Instytut Podstawowych Problemów Techniki
Polskiej Akademii Nauk

mgr inż. Paweł Kieś

Rozprawa doktorska p.t.:

Adaptacyjny dobór metody krzyżowania w binarnym algorytmie genetycznym

Promotor: prof. dr hab. Witold Kosiński

Warszawa 1999

Praca powstała w roku obchodów
50-lecia Informatyki Polskiej

Podziękowania

Autor chciałby serdecznie podziękować promotorowi prof. dr hab. Witoldowi Kosińskiemu za opiekę naukową w czasie stacjonarnych studiów doktoranckich, odbytych w Instytucie Podstawowych Problemów Techniki Polskiej Akademii Nauk w Warszawie w latach 1995-1999.

Autor składa podziękowania wszystkim członkom Samodzielnej Pracowni Optyczno–Komputerowych Metod Mechaniki za życzliwość oraz za udzielanie autorowi silnego wsparcia w jego rozwoju duchowym i naukowym.

Rozprawę przygotowano w ramach Grantu Promotorskiego Nr 8T11C 045 14, finansowanego przez Komitet Badań Naukowych w 1998 roku.

Spis treści

Rozdział 1

O rozprawie

1.1 Tematyka rozprawy

Współcześnie coraz częściej spotykamy się z sytuacją, gdy pojawia się przed nami praktyczny problem, który musimy rozwiązać, aby móc zrealizować powzięte zamierzenie, ale jednocześnie problem ten jest tak nowy i niespotykany dotychczas, iż opisująca go teoria nie zdążyła jeszcze powstać albo jest jeszcze bardzo niekompletna. Z drugiej strony mamy często do czynienia z problemami, których złożoność obliczeniowa jest tak duża, że zastosowanie znanego dokładnego algorytmu nie jest możliwe ze względu na istniejące ograniczenia czasowe. W obu sytuacjach jesteśmy zmuszeni poszukiwać innych podejść do rozwiązywanego problemu. Szczególną grupę stanowią tu *algorytmy probabilistyczne* [20]. Ich istotą jest to, że znajdują one rozwiązanie z prawdopodobieństwem mniejszym od 1, a przebieg ich działania jest zależny od czynników losowych. Oznacza to, że:

- dla pewnych zadań mogą nie znaleźć rozwiązania;
- czasami trzeba ponownie uruchomić algorytm dla tego samego zadania, gdyż nie każde uruchomienie daje rozwiązanie.

Stosowanie algorytmów probabilistycznych zwykle jest kompromisem. Najczęściej rezygnujemy z pewności znalezienia dokładnego rozwiązania być może w bardzo długim czasie, na rzecz znalezienia rozwiązania przybliżonego, jedynie w większości przypadków, ale za to w krótszym czasie. Oczywiście, jak wspomniano wcześniej, w pewnych sytuacjach algorytmy probabilistyczne są jedynymi dającymi jakiekolwiek rozwiązanie.

Dziedzina wiedzy nazywana *sztuczną inteligencją*, a zwłaszcza jej dział nazywany *algorytmami ewolucyjnymi*, dostarcza nam wiele algorytmów probabilistycznych, których wspólną cechą jest to, że opierają się one na pomysłach "podpatrzonych" w przyrodzie u istot żywych. Szczególną grupą algorytmów ewolucyjnych są Binarne Algorytmy Genetyczne ($\mathcal{BAG}$), będące tematem niniejszej rozprawy.

Działanie algorytmów genetycznych jest uzależnione od dużej liczby różnych parametrów. Okazuje się, że dla każdego zadania zwykle są odpowiednie inne wartości parametrów. Stąd bierze się dążenie do adaptacji parametrów $\mathcal{BAG}$ do rozwiązywanego zadania, w celu zwiększenia jego szeroko rozumianej sprawności. Podejście to potwierdza udowodnione ostatnio twierdzenie *"No Free Lunch Theorem"* mówiące, że nie istnieje uniwersalny algorytm jednakowo dobry do rozwiązywania każdego zadania [31]. Dlatego w niniejszej rozprawie będziemy się zajmowali adaptacją parametrów $\mathcal{BAG}$, a w szczególności *adaptacją off-line*, która, w przeciwieństwie do *adaptacji on-line*, polega na tym, że parametry algorytmu genetycznego nie są zmieniane w trakcie jego pracy, ale wyznaczane jednokrotnie przed jego uruchomieniem. Podstawą do wyznaczania parametrów będą pomiary własności rozwiązywanego zadania.

1.2 Cel i zakres rozprawy

W literaturze możemy spotkać opisy wielu udanych metod adaptacji on-line parametrów algorytmu genetycznego. Natomiast nie ma żadnych informacji o adaptacji off-line, a jedynie podejmowane są metody pomiarów właściwości rozwiązywanego zadania, bez konstruktywnych wniosków na temat doboru na tej podstawie parametrów $\mathcal{BAG}$. Eksperymenty przeprowadzone przez autora pokazują, że w pewnych sytuacjach adaptacja off-line algorytmu genetycznego jest możliwa i celowa, gdyż prowadzi do polepszenia jego sprawności. Dlatego sformułowaną następującą tezę niniejszej rozprawy:

> Dobór permutacji kodu rozwiązań w binarnym algorytmie genetycznym metodą adaptacji off-line może poprawić sprawność działania tego algorytmu.

Aby wykazać słuszność powyższej tezy postawiono następujące cele rozprawy:

1. Zmodyfikowanie klasycznego $\mathcal{BAG}$ przez wprowadzenie kodowania z permutacją bitów kodu.
2. Zbadanie wpływu permutacji na sprawność $\mathcal{BAG}$, oraz określenie, czy wpływ ten jest wystarczająco duży, aby permutacja mogła być wykorzystywana jako adaptowany parametr $\mathcal{BAG}$.
3. Klasyfikacja i porównanie właściwości znanych metod adaptacji algorytmów genetycznych.
4. Opracowanie, analiza i porównanie metod oszacowania sprawności $\mathcal{BAG}$ bez jego uruchamiania, czyli na podstawie pomiarów funkcji przystosowania (off-line).
5. Opracowanie ogólnej korelacyjnej metody adaptacji off-line permutacji kodu, wykorzystującej oszacowanie sprawności $\mathcal{BAG}$.
6. Zbadanie właściwości metaalgorytmu genetycznego użytego w korelacyjnej metodzie adaptacji off-line $\mathcal{BAG}$.
7. Stworzenie oprogramowania do pomiaru funkcji przystosowania oraz implementacja korelacyjnej metody adaptacji off-line wykorzystującej metaalgorytm genetyczny.
8. Dowiedzenie na drodze eksperymentalnej skuteczności proponowanej metody adaptacji oraz podjęcie próby analizy jej działania.

1.3 Streszczenie rozprawy

Rozprawa prezentuje ogół zagadnień związanych z korelacyjnymi metodami adaptacji off-line $\mathcal{BAG}$, ze szczególnym uwzględnieniem metod pomiaru funkcji przystosowania. Praca zawiera 9 rozdziałów oraz bibliografię. W Rozdziale 1 – niniejszym, przedstawiono w skrócie całą rozprawę, jej główne idee i genezę.

Rozdziały 2, 3 i 4 pokazują aktualny stan wiedzy w zakresie algorytmów genetycznych oraz metod ich adaptacji. W Rozdziale 2 omówiono strukturę i zasadę działania algorytmów genetycznych. Zwrócono uwagę na umiejscowienie ich w szerszej grupie algorytmów ewolucyjnych. W Rozdziale 3 wprowadzono elementy teorii algorytmów genetycznych, zwłaszcza podjęto zagadnienia mające szczególne zastosowanie w adaptacji $\mathcal{BAG}$. Dokonano próby usystematyzowania i uściślenia tych zagadnień związanych z $\mathcal{BAG}$, które zwykle są przedstawiane w literaturze w sposób niejednoznaczny i nieprecyzyjny. W Rozdziale 4 omówiono ideę adaptacji $\mathcal{BAG}$. Wiele pojęć związanych z adaptacją

zostało zdefiniowanych i uściślonych. Na podstawie literatury dokonano klasyfikacji opracowanych dotychczas metod adaptacji $\mathcal{BAG}$ oraz metod pomiaru funkcji przystosowania, przy czym starano się umiejscowić w tej strukturze także korelacyjne metody adaptacji.

Rozdziały 6, 7, 8 i 9 dotyczą nowych zagadnień, będących oryginalnym wynikiem pracy badawczej autora. W Rozdziale 6 przedstawiono zagadnienie adaptacji off-line oraz wprowadzono nową, korelacyjną metodę adaptacji off-line parametrów $\mathcal{BAG}$ wykorzystującą heurystyczne funkcje pomiarowe do pomiarów funkcji przystosowania. Zaproponowano także użycie permutacji kodu jako nowego adaptowalnego parametru $\mathcal{BAG}$. W Rozdziale 7 zdefiniowano trzy funkcje pomiarowe do, które mogą być wykorzystywane w korelacyjnych metodach adaptacji $\mathcal{BAG}$, z permutacja kodu jako adaptowanym parametrem. Dodatkowo zdefiniowano pomocniczą metodę pomiarową, mającą pomóc w zrozumieniu działania proponowanych metod adaptacji. W Rozdziale 8 szczegółowo omówiono przeprowadzone badania eksperymentalne. Zwrócono uwagę na precyzyjne określenie warunków eksperymentu oraz przyjętych założeń. W Rozdziale 9 sformułowano wnioski wynikające z przeprowadzonych badań, a także omówiono możliwe zastosowania osiągniętych wyników. Wskazano również kierunki dalszych badań, mających na celu rozszerzenie wiedzy na temat korelacyjnych metod adaptacji oraz wyjaśnienie powstałych w trakcie badań niejasności i wątpliwości.

Rozdział 2

Wprowadzenie

2.1 Geneza algorytmów genetycznych

W tym rozdziale omówimy w skrócie genezę oraz strukturę i działanie algorytmów genetycznych. Nawiążemy także do innych algorytmów wykonujących te same zadania. Z konieczności wiele zagadnień związanych z algorytmami genetycznymi jedynie zasygnalizujemy nie wdając sie w dokładne wyjaśnienia. Dlatego odsyłamy dociekliwego czytelnika do łatwo dostępnych polskojęzycznych książek, które na początku wprowadzają od podstaw ideę algorytmów genetycznych, a następnie omawiają bardziej szczegółowo większość z poruszanych tu zagadnień. Najważniejsze z tych książek to prace Michalewicza [37], Goldberga [28] i Cytowskiego [39].

2.1.1 Algorytmy optymalizacji

Algorytm genetyczny należy do szerokiej klasy *algorytmów ewolucyjnych*, a te z kolei zaliczają sie do algorytmów optymalizacyjnych, które należą do najszerszej grupy – do grupy algorytmów przeszukujących (patrz [13]). W najogólniejszym podejściu celem optymalizacji jest znalezienie takiego rozwiązania postawionego problemu, które maksymalizuje wartość pewnej funkcji skalarnej $f : X \to \mathbb{R}$, określonej na pewnej *przestrzeni rozwiązań* X. Funkcja powyższa jest nazywana zwykle *funkcją jakości.*

W funkcji jakości łączymy różne kryteria, których spełnienia wymagamy od pożądanego rozwiązania, jak również określamy ich względną ważność poprzez dobór odpowiednich współczynników wagowych. Dzięki zdefiniowaniu funkcji jakości, możemy łatwo powiedzieć, które z otrzymanych rozwiązań jest lepsze.

Najczęściej nie wszystkie elementy przestrzeni rozwiązań są dopuszczalne, co oznacza, że poza masymalizacją funkcji jakości pożądane rozwiązanie musi jeszcze spełniać pewne dodatkowe warunki lub ograniczenia, nazywane więzami [28]. Zwykle warunki te można zapisać w postaci układu nierówności.

$$h_i(x) \geq 0, \quad i = 1, 2, \ldots, n. \tag{2.1}$$

Wszystkie rozwiązania spełniające ten układ nazywamy *rozwiązaniami dopuszczalnymi.*

Najprostszy sposób postępowania z rozwiązaniami niedopuszczalnymi sprowadza się do ich odrzucania, przez przypisanie im bardzo niskiej wartości funkcji jakości. Jednak w sytuacji, gdy rozwiązań dopuszczalnych jest stosunkowo niewiele, albo warunki ograniczeń są wysoce skomplikowane, możemy próbować znaleźć rozwiązanie które, jedynie w niewielkim stopniu nie spełnia ograniczeń, a następnie próbować to rozwiązanie *naprawić* za pomocą *algorytmu naprawy* [37]. Naprawa zwykle sprowadza się do poszukania najbliższego rozwiazania dopuszczalnego w pewnym otoczeniu rozwiązania otrzymanego.

