmów genetycznych jest też ustalenie odpowiedniego warunku stopu algorytmu [18, 102, 144, 160], czyli warunku, w jakim algorytm zakończy swoje działanie zwracając najlepsze rozwiązanie. Od warunku stopu zależy, czy algorytm nie zakończy działania zbyt wcześnie, nie osiągając optymalnego rezultatu, czy też będzie wykonywał wiele niepotrzebnych powtórzeń operacji, które nie będą już poprawiały otrzymanego przez algorytm rozwiązania.

Możliwymi warunkami stopu dla algorytmów genetycznych mogą być:

- **ustalona z góry liczba generacji**, lub określona liczba generacji, w których nie uległa poprawie wartość funkcji przystosowania najlepszego osobnika, jest to najprostszy warunek stopu dla algorytmu genetycznego.
- **maksymalny czas** jaki został przeznaczony na poszukiwanie najlepszego rozwiązania przez algorytm genetyczny. W tym przypadku jako warunek stopu algorytmu genetycznego przyjmuje się maksymalny czas działania algorytmu, a bardzo często dodatkowo także wartość funkcji przystosowania której osiągnięcie powoduje zakończenie działania algorytmu (algorytm może zakończyć działanie wcześniej po osiągnięciu zadanej wartości funkcji przystosowania).
- **warunek różnorodności**, rozumiany jako odchylenie standardowe wartości funkcji przystosowania w populacji, zatrzymanie działania algorytmu ma miejsce w momencie, gdy zostanie utracona różnorodność w populacji. W przypadku utraty różnorodności w populacji pozostają jedynie osobniki o identycznym (lub niemal identycznym) genotypie. Rozwiązanie to jest jednak rzadko stosowane, częściej w przypadku utraty różnorodności w populacji stosuje się tzw. dostrajanie lokalne (omówione w podrozdziale 1.4.2) [83, 146].
- **specyficzne warunki stopu** (dobre do problemu, jaki ma być rozwiązany za pomocą algorytmu genetycznego). Na przykład w [22] autorzy proponują jako warunek stopu

osiągnięcie pewnego poziomu ufności, w artykule dla odmiany [14] autorzy proponują kryterium stopu oparte na wariancji w populacji najlepszych osobników.

### Dostrajanie algorytmu genetycznego

Niezwykle ważnym etapem, nie będącym bezpośrednio operacją w samym algorytmie genetycznym, ale czynnością, którą należy przeprowadzić, jest wstępne określenie optymalnych wartości jego parametrów. Jest to tzw. dostrajanie algorytmu genetycznego [52, 53, 54]. Dobór parametrów ma istotny wpływ na jakość rozwiązania otrzymanego za pomocą algorytmu genetycznego.

Podstawowe parametry algorytmu genetycznego przedstawiono w Tabeli 1.5.

**Tabela 1.5:** Podstawowe parametry algorytmu genetycznego

Parametr	Opis
długość osobnika	liczba genów w osobniku, determinowana przez dane wejściowe dla algorytmu
rozmiar populacji	liczba osobników w populacji. Zbyt mały utrudni znalezienie rozwiązania, zbyt duży może zamienić algorytm genetyczny w algorytm przeszukiwania wszystkich możliwych rozwiązań.
prawdopodobieństwo krzyżowania	określa prawdopodobieństwo wzięcia wystąpienia operacji krzyżowania
prawdopodobieństwo mutacji	określa prawdopodobieństwo wystąpienia operacji mutacji

Poza parametrami przedstawionymi w Tabeli 1.5, dla algorytmu genetycznego należy jeszcze określić sposób kodowania osobników, a także wybrać sposób przeprowadzenia operacji krzyżowania, selekcji, oraz mutacji. W przypadku wystąpienia w algorytmie genetycznym dodatkowych operacji, należy również określić prawdopodobieństwo ich wystąpienia.

Dobranie odpowiednich wartości parametrów algorytmu genetycznego ma bezpośredni wpływ na otrzymany wynik. Źle dostrojone parametry algorytmu genetycznego mają bezpośredni wpływ na obniżenie jakości wyniku działania algorytmu genetycznego (bądź też na wydłużenie czasu potrzebnego do uzyskania wyniku). Jednym z podstawowych parametrów jaki należy ustalić jest rozmiar populacji: wpływa on na jakość otrzymanego rozwiązania oraz na czas działania algorytmu.

Niewłaściwy dobór parametrów algorytmu genetycznego może również doprowadzić do przedwczesnej zbieżności, co spowoduje, że otrzymywane najlepsze wyniki stają się zbieżne do

pewnej granicy, będącej niestety ekstremum lokalnym. Aby temu zapobiec stosuje się czasami operację rekombinacji genotypu części osobników.

Ważnymi parametrami które również należy dobrać dla algorytmu genetycznego są wartości prawdopodobieństw wystąpienia poszczególnych operacji genetycznych. Przykładowo: zbyt duża wartość wystąpienia operacji mutacji może zmienić algorytm genetyczny w algorytm losowy.

Najczęściej spotykanym sposobem strojenia parametrów algorytmu genetycznego jest strojenie eksperymentalne, które polega na sprawdzeniu możliwych kombinacji parametrów i wybranie takich, dla których uzyskuje się najlepsze efekty. Takie podejście zostało również zastosowane w niniejszej pracy (por. Rozdział 4).

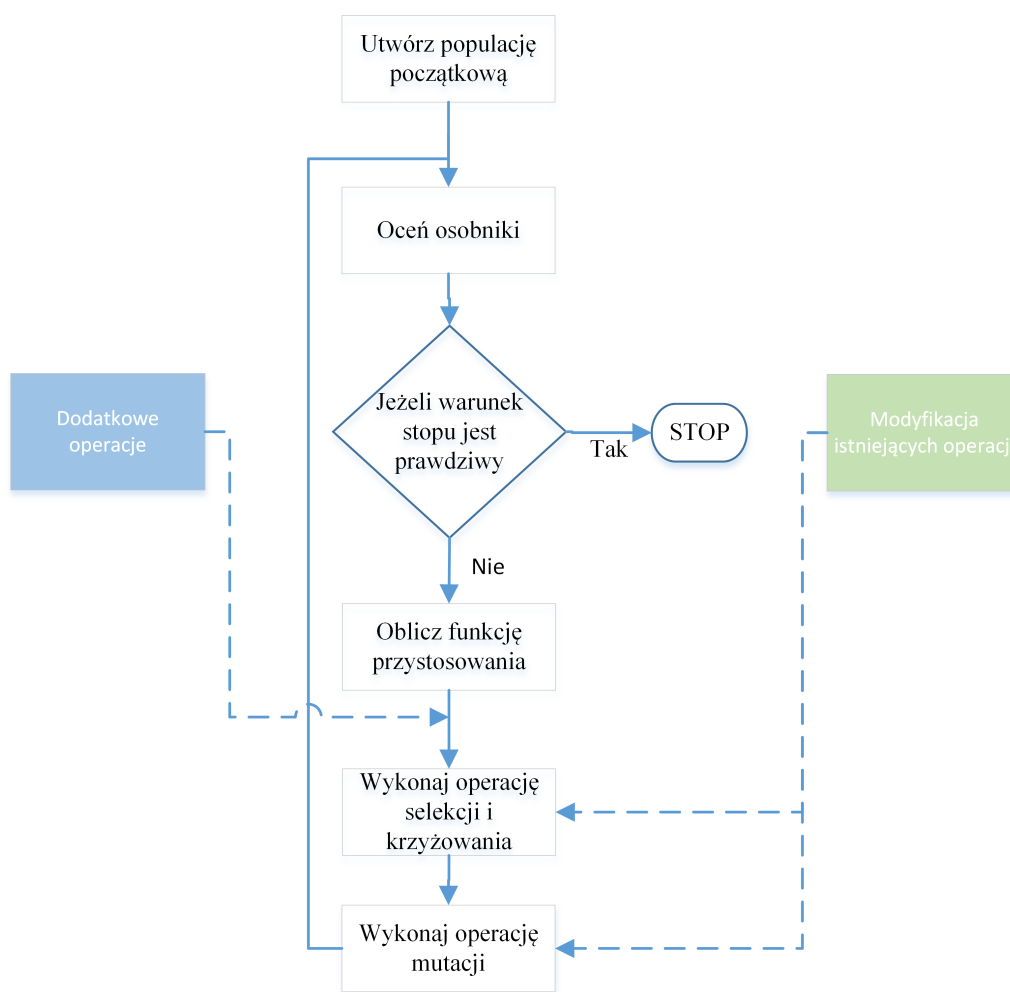
W przypadku strojenia algorytmu genetycznego często wykorzystuje się też wartości zaczerpnięte z literatury. Istnieją też, spotykane w literaturze, liczne próby automatyzacji procesu strojenia algorytmów genetycznych [1, 16, 37, 181].

### 1.4.3 Wybrane modyfikacje klasycznego algorytmu genetycznego

W podrozdziale tym zaprezentowany został przegląd literaturowy dostępnych modyfikacji algorytmów genetycznych. Modyfikacje występujące w algorytmach genetycznych można podzielić na cztery rodzaje. Pierwszy rodzaj dotyczy modyfikacji stanowiących nowe operacje w algorytmach genetycznych, drugi rodzaj to modyfikacje istniejących operacji. Trzecim rodzajem modyfikacji KAGsą modyfikacje parametrów, jako czwarty rodzaj możemy sklasyfikować pozostałe modyfikacje w tym mające na celu przyspieszenie działania KAG. Przykładowe miejsca włączenia modyfikacji do KAG pokazano na Rysunku 1.5.

Przykładowymi modyfikacjami z grupy pierwszej są:

- Operacja migracji zaproponowana w pracach [62, 122]. Operacja migracji polega na podziale populacji na podpopulacje, które przez pewną liczbę generacji pozostają w izolacji. W określonym momencie następuje wymiana osobników z pewnym prawdopodobieństwem  $P_m$  osobników między podpopulacjami. Jest to modyfikacja bazująca na modelu wyspowym i pojęciu niszowania [163].
- Operacja MGA, zaproponowana została w pracy [141]. Jej autorzy zaproponowali nową operację, która jest dodatkową selekcją najlepiej przystosowanych osobników z populacji po zastosowaniu operacji krzyżowania i mutacji. Na utworzonej w ten sposób podpopulacji przeprowadza się operację algorytmu genetycznego (krzyżowanie i mutację). W kolejnym



**Rysunek 1.5:** Rodzaje modyfikacji KAG

kroku porównuje się nowo otrzymanych osobników z ich rodzicami. Jeżeli otrzymane wartości funkcji przystosowania potomka są lepsze od wartości funkcji przystosowania rodzica to potomek zastępuje rodzica w danej generacji.

- Operacja tworzenia populacji początkowej, zaproponowana w artykule [132]. Tutaj autorzy proponują zastąpienie standardowego losowego tworzenia populacji początkowej, trzema sposobami tworzenia populacji początkowej (określonymi jako LHS, HSS, FSS), które mają lepiej pokryć przestrzeń możliwych rozwiązań dla algorytmu genetycznego.

Przykładowe modyfikacje z grupy drugiej to:

- Modyfikacja operacji krzyżowania, przedstawiona w pracy [137]: w zmodyfikowanej operacji krzyżowania oprócz osobników z danej populacji biorą też udział osobnicy z najlepszą wartością funkcji przystosowania z dwóch poprzednich generacji algorytmu genetycznego.
- Modyfikacja operacji krzyżowania polegająca na wyszukiwaniu dodatkowo na etapie krzyżowania u obu rodziców genu wpływającego w największym stopniu na pogorszenie war-

tości funkcji przystosowania i przeprowadzenie operacji krzyżowania w taki sposób aby dokonać zmiany wartości tego genu u potomków [6] [154].

- Modyfikacja operacji mutacji, polegająca na wprowadzeniu dodatkowego mechanizmu kontroli poziomu mutacji i jej wpływu na funkcję przystosowania osobnika [26].
- Modyfikacja operacji mutacji polegająca na przeprowadzeniu mutacji tylko dla genów najgorszych (w największym stopniu wpływające na pogorszenie wartości funkcji przystosowania) [73]. Dodatkowo w tym podejściu wprowadzony został mechanizm poszukiwania najgorszych genów.

Literaturowym przykładem modyfikacji z grupy trzeciej, jest: Zmiana reprezentacji chromosomów na etapie krzyżowania z binarnej (1,0) na zapis (+,-). Operacja krzyżowania polega na operacji logicznej AND (dla dwóch '+' do nowego osobnika przechodzi '+', w przeciwnym wypadku '-') (zaproponowana w [80]).

Innymi przykładami modyfikacji stosowanych w KAGsą (czwarty rodzaj):

- Zrównoleglenie i utworzenie modelu wyspowego algorytmu genetycznego, które zostało zaproponowane w algorytmach genetycznych już dość dawno ([177], 1964). W ramach modelu wyspowego istnieje kilka podpopulacji, które koegzystują niezależnie od siebie w algorytmie genetycznym (wykonywane są na nich osobno operacje algorytmu genetycznego). W modelu wyspowym pojawia się też dodatkowy element jakim jest połączenie pomiędzy subpopulacjami (migracja osobników między subpopulacjami);
- W pracy [110] zaproponowano Przyrostowy Algorytm Genetyczny (ang. *Incremental Genetic Algorithm*, IGA), który bazuje na KAG. W modyfikacji autorzy proponują podział działania algorytmu na 2 etapy: w pierwszym wykorzystywany jest KAG, za pomocą którego tworzona jest populacja najlepszych osobników, które biorą udział w drugim etapie (nazwanym IGA), w którym dodatkowo zaimplementowana została metoda modyfikująca problem, tak aby lepiej dopasować go do najlepszego rozwiązania;
- W pracy [15] autorzy proponują modyfikację polegającą na powtórzeniu operacji selekcji, krzyżowania i mutacji na etapie krzyżowania osobników. Jeżeli powstałe potomstwo (poddane dodatkowo operacji selekcji, krzyżowania i mutacji) jest lepiej przystosowane (ma lepszą wartość funkcji przystosowania) to zastępuje rodziców w populacji. Jeżeli nie, w populacji pozostają rodzice (eliminowany jest potomek);



- Modyfikacja polegająca na dodaniu tablicy przechowującej pewną grupę najlepszych osobników z kolejnych generacji została zaproponowana przez w [125]. Modyfikacja ta ma celu poprawę efektywności działania algorytmu genetycznego, autorzy tego artykułu deklarują poprawę względem algorytmu bez modyfikacji na poziomie 10%. Tak zmodyfikowany algorytm nazwano MEGA (ang. *Matrix Extension Genetic Algorithm*).

Przedstawione w niniejszym podrozdziale modyfikacje stanowią jedynie część wszystkich modyfikacji algorytmów genetycznych spotykanych w literaturze.

Niektóre z proponowanych modyfikacji mogą być zastosowane tylko w przypadku konkretnych algorytmów, i do rozwiązywania konkretnych problemów. Przykładem takich modyfikacji są zaproponowane modyfikacje algorytmu dla problemu komiwojażera przedstawione w artykule [139].

#### 1.4.4 Przykładowe zastosowania algorytmów genetycznych

Algorytmy genetyczne dobrze sprawdzają się w:

- rozwiązywaniu problemów optymalizacyjnych takich jak na przykład optymalizacja procesów produkcyjnych [19, 36, 91, 164, 171];
- analizie danych, mogą być pomocne w grupowaniu danych. Przykłady takich zastosowań zostały przedstawione w artykułach [78, 85, 112, 182]. Opisano również zastosowanie AG w wykrywaniu danych odstających [39, 136];
- optymalizacji tras przejazdu [38, 61, 94, 120];
- optymalizacji transportu, przy czym problem ten dotyczy zarówno optymalizacji transportu publicznego (ze względu na koszty, obciążenie tras itp.) [44, 71, 145, 176], jak i transportu towarów z uwzględnieniem opłacalności transportu, optymalizacji załadunku [45, 77, 98, 155];
- przetwarzaniu obrazów [79, 86, 161];
- optymalizacji funkcji [107, 180, 186];
- dziedzinach związanych z sieciami komputerowymi, zarówno w wyznaczaniu najkrótszej drogi dla pakietów [148], w optymalizacji budowy sieci [95, 133, 165] i w analizie bezpieczeństwa sieci [9, 103];

- w naukach biologicznych na przykład są wykorzystywane do przewidywania struktur białek [43, 111, 126], czy też do dopasowania sekwencji biologicznych (DNA, RNA i białek) [23, 66].

W rozdziale tym przedstawiono podstawowe informacje o problemach optymalizacyjnych jako o grupie problemów, które najczęściej rozwiązywane są za pomocą algorytmów genetycznych. Umieszczono w nim także podstawowe informacje o algorytmach ewolucyjnych i ich podstawach biologicznych.

Szerzej omówione zostały algorytmy genetyczne, które są tematem badań w niniejszej pracy. W przypadku algorytmów genetycznych poza omówieniem zasady działania algorytmu genetycznego, zamieszczone zostały również informacje o operacjach zachodzących w algorytmach genetycznych. W rozdziale zawarto również przykładowe modyfikacje algorytmów genetycznych, a także przykłady zastosowania algorytmów genetycznych spotykane w literaturze.

## Wybrane problemy optymalizacyjne

W rozdziale tym omówione zostały cztery wybrane problemy optymalizacyjne, wraz z algorytmami genetycznymi służącymi ich rozwiązaniu. Zaprezentowane algorytmy są algorytmami bazowymi *AB*, na których zastosowane zostały autorskie modyfikacje inspirowane epigenetyką (zaprezentowane w Rozdziale 3), będące przedmiotem badań zaprezentowanych w niniejszej pracy. Na potrzeby badań, w *AB* zastosowano modyfikacje inspirowane epigenetyką, w celu sprawdzenia, czy zaproponowane przez autora modyfikacje wpłyną na poprawę efektywności działania algorytmów genetycznych.

Do sprawdzenia skuteczności proponowanych operacji bazujących na procesach epigenetycznych posłużono się algorytmami genetycznymi służącymi kolejno do: optymalnego załadunku, grupowania danych, wykrywania danych odstających i dopasowania sekwencji biologicznych. Zaprezentowane algorytmy genetyczne (poza algorytmem do grupowania danych) są algorytmami zaczerpniętymi z literatury. Algorytm genetyczny wykonujący grupowanie danych jest autorskim algorytmem genetycznym, opublikowanym w [40].

### 2.1 Problem optymalizacji załadunku

Pierwszym przedstawionym problemem wraz z algorytmem genetycznym go rozwiązującym jest problem optymalnego załadunku kontenerów [21, 47, 114]. Problemem jaki należy rozwiązać jest znalezienie optymalnego załadunku kontenera, tak aby uzyskać największy zysk na przewożonym towarze, ale bez przekroczenia dopuszczalnej masy całkowitej kontenera. Problem załadunku jest niezwykle istotny z punktu widzenia firm spedycyjnych, gdzie określony towar trzeba dostarczyć w określone miejsce, przy poniesieniu jak najniższych kosztów.

Formalna definicja problemu optymalizacji załadunku może być sformułowana następująco [129]:

do dyspozycji jest kontener o maksymalnej ładowności  $V_K$ , oraz zbiór  $N$  elementów  $\{x_1, \dots, x_j, \dots, x_N\}$ ,

przy czym każdy element zbioru ma określoną wagę  $m_j$ , oraz wartość  $c_j$ ,  $j \in \mathbb{N}$ .

Zmaksymalizuj:

$$\sum_{j=1}^N c_j x_j \quad (2.1)$$

Przy założeniach:

$$\sum_{j=1}^N m_j x_j \leq V_K, \quad x_j = 0 \text{ lub } 1, \quad j = 1, \dots, N \quad (2.2)$$

Wykorzystany algorytm genetyczny do dopasowania optymalnego załadunku bazuje KAG (por. Podrozdział 1.4 i Algorytm 1). Zawiera standardowe operacje dla algorytmów genetycznych takie jak operacje krzyżowania, mutacji, doboru osobników. Standardowe operacje wykorzystane w algorytmie genetycznym do optymalizacji załadunku:

- a) **Kodowanie osobników:** w wykorzystanym algorytmie występuje binarne kodowanie osobników. Wartość 1 odpowiada informacji, że dany towar należy załadować do kontenera, wartość 0, oznacza że danego towaru nie należy ładować do kontenera.
- b) **Funkcja przystosowania:** W algorytmie zastosowano funkcję przystosowania obliczaną w oparciu o trzy parametry:
  - wagę towaru do przewiezienia;
  - zapłatę otrzymaną za przewiezienie danego towaru;
  - maksymalną ładowność kontenera.

Najlepiej przystosowany osobnik to taki, dla którego dostarczy się jak najwięcej towarów za jak największą kwotę, nie przekraczając przy tym maksymalnej ładowności kontenera.

- c) **Selekcja osobników:** w wykorzystanym algorytmie zastosowano turniejową metodę selekcji osobników.
- d) **Krzyżowanie osobników:** w implementacji algorytmu zastosowano jednopunktową operację krzyżowania.
- e) **Mutacja osobników:** zastosowano mutacje punktowe, zmieniające pojedyncze geny w genotypie osobnika.

## 2.2 Problem grupowania danych

Drugim algorytmem wykorzystanym jako bazowy w eksperymentach (zaprezentowanych w rozdziale 4) nad możliwościami zastosowania modyfikacji inspirowanych procesami epigenetycznymi (omówionych w rozdziale 3) był autorski algorytm do grupowania danych.

Grupowanie danych polega na podzieleniu elementów zbioru, na konkretne podgrupy według ustalonego kryterium. Formalny opis problemu może zostać sformułowany następująco:

Przedmiotem grupowania jest skończony zbiór  $X$  składający się  $N$  elementów, oraz skończony zbiór  $A$  zawierający  $m$  atrybutów opisujących obiekty  $x \in X$  przyjmujących określone wartości  $x_y$ , gdzie  $A_y$  oznacza konkretny atrybut:

$$X = \{x_y\}_{y \in Y} \quad (2.3)$$

Analizie podlega zbiór grup  $C = \{C_1, C_2, \dots, C_N\}$ , stanowiący zbiór podzbiorów  $X$ . Grupą to podzbiór  $X \supseteq C_i$ .

$$X = \{x_1, \dots, x_N\} = \bigcup_{i=1}^c C_i \quad (2.4)$$

W przypadku spełnienia warunku pełnego pokrycia zbioru  $X$  (równanie 2.4), zbiór taki nazywany jest segmentacją  $S(X)$ .

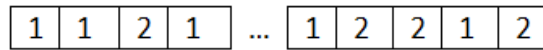
$$S(X) = \{C_1, \dots, C_N\} \quad (2.5)$$

Problem grupowania  $C$  polega na podziale obiektów na podzbiory (grupy), wyznacza się tak zwaną macierz przynależności dla każdego elementu  $x_i$  do poszczególnych grupy.

Sam problem grupowania danych może być problemem dość złożonym, nie zawsze istnieje oczywisty podział na grupy. Stąd też istnieje wiele algorytmów służących do grupowania danych, a ich odpowiedni dobór pozwala uzyskać oczekiwane rezultaty. Do najpopularniejszych należą algorytmy K-means [72], X-means [127], DBScan [55], grupowanie hierarchiczne [184], mapy samoorganizujące się [92] oraz wiele innych. Jednym z możliwych sposobów grupowania danych jest wykorzystanie algorytmów genetycznych. W literaturze można spotkać liczne przykłady algorytmów genetycznych służących rozwiązaniu tego problemu. W eksperymentach przeprowadzonych w niniejszej rozprawie wykorzystana została autorska modyfikacja algorytmu genetycznego do grupowania danych. Schemat działania algorytmu bazuje na KAG, przedstawionym w algorytmie 1.

W proponowanym algorytmie zastosowano standardowe operacje dla algorytmów genetycznych takie jak selekcja, krzyżowanie, mutacje. Poniżej przedstawiono sposób kodowania osobników i operacje jakie są wykorzystywane w proponowanym algorytmie genetycznym.

- a) **Kodowanie osobników:** w proponowanym algorytmie długość genotypu osobnika odpowiada liczbie elementów w zbiorze danych, który podlega grupowaniu. Kodowanie osobników odbywa się poprzez przypisanie każdemu z osobników numeru grupy (kodowanie liczbami całkowitymi). Przykład kodowania osobników przedstawiono na Rysunku 2.1.



**Rysunek 2.1:** Kodowanie osobnika w algorytmie do grupowania danych

- b) **Funkcja przystosowania:** w celu skonstruowania funkcji przystosowania osobników posłużono się miarą Akaike Information Criterion (AIC) [7, 147], odległością Hausdorfa między zbiorami, oraz gęstością zbioru. Wybór tych miar pozwolił na skonstruowanie funkcji przystosowania pozwalającej na osiągnięcie wysokich rezultatów grupowania danych przez proponowany algorytm. Sposób obliczania funkcji przystosowania pokazuje wzór 2.6.

$$fit_k = \frac{AIC(k)}{(1 + \sum_{i=1}^{n-1} d_H(i, i+1))^2} \left(1 + \sum_{i=1}^n dens(i)\right) \quad (2.6)$$

gdzie:

$fit_k$  – funkcja przystosowania osobnika  $k$ ,

$AIC(k)$  – miara AIC dla osobnika  $k$ ,

$d_H$  – miara Hausdorfa pomiędzy grupami,

$dens(i)$  – gęstość grupy,

$n$  – przewidywana liczba grup.

- c) **Selekcja osobników:** w proponowanym algorytmie do grupowania danych zastosowano turniejową metodę selekcji osobników.
- d) **Krzyżowania osobników:** w proponowanym algorytmie krzyżowanie osobników zachodzi na dwa sposoby (z równym prawdopodobieństwem wystąpienia). Pierwszym sposobem krzyżowania jest podział osobników na dwie części (krzyżowanie jednopunktowe). Oba osobniki wchodzące do procesu krzyżowania są dzielone w tym samym losowo wybranym miejscu (Rysunek 1.4a). W kolejnym kroku następuje wymiana materiału genetycznego pomiędzy osobnikami, w taki sposób, że każdy nowy osobnik posiada część genotypu jednego rodzica i część genotypu drugiego rodzica.
- Drugim wykorzystanym sposobem jest wybranie tego samego losowego fragmentu genotypu rodziców i ich wzajemna zamiana (krzyżowanie dwupunktowe Rysunek 1.4b).

- e) **Mutacja osobników:** W proponowanym algorytmie zastosowano najczęściej występujący rodzaj mutacji jakim jest mutacja punktowa. Ponieważ mutacja punktowa polega na zmianie pojedynczego genu w genotypie osobnika w przypadku proponowanego algorytmu zmiana skutkuje zmianą przynależności do grupy.

## 2.3 Problem wykrywania wartości odstających

Wykrywanie wartości odstających w zbiorach danych jest jednym z zadań analizy danych. Występowanie w zbiorze danych wartości odstających może mieć wpływ na przekłamanie wyników analizy danych. Dane odstające mogą też być tym, czego poszukuje się podczas analizy danych, są poszukiwaną anomalią, pojawiającą się w zbiorze danych, stąd ważne jest, żeby ich obecność wykryć na wczesnym etapie analizy danych.

Formalna definicja problemu wykrywania danych odstających, może być sformułowana następująco:

Dla danego zbioru  $D$  składającego się z  $N$  punktów  $\{x_1, x_2, \dots, x_N\}$ , gdzie każdy punkt jest wielowymiarowym wektorem  $m$  atrybutów  $A$  poszukuje się podzbioru  $O \subseteq D$  o rozmiarze  $K$ , w taki sposób, aby minimalizować entropię  $E(D - O)$  (czyli średnią ilość informacji):

$$\min_{O \subseteq D} E(D - O) \quad (2.7)$$

gdzie  $|O| = k$

W eksperymentach wykorzystano algorytm genetyczny do wykrywania danych odstających zaprezentowany w [11]. Algorytm bazuje na  $KAG$ (por. Algorytm 1).

W przypadku algorytmu do wykrywania wartości odstających zasady tworzenia populacji początkowej, sposób obliczania wartości funkcji przystosowania i podstawowe operacje w algorytmie są następujące:

- a) **Kodowanie osobników:** kodowanie osobników w wykorzystanym algorytmie jest binarne, gdzie 0 oznacza wartość niebędącą wartością odstającą, a 1 wartość odstającą. Długość osobnika odpowiada liczbie elementów w zbiorze danych.
- b) **Funkcja przystosowania:** do obliczenia wartości funkcji przystosowania posłużono się kryterium informacyjnym Akaikego [7], oraz karą za wartości odstające w zbiorze danych. Wzór na obliczenie funkcji przystosowania osobnika w populacji jest następujący:

$$fit_k = AIC(k) + \kappa m_d \log(n) \quad (2.8)$$

gdzie:

$fit_k$  – funkcja przystosowania osobnika  $k$ ,

$AIC(k)$  – miara AIC dla osobnika  $k$ ,

$\kappa$  – współczynnik kary za wartość odstającą,

$m_d$  – liczba wartości odstających w zbiorze,

$n$  – liczba osobników.

c) **Selekcja osobników**: w wykorzystanym algorytmie skorzystano z turniejowej metody selekcji osobników (por. Podrozdział 1.4.2).

d) **Mutacja osobników**: W algorytmie zastosowano mutację punktową.

## 2.4 Problem dopasowania sekwencji biologicznych

Ostatnim algorytmem genetycznym, który został wykorzystany do oceny skuteczności wprowadzonych modyfikacji jest algorytm do dopasowania sekwencji DNA, RNA i białek. Ustalenie podobieństwa pomiędzy sekwencjami jest niezwykle złożonym procesem. Porównywanie sekwencji DNA, RNA i białek ma duże znaczenie w ustalaniu za jakie procesy biologiczne odpowiada dana sekwencja. W przypadku sekwencji biologicznych, sekwencje o podobnej budowie mają bardzo często zbliżone funkcje. Porównywanie sekwencji jest również często wykorzystywane w badaniach nad przebiegiem ewolucji organizmów żywych. Algorytmem wykorzystanym w eksperymentach jest algorytm GAMSА (ang. *Genetic Algorithm for Multiple Sequence Alignment*) [68]. Formalna definicja problemu dopasowania przedstawia się następująco:

Niech  $\Sigma' = \Sigma \cup \{ \_ \}$  jest alfabetem rozszerzonym o znak reprezentujący pustą przestrzeń. Dane są sekwencje  $s \in \Sigma^*$  o długości  $m$ , oraz  $t \in \Sigma^*$  o długości  $N$ ,  $\alpha = \{s', t'\}$  jest dopasowaniem  $s$  i  $t$ , wtedy i tylko wtedy, gdy:

- $s', t' \in (\Sigma')^*$ ,
- $|s'| = |t'| = l$ , gdzie  $\max(m, n) \leq l \leq m + n$ ,
- $s$  jest sekwencją powstałą w wyniku usunięcia „\_” z  $s'$ , a  $t$  jest sekwencją powstałą w wyniku usunięcia „\_” z  $t'$ ,
- nie istnieje wartość  $i$ , dla której  $s'[i] = t'[i] = \_$

Pod względem schematu działania algorytm GAMSА nie różni się od klasycznego algorytmu genetycznego (por. Algorytm 1), różni się natomiast istotnie pod względem samych operacji tworzenia populacji, krzyżowania i mutacji.



- a) **Kodowanie osobników:** Populacja początkowa tworzona jest na podstawie zadanych do porównania sekwencji, w taki sposób, że każdy osobnik w populacji zawiera wszystkie porównywane sekwencje. Ze względu na to, że nie można dokonać zamiany miejsc kwasów nukleinowych czy aminokwasów w sekwencji – zmieniłoby to zadaną sekwencję na zupełnie inną, poszczególne osobniki w populacji tworzy się poprzez wstawienie przerw do sekwencji. Maksymalna liczba przerw wstawionych do najdłuższej sekwencji odpowiada długości tej sekwencji. Do pozostałych sekwencji dokłada się w losowe miejsca przerwy w takiej liczbie, aby ich długość była taka sama, jak najdłuższej sekwencji z wstawionymi przerwami. Przykład tworzenia osobnika na podstawie zadanych sekwencji przedstawiono na Rysunku 2.2.



*Rysunek 2.2: Operacja tworzenia osobnika w algorytmie GAMSА*

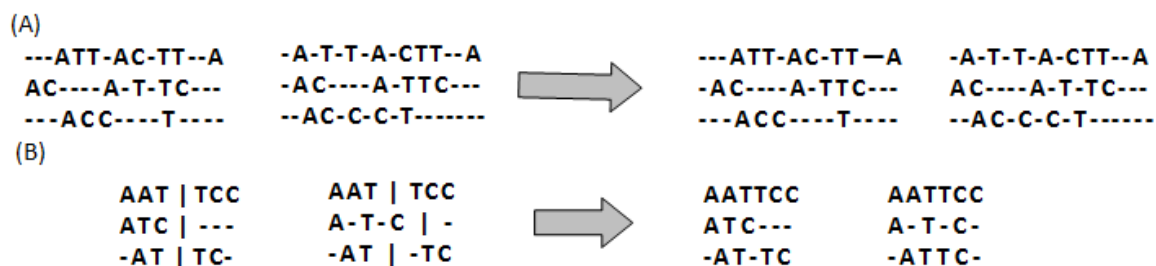
Utworzone w ten sposób osobniki stanowią populację początkową, na której wykonywane są kolejne operacje GA.

- b) **Funkcja przystosowania:** w algorytmie GAMSА do obliczenia funkcji przystosowania dla każdego osobnika wprowadza się nagrody za dobre dopasowanie i kary za błędne. Jednym ze sposobów obliczania funkcji przystosowania spotykanym w literaturze jest przyznawanie kar i nagród na zasadach:

- +1 za zgodność na pozycji, czyli ten sam aminokwas lub kwas nukleinowy w obu sekwencjach;
- -1 za niezgodność, czyli różne aminokwasy lub kwasy nukleinowe na tej samej pozycji;
- -2 kara za przerwę, gdy w jednej sekwencji występuje aminokwas lub kwas nukleinowy, a w drugiej w tym miejscu pojawia się przerwa.

- c) **Krzyżowanie osobników:** operacja krzyżowania również różni się od standardowej operacji krzyżowania występującej w algorytmach genetycznych. W algorytmie GAMSА występują dwie metody krzyżowania i obydwie zachodzą równocześnie w trakcie działania algorytmu. Pierwszym sposobem krzyżowania osobników jest krzyżowanie horyzontalne. W tym sposobie krzyżowania pomiędzy osobnikami wymieniana jest jedna cała sekwencja, proces krzyżowania horyzontalnego pokazano na Rysunku 2.3.A.

Drugim zastosowanym w algorytmie sposobem krzyżowania jest krzyżowanie wertykalne. Krzyżowanie wertykalne bardziej przypomina standardową operację krzyżowania, występu-



*Rysunek 2.3: Operacje krzyżowania w algorytmie GAMS*

jąca w klasycznym algorytmie genetycznym. Krzyżowanie to polega na przecięciu osobników w określonym miejscu i wymianę pomiędzy nimi fragmentów sekwencji, co prowadzi do powstania nowego osobnika. Dla pierwszej sekwencji biorącej udział w krzyżowaniu miejsce przecięcia wyznaczane jest w sposób losowy. W przypadku drugiej sekwencji miejsce przecięcia jest wyznaczane jako kolejne miejsce po wystąpieniu ostatniego aminokwasu lub kwasu nukleinowego z fragmentu z sekwencji pierwszej. Spowodowane jest to tym, że w przypadku operacji na sekwencjach nie można zmienić liczby i kolejności aminokwasów lub kwasów nukleinowych w sekwencji. Ewentualne braki lub nadmiary w długości nowych osobników niweluje się poprzez dołożenie lub usunięcie końcowych przerw. Przykład krzyżowania wertrykalnego pokazano na Rysunku 2.3.B.

- d) **Operacja mutacji:** pierwszą cechą odróżniającą mutację zastosowaną w GAMS od klasycznych mutacji, jest częstotliwość jej wystąpienia, w klasycznym algorytmie genetycznym mutacje zachodzą z niewielkim prawdopodobieństwem, wynoszącym do kilku procent, a w przypadku algorytmu GAMS prawdopodobieństwo wystąpienia mutacji wynosi od 50% do 100%. Operacja mutacji w algorytmie GAMS polega na losowym dołożeniu lub usunięciu przerw w sekwencji. Przykład zastosowania operatora mutacji pokazano na Rysunku 2.4.



*Rysunek 2.4: Operacja mutacji w algorytmie GAMS*

Przedstawione algorytmy genetyczne posłużyły za bazowe algorytmy do sprawdzenia możliwości zastosowania autorskich modyfikacji algorytmu genetycznego inspirowanych procesami epigenetycznymi (Rozdział 3), a wyniki eksperymentów po ich modyfikacji zostaną omówione w Rozdziale 4.

## Modyfikacje algorytmu genetycznego imitujące procesy epigenetyczne

W rozdziale tym zaprezentowano proponowane przez autora rozprawy modyfikacje algorytmów genetycznych. Autor stworzył trzy dodatkowe operacje inspirowane procesami epigenetycznymi. Proponowane nowe operacje są niezależne od siebie i mogą być stosowane niezależnie od siebie.

Proponowane autorskie operacje bazują na trzech procesach epigenetycznych, jakimi są:

- dziedziczenie za pomocą prionu,
- metylacja cytozyny,
- wyłączenie alleliczne.

W rozdziale tym przedstawiono odniesienie do biologicznych podstaw proponowanych modyfikacji ich ideę oraz sposób ich zastosowania w algorytmach genetycznych. Celem zastosowania proponowanych modyfikacji epigenetycznych jest zmniejszenie liczby generacji potrzebnych do znalezienia najlepszego rozwiązania przez algorytm genetyczny. Wyniki eksperymentów sprawdzające przydatność zastosowania omówionych w tym rozdziale modyfikacji przedstawiono w Rozdziale 4.

### Epigenetyka

Epigenetyka [8,31,48,119] jest nauką zajmującą się badaniem procesów dziedziczenia pozagenowego, a także wpływem czynników zewnętrznych na poziom ekspresji genów. Ekspresja genów determinuje cechy fenotypowe osobnika, czyli jego przystosowanie do środowiska, oraz sposób zachowania, wygląd itp. Od dłuższego czasu naukowcy zastanawiali się na przykład,

dlaczego istnieje różnica w wyglądzie i sposobie zachowywania się bliźniąt jednojajowych, czy też dlaczego sklonowane osobniki, mimo identycznych sekwencji genów, posiadają jednak inne ubarwienie, przejawiały inne zachowanie. Część tych zagadek dało się wytłumaczyć dzięki odkryciu mechanizmów będących przedmiotem badań epigenetyki.

Pojęcia związane z epigenetyką pojawiły się w momencie odkrycia, że część zmian w genotypie organizmów żywych nie jest bezpośrednio związana ze strukturą DNA, jej zmianami i procesami dziedziczenia. Zaczęto wtedy zastanawiać się, co może powodować te zmiany. Efektem badań było odkrycie licznych cząsteczek mających wpływ na procesy zachodzące w żywych organizmach, mających wpływ na to, w jaki sposób kod genetyczny zostanie odczytany. Inaczej mówiąc – jak zmieni się fenotyp osobnika, bez zmiany jego DNA.

Okazało się, że procesy epigenetyczne odgrywają znaczącą rolę w różnicowaniu się populacji i dostosowywaniu do nowych warunków. Modyfikacje epigenetyczne są też źródłem części chorób, a także mają wpływ na cechy osobnicze, takie jak na przykład postrzeganie świata, czy cechy osobowości. Można stwierdzić, że genotyp organizmów żywych jest miejscem przechowywania względnie statycznej informacji genetycznej, a procesy epigenetyczne są swoistymi dynamicznymi kontrolerami (inhibitorami lub katalizatorami), odpowiadającymi za aktywację pewnych informacji.