Inna metoda polega na odpowiednim przekształceniu funkcji jakości. Aby zmierzyć w jakim stopniu otrzymane rozwiązanie jest niedopuszczalne wprowadzamy *funkcję kary* $g(x)$ w następującej

postaci [28]:

$$g(x) = \sum_{1}^{n} \alpha_i \Phi(h_i(x)), \quad \text{gdzie} \quad \sum_{1}^{n} \alpha_i = 1 \tag{2.2}$$

Współczynniki α_i określają względną ważność każdego z ograniczeń. Funkcja Φ powinna być równa zeru dla argumentów dodatnich, natomiast dla argumentów ujemnych powinna być nierosnąca. Korzystając z tak zdefiniowanej funkcji kary definiujemy zastępczą funkcję jakości:

$$\tilde{f}(x) = f(x) - rg(x) \tag{2.3}$$

Przy spełnieniu pewnych warunków ciąg rozwiązań maksymalizujących $\tilde{f}(x)$ jest zbieżny do rozwiązania maksymalizującego $f(x)$, gdy współczynnik kary r dąży do nieskończoności [28]. W ten sposób każdy problem z ograniczeniami możemy przekształcić na odpowiedni problem bez ograniczeń.

Kolejna metoda radzenia sobie z ograniczeniami polega na takim przekształceniu przestrzeni rozwiązań, aby przekształconej przestrzeni każde rozwiązanie było dopuszczalne.

W niniejszej pracy, dla uproszczenia, będziemy zakładali, że niezbędne przekształcenia funkcji przystosowania albo przestrzeni rozwiązań zostały już dokonane i stąd wszystkie elementy przestrzeni rozwiązań są rozwiązaniami dopuszczalnymi.

Najprostszy algorytm optymalizacyjny polega na zwykłym obliczeniu rozwiązania optymalnego. Stosuje się go w tych przypadkach, gdy możliwe jest algebraiczne wyprowadzenie odpowiednich wzorów (np. gdy funkcja jakości ma postać wielomianu drugiego lub czwartego stopnia). Zwykle jednak, zadanie z którym się spotykamy w praktyce jest znacznie trudniejsze i powyższe rozwiązanie okazuje się niemożliwe. Staramy wtedy uzyskać jakiekolwiek dopuszczalne rozwiązanie, a następnie drogą kolejnych ulepszeń, próbować przybliżyć się do rozwiązania optymalnego. Ciąg rozwiązań prowadzący od rozwiązania początkowego do optymalnego tworzy *ścieżkę* w przestrzeni rozwiązań.

Sposób znajdowania kolejnego przybliżenia rozwiązania na podstawie przybliżeń wyznaczonych uprzednio jest istotą każdego algorytmu optymalizacji. Podstawowym algorytmem stosowanym tutaj jest *algorytm największego wzrostu*, który polega tym, że kolejnym przybliżeniem staje się najlepsze rozwiązanie będące w sąsiedztwie przybliżenia aktualnego. Inaczej mówiąc, aby uzyskać nowe przybliżenie poddajemy aktualne przybliżenie działaniu przekształcenia typu $m : X \times U(X) \to X$, gdzie $U(X)$ jest pewnym otoczeniem aktualnego przybliżenia. Jest oczywiste, że algorytm ten może być stosowany wyłącznie wtedy, gdy mamy pewność, iż znajdujemy się bardzo blisko maksimum globalnego, w przeciwnym razie znajdywane jest maksimum lokalne leżące najbliżej punktu startowego.

Próbą zaradzenia powyższej wadzie algorytmu najwiekszego wzrostu jest wprowadzenie do obliczeń szumu, tak, aby wynik działania algorytmu nie zależał już wyłącznie od punktu startowego, ale także od czynnika losowego. Dzięki temu, jeżeli nawet algorytm doprowadzi do maksimum lokalnego, to zawsze istnieje możliwość "ucieczki" z takiego punktu. Zwykle amplituda szumu jest zmniejszana w kolejnych krokach obliczeń, co umożliwia uzyskanie stabilnego rozwiązania po odpowiedniej liczbie kroków. Przykładem zastosowania opisanego pomysłu jest metoda symulowanego wyżarzania.

Naszkicowane powyżej algorytmy mają jedną wspólną cechę: w danym kroku obliczeń jest przetwarzane tylko jedno rozwiązanie. Dopuszczenie jednoczesnego przetwarzania grupy rozwiązań prowadzi nas do algorytmów ewolucyjnych.

2.1.2 Algorytmy ewolucyjne

Wśród badaczy algorytmów ewolucyjnych przyjęte jest używanie terminologii zaczerpniętej z biologii. I tak grupa rozwiązań przetwarzana w danym kroku jest nazywana *populacją*, rozwiązania należące do populacji nazywane są *osobnikami*, zamiast o kroku obliczeń mówi się o *pokoleniu*.

Nowe osobniki zwane *potomkami* są otrzymywane z osobników z poprzedniej populacji, zwanych *rodzicami*. Zamiast o funkcji jakości mówimy o *funkcji przystosowania*, a wartość funkcji przystosowania dla konkretnego osobnika nazywana jest jego *przystosowaniem*.

Idea algorytmu ewolucyjnego jest prosta. Tworzymy w sposób losowy początkową populację osobników. Następnie każdego osobnika staramy się ulepszyć. Możemy to robić metodami opisanymi powyżej, możemy także wykorzystywać informację specyficzne dla zadania, na przykład różnorodne heurystyki. Możemy, w końcu, modyfikować osobniki w sposób czysto losowy, mając nadzieję, że uda nam się przypadkiem uzyskać poprawę. Taka operacja losowej modyfikacji osobnika nazywa się *mutacją*.

Kolejny krok ma na celu zapewnienie możliwości "wymiany doświadczeń" pomiędzy osobnikami i odpowiada zastosowaniu przekształcenia typu $c : X^k \to X^k$ nazywanego *krzyżowaniem* (zwykle $k = 2$). Łączymy osobniki (rodziców) w pary (lub w większe grupy), a następnie na pdstawie każdej pary (grupy) rodzicielskiej budujemy parę (grupę) potomków w ten sposób, że obaj potomkowie mają w sobie część właściwości jednego rodzica i część drugiego. Mamy przez to nadzieję, że najlepsze cechy każdego z rodziców połączą się w jednym osobniku potomnym.

Następnym krokiem jest ocena każdego potomka, polegająca na wyliczeniu wartości funkcji przystosowania. Po tym etapie jesteśmy w stanie dokonać selekcji nowego pokolenia. W najprostszej wersji selekcja przebiega w ten sposób, że osobniki lepiej przystosowane (o wyższej wartości funkcji przystosowania) mają większą szansę dostania się do nowego pokolenia, czyli przeżycia. Po utworzeniu nowego pokolenia wykonujemy dalsze operacje jak dla populacji początkowej zamykając w ten sposób pętlę algorytmu.

Po każdym kroku oceny dokonujemy sprawdzenia warunku zakończenia. Zwykle jest tym warunkiem wygenerowanie określonej liczby pokoleń. Możemy również obliczać współczynnik poprawy aktualnego pokolenia w stosunku do poprzedniego i przerywać działanie algorytmu, gdy przez pewną liczbę pokoleń będzie się on utrzymywał poniżej zadanej wartości minimalnej. W sytuacjach laboratoryjnych, gdy znamy wartość funkcji przystosowania dla osobnika optymalnego, możemy jako warunek zakończenia wykorzystać osiągnięcie tej wartości (np. z zadaną dokładnością) przez najlepszego osobnika w populacji.

Można zauważyć, że dla każdego typu problemu musimy zdefiniować metody mutacji i krzyżowania, które będą zwykle uzależnione od reprezentacji osobników . Inne będą również metody ulepszania przedstawicieli populacji – zazwyczaj będą one zależne od przyjętej topologii przestrzeni X. Dlatego też powstał pomysł, aby ujednolicić działanie algorytmu ewolucyjnego dla różnych typów problemów. Pomysł ten prowadzi do algorytmów genetycznych.

2.1.3 Algorytmy genetyczne

Algorytm genetyczny charakteryzuje się tym, że elementy przestrzeni rozwiązań są w nim kodowane za pomocą wektorów liczb o ustalonej długości. Wektory te nazywane są chromosomami albo łańcuchami, ze względu na podobieństwo do chromosomów składających się z cząsteczek kwasu DNA, które w organizmach żywych służą do przechowywania i przekazywania materiału genetycznego.

W praktyce stosuje się dwa typy algorytmów genetycznych (patrz [37]):

- **zmiennoprzecinkowe** – w których chromosomy są wektorami liczb zmiennoprzecinkowych, oraz
- **binarne** – w których chromosomy są wektorami liczb dwójkowych.

W algorytmie genetycznym jedynym problemem, zależnym od rozwiązywanego zadania, jest sposób zakodowania elementów przestrzeni rozwiązań w chromosomy. Poza tym działanie algorytmu jest prawie niezależne od problemu, a operacje krzyżowania i mutacji działają nie na rozwiązaniach, alè na chromosomach. W przypadku algorytmów zmiennoprzecinkowych operacje te definiuje się

```
i := 0
tworzenie G_i
ocena G_i
while not warunek zakończenia do
  i:=i+1
  selekcja G_i z G_{i-1}
  krzyżowanie G_i
  mutacja G_i
  ocena G_i
end
```

Rys. 2.1: Schemat binarnego algorytmu genetycznego

zwykle korzystając z pewnych operacji arytmetycznych wykonywanych z pewnym prawdopodobieństwem na liczbach wchodzących w skład chromosomu. W binarnej wersji algorytmu dostępne operacje są znacznie mniej zróżnicowane i sprowaczają się wyłacznie do różnych modyfikacji pojedyńczych bitów lub ich ciągów.

W pracy Wolperta i Macready'ego [31] dowiedziono, że nie da się stworzyć jednego, uniwersalnego algorytmu do rozwiązywania wszelkich zagadnień, natomiast zawsze można starać się dopasować algorytm do rozwiązywanego problemu. Dzięki temu możemy uzyskać "wersję" algorytmu działającą wystarczająco dobrze dla pewnej grupy problemów, albo nawet dla pewnego konkretnego problemu.

Takie dostosowanie algorytmu genetycznego jest możliwe poprzez modyfikację jego parametrów, zamiast samej zasady działania. Można w tym celu stosować rozmaite algorytmy adaptacjne, modyfikujące parametry algorytmu genetyczxnego w trakcie jego pracy, można także badać funkcję przystosowania różnymi metodami i na tej podstawie dobierać odpowiednie wartości parametrów przed uruchomieniem algorytmu.

W niniejszej pracy rozważania nasze ograniczymy wyłącznie do binarnego algorytmu genetycznego ($\mathcal{BAG}$). Dostosowując algorytm do zadania będziemy brali pod uwagę tylko takie parametry algorytmu, które wpływają na działanie operatora krzyżowania.

2.2 Struktura i działanie binarnego algorytmu genetycznego

2.2.1 Struktura i działanie

Binarny algorytm genetyczny, jak wspomniano wcześniej, jast zaliczany do grupy algorytmów ewolucyjnych. Elementy przestrzeni rozwiązań są tutaj reprezentowane przez wektory binarne – złożone z liczb dwójkowych, nazywane także *łańcuchami binarnymi*, *chromosomami* lub *osobnikami*. W standardowym $\mathcal{BAG}$ przyjmuje się, że wszystkie chromosomy mają tę samą długość N, a więc zbiorem wszystkich chromosomów jest zbiór $\mathcal{B} = \{0,1\}^N$.

Wektor chromosomów o ustalonej długości tworzy *populację* nazywaną inaczej *pokoleniem*. Najogólniej działanie $\mathcal{BAG}$ można opisać następująco: w kolejnych krokach algorytm generuje aktualną populację G_i na podstawie populacji poprzedniej G_{i-1}, poczynając od pewnej populacji początkowej G_0. Działanie zostaje przerwane, gdy zostaje spełniony warunek zakończenia.

Dokładny schemat $\mathcal{BAG}$ przedstawiono na Rys. 2.2.1. Widać, że proces tworzenia nowej populacji składa się z pięciu podstawowych kroków:

- sprawdzenia warunku zakończenia,

- oceny,
- selekcji,
- krzyżowania,
- mutacji.