Procesy epigenetyczne, poprzez zmiany biochemiczne wpływają na strukturę DNA u organizmów żywych, a także na poziom ekspresji genów. Uruchomienie danego procesu epigenetycznego może odbywać się na skutek zmian zachodzących w organizmie (również w sposób losowy), a także w wyniku oddziaływania czynników zewnętrznych (środowiskowych). Obecne badania wskazują, że środowisko w jakim przebywa dany organizm, składniki diety, a także związki chemiczne mogą być impulsami do uruchomienia danego procesu epigenetycznego w organizmie. Naukowcy szacują że dokładne poznanie procesów epigenetycznych pozwoli opracować skuteczne terapie genowe na wiele chorób. Do głównych modyfikacji zachodzących w naszym genotypie o podłożu epigenetycznym można zaliczyć:

- modyfikacje powstałe w wyniku oddziaływania wolnych cząstek – w każdej komórce oprócz nici DNA znajdują liczne krótkie fragmenty przypominające strukturą nici DNA, które mogą przyłączać się do DNA i modyfikować białko kodowane przez dany fragment genotypu (np. dziedziczenie za pomocą prionu).
- modyfikacje na poziomie nici DNA – najczęściej prowadzą do wyciszenia ekspresji konkretnego genu, czyli zmniejszenia jego poziomu ekspresji (np. proces metylacji cytozyny);

- modyfikacje na poziomie histonów – histony są niewielkimi białkami, wokół których owinięta jest nić DNA; ich modyfikacja może wpływać na zwiększenie, bądź zmniejszenie syntezy białek zakodowanych w danym odcinku DNA (np. deacetylizacja, wyłączenie alleliczne);

W związku z tym, że epigenetyka jest stosunkowo młodą dziedziną nauki, a same procesy epigenetyczne zostały odkryte stosunkowo niedawno, to nie znalazły one jeszcze odzwierciedlenia w algorytmach ewolucyjnych. W związku z tym autor postanowił sprawdzić, czy tak jak w przypadku organizmów żywych (gdzie epigenetyka wpływa na szybsze różnicowanie się populacji i jej lepsze dopasowanie do warunków środowiskowych), również w przypadku algorytmów genetycznych zastosowanie operacji naśladujących procesy epigenetyczne poprawi działanie tychże algorytmów (poprzez szybsze znalezienie optymalnego wyniku przez dany algorytm genetyczny).

W kolejnych podrozdziałach zaprezentowane zostaną szczegóły trzech autorskich operacji algorytmów genetycznych inspirowanych procesami epigenetycznymi.

### 3.1 Modyfikacja imitująca dziedziczenie za pomocą prionu

Pierwszą autorską modyfikacją zaprezentowaną w pracy, jest modyfikacja naśladująca proces epigenetyczny dziedziczenia za pomocą prionu (*Epi\_1*). Proponowana modyfikacja jest dodatkową operacją przeprowadzaną na populacji w algorytmie genetycznym (por. Rysunek 1.5). Celem jej wprowadzenia jest redukcja liczby generacji algorytmu genetycznego potrzebnej do uzyskania najlepszego rozwiązania.

#### 3.1.1 Molekularne podstawy procesu dziedziczenia za pomocą prionu

Priony są cząsteczkami białkowymi występującymi w każdym żywym organizmie [109]. W normalnych warunkach priony są nieaktywne (priony nieinfekcyjne) i nie mają wpływu na procesy zachodzące w żywym organizmie. Jednakże możliwe jest wystąpienie impulsu z zewnątrz lub wewnątrz organizmu aby prion przeszedł ze stanu neutralnego do stanu inwazyjnego (prion infekcyjny). W przypadku czynnika zewnętrznego bardzo często następuje przejście prionu do stanu inwazyjnego u wszystkich (lub większości) przedstawicieli danej populacji poddanych działaniu tego samego czynnika zewnętrznego.

Przykładem takiego działania prionu jest aktywacja prionu, który umożliwia ochronę białek przed denaturacją termiczną u populacji muszki owocówki poddanej działaniu wysokiej temperatury. W stanie inwazyjnym priony mogą odpowiadać za pojawienie się chorób oraz za zmiany ewolucyjne (powodując, że organizm staje się lepiej przystosowany do nowych warunków).

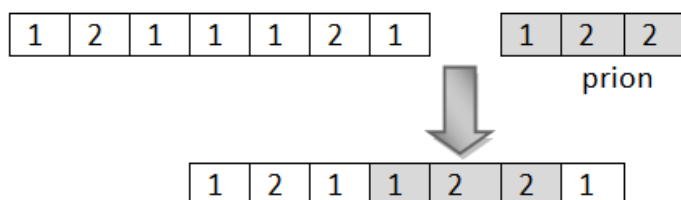
Priony mogą też oddziaływać na proces ekspresji genów, mogą mieć również wpływ na kod genetyczny organizmu poprzez wbudowanie się sekwencji kodującej prion do sekwencji DNA organizmu.

### 3.1.2 Opis modyfikacji imitującej dziedziczenie za pomocą prionu

W proponowanej modyfikacji odwzorowano proces dziedziczenia za pomocą prionu w obrębie populacji, na którą zadziałał ten sam czynnik zewnętrznym. Algorytmem bazowym był KAG, do którego wprowadzono dodatkowy element – symulujący wystąpienie czynnika zewnętrznego. W zmodyfikowanym algorytmie genetycznym, poza standardowymi operatorami, jak krzyżowanie i mutacja, wprowadzono jeszcze jeden proces: osobniki z danej populacji są z pewnym prawdopodobieństwem (optymalne prawdopodobieństwo wystąpienia modyfikacji było przedmiotem badań, których wyniki przedstawiono w rozdziale 4) wystawione na oddziaływanie pewnego czynnika zewnętrznego. Czynnikiem zewnętrznym jest niewielkiej długości fragment genotypu (symulacja cząsteczki prionu), odpowiadający pod względem kodowania kodowaniu osobników w algorytmie genetycznym. W uproszczeniu, działanie tego czynnika powoduje wystąpienie zmiany w kodzie osobników, zmiana jest taka sama u wszystkich osobników i pojawia się w tym samym miejscu w sekwencji DNA. Powoduje to zmianę fragmentu genotypu osobnika, w algorytmie genetycznym, na sekwencję prionu.

Sekwencja prionu jest generowana z równym prawdopodobieństwem (wynoszącym po 50% – jest to wartość wyznaczona eksperymentalnie) z genotypu osobnika o najwyższej wartości funkcji przystosowania lub w sposób losowy. Dziedziczeniu za pomocą prionu nie podlega osobnik z najlepszą wartością funkcji przystosowania – ma to na celu zabezpieczenie algorytmu przed utratą najlepszego rozwiązania (jest on w pewnym sensie najsilniejszy i powinien przetrwać zmiany środowiska). Na rysunku 3.1 pokazano proces dziedziczenia za pomocą prionu.

Długość generowanej sekwencji symulującej prion jest losowa, przy czym nie może być dłuższa niż 10% długości genotypu osobnika (maksymalna długość prionu na poziomie do 10% została ustalona, aby prion nie dokonywał zmian zbyt dużego fragmentu sekwencji osobnika, co mogłoby prowadzić do zaniku różnorodności w populacji). Ogólny zapis algorytmu realizującego modyfikację symulującą proces dziedziczenia za pomocą prionu przedstawiono jako Algorytm 2.



*Rysunek 3.1: Operacja imitująca dziedziczenie za pomocą prionu – Epi\_1*

---

**Algorytm 2:** Algorytm operacji imitującej dziedziczenie za pomocą prionu – Epi\_1

---

```

1 Dane wejściowe: populacja osobników, prawdopodobieństwo prionu ( $P_p$ )
2 Wynik: populacja osobników poddana działaniu prionu
3 begin
4   | Wylosuj długość prionu z zakresu  $\langle 0; 10\% \rangle$  długości osobnika;
5   | Wylosuj miejsce wbudowania prionu;
6   | Wylosuj liczbę binarną;
7   | if liczba = 0 then
8   |   | Wygeneruj prion z osobnika z najlepszą wartością funkcji przystosowania;
9   | end
10  | else
11  |   | utwórz prion w sposób losowy
12  | end
13  | Z  $P_p$  wybierz osobników poddanych działaniu prionu;
14  | foreach wybrany osobnik do
15  |   | wbuduj prion w genotyp osobnika;
16  | end
17 end

```

---

Na podstawie Algorytmu 2, operacja przeprowadzana jest na wybranych z pewnym prawdopodobieństwem (będącym przedmiotem badań) osobnikach z populacji. W przeprowadzonej operacji generowany jest genotyp prionu, odpowiadający pod względem kodowania genotypowi osobników. Zmiana w genotypie osobnika odbywa się poprzez wbudowanie sekwencji prionu w genotyp osobnika w z góry ustalonym miejscu. Genotyp prionu generowany jest z osobnika o najlepszej wartości funkcji przystosowania, lub w sposób losowy. Do genotypu osobników wybranych do operacji *Epi\_1* wbudowywany jest genotyp prionu zastępując genotyp osobnika. Algorytm 2 przedstawia zapis tylko proponowanej operacji *Epi\_1*. Pozostałe operacje algorytmu genetycznego pozostają bez zmian.

Wyniki badań w których sprawdzono, czy zastosowanie modyfikacji *Epi\_1* w algorytmach genetycznych poprawia działanie tychże algorytmów, zostały udokumentowane w Rozdziale 4.

## 3.2 Modyfikacja imitująca metylacje cytozyny

Druga autorska modyfikacja (nazwana *Epi\_2*) jest modyfikacją operacji krzyżowania w algorytmie genetycznym (por. Rysunek 1.5), naśladującą proces epigenetyczny metylacji cytozyny. Wystąpienie proponowanej modyfikacji powoduje zmianę sposobu krzyżowania osobników.

### 3.2.1 Molekularne podstawy procesu metylacji cytozyny

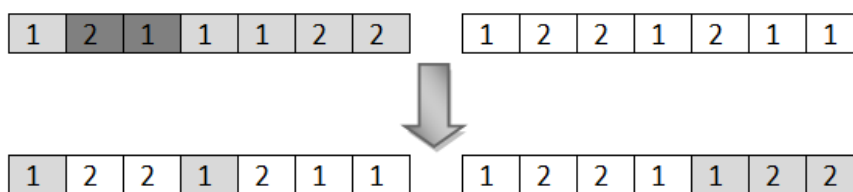
Metylacja cytozyny jest procesem polegającym na przyłączeniu się grup metylowych ( $-CH_3$ ) do zasad azotowych nukleotydów (podstawowy składnik budulcowy kwasów nukleinowych DNA i RNA). Przyłączenie grupy metylowej do nukleotydów wpływa na redukcję poziomu ekspresji genów kodowanych przez dany fragment DNA. W przypadku zmetylowania dużego fragmentu nici DNA może dojść do blokowania fragmentu sekwencji, tak aby nie dokonano odczytania genu. Proces metylacji cytozyny może mieć też wpływ na przekazywanie informacji genetycznych poprzez uniemożliwienie przekazania określonego genu w procesie dziedziczenia [34, 104, 149, 178, 183]. Metylacja cytozyny u organizmów żywych wpływa również na proces formowania się tkanek (na specjalizację komórek). Jest również niezwykle istotny w ogólnym funkcjonowaniu organizmu, bowiem jedynie niewielka liczba genów ma stały poziom ekspresji (tzw. geny metabolizmu podstawowego ang. *housekeeping genes* – odpowiadające za podstawowe funkcje życiowe). Pozostałe geny, w wyniku różnych procesów biologicznych, między innymi metylacji cytozyny, mają zmienne poziomy ekspresji, co ma wpływ na prawidłowe funkcjonowanie organizmu. Metylacja cytozyny może też pomagać organizmowi w pozbyciu się kodu DNA pochodzącego z zewnątrz (np. od wirusów), co powoduje zabezpieczenie organizmu przed wpływem tego kodu. Zaburzenie procesu metylacji może prowadzić do rozwoju na przykład chorób nowotworowych, czy też innych chorób o podłożu genetycznym (np. zespołów Angelmana, Pradera-Williego, Beckwitha-Wiedemana).

### 3.2.2 Opis modyfikacji imitującej metylację cytozyny

W proponowanej modyfikacji zasymulowano proces blokowania fragmentu sekwencji genotypu osobnika. Modyfikacja *Epi\_2* została zaimplementowana w operacji krzyżowania osobników, modyfikując tę operację. Osobniki potomne zawierają fragmenty genotypu każdego z rodziców. Jeżeli w operacji krzyżowania wystąpi proces symulujący metylację cytozyny, następuje zablokowanie określonego fragmentu genotypu osobnika, przez co nie bierze on udziału w krzyżowaniu. Jest to równoznaczne z tym, że fragment genomu podlegający procesowi metylacji cytozyny nie zostanie przekazany potomkom, czyli nie pojawi się w nowej populacji.



Na Rysunku 3.2 pokazano schemat przebieg procesu metylacji cytozyny w procesie krzyżowania osobników. Ciemnoszary fragment genotypu osobnika uległ procesowi metylacji, czyli nie bierze udziału w procesie krzyżowania.



**Rysunek 3.2:** Operacja  $Epi\_2$  – imitująca proces metylacji cytozyny (opis w tekście)

W celu zapobieżenia utracie najlepszych rozwiązań w procesie krzyżowania proces blokowania sekwencji występuje u osobników z najniższymi wartościami funkcji przystosowania, co ma na celu przekazanie osobnikom potomnym większego fragmentu genotypu osobnika lepiej przystosowanego. Oznacza to, że oba osobniki potomne otrzymują jedynie określony fragment genotypu od rodzica mającego lepszą wartość funkcji przystosowania. Dzięki temu następuje „wyciszenie” sekwencji genów o gorszej funkcji przystosowania (od osobnika gorzej przystosowanego).

Algorytm 3 przedstawia pseudokod procesu symulującego proces metylacji cytozyny ( $Epi\_2$ ). Proponowana modyfikacja  $Epi\_2$  zachodzi w procesie krzyżowania z odpowiednim prawdopodobieństwem, którego optymalna wartość została wyznaczona empirycznie (por. Rozdział 4).

---

**Algorytm 3:** Zmodyfikowany algorytm krzyżowania z uwzględnieniem symulacji procesu metylacji cytozyny

---

```

1 Dane wejściowe: populacja osobników, prawdopodobieństwo metylacji ( $P_m$ ),
  funkcja przystosowania osobników  $F_p$ 
2 Wynik: nowa populacja utworzona w wyniku krzyżowania z modyfikacją  $Epi\_2$ 
3 wybierz z prawdopodobieństwem  $P_m$  pary osobników które zostaną poddane
  procesowi krzyżowania z modyfikacją  $Epi\_2$ ;
4 foreach wybranej pary osobników do
5   | wylosuj długość i lokalizację zablokowanego fragmentu;
6   | zablokuj fragment genotypu u wybranego osobnika;
7   | if  $F_p(\text{osobnik1}) > F_p(\text{osobnik2})$  then
8   |   | zablokuj fragment genotypu u osobnika2;
9   | end
10  | else
11  |   | zablokuj fragment genotypu u osobnika1;
12  | end
13  | przeprowadź operację krzyżowania bez udziału zablokowanego fragmentu;
14 end

```

---

Przedstawiony algorytm modyfikacji *Epi\_2* (Algorytm 3) przedstawia zmodyfikowaną operację krzyżowania w momencie wystąpienia modyfikacji *Epi\_2*, reszta działania algorytmu odpowiada KAG. Na podstawie Algorytmu 3 pewna grupa osobników wybrana spośród całej populacji podlega zmodyfikowanej operacji krzyżowania, pozostałe osobniki podlegają standardowej operacji krzyżowania. W pierwszym kroku w sposób losowy generowana jest długość genotypu podlegającego blokowaniu w ramach modyfikacji *Epi\_2*, oraz miejsce wystąpienia blokowania. W kolejnym kroku porównywane są wartości funkcji przystosowania osobników biorących udział w krzyżowaniu. U osobnika z niższą wartością funkcji przystosowania następuje zablokowanie przekazania do potomków fragmentu genotypu, którego lokalizacja i długość została ustalona we wcześniejszym kroku. Na osobnikach przeprowadza się operację krzyżowania bez udziału zablokowanego fragmentu genotypu jednego z rodziców.

Badania nad oceną skuteczności wprowadzenia modyfikacji *Epi\_2* do algorytmu genetycznego (na przykładzie 4 wybranych zastosowań algorytmów genetycznych) zostały udokumentowane w Podrozdziale 4 niniejszej pracy.

### 3.3 Modyfikacja imitująca wyłączenie alleliczne

Trzecią proponowaną modyfikacją algorytmu genetycznego (oznaczoną jako *Epi\_3*), jest modyfikacją w której zasymulowany został proces epigenetyczny wyłączenia allelicznego. Modyfikacja *Epi\_3* jest dodatkową operacją występującą w algorytmie genetycznym (por. Rysunek 1.5). W niniejszym podrozdziale przedstawiono podstawy biologiczne modyfikacji oraz opis jej implementacji w algorytmach genetycznych.

#### 3.3.1 Molekularne podstawy procesu wyłączenia allelicznego

Wyłączenie alleliczne jest procesem biologicznym polegającym na wyciszeniu fragmentu DNA odpowiedzialnego za kodowanie danego genu (allel patrz Tabela 1.3). W wyniku jego wyciszenia następuje aktywacja innego fragmentu kodującego na przykład zmodyfikowaną wersję danego genu. W wyniku wyłączenia allelicznego powstają zmiany w białkach kodowanych przez nie DNA [48, 119]. Przykładem wyłączenia allelicznego jest rozwój cech płciowych u drożdży, gdzie wyłączenie odpowiedniego fragmentu sekwencji nukleotydowej i zastąpienie jej innym determinuje płeć. W przypadku dziedziczenia dwóch rodzajów genów od rodziców odpowiadających za kodowanie tego samego białka w różnych izoformach, u eukariotów (organizmów wielokomórkowych) wyłączenie alleliczne może mieć wpływ na to, która wersja kodu genetycznego zostanie aktywowana (czyli którą formę białka będzie produkował organizm), tak aby

u organizmu potomnego występowała tylko jedna forma danego białka. Na przykład w przypadku wyłączenia allelicznego występującego u limfocytów B następuje rearanżacja genomu kodującego dane białko (czyli przebudowanie danego fragmentu genotypu).

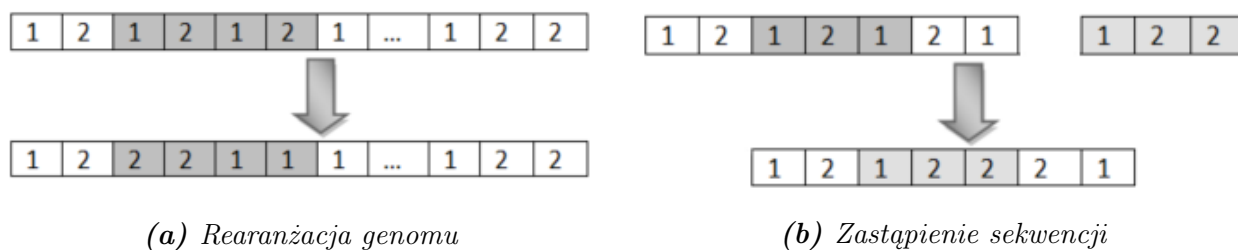
### 3.3.2 Opis modyfikacji

Proponowana modyfikacja  $Epi\_3$ , naśladująca proces wyłączenia allelicznego, stanowi dodatkową operację występującą w algorytmach genetycznych. Prawdopodobieństwo wystąpienia modyfikacji  $Epi\_3$  było przedmiotem badań eksperymentalnych, których wyniki zostały udokumentowane w Rozdziale 4.

W proponowanej modyfikacji proces wyłączenia allelicznego jest zasymulowany jako proces modyfikujący określony fragment sekwencji genotypu osobnika. Następuje dezaktywacja fragmentu genotypu i zastąpienie go nowym fragmentem. Modyfikacja fragmentu genotypu osobnika w procesie wyłączenia allelicznego odbywa się na dwa sposoby (każde występuje z prawdopodobieństwem wynoszącym 50%), jako dwie osobne (niezależne) operacje:

- **rearanżacja genomu** – zdezaktywowany fragment genotypu podlega procesowi rearanżacji, to znaczy jego geny zostają w losowy sposób zamienione pozycjami w danym fragmencie genotypu osobnika (Rysunek 3.3a);
- **zastąpienie sekwencji** – zdezaktywowany fragment genotypu osobnika zastępowany jest przez inny fragment odpowiadający długością fragmentowi zdezaktywowanemu. W proponowanym rozwiązaniu dezaktywowany fragment zastępowany jest nowym utworzonym w sposób losowy, zgodny ze sposobem kodowania osobników w danym algorytmie genetycznym (Rysunek 3.3b).

Na Rysunku 3.3 przedstawiono schematycznie obie składowe operacji  $Epi\_3$  na osobniku z populacji w algorytmie genetycznym.



**Rysunek 3.3:** Sposób przeprowadzenia operacji  $Epi\_3$  na genotypie osobnika

W Algorytmie 4 przedstawiono sposób realizacji proponowanej modyfikacji  $Epi\_3$  w algorytmie genetycznym.

---

**Algorytm 4:** Algorytm operacji wyłączenia allelicznego – Epi\_3

---

```
1 Dane wejściowe: populacja osobników, prawdopodobieństwo wystąpienia
  modyfikacji ( $P_m$ )
2 Wynik: populacja osobników poddana działaniu wyłączenia allelicznego
3 Utwórz pod populację osobników najgorzej przystosowanych - odwrotna selekcja;
4 foreach wybrany osobnik do
5   | wylosuj typ wyłączenia allelicznego;
6   | wylosuj długość zmiany w genotypie;
7   | wylosuj miejsce wystąpienia modyfikacji;
8   | if typ = rearanżacja genomu then
9     |   wykonaj rearanżację genomu w wybranym miejscu;
10  | end
11  | else
12  |   Utwórz nowy fragment genotypu;
13  |   Zastąp fragment genotypu nowym fragmentem;
14  | end
15 end
```

---

Na podstawie Algorytmu 4, w pierwszym kroku operacji *Epi\_3* wykonywana jest operacja odwrotnej selekcji osobników. W standardowej operacji selekcji chodzi o to, aby wybrać osobniki lepiej przystosowane, w przypadku odwrotnej selekcji wybrani zostają osobnicy najgorzej przystosowani. Jednym z założeń modyfikacji *Epi\_3* jest to, aby modyfikacji ulegały częściej osobniki z gorszą wartością funkcji przystosowania, czyli selekcja ma zwiększyć prawdopodobieństwo wyboru do operacji *Epi\_3* osobników najgorzej przystosowanych. W kolejnym kroku na wybranej podpopulacji osobników z równym prawdopodobieństwem wystąpienia (po 50%) wykonywana jest operacja rearanżacji genomu, lub zastąpienia fragmentu genotypu. Rearanżacja polega na zamianie kolejności wystąpienia genów w wybranym fragmencie genotypu. W przypadku zastąpienia fragmentu genotypu, wybrany losowo fragment genotypu zastępowany jest losowym fragmentem, odpowiadającym pod względem kodowania genotypu kodowaniu osobników w algorytmie – miejsce zmiany genotypu i sekwencja generowane są dla każdego osobnika indywidualnie. Zmodyfikowany osobnik, poddany procesowi wyłączenia allelicznego, pozostaje w populacji osobników i bierze udział w kolejnych krokach algorytmu genetycznego.

Wyniki badań nad oceną skuteczności wprowadzenia modyfikacji *Epi\_3* do algorytmu genetycznego (na bazie przykładowych algorytmów genetycznych) zostały przedstawione w Rozdziale 4.

## Eksperymenty

W rozdziale tym przedstawione zostały wyniki eksperymentów, mające na celu sprawdzenie, czy i w jakim stopniu proponowane przez autora pracy modyfikacje naśladujące procesy epigenetyczne mają wpływ na poprawę skuteczności działania algorytmów genetycznych. W rozdziale tym przedstawiono również wyniki badań nad empirycznym doбором parametrów proponowanych modyfikacji. Ocenie skuteczności podlegają trzy zaproponowane przez autora rozprawy modyfikacje algorytmów genetycznych, zaprezentowane w Rozdziale 3 (*Epi\_1*, *Epi\_2*, *Epi\_3*). Modyfikacje zostały zastosowane dla algorytmów genetycznych zaprezentowanych w Rozdziale 2 (zwanych dalej algorytmami bazowymi, *GAB*). W eksperymentach empirycznie wyznaczono wartość optymalnego prawdopodobieństwa wystąpienia danej modyfikacji, to znaczy dla jakiego prawdopodobieństwa wystąpienia modyfikacji uzyskuje się najlepszą poprawę działania algorytmu.

Proponowane modyfikacje zostały sprawdzone pod kątem poprawy działania algorytmów genetycznych, rozumianych jako redukcję liczby generacji potrzebnych do uzyskania oczekiwanego wyniku. Otrzymane wyniki zostały porównane z *GAB*. Algorytmy z proponowanymi modyfikacjami zostały również ocenione pod kątem czasu potrzebnego na przeprowadzenie kroków algorytmu genetycznego po dodaniu modyfikacji w pojedynczej generacji algorytmu, a wyniki porównane z rezultatami otrzymanymi dla *GAB*.

### 4.1 Warunki eksperymentów

Implementacje algorytmów zostały w całości wykonane przez autora wykorzystując skryptowy język programowania R, dostępny w ramach The R project [135]. Do pomiarów czasów przeprowadzonych eksperymentów wykorzystany został pakiet `Microbenchmark` [115]. Jako środowisko programistyczne wykorzystane zostało narzędzie R-Studio [143]. Wszystkie eksperymenty zostały przeprowadzone na komputerze klasy PC, o parametrach:

- **System operacyjny:** Windows 10 64 bit;
- **Procesor:** dwurdzeniowy Intel Core I7-6500U, o taktowaniu procesora 2,5 GHz;
- **Pamięć operacyjna:** 8 GB;
- **Wersja języka R:** 3.4.1 (64 bit);
- **Wersja R-Studio:** 1.1.463.

## 4.2 Metodyka przeprowadzania eksperymentów

Przeprowadzane eksperymenty zostały usystematyzowane ze względu na algorytmy bazowe, na których sprawdzano proponowane modyfikacje.

W celu zapewnienia jednoznaczności i zwięzłości opisu eksperymentów zastosowano skróty nazw poszczególnych algorytmów według schematu nazewniczego przedstawionego w Tabeli 4.1.

**Tabela 4.1:** Konwencja nazewnicza oznaczeń algorytmów stosowana w opisach eksperymentów

		Algorytm do rozwiązania problemu optymalizacyjnego:			
		Optymalizacji Załadunku	Grupowania Danych	Wykrywania Danych Odstających	Dopasowania Sekwencji
Modyfikacja algorytmu genetycznego	brak	GAOZ	GAGR	GADO	GAMSA
	$Epi_x, x = \{1, 2, 3\}$	GAOZ Epi_1	GAGR Epi_1	GADO Epi_1	GAMSA Epi_1
		GAOZ Epi_2	GAGR Epi_2	GADO Epi_2	GAMSA Epi_2
		GAOZ Epi_3	GAGR Epi_3	GADO Epi_3	GAMSA Epi_3

Algorytmami bazowymi (odniesienia) dla eksperymentów są algorytmy: GAOZ, GAGR, GADO oraz GAMSA.

Dla każdego wybranego do eksperymentów algorytmu genetycznego (patrz Rozdział 2) umożliwiającego rozwiązanie określonych problemów optymalizacyjnych (patrz Rozdział 2) zastosowano kolejno zaproponowane w pracy modyfikacje inspirowane procesami epigenetycznymi (por. Rozdział 3). Wpływ proponowanych modyfikacji na efektywność działania bazowych algorytmów genetycznych weryfikowany był niezależnie, co oznacza, że modyfikacje wprowadzane były do GAB po kolei, a otrzymane w wyniku wprowadzenia danej modyfikacji porównywane były z wynikami otrzymanymi dla algorytmów bazowych. Dla każdego z algorytmów, w pierwszej części poświęconego mu podrozdziału znajdują się informacje o zbiorach danych użytych do

eksperymentów (zbiory testowe), a w kolejnym przedstawione są parametry podstawowe dla algorytmu genetycznego. Przedstawione parametry podstawowe algorytmu genetycznego, zostały użyte zarówno w przypadku *GAB* jak i algorytmów z wprowadzonymi kolejno modyfikacjami *Epi\_x*.

W kolejnych podrozdziałach prezentowane są wyniki badań nad oceną skuteczności proponowanych w pracy modyfikacji *Epi\_x*. Dla każdej proponowanej modyfikacji jako pierwsze przedstawione zostały wyniki badań nad empirycznym doбором optymalnego prawdopodobieństwa wystąpienia danej modyfikacji.

W przeprowadzonych badaniach dokonano wyznaczenia mediany liczby generacji dla 9 iteracji poszczególnych algorytmów genetycznych z różnymi prawdopodobieństwami wystąpienia proponowanych modyfikacji *Epi\_x*, oraz dla *GAB*. Badania miały na celu sprawdzenie dla jakiego prawdopodobieństwa wystąpienia proponowanych modyfikacji uzyskuje się największą redukcję liczby generacji potrzebnych do uzyskania oczekiwanego wyniku.

W drugim etapie badań dokonano porównania zmiany wartości funkcji przystosowania najlepszego osobnika w kolejnych generacjach dla danego algorytmu z uwzględnioną wybraną modyfikacją *Epi\_x* w stosunku do *GAB*. Ten etap badań miał na celu sprawdzenie w jakim stopniu wprowadzenie danej modyfikacji *Epi* wpłynie na dynamikę zmiany wartości funkcji przystosowania najlepszego osobnika w stosunku do dynamiki zmiany wartości funkcji przystosowania dla najlepszego osobnika w *GAB*. Do porównania wartości funkcji przystosowania zostały wybrane osobniki, dla których liczba generacji potrzebna do uzyskania oczekiwanego wyniku stanowiła wartość mediany dla wszystkich iteracji danego algorytmu.

W trzecim etapie badań przedstawione zostaje porównanie czasu potrzebnego do przeprowadzenia wszystkich operacji w pojedynczej generacji algorytmu z daną modyfikacją *Epi* w stosunku do *GAB*. Badania miały na celu sprawdzenie wpływu wprowadzenia proponowanych modyfikacji na czas działania danego algorytmu genetycznego. W badaniach sprawdzano, czy mimo redukcji liczby generacji potrzebnych do uzyskania oczekiwanego rezultatu przez algorytm genetyczny z zastosowaną jedną z proponowanych modyfikacji nie wydłuży się całkowity czas potrzebny na uzyskanie wyniku w stosunku do *GAB*. Pomiar czasów wykonano dla 100 powtórzeń operacji dla pojedynczej generacji.

Na wykresach prezentujących medianę liczby generacji potrzebną do uzyskania oczekiwanego wyniku za pomocą algorytmów *GAB* i *GAB Epi\_x* (podrozdziały z doбором optymalnego prawdopodobieństwa), wartość liczby generacji została przedstawiona procentowo, gdzie za 100% uznaje się medianę liczby generacji w algorytmie *GAB*. Dla algorytmów *Epi\_x* wartości

poniżej 100% oznaczają redukcję liczby generacji w odniesieniu do algorytmu *GAB*, powyżej zwiększenie liczby generacji (przykładowo Rysunek 4.1).

W celu łatwiejszego porównania wyników dla poszczególnych zbiorów testowych, zmiany wartości funkcji przystosowania w kolejnych generacjach *GAB* i *GAB Epi\_x* przedstawione na wykresach w podrozdziałach dotyczących porównania *GAB* z ich odpowiednikami z dołożonymi operacjami *Epi\_x* zostały znormalizowane do zakresu  $[0,100]$  (przykładowo Rysunek 4.2).

### 4.3 Badanie algorytmu optymalizacji załadunku

Pierwszym algorytmem genetycznym na którym sprawdzono, czy zastosowanie w nim proponowanych modyfikacji epigenetycznych poprawi jego działanie (poprzez redukcję liczby generacji potrzebnych do uzyskania optymalnego rozwiązania) był algorytm do rozwiązywania problemu optymalnego załadunku (*GAOZ*) (omówiony w Podrozdziale 2.1).

#### 4.3.1 Dane wykorzystane w eksperymentach

W eksperymentach dla *GAOZ* wykorzystano pięć zbiorów danych, różniących się możliwą liczbą towarów do przewiezienia i maksymalną dopuszczalną masą przewożonego towaru w kontenerze. Zbiory danych są zbiorami sztucznymi, utworzonymi na potrzeby eksperymentów. Opis zbiorów danych dla algorytmu optymalizacji załadunku przedstawiono w Tabeli 4.2.

*Tabela 4.2: Dane wykorzystane w eksperymentach z GAOZ*

Zbiór	Liczba towarów do przewiezienia	Maksymalny dozwolony ciężar ładunku
zbiór 1	10	20
zbiór 2	20	40
zbiór 3	40	80
zbiór 4	80	160
zbiór 5	160	320

Każdy towar do przewiezienia (w danym zbiorze danych) posiadał dodatkowo takie wartości jak waga własna, oraz zysk za dostarczenie.

#### 4.3.2 Podstawowe parametry algorytmu

Dla algorytmu *GAOZ* ustalone zostały parametry, takie jak prawdopodobieństwo krzyżowania, mutacji oraz warunek stopu. Parametry ustalone dla algorytmu optymalizacji załadunku są standardowymi, zalecanymi w literaturze wartościami. Parametry algorytmu genetycznego do optymalizacji załadunku zostały przedstawione w Tabeli 4.3.



Ustalone parametry dla GA do optymalizacji załadunku zostały użyte zarówno w przypadku algorytmu bazowego, jak i algorytmów zmodyfikowanych o operacje imitujące procesy epigenetyczne.

**Tabela 4.3:** Parametry algorytmu genetycznego wykorzystane w eksperymentach

Parametr	Wartość
prawdopodobieństwo krzyżowania	90%
prawdopodobieństwo mutacji	1%
metoda selekcji osobników	turniejowa
warunek stopu	warunkowana długością zbioru liczba generacji bez poprawy wartości funkcji przystosowania

W kolejnych podrozdziałach przedstawiono wyniki eksperymentów przeprowadzonych na *GAOZ* z zastosowanymi operacjami *Epi\_x*. Otrzymane wyniki, dla zmodyfikowanych wersji algorytmu genetycznego, zostały też porównane z wynikami uzyskanymi za pomocą *GAOZ*.

### 4.3.3 Operacja *Epi\_1*

W niniejszym podrozdziale przedstawione zostały wyniki eksperymentów przeprowadzonych w celu sprawdzenia, czy zastosowanie w *GAOZ* operacji naśladującej epigenetyczny proces dziedziczenia za pomocą prionu poprawi efektywność działania tego algorytmu (*GAOZ Epi\_1*).

W pierwszym kroku sprawdzone zostało, czy wystąpienie operacji *Epi\_1* wpływa na liczbę iteracji potrzebnych do uzyskania optymalnego wyniku. W Tabeli 4.4 przedstawiono liczbę generacji potrzebną do uzyskania optymalnego rozwiązania w zależności od prawdopodobieństwa wystąpienia proponowanej operacji dla poszczególnych zbiorów testowych.

Na podstawie tabeli 4.4 można zauważyć, że najmniejsza liczba generacji potrzebna do uzyskania optymalnego rozwiązania dla większości zbiorów testowych występuje w przypadku 40% prawdopodobieństwa wystąpienia operacji *Epi\_1*. Wyjątkiem jest zbiór testowy 3, w którym minimalna liczba generacji osiągnięta została w przypadku prawdopodobieństwa modyfikacji wynoszącego 30%. W kolejnych sekcjach przedstawiono poszczególne wyniki otrzymane przy wprowadzeniu operacji *Epi\_1* do *GAOZ* w zależności od zbioru testowego.

### **Eksperymentalny dobór optymalnego prawdopodobieństwa wystąpienia operacji *Epi\_1***

Na rysunku 4.1 przedstawiono procentową wartość liczby generacji dla różnych prawdopodobieństw wystąpienia modyfikacji *Epi\_1* w *GAOZ Epi\_1* w stosunku do liczby generacji

**Tabela 4.4:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji  $Epi\_1$  (Pogrubioną kursywą zaznaczono najmniejszą liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania				
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5
0% (bez modyfikacji)	154	631	973	1355	1513
5%	178	624	968	1454	1574
10%	172	614	972	1286	1530
20%	144	578	690	1103	1588
30%	126	430	<b>641</b>	1186	1591
40%	<b>113</b>	<b>406</b>	644	<b>889</b>	<b>1404</b>
50%	131	419	658	1016	1459
60%	148	425	741	1021	1589
70%	152	437	951	1165	1598
80%	159	410	963	1302	1581
90%	166	623	967	1398	1598
100%	185	645	980	1465	1597

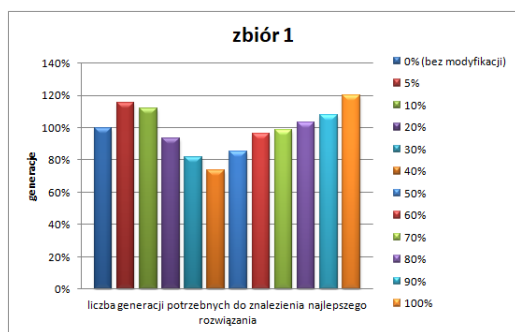
w GAOZ. Procentowe (względne) przedstawienie różnic w liczbie generacji pozwala w sposób bardziej obiektywny porównać otrzymane wyniki dla różnych zbiorów testowych.

Dla wszystkich zbiorów testowych zauważalna jest poprawa wydajności działania w przypadku  $GAOZEpi\_1$  z prawdopodobieństwem modyfikacji na poziomie 30% - 40%. Dla mniejszych wartości prawdopodobieństwa wystąpienia modyfikacji  $Epi\_1$  poprawa działania jest niezauważalna, a nawet w przypadku zbioru pierwszego i czwartego nastąpiło nieznaczne pogorszenie efektywności działania  $GAOZEpi\_1$  (wykres na rysunkach 4.1a oraz 4.1d). Największy spadek liczby generacji potrzebnych do znalezienia optymalnego rozwiązania dla problemu optymalnego załadunku uzyskano dla zbioru drugiego, trzeciego i czwartego (wykresy 4.1b, 4.1c, oraz 4.1d). W przypadku wszystkich zbiorów testowych zbyt duże prawdopodobieństwo wystąpienia operacji  $Epi\_1$  powoduje pogorszenie efektywności.

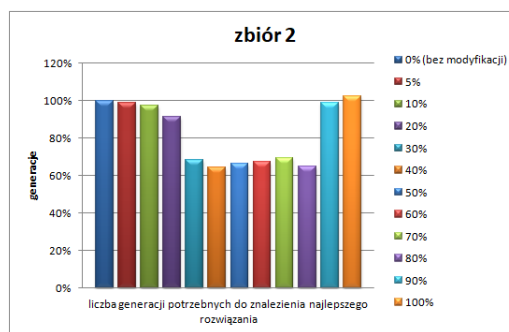
### Porównanie efektywności działania GAOZ i GAOZ $Epi\_1$

W sekcji tej zaprezentowano wyniki porównania zmiany funkcji przystosowania dla GAOZ z algorytmami GAOZ  $Epi\_1$  z optymalnym prawdopodobieństwem wystąpienia modyfikacji  $Epi\_1$  dla poszczególnych zbiorów testowych. Wyniki porównania zostały zaprezentowane na wykresach na Rysunku 4.2. Z wykresów można odczytać jak szybko wzrasta wartość funkcji przystosowania najlepszego osobnika wraz z kolejnymi generacjami GAOZ i GAOZ  $Epi\_1$ .