W fazie sprawdzenia warunku zakończenia możliwe jest opuszczenie głównej pętli $\mathcal{BAG}$ w przypadku spełnienia tego warunku. Warunek zwykle ten ma charakter heurystyczny i powinien powodować przerwanie działania algorytmu zarówno w przypadku przypuszczenia, że optimum globalne zostało już znalezione, jak i w przypadku przypuszczenia, że dalsza praca algorytmu nie ma sensu gdyż algorytm "wpadł" w optimum lokalne i jest mało prawdopodobne, że z niego się wydostanie.

Faza oceny polega na wyliczeniu przystosowania każdego osobnika. Podobnie jak w biologii, określa ona stopień przystosowania danego osobnika do warunków stwarzanych przez środowisko, czyli jego zdolność do przetrwania, albo inaczej: jego udział w tworzeniu nowego pokolenia.

W fazie selekcji przeprowadzany jest wybór tych osobników z poprzedniego pokolenia, które przejdą do nowego. Metoda wyboru ma charakter losowy. Zapewnia ona, że wartość oczekiwana liczby kopii danego osobnika w nowej populacji jest tym wyższa, im wyższe jest jego przystosowanie. Czyli, korzystając z analogii biologicznej, można powiedzieć, że bardziej przystosowany osobnik powinien mieć większą szansę przetrwania.

Krzyżowanie odpowiada za "wymianę doświadczeń" znaną z algorytmów ewolucyjnych. Polega ono na wzajemnej wymianie fragmentów chromosomów między osobnikami losowo dobranymi w pary.

Mutacja jest przeprowadzaną losowo próbą ulepszenia każdego osobnika w populacji. Do jej zadań należy zapobieganie przedwczesnej zbieżności populacji do lokalnego maksimum.

Poniżej zostaną omówiane zagadnienia istotne z punktu widzenia niniejszej pracy. Natomiast dokładniejszy opis każdego z elementów $\mathcal{BAG}$ można znaleźć w książkach: Michalewicza [37] i Goldberga [28].

2.2.2 Wybór nowego pokolenia – selekcja

Selekcja może być zdefiniowana jako odwzorowanie typu $\mathcal{B}^{M_1} \rightarrow \mathcal{B}^{M_2}$ tworzące nową populację osobników na podstawie starej. W standardowym $\mathcal{BAG}$przyjmujemy, że liczebność populacji jest niezmienna w trakcie pracy algorytmu, czyli $M_1 = M_2 = M$.

W większości stosowanych alorytmów selekcji będziemy się spotykać z omówianymi poniżej terminami:

- **funkcja skalująca** – ponieważ większość algorytmów selekcji wymaga, aby wartość funkcji przystosowania była większa od zera, oryginalne wartości (wyliczone na podstawie definicji problemu) muszą być niekiedy poddane skalowaniu. Za pomocą funkcji skalujących można także wpływać na różnorodność populaji, uwypuklając albo zmniejszając różnice pomiędzy osobnikami. Omówienie ważniejszych metod skalowania można znaleźć u Goldberga [28] na str. 138;
- **względne przystosowanie osobnika** – stosunek przystosowania danego osobnika do średniego przystosowania całej populacji;
- **prawdopodobieństwo przeżycia** p_{si} – prawdopodobieństwo, że i-ty osobnik zostanie wybrany do nowego pokolenia;
- **wartość oczekiwana liczby kopii osobnika w nowej populacji** m_{ci} – obliczana ze wzoru: $m_{ci} = p_{si}M$.

W standardowym $\mathcal{BAG}$ wyboru osobników, które wejdą w skład nowego pokolenia (przeżyją) dokonuje się tzw. metodą ruletki. Wyobraźmy sobie koło ruletki. Znając wartości funkcji przystosowania osobników, możemy każdemu z nich przydzielić wycinek koła zgodnie z zasadą, że kąt wycinka powinien być większy dla większej wartości funkcji przystosowania. Jeśli zakręcimy teraz kołem to osobniki o większej wartości funkcji przystosowania zostaną wybrane z większym prawdopodobieństwem. Oczywiście kąty spełniające powyższe warunki możemy wyznaczyć na nieskończenie wiele sposobów.

Podstawowa metoda ruletki ma wiele wad. Na przykład łatwo powoduje zdominowanie całej populacji przez jednego "wybitnego" osobnika. Po za tym rzeczywista liczba kopii każdego osobników może się znacznie różnić od oczekiwanej, co przy dużym podobieństwie osobników może nawet uniemożliwić osiągnięcie zbieżności.

Z tego względu zostało zaproponowanych i przetestowanych wiele udoskonaleń tej metody. W pracy doktorskiej de Jonga [6] czytelnik może znaleźć analizę działania wielu modyfikacji $\mathcal{BAG}$, w tym także algorytmu selekcji. Jedna z nich jest nazywana *elitarystycznym modelem selekcji.* Polega ona na tym, że zawsze jest zachowywany najlepszy osobnik z poprzedniego pokolenia. W przypadku, gdy konieczne jest znalezienie dokładnej wartości maksimum, czyli istotne jest lokalne poszukiwanie, pozwala ona na radykalne skrócenie czasu działania algorytmu. Dzieje się tak dlatego, że klasyczna metoda ruletki nie zabezpiecza najlepszych osobników przed zniknięciem z populacji w trakcie długotrwałych poszukiwań lokalnych. Gdy tak się stanie, trzeba zacząć poszukiwania utraconych osobników od początku. W przypadku wielomodalnych funkcji przystosowania sprawność $\mathcal{BAG}$ z modelem elitarystycznym może spaść, gdyż, jak przypuszcza de Jong, poprawa lokalnego poszukiwania następuje kosztem "perspektywy globalnej".

Badania nad nowymi metodami selekcji mającymi na celu zredukowanie rozrzutu metody ruletki przeprowadziła Brindle w swojej pracy doktorskiej [10]. Można znaleźć tam omówienie deterministycznej metody selekcji, oraz różnych metod, w których zmniejszono wpływ czynnika losowego na wynik selekcji.

2.2.3 Wymiana doświadczeń – krzyżowanie

Jak wspomniano wcześniej, krzyżowanie możemy zdefiniować jako odwzorowanie $c : \mathcal{B}\times\mathcal{B} \to \mathcal{B}\times\mathcal{B}$, przekształcające parę rodziców w parę potomków.

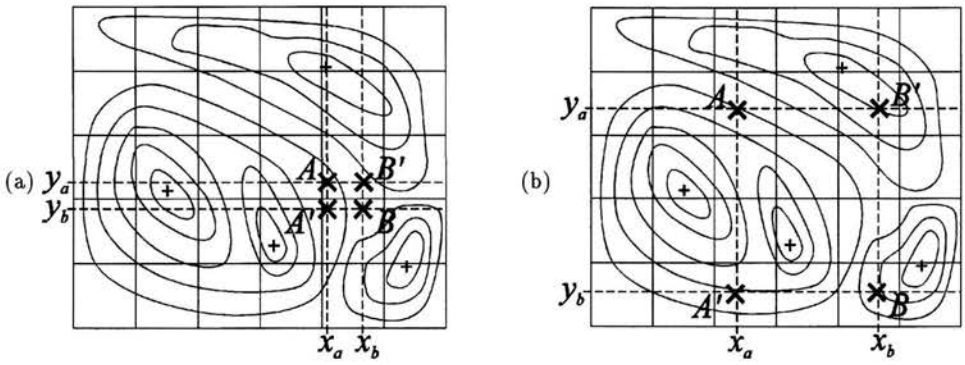
W standardowym $\mathcal{BAG}$ jest stosowana najprostsza wersja krzyżowania – *krzyżowanie jednopunktowe.* Polega ona na losowym połączeniu chromosomów z populacji w pary. Dla każdej pary losowana jest pozycja krzyżowania i_c – liczba naturalna ze zbioru $\{1,\ldots,N\}$, dla chromosomu długości N, przy czym dla wartości $i_c = N$ otrzymujemy krzyżowanie tożsamościowe. Następnie bity o numerach pozycji większych od i_c są wymieniane pomiędzy sparowanymi osobnikami, jak widać na przykładzie chromosomów A i B dla $i_c = 9$ (po przeprowadzeniu krzyżowania):

$$\begin{array}{ll} A: & a\,a\,a\,a\,a\,a\,a\,a\,a\,b\,b\,b\,b\,b\,b\,b \\ & \qquad\qquad\qquad\quad \updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow \\ B: & b\,b\,b\,b\,b\,b\,b\,b\,b\,a\,a\,a\,a\,a\,a\,a \end{array}$$

Podobnie definiuje się krzyżowanie dwupunktowe, które może być rozpatrywane jako złożenie dwóch krzyżowań jednopunktowych. W tym przypadku losujemy dwie liczby: $i_{c1}, i_{c2} \in \{0,\ldots,N\}$ i wymieniamy bity na pozycjach od i_{c1} do i_{c2} włącznie. Przykładowo dla $i_{c1} = 4$ i $i_{c2} = 11$ mamy:

$$\begin{array}{ll} A: & a\,a\,a\,b\,b\,b\,b\,b\,b\,b\,a\,a\,a\,a\,a \\ & \qquad\quad \updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow\updownarrow \\ B: & b\,b\,b\,a\,a\,a\,a\,a\,a\,a\,b\,b\,b\,b\,b \end{array}$$

Widać, że dla krzyżowania jednopunktowego dokonujemy wyboru jednej z $N-1$ operacji, a dla dwupunktowego – jednej z $\binom{N}{2}$. Propozycje innych metod krzyżowania czytelnik może znaleźć



Rys. 2.2: Przykład znaczenia krzyżowania na płaszczyźnie $\mathbb{R}^2$. Krzyżowanie dwóch obiektów A i B polega tu na wymianie między nimi współrzędnej y. W wyniku powstają obiekty A' i B'. Widać, że obszar przeszukiwania jest mniejszy przy krzyżowaniu bardziej "podobnych" obiektów.

u Goldberga [28] na str. 134; dodatkowo uogólninione metody krzyżowania zostaną omówione w dalszych częściach rozprawy.

Zazwyczaj przyjmuje się, że operacją krzyżowania nie jest obligatoryjna, ale wykonywana z *prawdopodobieństwem krzyżowania* p_c zwykle równym liczbie od 0.8 do 1. Oznacza to, że dopuszcza się możliwość pozostawienia pary w stanie niezmienionym.

Ze względu na łatwość wymiany łańcuchów o dowolnych długościach i położeniu przyjęto, że algorytm genetyczny wykorzystywany w niniejszej pracy korzysta z krzyżowania dwupunktowego.

Idea krzyżowania opiera się na założeniu, że jest duża szansa, że potomek stworzony z fragmentów rodziców może osiągnąć wyższe przystosowanie niż jego rodzice. Oczywiście można wskazać problemy, w których krzyżowanie wogóle nie polepsza działania $\mathcal{BAG}$ i może być zastąpione makromutacją, która polega na zastąpieniu fragmentu chromosomu losowym lancuchem binarnym (patrz praca Jonesa [36]).

Drugim zadaniem krzyżowania, oprócz "wymiany doświadczeń" jest zawężanie przeszukiwanego przez algorytm obszaru w miarę jak chromosomy należące do populacji stają się do siebie coraz bardziej podobne (patrz Rys. 2.2). Można zauważyć że w trakcie krzyżowania modyfikacji mogą podlegać wyłącznie te bity łańcucha, które są różne u obu rodziców. Takie działanie w połączeniu z selekcją, przy braku mutacji, nieuchronnie prowadziłoby do zdominowania całej populacji przez jednego osobnika. Jeżeli w trakcie działania algorytmu okazałoby się, że wszystkie chromosomy w populacji mają na pewnej pozycji bit o takiej samej wartości 0 lub 1, to nie byłoby już żadnej możliwości zmiany wartości tego bitu w następnych pokoleniach. Zjawisku temu skutecznie zapobiegają losowe zakłócenia wprowadzane przez mutację.

Widać z powyższego, że dla populacji o skończonej liczebności, krzyżowanie wraz z selekcją powodują utratę "materiału genetycznego", pomimo że samo krzyżowanie nie powoduje żadnych strat informacji. Nie ma tej wady algorytm teoretyczny, pracujący na populacji nieskończonej, opisywany w pracy Qi i Palmieriego [24]. W takim przypadku krzyżowanie i selekcja oddziaływują jedynie na "gęstość" występowania danego chromosomu w populacji, która zawsze jest większa od zera.

2.2.4 Losowe zakłócenia – mutacja

Podstawowy algorytm mutacji polega na zanegowaniu niektórych bitów w osobnikach należących do populacji. Modyfikacja ta jest dokonywana z pewnym małym *prawdopodobieństwem mutacji*

p_m. Zwykle przyjmuje się $p_m < 0.1$. Stąd wprost wynika średnia liczba zmutowanych bitów w populacji:

$$n_m = p_m N M \tag{2.4}$$

gdzie N jest długością chromosomu, a M – liczebnością populacji.

Jak powiedziano wyżej, mutacja zapobiega wystąpieniu niekorzystnych ubocznych efektów krzyżowania. Jest ona także odpowiedzialna za intensywność przeszukiwania nowych obszarów przestrzeni rozwiązań. Intensywność przeszukiwania jest tym wyższa im większa jest wartość p_m. Stąd wydaje się słuszna koncepcja zmniejszania p_m w miarę przybliżania się do maksimum globalnego. Doświadczenia praktyczne wskazują, że p_m należy dobierać w zależności od rozwiązywanego problemu. Można do tego celu wykorzystać intuicję badacza, ale możliwe jest także zastosowanie np. sieci neuronowych lub systemów ekspertowych.

Rozdział 3

Elementy teorii algorytmów genetycznych

3.1 Funkcja przystosowania i funkcja kodująca

Załóżmy że naszym zadaniem jest znaleźć taki element x_{opt} pewnej **przestrzeni poszukiwań** X, że:

$$\hat{f}(x_{opt}) = \max_{x \in X} \hat{f}(x). \tag{3.1}$$

gdzie: $\hat{f} : X \to \mathbb{R}$ jest funkcją jakości (funkcją przystosowania).

W ogólnym przypadku nie nakładamy na X żadnych ograniczeń. Aby jednak można było do tego problemu zastosować algorytm genetyczny, należy zbiór X przekształcić na zbiór o skończonej liczbie elementów (zbiór dyskretny). Przekształcenie to wykonujemy w trzech krokach:

1. definiujemy relację równoważności $R \subset X \times X$, tak, aby przestrzeń ilorazowa $\tilde{X} = X/R$ była dyskretna;
2. bierzemy zbiór $\tilde{X}$, będący rodziną klas abstrakcji relacji R, jako zastępczą dyskretną przestrzeń poszukiwań;
3. definiujemy sposób obliczania przystosowania elementów $\tilde{x} \in \tilde{X}$ poprzez funkcję $\tilde{f} : \tilde{X} \to \mathbb{R}$; zwykle przyjmujemy, że $\tilde{f}(\tilde{x}) \equiv \hat{f}(x)$, gdzie $x \in \tilde{x}$ jest wybranym w ustalony sposób elementem klasy abstrakcji $\tilde{x}$.

Dla uproszczenia rozważań bez straty ogólności, w dalszej części niniejszej pracy będziemy zakładali, że X jest już zbiorem skończonym i nie wymaga powyższych przekształceń.

Kolejnym krokiem jest zakodowanie elementów X w postaci N-elementowych ciągów kodowych składających się ze zbioru symboli $V = \{0, 1\}$. Ciągi kodowe są nazywane łańcuchami (wektorami) dwójkowymi lub **chromosomami**. Zbiór wszystkich N-elementowych chromosomów oznaczamy przez $\mathcal{B} = \{0, 1\}^N$ i nazywamy **przestrzenią kodową**. Stąd widać, że $|\mathcal{B}| = |V|^N = 2^N$. Elementy przestrzeni kodowej będziemy oznaczali jak odpowiednie liczby dwójkowe – tzn. za pomocą ciągów zer i jedynek, bez oddzielania składowych przecinkami, jak na przykład dla $N = 16$:

$$\underline{b} = 0110010101001110 \in \mathcal{B}.$$

Dodatkowo przyjmujemy, że funkcja $ones : \mathcal{B} \to \{0, \ldots, N\}$ dla każdego chromosomu z $\mathcal{B}$ zwraca liczbę bitów o wartości "1", czyli dla powyższego chromosomu otrzymamy $ones(\underline{b}) = 8$.

Wybór długości chromosomu N zależy od tego, z jaką dokładnością chcemy przeszukiwać przestrzeń X. Wymagana jest przy tym pewna wiedza na temat kształtu funkcji przystosowania, a zwłaszcza szybkości jej zmian, gdzyż przyjęcie zbyt małej dokładności kodowania przy dużej zmienności $\hat{f}$ może spowodować, że pewne jej właściwości staną się po zakodowaniu niedostępne.

Definicja 3.1 *Niech $X, \mathcal{B}$ będą jak wyżej, oraz niech $|X| = |\mathcal{B}|$. Definiujemy* **funkcję kodującą** Ψ *jako dowolną bijekcję*

$$\Psi : X \to \mathcal{B}$$

przyporządkowującą każdemu elementowi $x \in X$ chromosom $\Psi(x) \in \mathcal{B}$. □

Warunek $|X| = |\mathcal{B}|$ jest konieczny, aby Ψ mogła być bijekcją. Jeśli nie jest spełniony, to możemy doprowadzić do jego spełnienia na trzy sposoby:

1. jeżeli $|X| > |\mathcal{B}|$, to możemy zamiast całego X wziąć jego podzbiór o odpowiedniej liczebności, albo
2. jeżeli $|X| < |\mathcal{B}|$, to możemy sztucznie zwiększyć liczbę elementów X poprzez powielenie niektórych, albo
3. możemy wybrać odpowiednią relację równoważności R, jeżeli oryginalna przestrzeń poszukiwań jest ciągła (tzn. mocy continuum).

Przykład 3.1 *(por. [37]) Chcemy maksymalizować funkcję k zmiennych $f(x_1, \ldots, x_k) : \mathbb{R}^k \to \mathbb{R}$, gdzie zmienna x_i może przyjmować wartości z przedziału $D_i = \langle a_i, b_i \rangle$. Żądamy, aby każda zmienna była kodowana z dokładnością 6 cyfr znaczących, co oznacza, że każdy przedział D_i należy podzielić przynajmniej na $(b_i - a_i) \cdot 10^6$ równych podprzedziałow. Musimy teraz znaleźć najmniejszą liczbę całkowitą n_i taką, że $(b_i - a_i) \cdot 10^6 \leq 2^{n_i} - 1$. Tak zdefiniowana liczba n_i jest minimalną liczbą bitów, na których należy zakodować zmienną x_i, aby uzyskać założoną dokładność. Dziesiętną reprezentację kodu odpowiadającego x_i obliczamy następująco:*

$$b_i = round\left(\frac{x_i - a_i}{b_i - a_i}(2^{n_i} - 1)\right).$$

W ten sposób kodujemy elementy przestrzeń poszukiwań za pomocą chromosomów w postaci

$$\beta_1 \beta_2 \ldots \beta_k,$$

gdzie $\beta_i \in \{0,1\}^{n_i}$, o łącznej długości

$$N = \sum_{i=1}^{k} n_i,$$

przy czym kolejne fragmenty chromosomu β_i odpowiadają kolejnym zmiennym x_i. □

Wniosek 3.1 *Funkcja przystosowania $\hat{f}$ może być przedstawiona jako złożenie*

$$\hat{f} = f \circ \Psi, \tag{3.2}$$

gdzie $f : \mathcal{B} \to \mathbb{R}$ jest odwzorowaniem przypisującym każdemu chromosomowi wartość przystosowania odpowiadającego mu elementu X. Stąd mamy że:

$$f = \hat{f} \circ \Psi^{-1}, \tag{3.3}$$

co oznacza, że w funkcji f występuje ukryta zależność od sposobu kodowania. □

W dalszej części niniejszej pracy będziemy zakładali, że pojęcie "funkcja przystosowania" oznacza pewną funkcję f typu $\mathcal{B} \to \mathbb{R}$, którą będziemy wykorzystywali do obliczania przystosowania chromosomu. Dzięki temu będziemy mogli uogólnić nasze rozważania uniezależniając się od postaci zbioru X. Umożliwi to nam również uniezależnienie sposobu działania algorytmu genetycznego od przyjętej funkcji kodującej poprzez "ukrycie" sposobu kodowania w funkcji f.

Definicja 3.2 *Zbiór wszystkich odwzorowań typu $\mathcal{B} \to \mathbb{R}$, gdzie $\mathcal{B}$ jest zbiorem N-bitowych chromosomów dla pewnego ustalonego N, nazwiemy* **rodziną funkcji przystosowania** *i oznaczymy symbolem $\mathcal{F}$.* □

W praktyce często działamy na pewnych podzbiorach zbioru $\mathcal{B}$. Dlatego podamy teraz definicję podstawowego pojęcia związanego z funkcją przystosowania i dotyczącego podzbiorów $\mathcal{B}$.

Definicja 3.3 *Niech $f \in \mathcal{F}$ będzie funkcją przystosowania.* **Przystosowaniem podzbioru chromosomów** *$B \subseteq \mathcal{B}$ nazywamy wartość $f(B)$ zdefiniowaną następująco:*

$$f(B) = \frac{1}{|B|} \sum_{\underline{b} \in B} f(\underline{b}) \tag{3.4}$$

W przypadku, gdy $B = \mathcal{B}$, wartość $\overline{f} = f(\mathcal{B})$ będziemy nazywali **średnim przystosowaniem.** □

Aby uniknąć zwiększania liczby oznaczeń, będziemy oznaczali za pomocą funkcji f zarówno przystosowanie podzbioru chromosomów, jak i przystosowanie chromosomu. Ponieważ w obu przypadkach dziedzina jest różna, więc nie będzie to prowadziło do niejednoznaczności.

3.2 Właściwości populacji

Algorytm genetyczny dokonuje optymalizacji, działając na populacjach rozwiązań w ten sposób, że w kroku k tworzy następną, k-tą populacje przekształcając populację $k-1$. Populacja jest jednym z istotniejszych pojęć $\mathcal{BAG}$, dlatego podamy teraz jej precyzyjną definicję, oraz definicje ważniejszych pojęć towarzyszących. W przypadku ogólnym rozmiar populacji może się zmieniać w trakcie działania algorytmu. My jednak ograniczymy nasze rozważania do takiej klasy algorytmów genetycznych, w której populacja jest stała.

Definicja 3.4 *Niech $\mathcal{B}$ będzie zbiorem wszystkich łańcuchów o długości N.* **Populacją** *o rozmiarze $M \in \mathbb{N}$ nazwiemy każdy wektor $G \in \mathcal{B}^M$. Kolejne osobniki należące do populacji, czyli składowe wektora G, będziemy oznaczali przez $\underline{G}_k \in \mathcal{B}$, gdzie $k = 1, \ldots, M$ jest numerem osobnika w populacji.* □

Definicja 3.5

1. **Zbiorem osobników populacji** *$G \in \mathcal{B}^M$ nazwiemy minimalny podzbior $\mathcal{B}$, oznaczany przez $set(G)$, zawierający wszystkie osobniki należące do populacji, czyli*

$$set(G) = \left\{ \underline{b} \in \mathcal{B} : \bigvee_{k \in \langle 1, M \rangle} \underline{G}_k = \underline{b} \right\}. \tag{3.5}$$

2. **Liczbą kopii łańcucha** *$\underline{b} \in \mathcal{B}$ w populacji $G \in \mathcal{B}^M$ nazwiemy liczbę:*

$$m(\underline{b}, G) = |\{k \in \langle 1, M \rangle : \underline{G}_k = \underline{b}\}|. \tag{3.6}$$

3. **Przystosowaniem populacji** *$G \in \mathcal{B}^M$ nazwiemy liczbę:*

$$f(G) = \frac{1}{M} \sum_{k=1}^{M} f(\underline{G}_k). \tag{3.7}$$

□

Podobnie jak w przypadku przystosowania podzbioru chromosomów, w celu uniknięcia zwiększania liczby oznaczeń, przystosowanie populacji będziemy oznaczaąć również przez funkcję f.

Wniosek 3.2 *Na podstawie punktów 1 i 2 Definicji 3.5 można udowodnić prawdziwość zależności:*

$$\sum_{\underline{b}\in set(G)} m(\underline{b}, G) = \sum_{\underline{b}\in\mathcal{B}} m(\underline{b}, G) = M.$$

□

Z punktu widzenia działania $\mathcal{BAG}$, jak również oceny przystosowania populacji, nie jest istotna kolejność ustawienia osobników w populacji, a jedynie liczba ich kopii. Dlatego wprowadzimy teraz pojęcie równoważności populcji.