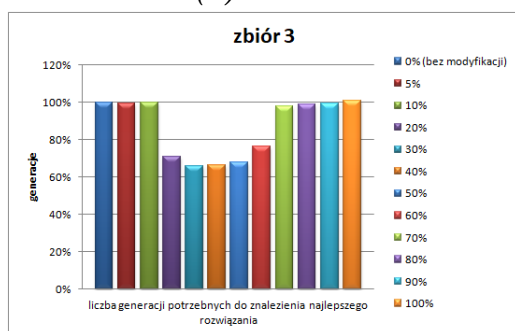
Na podstawie wykresów na rysunku 4.2 można zauważyć, że w przypadku algorytmu genetycznego z modyfikacją  $Epi\_1$  następował szybszy wzrost wartości funkcji przystosowania



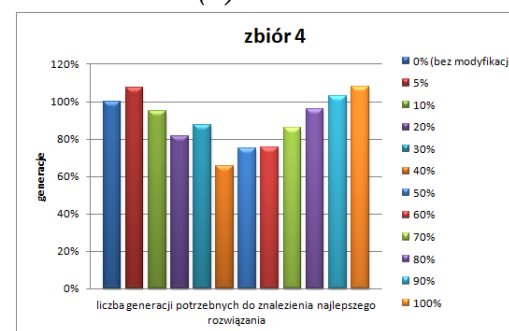
(a) zbiór 1



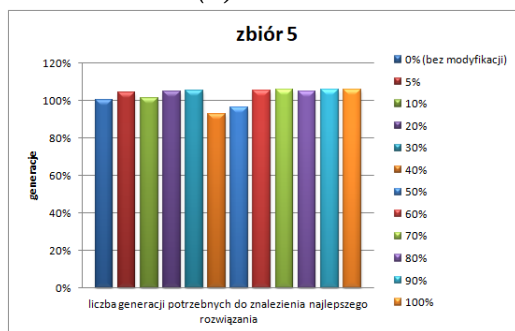
(b) zbiór 1



(c) zbiór 3



(d) zbiór 4



(e) zbiór 5

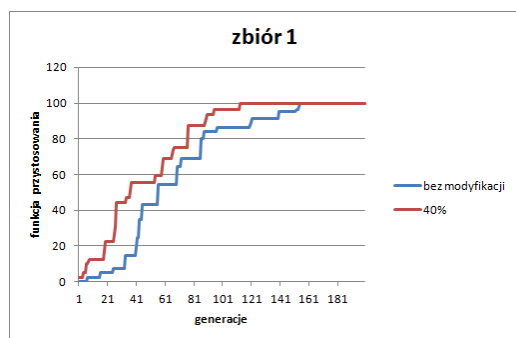
**Rysunek 4.1:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji  $Epi_1$

najlepszego osobnika w porównaniu do algorytmu bez modyfikacji. Największe różnice we wzroście wartości funkcji przystosowania dla  $GAOZEpi_1$  w stosunku do  $GAOZ$  można zauważyć dla zbiorów pierwszego i piątego (wykresy 4.2a, oraz 4.2e). Natomiast najmniejsze różnice w zmianie wartości funkcji przystosowania, można zaobserwować w przypadku zbioru 4 (wykres 4.2d).

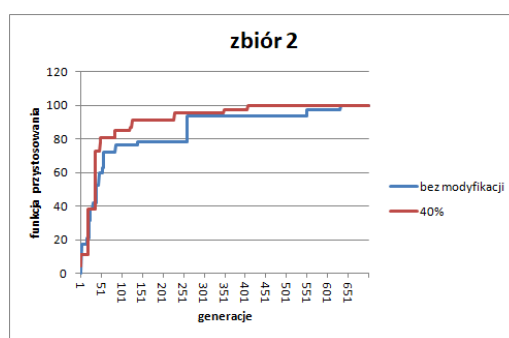
### Porównanie czasów działania algorytmów

W sekcji tej przedstawiono porównanie czasów jednej generacji  $GAOZEpi_1$  i  $GAOZ$ , wyniki czasów (średnie wartości czasu i wartość odchylenia standardowego) pokazano w Tabeli 4.5 (zmianę procentową określono w stosunku do algorytmu  $GAOZ$ ).

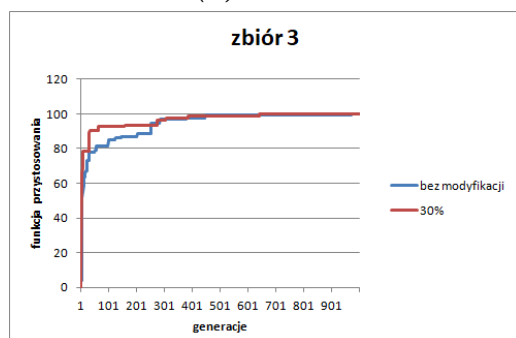
W przypadku zastosowania operacji  $Epi_1$  w algorytmie  $GAOZ$  zaobserwowano wzrost wartości czasu potrzebnego do wykonania jednej generacji algorytmu. Minimalny wzrost czasu



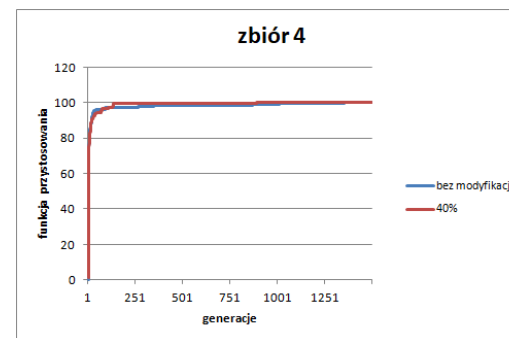
(a) zbiór 1



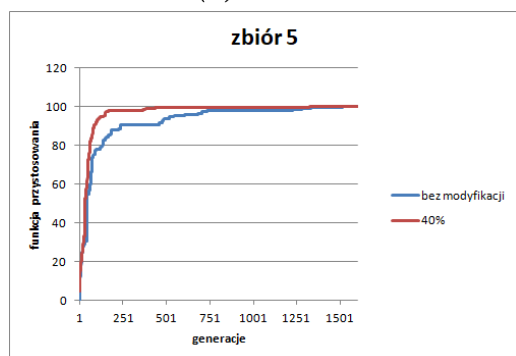
(b) zbiór 2



(c) zbiór 3



(d) zbiór 4



(e) zbiór 5

**Rysunek 4.2:** Zmiany wartości funkcji przystosowania dla generacji w przypadku GAOZ i GAOZ Epi\_1

**Tabela 4.5:** Porównanie czasów jednej generacji dla GAOZEpi\_1 i GAOZ

zbiory	czas [s]				zmiana %
	GAOZEpi_1		GAOZ		
	średnia	sd	średnia	sd	
zbiór 1	0,064	0,006	0,063	0,012	1,04%
zbiór 2	0,080	0,011	0,072	0,010	10,01%
zbiór 3	0,110	0,015	0,098	0,021	11,10%
zbiór 4	0,181	0,016	0,169	0,022	6,67%
zbiór 5	0,339	0,115	0,318	0,115	6,39%

zaobserwowano w przypadku zbioru 1, wynosił on 1,04%. Największy wzrost czasu zaobserwowano w przypadku zbioru 3, czas potrzebny do wykonania operacji dla jednej generacji wzrósł średnio o 11,10%. W przypadku algorytmu do optymalizacji załadunku średni wzrost czasu (dla wszystkich zbiorów) przeprowadzenia operacji dla pojedynczej generacji wzrósł o średnio

w przybliżeniu 7%.

#### 4.3.4 Modyfikacja *Epi\_2*

W podrozdziale tym zaprezentowane zostały wyniki eksperymentów mające na celu sprawdzenie zaproponowanej w rozprawie modyfikacji w algorytmach genetycznych naśladującej proces epigenetyczny metylacji cytozyny (*Epi\_2*). Podobnie jak w przypadku wcześniejszej modyfikacji *Epi\_1*, przeprowadzone eksperymenty miały na celu sprawdzenie, czy zastosowanie modyfikacji *Epi\_2* wpłynie na poprawę efektywności działania algorytmów genetycznych poprzez redukcję liczby generacji potrzebnych do uzyskania optymalnego wyniku.

W pierwszym etapie eksperymentów sprawdzono, czy zastosowanie modyfikacji *Epi\_2* w użytym do eksperymentów algorytmie *GAOZ* (*GAOZ Epi\_2*) w ogóle poprawi działanie, oraz jakie będzie optymalne prawdopodobieństwo wystąpienia proponowanej modyfikacji. W Tabeli 4.6 przedstawiono liczbę generacji potrzebną do znalezienia optymalnego załadunku przez *GAOZ*, oraz *GAOZ Epi\_2* z różnym prawdopodobieństwem wystąpienia modyfikacji *Epi\_2*.

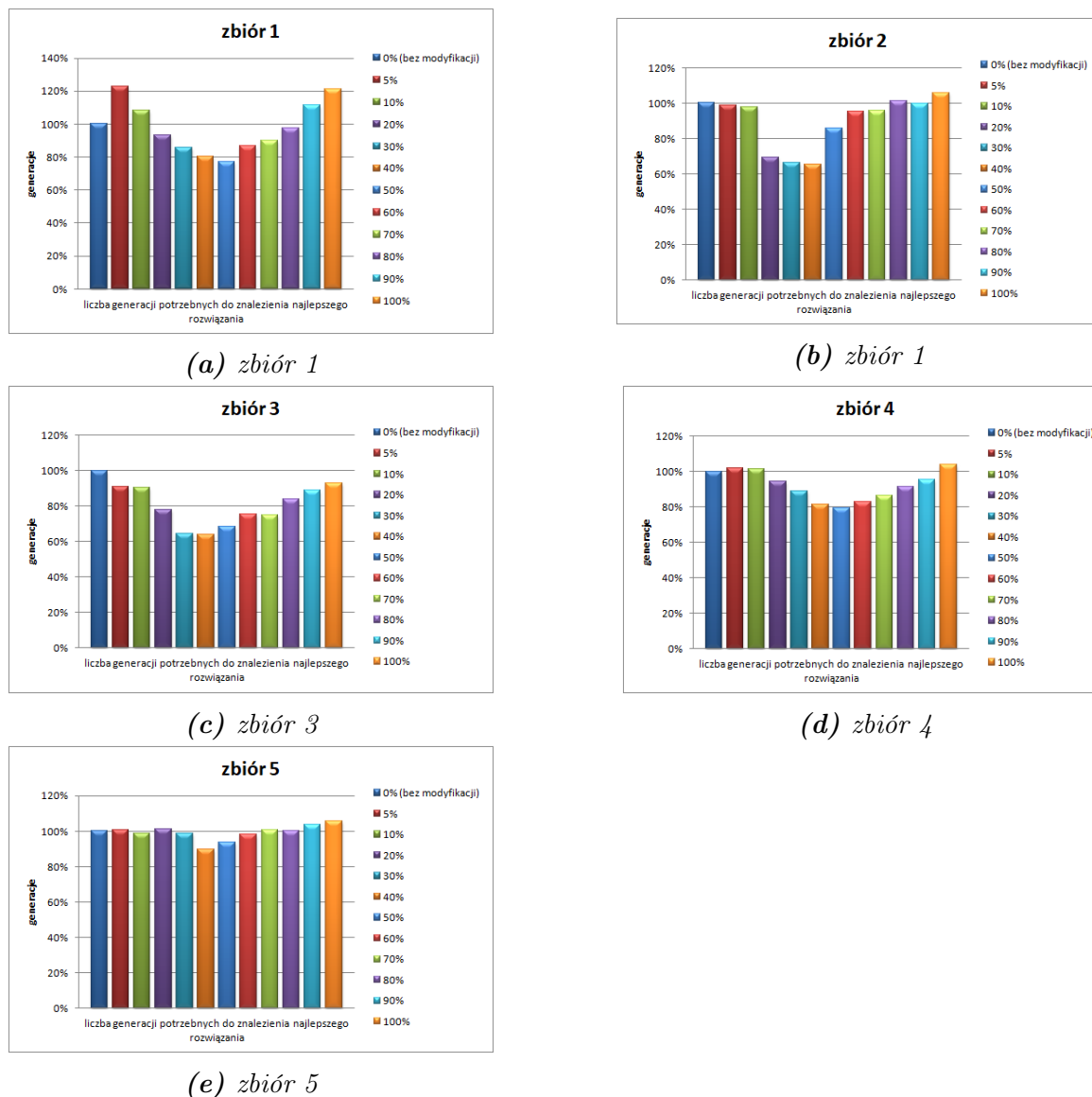
**Tabela 4.6:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* (Pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania				
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5
0% (bez modyfikacji)	154	631	973	1355	1513
5%	189	623	885	1378	1523
10%	167	617	877	1369	1492
20%	144	436	760	1278	1533
30%	132	419	627	1202	1490
40%	124	<b>413</b>	<b>620</b>	1101	<b>1356</b>
50%	<b>119</b>	541	665	<b>1077</b>	1415
60%	134	602	736	1123	1489
70%	139	604	731	1169	1523
80%	150	640	816	1235	1519
90%	172	630	867	1294	1565
100%	187	666	906	1403	1598

Na podstawie tabeli 4.6 można zaobserwować, że przy zastosowaniu odpowiedniego prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* można uzyskać poprawę działania algorytmu genetycznego, poprzez redukcję liczby generacji potrzebnych do uzyskania optymalnego wyniku.

## Eksperymentalny dobór optymalnego prawdopodobieństwa wystąpienia modyfikacji $Epi\_2$

Na rysunku 4.3 przedstawiono procentową wartość liczby generacji dla różnych prawdopodobieństw wystąpienia modyfikacji  $Epi\_2$  w  $GAOZ Epi\_2$  w stosunku do  $GAOZ$ .



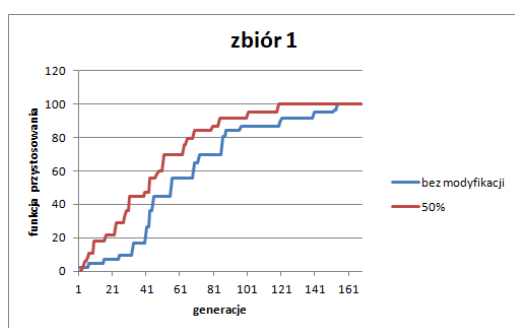
**Rysunek 4.3:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji  $Epi\_2$

Na podstawie wykresów na rysunku 4.3 można zauważyć, że największą redukcję liczby generacji uzyskano w przypadku zbioru 2, dla prawdopodobieństwa wystąpienia modyfikacji  $Epi\_2$  wynoszącym 40% (wykres 4.3b). Najmniejszą różnicę w liczbie generacji pomiędzy  $GAOZ$ , a algorytmem  $GAOZ Epi\_2$  można zaobserwować dla zbioru 5 (wykres 4.1e). Przy większym prawdopodobieństwie wystąpienia modyfikacji  $Epi\_2$  w  $GAOZ Epi\_2$  (prawdopodobieństwo powyżej 60%) można zaobserwować zwiększenie liczby generacji potrzebnych do

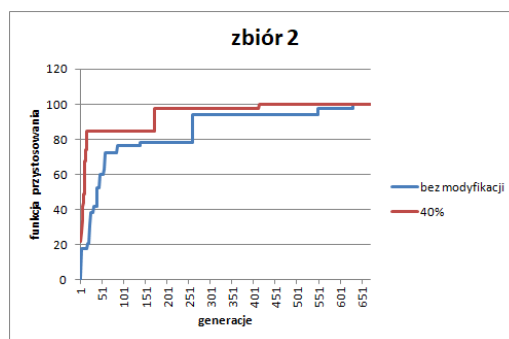
znalezienia optymalnego rozwiązania przez zmodyfikowany algorytm, a więc pogorszenie efektywności działania algorytmu.

### Porównanie efektywności działania GAOZz GAOZEpi\_2

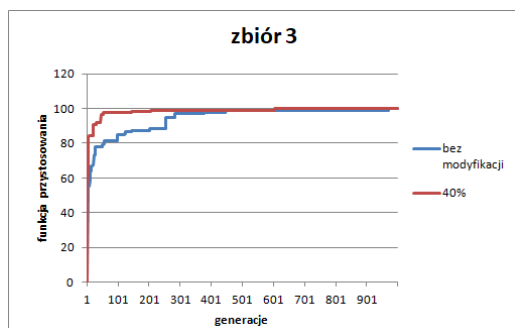
Dla optymalnych wartości prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* w GAOZ *Epi\_2* dokonano porównania zmiany wartości funkcji przystosowania w kolejnych generacjach względem zmiany wartości funkcji przystosowania w GAOZ. Porównanie to ma celu sprawdzenie, czy zmodyfikowany algorytm powoduje wzrost prędkości zmian wartości funkcji przystosowania wraz z kolejnymi generacjami algorytmu. Wyniki porównania funkcji przystosowania pokazano na wykresach na Rysunku 4.4.



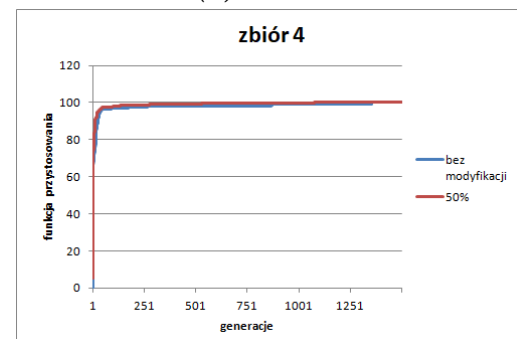
(a) zbiór 1



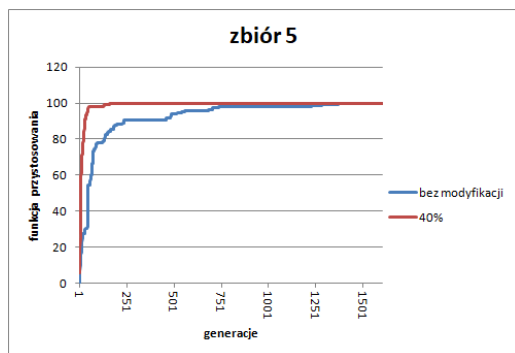
(b) zbiór 1



(c) zbiór 3



(d) zbiór 4



(e) zbiór 5

**Rysunek 4.4:** Zmiana wartości funkcji przystosowania w kolejnych generacjach GAOZEpi\_2 i GAOZ, dla poszczególnych zbiorów danych

Na podstawie wykresów na Rysunku 4.4 można zaobserwować większe tempo zmiany

wartości funkcji przystosowania w *GAOZ Epi\_2* (w porównaniu do *GAOZ*) w prawie wszystkich testowych zbiorach danych. Najmniejszą różnicę w tempie zmian wartości funkcji przystosowania pomiędzy *GAOZ* a *GAOZ Epi\_2* uzyskano w przypadku zbioru 4 (wykres 4.2d).

### Porównanie czasów działania pojedynczej generacji *GAOZ* i *GAOZ Epi\_2*

W Tabeli 4.7 przedstawiono średnią wartość czasów i wartość odchylenia standardowego dla wszystkich operacji algorytmu genetycznego dla pojedynczej generacji algorytmu. Dodatkowo, w ostatniej kolumnie przedstawiono zmianę procentową czasu pojedynczej generacji w *GAOZEpi\_2* względem *GAOZ* (o ile procentowo wydłużył się czas pojedynczej generacji).

**Tabela 4.7:** Porównanie czasów pojedynczej generacji w *GAOZ* względem *GAOZEpi\_2*

zbiór	czas [s]				zmiana %
	<i>GAOZ Epi_2</i>		<i>GAOZ</i>		
	średnia	sd	średnia	sd	
zbiór 1	0,072	0,031	0,063	0,012	12,70%
zbiór 2	0,081	0,005	0,072	0,010	11,35%
zbiór 3	0,109	0,010	0,098	0,021	10,55%
zbiór 4	0,182	0,017	0,169	0,022	7,05%
zbiór 5	0,345	0,061	0,318	0,115	8,05%

Na podstawie tabeli 4.7 można zaobserwować, że nastąpiło wydłużenie czasu potrzebnego na przeprowadzenie wszystkich operacji algorytmu genetycznego dla pojedynczej generacji w przypadku *GAOZ Epi\_2* względem *GAOZ*. Procentowa wartość wydłużenia czasu pojedynczej generacji wahała się od 7,05% dla zbioru 4, do 12,70% dla zbioru 1.

W przypadku zastosowania modyfikacji *Epi\_2* w algorytmie *GAOZ*, na podstawie przeprowadzonych eksperymentów można zauważyć poprawę efektywności działania algorytmu. Poprawa efektywności ma miejsce poprzez redukcję liczby generacji potrzebnych do uzyskania optymalnego rezultatu. Wprowadzenie modyfikacji powoduje jednak wydłużenie czasu potrzebnego do przeprowadzenia wszystkich operacji w algorytmie genetycznym dla pojedynczej generacji.

#### 4.3.5 Operacja *Epi\_3*

W podrozdziale tym przedstawione zostały wyniki eksperymentów nad zastosowaniem operacji *Epi\_3* w algorytmie *GAOZ* (*GAOZ Epi\_3*). Przeprowadzone eksperymenty miały na celu sprawdzenie, czy zastosowanie operacji *Epi\_3* w algorytmie genetycznym do optymalizacji załadunku pozwoli zredukować liczbę generacji potrzebną do uzyskania oczekiwanego wyniku.



Pierwszym etapem przeprowadzonych eksperymentów, było sprawdzenie czy wprowadzenie operacji  $Epi\_3$  do algorytmu optymalizacji załadunku spowoduje w ogóle redukcję liczby generacji potrzebnych do uzyskania prawidłowego wyniku, oraz jakie jest optymalne prawdopodobieństwo wystąpienia operacji  $Epi\_3$ . Medianę liczby generacji w  $GAOZ$ , oraz  $GAOZ Epi\_3$  z różnym prawdopodobieństwem wystąpienia modyfikacji  $Epi\_3$  przedstawiono w Tabeli 4.8.

**Tabela 4.8:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji  $Epi\_3$  (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

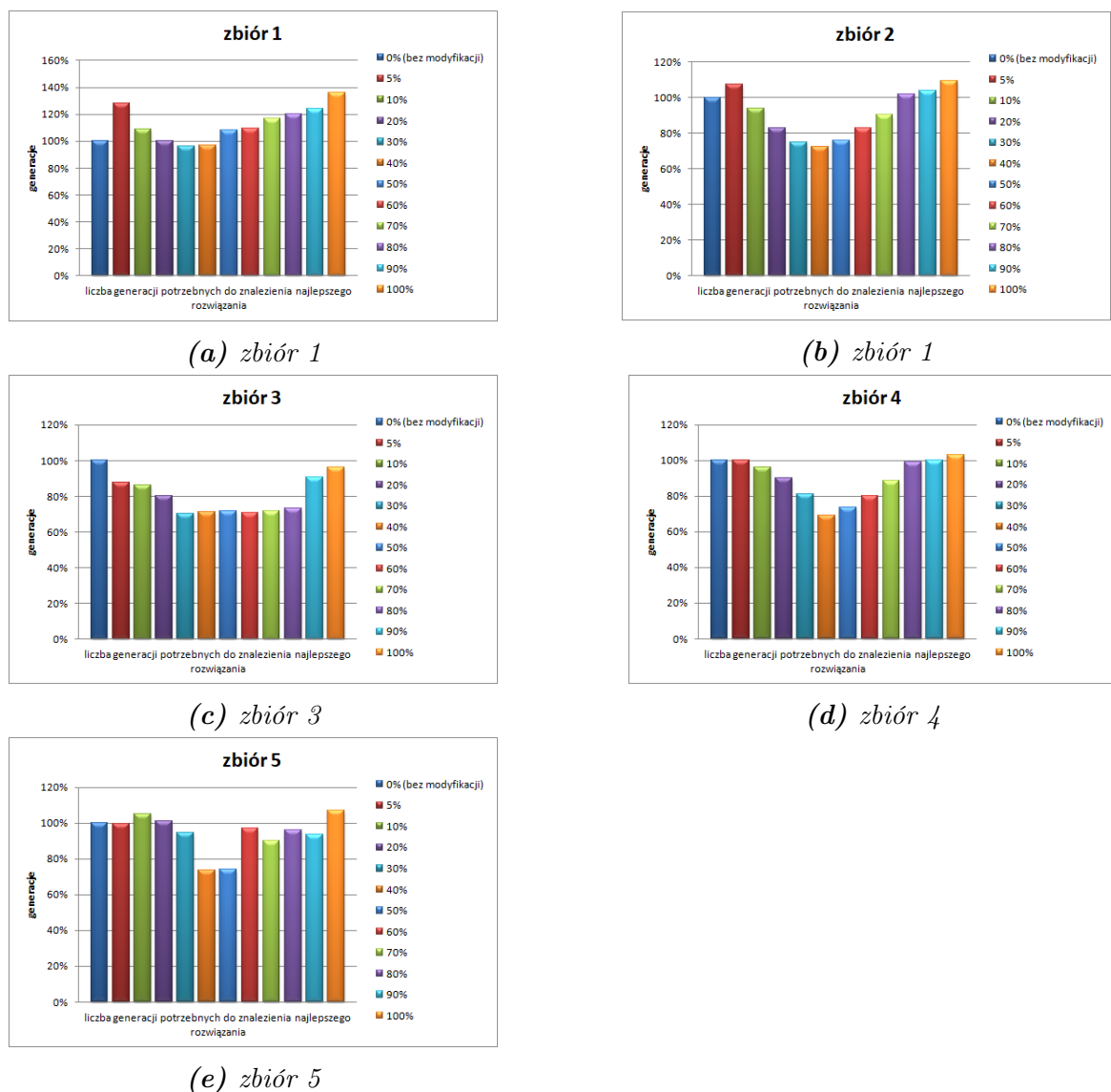
Prawdopodobieństwo wystąpienia modyfikacji	Liczba generacji potrzebnych do znalezienia najlepszego rozwiązania				
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5
0% (bez modyfikacji)	154	631	973	1355	1513
5%	197	678	853	1356	1503
10%	168	592	841	1302	1589
20%	154	521	780	1222	1532
30%	<b>148</b>	472	<b>682</b>	1102	1432
40%	149	<b>455</b>	692	<b>938</b>	<b>1115</b>
50%	167	479	696	998	1123
60%	169	523	689	1087	1471
70%	180	571	699	1201	1365
80%	185	643	710	1345	1452
90%	191	654	880	1358	1415
100%	209	689	934	1398	1623

Na podstawie Tabeli 4.8 można zauważyć, że przy prawdopodobieństwie wystąpienia modyfikacji  $Epi\_3$  poniżej 60% uzyskuje się redukcję liczby generacji potrzebną do uzyskania optymalnego załadunku towarów. Najlepsze rezultaty, poprzez największą redukcję liczby generacji w  $GAOZ Epi\_3$  w stosunku do  $GAOZ$  uzyskuje się dla prawdopodobieństwa wystąpienia modyfikacji  $Epi\_3$  z zakresu 30-40%.

### **Eksperymentalny dobór optymalnego prawdopodobieństwa wystąpienia modyfikacji $Epi\_3$**

Dla każdego ze zbiorów testowych dokonano procentowego porównania liczby generacji potrzebnych do uzyskania optymalnego wyniku działania  $GAOZ$  i  $GAOZ Epi\_3$ . Porównanie to zostało przedstawione na wykresach na Rysunku 4.5.

Na wykresach na Rysunku 4.5 pokazano procentową wartość liczby generacji, dla danego prawdopodobieństwa wystąpienia modyfikacji  $Epi\_3$  (w  $GAOZ Epi\_3$ ) w stosunku do  $GAOZ$ . Spośród wykorzystanych zbiorów danych testowych największą redukcję liczby generacji osiągnięto w przypadku zbiorów 3, 4 i 5 (wykresy 4.5c, 4.5d, oraz 4.5e). Dla tych trzech zbiorów największą redukcję liczby generacji osiągnięto przy zastosowaniu modyfikacji  $Epi\_3$  z



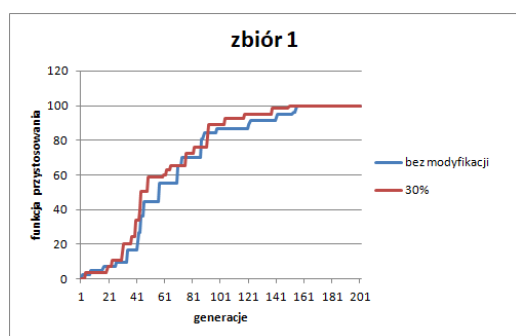
**Rysunek 4.5:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji  $Epi\_3$

prawdopodobieństwem jej wystąpienia na poziomie 40%. Najmniejszą redukcję liczby generacji w  $GAOZ\ Epi\_3$  można zauważyć w 1 zbiorze danych (wykres 4.5a). Dla zbioru 1 można też zaobserwować w największym stopniu, że zbyt duże prawdopodobieństwo wystąpienia operacji  $Epi\_3$  w  $GAOZEpi\_3$  może doprowadzić do zwiększenia liczby generacji w stosunku do  $GAOZ$ . Ogólnie dla wszystkich zbiorów testowych zbyt duże prawdopodobieństwo wystąpienia operacji  $Epi\_3$  doprowadza do pogorszenia efektywności działania tego algorytmu, poprzez zwiększenie liczby generacji potrzebnych do uzyskania optymalnego wyniku.

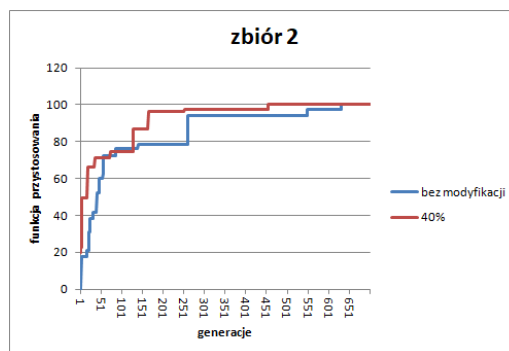
### Porównanie efektywności działania $GAOZ$ i $GAOZEpi\_3$

Dla optymalnego prawdopodobieństwa wystąpienia operacji  $Epi\_3$  w  $GAOZ$  dla danego zbioru testowego dokonano porównania zmiany wartości funkcji przystosowania wraz z kolejnymi generacjami dla algorytmu zmodyfikowanego i bazowego. Wyniki tego porównania zostały

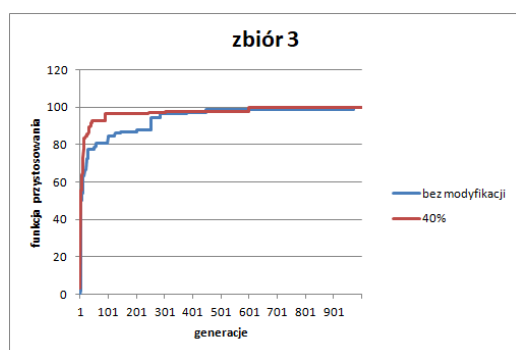
zaprezentowane na wykresach na Rysunku 4.6.



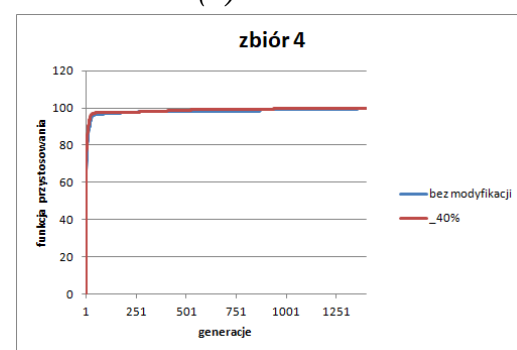
(a) zbiór 1



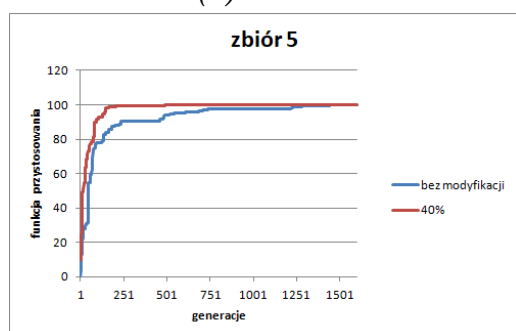
(b) zbiór 2



(c) zbiór 3



(d) zbiór 4



(e) zbiór 5

**Rysunek 4.6:** Zmiana wartości funkcji przystosowania dla kolejnych generacji algorytmu genetycznego, w zależności od zastosowania operacji  $Epi\_3$  i bez niej

Na podstawie wykresów na rysunku 4.6 można zauważyć, że jedynie w przypadku zbioru 4 różnice w tempie zmian wartości funkcji przystosowania w  $GAOZ$  i w  $GAOZ Epi\_3$  z optymalnym prawdopodobieństwem wystąpienia operacji  $Epi\_3$  są praktycznie niezauważalne (wykres 4.6d). Dla pozostałych zbiorów testowych można zauważyć większą dynamikę zmiany wartości funkcji przystosowania w  $GAOZ Epi\_3$  względem  $GAOZ$ . Największe różnice w tempie zmiany wartości funkcji przystosowania najlepszego osobnika można zaobserwować w przypadku zbioru 2 (wykres 4.6b).

## Porównanie czasów

W Tabeli 4.9 przedstawiono średni czas i odchylenie standardowe wykonania operacji algorytmu genetycznego dla pojedynczej generacji w *GAOZ Epi\_3* i w *GAOZ*. Dodatkowo w ostatniej kolumnie przedstawiono procentową wartość wydłużenia czasu w algorytmie z dołożoną operacją *Epi\_3* względem algorytmu bez modyfikacji.

**Tabela 4.9:** Porównanie czasów pojedynczej generacji *GAOZ* i *GAOZ Epi\_3*

zbiór	czas [s]				zmiana %
	<i>GAOZ Epi_3</i>		<i>GAOZ</i>		
	średnia	sd	średnia	sd	
zbiór 1	0,077	0,036	0,063	0,012	18,17%
zbiór 2	0,075	0,053	0,072	0,010	3,53%
zbiór 3	0,112	0,064	0,098	0,021	12,65%
zbiór 4	0,180	0,066	0,169	0,022	6,16%
zbiór 5	0,372	0,179	0,318	0,115	14,76%

Na podstawie Tabeli 4.9 można zaobserwować wydłużenie się czasu potrzebnego do przeprowadzenia wszystkich operacji w pojedynczej generacji *GAOZ Epi\_3* względem *GAOZ*. Średnio czas operacji dla pojedynczej generacji wzrósł o 11,05%. Największy zaobserwowano w przypadku zbioru 1 (wzrost o 18,17%), najmniejszy w przypadku zbioru 2 (wzrost o 3,53%),

### 4.3.6 Podsumowanie wyników

W niniejszym podrozdziale zaprezentowane zostały wyniki eksperymentów nad oceną skuteczności zastosowania trzech proponowanych w pracy operacji naśladujących procesy epigenetyczne w algorytmie genetycznym do optymalizacji załadunku. W przeprowadzonych badaniach sprawdzono optymalne prawdopodobieństwo wystąpienia proponowanych dodatkowych operacji. Optymalne prawdopodobieństwo wystąpienia proponowanych w pracy operacji w algorytmie do optymalizacji załadunku przedstawiono w Tabeli 4.10.

**Tabela 4.10:** Optymalne prawdopodobieństwo wystąpienia operacji *Epi* w algorytmie *GAOZ*

Operacja	Optymalne prawdopodobieństwo
<i>Epi_1</i>	40%
<i>Epi_2</i>	40%
<i>Epi_3</i>	30%

Za optymalne prawdopodobieństwo wystąpienia operacji proponowanych niniejszej pracy uznano takie, przy którym nastąpiła największa redukcja liczby generacji potrzebna do uzyskania oczekiwanego rezultatu działania algorytmu *GAOZ*. Na podstawie Tabeli 4.10, można

zauważyć, że optymalne prawdopodobieństwo wystąpienia operacji  $Epi\_1$  i  $Epi\_2$  w algorytmie GAOZ wynosi 40%. Dla operacji  $Epi\_3$  optymalne prawdopodobieństwo wystąpienia wynosi 30%.

W Tabeli 4.11 przedstawiono procentową wartość średniej redukcji liczby generacji potrzebnych do uzyskania optymalnego wyniku w algorytmie z zastosowaniem operacji  $Epi$ , względem algorytmu GAOZ. W Tabeli 4.11 przedstawiono również o ile procent zastosowanie proponowanych operacji w algorytmie genetycznym wydłużyło czas pojedynczej generacji algorytmu genetycznego, względem algorytmu bez dodatkowych operacji.

**Tabela 4.11:** Procentowa wartość średniej redukcji liczby generacji oraz procentowe wydłużenie czasu pojedynczej generacji GAOZ  $Epi\_x$ , względem GAOZ

Proponowana operacja	średnia redukcja liczby generacji	średnie % wydłużenie czasu pojedynczej generacji
$Epi\_1$	↓ 29%	↑ 7,00%
$Epi\_2$	↓ 25%	↑ 9,94%
$Epi\_3$	↓ 24%	↑ 11,05%

Na podstawie Tabeli 4.11, zastosowanie proponowanych w pracy dodatkowych operacji bazujących na procesach epigenetycznych w algorytmie GAOZ poprawia działanie tegoż algorytmu poprzez redukcję liczby generacji potrzebnych do uzyskania oczekiwanego wyniku. Największą redukcję liczby generacji uzyskano przy zastosowaniu operacji  $Epi\_1$ , osiągając średnią redukcję rzędu 29%. Zastosowanie dodatkowych operacji wiąże się też z wydłużeniem czasu potrzebnego na przeprowadzenie wszystkich operacji algorytmu genetycznego w pojedynczej generacji algorytmu. Najmniejszy wzrost czasu zaobserwowano w przypadku zastosowania operacji  $Epi\_1$  (Tabela 4.11), wynosił on 7,00%. Największy zaś zaobserwowano w przypadku zastosowania operacji  $Epi\_3$ , wzrost wynosił 11,05%.

Na podstawie przeprowadzonych eksperymentów w GAOZ najlepsze rezultaty uzyskano przy zastosowaniu operacji  $Epi\_1$ . Uzyskano największą redukcję liczby generacji, przy najmniejszym wzroście czasu operacji pojedynczej generacji.

## 4.4 Badanie algorytmu grupowania danych

Kolejnym algorytmem, dla którego sprawdzono czy zastosowanie proponowanych w pracy dodatkowych operacji bazujących na procesach epigenetycznych ( $Epi$ ) poprawi działanie tego algorytmu, był algorytm genetyczny do grupowania danych (GAGR). Wykorzystany algorytm jest autorskim algorytmem, dlatego w podrozdziale tym zamieszczono dodatkowo wyniki badań nad oceną skuteczności działania samego algorytmu. W niniejszym podrozdziale podob-

nie jak w poprzednim przedstawiono wyniki eksperymentów nad oceną skuteczności działania proponowanych w pracy operacji wzorowanych na procesach epigenetycznych.

#### 4.4.1 Dane użyte w eksperymentach

W przeprowadzonych eksperymentach, dla algorytmu *GAGR* posłużono się 6 zbiorami danych. Spośród wykorzystanych zbiorów zbiór 2 jest zbiorem własnym wygenerowanym przez autora, pozostałe zbiory są zbiorami dostępnymi w literaturze służącymi ocenie metod grupowania danych. Charakterystykę zbiorów użytych w eksperymentach zaprezentowano w Tabeli 4.12.

*Tabela 4.12: Zbiory eksperymentalne*

zbiór	nazwa zbioru	liczba elementów	wymiary	liczba grup
zbiór 1	Mouse [93]	490	2	3
zbiór 2	Zbiór własny	5000	3	3
zbiór 3	Irys [56]	150	3	3
zbiór 4	S-sets [60]	5000	2	15
zbiór 5	Gaussian clusters dataset [59]	2048	2	2
zbiór 6	Unbalance [59]	6500	2	8

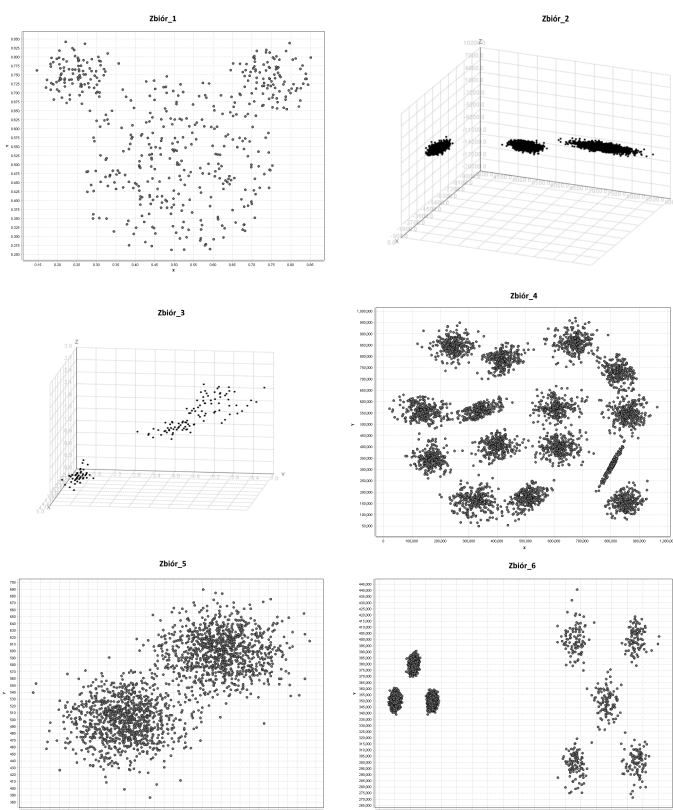
Wykorzystane w eksperymentach zbiory różnią się zarówno licznością zbioru, jak i liczbą grup, i wymiarów. Graficzną prezentację wykorzystanych w eksperymentach zbiorów przedstawiono na Rysunku 4.7.