Definicja 3.6 *Mówimy, że populacje $G_1, G_2 \in \mathcal{B}^M$ są* **równoważne** *jeżeli jest spełniony warunek*

$$\bigwedge_{\underline{b}\in\mathcal{B}} m(\underline{b}, G_1) = m(\underline{b}, G_2). \tag{3.8}$$

Spełnienie powyższego warunku oznaczamy przez $G_1 \cong G_2$. □

Można łatwo zauważyć, że warunek $set(G_1) = set(G_2)$ jest warunkiem koniecznym, ale nie wystarczającym, aby zachodziło $G_1 \cong G_2$.

3.3 Własności schematów

3.3.1 Definicje

Podstawowym pojęciem w teorii algorytmów genetycznych jest pojęcie schematu wprowadzone przez Hollanda w pracy [5]. Schematy reprezentują podprzestrzenie przestrzeni kodowej $\mathcal{B}$ zawierające podobne do siebie chromosomy, to znaczy takie, które na pewnych pozycjach mają bity o tych samych wartościach.

Najpowszechniej spotykana interpretacja schematów mówi, że schematy są opisami idei albo koncepcji rozwiązań problemu, natomiast podstawowym zadaniem algorytmu genetycznego jest przetwarzanie tych koncepcji. Polega ono na tym, że lepsze koncepcje – schematy zyskują więcej reprezentujących je chromosomów, a gorsze – mniej. Zdefiniujemy teraz pojęcie schematu, oraz inne związane z nim, najczęściej spotykane pojęcia.

Definicja 3.7 *Niech $\mathcal{B} = \{0,1\}^N$ będzie zbiorem N-bitowych chromosomów.* **Schematem** *nazywamy dowolny ciąg $\underline{s} = (s_1, \ldots, s_N)$ należący do zbioru $\mathcal{S} = \{0,1,*\}^N$. Każdy schemat $\underline{s}$ jednoznacznie definiuje podprzestrzeń przestrzeni $\mathcal{B}$, określoną następującym wzorem:*

$$\underline{s} = \left\{ \underline{b} \in \mathcal{B} : \bigwedge_{i\in\{1,\ldots,N\}} b_i = \begin{cases} s_i, & dla\ s_i \neq * \\ 0\ lub\ 1, & dla\ s_i = * \end{cases} \right\}, \tag{3.9}$$

gdzie '' (gwiazdka) jest symbolem uniwersalnym, a 0 i 1 oznaczają, że pozycje są ustalone. Chromosomy należące do tej podprzestrzeni nazywamy* **reprezentantami schematu** $\underline{s}$. □

Z powyższej definicji wynika, że każdy schemat jest podzbiorem przestrzeni $\mathcal{B}$, ale nie każdy podzbiór $\mathcal{B}$ jest schematem. Można także łatwo wywnioskować, że każdy chromosom jest reprezentantem 2^N schematów, które możemy otrzymać zastępując kolejne bity chromosomu symbolem uniwersalnym.

Schematy będziemy oznaczali podobnie jak chromosomy – za pomocą ciągów zer, jedynek i gwiazdek, bez oddzielania składowych przecinkami, jak na przykład dla $N = 16$:

$$\underline{s} = 011***1010100*11* \in \mathcal{S}.$$

Definicja 3.8

1. Przystosowaniem schematu $\underline{s}$ *nazywamy wartość* $f(\underline{s})$ *zdefiniowaną jako*

$$f(\underline{s}) = \frac{1}{|\underline{s}|} \sum_{\underline{b} \in \underline{s}} f(\underline{b}). \tag{3.10}$$

2. Liczbą reprezentantów schematu $\underline{s}$ w podzbiorze $B \subseteq \mathcal{B}$ *nazywamy wartość* $m(\underline{s}, B)$ *zdefiniowaną jako*

$$m(\underline{s}, B) = |B \cap \underline{s}|. \tag{3.11}$$

3. Przystosowaniem schematu $\underline{s}$ w podzbiorze $B \subseteq \mathcal{B}$ *nazywamy wartość* $f(\underline{s}, B)$ *zdefiniowaną jako*

$$f(\underline{s}, B) = \frac{1}{m(\underline{s}, B)} \sum_{\underline{b} \in B \cap \underline{s}} f(\underline{b}), \tag{3.12}$$

dla $m(\underline{s}, B) \neq 0$.

4. Liczbą reprezentantów schematu $\underline{s}$ w populacji $G \in \mathcal{B}^M$ *nazywamy wartość* $m(\underline{s}, G)$ *zdefiniowaną jako*

$$m(\underline{s}, G) = \sum_{\underline{b} \in \underline{s}} m(\underline{b}, G) \tag{3.13}$$

5. Przystosowaniem schematu $\underline{s}$ w populacji $G \in \mathcal{B}^M$ *nazywamy wartość* $f(\underline{s}, G)$ *zdefiniowaną jako*

$$f(\underline{s}, G) = \frac{1}{m(\underline{s}, G)} \sum_{\underline{b} \in \mathcal{B}} m(\underline{b}, G)\, f(\underline{b}) \tag{3.14}$$

dla $m(\underline{s}, G) \neq 0$.

6. Rzędem schematu $\underline{s}$, *który oznaczamy przez* $o(\underline{s})$, *nazywamy liczbę ustalonych pozycji w* $\underline{s}$.

7. Rozpiętością schamatu $\underline{s}$, *którą oznaczamy przez* $\delta(\underline{s})$, *nazywamy odległość między dwiema skrajnymi pozycjami ustalonymi.*

□

Podobnie jak to zrobiliśmy wcześniej, rozszerzamy tutaj użycie oznaczeń m i f wykorzystując je do oznaczania pojęć pokrewnych znaczeniowo pojęciom zdefiniowanym uprzednio (por. Def. 3.5).

3.3.2 Przetwarzanie schematów w $\mathcal{BAG}$

Elementem $\mathcal{BAG}$ bezpośrednio wpływającym na schematy reprezentowane w populacji są operatory genetyczne, np. mutacja i krzyżowanie, które zostaną szczegółowo omówione w Podrozdziale 3.4.2. Operatory te, przetwarzając bezpośrednio chromosomy, przetwarzają pośrednio także schematy. Dokonują tego zmieniając liczbę reprezentantów poszczególnych schematów w kolejnych populacjach. Z drugiej strony, każda modyfikacja chromosomu powoduje zmianę zbioru schematów, których jest on reprezentantem. Jeżeli w wyniku modyfikacji chromosom przestał być reprezentantem pewnego schematu, to mówimy o takim schemacie, że został on zniszczony, w przypadku przeciwnym zostaje on zachowany.

Prześledźmy w skrócie wpływ najpopularniejszych operatorów genetycznych na schematy:

1. Standardowa mutacja powoduje zniszczenie schematu, jeśli w wyniku jej działania zostaje zmieniona pozycja ustalona w schemacie.

2. Krzyżowanie jednopunktowe zawsze zachowuje schemat tylko wtedy, gdy pozycja krzyżowania nie rozdziela pozycji ustalonych. Jeśli pozycje ustalone są rozdzielone, to zachowanie schematu zależy od drugiego z krzyżowanych chromosomów. W tym przypadku im schemat jest węższy tym mniejsze jest prawdopodobieństwo jego zniszczenia.

Korzystając ze zdefiniowanych uprzednio pojęć, wyprowadza się wzór na liczbę reprezentantów danego schematu w kolejnych populacjach, zwanych inaczej pokoleniami, tworzonych przez algorytm genetyczny (dokładne omówienie i wyprowadzenie tego wzoru dla krzyżowania jednopunktowego można znaleźć w książkach Michalewicza [37] i Goldberga [28]):

$$\mathbf{E}\left(m(\underline{s}, G_{k+1})\right) \geq m(\underline{s}, G_k)\frac{f(\underline{s}, G_k)}{f(G_k)}\left(1 - p_c\frac{\delta(\underline{s})}{N-1} - o(\underline{s})\,p_m\right) \tag{3.15}$$

gdzie $\mathbf{E}(\cdot)$ jest wartością oczekiwaną, $k = 1, 2, \ldots$ jest kolejnym numerem populacji, natomiast $G_k \in \mathcal{B}^M$ jest populacją stanowiącą k-te pokolenie w $\mathcal{BAG}$. Na podstawie wzoru (3.15) dowodzi się Twierdzenie o schematach zwane też Podstawowym twierdzeniem algorytmów genetycznych:

Twierdzenie 3.1 (o schematach) *Wąskie (o małej rozpiętości) niskiego rzędu o przystosowaniu wyższym niż średnie schematy uzyskują w kolejnych pokoleniach wykładniczo rosnącą liczbę reprezentantów.* □

Należy zwrócić uwagę, że twierdzenie oparte jest na analizie działania krzyżowania jednopunktowego i standardowej mutacji – jest prawdziwe jedynie dla tych operatorów. Przytaczamy je ze względów historycznych, nadmieniając, że podobne twierdzenia zostały opracowane także dla innych operatorów (por. prace Spearsa i DeJonga [18] i [21]).

3.3.3 Hipoteza cegiełek

Hipoteza 3.1 (cegiełek lub bloków budujących) *Algorytm genetyczny poszukuje chromosomu optymalnego $\underline{b}_{opt}$ przez zestawianie wąskich, niskiego rzędu schematów o wyższym niż średnie przystosowaniu, zwanych cegiełkami lub blokami budującymi.* □

Hipoteza powyższa, dawniej przyjmowana z dużym optymizmem, w ostatnim okresie wielokrotnie była podważana i krytykowana. Jako przykład sytuacji, w której hipoteza cegiełek nie działa, wprowadza się pojęcie zwodniczości funkcji przystosowania (lub problemu). Przyjmuje się, że problemem zwodniczym jest taki problem, dla którego algorytm genetyczny może zbiegać do rozwiązania suboptymalnego, ponieważ istnieją schematy o wysokim przystosowaniu, które jednakże nie zawierają rozwiązania optymalnego. Aby ocenić pod tym względem dany problem, definiuje się warunek zwodniczości.

Definicja 3.9 *Mówimy, że funkcja przystosowania $f \in \mathcal{F}$ jest zwodnicza, jeśli spełniony jest warunek:*

$$\bigvee_{\underline{s}_1, \underline{s}_2 \in \mathcal{S}} \underline{b}_{opt} \notin \underline{s}_1 \wedge \underline{b}_{opt} \in \underline{s}_2 \wedge f(\underline{s}_1) > f(\underline{s}_2), \tag{3.16}$$

gdzie $\underline{b}_{opt} \in \mathcal{B}$ jest globalnym optimum funkcji f. □

Spełnienie powyższego warunku oznacza, że rozwiązanie optymalne $\underline{b}_{opt}$ jest izolowane, czyli otoczone rozwiązaniami słabymi, należącymi do schematu $\underline{s}_2$, co może błędnie skierować algorytm genetyczny do rozwiązań suboptymalnych, należących do $\underline{s}_1$.

Przykład 3.2 *Minimalny problem zwodniczy (ang. minimal deceptive problem) (patrz [28] str. 63) można łatwo zdefiniować na dwóch bitach. Załóżmy, że dla $N = 6$ mamy $\underline{b}_{opt} = 111111$ oraz, że mamy dane przystosowania schematów rzędu 2 zgodnie z poniższą tabelką.*

$\underline{s}$	$f(\underline{s})$
$*0*0**$	3
$*0*1**$	5
$*1*0**$	0
$*1*1**$	6

Stąd możemy obliczyć, że $f(*0****) = 4$ *oraz* $f(*1****) = 3$, *czyli spełniony jest warunek zwodniczości zgodnie z Definicją 3.9, gdyż* $\underline{b}_{opt} \in *1****$. □

Zwodniczość jest ściśle związana z pojęciem **epistazy**, oznaczającym silne uzależnienie między bitami będącymi na różnych pozycjach w chromosomie. Przejawia się ono w tym, że wpływ zmiany danego bitu na wartość funkcji przystosowania jest silnie zależny od wartości innych bitów w chromosomie. Stąd widać, że w przypadku silnej epistazy nie możemy traktować fragmentów chromosomu jak cegiełek do budowania rozwiązania optymalnego.

3.4 Operatory genetyczne

3.4.1 Rodzaje operatorów genetycznych

Zgodnie z Rys. 2.2.1 w każdym wykonaniu głównej pętli $\mathcal{BAG}$ następuje zbudowanie populacji aktualnej G_i na podstawie populacji poprzednej G_{i-1}. Czynność ta składa się z kilku faz. W każdej fazie następuje przetwarzanie łańcuchów populacji poprzedniej przy wykorzystaniu właściwych dla danej fazy algorytmów, których istotną cechą jest wykonywanie pewnych prostych operacji bezpośrednio na łańcuchach należących do populacji. Przyjmijmy, że operacje te są wykonywane przez odwzorowania nazywane **operatorami genetycznymi**. Istotną cechą operatorów genetycznych jest to, że są one zwykle operatorami losowymi, to znaczy ich wynik działania jest zależny nie tylko od ich argumentów, ale także od wartości realizacji pewnej zmiennej losowej.

Ze względu na dziedzinę możemy operatory genetyczne podzielić na dwie grupy:

- operatory działające na osobnikach z populacji, np. operatory mutacji i krzyżowania;
- operatory działające na całych populacjach, np. operator selekcji metodą ruletki.

Ze względu na zależność działania operatora od przystosowania osobników, możemy operatory podzielić na dwie grupy:

- niezależne od przystosowania, np. operatory mutacji i krzyżowania;
- zależne od przystosowania, np. operator selekcja metodą ruletki.

Omówimy teraz bardziej szczegółowo grupę operatorów należących do pierwszych pozycji w obu powyższych podziałach, czyli działających na osobnikach i niezależnych od przystosowania. Operatory należące do tej grupy będziemy nazywali **operatorami przemieszczenia**.

3.4.2 Wprowadzenie operatora przemieszczenia

Definicja 3.10 *Niech trójka uporządkowana* $(\Omega, \mathcal{C}, P)$ *będzie przestrzenią probabilistyczną, gdzie* Ω *jest skończoną przestrzenią zdarzeń elementarnych.* **Operatorem przemieszczenia** *nazywamy dowolne odwzorowanie postaci:*

$$\phi : \mathcal{B}^d \times \Theta \to \mathcal{B}^d, \tag{3.17}$$

gdzie Θ *jest jest zbiorem rozłącznych podzbiorów przestrzeni* Ω *takich, że* $\bigcup(\theta \in \Theta) = \Omega$ *oraz liczba naturalna* $d > 0$ *jest wymiarem operatora. Drugi argument operatora* ϕ *będziemy nazywali* **argumentem losowym**. □

Operator przemieszczenia może być jednoznacznie scharakteryzowany przez warunkowy dyskretny rozkład prawdopodobieństwa w postaci:

$$\mathbf{P}\left(\phi(lB,\theta)|B\right), \tag{3.18}$$

gdzie $\theta \in \Theta$, przy czym $B \in \mathcal{B}^d$ oraz $\phi(B,\theta) \in \mathcal{B}^d$ są wektorami łańcuchów w postaci $(\underline{b}_1,\ldots,\underline{b}_d)$, gdzie $\underline{b}_i \in \mathcal{B}$ dla $i = 1,\ldots,d$. Rozkład ten może być przedstawiony w postaci macierzy kwadratowej o wymiarze $|\mathcal{B}|^d$.

W niniejszej pracy będziemy często używali uproszczonego zapisu operatorów przemieszczenia, $\phi : \mathcal{B}^d \to \mathcal{B}^d$, rezygnując z jawnego uwzględniania argumentu losowego. Zwykle właściwości tego argumentu, czyli definicja zbioru Θ oraz rozkład prawdopodobieństwa jego elementów, wynikają jednoznacznie z definicji danego operatora, stąd takie uproszczenie nie prowadzi do niejednoznaczności. Należy jednak mieć na uwadze, że właściwości tego argumentu mają fundamentalny wpływ na działanie operatora.

Przykład 3.3 *Standardowy operator mutacji jest przykładem operatora przemieszczenia dla* $d = 1$. *Możemy przyjąć, że zbiór wartości argumentu losowego jest łańcuchem binarnym o długości* N, *czyli* $\Theta = \mathcal{B}$. *Każdy z bitów przyjmuje wartość* 1 *z prawdopodobieństwem* p_m *i wartość* 0 *– z prawdopodobieństwem* $1 - p_m$. *Wartość* 1 *oznacza, że odpowiedni bit mutowanego łańcucha będzie zmieniony na przeciwny. Fragment rozkładu prawdopodobieństwa dla łańcucha 3-bitowego podano w Przykładzie 3.6.* □

Przykład 3.4 *Operator krzyżowania jednopunktowego jest przykładem operatora przemieszczenia dla* $d = 2$. *Wartość argumentu losowego w tym przypadku określa pozycję krzyżowania, jego zbiór wartości jest podzbiorem liczb naturalnych:* $\Theta = \{1,\ldots,N\}$, *a jego rozkład prawdopodobieństwa jest jednorodny.* □

3.4.3 Opis właściwości operatorów przemieszczenia przy pomocy relacji sąsiedztwa

Jak wspomniano wcześniej, każdy problem rozwiązywany przez $\mathcal{BAG}$ jest jednoznacznie definiowany przez jego funkcję przystosowania $f \in \mathcal{F}$ (patrz Definicja 3.2). W pewnych sytuacjach wygodne jest zdefiniowanie dodatkowo topologii w dziedzinie funkcji przystosowania, czyli w zbiorze $\mathcal{B}$, poprzez wprowadzenie pojęcia **relacji sąsiedztwa** oraz pojęcia **sąsiedztwa**.

Definicja 3.11 *Niech* X *będzie dowolnym zbiorem skończonym.* **Relacją sąsiedztwa** *w* X *nazywamy dowolny podzbiór* $\mathcal{N} \subset X \times X$. $\mathcal{N}$-**Sąsiedztwem** *dowolnego elementu* $x_0 \in X$ *nazywamy zbiór*

$$\mathcal{N}(x_0) = \{x : (x_0, x) \in \mathcal{N}\}. \tag{3.19}$$

□

Każdy operator przemieszczenia definiuje jednoznacznie pewną relację sąsiedztwa w $\mathcal{B}$. Dla dowolnego operatora przemieszczenia $\phi : \mathcal{B} \to \mathcal{B}$ możemy zdefiniować relację sąsiedztwa $\mathcal{N}$ następująco:

$$\mathcal{N} = \{(\underline{b}_1, \underline{b}_2) : \mathbf{P}\left(\phi(\underline{b}_1) = \underline{b}_2\right) > 0\}. \tag{3.20}$$

Przykład 3.5 *Rozważmy standardowy operator mutacji o prawdopodobieństwie* p_m. *W tym przypadku zapis* $\mathcal{N}(\underline{b})$ *oznacza zbiór wszystkich łańcuchów które możemy uzyskać w wyniku zmutowania łańcucha* $\underline{b}$. *Nietrudno zauważyć, że jeżeli* $p_m > 0$ *to otrzymujemy że* $\mathcal{N}(\underline{b}) = \mathcal{B}$ *dla każdego* $\underline{b} \in \mathcal{B}$, *ponieważ nie ma ograniczenia na liczbę mutowanych bitów. Otrzymana w ten sposób relacja* $\mathcal{N} = \mathcal{B} \times \mathcal{B}$ *jest relacją symetryczną.* □

W $\mathcal{BAG}$ stosuje się głównie takie operatory przemieszczenia, których istotną właściwością jest, że w wyniku ich działania możemy otrzymać z różnym prawdopodobieństwem każdy łańcuch z $\mathcal{B}$, jak w powyższym przykładzie. Dlatego, aby pełniej opisać właściwości danego operatora, wprowadza się pojęcie **rozmytej relacji sąsiedztwa** oraz **sąsiedztwa rozmytego.**

Definicja 3.12 *Niech X będzie dowolnym zbiorem skończonym.* **Rozmyta relacja sąsiedztwa** *w X jest definiowana przez dowolne odwzorowanie $\mu : X \times X \to \langle 0,1\rangle$ nazywane* **funkcją przynależności.** *Natomiast* **α-sąsiedztwem** *$\mathcal{N}_\alpha(x)$ dowolnego elementu $x_0 \in X$, dla $\alpha \in \langle 0,1\rangle$, nazywamy zbiór*

$$\mathcal{N}_\alpha(x_0) = \{x : \mu(x_0, x) \geq \alpha\}. \tag{3.21}$$

□

Pojęcie rozmytej relacji sąsiedztwa jest zdefiniowane zgodnie z teorią zbiorów rozmytych, której dokładny opis może czytelnik znaleźć na przykład w skrypcie Czogały i Pedrycza [11]. Tam też można znaleźć omówienie związków między klasycznym pojęciem zbioru a pojęciem zbioru rozmytego.

Podobnie jak w przypadku relacji sąsiedztwa, każdy operator przemieszczenie także definiuje jednoznacznie pewną rozmytą relację sąsiedztwa w $\mathcal{B}$. Zwykle wartość funkcji przynależności $\mu(\underline{b}_1, \underline{b}_2)$ dla dowolnych łańcuchów $\underline{b}_1, \underline{b}_2 \in \mathcal{B}$ interpretuje się jako stopień "powiązania" istniejący między tymi łańcuchami. Dlatego przyjmiemy, że wartości funkcji μ będą wyznaczane z warunkowego dyskretnego rozkładu prawdopodobieństwa:

$$\bigwedge_{\underline{b}_1, \underline{b}_2 \in \mathcal{B}} \mu(\underline{b}_1, \underline{b}_2) = \mathbf{P}(\phi(\underline{b}) = \underline{b}_2 | \underline{b} = \underline{b}_1) \tag{3.22}$$

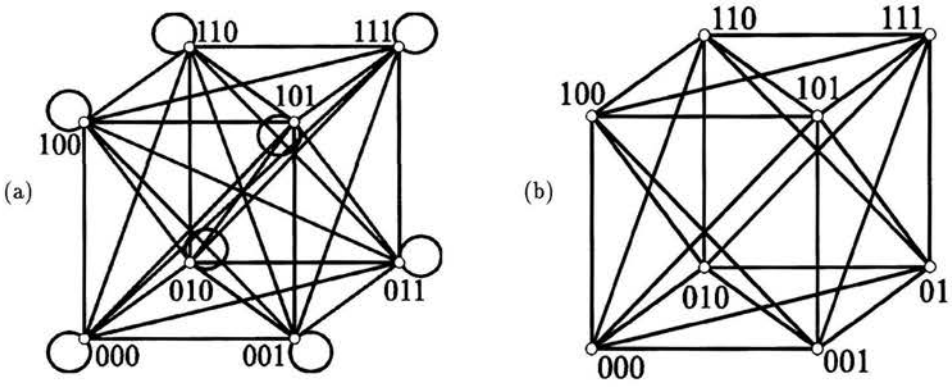
gdzie ϕ jest operatorem przemieszczenia oraz $\underline{b} \in \mathcal{B}$.

Przykład 3.6 *Załóżmy, że $\mathcal{B}$ jest przestrzenią łańcuchów o długości $N = 3$. Przyjmiemy, że rozmytą relację sąsiedztwa tworzymy za pomocą standardowej mutacji o prawdopodobieństwie $p_m = 0.1$. Wtedy przykładowo dla pierwszego argumentu funkcji przynależności μ równego* 000 *otrzymamy następujące wartości:*

$\underline{b}$	$\mu(000, \underline{b})$
000	$0.9^3 = 0.729$
001	$0.1 \cdot 0.9^2 = 0.081$
010	$0.1 \cdot 0.9^2 = 0.081$
011	$0.1^2 \cdot 0.9 = 0.009$
100	$0.1 \cdot 0.9^2 = 0.081$
101	$0.1^2 \cdot 0.9 = 0.009$
110	$0.1^2 \cdot 0.9 = 0.009$
111	$0.1^3 = 0.001$

□

W podobny sposób możemy wyznaczyć rozmytą relacją sąsiedztwa dla operatorów przemieszczenia, w których $d > 1$, np. dla krzyżowania ($d = 2$). Najpierw wyznaczamy d rozkładów warunkowych $\mathbf{P}(\phi_k(B, \theta)|\underline{b}_k)$ dla kolejnych składowych wartości operatora, gdzie $B = (\underline{b}_1, \ldots, \underline{b}_k, \ldots, \underline{b}_d)$. Następnie otrzymane rozkłady uśredniamy i stąd dla każdej pary $(\underline{b}_1, \underline{b}_2)$ wyznaczamy wartość funkcji przynależności μ.



Rys. 3.1: Grafy odpowiadające przestrzeni chromosomów 3-bitowych z różnymi operatorami sąsiedztwa: (a) standardowy operator mutacji, (b) operator zakłócenia o szerokości zakłócenia $w = 2$, który zostanie zdefiniowany w Def. 7.4. Można zauważyć, że operator sąsiedztwa w przypadku (a) jest odwzorowaniem zwrotnym, a w (b) – przeciwzwrotnym.