#### 4.4.2 Parametry algorytmu

W niniejszej sekcji przedstawione zostały parametry jakie zostały ustawione dla algorytmu *GAGR*. Parametry *GAGR* dla przeprowadzonych eksperymentów przedstawiono w Tabeli 4.13.

*Tabela 4.13: Parametry algorytmu genetycznego wykorzystane w eksperymentach*

Parametr	Wartość
liczba osobników w populacji	10% liczby elementów z zbiorze danych
prawdopodobieństwo krzyżowania punktowego	40%
prawdopodobieństwo krzyżowania dwu punktowego	40%
prawdopodobieństwo mutacji	5%
metoda selekcji osobników	turniejowa
warunek stopu	warunkowana długością zbioru liczba generacji bez poprawy wartości funkcji przystosowania



**Rysunek 4.7:** Wizualizacja zbiorów użytych w eksperymentach

Przedstawione parametry *GAGR* (patrz Tabela 4.13) zostały wykorzystane zarówno w przypadku wykorzystania algorytmu bazowego (*GAGR*) jak i algorytmów z wprowadzonymi modyfikacjami (*GAGR Epi\_x*), zaproponowanymi w niniejszej pracy, naśladującymi procesy epigenetyczne.

Dobór odpowiednich parametrów algorytmu genetycznego został dokonany eksperymentalnie w ramach wcześniejszych badań mających na celu opracowanie skutecznego algorytmu genetycznego do grupowania danych [40].

#### 4.4.3 Ocena skuteczności działania algorytmu genetycznego do grupowania danych

W związku z tym, że omówiony w niniejszym podrozdziale algorytm jest autorską modyfikacją algorytmu genetycznego w sekcji tej zaprezentowano wyniki eksperymentów potwierdzających skuteczność działania zaprezentowanego algorytmu do grupowania danych.

W Tabeli 4.14 zaprezentowano porównanie działania proponowanego *GAGR* w stosunku do popularnych metod grupowania danych. Przedstawione wyniki otrzymano dla zbiorów przedstawionych w Tabeli 4.12. Kolumny oznaczone jako T przedstawiają informację o liczbie elementów ze zbioru danych, które zostały prawidłowo przydzielone do odpowiednich grup. W kolumnach

oznaczonych jako F, zamieszczono informacje o liczbie elementów ze zbioru danych, które nie zostały prawidłowo przydzielone do odpowiedniej grupy. Dodatkowo dla algorytmu DBScan została podana liczba grup na jakie zostały podzielone dane zbiory przy użyciu tego algorytmu (algorytm DBScan jako jedyny z przedstawionych nie wymaga określenia na początku liczby grup).

**Tabela 4.14:** Porównanie skuteczności proponowanego GAGR, z popularnymi metodami grupowania danych

zbiór	GAGR		K-means		X-means		DBScan		
	T	F	T	F	T	F	liczba grup	T	F
zbiór 1	481	9	411	79	326	164	4	470	20
zbiór 2	500	0	500	0	416	84	4	500	0
zbiór 3	149	1	147	3	141	9	3	145	5
zbiór 4	238	0	237	1	179	59	3	233	5
zbiór 5	373	0	295	78	240	133	3	371	2
zbiór 6	700	0	615	85	560	140	2	640	60

Dodatkowo dla porównanych algorytmów obliczona została wartość precyzji, obliczana wzorem 4.1:

$$PPV = \frac{TP}{TP + FP} \quad (4.1)$$

Uzyskana wartość precyzji dla poszczególnych zbiorów uzyskane przez porównywane algorytmy została przedstawiona w Tabeli 4.15.

**Tabela 4.15:** Wartości precyzji dla poszczególnych zbiorów uzyskane przez porównywane metody

zbiory	Precision			
	GAGR	K-means	X-means	DBScan
zbiór 1	0,98	0,84	0,67	0,96
zbiór 2	1,00	1,00	0,83	1,00
zbiór 3	0,99	0,98	0,94	0,97
zbiór 4	1,00	1,00	0,75	0,98
zbiór 5	1,00	0,79	0,64	0,99
zbiór 6	1,00	0,88	0,80	0,91

Na podstawie zaprezentowanych wyników w Tabeli 4.14 i Tabeli 4.15 można stwierdzić wysoką skuteczność proponowanego algorytmu genetycznego do grupowania danych. Proponowany algorytm został więc również wykorzystany w badaniach nad możliwością zastosowania procesów naśladujących procesy epigenetyczne w algorytmach genetycznych.

#### 4.4.4 Operacja *Epi\_1*

W niniejszym podrozdziale przedstawione zostały wyniki eksperymentów przeprowadzonych w celu sprawdzenia, czy zastosowanie dodatkowej operacji naśladującej epigenetyczny



proces dziedziczenia za pomocą prionu poprawi efektywność  $GAGR(GAGR\ Epi\_1)$ .

W Tabeli 4.16 przedstawiono liczbę generacji potrzebną do uzyskania rozwiązania (znalezienia optymalnych grup dla danych) dla  $GAGR$ , oraz  $GAGR\ Epi\_1$  z różnym prawdopodobieństwem wystąpienia operacji  $Epi\_1$ .

**Tabela 4.16:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji  $Epi\_1$  w  $GAGR$  (Pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania					
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5	zbiór 6
0% (bez modyfikacji)	758	181	1151	270	450	614
5%	678	155	1104	268	444	615
10%	<b>580</b>	95	<b>992</b>	261	431	601
20%	616	<b>73</b>	998	255	<b>426</b>	<b>556</b>
30%	699	82	1056	246	433	572
40%	766	103	1089	<b>236</b>	439	569
50%	784	111	1187	238	448	583
60%	789	168	1233	265	463	589
70%	816	176	1239	288	479	604
80%	855	192	1289	310	478	629
90%	862	199	1355	319	488	633
100%	877	208	1378	405	487	637

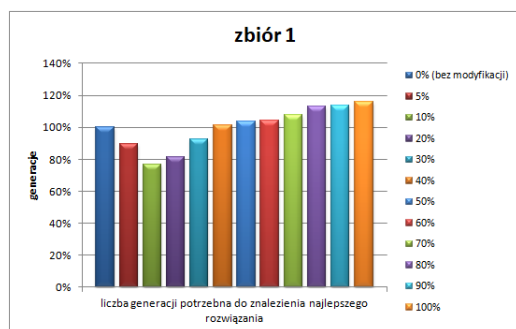
Na podstawie Tabeli 4.16 można zaobserwować, że najmniejszą liczbę generacji potrzebną do optymalnego pogrupowania danych potrzebuje algorytm z operacją epigenetyczną  $Epi\_1$  z prawdopodobieństwem wystąpienia tej modyfikacji na poziomie 10-20%. W kolejnych sekcjach zaprezentowano wyniki dla poszczególnych zbiorów testowych.

## Empiryczny dobór optymalnego prawdopodobieństwa wystąpienia modyfikacji

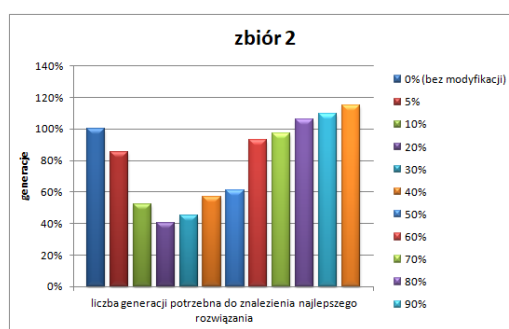
### $Epi\_1$

W przypadku  $GAGR$  który posłużył do badań nad wykorzystaniem modyfikacji  $Epi\_1$  w celu zmniejszenia liczby generacji potrzebnych do uzyskania optymalnego wyniku, w pierwszym kroku podjęto się sprawdzenia optymalnego prawdopodobieństwa wystąpienia modyfikacji  $Epi\_1$  w wykorzystanym GA. Medianę liczby generacji dla każdego ze zbiorów testowych dla  $GAGR$  i zmodyfikowanego ( $GAGR\ Epi\_1$ ) z różnym prawdopodobieństwem wystąpienia proponowanej operacji  $Epi\_1$  pokazano na wykresach na Rysunku 4.8.

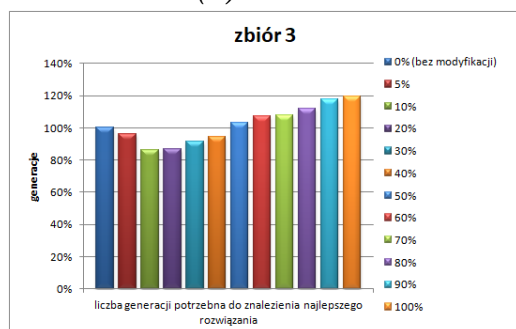
Na podstawie wykresów na Rysunku 4.8 można zauważyć zmniejszenie liczby generacji potrzebnych do prawidłowego pogrupowania testowych zbiorów danych dla  $GAGR\ Epi\_1$ , przy prawdopodobieństwie wystąpienia modyfikacji  $Epi\_1$  10-20%, w stosunku do  $GAGR$ . Naj-



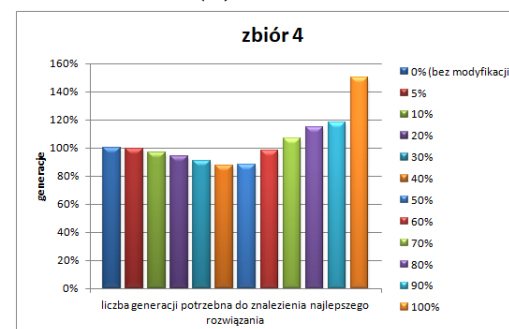
(a) zbiór 1



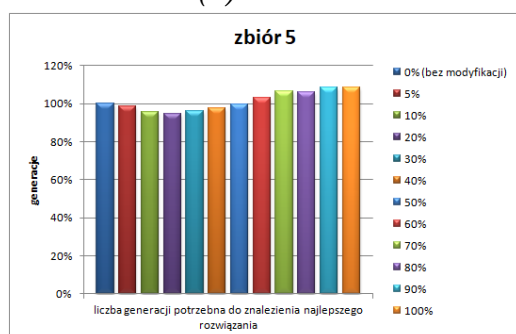
(b) zbiór 2



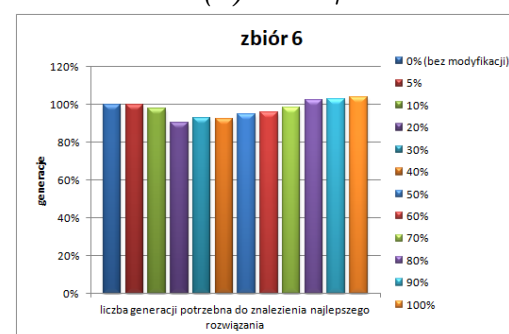
(c) zbiór 3



(d) zbiór 4



(e) zbiór 5



(f) zbiór 6

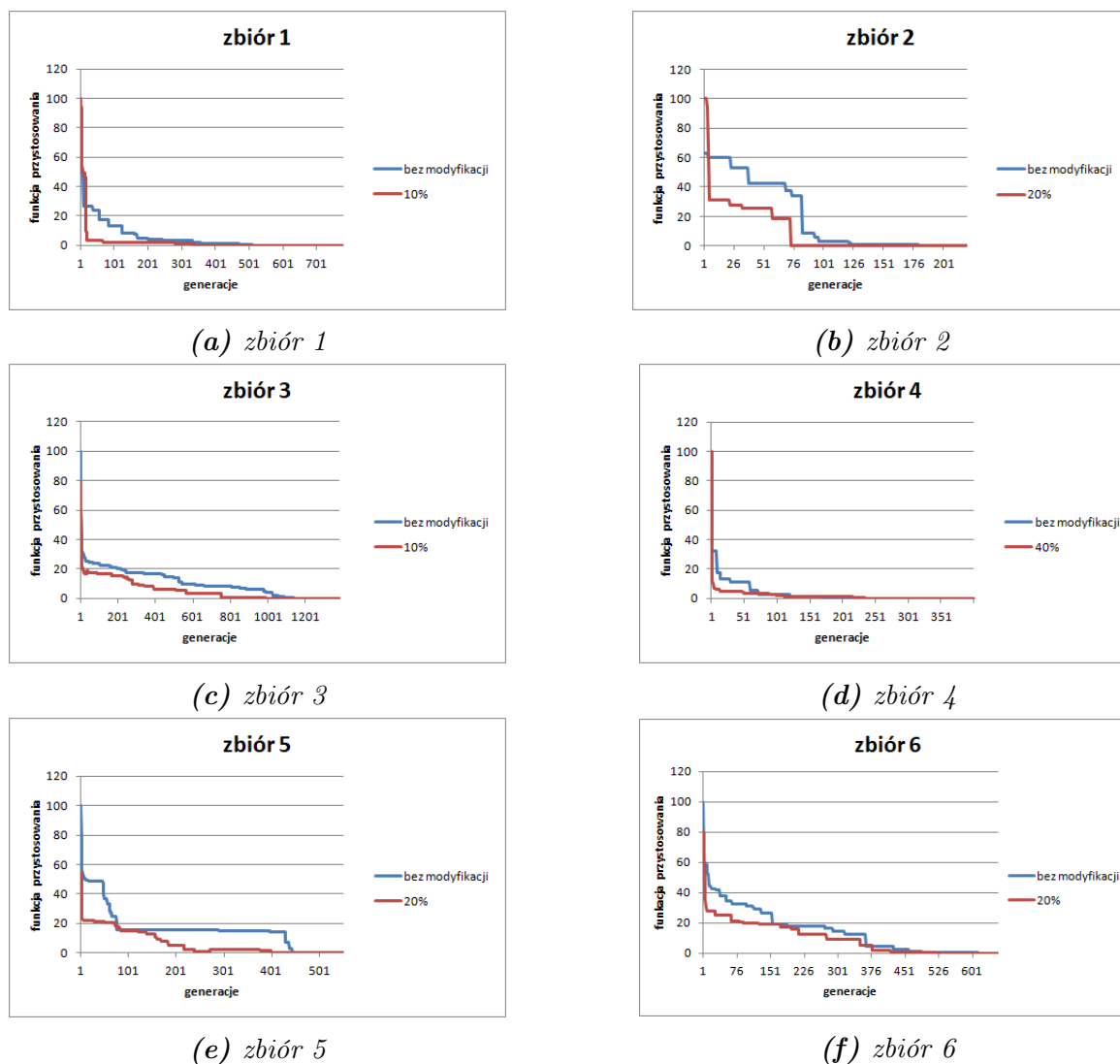
**Rysunek 4.8:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji  $Epi\_1$  w GAGR

większy spadek liczby generacji potrzebnych do prawidłowego pogrupowania zbiorów testowych można zauważyć w przypadku zbioru 2 (wykres 4.8b). Najmniejsze różnice zauważalne są w przypadku zbioru 5 (wykres 4.8e). Dla wszystkich testowych zbiorów danych, zbyt duże prawdopodobieństwo wystąpienia modyfikacji  $Epi\_1$  powoduje pogorszenie działania GAGR  $Epi\_1$  w stosunku do GAGR, poprzez zwiększenie liczby generacji potrzebnych do prawidłowego pogrupowania danych.

### Porównanie GAGR z GAGR $Epi\_1$

Dla optymalnej wartości prawdopodobieństwa wystąpienia operacji  $Epi\_1$  w GAGR, dokonano porównania spadku wartości funkcji przystosowania wraz z kolejnymi generacjami, w stosunku GAGR. Wyniki porównania wartości funkcji przystosowania pokazane zostały na wykresach na Rysunku 4.9. Na osi pionowej znajdują się wartości funkcji przystosowania, os

pozioma to kolejne generacje algorytmu *GAGR*.



**Rysunek 4.9:** Zmiany wartości funkcji przystosowania dla generacji algorytmów *GAGR* i *GAGR Epi\_1*

Na podstawie Rysunku 4.9 można stwierdzić, że dla wszystkich zbiorów testowych zastosowanie modyfikacji *Epi\_1* powoduje poprawę działania algorytmu. Dodatkowo można też zaobserwować szybszy spadek wartości funkcji przystosowania w *GAGR Epi\_1* w stosunku do algorytmu bez modyfikacji. W przypadku piątego zbioru danych (wykres 4.9e) można zaobserwować, że algorytm bez modyfikacji w pewnym momencie utyka na jednej wartości funkcji przystosowania, i przez znaczną liczbę kolejnych generacji nie ulegają poprawie wartości dopasowania osobników. W przypadku algorytmu z modyfikacją *Epi\_1*, w związku z większym zróżnicowaniem genotypu osobników utknięcie algorytmu w minimum lokalnym nie występuje.

### Porównanie czasów

W Tabeli 4.17 przedstawione zostało porównanie średnich czasów wykonania operacji algorytmu genetycznego do grupowania danych dla pojedynczej generacji *GAGR Epi\_1* i *GAGR*.

**Tabela 4.17:** Porównanie czasów pojedynczej generacji *GAGR* i *GAGR Epi\_1*

zbiory	czas [s]				zmiana %
	Epi_1		podstawowy		
	średnia	sd	średnia	sd	
zbiór 1	3,536	0,711	3,287	0,216	7,06%
zbiór 2	4,059	0,796	3,515	0,313	13,42%
zbiór 3	1,733	0,138	1,610	0,098	7,13%
zbiór 4	0,825	0,140	0,756	0,088	8,41%
zbiór 5	1,512	1,834	1,285	0,085	15,04%
zbiór 6	5,531	0,570	5,208	0,209	5,84%

Na podstawie otrzymanych średnich czasów pojedynczej generacji w *GAGR Epi\_1* i *GAGR* zaobserwowano wydłużenie czasu potrzebnego do przeprowadzanie operacji dla pojedynczej generacji w *GAGR Epi\_1* względem *GAGR*. Najmniejsze różnice zaobserwowano w przypadku zbioru 6, był to wzrost o 5,84%. Największą różnicę można zaobserwować w przypadku zbioru 2, gdy wzrost wyniósł 13,42%. Średnia zmiana procentowa czasu potrzebnego do przeprowadzenia wszystkich operacji algorytmu genetycznego dla pojedynczej generacji w *GAGR Epi\_1* względem *GAGR* wynosiła około +9,5%.

#### 4.4.5 Modyfikacja *Epi\_2*

W podrozdziale tym zaprezentowane zostały wyniki eksperymentów mające na celu sprawdzenie zaproponowanej w rozprawie modyfikacji naśladującej proces epigenetyczny metylacji cytozyny (*Epi\_2*).

Pierwszym krokiem w podjętych badaniach było sprawdzenie czy i z jakim optymalnym prawdopodobieństwem modyfikacja *Epi\_2* zmniejsza liczbę generacji w *GAGR(GAGR Epi\_2)*. Wyniki porównania liczb generacji w *GAGR*, oraz w *GAGR Epi\_2* przy różnym prawdopodobieństwie wystąpienia modyfikacji *Epi\_2* przedstawiono w Tabeli 4.18.

Na podstawie Tabeli 4.18 można zaobserwować, że przy odpowiednio dobranym prawdopodobieństwie modyfikacji *Epi\_2* można uzyskać redukcję liczby generacji w *GAGR Epi\_2* względem *GAGR*.

#### Wybór optymalnego prawdopodobieństwa wystąpienia modyfikacji *Epi\_2*

Na Rysunku 4.10 zaprezentowane zostały procentowe wartości liczby generacji potrzebne do uzyskania prawidłowego podziału na grupy w *GAGR Epi\_2* względem *GAGR*.

Na podstawie Rysunku 4.10 można zauważyć, że w przypadku zastosowania modyfikacji *Epi\_2* w *GAGR* z prawdopodobieństwem poniżej 50% można uzyskać zmniejszenie liczby ge-

**Tabela 4.18:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* (Pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

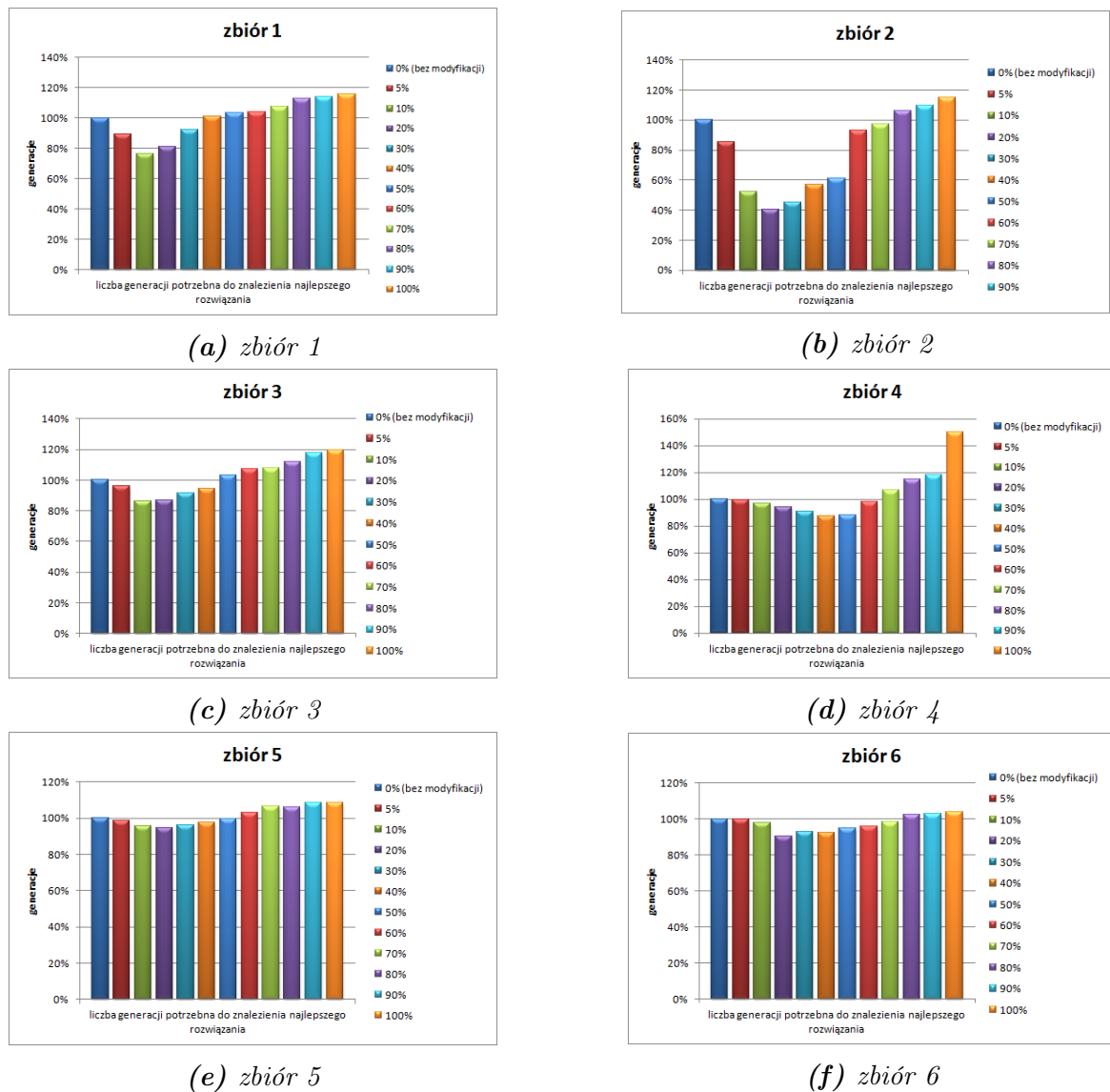
prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania					
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5	zbiór 6
0% (bez modyfikacji)	758	181	1151	270	450	614
5%	722	180	1098	241	445	598
10%	<b>675</b>	177	1016	<b>230</b>	441	591
20%	698	175	<b>955</b>	249	<b>435</b>	584
30%	757	172	988	268	438	<b>580</b>
40%	782	<b>170</b>	1072	287	449	582
50%	812	179	1093	302	456	596
60%	825	188	1133	308	463	608
70%	837	192	1169	321	478	616
80%	881	194	1173	315	476	631
90%	893	193	1199	326	482	638
100%	906	197	1208	331	506	644

neracji potrzebnych do uzyskania prawidłowego podziału na grupy (czyli poprawę efektywności działania algorytmu genetycznego). Największą redukcję liczby generacji uzyskano w przypadku zbioru 2 (wykres 4.10b), redukcja o 60% względem *GAGR*. Najmniejsze różnice w liczbie generacji pomiędzy *GAGREpi\_2* a *GAGR* można zaobserwować w przypadku zbiorów 5 i 6 (wykresy 4.10e, 4.10f). Zbyt duże prawdopodobieństwo wystąpienia modyfikacji *Epi\_2* w *GAGR* może wpłynąć negatywnie na efektywność działania algorytmu, poprzez zwiększenie liczby generacji potrzebnych do prawidłowego podziału na grupy.

### Porównanie efektywności działania *GAGR* z *GAGR Epi\_2*

Po ustaleniu optymalnego prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* przystąpiono do porównania zmiany wartości funkcji przystosowania wraz z kolejnymi generacjami GA, względem algorytmu podstawowego. Porównanie funkcji przystosowania dla poszczególnych zbiorów testowych pokazano na Rysunku 4.11.

Na podstawie Rysunku 4.11 można zauważyć większą dynamikę zmiany wartości funkcji przystosowania w *GAGR Epi\_2* względem *GAGR*. Największą różnicę w tempie zmian wartość funkcji przystosowania między algorytmem *GAGR Epi\_2* i *GAGR* można zaobserwować w przypadku zbioru 2 (wykres 4.11f).



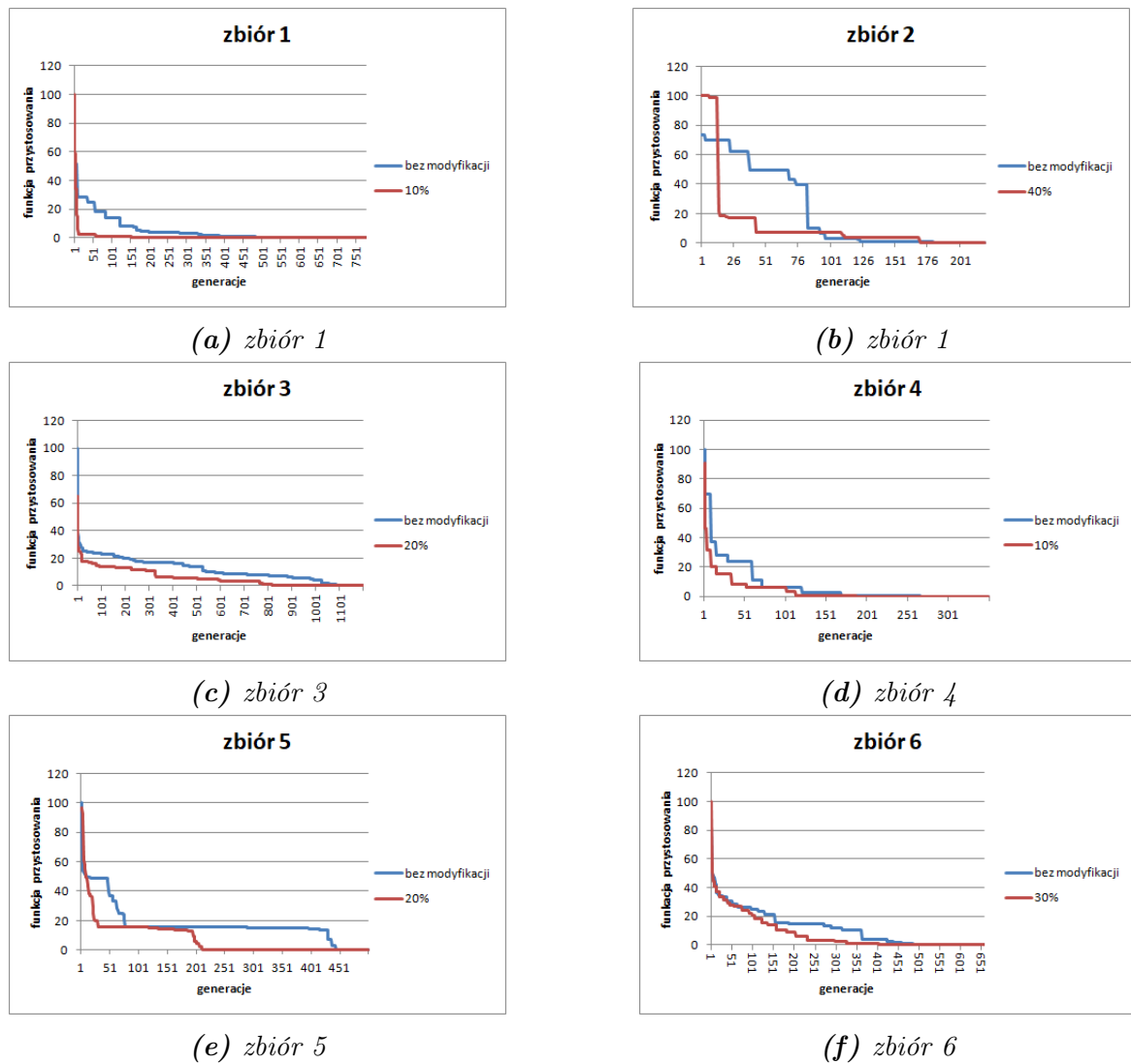
**Rysunek 4.10:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji  $Epi\_2$  w GAGR

## Porównanie czasów

W Tabeli 4.19 przedstawiono zestawienie średnich czasów i wartość odchylenia standardowego dla operacji algorytmów  $GAGR$  i  $GAGR Epi\_2$  w pojedynczej generacji algorytmu. Dodatkowo w ostatniej kolumnie przedstawiono procentową zmianę czasu w  $GAGR Epi\_2$  względem  $GAGR$  dla poszczególnych zbiorów użytych w eksperymentach.

Na podstawie Tabeli 4.19 można zauważyć wydłużenie czasu przeprowadzenia operacji algorytmu  $GAGR ed$  (w stosunku do  $GAGR$ ) dla pojedynczej generacji. Procentowe wydłużenie czasu działania algorytmu wynosi od 3,05% dla zbioru 1, do 16,30% dla zbioru 5.

Zastosowanie modyfikacji  $Epi\_2$  w algorytmie  $GAGR$  poprawia efektywność działania tego algorytmu poprzez redukcję liczby generacji potrzebnych do otrzymania prawidłowego podziału na grupy. Następuje jednak wydłużenie czasu operacji dla pojedynczej generacji algo-



**Rysunek 4.11:** Zmiany wartości funkcji przystosowania w kolejnych generacjach algorytmów GAGR Epi\_2 i GAGR

**Tabela 4.19:** Porównanie czasów pojedynczej generacji w GAGR Epi\_2i GAGR

zbiór	czas [s]				zmiana %
	GAGREpi_2		GAGR		
	średnia	sd	średnia	sd	
zbiór 1	3,390	0,571	3,287	0,216	3,05%
zbiór 2	4,059	0,796	3,515	0,313	13,42%
zbiór 3	1,752	0,185	1,610	0,098	8,11%
zbiór 4	0,824	0,171	0,756	0,088	8,28%
zbiór 5	1,535	1,428	1,285	0,085	16,30%
zbiór 6	5,476	0,735	5,208	0,209	4,90%

rytmu genetycznego.

#### 4.4.6 Operacja Epi\_3

Trzecią proponowaną w pracy operacją imitującą proces epigenetyczny jest wyłączenia alleliczne (Epi\_3). W pierwszym etapie eksperymentów sprawdzone zostało, czy zastosowanie

operacji  $Epi\_3$  w  $GAGR$  poprawia jego efektywność. W Tabeli 4.20 przedstawiono medianę liczby generacji dla  $GAGREpi\_3$  z różnym prawdopodobieństwem wystąpienia operacji  $Epi\_3$  oraz dla algorytmu  $GAGR$ .

**Tabela 4.20:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji  $Epi\_3$  w  $GAGR$  (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania					
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5	zbiór 6
0% (bez modyfikacji)	758	181	1151	270	450	614
5%	762	133	1150	271	444	616
10%	731	<b>125</b>	1139	266	439	610
20%	655	142	1115	257	435	572
30%	<b>612</b>	144	1103	242	432	<b>549</b>
40%	649	162	1098	<b>238</b>	<b>429</b>	557
50%	682	166	<b>1093</b>	243	433	563
60%	749	179	1106	254	446	584
70%	801	188	1140	268	456	593
80%	899	202	1147	277	461	611
90%	965	216	1152	289	465	637
100%	1056	228	1171	301	482	668

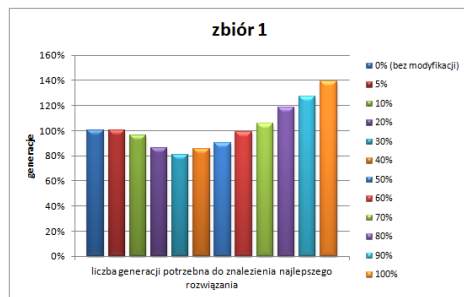
Na podstawie Tabeli 4.20 można zauważyć, że zastosowanie operacji  $Epi\_3$  z odpowiednim prawdopodobieństwem jej wystąpienia redukuje liczbę generacji w algorytmie potrzebnych do prawidłowego podziału na grupy zbioru danych. Optymalna wartość prawdopodobieństwa jest niestety zależna od zbioru danych i waha się od 10% do 50%.

### Empiryczne wyznaczenie optymalnego prawdopodobieństwa wystąpienia modyfikacji $Epi\_3$

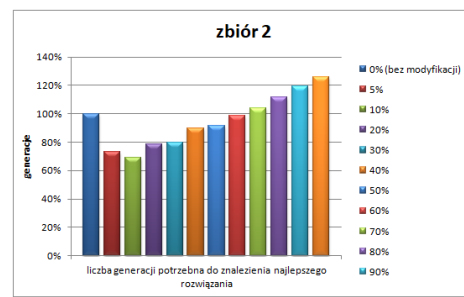
Na Rysunku 4.12 przedstawiono procentową zależność mediany liczby generacji w algorytmach z różnym prawdopodobieństwem wystąpienia operacji  $Epi\_3$  ( $GAGR Epi\_3$ ) w stosunku do liczby generacji w  $GAGR$ . Za wartość 100% przyjęto medianę liczby generacji w algorytmie bez modyfikacji.

Na podstawie Rysunku 4.12 można zauważyć, że odpowiednio dobrane prawdopodobieństwo wystąpienia operacji  $Epi\_3$  pozwala zredukować liczbę potrzebnych generacji do uzyskania optymalnego wyniku w algorytmie  $GAGR Epi\_3$  w stosunku do  $GAGR$ . Największą redukcję liczby generacji uzyskano w przypadku zbioru 2 dla prawdopodobieństwa wystąpienia modyfikacji  $Epi\_3$  równym 10% (redukcja liczby generacji o około 30%, wykres 4.12b). Najmniejszą redukcję liczby generacji otrzymano w przypadku zbioru 3 i 5 (wykresy 4.12c i 4.12e). W przypadku wszystkich testowych zbiorów danych zbyt duże prawdopodobieństwo wystąpienia

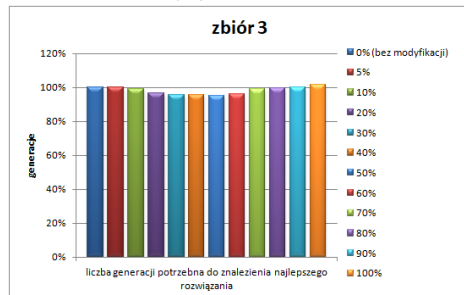




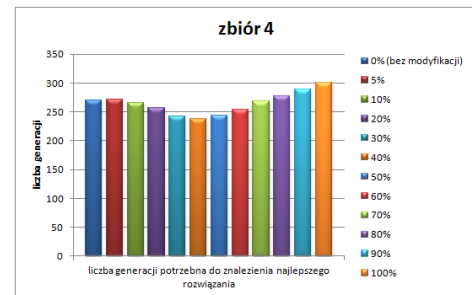
(a) zbiór 1



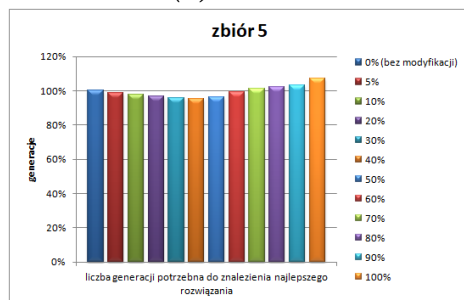
(b) zbiór 2



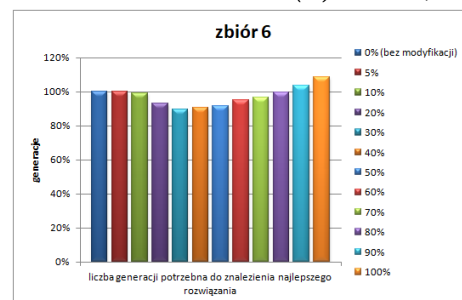
(c) zbiór 3



(d) zbiór 4



(e) zbiór 5



(f) zbiór 6

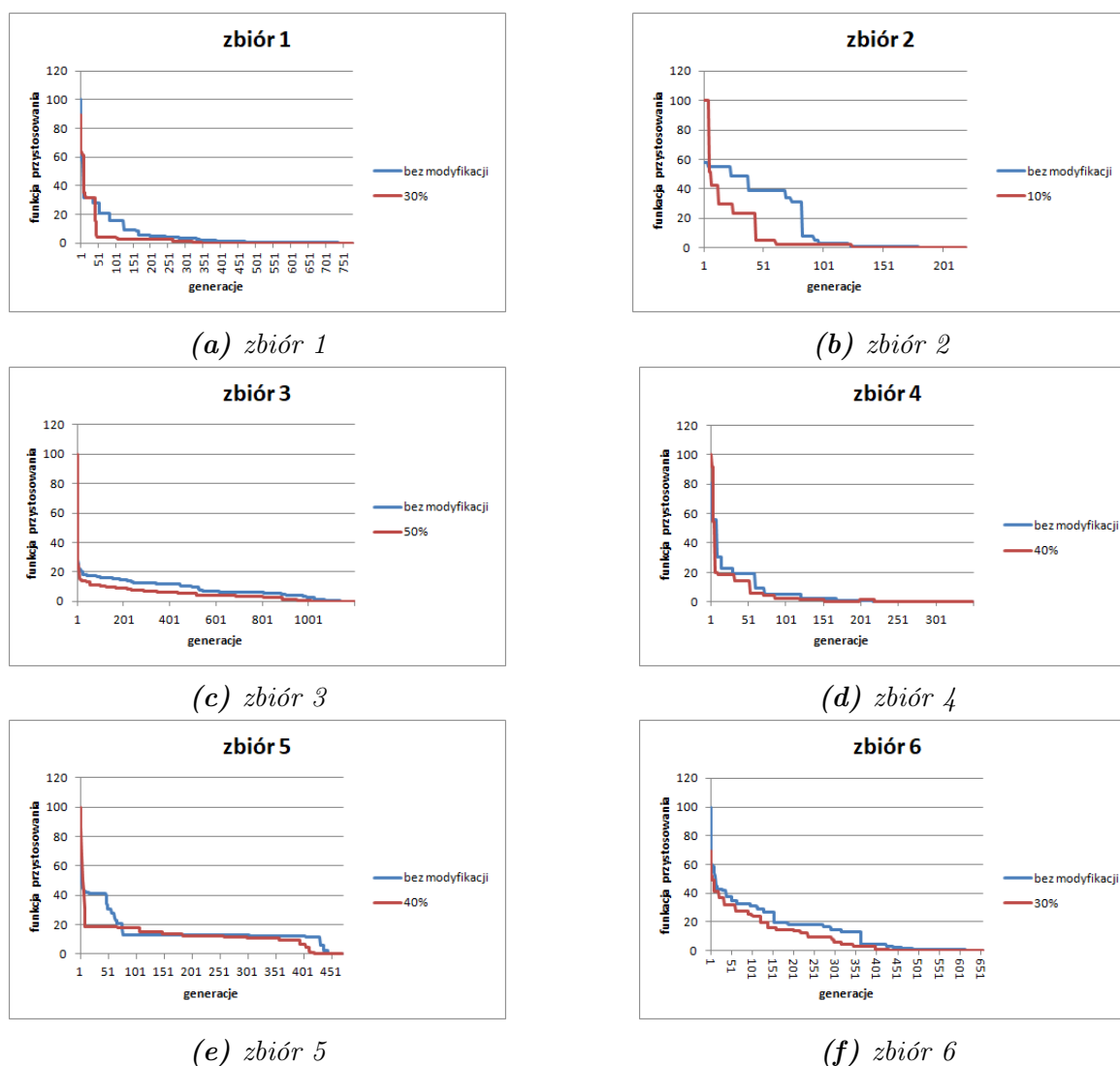
**Rysunek 4.12:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji  $Epi\_3$

operacji  $Epi\_3$  w  $GAGR Epi\_3$  powoduje pogorszenie efektywności działania algorytmu poprzez zwiększenie liczby generacji potrzebnych do uzyskania prawidłowego podziału na grupy testowych zbiorów danych, względem  $GAGR$ .

### Porównanie efektywności działania $GAGR Epi\_3$ z $GAGR$

Dla optymalnego prawdopodobieństwa wystąpienia operacji  $Epi\_3$  dla danego zbioru testowego dokonano porównania dynamiki zmiany wartości funkcji przystosowania najlepszego osobnika w  $GAGR Epi\_3$  względem  $GAGR$ . Otrzymane wyniki zostały przedstawione na Rysunku 4.13.

Wartości funkcji przystosowania dla poszczególnych generacji w poszczególnych zbiorach testowych, przedstawiono na Rysunku 4.13. Na podstawie Rysunku 4.13 można zauważyć większą dynamikę zmiany wartości funkcji przystosowania najlepszego osobnika w algorytmie  $GAGR Epi\_3$  z optymalnym prawdopodobieństwem wystąpienia operacji  $Epi\_3$  w stosun-



**Rysunek 4.13:** Zmiany wartości funkcji przystosowania dla generacji algorytmu GAGR i GAGR Epi\_3

ku do GAGR. Największą różnicę w tempie zmiany wartości funkcji przystosowania wraz z kolejnymi generacjami można zauważyć w przypadku zbioru 2 (wykres 4.13b).

### Porównanie czasów

W Tabeli 4.21 przedstawiono średnie wartości czasu i wartość odchylenia standardowego dla operacji w pojedynczej generacji GAGREpi\_3 i GAGR. W ostatniej kolumnie tabeli przedstawiono dodatkowo o ile procentowo wydłużył się czas przeprowadzenia operacji w pojedynczej generacji algorytmu genetycznego w przypadku GAGR Epi\_3 w stosunku do GAGR.

Na podstawie Tabeli 4.21 można zauważyć wydłużenie czasu wykonywania operacji w pojedynczej generacji w algorytmie z operacją Epi\_3 względem algorytmu bazowego bez modyfikacji. Średnio czas pojedynczej generacji wydłużył się o 10,79%. Największe wydłużenie czasu można zaobserwować w przypadku zbioru 3 (wzrost o 13,95%), najmniejsze w przypadku zbioru 6 (wzrost o 6,45%).

**Tabela 4.21:** Porównanie czasów pojedynczej generacji w GAGR Epi\_3 i GAGR

zbiór	czas [s]				zmiana %
	GAGREpi_3		GAGR		
	średnia	sd	średnia	sd	
zbiór 1	3,650	0,869	3,287	0,216	9,94%
zbiór 2	4,058	0,742	3,515	0,313	13,39%
zbiór 3	1,871	0,503	1,610	0,098	13,95%
zbiór 4	0,845	0,394	0,756	0,088	10,49%
zbiór 5	1,436	0,463	1,285	0,085	10,54%
zbiór 6	5,567	0,817	5,208	0,209	6,45%

Na podstawie przeprowadzonych eksperymentów można stwierdzić, że wprowadzenie modyfikacji Epi\_3 do GAGR pozwala zredukować liczbę generacji potrzebną do prawidłowego podziału danych na grupy. W przypadku zastosowania modyfikacji Epi\_3 należy wziąć pod uwagę, że zbyt duże prawdopodobieństwo wystąpienia modyfikacji Epi\_3 w GAGR może zwiększyć liczbę generacji, a przez co doprowadzić do zwiększenia czasu potrzebnego na pogrupowanie danych. Wprowadzenie modyfikacji Epi\_3 wpływa również na czas wykonywania operacji w pojedynczej generacji algorytmu, wydłużając go.

#### 4.4.7 Podsumowanie wyników

Podrozdział ten zawiera wyniki eksperymentów nad oceną skuteczności zastosowania trzech proponowanych w pracy operacji naśladujących procesy epigenetyczne w autorskim algorytmie genetycznym do grupowania danych. W ramach przeprowadzonych badań dokonano doboru optymalnych prawdopodobieństw wystąpienia proponowanych w pracy dodatkowych operacji algorytmu genetycznego. Optymalne prawdopodobieństwo wystąpienia proponowanych w pracy operacji w algorytmie do grupowania danych przedstawiono w Tabeli 4.22.

**Tabela 4.22:** Optymalne prawdopodobieństwo wystąpienia operacji Epi\_x w algorytmie GAGR

Modyfikacja	Optymalne prawdopodobieństwo
Epi_1	20%
Epi_2	20%
Epi_3	30%

Optymalne prawdopodobieństwo wystąpienia operacji Epi\_1 i Epi\_2 w algorytmie GAGR wynosi około 20% (Tabela 4.22). Dla operacji Epi\_3 optymalne prawdopodobieństwo wystąpienia wynosi 30%.

W Tabeli 4.23 przedstawiono procentową wartość średniej redukcji liczby generacji potrzebnych do uzyskania optymalnego wyniku w algorytmach GAGREpi\_x, względem algorytmu GAGR.

Tabela 4.23 zawiera również informację o ile procent zastosowanie proponowanych w pracy operacji w algorytmie genetycznym wydłużyło czas pojedynczej generacji algorytmów *GAGR Epi\_x* względem algorytmu *GAGR*.

**Tabela 4.23:** Procentowa wartość średniej redukcji liczby generacji oraz procentowe wydłużenie czasu pojedynczej generacji algorytmów *GAGR Epi\_x* względem algorytmu *GAGR*

Proponowana operacja	średnia redukcja liczby generacji	średnie % wydłużenie czasu pojedynczej generacji
<i>Epi_1</i>	↓ 20%	↑ 9,50%
<i>Epi_2</i>	↓ 21%	↑ 9,01%
<i>Epi_3</i>	↓ 14%	↑ 10,79%

Zastosowanie proponowanych w pracy dodatkowych operacji bazujących na procesach epigenetycznych w algorytmie *GAGR* poprawia działanie tego algorytmu poprzez redukcję liczby generacji potrzebnych do uzyskania oczekiwanego wyniku, Tabela 4.23. Największą redukcję liczby generacji uzyskano przy zastosowaniu modyfikacji *Epi\_2*, średnia redukcja o 21%. Zastosowanie dodatkowych operacji wiąże się jednocześnie z wydłużeniem czasu potrzebnego na przeprowadzenie wszystkich operacji algorytmu genetycznego w pojedynczej generacji algorytmu. Najmniej wydłużył się czas operacji w pojedynczej generacji w przypadku zastosowania modyfikacji *Epi\_2* (Tabela 4.23), wzrost o 9,01%. Najwięcej w przypadku dołożenia do algorytmu operacji *Epi\_3*, wzrost o 10,79%.

Na podstawie badań nad możliwością zastosowania proponowanych w pracy dodatkowych operacji w algorytmie *GAGR* rezultaty uzyskano przy zastosowaniu operacji *Epi\_2*. Uzyskano największą redukcję liczby generacji, przy najmniejszym wzroście czasu operacji w pojedynczej generacji.

## 4.5 Badania algorytmu wykrywania wartości odstających

Kolejnym algorytmem genetycznym na którym sprawdzono możliwość poprawy efektywności działania poprzez zastosowanie dodatkowych operacji wzorowanych na procesach epigenetycznych, był algorytm do wykrywania danych odstających (*GADO*). W algorytmie tym, tak jak w przypadku poprzednich zostały dołożone dodatkowe operacje proponowane w pracy (*GADO Epi\_x*), a otrzymane wyniki badań ich działania zostały porównane z bazowym algorytmem *GADO*.

### 4.5.1 Dane wykorzystane w eksperymentach

W eksperymentach dotyczących możliwości zastosowania operacji naśladujących procesy epigenetyczne w *GADO* posłużono się 6 zbiorami danych. Zbiór 1 jest zbiorem utworzonym w sposób sztuczny przez autora pracy, pozostałe zbiory są ogólnodostępnymi zbiorami służącymi do testowania metod wykrywania wartości odstających. W Tabeli 4.24 przedstawiono wykorzystane w eksperymentach zbiory danych wraz z ich charakterystyką.

**Tabela 4.24:** Zbiory danych wykorzystane w eksperymentach przy wykorzystaniu *GADO*

Zbiór	Nazwa	Rozmiar zbioru	Liczba kolumn	Liczba wartości odstających
zbiór 1	Zbiór sztuczny	2000	3	190
zbiór 2	Thyroid Disease [134]	3772	6	93
zbiór 3	Breast Cancer Wisconsin [175]	570	30	212
zbiór 4	Pima Indians Diabetes [101]	768	8	268
zbiór 5	Glass Identification [89]	214	7	9
zbiór 6	Pen-Based Recognition of Handwritten [89]	6870	16	156

Dla wszystkich zbiorów danych wykorzystanych w eksperymentach znana była liczba wartości odstających w zbiorach, oraz to które wartości powinny zostać uznane za odstające. Zbiory różniły się rozmiarem i liczbą kolumn braną pod uwagę w procesie wykrywania wartości odstających.

### 4.5.2 Parametry algorytmu

Dla algorytmu *GADO* na potrzeby przeprowadzonych eksperymentów konieczne było ustalenie parametrów algorytmu. Parametrami koniecznymi do ustalenia były prawdopodobieństwa wystąpienia operacji krzyżowania, mutacji, metoda selekcji i warunek stopu. Wartości parametrów dla *GADO* zostały przedstawione w Tabeli 4.25. Wartości parametrów są wartościami literaturowymi i zostały zaczerpnięte z publikacji autora algorytmu [11].

**Tabela 4.25:** Parametry algorytmu *GADO* wykorzystane w eksperymentach

Parametr	Wartość
liczba osobników w populacji	10% liczby elementów z zbiorze danych
prawdopodobieństwo krzyżowania	100%
prawdopodobieństwo mutacji	1%
metoda selekcji osobników	turniejowa
warunek stopu	warunkowana długością zbioru liczba generacji bez poprawy wartości funkcji przystosowania

Przedstawione parametry zostały wykorzystane w eksperymentach zarówno dla algorytmu bazowego (*GADO*) jak i algorytmów z dołożonymi proponowanymi operacjami naśladujące procesy epigenetyczne (*GADO Epi\_x*).

### 4.5.3 Operacja *Epi\_1*

W niniejszym podrozdziale przedstawione zostały wyniki eksperymentów przeprowadzonych w celu sprawdzenia, czy uzupełnienie algorytmu *GADO* o dodatkową operację naśladującą epigenetyczny proces dziedziczenia za pomocą prionu (*Epi\_1*) poprawia efektywność tego algorytmu (*GADO Epi\_1*). Pierwszym krokiem było dobranie optymalnego prawdopodobieństwa modyfikacji. W kolejnym kroku uzyskane wartości funkcji przystosowania dla optymalnego prawdopodobieństwa wystąpienia modyfikacji zostały porównane z wartościami funkcji przystosowania dla algorytmu *GADO*.

W Tabeli 4.26 pokazano zbiorcze zestawienie liczby generacji potrzebnych do wykrycia wszystkich wartości odstających w sześciu testowych zbiorach danych przez *GADO* oraz *GADO Epi\_1* z różnym prawdopodobieństwem wystąpienia modyfikacji *Epi\_1*.

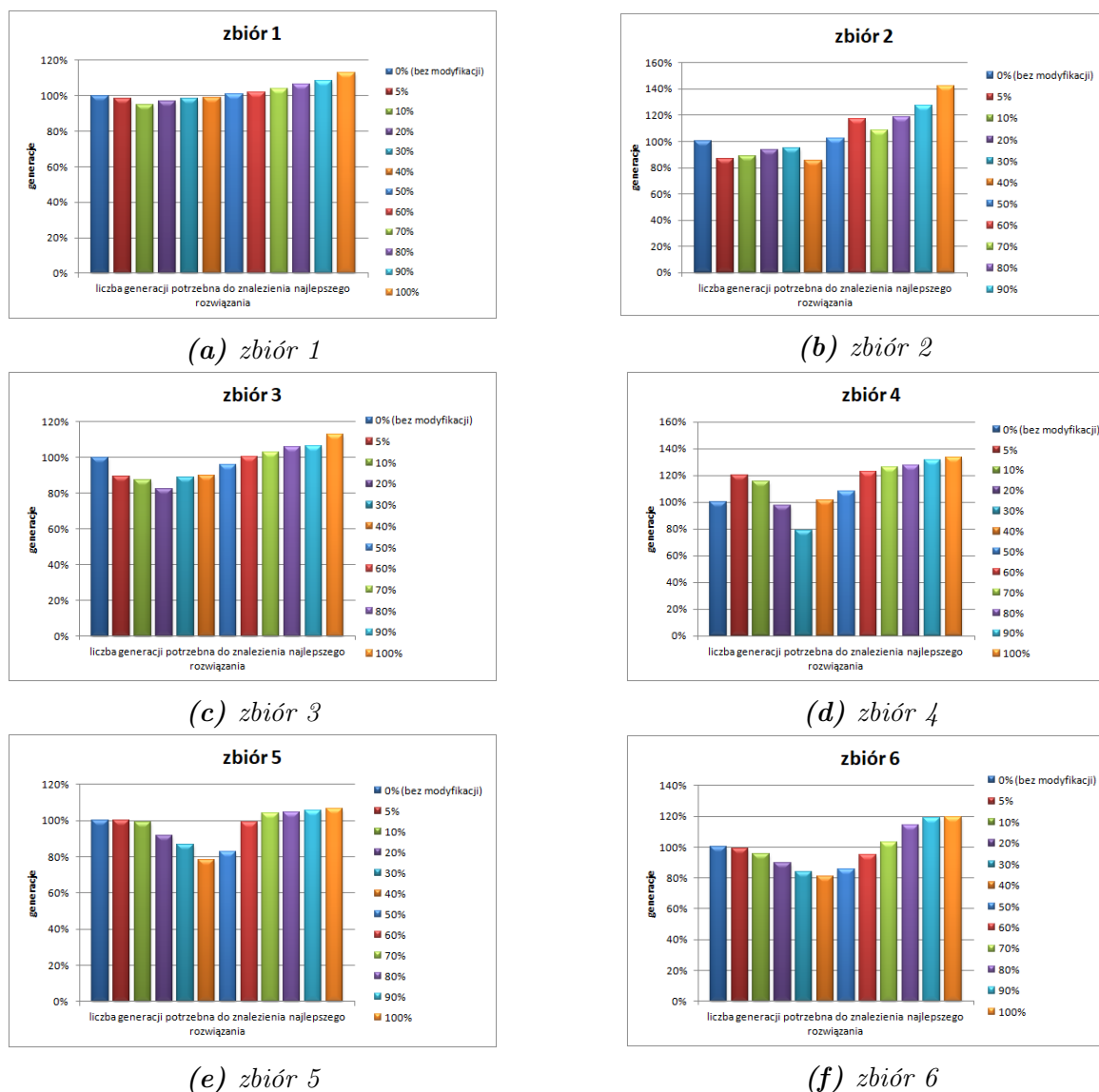
**Tabela 4.26:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji *Epi\_1* (Pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania					
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5	zbiór 6
0% (bez modyfikacji)	2675	2175	661	740	1809	1479
5%	2627	<b>1883</b>	592	887	1813	1468
10%	<b>2543</b>	1934	577	855	1788	1411
20%	2589	2032	<b>545</b>	721	1655	1325
30%	2633	2066	587	<b>585</b>	1564	1237
40%	2649	1851	594	753	<b>1419</b>	<b>1195</b>
50%	2703	2221	633	798	1498	1264
60%	2729	2553	664	911	1789	1406
70%	2781	2354	682	935	1877	1522
80%	2845	2574	701	943	1892	1689
90%	2897	2765	702	971	1906	1755
100%	3015	3100	746	990	1923	1769

### Empiryczny dobór optymalnego prawdopodobieństwa wystąpienia modyfikacji *Epi\_1*

Dla sześciu testowych zbiorów danych, została porównana liczba generacji (w zależności od prawdopodobieństwa wystąpienia modyfikacji *Epi\_1*) potrzebnych do wykrycia wszystkich

wartości odstających w zbiorach danych. Liczbę generacji w *GADO* potrzebną do wykrycia wszystkich wartości odstających w zbiorach testowych pokazano na Rysunku 4.14.



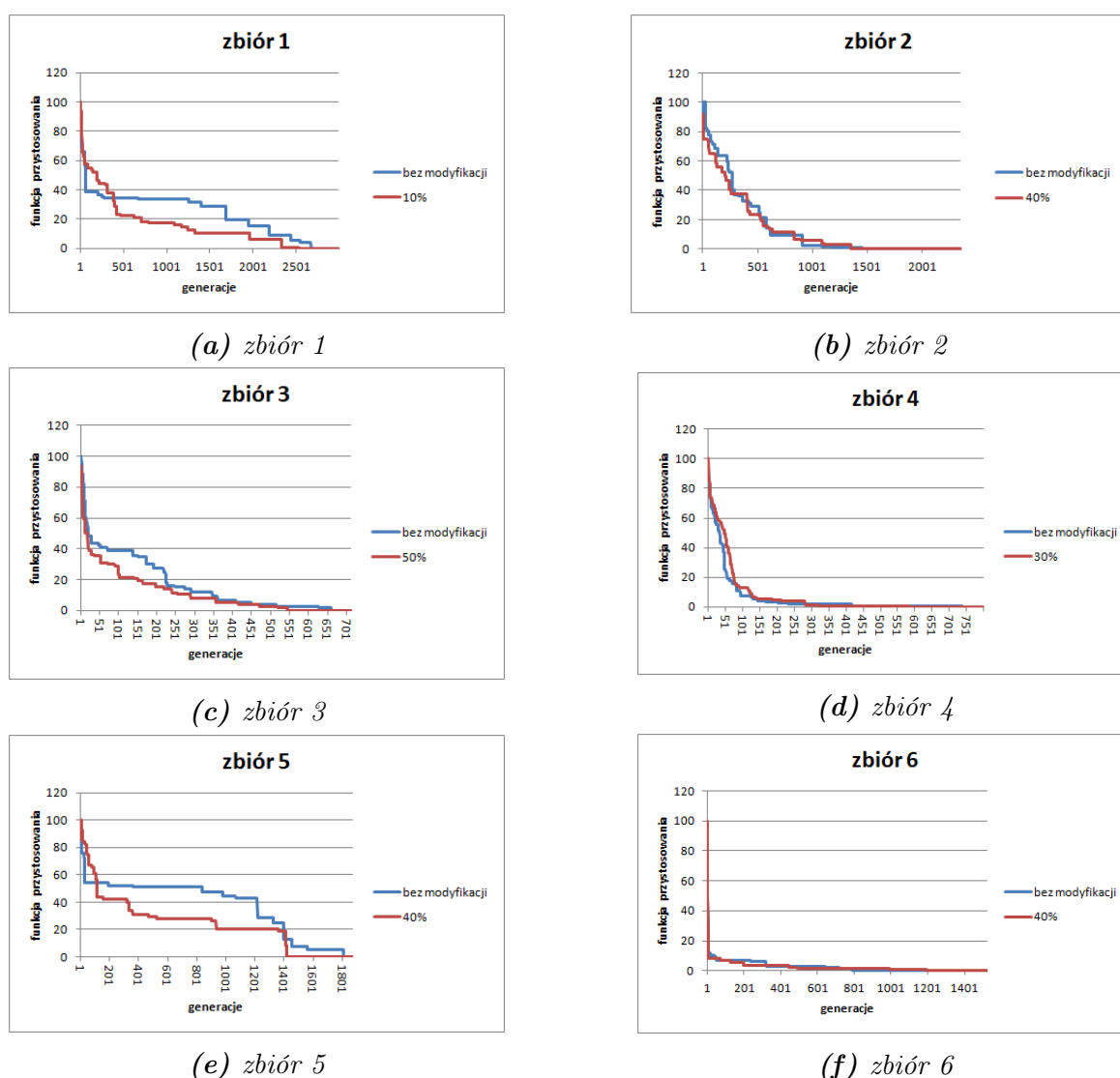
**Rysunek 4.14:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji *Epi\_1*

Na podstawie Rysunku 4.14 można zauważyć, że w przypadku prawdopodobieństwa wystąpienia modyfikacji *Epi\_1* w *GADO* poniżej 60% uzyskuje się poprawę działania *GADO*, poprzez zmniejszenie liczby generacji potrzebnych do wykrycia wartości odstających w zbiorze danych. Największy spadek liczby generacji można zaobserwować w przypadku zbioru 4, dla prawdopodobieństwa wystąpienia modyfikacji 30% (wykres 4.14d), oraz dla zbioru 5 i 6 dla prawdopodobieństwa wystąpienia modyfikacji *Epi\_1* wynoszącym 40% (wykresy 4.14e, oraz 4.14f). Najmniejszą różnicę w liczbie generacji potrzebnych do wykrycia wartości odstających można zaobserwować w przypadku zbioru 1 (wykres 4.14a). W przypadku dużego prawdopodobieństwa wystąpienia modyfikacji *Epi\_1* (powyżej 60%) można zaobserwować pogorszenie

działania *GADO Epi\_1*, poprzez zwiększenie liczby generacji potrzebnych do wykrycia wszystkich wartości odstających w stosunku do algorytmu bez modyfikacji.

### Porównanie efektywności działania *GADO* z *GADO Epi\_1*

Dla prawdopodobieństwa wystąpienia modyfikacji *Epi\_1* przy którym uzyskano największy spadek liczby generacji potrzebnych do wykrycia wszystkich wartości odstających w zbiorach testowych dokonano porównania zmiany wartości funkcji przystosowania w algorytmie *GADO Epi\_1* z algorytmem *GADO*. Wartości funkcji przystosowania dla kolejnych generacji algorytmu z operacją *Epi\_1* i algorytmu bez modyfikacji pokazano na wykresach na Rysunku 4.15.



**Rysunek 4.15:** Wartości funkcji przystosowania dla kolejnych generacji algorytmu *GADO* i *GADO Epi\_1*

Na podstawie Rysunku 4.15 można zaobserwować, że największe różnice w dynamice zmiany wartości funkcji przystosowania dla *GADO Epi\_1* w stosunku do *GADO* występują w



przypadku zbiorów 1 i 5 (wykresy 4.15a, oraz 4.15e). Natomiast najmniejsze różnice w dynamice zmiany wartości funkcji przystosowania można zaobserwować dla zbiorów 2, 4 i 6 (wykresy 4.15b, 4.15d, oraz 4.15f).

### Porównanie czasów działania *GADO* i *GADO Epi\_1*

W Tabeli 4.27 przedstawione zostały średnie wartości czasów, oraz wartość odchylenia standardowego przeprowadzenia wszystkich operacji algorytmu genetycznego dla pojedynczej generacji algorytmu do wykrywania danych odstających. W tabeli przedstawiono również wartość procentowej zmiany wartości czasu *GADO Epi\_1* względem *GADO*.

**Tabela 4.27:** Porównanie czasów pojedynczej generacji dla *GADO* oraz *GADO Epi\_1*

zbiory	czas [s]				zmiana %
	<i>GADOEpi_1</i>		<i>GADO</i>		
	średnia	sd	średnia	sd	
zbiór 1	0,619	0,042	0,567	0,136	8,37%
zbiór 2	0,362	0,025	0,331	0,035	8,75%
zbiór 3	0,254	0,019	0,225	0,027	11,25%
zbiór 4	0,169	0,019	0,161	0,018	5,09%
zbiór 5	1,179	0,068	1,135	0,077	3,70%
zbiór 6	0,370	0,034	0,341	0,014	7,87%

Na podstawie Tabeli 4.27 można zaobserwować, że wprowadzenie do *GADO* operacji *Epi\_1* wpłynęło na czas operacji w pojedynczej generacji, powodując wydłużenie go. Spośród wykorzystanych w eksperymentach zbiorów danych najmniejszy wzrost czasu pojedynczej generacji dla *GADO Epi\_1* (względem *GADO*) można zaobserwować w przypadku zbioru 5, wzrost wyniósł tylko 3,7%. Natomiast największy wzrost czasu można zaobserwować w przypadku zbioru 3, wzrost o 11,25%. Średni wzrost czasu potrzebnego do przeprowadzania operacji w pojedynczej generacji w algorytmie z modyfikacją *Epi\_1* wynosił 7,5%.

Na podstawie otrzymanych wyników dla *GADO*, oraz *GADO Epi\_1* można zauważyć, że w przypadku tego algorytmu genetycznego, również zaobserwowano poprawę działania *GADO Epi\_1*, w stosunku do *GADO*, poprzez zmniejszenie liczby generacji potrzebnych do wykrycia wartości odstających w zbiorach danych.

#### 4.5.4 Modyfikacja *Epi\_2*

W podrozdziale tym zaprezentowane zostały wyniki eksperymentów mające na celu sprawdzenie zaproponowanej w pracy modyfikacji algorytmu genetycznego naśladującej proces epigenetyczny metylacji cytozyny (*Epi\_2*). Tak jak w przypadku wcześniejszych eksperymentów, w pierwszym kroku sprawdzone zostało czy przy zastosowaniu modyfikacji *Epi\_2*

w algorytmie genetycznym (*GADO Epi\_2*) nastąpi redukcja liczby generacji potrzebnych do wykrycia wszystkich danych odstających w zbiorach testowych. W przypadku redukcji liczby generacji w *GADO Epi\_2* należy też ustalić jakie jest optymalne prawdopodobieństwo wystąpienia modyfikacji *Epi\_2* w *GADO*. Liczba generacji potrzebna do wykrycia wszystkich danych odstających dla poszczególnych zbiorów testowych przez *GADO* i *GADO Epi\_2* z różnym prawdopodobieństwem wystąpienia modyfikacji *Epi\_2* przedstawiono w Tabeli 4.28.

**Tabela 4.28:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

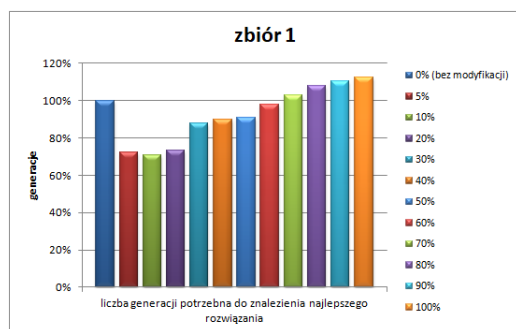
prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania					
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5	zbiór 6
0% (bez modyfikacji)	2675	2175	661	740	1809	1479
5%	1932	1568	415	735	1801	1481
10%	<b>1898</b>	1564	406	551	1655	1472
20%	1954	<b>1558</b>	397	<b>467</b>	<b>1510</b>	<b>1419</b>
30%	2354	1669	<b>390</b>	482	1582	1359
40%	2398	1857	409	563	1691	1305
50%	2425	1884	421	609	1709	1378
60%	2619	1928	478	869	1753	1441
70%	2756	1950	523	1052	1806	1522
80%	2887	2106	614	1104	1823	1541
90%	2954	2215	674	1129	1893	1603
100%	3012	2239	687	1206	1933	1622

Na podstawie Tabeli 4.28 można zauważyć, że dla algorytmu *GADO Epi\_2* przy zastosowaniu odpowiedniego prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* uzyskuje się redukcję liczby generacji względem *GADO*.

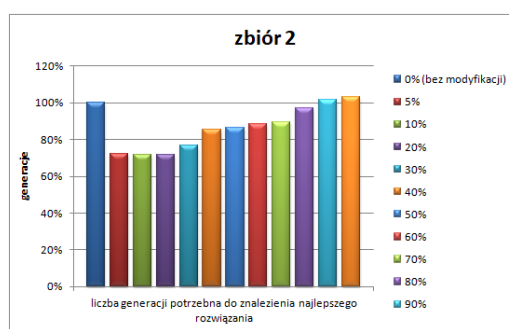
### Empiryczny dobór optymalnego prawdopodobieństwa wystąpienia modyfikacji *Epi\_2*

Na Rysunku 4.16 przedstawiono procentową zależność liczby generacji pomiędzy *GADO*, a *GADO Epi\_2* różnym prawdopodobieństwem wystąpienia modyfikacji *Epi\_2*. Jako wartość 100% przyjęto medianę (dla 9 powtórzeń uruchomienia GA) liczby generacji dla algorytmu bez modyfikacji. Wartości poniżej 100% oznaczają, że uzyskano redukcję liczby generacji potrzebną do wykrycia wszystkich wartości odstających w algorytmie z modyfikacją *Epi\_2*, względem algorytmu podstawowego.

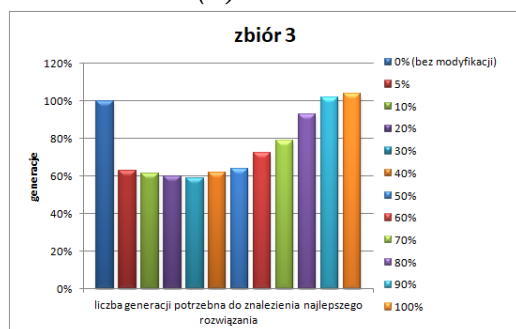
Na podstawie Rysunku 4.16 można zauważyć, że przy odpowiednim doborze prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* w *GADO Epi\_2*, można uzyskać poprawę efektywności działania algorytmu w postaci redukcji liczby generacji potrzebnych do wykrycia



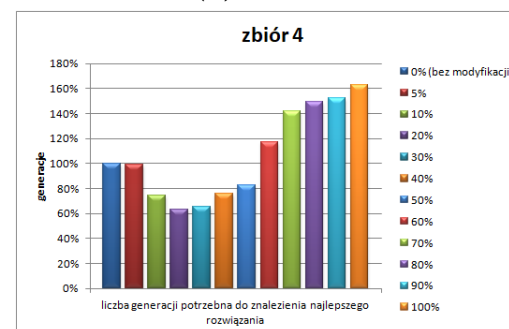
(a) zbiór 1



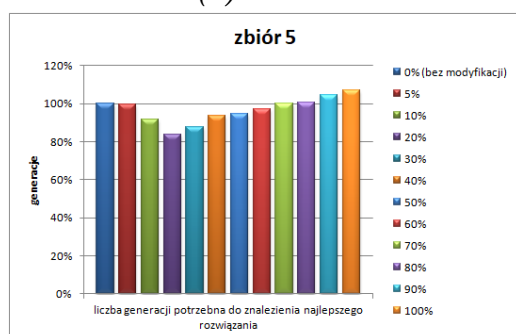
(b) zbiór 2



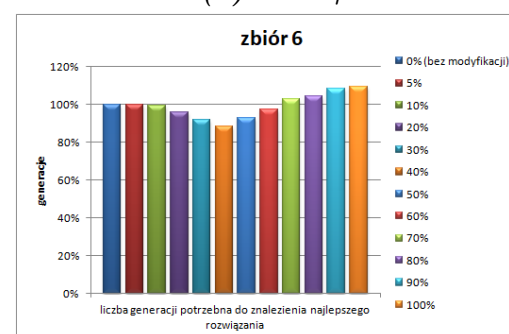
(c) zbiór 3



(d) zbiór 4



(e) zbiór 5



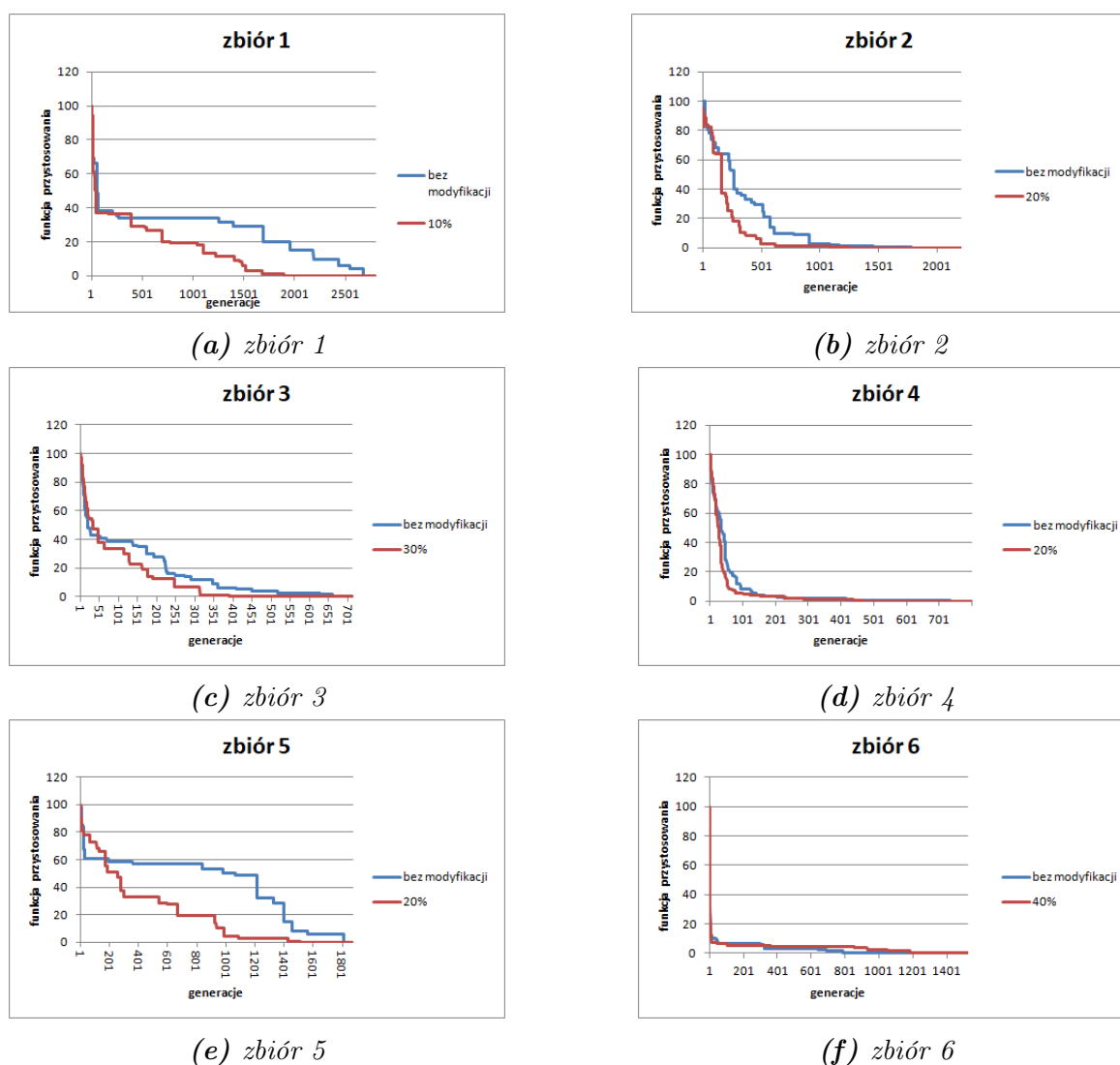
(f) zbiór 6

**Rysunek 4.16:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji  $Epi_2$

wszystkich wartości odstających względem  $GADO$ . Najlepsze rezultaty osiągnięto w przypadku niewielkich prawdopodobieństw wystąpienia modyfikacji  $Epi_2$  na poziomie od 10-30%. Największą procentową redukcję liczby generacji uzyskano w przypadku zbioru 3 (wykres 4.16c). Najmniej zauważalne różnice w liczbie generacji pomiędzy algorytmem  $GADO$ , a różnymi wersjami algorytmu  $GADO Epi_2$  uzyskano w zbiorze 6 (wykres 4.16f). W przypadku zbyt dużego prawdopodobieństwa wystąpienia modyfikacji  $Epi_2$  dochodzi do pogorszenia efektywności działania  $GADO Epi_2$  (względem  $GADO$ , poprzez zwiększenie liczby generacji potrzebnych do wykrycia danych odstających). Największe pogorszenie działania  $GADO Epi_2$  można zaobserwować dla dużych prawdopodobieństw wystąpienia modyfikacji  $Epi_2$  (powyżej 60%) w przypadku zbioru 4 (wykres 4.16d).

## Porównanie efektywności działania *GADO Epi\_2* z *GADO*

Dla optymalnych wartości prawdopodobieństw wystąpienia modyfikacji *Epi\_1* w testowych zbiorach, dokonano porównania zmiany wartości funkcji przystosowania dla algorytmu *GADO Epi\_2* i algorytmu podstawowego. Zmianę wartości funkcji przystosowania dla *GADO* i *GADO Epi\_2* z optymalnym prawdopodobieństwem wystąpienia modyfikacji *Epi\_2* pokazano na wykresach na Rysunku 4.17.



**Rysunek 4.17:** Zmiana wartości funkcji przystosowania dla kolejnych generacji algorytmu genetycznego, w zależności od zastosowania modyfikacji *Epi\_2* i bez niej

Na podstawie Rysunku 4.17 można zauważyć, że prawie w przypadku wszystkich zbiorów testowych uzyskano przy użyciu *GADO Epi\_2* zwiększenie dynamiki zmiany wartości funkcji przystosowania względem *GADO*. Wyjątkiem jest zbiór 6 (wykres 4.17f), gdzie różnice w zmianie wartości funkcji przystosowania są niemal niezauważalne. W przypadku zbioru 6, zastosowanie modyfikacji *Epi\_2* w najmniejszym stopniu wpłynęło na redukcję liczby generacji *GADO* (na podstawie wykresu 4.16f). Największy wzrost dynamiki zmiany wartości funkcji

przystosowania można zaobserwować w przypadku zbiorów testowych 1 i 5 (wykresy 4.17a, oraz 4.17e). W przeprowadzonych eksperymentach uzyskano poprawę efektywności działania *GADO* poprzez zastosowanie z odpowiednim prawdopodobieństwem modyfikacji *Epi\_2*. Poprawa efektywności polega na zmniejszeniu liczby generacji potrzebnych do wykrycia wszystkich danych odstających w zbiorach testowych.

### Porównanie czasów działania *GADO* i *GADO Epi\_2*

W Tabeli 4.29 pokazano średnie czasy i wartość odchylenia standardowego dla wszystkich operacji algorytmów *GADO* i *GADO Epi\_2* przeprowadzanych w ramach pojedynczej generacji, dla poszczególnych zbiorów danych. Dodatkowo w ostatniej kolumnie tabeli umieszczono procentową wartość zmiany czasu (o ile wydłużył się czas) algorytmu *GADO Epi\_2* względem *GADO*.

**Tabela 4.29:** Porównanie czasów pojedynczej generacji w *GADO* i *GADO Epi\_2*

zbiór	czas [s]				zmiana %
	<i>GADO Epi_2</i>		<i>GADO</i>		
	średnia	sd	średnia	sd	
zbiór 1	0,592	0,149	0,567	0,136	4,18%
zbiór 2	0,344	0,011	0,331	0,035	3,97%
zbiór 3	0,236	0,016	0,225	0,027	4,68%
zbiór 4	0,171	0,037	0,161	0,018	5,94%
zbiór 5	1,144	0,039	1,135	0,077	0,80%
zbiór 6	0,356	0,028	0,341	0,014	4,24%

Na podstawie Tabeli 4.29 można zaobserwować wydłużenie czasu operacji pojedynczej generacji *GADO Epi\_2* względem *GADO*. Wydłużenie czasu operacji *GADO Epi\_2* nie jest jednak duże, zaobserwowano procentowy wzrost od 0,80% dla zbioru 5, do 5,94% dla zbioru 4.