3.5 Opis krajobrazów przystosowania za pomocą grafów

3.5.1 Idea krajobrazu przystosowania

Pojęcie **krajobrazu przystosowania** *(ang. fitness landscape)* łączy w sobie definicję problemu, czyli funkcję przystosowania wraz z topologią przestrzeni kodowej. Różni autorzy w różny sposób definiują to pojęcie, często włączając do niego jeszcze dodatkowe elementy (por. [34]). W niniejszej rozprawie będziemy oznaczali krajobraz przystosowania jako parę uporządkowaną

$$L = (f, \mathcal{N}), \tag{3.23}$$

gdzie $f \in \mathcal{F}$ jest funkcją przystosowania, a $\mathcal{N}$ – relacją sąsiedztwa zwykłego lub rozmytego.

Najczęściej spotykanym w literaturze sposobem opisu sąsiedztwa, używanym zamiast relacji sąsiedztwa $\mathcal{N}$, jest graf (patrz [30], [38], [34] i inne). Zwykle przyjmuje się, że elementy przestrzeni kodowej (chromosomy w $\mathcal{BAG}$) są wierzchołkami grafu, a istnienie lub nieistnienie krawędzi między dwoma wierzchołkami albo waga przypisana tej krawędzi określa relację sąsiedztwa między tymi wierzchołkami. Podejście to zostanie omówione poniżej.

3.5.2 Grafy zwykłe

Podstawy teorii grafów może czytelnik znaleźć w bardzo wielu publikacjach. Szczególnie przystępnie została ona przedstawiona w książce [9] – z niej zaczerpnięto większość poniższych definicji. Natomiast w pracy Stadlera [30] opisano obszernie te elementy teorii grafów, które są szczególnie wykorzystywane w badaniach krajobrazów przystosowania.

Definicja 3.13 Grafem nieskierowanym *nazywamy parę* $G = (V, E)$ *złożoną ze zbioru* wierzchołków $V = \{v_1, v_2, \ldots\}$ *oraz zbioru* krawędzi $E = \{e_1, e_2, \ldots\}$. *Krawędź* e_k *utożsamia się z nieuporządkowaną parą wierzchołków* $\{v_i, v_j\}$. *Wierzchołki* v_i, v_j *nazywamy wierzchołkami końcowymi krawędzi* e_k *i mówimy, że krawędź* e_k *jest* incydentna do *i-tego oraz j-tego wierzchołka.* □

Definicja 3.14 Grafem skierowanym *nazywamy parę* $G = (V, E)$ *złożoną ze zbioru* wierzchołków $V = \{v_1, v_2, \ldots\}$ *oraz zbioru* krawędzi skierowanych $E = \{e_1, e_2, \ldots\}$. *Krawędź skierowaną* e_k *utożsamia się z uporządkowaną parą wierzchołków* (v_i, v_j). *Wierzchołki* v_i, v_j *nazywamy odpowiednio,*

wierzchołkiem początkowym i końcowym krawędzi e_k, oraz mówimy, że krawędź e_k jest **incydentna z** *wierzchołka v_i oraz* **incydentna w** *wierzchołek v_j.* □

Zauważmy, że z każdego grafu skierowanego G_s możemy utworzyć graf nieskierowany, pomijając orientację krawędzi. Taki graf będziemy nazywali **grafem nieskierowanym odpowiadającym** G_s. Podobnie jeżeli przypiszemy pewną orientację krawędziom grafu nieskierowanego G_{ns}, to otrzymamy graf skierowany. Graf taki będziemy nazywali **grafem skierowanym związanym z** G_{ns}. Jest oczywiste, że każdemu grafowi nieskierowanemu odpowiada wiele grafów skierowanych.

Definicja 3.15 **Pętlą własną** *grafu nazywamy dowolną krawędź postaci $\{v, v\}$ - dla grafu nieskierowanego lub (v, v) - dla grafu skierowanego, gdzie $v \in V$.* □

Definicja 3.16 **Grafem pełnym** *nieskierowanym (lub skierowanym) nazwiemy taki graf, w którym istnieje krawędź między każdą parą wierzchołków, czyli*

$$\bigwedge_{v,w \in V} \{v, w\} \in E \quad (lub\ (v, w) \in E). \tag{3.24}$$

□

W przypadku grafów pełnych, liczba krawędzi jest związana z liczbą wierzchołków następującymi zależnościami:

1. dla grafu skierowanego (ze wzoru na wariacje z powtórzeniami $\bar{C}_n^k = n^k$):

$$|E| = |V|^2, \tag{3.25}$$

2. dla grafu nieskierowanego (ze wzoru na kombinacje z powtórzeniami $C_n^k = \binom{n+k-1}{k}$):

$$|E| = \frac{|V|\,(|V|+1)}{2}. \tag{3.26}$$

Przykład 3.7 *Weźmy standardowy operator mutacji jak w Przykładzie 3.5, jako operator przemieszczenia tworzący krajobraz przystosowania $L = (f, \mathcal{N})$, gdzie $f \in \mathcal{F}$, a $\mathcal{N}$ jest relacją symetryczną. Aby przedstawić L za pomocą grafu nieskierowanego przyjmiemy, że zbiór wierzchołków $V = \mathcal{B}$, a zbiór nieskierowanych krawędzi*

$$E = \{\{v, w\} : (v, w)\ lub\ (w, v) \in \mathcal{N}\}, \tag{3.27}$$

Na przykład dla $N = 3$ otrzymamy:

$$\begin{aligned} E = \{\ & \{000,000\}, \ldots, \{000,111\}, \{001,001\}, \ldots, \{001,111\}, \\ & \{010,010\}, \ldots, \{010,111\}, \{011,011\}, \ldots, \{011,111\}, \\ & \{100,100\}, \ldots, \{100,111\}, \{101,101\}, \ldots, \{101,111\}, \\ & \{110,110\}, \{110,111\}, \{111,111\}\ \}, \end{aligned} \tag{3.28}$$

przy czym moc $|E| = 36$ (ze wzoru (3.26)). W ten sposób zdefiniowany graf przedstawiono na Rys. 3.1a. □

Przykład 3.8 *Weźmy graf nieskierowany, odpowiadający standardowemu operatorowi mutacji, opisany w Przykładzie 3.7. Możemy utworzyć odpowiadający mu graf skierowany zastępując każdą krawędź nieskierowaną $\{v, w\}$ (dla $v \neq w$) parą krawędzi skierowanych (v, w) i (w, v), oraz każdą pętlę własną nieskierowaną $\{v, v\}$ - przez pętlę skierowaną (v, v).* □

Definicja 3.17 *Niech G będzie grafem nieskierowanym (skierowanym) o n wierzchołkach.* Macierzą przyległości *grafu G nazwiemy macierz A o wymiarach $n \times n$ o elementach*

$$A_{ij} = \begin{cases} 1, & \text{gdy istnieje krawędź } \{v_i, v_j\} \text{ (krawędź skierowana } (v_i, v_j)); \\ 0, & \text{wpp.} \end{cases} \tag{3.29}$$

□

Przykład 3.9 *Macierzą przyległości dla obu grafów opisanych w Przykładach 3.7 i 3.8 będzie macierz o wymiarach 8×8 o wszystkich elementach równych 1.* □

Przykład 3.10 *Rozważmy graf dla relacji sąsiedztwa określonej przez operator zakłócenia zdefiniowany w Def. 7.4, przedstawiony na Rys. 3.1b. Widać że krawędzie występują jedynie między między takimi wierzchołkami, których odległość Hamminga jest równa 1 lub 2. Stąd otrzymamy macierz przyległości o wymiarach 8×8 o elementach:*

$$A_{ij} = \begin{cases} 1, & \text{gdy } 0 < d_H(\underline{b}_i, \underline{b}_j) < 3; \\ 0, & \text{wpp.} \end{cases} \tag{3.30}$$

Otrzymany graf odpowiada relacji symetrycznej, gdyż $d_H(\underline{b}_i, \underline{b}_j) \equiv d_H(\underline{b}_j, \underline{b}_i)$ dla wszystkich i, j. □

Każdy graf skierowany jest reprezentacją pewnej relacji $R \subset V \times V$. Natomiast każdy graf nieskierowany jest reprezentacją pewnej relacji symetrycznej $R_s \subset V \times V$. Dlatego w teorii relacji macierz przyległości jest nazywana macierzą relacji [9]. Z takiego podejścia wynika także definicja sąsiedztwa wierzchołka grafu.

Definicja 3.18 Sąsiedztwem *wierzchołka $v \in V$ nazywamy podzbiór $\mathcal{N}(v) \subset V$ zdefiniowany*

1. *dla grafu nieskierowanego jako: $\mathcal{N}(v) = \{w \in V : \{v, w\} \in E\}$,*
2. *dla grafu skierowanego jako: $\mathcal{N}(v) = \{w \in V : (v, w) \in E\}$,*

gdzie E jest zbiorem krawędzi grafu. □

Definicja 3.19 *Niech G będzie grafem nieskierowanym o n wierzchołkach.* Stopniem *i-tego wierzchołka $d(v_i)$ nazywamy liczbę incydentych do niego krawędzi, czyli*

$$d(v_i) = \sum_j A_{ij} = |\mathcal{N}(v_i)|. \tag{3.31}$$

Macierz diagonalną D o wymiarach $n \times n$ o elementach $D_{ii} = d(v_i)$ nazywamy macierzą stopni. *Graf nazywamy* regularnym, *jesli wszystkie jego wierzchołki są tego samego stopnia d_0. W tym przypadku $D_{ii} = d_0 I$, gdzie I jest macierzą jednostkową.* □

Definicja 3.20 *Niech G będzie grafem nieskierowanym bez pętli własnych o n wierzchołkach i m krawędziach.* Macierzą incydencji *grafu G nazwiemy macierz ∇ o wymiarach $n \times m$ o elementach*

$$\nabla_{ij} = \begin{cases} 1, & \text{gdy } j\text{-ta krawędź jest incydentna do } i\text{-tego wierzchołka;} \\ 0, & \text{wpp.} \end{cases} \tag{3.32}$$

□

Definicja 3.21 *Niech G będzie grafem skierowanym bez pętli własnych o n wierzchołkach i m krawędziach.* Macierzą incydencji *grafu G nazwiemy macierz ∇ o wymiarach $n \times m$ o elementach*

$$\nabla_{ij} = \begin{cases} 1, & \text{gdy } j\text{-ta krawędź jest incydentna z } i\text{-tego wierzchołka;} \\ -1, & \text{gdy } j\text{-ta krawędź jest incydentna w } i\text{-ty wierzchołek;} \\ 0, & \text{wpp.} \end{cases} \tag{3.33}$$

□

Definicja 3.22 *Niech G będzie grafem skierowanym bez pętli własnych, a ∇ – jego macierzą incydencji. Niech D i A będą odpowiednio macierzą stopni i przyległości grafu nieskierowanego odpowiadającego G. Definiujemy* **laplasjan grafu** *jako*

$$\Delta = A - D = -\nabla\nabla^T. \tag{3.34}$$

□

Dowód równości $A - D = -\nabla\nabla^T$, a także omówienie innych właściwości laplasjanu grafu można znaleźć w pracy Biggsa [23].

Wybór symboli ∇ i Δ jest nieprzypadkowy. Dla dowolnej funkcji $f : V \to \mathbb{R}$ możemy zdefiniować funkcję $\nabla f : E \to \mathbb{R}$ przez wyrażenie

$$(\nabla f)(e) = f(v) - f(w) \tag{3.35}$$

gdzie $v, w \in V$, $e \in E$ oraz $e = (v, w)$. Operacja ta odpowiada przyrostowi kierunkowemu pola skalarnego w $\mathbb{R}^n$, przy czym w przypadku grafu, kierunek jest definiowany przez krawędź e.

Podobnie przyjmuje się, że macierz Δ odpowiada laplasjanowi pola skalarnego, zdefiniowanemu w $\mathbb{R}^n$ jako $\Delta = \sum_{i=1}^{n} \frac{\partial}{\partial x_i^2}$. Dla dowolnej funkcji $f : V \to \mathbb{R}$ możemy zdefiniować funkcję $\Delta f : V \times \mathbb{R}$ przez wyrazenie

$$(\Delta f)(w) = \sum_{v \in \mathcal{N}(w)} f(v) - f(w) \tag{3.36}$$

gdzie $v, w \in V$ oraz $e = (v, w) \in E$. Rownież inne właściwości obu macierzy są podobne do właściwości odpowiednich operatorów teorii pola. Dzięki temu staje się możliwe stosowanie narzędzi teorii pola do badania właściwości grafów. Zasygnalizowane powyżej podejście zostało szczególnie rozwinięte i omówione w pracach Stadlera i in. [30], [38], [35], [41].