Na podstawie przeprowadzonych eksperymentów można stwierdzić, że wprowadzenie do algorytmu genetycznego do wykrywania wartości odstających modyfikacji *Epi\_2* redukuje liczbę generacji potrzebną do uzyskania optymalnego wyniku (wykrycia wszystkich wartości odstających w zbiorach danych). Wprowadzanie modyfikacji *Epi\_2* wpływa też na czas działania algorytmu, wydłużając czas potrzebny na przeprowadzenie operacji dla pojedynczej generacji algorytmu.

#### 4.5.5 Operacja *Epi\_3*

W podrozdziale tym przedstawione zostały wyniki eksperymentów nad zastosowaniem proponowanej dodatkowej operacji *Epi\_3* w algorytmie genetycznym do wykrywania danych

odstających (*GADO*). Sprawdzenia wpływu na liczbę generacji przy zastosowaniu operacji *Epi\_3* w algorytmie *GADO Epi\_3* względem *GADO* dokonano dla różnych prawdopodobieństw wystąpienia modyfikacji. Wyniki porównania liczby generacji potrzebnych do wykrycia wszystkich wartości odstających w testowych zbiorach danych przedstawiono w Tabeli 4.30.

**Tabela 4.30:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji *Epi\_3* (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania					
	zbiór 1	zbiór 2	zbiór 3	zbiór 4	zbiór 5	zbiór 6
0% (bez modyfikacji)	2675	2175	661	740	1809	1479
5%	2501	2112	639	729	1797	1466
10%	2497	2085	<b>627</b>	688	1781	1419
20%	2456	1658	645	<b>656</b>	1728	1325
30%	<b>2445</b>	<b>1107</b>	658	692	<b>1693</b>	<b>1230</b>
40%	2498	1243	672	778	1699	1289
50%	2567	1265	679	794	1756	1355
60%	2623	1639	689	824	1805	1417
70%	2688	1912	719	833	1922	1482
80%	2731	1958	733	851	1919	1555
90%	2789	1986	779	853	1955	1579
100%	2834	2233	803	894	2098	1602

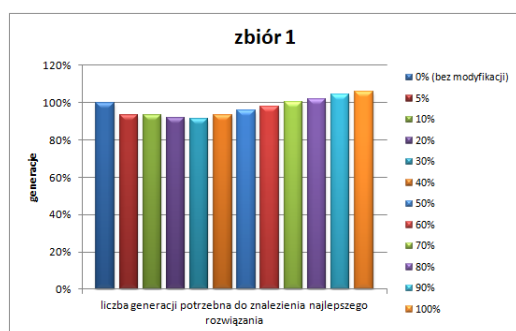
Na podstawie Tabeli 4.30 można zauważyć, że wprowadzenie z odpowiednim prawdopodobieństwem modyfikacji *Epi\_3* do *GADO* redukuje liczbę generacji potrzebną do wykrycia wszystkich wartości odstających w zbiorach testowych.

## Empiryczny dobór optymalnego prawdopodobieństwa wystąpienia modyfikacji

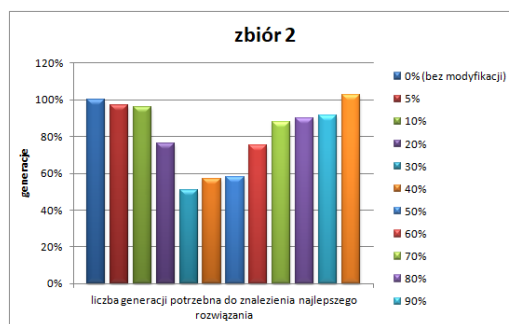
### *Epi\_3*

Na Rysunku 4.18 zaprezentowane zostały procentowe wartości mediany liczby generacji dla algorytmów *GADO Epi\_3* (dla różnego prawdopodobieństwa wystąpienia modyfikacji *Epi\_3*), względem algorytmu *GADO*.

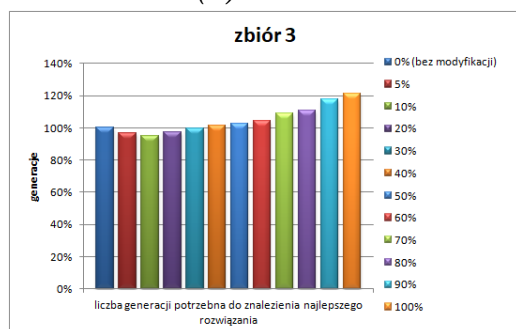
Na podstawie Rysunku 4.18 można stwierdzić, że przy odpowiednim prawdopodobieństwie wystąpienia operacji *Epi\_3* uzyskuje się redukcję liczby generacji w *GADO*. Największą poprawę efektywności działania algorytmu genetycznego (czyli największą redukcję liczby generacji w algorytmie) można zaobserwować w przypadku zbioru 2. Dla zbioru 2 udało się uzyskać redukcję liczby generacji na poziomie około 50%, przy prawdopodobieństwie wystąpienia modyfikacji *Epi\_3* na poziomie 30% (wykres 4.18b). Należy jednak zwrócić uwagę również na fakt, że zbyt duże prawdopodobieństwo wystąpienia modyfikacji *Epi\_3* może doprowadzić do po-



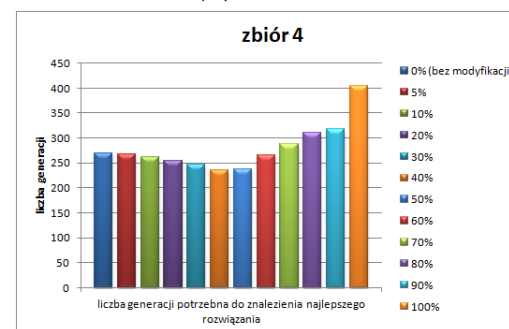
(a) zbiór 1



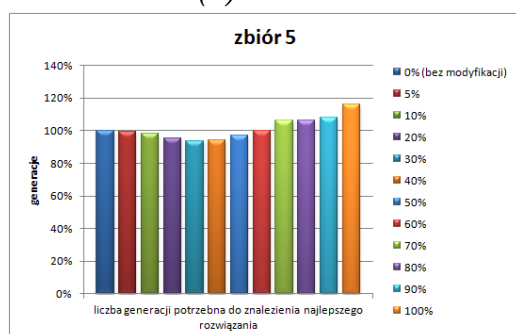
(b) zbiór 2



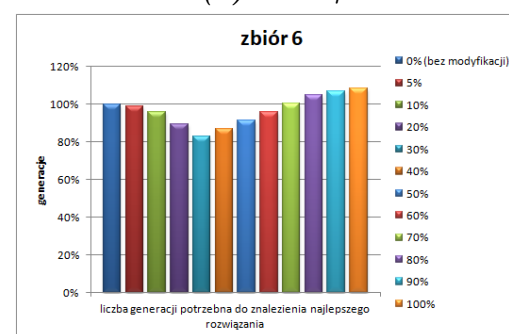
(c) zbiór 3



(d) zbiór 4



(e) zbiór 5



(f) zbiór 6

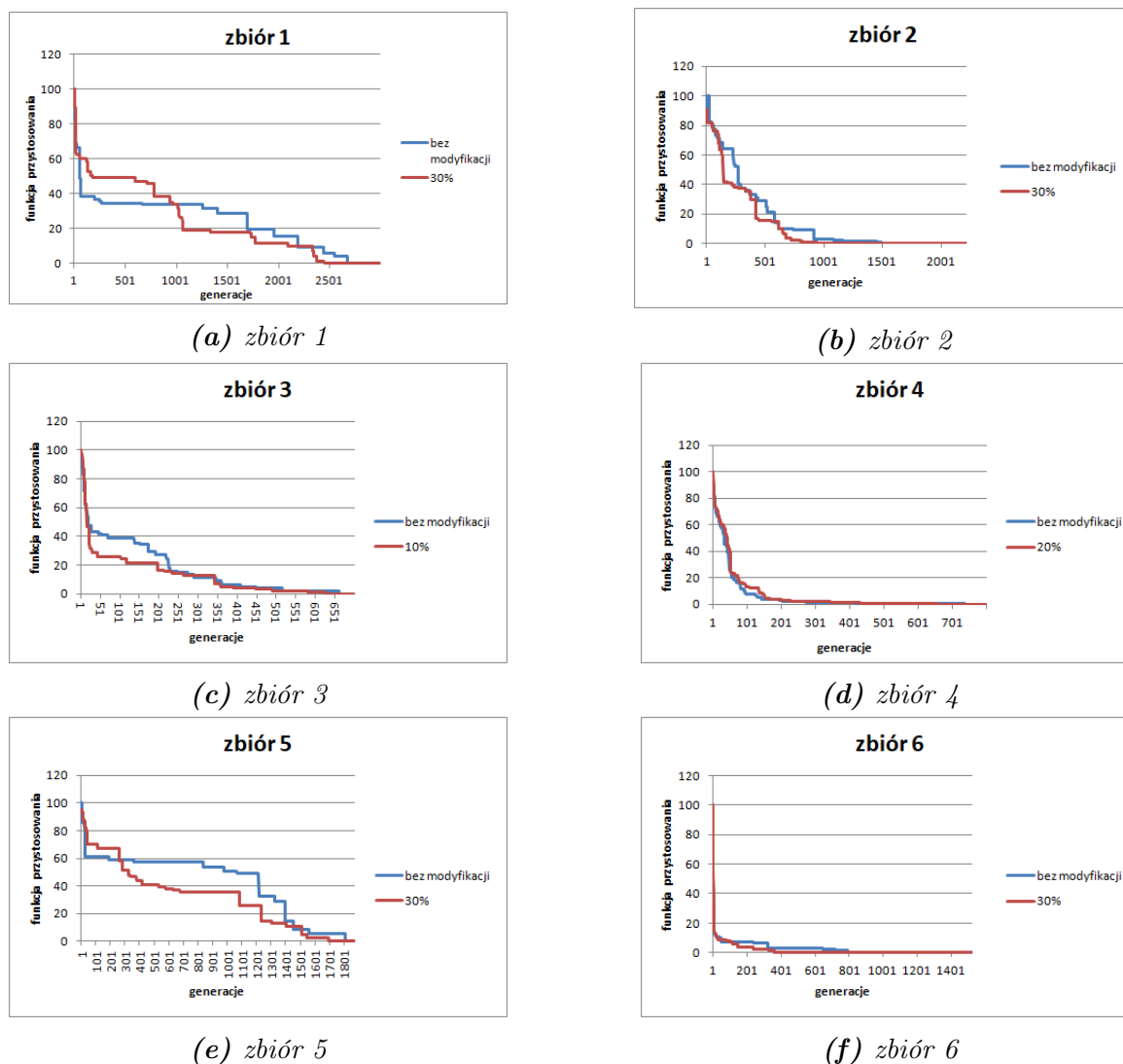
**Rysunek 4.18:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji  $Epi\_3$

gorszenia efektywności działania algorytmu genetycznego, poprzez zwiększenie liczby generacji w  $GADO$ . Zwiększenie liczby generacji w  $GADO$  może mieć wpływ na czas działania  $GADO$ .

### Porównanie efektywności działania $GADO Epi\_3$ z $GADO$

Dla optymalnego prawdopodobieństwa wystąpienia modyfikacji  $Epi\_3$  w  $GADO Epi\_3$  dokonano porównania tempa zmiany wartości funkcji przystosowania względem tempa zmiany wartości funkcji przystosowania dla  $GADO$ . Wyniki porównania zostały przedstawione na Rysunku 4.19.

Na podstawie Rysunku 4.19 można zauważyć, że w przypadku zbiorów 1, 3 i 5 istnieje spora różnica w tempie zmiany wartości funkcji przystosowania w  $GADO Epi\_3$  względem  $GADO$ . W tych zbiorach znacznie szybciej zmieniają się wartości funkcji przystosowania najlepszego osobnika w algorytmie z modyfikacją  $Epi\_3$ , co ostatecznie prowadzi do redukcji liczby



**Rysunek 4.19:** Zmiana wartości funkcji przystosowania dla generacji algorytmu genetycznego, w zależności od zastosowania operacji *Epi\_3* i bez niej

generacji potrzebnych do wykrycia wszystkich wartości odstających (wykresy 4.19a, 4.19c, oraz 4.19e). Dla pozostałych zbiorów testowych różnice w tempie zmiany wartości funkcji przystosowania pomiędzy *GADO*, a *GADO Epi\_3*, nie są już aż tak wyraźne, ale mimo to *GADO Epi\_3* i tak szybciej (w mniejszej liczbie generacji) wykrywa wszystkie wartości odstające w zbiorach testowych, niż *GADO*.

### Porównanie czasów działania *GADO* i *GADO Epi\_3*

W Tabeli 4.31 przedstawiono średnie wartości czasu i wartości odchylenia standardowego (dla poszczególnych zbiorów testowych) dla operacji wykonywanych w pojedynczej generacji *GADO Epi\_3* i algorytmu *GADO*. W ostatniej kolumnie tabeli przedstawiono o ile procentowo wydłużył się czas potrzebny na przeprowadzenie operacji w pojedynczej generacji w *GADO Epi\_3* względem *GADO Epi\_3*.

Na podstawie Tabeli 4.31 można zauważyć wzrost czasu potrzebnego do przeprowadzenie



**Tabela 4.31:** Porównanie czasów pojedynczej generacji w *GADO* i *GADO Epi\_3*

zbiór	czas [s]				zmiana %
	<i>GADOEpi_3</i>		<i>GADO</i>		
	średnia	sd	średnia	sd	
zbiór 1	0,592	0,030	0,567	0,136	4,18%
zbiór 2	0,356	0,054	0,331	0,035	7,22%
zbiór 3	0,246	0,050	0,225	0,027	8,35%
zbiór 4	0,170	0,215	0,161	0,018	5,50%
zbiór 5	1,152	0,285	1,135	0,077	1,48%
zbiór 6	0,355	0,095	0,341	0,014	3,75%

operacji w pojedynczej generacji algorytmu genetycznego w *GADO Epi\_3* względem *GADO*, średnio o 5,08%. Największy średni wzrost czasu zaobserwować można w przypadku zbioru 3 (wzrost o 8,35%), a najmniejszy w przypadku zbioru 5 (wzrost o 1,48%).

Na podstawie przeprowadzonych eksperymentów stwierdzono, że zastosowanie z odpowiednim prawdopodobieństwem operacji *Epi\_3* w *GADO* prowadzi do redukcji liczby generacji w *GADO*. Redukcja liczby generacji zmniejsza liczbę powtórzeń operacji w *GADO* takich jak selekcja, krzyżowanie, mutacja. Wprowadzenie modyfikacji *Epi\_3* do algorytmu *GADO* wpływa jednak na wydłużenie czasu operacji w pojedynczej generacji algorytmu.

#### 4.5.6 Podsumowanie wyników

W podrozdziale przedstawiono wyniki badań nad oceną skuteczności zastosowania trzech proponowanych w pracy operacji naśladujących procesy epigenetyczne w algorytmie *GADO*. W ramach przeprowadzonych badań dokonano doboru optymalnych prawdopodobieństw wystąpienia proponowanych w pracy dodatkowych operacji (*Epi*) algorytmu genetycznego. Ustalone empirycznie optymalne prawdopodobieństwo wystąpienia operacji *Epi* w algorytmie *GADO* przedstawiono w Tabeli 4.32.

**Tabela 4.32:** Optymalne prawdopodobieństwo wystąpienia operacji *Epi* w algorytmie *GADO*

Modyfikacja	Optymalne prawdopodobieństwo
<i>Epi_1</i>	20%
<i>Epi_2</i>	20%
<i>Epi_3</i>	30%

Optymalne prawdopodobieństwo wystąpienia operacji *Epi\_1* i *Epi\_2* w algorytmie do grupowania danych wynosi 20% (tabela 4.32). Dla operacji *Epi\_3* optymalne prawdopodobieństwo wystąpienia wynosi 30%.

W Tabeli 4.33 przedstawiono procentową wartość średniej redukcji liczby generacji potrzebnych

do uzyskania optymalnego wyniku w algorytmach *GADO Epi\_x*, względem algorytmu *GADO*. Tabela 4.33 zawiera również informację o ile procent zastosowanie operacji *Epi* w algorytmie *GADO Epi\_x* wydłużyło czas pojedynczej generacji względem algorytmu *GADO*.

**Tabela 4.33:** Procentowa wartość średniej redukcji liczby generacji oraz procentowe wydłużenie czasu pojedynczej generacji algorytmu *GADO Epi\_x* względem *GADO*

Proponowana operacja	średnia redukcja liczby generacji	średnie % wydłużenie czasu pojedynczej generacji
<i>Epi_1</i>	↓ 17%	↑ 7,50%
<i>Epi_2</i>	↓ 27%	↑ 3,98%
<i>Epi_3</i>	↓ 16%	↑ 5,08%

Zastosowanie proponowanych w pracy dodatkowych operacji bazujących na procesach epigenetycznych w algorytmie genetycznym do wykrywania danych odstających poprawia działanie tego algorytmu (poprzez redukcję liczby generacji potrzebnych do uzyskania oczekiwanego wyniku, por. Tabela 4.33. Największą redukcję liczby generacji uzyskano przy zastosowaniu w algorytmie *GADO* modyfikacji *Epi\_2*, średnia redukcja wynosiła 27%. Zastosowanie dodatkowych operacji wiąże się też z wydłużeniem czasu potrzebnego na przeprowadzenie wszystkich operacji algorytmu genetycznego w pojedynczej generacji algorytmu. Najmniej wydłużył się czas operacji w pojedynczej generacji w przypadku zastosowania modyfikacji *Epi\_2* (Tabela 4.33), wzrost o 3,98%. Najwięcej w przypadku zastosowania operacji *Epi\_1*, wzrost o 7,50%. Na podstawie badań nad możliwością zastosowania proponowanych w pracy dodatkowych operacji w algorytmie do wykrywania danych odstających najlepsze rezultaty uzyskano przy zastosowaniu operacji *Epi\_2*. Uzyskano największą redukcję liczby generacji, przy najmniejszym wzroście czasu operacji w pojedynczej generacji.

## 4.6 Algorytm dopasowania sekwencji biologicznych

Ostatnim algorytmem dla którego sprawdzono możliwość dodatkowych operacji proponowanych w pracy (*Epi\_x*) jest algorytm dopasowania sekwencji biologicznych (*GAMSA*). Tak jak w przypadku poprzednich eksperymentów badania polegały na dołożeniu proponowanych operacji do bazowego algorytmu do porównywania sekwencji biologicznych. Otrzymane wyniki badań przy użyciu algorytmu genetycznego *GAMSA* z dodatkowymi operacjami *Epi* (*GAMSA Epi\_x*) porównano z algorytmem bazowym (*GAMSA*).

### 4.6.1 Dane wykorzystane w eksperymentach

W eksperymentach sprawdzających możliwość zastosowania dodatkowych, zaproponowanych w pracy, operacji w algorytmach genetycznych naśladowujących procesy epigenetyczne dla algorytmu *GAMSA* wykorzystano cztery zbiory sekwencji biologicznych. Każdy zbiór sekwencji zawiera trzy sekwencje biologiczne, dla których wyszukiwane było najlepsze dopasowanie. Zbiory sekwencji wraz z ich charakterystyką przedstawiono w Tabeli 4.34.

**Tabela 4.34:** Zbiory sekwencji biologicznych wykorzystane w eksperymentach

zbiór	Rodzaj sekwencji	Liczba sekwencji w zbiorze	Długość sekwencji	ID w bazach NCBI
zbiór 1	DNA	3	686	KR362841 [153]
			280	KT004640 [75]
			493	KT198657 [169]
zbiór 2	białko	3	188	ACI96295 [12]
			190	AIG05829 [113]
			188	NP619757 [138]
zbiór 3	białko	3	285	AAC62750 [35]
			253	AAH12844 [159]
			254	AIG55221 [166]
zbiór 4	cRNA	3	856	GU585905 [99]
			624	FJ434381 [28]
			535	KX290116 [106]

Wszystkie wykorzystane w eksperymentach sekwencje pochodziły z ogólnodostępnych baz białkowych i nukleotydowych dostępnych na stronach NCBI (National Center for Biotechnology Information, <https://www.ncbi.nlm.nih.gov>). Dwa zbiory sekwencji są to sekwencje zawierają po 3 sekwencje białkowe (zbiór 2 i 3), zbiór 1 – zawiera 3 sekwencje DNA, a zbiór 4 – 3 sekwencje cRNA.

### 4.6.2 Parametry algorytmu

Parametry, które zostały ustawione w algorytmie *GAMSA* i dla przeprowadzonych eksperymentów, zarówno w algorytmie bazowym jak i w algorytmach *GAMSA Epi\_x* przedstawiono w Tabeli 4.35. Wartości parametrów zostały dobrane na podstawie literatury.

W algorytmie *GAMSA* wykorzystanym w eksperymentach, użyto dwóch metod krzyżowania. Prawdopodobieństwo mutacji zostało ustawione na wartość średnią zalecanych wartości spotykanych w literaturze.

**Tabela 4.35:** Parametry algorytmu genetycznego wykorzystane w eksperymentach

Parametr	Wartość
liczba osobników w populacji	10% długości najdłuższej sekwencji
prawdopodobieństwo krzyżowania horyzontalnego i wertykalnego	70%
prawdopodobieństwo mutacji	75%
metoda selekcji osobników	turniejowa
warunek stopu	warunkowana długością zbioru liczba generacji bez poprawy wartości funkcji przystosowania

### 4.6.3 Operacja *Epi\_1*

W niniejszym podrozdziale przedstawione zostały wyniki eksperymentów przeprowadzonych w celu sprawdzenia, czy uzupełnienie algorytmu *GAMSA* operacją *Epi\_1* poprawia efektywność tego algorytmu (*GAMSA Epi\_1*). W Tabeli 4.36 zaprezentowano liczbę generacji potrzebną do znalezienia optymalnego dopasowania sekwencji biologicznych w *GAMSA*, oraz w *GAMSA Epi\_1* z różnym prawdopodobieństwem wystąpienia operacji *Epi\_1*.

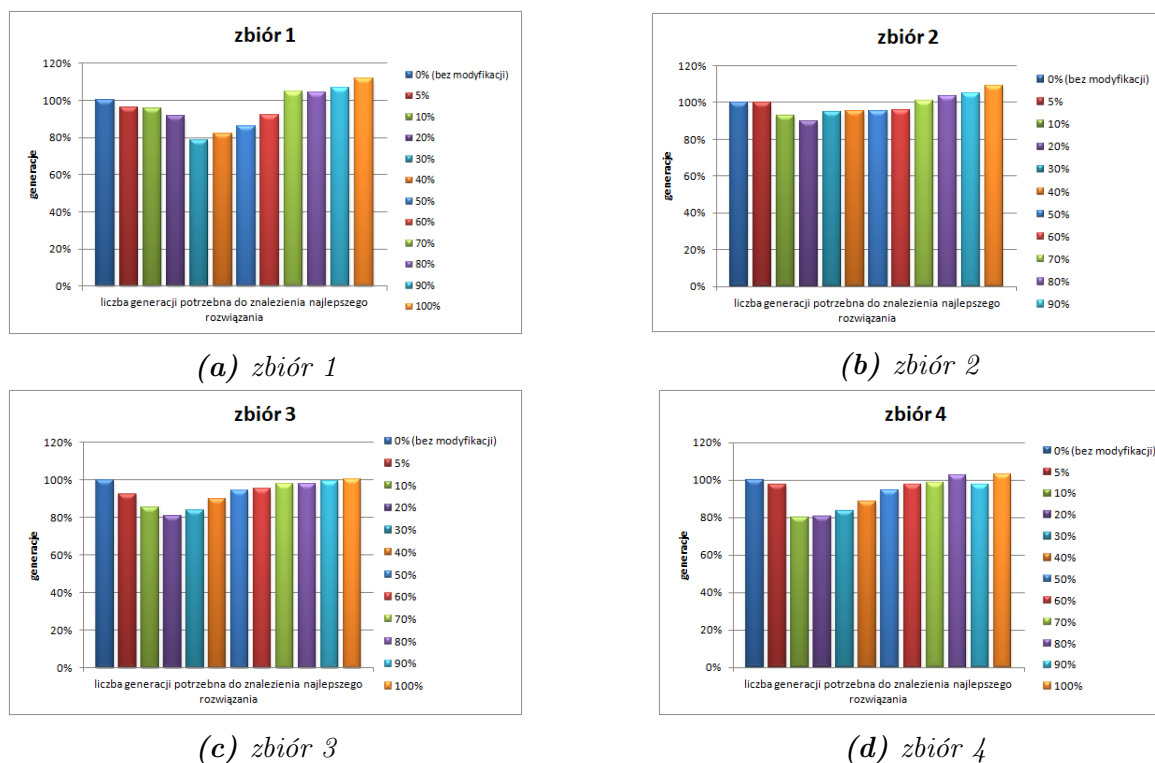
**Tabela 4.36:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji *Epi\_1* (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia operacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania			
	zbiór 1	zbiór 2	zbiór 3	zbiór 4
0% (bez modyfikacji)	1375	1350	1488	1415
5%	1322	1348	1375	1378
10%	1313	1255	1272	<b>1134</b>
20%	1262	<b>1210</b>	<b>1207</b>	1144
30%	<b>1081</b>	1280	1246	1183
40%	1127	1285	1339	1251
50%	1185	1288	1408	1336
60%	1268	1297	1419	1384
70%	1442	1360	1456	1394
80%	1430	1400	1458	1449
90%	1467	1416	1476	1382
100%	1537	1470	1498	1457

Liczba generacji przedstawiona w tabeli 4.36 stanowi wartość mediany dla 9 powtórzeń działania GA.

### Empiryczny dobór optymalnego prawdopodobieństwa wystąpienia operacji *Epi\_1*

Na Rysunku 4.20 przedstawione zostały procentowe wartości liczby generacji dla *GAMSA Epi\_1* w stosunku do liczby generacji *GAMSA*.



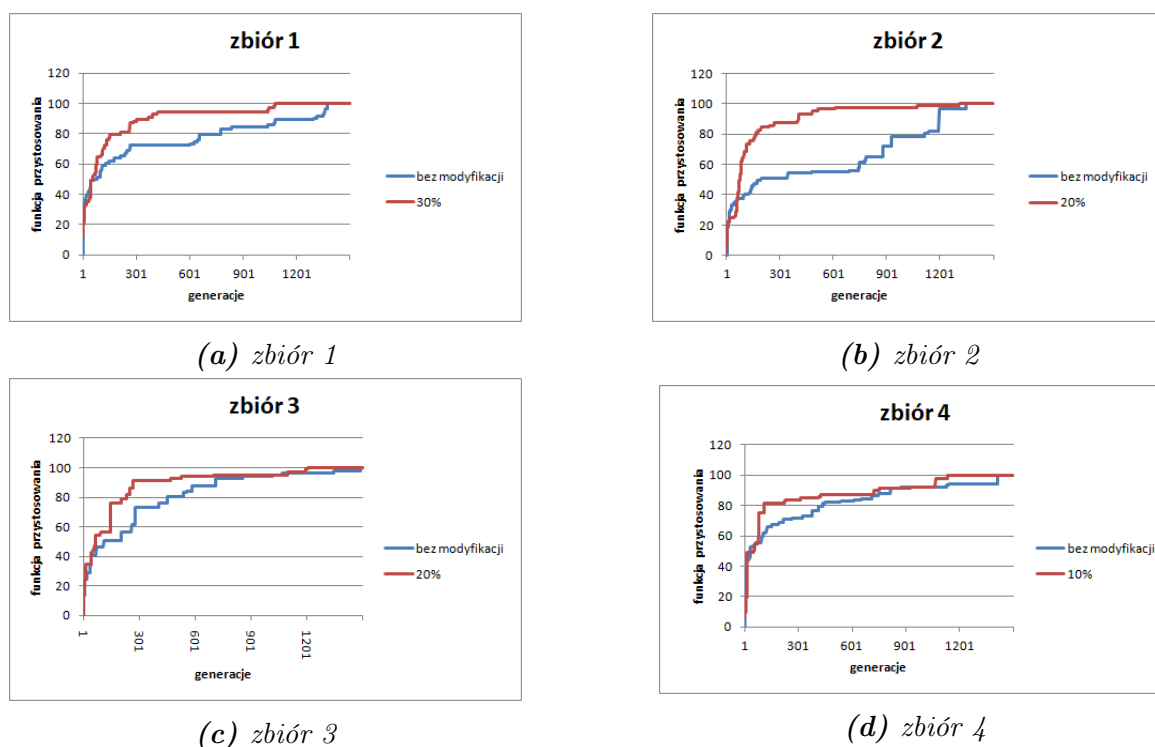
**Rysunek 4.20:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów sekwencji testowych w zależności od prawdopodobieństwa wystąpienia operacji  $Epi_1$

Na podstawie wykresów na Rysunku 4.20 można zauważyć redukcję liczby generacji dla  $GAMSA_{Epi_1}$  w stosunku do  $GAMSA$ , dla prawdopodobieństw wystąpienia modyfikacji poniżej 60%. Największy spadek liczby generacji uzyskano w przypadku 30% prawdopodobieństwa wystąpienia modyfikacji dla zbioru 1 (wykres 4.20a). Najmniejszy wpływ wprowadzenie operacji  $Epi_1$  do algorytmu  $GAMSA$  miało miejsce w przypadku zbioru 2 sekwencji. Przy zbyt wysokim prawdopodobieństwie wystąpienia modyfikacji  $Epi_1$  w  $GAMSA_{Epi_1}$ , można zaobserwować zwiększenie liczby generacji potrzebnych do optymalnego dopasowania sekwencji względem  $GAMSA$ .

### Porównanie efektywności działania $GAMSA_{Epi_1}$ z $GAMSA$

Dla optymalnego prawdopodobieństwa wystąpienia operacji  $Epi_1$  w algorytmie  $GAMSA_{Epi_1}$  dokonano porównania zmiany wartości funkcji przystosowania dla algorytmu  $GAMSA_{Epi_1}$  i  $GAMSA$ . Wyniki tego porównania zostały zaprezentowane na Rysunku 4.21.

Na Rysunku 4.21 można zauważyć, że w przypadku wszystkich zbiorów sekwencji biologicznych uzyskano zwiększenie dynamiki zmiany wartości funkcji przystosowania dla algorytmu  $GAMSA_{Epi_1}$  względem algorytmu  $GAMSA$ . Największą różnicę w dynamice zmiany wartości funkcji przystosowania można zaobserwować dla zbioru 2 sekwencji (wykres 4.21b). Najmniejsze różnice w zmianach wartości funkcji przystosowania można zaobserwować w przy-



**Rysunek 4.21:** Wartość funkcji przystosowania dla kolejnych generacji algorytmu GAMSA, w zależności od wystąpienia operacji  $Epi\_1$  i bez niej

padku zbioru 4 (wykres 4.21d).

### Porównanie czasów

W Tabeli 4.37 przedstawiono średnie wartości pomiaru czas i wartości odchylenia standardowego dla operacji przeprowadzanych w GAMSA i GAMSA  $Epi\_1$  w pojedynczej generacji. Dodatkowo w ostatniej kolumnie znajduje się informacja o procentowej wartości średniej zmiany czasu trwania pojedynczej generacji w porównywanych algorytmach genetycznych.

**Tabela 4.37:** Porównanie czasów pojedynczej generacji w algorytmie GAMSA i GAMSA  $Epi\_1$

zbiory	czas [s]				zmiana %
	GAMSA $Epi\_1$		GAMSA		
	średnia	sd	średnia	sd	
zbiór 1	0,737	0,035	0,728	0,059	1,15%
zbiór 2	0,310	0,028	0,285	0,040	8,14%
zbiór 3	0,400	0,043	0,367	0,032	8,33%
zbiór 4	0,899	0,064	0,881	0,209	2,01%

Na podstawie Tabeli 4.37 zaobserwowano wydłużenie czasu działania operacji w ramach pojedynczej generacji w algorytmie GAMSA  $Epi\_1$ , względem algorytmu GAMSA. Średnia wartość wydłużenia czasu działania wynosiła około 4,9%. Najmniejszą różnicę w czasie wykonywania operacji w pojedynczej generacji GAMSA  $Epi\_1$  względem GAMSA można

zaobserwować w przypadku zbioru 1, zmiana wynosi 1,15%. Największą zmianę, wynoszącą 8,33%, zaobserwowano w przypadku zbioru 3.

Na podstawie otrzymanych wyników autor stwierdza, że w przypadku algorytmu *GAMSA Epi\_1* zaobserwowano poprawę działania tego algorytmu w stosunku do *GAMSA*. Poprawa działania polega na zmniejszeniu liczby generacji algorytmu potrzebnych do prawidłowego dopasowania sekwencji biologicznych.

#### 4.6.4 Modyfikacja *Epi\_3*

W podrozdziale tym zaprezentowane zostały wyniki eksperymentów mające na celu sprawdzenie kolejnej zaproponowanej w rozprawie modyfikacji algorytmów genetycznych naśladującej proces epigenetyczny metylacji cytozyny (*Epi\_2*). Badania miały na celu sprawdzenie czy zastosowanie modyfikacji *Epi\_2* poprawia efektywność działania algorytmu do dopasowywania sekwencji biologicznych (*GAMSA*).

Zestawienie liczby generacji potrzebnych do znalezienia optymalnego dopasowania testowych sekwencji biologicznych dla algorytmu *GAMSA* i *GAMSA Epi\_2* różnym prawdopodobieństwem wystąpienia modyfikacji *Epi\_2* przedstawiono w Tabeli 4.38.

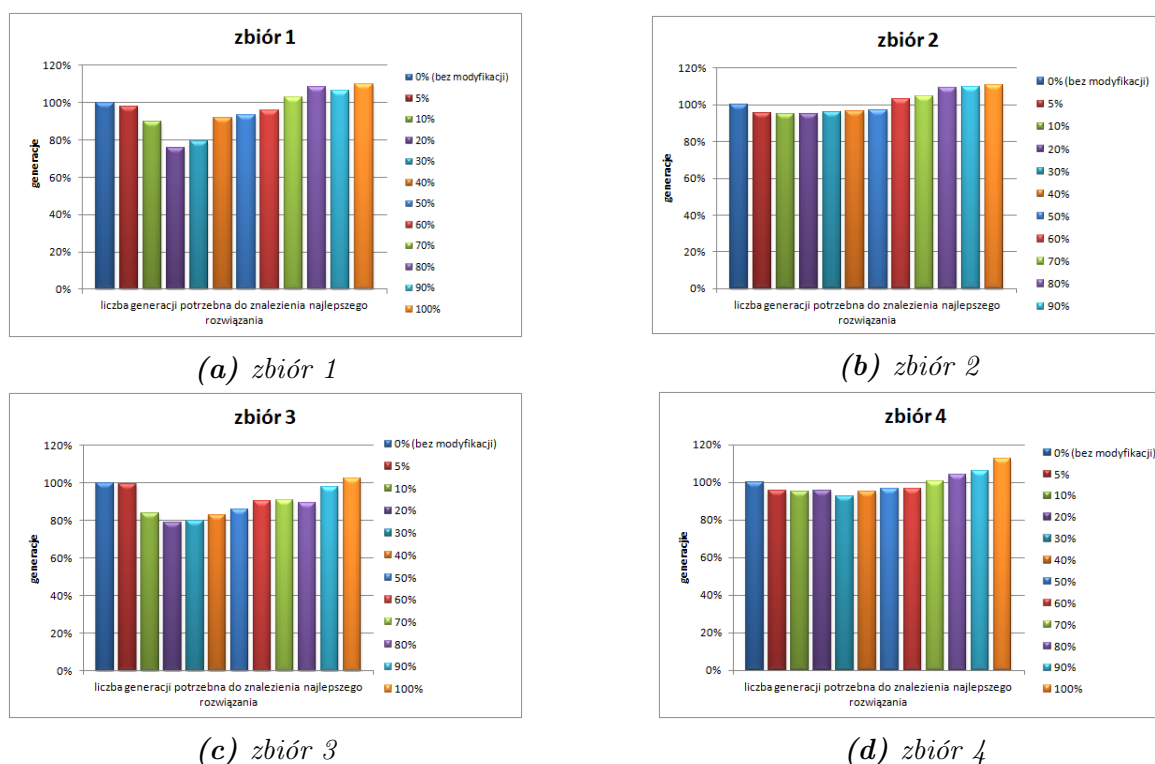
**Tabela 4.38:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia modyfikacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania			
	zbiór 1	zbiór 2	zbiór 3	zbiór 4
0% (bez modyfikacji)	1375	1350	1488	1415
5%	1346	1292	1480	1354
10%	1238	1282	1249	1343
20%	<b>1044</b>	<b>1280</b>	<b>1173</b>	1350
30%	1089	1297	1189	<b>1314</b>
40%	1265	1305	1231	1344
50%	1281	1309	1277	1369
60%	1320	1389	1347	1370
70%	1413	1413	1353	1421
80%	1492	1473	1334	1475
90%	1464	1483	1458	1499
100%	1509	1492	1524	1591

Na podstawie Tabeli 4.38 można zauważyć, że zastosowanie modyfikacji *Epi\_2* z odpowiednim prawdopodobieństwem (ok. 20%) pozwala osiągnąć redukcję liczby generacji potrzebnych do znalezienia optymalnego dopasowania sekwencji przez *GAMSA Epi\_2* w stosunku do *GAMSA*.

## Empiryczne wyznaczenie optymalnego prawdopodobieństwa wystąpienia modyfikacji $Epi\_2$

Na Rysunku 4.22 przedstawione zostały procentowe wartości liczby generacji potrzebnych do uzyskania optymalnego dopasowania testowych zbiorów sekwencji biologicznych. Jako wartość 100% ustalona została liczba generacji potrzebna do dopasowania sekwencji dla algorytmu GAMSА bez modyfikacji. Dla zGAMSА  $Epi\_2$  różnymi wartościami prawdopodobieństwa wystąpienia modyfikacji  $Epi\_2$  przedstawiono jaki procent z liczby generacji algorytmu GAMSА potrzebnych było do uzyskania optymalnego dopasowania sekwencji.



**Rysunek 4.22:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji  $Epi\_2$

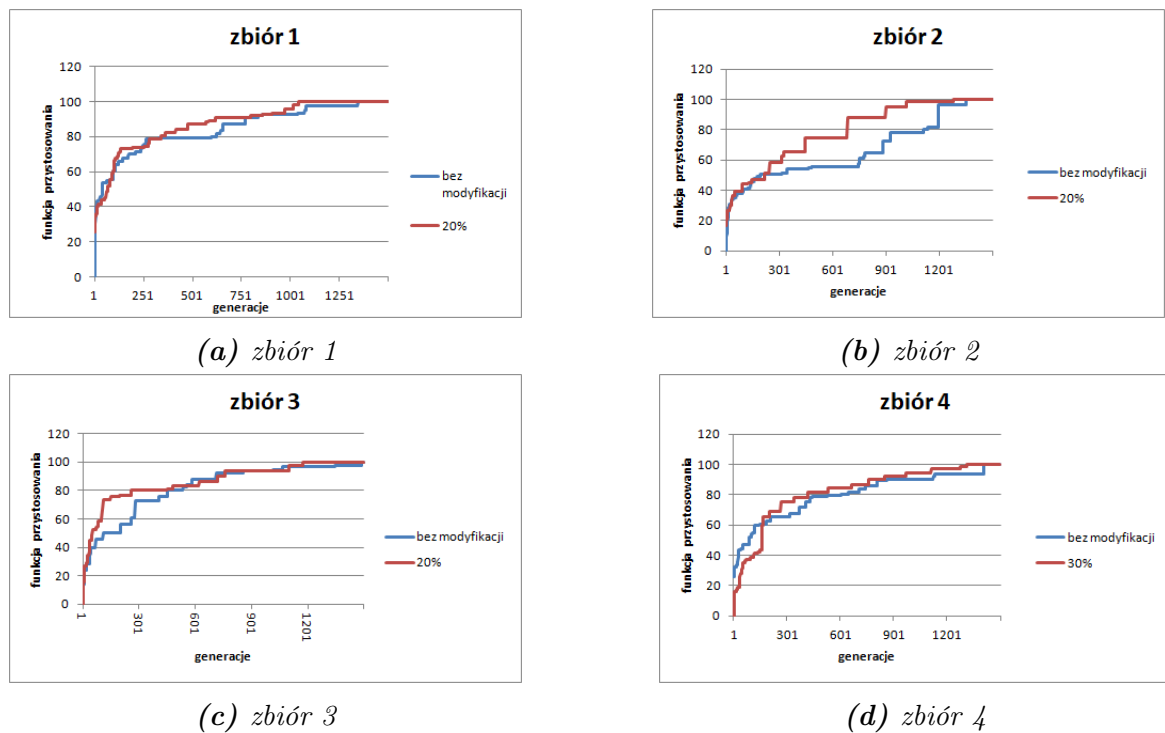
Największe procentowe zmniejszenie liczby generacji uzyskano w przypadku zbioru 1 testowych sekwencji biologicznych (wykres 4.22a), dla prawdopodobieństwa wystąpienia modyfikacji na poziomie 20%. Najmniejsze procentowe różnice w liczbie generacji uzyskano w przypadku zbioru 4 sekwencji biologicznych (wykres 4.22d).

### Porównanie GAMSА $Epi\_2$ z GAMSА

Dla prawdopodobieństw, dla których uzyskano dla danego zbioru sekwencji największą redukcję liczby generacji, dokonano porównania zmiany wartości funkcji przystosowania wraz z kolejnymi generacjami GAMSА  $Epi\_2$ . Porównania tego dokonano ze zmianą wartości funkcji



przystosowania w algorytmie GAMSA . Wyniki tego porównania dla poszczególnych testowych zbiorów sekwencji biologicznych przedstawiono na Rysunku 4.23.



**Rysunek 4.23:** Zmiana wartości funkcji przystosowania wraz z kolejnymi generacjami algorytmu genetycznego, w zależności od zastosowania modyfikacji *Epi\_2* i bez niej

Na Rysunku 4.23 można zauważyć szybszą zmianę wartości funkcji przystosowania najlepszego osobnika wraz z kolejnymi generacjami w przypadku algorytmu *GAMSA Epi\_2* względem *GAMSA* . Największą różnicę w tempie zmian wartości funkcji przystosowania można zaobserwować w przypadku zbioru 2 sekwencji biologicznych (wykres 4.23b). Najmniejszą różnicę w zmianie wartości funkcji przystosowania pomiędzy *GAMSA* , a *GAMSA Epi\_2* można zaobserwować w przypadku zbioru 4 (wykres 4.23d). W przypadku zbioru 4 wprowadzenie modyfikacji *Epi\_2* do algorytmu *GAMSA* miało też spośród wszystkich zbiorów testowych najmniejszy wpływ na redukcję liczby generacji potrzebnych do znalezienia optymalnego dopasowania sekwencji (wykres 4.22d).

### Porównanie czasów

W Tabeli 4.39 przedstawiono średni czas i wartość odchylenia standardowego dla pojedynczej generacji algorytmu genetycznego *GAMSA Epi\_2* i algorytmu *GAMSA* . W ostatniej kolumnie tabeli pokazano wartość procentową zmiany czasu, to znaczy o ile procentowo wydłużył się czas operacji w algorytmie *GAMSA Epi\_2* względem algorytmu *GAMSA* , dla pojedynczej generacji algorytmu.

**Tabela 4.39:** Porównanie czasów pojedynczej generacji w *GAMSA* i *GAMSA Epi\_2*

zbiór	czas [s]				zmiana %
	<i>GAMSA Epi_2</i>		<i>GAMSA</i>		
	średnia	sd	średnia	sd	
zbiór 1	0,739	0,058	0,728	0,059	1,48%
zbiór 2	0,290	0,026	0,285	0,040	1,73%
zbiór 3	0,372	0,029	0,367	0,032	1,31%
zbiór 4	0,966	0,087	0,881	0,209	8,80%

Na podstawie Tabeli 4.39 można zaobserwować, że wprowadzenie modyfikacji *Epi\_2* do algorytmu *GAMSA* powoduje wydłużenie czasu operacji w pojedynczej generacji algorytmu. Średnio czas pojedynczej generacji wzrasta o 3,33% dla algorytmu *GAMSA Epi\_2*. Najmniejszy wzrost czasu można zaobserwować w przypadku zbioru 3 sekwencji (wzrost o 1,31%), największy w przypadku zbioru 4 sekwencji (wzrost o 8,80%).

Na podstawie przeprowadzonych eksperymentów dla algorytmu *GAMSA* można zaobserwować, redukcję liczby generacji potrzebnych do uzyskania optymalnego dopasowania sekwencji w przypadku zastosowania modyfikacji *Epi\_2*, czyli poprawia się efektywność algorytmu genetycznego. Następuje jednak wydłużenie czasu operacji pojedynczej generacji algorytmu genetycznego, lecz jest ono niewielkie.

#### 4.6.5 Operacja *Epi\_3*

W podrozdziale tym przedstawione zostały wyniki eksperymentów nad zastosowaniem dodatkowej operacji *Epi\_3* w algorytmie *GAMSA*. W pierwszym etapie badań sprawdzono czy wprowadzenie operacji poprawi działanie algorytmu *GAMSA*, poprzez redukcję liczby generacji potrzebnych do uzyskania najlepszego dopasowania sekwencji. W przypadku stwierdzenia poprawy działania algorytmu *GAMSA Epi\_3* w stosunku do *GAMSA*, ważne również było ustalenie optymalnego prawdopodobieństwa wystąpienia operacji *Epi\_3* w algorytmie *GAMSA*. Medianę liczby generacji *GAMSA*, oraz *GAMSA Epi\_3* z różnym prawdopodobieństwem wystąpienia operacji *Epi\_3*, dla poszczególnych zbiorów sekwencji testowych przedstawiono w Tabeli 4.40.

W Tabeli 4.40 pogrubioną kursywą zaznaczone zostały wartości najmniejszej mediany liczby generacji dla danego zbioru przy wykorzystaniu *GAMSA* i *GAMSA Epi\_3* z różnym prawdopodobieństwem wystąpienia operacji *Epi\_3*. Na podstawie Tabeli 4.40 można stwierdzić redukcję liczby generacji przy odpowiednim prawdopodobieństwie wystąpienia operacji w algorytmie *GAMSA Epi\_3* względem *GAMSA*. Optymalna wartość prawdopodobieństwa wystąpienia operacji *Epi\_3*, czyli taka dla której otrzymano największą redukcję liczby generacji

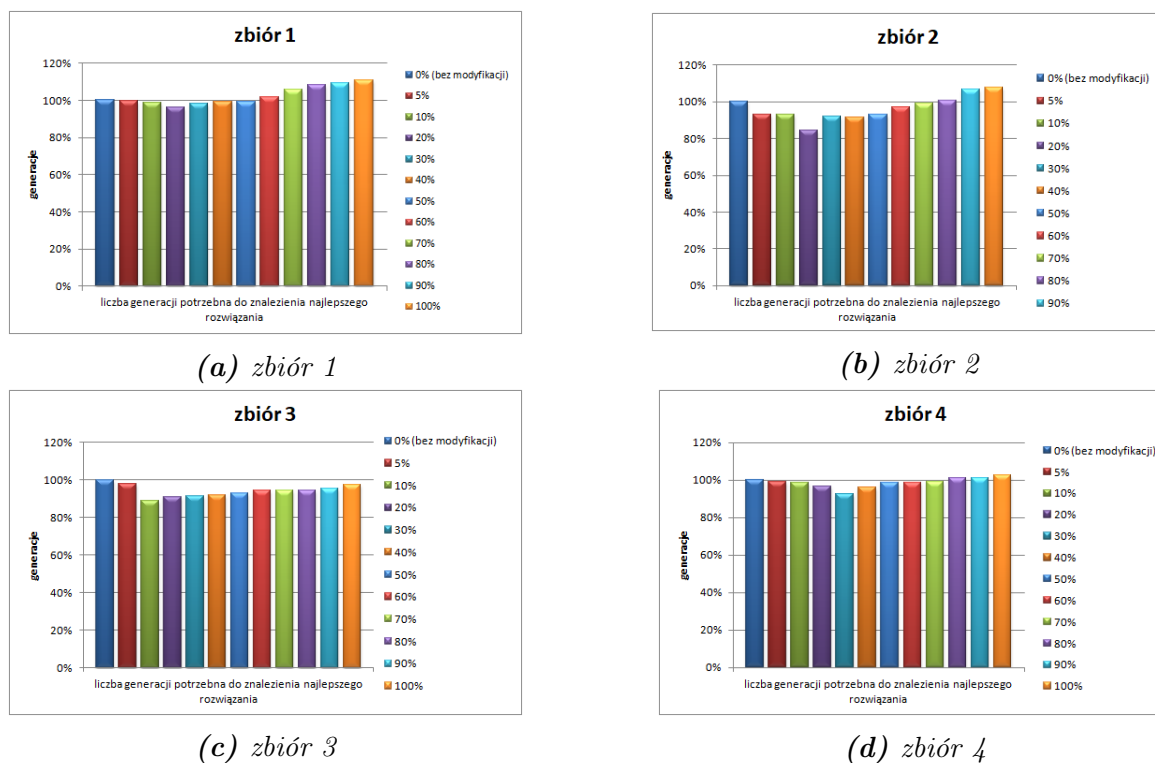
**Tabela 4.40:** Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji *Epi\_3* (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego)

prawdopodobieństwo wystąpienia operacji	liczba generacji potrzebnych do znalezienia najlepszego rozwiązania			
	zbiór 1	zbiór 2	zbiór 3	zbiór 4
0% (bez modyfikacji)	1375	1350	1488	1415
5%	1371	1257	1457	1400
10%	1358	1255	<b>1323</b>	1394
20%	<b>1325</b>	<b>1142</b>	1350	1364
30%	1350	1239	1363	<b>1313</b>
40%	1362	1238	1366	1360
50%	1363	1256	1380	1398
60%	1401	1310	1408	1399
70%	1457	1337	1402	1402
80%	1491	1357	1403	1429
90%	1505	1443	1419	1429
100%	1521	1451	1449	1455

w algorytmie *GAMSA* oscyluje wokół wartości 20%.

### Empiryczny dobór optymalnego prawdopodobieństwa wystąpienia operacji *Epi\_3*

Na Rysunku 4.24 zaprezentowano procentowe wartości liczby generacji w *GAMSA Epi\_3* z różnymi prawdopodobieństwami wystąpienia operacji *Epi\_3* względem *GAMSA*.

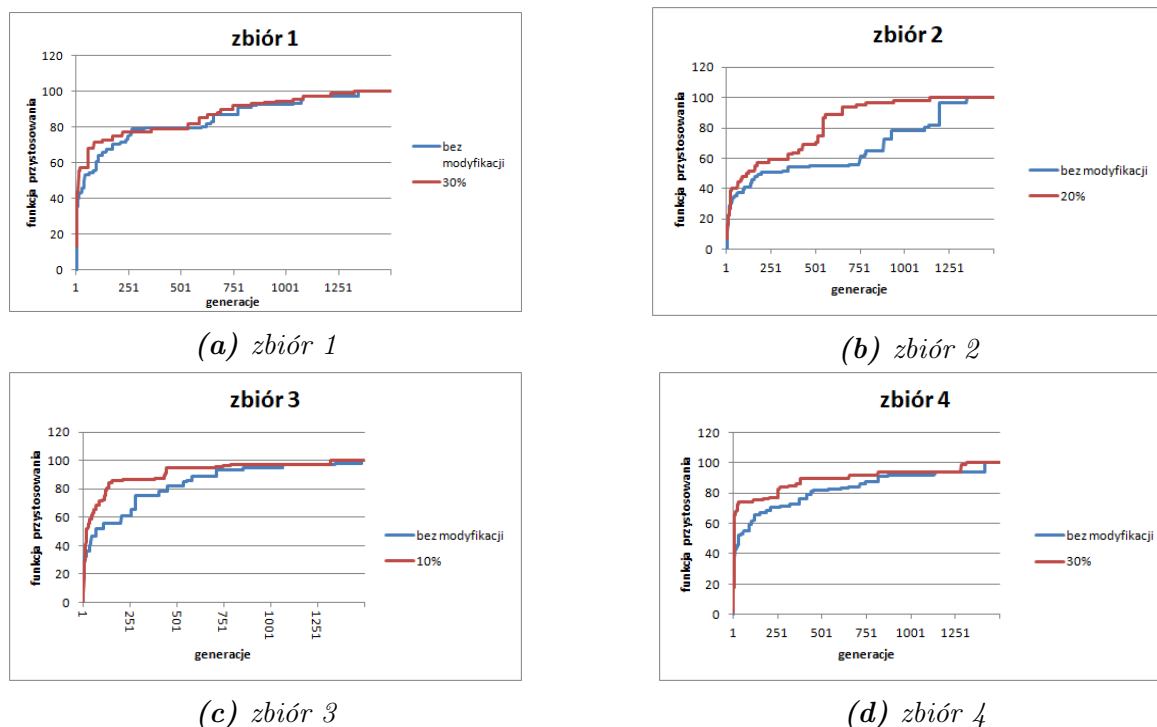


**Rysunek 4.24:** Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji *Epi\_3* w *GAMSA*

Na podstawie Rysunku 4.24 można stwierdzić redukcję liczby generacji w *GAMSA Epi\_3*, przy odpowiednim doborze prawdopodobieństwa wystąpienia modyfikacji *Epi\_3*, względem *GAMSA*. Największą redukcję liczby generacji uzyskano dla prawdopodobieństwa wystąpienia operacji *Epi\_3* na poziomie 20% w zbiorze 2 sekwencji testowych (wykres 4.24b). W przypadku zbyt dużego prawdopodobieństwa wystąpienia modyfikacji *Epi\_3* w *GAMSA Epi\_3* dochodzi do pogorszenia efektywności działania algorytmu, poprzez zwiększenie liczby generacji potrzebnych do uzyskania optymalnego dopasowania sekwencji biologicznych. Zwiększenie liczby generacji algorytmu *GAMSA*, może przekładać się na wydłużenia czasu potrzebnego na znalezienie optymalnego dopasowania sekwencji biologicznych.

### Porównanie efektywności działania *GAMSA Epi\_3* z *GAMSA*

Dla optymalnych wartości prawdopodobieństwa wystąpienia operacji *Epi\_3* w *GAMSA* dla testowych zbiorów sekwencji biologicznych dokonano porównania tempa zmiany wartości funkcji przystosowania najlepszego osobnika w danej generacji dla algorytmu *GAMSA Epi\_3*, z *GAMSA*. Wyniki tego porównania zaprezentowane zostały na Rysunku 4.25.



**Rysunek 4.25:** Zmiana wartości funkcji przystosowania dla generacji algorytmu *GAMSA* i *GAMSA Epi\_3*

Na podstawie Rysunku 4.25 można zauważyć większą dynamikę zmiany wartości funkcji przystosowania najlepszego osobnika wraz z kolejnymi generacjami w algorytmie *GAMSA Epi\_3* w stosunku do bazowego *GAMSA*. Największą różnicę w tempie zmiany wartości funkcji przystosowania najlepszego osobnika można zauważyć w przypadku zbioru 2 sekwencji (wykres

4.25b). Najmniejsze różnice występują w przypadku zbioru 1 sekwencji (wykres 4.25a). Dla wszystkich zbiorów sekwencji *GAMSA Epi\_3* szybciej osiąga wartość funkcji przystosowania odpowiadającej najlepszemu dopasowaniu sekwencji, niż *GAMSA*.

### Porównanie czasów

W Tabeli 4.41 przedstawiono średni czas i wartość odchylenia standardowego dla operacji w pojedynczej generacji dla *GAMSA Epi\_3* i *GAMSA*. W ostatniej kolumnie umieszczono dodatkowo informację o średnim procentowym wzroście czasu potrzebnego do przeprowadzenia operacji w pojedynczej generacji w algorytmie *GAMSA Epi\_3* względem *GAMSA*.

**Tabela 4.41:** Porównanie czasów pojedynczej generacji w algorytmie *GAMSA* i algorytmie *GAMSA Epi\_3*

zbiór	czas [s]				zmiana %
	<i>GAMSA Epi_3</i>		<i>GAMSA</i>		
	średnia	sd	średnia	sd	
zbiór 1	0,749	0,067	0,728	0,059	2,80%
zbiór 2	0,290	0,028	0,285	0,040	1,73%
zbiór 3	0,372	0,033	0,367	0,032	1,31%
zbiór 4	0,896	0,068	0,881	0,209	1,68%

Na podstawie Tabeli 4.41 można zaobserwować wydłużenie się czasu potrzebnego do przeprowadzenia operacji algorytmu genetycznego w pojedynczej generacji *GAMSA Epi\_3* względem *GAMSA* o średnio 1,88%. Największe wydłużenie czasu zaobserwowano w przypadku zbioru 1 sekwencji biologicznych (wzrost o 2,80%), najmniejsze w przypadku zbioru 3 (wydłużenie o 1,31%).

Na podstawie przeprowadzonych eksperymentów można uznać, że zastosowanie z odpowiednim prawdopodobieństwem operacji *Epi\_3* w algorytmie *GAMSA* poprawia jego efektywność poprzez redukcję liczby generacji w algorytmie potrzebnych do uzyskania najlepszego dopasowania sekwencji biologicznych. Zastosowanie operacji *Epi\_3* wpływa niekorzystnie na czas przeprowadzenia operacji algorytmu genetycznego w pojedynczej generacji, wydłużając go.

### 4.6.6 Podsumowanie wyników

Podrozdział ten zawiera wyniki eksperymentów nad oceną skuteczności zastosowania trzech proponowanych w pracy operacji naśladujących procesy epigenetyczne w algorytmie genetycznym do porównywania sekwencji biologicznych (*GAMSA*). W ramach przeprowadzonych badań dokonano doboru optymalnych prawdopodobieństw wystąpienia proponowanych w pracy

dotychczasowych operacji algorytmu genetycznego. Optymalne prawdopodobieństwo wystąpienia proponowanych w pracy operacji w algorytmie *GAMSA* przedstawiono w Tabeli 4.42.

**Tabela 4.42:** Optymalne prawdopodobieństwo wystąpienia operacji *Epi\_x* w algorytmie *GAMSA*

Modyfikacja	Optymalne prawdopodobieństwo
<i>Epi_1</i>	20%
<i>Epi_2</i>	20%
<i>Epi_3</i>	20%

Optymalne prawdopodobieństwo wystąpienia proponowanych operacji *Epi\_x* w algorytmie *GAMSA* wynosiło 20% (Tabela 4.42).

W Tabeli 4.43 przedstawiono procentową wartość średniej redukcji liczby generacji potrzebnych do uzyskania optymalnego wyniku w *GAMSA Epi\_x*, względem algorytmu bazowego *GAMSA*. Tabela 4.43 zawiera również informację o ile procent zastosowanie proponowanych w pracy operacji w algorytmie genetycznym wydłużyło czas pojedynczej generacji algorytmu genetycznego, względem algorytmu bez dodatkowych operacji.

**Tabela 4.43:** Procentowa wartość średniej redukcji liczby generacji oraz procentowe wydłużenie czasu pojedynczej generacji algorytmu *GAMSA Epi\_x*, względem *GAMSA*

Proponowana operacja	średnia redukcja liczby generacji	średnie % wydłużenie czasu pojedynczej generacji
<i>Epi_1</i>	↓ 17%	↑ 4,90%
<i>Epi_2</i>	↓ 14%	↑ 3,33%
<i>Epi_3</i>	↓ 9%	↑ 1,88%

Zastosowanie proponowanych w pracy dodatkowych operacji bazujących na procesach epigenetycznych w algorytmie *GAMSA* poprawia działanie tego algorytmu poprzez redukcję liczby generacji potrzebnych do uzyskania oczekiwanego wyniku, Tabela 4.43. Największą redukcję liczby generacji uzyskano przy uzupełnieniu algorytmu operacją *Epi\_1*, średnia redukcja wyniosła 17%. Zastosowanie dodatkowych operacji wiąże się też z wydłużeniem czasu potrzebnego na przeprowadzenie wszystkich operacji algorytmu genetycznego w pojedynczej generacji algorytmu. Najmniej wydłużył się czas operacji w pojedynczej generacji w przypadku zastosowania operacji *Epi\_3* (Tabela 4.43), wzrost o 1,88%, jednak w przypadku operacji *Epi\_3* uzyskano również najmniejszą redukcję liczby generacji, redukcja o 9%. Największy wzrost czasu pojedynczej generacji zaobserwowano w przypadku dołożenia do algorytmu operacji *Epi\_1*, wzrost o 4,90%.

Na podstawie badań nad możliwością zastosowania proponowanych w pracy dodatkowych operacji w algorytmie *GAMSA* najlepsze rezultaty uzyskano przy zastosowaniu operacji *Epi\_1*.

Uzyskano największą redukcję liczby generacji, jednak przy największym wzroście czasu operacji w pojedynczej generacji.

## 4.7 Podsumowanie eksperymentów

W podrozdziale tym zamieszczono podsumowanie wyników eksperymentów, których rezultaty zamieszczono w rozdziale 4. Przedstawione eksperymenty dotyczyły badań nad celowością zastosowania w algorytmach genetycznych zaproponowanych w pracy nowych operacji inspirowanych procesami epigenetycznymi (zaprezentowanymi w Rozdziale 3). Zaproponowane modyfikacje naśladowały procesy epigenetyczne: dziedziczenia za pomocą prionu (*Epi\_1*), metylację cytozyny (*Epi\_2*) i wyłączenie allelicznego (*Epi\_3*). Algorytmy genetyczne z zaproponowanymi modyfikacjami zostały sprawdzone pod kątem efektywności działania na czterech bazowych algorytmach genetycznych służących do rozwiązania problemu: optymalizacji załadunku (*GAOZ*), grupowania danych (*GAGR*), wykrywania danych odstających (*GADO*), oraz dopasowania sekwencji biologicznych (*GAMSA*) (omówionych w Rozdziale 2). Poprzez poprawę efektywności rozumiano zredukowanie liczby generacji zmodyfikowanego o procesy epigenetyczne algorytmu genetycznego w stosunku do algorytmu bazowego.

Dla każdego algorytmu bazowego eksperymenty przeprowadzono w dwóch etapach: w pierwszej części dokonywano empirycznego wyznaczenia optymalnej wartości prawdopodobieństwa wystąpienia modyfikacji symulującej proces epigenetyczny, w drugiej weryfikowano czy proponowane operacje mają wpływ na redukcję liczby generacji. W Tabeli 4.44 zawarto zbiorcze wyniki ustalonych optymalnych prawdopodobieństw dla wprowadzanych w algorytmy bazowe operacji *Epi\_x*.

**Tabela 4.44:** Optymalne prawdopodobieństwo wystąpienia proponowanych operacji *Epi\_x* w badanych algorytmach

Proponowana operacja	Optymalne prawdopodobieństwo			
	<i>GAOZ</i>	<i>GAGR</i>	<i>GADO</i>	<i>GAMSA</i>
<i>Epi_1</i>	40%	20%	20%	20%
<i>Epi_2</i>	40%	20%	20%	20%
<i>Epi_3</i>	30%	30%	30%	20%

Na podstawie Tabeli 4.44 można zauważyć, że optymalne prawdopodobieństwo jest różne dla różnych operacji *Epi\_x* w poszczególnych algorytmach – jednak we wszystkich przypadkach zawiera się w przedziale od 20%-40%. Na podstawie przeprowadzonych badań Autor pracy zaleca jednak, aby samodzielnie dokonać doboru optymalnego prawdopodobieństwa wystąpienia proponowanych operacji na etapie dostrajania algorytmu genetycznego.

Wyniki eksperymentów związanych z redukcją liczby generacji w algorytmach genetycznych z zastosowanymi operacjami  $Epi\_x$  w stosunku do algorytmów bazowych ( $GAB$ ), przedstawiono w Tabeli 4.45.

**Tabela 4.45:** Średnia redukcja liczby generacji po zastosowaniu autorskich operacji w algorytmach bazowych

Proponowana operacja	Średnia redukcja liczby generacji			
	$GAOZ$	$GAGR$	$GADO$	$GAMSA$
$Epi\_1$	↓ 29%	↓ 20%	↓ 17%	↓ 17%
$Epi\_2$	↓ 25%	↓ 21%	↓ 27%	↓ 14%
$Epi\_3$	↓ 24%	↓ 14%	↓ 16%	↓ 9%

Największą średnią redukcję liczby generacji uzyskano przy zastosowaniu operacji  $Epi\_1$  w algorytmie  $GAGR$ , wynosiła ona 29% (por. Tabela 4.45), najmniejszy wpływ na redukcję liczby generacji w algorytmie genetycznym zaobserwowano w przypadku operacji  $Epi\_3$  w algorytmie  $GAMSA$  i wyniosła ona 9%. Wyniki te potwierdzają, że zastosowanie operacji  $Epi\_x$  wpływa na zmniejszenie liczby generacji algorytmu niezbędnej do uzyskania oczekiwanego wyniku.

Okazało się jednak, że wprowadzenie dodatkowych operacji do algorytmu genetycznego wiązało się też ze wzrostem czasu potrzebnego na przeprowadzenie wszystkich operacji w ramach pojedynczej generacji algorytmu. Przeprowadzone badanie wpływu wprowadzenia operacji  $Epi\_x$  na czas pojedynczej generacji algorytmu miało na celu sprawdzenie, czy mimo redukcji liczby generacji potrzebnych do uzyskania oczekiwanego rezultatu, czas wykonywania pozostanie akceptowalny (ten sam poziom lub niewielki wzrost). W Tabeli 4.46 przedstawiono średni wzrost czasu potrzebnego do przeprowadzenia wszystkich operacji w pojedynczej generacji dla poszczególnych algorytmów z operacjami  $Epi\_x$  względem algorytmów bazowych.

**Tabela 4.46:** Średni wzrost czasu dla pojedynczej generacji po zastosowaniu operacji  $Epi\_x$

Proponowana operacja	Średnie wydłużenie czasu pojedynczej generacji			
	$GAOZ$	$GAGR$	$GADO$	$GAMSA$
$Epi\_1$	↑ 7,00%	↑ 9,50%	↑ 7,50%	↑ 4,90%
$Epi\_2$	↑ 9,94%	↑ 9,01%	↑ 3,98%	↑ 3,33%
$Epi\_3$	↑ 11,05%	↑ 10,79%	↑ 5,08%	↑ 1,88%

Najmniejszy wzrost czasu potrzebnego do przeprowadzenia wszystkich operacji w pojedynczej generacji można zaobserwować dla operacji  $Epi\_1$ . W przypadku wybranych do badań algorytmów zastosowanie proponowanych operacji  $Epi\_x$  w najmniejszym stopniu wpłynęło na czas operacji w pojedynczej generacji w algorytmie  $GAMSA$ .

Wydłużenie czasu pojedynczej generacji przy zastosowaniu proponowanych w rozprawie modyfikacji okazał się na tyle niewielki, że przy otrzymanej redukcji liczby generacji można



spodziewać się redukcji całkowitego czasu potrzebnego do uzyskania optymalnego wyniku dzięki modyfikacjom *Epi\_x*. Przykładowo dla algorytmu *GAOZ* w przypadku zbioru 3 obliczony (na podstawie średniego czasu potrzebnego na przeprowadzenie operacji w pojedynczej generacji algorytmu genetycznego i mediany liczby generacji), średni łączny czas działania algorytmu bazowego wynosił 95,5 sekund, a w przypadku algorytmu *GAOZ Epi\_2* wynosił 67,5 sekund. Oznacza to, że mimo zwiększenia czasu potrzebnego na wykonanie operacji w ramach pojedynczej generacji, dzięki zastosowaniu operacji *Epi\_2* uzyskujemy skrócenie łącznego czasu o średnio 28 sekund (redukcja średniego łącznego czasu o około 30%, na podstawie Tabela 4.6 i Tabela 4.7).

Bazując na przeprowadzonych eksperymentach, potwierdzono, że zaproponowane w pracy autorskie operacje *Epi\_x* wpływają na poprawę efektywności działania algorytmów genetycznych: proponowane operacje inspirowane procesami epigenetycznymi wpływają na redukcję liczby generacji potrzebną do uzyskania oczekiwanego wyniku, a dzięki redukcji liczby generacji (pomimo wydłużenia czasu pojedynczej generacji) prowadzi to do redukcji łącznego czasu potrzebnego na znalezienie optymalnego rozwiązania przez algorytm genetyczny.

---

## Podsumowanie i wnioski

Rozprawa poświęcona została zagadnieniu związanemu z zastosowaniem algorytmów genetycznych w problemach optymalizacyjnych. Celem rozprawy było opracowanie autorskich operacji w algorytmach genetycznych, wzorowanych na procesach epigenetycznych i ich weryfikacja pod kątem zwiększenia efektywności działania algorytmów genetycznych. W rozprawie dokonano również przeglądu istniejących modyfikacji algorytmów genetycznych (Podrozdział 1.4.3). W rozprawie zaproponowano trzy nowe operacje w algorytmach genetycznych (Rozdział 3), zaproponowane modyfikacje, to:

- *Epi\_1* – operacja naśladująca proces epigenetyczny dziedziczenia za pomocą prionu;
- *Epi\_2* – modyfikacja naśladująca proces epigenetyczny metylacji cytozyny;
- *Epi\_3* – operacja naśladująca proces epigenetyczny wyłączenia allelicznego.

Na potrzeby badań eksperymentalnych zaimplementowano w języku **R** cztery przykładowe algorytmy genetyczne do rozwiązywania problemów optymalizacyjnych. Wybrane do przeprowadzenia eksperymentów algorytmy służyły do: optymalizacji załadunku, grupowania danych, wykrywania danych odstających i dopasowania sekwencji genetycznych (por. Rozdział 2). Algorytm genetyczny do grupowania danych jest algorytmem autorskim, pozostałe algorytmy są algorytmami zaczerpniętymi z literatury. Dla każdego spośród wybranych do eksperymentów algorytmów dokonano implementacji autorskich operacji inspirowanych epigenetyką. W części eksperymentalnej dokonano porównania wpływu proponowanych autorskich operacji na redukcję liczby generacji w algorytmie genetycznym, oraz na czas potrzebny na wykonanie wszystkich operacji w ramach pojedynczej generacji algorytmu genetycznego.

Pierwszą modyfikacją algorytmu genetycznego zaproponowaną w rozprawie była operacja naśladująca proces dziedziczenia za pomocą prionu (*Epi\_1*, Podrozdział 3.1). Operacja to została wprowadzona jako dodatkowa operacja do bazowych algorytmów genetycznych (wyniki

eksperymentów przedstawiono w Podrozdziałach 4.3.3, 4.4.4, 4.5.3, 4.6.3). Przy zastosowaniu operacji *Epi\_1* największą redukcję liczby generacji w stosunku do algorytmu bazowego uzyskano w przypadku algorytmu do optymalizacji załadunku (Tabela 4.4). Najmniejszą redukcję liczby generacji uzyskano natomiast w przypadku algorytmu do wykrywania wartości odstających i algorytmu do dopasowania sekwencji biologicznych (Tabele 4.26, 4.36). Duży wpływ na redukcję liczby generacji w algorytmie z wprowadzoną operacją *Epi\_1* miał dobór optymalnego prawdopodobieństwa wystąpienia tej operacji: jego zbyt duża wartość pogarszała działanie algorytmów *Epi\_1* względem algorytmów bazowych (Rysunki 4.1, 4.8, 4.8, 4.20). Na podstawie przeprowadzonych eksperymentów określono, że wartość optymalnego prawdopodobieństwa dla wystąpienia operacji *Epi\_1* w algorytmie genetycznym powinna zawierać się w przedziale od 20%-40%. Wprowadzenie modyfikacji *Epi\_1* do wybranych algorytmów genetycznych miało też wpływ na czas wykonywania wszystkich operacji w ramach pojedynczej generacji algorytmu, powodując jego wydłużenie. Najmniejszy wpływ wprowadzenia operacji *Epi\_1* na czas operacji w pojedynczej generacji zaobserwowano w przypadku algorytmu do dopasowania sekwencji biologicznych (Tabela 4.37). Największy wzrost czasu zaobserwowano w przypadku zastosowania dodatkowej operacji *Epi\_1* w algorytmie do grupowania danych (Tabela 4.17). Na podstawie przeprowadzonych eksperymentów na wybranych algorytmach genetycznych można stwierdzić poprawę działania algorytmów genetycznych przy zastosowaniu operacji *Epi\_1* w stosunku do algorytmów bazowych.

Drugą poddaną eksperymentom proponowaną operacją była operacja inspirowana procesem epigenetycznym metylacji cytozyny (*Epi\_2*, Podrozdział 3.2). Zaproponowana operacja powoduje zmodyfikowanie operacji krzyżowania (jest modyfikacją operacji krzyżowania). Dla operacji *Epi\_3* zaprezentowano wyniki eksperymentów nad oceną wpływu operacji na poprawę efektywności działania wybranych algorytmów genetycznych w następujących podrozdziałach (Podrozdziały 4.3.4, 4.4.5, 4.5.4, 4.6.4). Zastosowanie operacji *Epi\_2* w każdym przypadku spowodowało redukcję liczby generacji potrzebną do uzyskania oczekiwanego wyniku działania algorytmu. Największą redukcję liczby generacji uzyskano w przypadku algorytmu do wykrywania wartości odstających (Tabela 4.6), a najmniejszą – w przypadku algorytmu do dopasowania sekwencji biologicznych (Tabela 4.38). Na redukcję liczby generacji, tak jak i w przypadku poprzedniej operacji, duży wpływ miał dobór optymalnego prawdopodobieństwa wystąpienia modyfikacji *Epi\_2* w algorytmie genetycznym. Zbyt małe prawdopodobieństwo wystąpienia operacji *Epi\_2* nie wpływało na poprawę efektywności działania algorytmu, zbyt duże pogarszało działanie algorytmu (Rysunki 4.3, 4.10, 4.16, 4.22) (zalecane ustalone empirycznie optymalne prawdopodobieństwo wystąpienia operacji *Epi\_2* wynosi 20%). Proponowana operacja *Epi\_3*

wpływała jednak na wydłużenie czasu potrzebnego do przeprowadzenia wszystkich operacji w ramach pojedynczej generacji w wybranych algorytmach. Najmniejszy wpływ na wzrost czasu zaobserwowano w przypadku zastosowania operacji *Epi\_2* w algorytmach do wykrywania wartości odstających i dopasowania sekwencji biologicznych (Tabele 4.29, 4.39).

Trzecią zaproponowaną operacją inspirowaną epigenetyką była operacja naśladująca proces wyłączenia allelicznego (*Epi\_3*, Podrozdział 3.3). Operacja ta *Epi\_3* jest dodatkową operacją występującą w algorytmie genetycznym i tak jak w poprzednio, sprawdzono jej wpływ na poprawę efektywności działania bazowych algorytmów genetycznych (Podrozdziały 4.3.5, 4.4.6, 4.5.5, 4.6.5). Największą poprawę efektywności działania algorytmu genetycznego, poprzez największą redukcję liczby generacji zaobserwowano w przypadku algorytmu do optymalizacji załadunku (Tabela 4.8), a najmniejszą w algorytmie do dopasowania sekwencji biologicznych (Tabela 4.40). W eksperymentach sprawdzono wpływ prawdopodobieństwa wystąpienia operacji *Epi\_3* na efektywność działania algorytmów genetycznych. W przypadku zbyt dużego prawdopodobieństwa wystąpienia operacji *Epi\_3* obserwowano pogorszenie działania wybranych algorytmów genetycznych, w przypadku zbyt małego nie zaobserwowano znaczących różnic w porównaniu do algorytmów bazowych (Rysunki 4.5, 4.12, 4.18, 4.24). Na podstawie przeprowadzonych eksperymentów zaleca się, aby prawdopodobieństwo wystąpienia operacji *Epi\_3* mieściło się w zakresie od 20%-30%. Z przeprowadzonych badań wynika również, że wprowadzenie operacji *Epi\_3* do algorytmów genetycznych spowodowało wydłużenie czasu wykonywania operacji w ramach pojedynczej generacji algorytmu genetycznego. Najmniejsze wydłużenie czasu operacji w pojedynczej generacji zaobserwowano w przypadku algorytmu do dopasowania sekwencji biologicznych (Tabela 4.41), a największy w przypadku algorytmu do optymalizacji załadunku (Tabela 4.9).

Na podstawie przeprowadzonych eksperymentów udokumentowanych w rozprawie, należy stwierdzić, że teza rozprawy została potwierdzona: zastosowanie mechanizmów wzorowanych na procesach epigenetycznych poprawia efektywność działania algorytmów genetycznych. Spośród wszystkich zaproponowanych operacji inspirowanych epigenetyką najlepsze rezultaty uzyskano przy zastosowaniu operacji naśladującej proces metylacji cytozyny (*Epi\_2*) (Tabela 4.45).

Osiągnięto również cele dodatkowe, postawione w rozprawie. W tym:

- przeanalizowano operacje epigenetyczne pod kątem ich możliwości ich zastosowania w algorytmach genetycznych (Rozdział 3),
- przeanalizowano istniejące modyfikacje algorytmów genetycznych (Podrozdział 1.4.3),

- zaprojektowano i zaimplementowano zaproponowane operacje bazujące na procesach epigenetycznych w algorytmach genetycznych (Rozdziały 3, 4),
- przeprowadzono badania eksperymentalne zaproponowanych operacji pod kątem wpływu na poprawę efektywności działania algorytmów genetycznych (Rozdział 4),
- zbadano wpływ parametrów proponowanych operacji na efektywność działania algorytmów genetycznych (Rozdział 4),
- przeprowadzono badania eksperymentalne pod kątem oceny wpływu wprowadzenia zaproponowanych operacji na czas działania algorytmu genetycznego (Rozdział 4),
- porównano działanie algorytmów genetycznych z wprowadzonymi proponowanymi operacjami z algorytmami bazowymi (bez dodatkowych operacji epigenetycznych) (Rozdział 4).

Zastosowanie operacji naśladujących procesy epigenetyczne w algorytmach genetycznych daje pozytywne rezultaty, w związku z tym istnieje wiele możliwości kontynuowania badań, takie jak:

1. Sprawdzenie możliwości zastosowania operacji bazujących na innych procesach epigenetycznych. Nie wszystkie zagadnienia związane z epigenetyką zostały już poznane, w związku z tym można przypuszczać, że będą odkrywane nowe mechanizmy epigenetyczne, których odwzorowanie w algorytmach genetycznych może poprawiać ich działanie.
2. Przeprowadzenie badań nad zastosowaniem operacji bazujących na procesach epigenetycznych w algorytmach genetycznych zastosowanych dla innych niż pokazano w pracy, problemów optymalizacyjnych.
3. Dokonanie oceny wpływu połączenia proponowanych operacji naśladujących procesy epigenetyczne w ramach jednego algorytmu genetycznego.
4. Opracowanie algorytmu epigenetycznego (jako nowego typu algorytmów ewolucyjnych), który bazował będzie tylko na procesach epigenetycznych i porównanie efektywności jego działania z klasycznymi algorytmami genetycznymi.
5. Opracowanie nowych algorytmów genetycznych zawierających z założenia operacje bazujące na epigenetyce.

---

## Bibliografia

- [1] Y.L. Abdel-Magid and M.M. Dawoud. Optimal agc tuning with genetic algorithms. *Electric Power Systems Research*, 38(3):231–238, sep 1996.
- [2] O. Abdoun, , and J. Abouchabaka. A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. *International Journal of Computer Applications*, 31(11), 2012.
- [3] M. A. Abido. Multiobjective evolutionary algorithms for electric power dispatch problem. In *Intelligent Systems Reference Library*, pages 47–82. Springer Berlin Heidelberg, 2009.
- [4] I. Abuiziah and N. Shakarneh. A review of genetic algorithm optimization: Operations and applications to water pipeline systems. *International Journal of Mathematical and Computational Sciences*, 7(12), 2013.
- [5] J. Ackora-Prah, S. A. Gyamerah, and P. S. Andam. A heuristic crossover for portfolio selection. *Applied Mathematical Sciences*, 8:3215–3227, 2014.
- [6] Z. H. Ahmed. Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator. *International Journal of Biometrics & Bioinformatics*, 3(6):96, 2010.
- [7] H. Akaike. Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, 1973.
- [8] R. Al-Haddad and et al. Epigenetic changes in diabetes. *Neuroscience Letters*, 625:64–69, jun 2016.
- [9] M. Alhomidi and M. Reed. A genetic algorithm approach for the most likely attack path problem. In *2013 International Conference on Availability, Reliability and Security*, pages 360–366, Sep. 2013.

- [10] L. A. Allison. *Podstawy biologii molekularnej*. Wydawnictwo Uniwersytetu Warszawskiego, 2009.
- [11] Ö. G. Alma. *Outlier Detection Methods: Genetic Algorithms Based Outlier Detection using Information Criteria*. LAP LAMBERT Academic Publishing, 2010.
- [12] C.V. Almeyda-Becerra, B.E. Lockhart, and A.S. Swenson. Characterization of tomato yellow blotch virus, a new tymovirus occurring in greenhouse-grown tomatoes in minnesota. zgłoszone do czasopisma Plant Pathology (29.08.2011).
- [13] S. N. Sivan andam and S. N. Deepa. *Introduction to Genetic Algorithms*. Springer-Verlag GmbH, 2007.
- [14] B. Dinab andhu, M. C. A., and P. Sankar. Variance as a stopping criterion for genetic algorithms with elitist model. *Fundamenta Informaticae*, 120(2):145–164, 2012.
- [15] M. Angelova, P. Melo-Pinto, and T. Pencheva. Modified simple genetic algorithms improving convergence time for the purposes of fermentation process parameter identification. *WSEAS TRANSACTIONS on SYSTEMS*, 11:256–267, 2012.
- [16] M. Angelova and T. Pencheva. Tuning genetic algorithm parameters to improve convergence time. *International Journal of Chemical Engineering*, 2011:1–7, 2011.
- [17] J. Arabas. *Wykłady z algorytmów ewolucyjnych*. WNT, 2004.
- [18] J. Arabas. *Wykłady z algorytmów ewolucyjnych*. Wydawnictwo WNT, 2004.
- [19] P.K. Arora, A. Haleem, M.K. Singh, H. Kumar, and M. Kaushik. Design of a production system using genetic algorithm. *Procedia Technology*, 14:390–396, 2014.
- [20] D. Ashlock. *Evolutionary Computation for Modeling and Optimization*. Springer-Verlag GmbH, 2005.
- [21] I. Ayachi, R. Kammarti, M. Ksouri, and P. Borne. A genetic algorithm to solve the container storage space allocation problem. In *2010 International conference on Computational Intelligence and Vehicular System*, 2010.
- [22] H. Aytug and G. J. Koehler. New stopping criterion for genetic algorithms. *European Journal of Operational Research*, 126(3):662–674, nov 2000.
- [23] G. A. Azim. Pairwise sequence alignment revisited - genetic algorithms and cosine functions. *Applied Mathematics, Simulation, Modelling 2008*, 2008.

- [24] J. Bal. *Genetyka medyczna i molekularna*. Wydawnictwo Naukowe PWN, 2017.
- [25] W. Banzhaf, P. Nordin, and R. Keller. *Genetic Programming: An Introduction*. MORGAN KAUFMANN PUBL INC, 1997.
- [26] E. Bas and et al. A modified genetic algorithm for forecasting fuzzy time series. *Applied Intelligence*, 41(2):453–463, September 2014.
- [27] T. Blickle and L. Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394, dec 1996.
- [28] A. Bogoyavlenski, V. Berezin i A. Prilipov, and et al. Newcastle disease outbreaks in kazakhstan and kyrgyzstan during 1998, 2000, 2001, 2003, 2004, and 2005 were caused by viruses of the genotypes viib and viid. *Virus Genes*, 39 (1):94–101, 2009.
- [29] A. Brabazon, M. O’Neill, and S. McGarraghy. Introduction to evolutionary computing. In *Natural Computing Algorithms*, pages 17–20. Springer Berlin Heidelberg, 2015.
- [30] J. R. Bradley, B. R. Pober, and D. R. Johnson. *Genetyka medyczna*. PZWL, 2009.
- [31] T. A. Brown. *Genomy*. Wydawnictwo Naukowe PWN, 2013.
- [32] S. B. Bukhari, A. Ahmad, S. Auon Raza, and N. Siddique. A ring crossover genetic algorithm for unit commitment problem. *Turkish Journal of Electrical Engineering and Computer Sciences*, 24:3862 – 3876, 05 2016.
- [33] F. J. Burkowski. Shuffle crossover and mutual information. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1574–1580, 1999.
- [34] F. Capuano, M. Mülleder, R. Kok, H. Blom, and M. Ralser. Cytosine dna methylation is found in drosophila melanogaster but absent in saccharomyces cerevisiae, schizosaccharomyces pombe, and other yeast species. *Analytical chemistry*, 86:3697–3702, April 2014.
- [35] L. Cervenakova, J.J. Halperin, G. Stone, I. Taller, and C.J. Gibbs. A new creutzfeldt-jakob disease family with 96bp insertion mutation in the prnp gene. nieopublikowany.
- [36] S. S. Chaudhry and W. Luo. Application of genetic algorithms in production and operations management: a review. *International Journal of Production Research*, 43(19):4083–4101, oct 2005.



- [37] O. Chebbi and J. Chaouachi. Effective parameter tuning for genetic algorithm to solve a real world transportation problem. In *2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, aug 2015.
- [38] R-C. Chen, C-T. Chen, and J-Y. Li. A genetic algorithm for planning travel route with minimum transportation carbon footprint. In *Information and Business Intelligence*, pages 57–63. Springer Berlin Heidelberg, 2012.
- [39] L. Chomatek and A. Duraj. Multiobjective genetic algorithm for outliers detection. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 379–384, July 2017.
- [40] K. Chromiński, M. Tkacz, and M. Boryczka. Epigenetically inspired genetic algorithm for clustering data. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, jul 2018.
- [41] Kornel Chromiński and Mariusz Boryczka. Epigenetically inspired modification of genetic algorithm and his efficiency on biological sequence alignment. In *Intelligent Decision Technologies 2016*, pages 95–105. Springer International Publishing, 2016.
- [42] B.G.W. Craenen, A.E. Eiben, and J.I. van Hemert. Comparing evolutionary algorithms on binary constraint satisfaction problems. *IEEE Transactions on Evolutionary Computation*, 7(5):424–444, oct 2003.
- [43] F. Lima Custodio, H. J.C. Barbosa, and L. E. Dardenne. A multiple minima genetic algorithm for protein structure prediction. *Applied Soft Computing*, 15:88–99, feb 2014.
- [44] M. de los Angeles Saez Blazquez and et al. Genetic algorithm for scheduling routes in public transport. In *Advances in Intelligent Systems and Computing*, pages 399–406. Springer International Publishing, 2014.
- [45] A. Homem Dornas and et al. Real-polarized genetic algorithm for the three-dimensional bin packing problem. In *Proceedings of the Genetic and Evolutionary Computation Conference on - 17*. ACM Press, 2017.
- [46] H. Du, Z. Wang, W. Zhan, and J. Guo. Elitism and distance strategy for selection of evolutionary algorithms. *IEEE Access*, 6:44531–44541, 2018.
- [47] O. Dubrovsky, G. Levitin, and M. Penn. A genetic algorithm with a compact solution encoding for the container ship stowage problem. *Journal of Heuristics*, 8(6):585–599, Nov 2002.

- [48] C. Dupont, D. Armant, and C. Brenner. Epigenetics: Definition, mechanisms and clinical perspective. *Seminars in Reproductive Medicine*, 27(05):351–357, aug 2009.
- [49] R. Eberhart and Y. Shi. Comparison between genetic algorithms and particle swarm optimization. In *International conference on evolutionary programming*, pages 611–616. Springer, 1998.
- [50] Eiben, E. Agoston, and J. E. Smith. Introduction to evolutionary computing. *Heidelberg: Springer*, 2003.
- [51] A. Eiben and H. M. Van Kemenade. Diagonal crossover in genetic algorithms for numerical optimization. *Control and Cybernetics*, 26, 06 1997.
- [52] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 3(2):124, 1999.
- [53] A.E. Eiben and S.K. Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1(1):19–31, 2011.
- [54] A.E. Eiben and S.K. Smit. Evolutionary algorithm parameters and methods to tune them. *Autonomous Search*, pages 25–38, Springer, 2012.
- [55] M. Ester, H-P. Kriegel, J. S ander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, pages 226–231. AAAI Press, 1996.
- [56] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 (2):179–188, 1936.
- [57] D. B. Fogel. Evolutionary algorithms in theory and practice. *Complexity*, 2(4):26–27, mar 1997.
- [58] F. D. Francone, M. Conrads, W. Banzhaf, and P. Nordin. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO ’99)*. Morgan Kaufmann, 1999.
- [59] Pasi Franti and et al. Clustering datasets. *Clustering datasets*, 2016.
- [60] P. Fränti and O. Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39 (5):761–765, 2006.

- [61] M. Gao, G. Shi, W. Li, Y. Wang, and D. Liu. An improved genetic algorithm for island route planning. *Procedia Engineering*, 174:433–441, 2017.
- [62] D. Gladwin, P. Stewart, and J. Stewart. A controlled migration genetic algorithm operator for hardware-in-the-loop experimentation. *Engineering Applications of Artificial Intelligence*, 24(4):586–594, jun 2011.
- [63] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989.
- [64] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Elsevier, 1991.
- [65] D.E. Goldberg. *Algorytmy genetyczne i ich zastosowania*. WNT Warszawa, 1995.
- [66] C. Gondro and B.P. Kinghorn. A simple genetic algorithm for multiple sequence alignment. *Genetics and Molecular Research*, 6(4), 2007.
- [67] R.N. Greenwell, J.E. Angus, and M. Finck. Optimal mutation probability for genetic algorithms. *Mathematical and Computer Modelling*, 21(8):1–11, apr 1995.
- [68] R. Gupta, P. Agarwal, and A.K. Son. Genetic algorithm based approach for obtaining alignment of multiple sequences. *International Journal of Advanced Computer Science and Applications*, 3(12), 2012.
- [69] T.D. Gwiazda. *Algorytmy genetyczne. Kompendium*. PWN, 2007.
- [70] W. W. Hager. Computational optimization and applications, 2017.
- [71] S. Haldenbilen and H. Ceylan. Transport demand management in turkey: A genetic algorithm approach. *Transportation Planning and Technology*, 28(6):403–426, dec 2005.
- [72] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.
- [73] A. Hassanat and et al. Enhancing genetic algorithms using multi mutations: Experimental results on the travelling salesman problem. *International Journal of Computer Science and Information Security*, 14:785–801, 09 2016.
- [74] F. Herrera, M. Lozano, and A. M. Sánchez. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems*, 18(3):309–338, feb 2003.

- [75] P. E. Hewitt, S. Ijaz, R. Brailsford, and et al. Hepatitis e virus in blood components: a prevalence and transmission study in southeast england. *Lancet (London, England)*, 384:1766–1773, November 2014.
- [76] S. A. Hofmeyr and S. Forrest. *Immunity by Design: An Artificial Immune System*. Proceedings of 1999 GECCO Conference, 1999.
- [77] E. Hopper and B. Turton. Application of genetic algorithms to packing problems. a review. In *Soft Computing in Engineering Design and Manufacturing*, pages 279–288. Springer London, 1998.
- [78] E. R. Hruschka and N. F.F. Ebecken. A genetic algorithm for cluster analysis. *Intelligent Data Analysis*, 7(1):15–25, Feb 2003.
- [79] J-S. Huang and H-C. Liu. Object recognition using genetic algorithms with a hopfields neural model. *Expert Systems with Applications*, 13(3):191–199, oct 1997.
- [80] A. Hussain, Y. Muhammad, and A. Nawaz. Optimization through genetic algorithm with a new and efficient crossover operator. *International Journal of Advances in Mathematics*, 2018(1):1–14, 2018.
- [81] Ch. Igel, T. Suttorp, and N. Hansen. *Steady-State Selection and Efficient Covariance Matrix Update in the Multi-objective CMA-ES*, volume 4403. Springer-Verlag GmbH, 2007.
- [82] K. Jebari. Selection methods for genetic algorithms. *International Journal of Emerging Sciences*, 3:333–344, 12 2013.
- [83] K. Jin-Lee. Genetic algorithm stopping criteria for optimization of construction resource scheduling problems. *Construction Management and Economics*, 31(1):3–19, jan 2013.
- [84] Vedavyasrao Jorapur, V. S. Puranik, A. S. Deshp ande, and M. R. Sharma. Comparative study of different representations in genetic algorithms for job shop scheduling problem. *Journal of Software Engineering and Applications*, 07(07):571–580, 2014.
- [85] T. Jun-shan, H. Wei, and Q. Yan. Application of genetic algorithm in data mining. In *2009 First International Workshop on Education Technology and Computer Science*, volume 2, pages 353–356, March 2009.
- [86] M. Kaur. A review on pattern recognition using genetic algorithms. *International journal of Emerging Trends in Science and Technology*, 04(05):5213–5220, jun 2017.

- [87] Y. KAYA, M. UYAR, and R. TEKIN. A novel crossover operator for genetic algorithms: Ring crossover. *CoRR*, abs/1105.0355, 2011.
- [88] Y. Kaya, M. Uyar, and R. Tekin. A novel crossover operator for genetic algorithms: Ring crossover. *CoRR*, 2011.
- [89] F. Keller, , E. Muller, and K. Bohm. Hics: High-contrast subspaces for density-based outlier ranking. *ICDE*, 2012.
- [90] G. M. Khan. Evolutionary computation. In *Evolution of Artificial Neural Development*, pages 29–37. Springer, 2018.
- [91] M. Kljajic, I. Bernik, and U. Breskvar. Production planning using simulation and genetic algorithms in multi-criteria scheduling optimization. *Proceedings of the 8th WSEAS International Conference on Evolutionary Computing*, 12 2018.
- [92] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag GmbH, 2000.
- [93] H. P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Loop: Local outlier probabilities. *Proc. 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, 2009.
- [94] A. J. S. Kumar, J.Arunadevi, and V.Mohan. Intelligent transport route planning using genetic algorithms in path computation algorithms. *European Journal of Scientific Research*, 25(3):463–468, 2009.
- [95] H. K. Lai. Genetic algorithm based optimization in computer network applications. In *Conference: 17th Annual Rensselaer at Hartford Computer Science Seminar*, 2001.
- [96] W. B. Langdon. Size fair and homologous tree crossovers for tree genetic programming. *Genetic Programming and Evolvable Machines*, 1(1/2):95–119, 2000.
- [97] C-Y. Lee. Entropy-boltzmann selection in the genetic algorithms. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 33:138–49, 02 2003.
- [98] X. Li, Z. Zhao, and K. Zhang. A genetic algorithm for the three-dimensional bin packing problem with heterogeneous bins. In *Industrial and Systems Engineering Research*, 05 2014.

- [99] A.M. Linde, M. Munir, and S. Zohari et al. Complete genome characterisation of a newcastle disease virus isolated during an outbreak in sweden in 1997. *Virus Genes*, 41(2):165–173, 2010.
- [100] A. Lipowski and D. Lipowska. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6):2193–2196, 2012.
- [101] F. T. Liu, K. Ming Ting, and Zhi-Hua Zhou. Isolation forest. *Eighth IEEE International Conference on Data Mining*, 2008.
- [102] F. G. Lobo, C. F. Lima, and Z. Michalewicz. *Parameter Setting in Evolutionary Algorithms*. Springer Berlin Heidelberg, 2007.
- [103] L. M. R. J. Lobo and S. B. Chavan. Use of genetic algorithm in network security. *International Journal of Computer Applications*, 53(8):1–7, sep 2012.
- [104] V. I. Loginov, S. V. Rykov, M. V. Fridman, and E. A. Braga. Methylation of mirna genes and oncogenesis. *Biochemistry (Moscow)*, 80(2):145–162, feb 2015.
- [105] J. Magalhaes-Mendes. A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. *WSEAS TRANSACTIONS on COMPUTERS*, 12(4):164–173, 2013.
- [106] S. Maibam, P Roychoudhury, P.K. Subudhi, and et al. Molecular pathotyping and genotyping of newcastle disease virus isolates from different species of birds in north eastern india. *Department of Veterinary Microbiology, College of Veterinary Science, India*, 2016.
- [107] K. L. Mak and Y. S. Wong. FUNCTION OPTIMIZATION BY USING GENETIC ALGORITHMS WITH INDIVIDUALS HAVING DIFFERENT BIRTH AND SURVIVAL RATES. *Engineering Optimization*, 33(6):749–777, aug 2001.
- [108] K. Man, K-S. Tang, and S. Kwong. *Genetic Algorithms*. Springer London, 2012.
- [109] J. Manjrekar. Epigenetic inheritance, prions and evolution. *Journal of Genetics*, 96(3):445–456, jul 2017.
- [110] N. Mansour and et al. Incremental genetic algorithm. *The International Arab Journal of Information Technology*, 3(1):42–47, 2006.
- [111] N. Mansour, F. Kanj, and H. Khachfe. Evolutionary algorithm for protein structure prediction. In *2010 Sixth International Conference on Natural Computation*, volume 8, pages 3974–3977, Aug 2010.

- [112] U. Maulik and S. Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern Recognition*, 33(9):1455–1465, sep 2000.
- [113] F. Melo, J. Fernandes, B. Ribeiro, and S. Ribeiro. Complete genome sequence of a tobacco-infecting, tomato-blistering mosaic virus. *Genome announcements*, 2, August 2014.
- [114] D. Menghani and A. Guha. Packing boxes into multiple containers using genetic algorithm. *Journal of The Institution of Engineers (India): Series C*, 97(3):441–450, jun 2016.
- [115] O. Mersmann and et al. *Accurate Timing Functions*, October 2018. wersja 1.4-6.
- [116] Z. Michalewicz. *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. WNT Warszawa, 2003.
- [117] Z. Michalewicz. *Algorytmy genetyczne+struktury danych=programy ewolucyjne*. WNT, 2003.
- [118] B. Miller and D. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212, 1995.
- [119] D. Moore and S. David. *The Developing Genome: An Introduction to Behavioral Epigenetics*. OXFORD UNIV PR, 2015.
- [120] S. C. Nanayakkara and et al. Genetic algorithm based route planner for large urban street networks. In *2007 IEEE Congress on Evolutionary Computation*. IEEE, sep 2007.
- [121] S. Nitasha and K. Tapas. Study of various mutation operators in genetic algorithms. *International Journal of Computer Science and Information Technologies*, 5(3):4519–4521, 2014.
- [122] F. Ornelas and et al. Genetic algorithm using migration and modified gx as support in the diversification of populations. *2010 Ninth Mexican International Conference on Artificial Intelligence(MICAI)*, 01 2009.
- [123] D. Ortiz-Boyer, C. Hervas-Martinez, and N. Garcia-Pedrajas. Cxl2: A crossover operator for evolutionary algorithms based on population features. *Journal of Artificial Intelligence Research*, 24:1–48, 2005.
- [124] P. *Bioinformatyka i ewolucja molekularna*. Wydawnictwo Naukowe PWN, 2008.

- [125] I G. Pasek and et al. Traffic light signal parameters optimization using modification of multielement genetic algorithm. *International Journal of Electrical and Computer Engineering (IJECE)*, 8(1):246, feb 2018.
- [126] J. T Pedersen and J. Moulton. Genetic algorithms for protein structure prediction. *Current Opinion in Structural Biology*, 6(2):227–231, apr 1996.
- [127] D. Pelleg and A. W. Moore. *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*. Morgan Kaufmann, 2000.
- [128] T. Pencheva, K. Atanasov, and A. Shannon. Modelling of a stochastic universal sampling selection operator in genetic algorithms using generalized nets. In *Proceedings of the Tenth International Workshop on Generalized Nets, Sofia*, pages 1–7, 2009.
- [129] Ulrich Pferschy and Joachim Schauer. The knapsack problem with conflict graphs. *Journal of Graph Algorithms and Applications*, 13(2):233–249, 2009.
- [130] S. Picek and M. Golub. Comparison of a crossover operator in binary-coded genetic algorithms. *WSEAS TRANSACTIONS on COMPUTERS*, 9(9):1064–1074, 2010.
- [131] R. Poli, William, B. Langdon, and N. Freitag McPhee. *A Field Guide to Genetic Programming*. Lulu Enterprises, UK Ltd, 2008.
- [132] C. Preechakul and S. Kheawhom. Modified genetic algorithm with sampling techniques for chemical engineering optimization. *Journal of Industrial and Engineering Chemistry*, 15(1):110–118, jan 2009.
- [133] Y. Qiu, F. Liu, and X. Huang. Network optimization based on genetic algorithm and estimation of distribution algorithm. In *2008 International Conference on Computer Science and Software Engineering*, volume 4, pages 1058–1061, Dec 2008.
- [134] J.R. Quinlan, P.J. Compton, K.A. Horn, and L. Lazurus. Inductive knowledge acquisition: A case study. *Proceedings of the Second Australian Conference on Applications of Expert Systems*, 1686.
- [135] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008.
- [136] P. Vishnu Raja and V.Murali Bhaskaran. An effective genetic algorithm for outlier detection. *International Journal of Computer Applications*, 38(6):30–33, jan 2012.



- [137] M. Ramanauskas and et al. Genetic algorithm with modified crossover for grillage optimization. *International Journal of Computers Communications & Control*, 12(3):393, apr 2017.
- [138] C. T. Ranjith-Kumar, K. Gopinath, A. N. Jacob, V. Srividhya, P. Elango, and H.S. Savithri. Genomic sequence of physalis mottle virus and its evolutionary relationship with other tymoviruses. *Archives of virology*, 143:1489–1500, 1998.
- [139] S. S. Ray, S. B andyopadhyay, and S. K. Pal. New operators of genetic algorithms for traveling salesman problem. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 497–500 Vol.2, Aug 2004.
- [140] N. M. Razali and J. Geraghty. Genetic algorithm performance with different selection strategies in solving tsp. *Proceedings of the World Congress on Engineering*, 2, 2011.
- [141] O. Roeva. A modified genetic algorithm for a parameter identification of fermentation processes. *Biotechnology & Biotechnological Equipment*, 20(1):202–209, jan 2006.
- [142] F. Rothlauf. *Representations for Genetic and Evolutionary Algorithms*. Springer, 2006.
- [143] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA, 2015.
- [144] K. Rutczyńska-Wdowiak. Dobór kryterium zatrzymania algorytmu genetycznego na przykładzie identyfikacji parametrycznej modelu matematycznego silnika indukcyjnego. *PRZEGLĄD ELEKTROTECHNICZNY*, 1(12):285–290, dec 2016.
- [145] S. Saeidi. A genetic algorithm for route optimization in public transportation problem. *2014 International Conference on Business and Information*, 07 2014.
- [146] M. Safe, J. Carballido, I. Ponzoni, and N. Brignole. On stopping criteria for genetic algorithms. In *Advances in Artificial Intelligence – SBIA 2004*, pages 405–413. Springer Berlin Heidelberg, 2004.
- [147] Y. Sakamoto, M. Ishiguro, and G. Kitagawa. Akaike information criterion statistics. *D. Reidel Publishing Company*, 1986.
- [148] M. Sarfraz, Y. Javed, A. Anjum, and S. Sohail. Routing optimization strategy using genetic algorithm utilizing bandwidth and delay. In *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, volume 4, pages 158–162, Feb 2010.

- [149] D. Schübeler. Function and information content of dna methylation. *Nature*, 517:321–326, January 2015.
- [150] R. R. Sharapov. Genetic algorithms: Basic ideas, variants and analysis. *Vision Systems: Segmentation and Pattern Recognition*, pages 407–422, 2007.
- [151] D. Simon. *Evolutionary Optimization Algorithms*. Wiley, 2013.
- [152] R. Sivaraj and T. Ravich andran. Review of selection methods in genetic algorithm. *International Journal of Engineering Science and Technology (IJEST)*, 3(5), 2011.
- [153] D. B. Smith, S. Ijaz, R. S. Tedder, and et al. Variability and pathogenicity of hepatitis e virus genotype 3 variants. *The Journal of general virology*, 96:3255–3264, November 2015.
- [154] P. Snaselova and F. Zboril. Genetic algorithm using theory of chaos. *Procedia Computer Science*, 51:316–325, 2015.
- [155] R. Sridhar and et al. Adaptive genetic approach for optimization of heterogeneous bin packing problems. *International Journal of Pure and Applied Mathematics*, 2017.
- [156] M. Srinivas and L. M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):656–667, 1994.
- [157] M. Srinivas and L.M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):656–667, apr 1994.
- [158] Lieven Vandenberghe Stephen Boyd. *Convex Optimization*. Cambridge University Press, 2019.
- [159] R. Strausberg, E. Feingold, and L. Grouse et al. Generation and initial analysis of more than 15,000 full-length human and mouse cDNA sequences. *Proceedings of the National Academy of Sciences of the United States of America*, 99:16899–16903, December 2002.
- [160] M. Studniarski. Stopping criteria for genetic algorithms with application to multiobjective optimization. In *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part I, PPSN’10*, pages 697–706, Berlin, Heidelberg, 2010. Springer-Verlag.
- [161] P.N. Suganthan. Structural pattern recognition using genetic algorithms. *Pattern Recognition*, 35(9):1883–1893, sep 2002.

- [162] R. Tadeusiewicz. *Sieci neuronowe*. Akademicka Oficyna Wydawnicza RM, 1993.
- [163] C. Tajani, O. Abdoun, and A.I. Lahjouji. Genetic algorithm adopting immigration operator to solve the asymmetric traveling salesman problem. *International Journal of Pure and Applied Mathematics*, 115(4), aug 2017.
- [164] M. Tavakkolian, F. Jalali F., and M.A. Emadi. Production optimization using genetic algorithm approach. In *Nigeria Annual International Conference and Exhibition*. Society of Petroleum Engineers, 2004.
- [165] P.K. Tripathy, R.K. Dash, and C.R. Tripathy. A genetic algorithm based approach for topological optimization of interconnection networks. *Procedia Technology*, 6:196–205, 2012.
- [166] K. Tsangaras, S. Kolokotronis, R. Ulrich, S. Mor, , and J. Michaux. Negative purifying selection drives prion and doppel protein evolution. *Journal of molecular evolution*, 79:12–20, August 2014.
- [167] A.J. Umbarkar and P.D. Sheth. Crossover operators in genetic algorithms: a review. *ICTACT Journal on Soft Computing*, 06(01):1083–1092, oct 2015.
- [168] A. Urso. *Encyclopedia of Bioinformatics and Computational Biology*. Elsevier Inc., 2018.
- [169] M.D.B. van de Garde, S.D. Pas, and G. van der Net et al. Hepatitis e virus genotype 3 infection of human liver chimeric mice and a549 cells as novel models for the study of chronic hev infection. zgłoszone do *Viroscience* (21.06.2015).
- [170] M. J. Varnamkhasti, L. S. Lee, M. R. Abu Bakar, and W. J. Leong. A genetic algorithm with fuzzy crossover operator and probability. *Advances in Operations Research*, 2012:1–16, 2012.
- [171] M. Wael. Optimization of production systems using genetic algorithms. *International Journal of Computational Intelligence and Applications*, 03(03):233–248, sep 2003.
- [172] B. Webb. Swarm intelligence: From natural to artificial systems. *Connection Science*, 14(2):163–164, jun 2002.
- [173] M. D. Weight and H. Harpending. SOME USES OF MODELS OF QUANTITATIVE GENETIC SELECTION IN SOCIAL SCIENCE. *Journal of Biosocial Science*, 49(01):15–30, feb 2016.

- [174] T. Weise. *Global Optimization Algorithms - Theory and Application*. Thomas Weise, 2007.
- [175] W.H. Wolberg and O.L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87:9193–9196, 1990.
- [176] A. Wren and D. O. Wren. A genetic algorithm for public transport driver scheduling. *Computers & Operations Research*, 22(1):101–110, jan 1995.
- [177] S. Wright. Stochastic processes in evolution. *Stochastic models in medicine and biology*, 25:199–241, 1964.
- [178] P. Węgleński. *Genetyka molekularna*. PWN Warszawa, 2006.
- [179] S. L. Yadav and A. Sohal. Comparative study of different selection techniques in genetic algorithm. *International Journal of Engineering, Science and Mathematics*, 6(3):174–180, 2017.
- [180] C. Yang, Q. Qian, F. Wang, and M. Sun. An improved adaptive genetic algorithm for function optimization. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 675–680, Aug 2016.
- [181] E. Yeguas, M.V. Luzon, R. Pavon, R. Laza, G. Arroyo, and F. Diaz. Automatic parameter tuning for evolutionary algorithms using a bayesian case-based reasoning system. *Applied Soft Computing*, 18:185–195, may 2014.
- [182] D. Q. Zeebaree and et al. Combination of k-means clustering with genetic algorithm: A review. *International Journal of Applied Engineering Research*, 12(24):14238–14245, 2017.
- [183] A. Zemach, I. McDaniel, P. Silva, and D. Zilberman. Genome-wide evolutionary analysis of eukaryotic dna methylation. *Science (New York, N.Y.)*, 328:916–919, May 2010.
- [184] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the eleventh international conference on Information and knowledge management - CIKM*. ACM Press, 2002.
- [185] C. Zhou, J. Zheng, and W. Li. An improved heuristic crossover operator for tsp. In *2008 Fourth International Conference on Natural Computation*, volume 1, pages 541–545, Oct 2008.

- 
- [186] M.J. Valadan Zoej, M. Mokhtarzade, A. Mansourian, H. Ebadi, and S. Sadeghian. Rational function optimization using genetic algorithms. *International Journal of Applied Earth Observation and Geoinformation*, 9(4):403–413, dec 2007.

---

## Spis tabel

1.1	Podział problemów optymalizacyjnych według różnych kryteriów . . . . .	13
1.2	Przykłady problemów optymalizacyjnych w zależności od typu problemu optymalizacyjnego . . . . .	14
1.3	Pojęcia związane z algorytmami genetycznymi i odniesienie do pojęć stosowanych w naukach o życiu . . . . .	19
1.4	Sposoby kodowania osobników w <i>KAG</i> . . . . .	20
1.5	Podstawowe parametry algorytmu genetycznego . . . . .	28
4.1	Konwencja nazewnicza oznaczeń algorytmów stosowana w opisach eksperymentów .	53
4.2	Dane wykorzystane w eksperymentach z <i>GAOZ</i> . . . . .	55
4.3	Parametry algorytmu genetycznego wykorzystane w eksperymentach . . . . .	56
4.4	Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_1</i> (Pogrubioną kursywą zaznaczono najmniejszą liczbę iteracji dla danego zbioru testowego) . . . . .	57
4.5	Porównanie czasów jednej generacji dla <i>GAOZEpi_1</i> i <i>GAOZ</i> . . . . .	59
4.6	Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_2</i> (Pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	60
4.7	Porównanie czasów pojedynczej generacji w <i>GAOZ</i> względem <i>GAOZEpi_2</i> . . . . .	63
4.8	Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_3</i> (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	64

4.9	Porównanie czasów pojedynczej generacji <i>GAOZ</i> i <i>GAOZ Epi_3</i> . . . . .	67
4.10	Optymalne prawdopodobieństwo wystąpienia operacji <i>Epi</i> w algorytmie <i>GAOZ</i> . . . . .	67
4.11	Procentowa wartość średniej redukcji liczby generacji oraz procentowe wydłużenie czasu pojedynczej generacji <i>GAOZ Epi_x</i> , względem <i>GAOZ</i> . . . . .	68
4.12	Zbiory eksperymentalne . . . . .	69
4.13	Parametry algorytmu genetycznego wykorzystane w eksperymentach . . . . .	69
4.14	Porównanie skuteczności proponowanego <i>GAGR</i> , z popularnymi metodami grupowania danych . . . . .	71
4.15	Wartości precyzji dla poszczególnych zbiorów uzyskane przez porównywane metody . . . . .	71
4.16	Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_1</i> w <i>GAGR</i> (Pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	72
4.17	Porównanie czasów pojedynczej generacji <i>GAGR</i> i <i>GAGR Epi_1</i> . . . . .	75
4.18	Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_2</i> (Pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	76
4.19	Porównanie czasów pojedynczej generacji w <i>GAGR Epi_2</i> i <i>GAGR</i> . . . . .	78
4.20	Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_3</i> w <i>GAGR</i> (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	79
4.21	Porównanie czasów pojedynczej generacji w <i>GAGR Epi_3</i> i <i>GAGR</i> . . . . .	82
4.22	Optymalne prawdopodobieństwo wystąpienia operacji <i>Epi_x</i> w algorytmie <i>GAGR</i> . . . . .	82
4.23	Procentowa wartość średniej redukcji liczby generacji oraz procentowe wydłużenie czasu pojedynczej generacji algorytmów <i>GAGR Epi_x</i> względem algorytmu <i>GAGR</i> . . . . .	83
4.24	Zbiory danych wykorzystane w eksperymentach przy wykorzystaniu <i>GADO</i> . . . . .	84
4.25	Parametry algorytmu <i>GADO</i> wykorzystane w eksperymentach . . . . .	84
4.26	Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_1</i> (Pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	85
4.27	Porównanie czasów pojedynczej generacji dla <i>GADO</i> oraz <i>GADO Epi_1</i> . . . . .	88

4.28 Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_2</i> (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	89
4.29 Porównanie czasów pojedynczej generacji w <i>GADO</i> i <i>GADO Epi_2</i> . . . . .	92
4.30 Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_3</i> (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	93
4.31 Porównanie czasów pojedynczej generacji w <i>GADO</i> i <i>GADO Epi_3</i> . . . . .	96
4.32 Optymalne prawdopodobieństwo wystąpienia operacji <i>Epi</i> w algorytmie <i>GADO</i> . . . . .	96
4.33 Procentowa wartość średniej redukcji liczby generacji oraz procentowe wydłużenie czasu pojedynczej generacji algorytmu <i>GADO Epi_x</i> względem <i>GADO</i> . . . . .	97
4.34 Zbiory sekwencji biologicznych wykorzystane w eksperymentach . . . . .	98
4.35 Parametry algorytmu genetycznego wykorzystane w eksperymentach . . . . .	99
4.36 Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_1</i> (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	99
4.37 Porównanie czasów pojedynczej generacji w algorytmie <i>GAMSA</i> i <i>GAMSA Epi_1</i> . . . . .	101
4.38 Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_2</i> (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	102
4.39 Porównanie czasów pojedynczej generacji w <i>GAMSA</i> i <i>GAMSA Epi_2</i> . . . . .	105
4.40 Liczba generacji potrzebna do uzyskania najlepszego wyniku dla poszczególnych zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_3</i> (pogrubioną kursywą zaznaczono minimalną liczbę iteracji dla danego zbioru testowego) . . . . .	106
4.41 Porównanie czasów pojedynczej generacji w algorytmie <i>GAMSA</i> i algorytmie <i>GAMSA Epi_3</i> . . . . .	108
4.42 Optymalne prawdopodobieństwo wystąpienia operacji <i>Epi_x</i> w algorytmie <i>GAMSA</i> . . . . .	109
4.43 Procentowa wartość średniej redukcji liczby generacji oraz procentowe wydłużenie czasu pojedynczej generacji algorytmu <i>GAMSA Epi_x</i> , względem <i>GAMSA</i> . . . . .	109



---

4.44	Optymalne prawdopodobieństwo wystąpienia proponowanych operacji $Epi_x$ w badanych algorytmach . . . . .	110
4.45	Średnia redukcja liczby generacji po zastosowaniu autorskich operacji w algorytmach bazowych . . . . .	111
4.46	Średni wzrost czasu dla pojedynczej generacji po zastosowaniu operacji $Epi_x$ . . .	111

---

## Spis rysunków

1.1	Przykładowa klasyfikacja algorytmów inspirowanych naturą . . . . .	11
1.2	Budowa i miejsce występowania nici DNA [168] . . . . .	15
1.3	Podział algorytmów ewolucyjnych . . . . .	17
1.4	Schemat najczęstszych operacji krzyżowania występujących w algorytmach genetycznych . . . . .	25
1.5	Rodzaje modyfikacji <i>KAG</i> . . . . .	30
2.1	Kodowanie osobnika w algorytmie do grupowania danych . . . . .	37
2.2	Operacja tworzenia osobnika w algorytmie GAMSА . . . . .	40
2.3	Operacje krzyżowania w algorytmie GAMSА . . . . .	41
2.4	Operacja mutacji w algorytmie GAMSА . . . . .	41
3.1	Operacja imitująca dziedziczenie za pomocą prionu – <i>Epi_1</i> . . . . .	46
3.2	Operacja <i>Epi_2</i> – imitująca proces metylacji cytozyny (opis w tekście) . . . . .	48
3.3	Sposób przeprowadzenia operacji <i>Epi_3</i> na genotypie osobnika . . . . .	50
4.1	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_1</i> . . . . .	58
4.2	Zmiany wartości funkcji przystosowania dla generacji w przypadku <i>GAOZ</i> i <i>GAOZ Epi_1</i> . . . . .	59
4.3	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_2</i> . . . . .	61
4.4	Zmiana wartości funkcji przystosowania w kolejnych generacjach <i>GAOZ Epi_2</i> i <i>GAOZ</i> , dla poszczególnych zbiorów danych . . . . .	62
4.5	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_3</i> . . . . .	65

4.6	Zmiana wartości funkcji przystosowania dla kolejnych generacji algorytmu genetycznego, w zależności od zastosowania operacji <i>Epi_3</i> i bez niej . . . . .	66
4.7	Wizualizacja zbiorów użytych w eksperymentach . . . . .	70
4.8	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_1</i> w <i>GAGR</i> . . . . .	73
4.9	Zmiany wartości funkcji przystosowania dla generacji algorytmów <i>GAGR</i> i <i>GAGR Epi_1</i> . . . . .	74
4.10	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_2</i> w <i>GAGR</i> . . . . .	77
4.11	Zmiany wartości funkcji przystosowania w kolejnych generacjach algorytmów <i>GAGR Epi_2</i> i <i>GAGR</i> . . . . .	78
4.12	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_3</i> . . . . .	80
4.13	Zmiany wartości funkcji przystosowania dla generacji algorytmu <i>GAGR</i> i <i>GAGR Epi_3</i> . . . . .	81
4.14	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_1</i> . . . . .	86
4.15	Wartości funkcji przystosowania dla kolejnych generacji algorytmu <i>GADO</i> i <i>GADO Epi_1</i> . . . . .	87
4.16	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_2</i> . . . . .	90
4.17	Zmiana wartości funkcji przystosowania dla kolejnych generacji algorytmu genetycznego, w zależności od zastosowania modyfikacji <i>Epi_2</i> i bez niej . . . . .	91
4.18	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_3</i> . . . . .	94
4.19	Zmiana wartości funkcji przystosowania dla generacji algorytmu genetycznego, w zależności od zastosowania operacji <i>Epi_3</i> i bez niej . . . . .	95
4.20	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów sekwencji testowych w zależności od prawdopodobieństwa wystąpienia operacji <i>Epi_1</i> . . . . .	100
4.21	Wartość funkcji przystosowania dla kolejnych generacji algorytmu <i>GAMSA</i> , w zależności od wystąpienia operacji <i>Epi_1</i> i bez niej . . . . .	101
4.22	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia modyfikacji <i>Epi_2</i> . . . . .	103

---

4.23	Zmiana wartości funkcji przystosowania wraz z kolejnymi generacjami algorytmu genetycznego, w zależności od zastosowania modyfikacji $Epi\_2$ i bez niej . . . . .	104
4.24	Liczba generacji potrzebna do uzyskania najlepszego rezultatu dla zbiorów testowych w zależności od prawdopodobieństwa wystąpienia operacji $Epi\_3$ w $GAMSA$ . . . . .	106
4.25	Zmiana wartości funkcji przystosowania dla generacji algorytmu $GAMSA$ i $GAMSA$ $Epi\_3$ . . . . .	107

---

## Spis algorytmów

1	Klasyczny algorytm genetyczny . . . . .	19
2	Algorytm operacji imitującej dziedziczenie za pomocą prionu – Epi_1 . . . . .	46
3	Zmodyfikowany algorytm krzyżowania z uwzględnieniem symulacji procesu metylacji cytozyny . . . . .	48
4	Algorytm operacji wyłączenia allelicznego – Epi_3 . . . . .	51