3.5.3 Grafy ważone

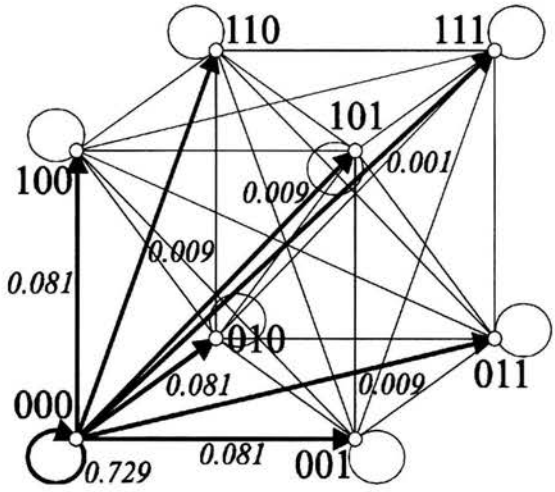
Dotychczas wprowadzone rodzaje grafów były wystarczające do reprezentacji relacji sąsiedztwa zgodnej z Definicją 3.11. W celu wygodnej reprezentacji rozmytej relacji sąsiedztwa wprowadza się rozszerzone pojęcie grafu – grafu ważonego. Zagadnienia te zostały szczegółowo zdefiniowane i omówione np. w książce [16].

Definicja 3.23 **Grafem ważonym** *lub* **siecią** $G = (V, E, \varpi)$ *nazywamy taki graf nieskierowany lub skierowany, z którego każdą krawędzią związano pewną* **funkcję wagi** $\varpi : E \to \mathbb{R}$. *Funkcja ta może być zapisywana w alternatywnej formie $\varpi_{,} : V \times V \to \mathbb{R}$ – jest ona wtedy określona tylko dla tych par wierzchołków, między którymi istnieje krawędź.* □

Funkcja wagi zwykle jest interpretowana jako "siła" powiązania między dwoma wierzchołkami, ale może także określać koszt danej krawędzi fizycznie przedstawiany jako odległość między końcowymi wierzchołkami. Pierwszy sposób interpretacji będzie dla nas szczególnie interesujący jeżeli ograniczymy zbiór wartości funkcji wagowej do przedziału $\langle 0, 1\rangle$. Każdy graf ważony nieskierowany (lub skierowany), spełniający to ograniczenie, będzie reprezentacją pewnej relacji rozmytej zdefiniowanej przez funkcję przynależności $\mu : V \times V \to \langle 0, 1\rangle$, zgodnie z Definicją 3.12, takiej że

$$\mu(v, w) = \begin{cases} \varpi(v, w), & \text{gdy } \{v, w\} \in E \text{ (lub } (v, w) \in E); \\ 0, & \text{wpp;} \end{cases} \tag{3.37}$$

gdzie $v, w \in V$ są wierzchołkami, E jest zbiorem krawędzi. Z drugiej strony możemy zauważyć, że wykorzystując powyższą interpretację funkcję wagi, można w prosty sposób reprezentować dowolny graf ważony niepełny za pomocą grafu ważonego pełnego, przypisując nieistniejącym krawędziom zerową wagę. Przy tak przyjętej funkcji wagowej możemy łatwo zdefiniowac macierz przyległości grafu ważonego.



Rys. 3.2: Fragment grafu ważonego skierowanego opisanego w Przykładzie 3.11 dla danych liczbowych z tabelki z Przykładu 3.6, zawierający krawędzie wychodzące z węzła 000. Przy każdej z krawędzi podano jej wagę równą prawdopodobieństwu jej wybrania, przy założeniu że prawdopodobieństwo mutacji $p_m = 0.1$.

Definicja 3.24 *Niech pełny graf ważony nieskierowany (skierowany) $G = (V, V \times V, \varpi)$ o n wierzchołkach będzie reprezentantem relacji rozmytej definiowanej przez funkcje przynależności μ.* **Macierzą przyległości grafu ważonego** *G nazwiemy macierz A o wymiarach $n \times n$ o elementach*

$$A_{ij} = \varpi(v_i, v_j) = \mu(v_i, v_j). \tag{3.38}$$

□

Przykład 3.11 *Rozważmy opis standardowego operatora mutacji za pomocą relacji rozmytej z Przykładu 3.6. Uzyskamy równoważny mu opis za pomocą grafu ważonego skierowanego przyjmując funkcję wagi $\varpi(\underline{b}_1, \underline{b}_2) = \mu(\underline{b}_1, \underline{b}_2)$ dla wszystkich możliwych par $(\underline{b}_1, \underline{b}_2)$. Jeśli przyjmiemy $p_m = 0.1$, to biorąc dane z tabelki z omawianego przykładu, możemy graficznie przedstawić uzyskany graf zgodnie z Rys. 3.2. Ponieważ mamy do czynienia z symetryczną relacją rozmytą, więc możemy ją przedstawić za pomocą grafu ważonego nieskierowanego, który można narysować podobnie jak na Rys. 3.2; należy jedynie pominąć orientację krawędzi.* □

3.5.4 Ścieżki

Jedną z najważniejszych operacji na grafie jest przeglądanie grafu, czyli przechodzenie między kolejnymi wierzchołkami połączonymi krawędziami. Sposób przeglądania grafu jest opisywany ścieżką.

Definicja 3.25 **Ścieżką** *(ang. path) (inaczej marszrutą w* [16] *lub drogą w* [9]*) o długości l w grafie $G = (V, E)$ nazywamy dowolny ciąg $\Pi = (v_0, e_1, v_1, e_2, v_2, \ldots, v_{l-1}, e_{l-1}, v_l)$, dla którego spełniony jest warunek*

$$\bigwedge_{i \in \{1,\ldots,l\}} e_i = (v_{i-1}, v_i), \tag{3.39}$$

gdzie $v_i \in V$ oraz $e_i \in E$. □

W praktyce często ścieżki zapisuje się w formie skróconej, pomijając oznaczenia krawędzi, na przykład ścieżkę z powyższej definicji możemy zapisać: $\Pi = (v_0, v_1, v_2, \ldots, v_{l-1}, v_l)$ - dlatego możemy przyjąć, że $\Pi \in V^l$. Zauważmy, że gdy $l = \infty$ otrzymujemy nieskończoną ścieżkę o początku v_0; piszemy wtedy $\Pi = (v_0, v_1, \ldots) \in V^\infty$.

Pewne informacje o ścieżkach w grafie zwykłym (nieważonym) możemy uzyskać obliczając potęgi macierzy przyległości. Wyjaśnia to poniższy lemat (por. [9] i [30]).

Lemat 3.1 *Niech A będzie macierzą przyległości grafu zwykłego. Element i, j w macierzy A^k (k-ta potęga A) jest równy liczbie różnych ścieżek o długości k z wierzchołka i-tego do j-tego.* □

Definicja 3.26 *Niech pełny graf ważony $G = (\mathcal{B}, \mathcal{B} \times \mathcal{B}, \varpi)$ będzie reprezentacją pewnego operatora przemieszczenia $\phi : \mathcal{B} \to \mathcal{B}$ i będzie zbudowany na podstawie rozmytej relacji sąsiedztwa tego operatora, zgodnie ze wzorem (3.22).* Ścieżką losową z $\underline{b}$ *(ang. random walk) nazwiemy dowolną ścieżkę, której i-ty element będzie równy i-krotnemu działaniu operatora na $\underline{b}$, czyli $\underline{b}_i = \phi^i(\underline{b})$ dla $i = 0, 1, 2, \ldots$. Ścieżki losowe nieskończone będziemy oznaczali symbolem $\Pi_{\underline{b},\phi} \in \mathcal{B}^\infty$, natomiast ścieżki skończone składające się z $l+1$ elementów początkowych ścieżki (dla $i = 0, 1, \ldots, l$) - symbolem $\Pi_{\underline{b},\phi,l} \in \mathcal{B}^l$.* □

Prawdopodobieństwo uzyskania na podstawie powyższej definicji pewnej skończonej ścieżki $\hat{\Pi} = (\underline{b}_0, \underline{b}_1, \ldots, \underline{b}_l)$ wynosi

$$\mathbf{P}\left(\Pi_{\underline{b}_0,\phi,l} = \hat{\Pi}\right) = \mathbf{P}\left(\left(\underline{b}_0, \phi(\underline{b}_0), \phi^2(\underline{b}_0), \ldots, \phi^l(\underline{b}_0)\right) = \hat{\Pi}\right) = \prod_{i=1}^{l} \varpi(\underline{b}_{i-1}, \underline{b}_i), \tag{3.40}$$

co zarazem definiuje nam dyskretny rozkład prawdopodobieństwa ścieżki losowej będącej w istocie l-wymiarową dyskretną zmienną losową. Ścieżka jako zmienna losowa jest funkcją wektora zmiennych losowych, którego składowe, odpowiadająca kolejnym użyciom operatora ϕ, są równe zmiennej losowej θ, ukrytej w operatorze przemieszczenia (patrz Def. 3.10). Dokładne omówienie właściwości losowych ścieżek znajdzie czytelnik na przykład w pracy Spitzera [7].

3.6 Uogólniony operator krzyżowania

3.6.1 Definicje

Pomimo, że w niniejszej rozprawie ograniczamy się niemal wyłącznie do analizy operatora krzyżowania działającego na osobnikach będących binarnymi chromosomami, należy zauważyć, że jego cenne właściwości nie koniecznie muszą być związane ze szczególną postacią osobników.

Wprowadzimy teraz za Kolonką [29] uogólniony operator krzyżowania działający na osobnikach należących do dowolnego zbioru I.

Definicja 3.27 *Niech I będzie dowolnym zbiorem osobników, U - przestrzenią parametrów krzyżowania. Dla każdego $u \in U$ definiujemy operator krzyżowania $c_u : I \times I \to I \times I$ jako parę odwzorowań $C_u, \overline{C}_u : I \times I \to I$, takich że dla pary rodziców $x, y \in I$, $C_u(x, y)$ i $\overline{C}_u(x, y)$ jest parą ich potomków.*

Dodatkowo dla każdego $u \in U$ definiujemy operatory $\varphi_u : I \to I$ oraz $\overline{\varphi}_u : I \to I$, dokonujące podziału dowolnego osobnika $x \in I$ na komplementarne fragmenty $\varphi_u(x)$ i $\overline{\varphi}_u(x)$. Przyjmujemy dla ułatwienia, że fragmenty osobników również są elementami I.

Definiujemy także odwzorowanie $h : I \times I \to I$ łączące fragmenty osobników w nowe osobniki, takie że:

$$C_u(x, y) = h(\varphi_u(x), \overline{\varphi}_u(y)) \text{ oraz } \overline{C}_u(x, y) = h(\varphi_u(y), \overline{\varphi}_u(x)) \tag{3.41}$$

Zakładamy przy tym, że odwzorowania $\varphi_u, \overline{\varphi}_u$ i h spełniają warunek:

$$\bigwedge_{x,y,z\in I}\bigwedge_{u\in U} h(\varphi_u(x),\overline{\varphi}_u(y)) = z \iff \varphi_u(x) = \varphi_u(z) \wedge \overline{\varphi}_u(y) = \overline{\varphi}_u(z), \tag{3.42}$$

który zapewnia, że podczas krzyżowania nie następuje utrata informacji (materiału genetycznego). □

Lemat 3.2 *Dla każdego $x, y \in I$ i dla każdego $u \in U$ zachodzą:*

1. $C_u(x,x) = \overline{C}_u(x,x) = x$;
2. $C_u\left(C_u(x,y),\overline{C}_u(x,y)\right) = x$ *i* $\overline{C}_u\left(C_u(x,y),\overline{C}_u(x,y)\right) = y$;
3. $c_u : I \times I \to I \times I$ *jest bijekcją, oraz* $c_u^{-1} \equiv c_u$;
4. *dla każdego podzbioru* $F \subset I \times I$ *mamy:* $c_u(F \cup c_u(F)) = F \cup c_u(F)$

□

Dowód

1. Wynika z zastosowania części '$\Longleftarrow$' warunku (3.42).
2. Weźmy $v := C_u(x,y)$ i $w := \overline{C}_u(x,y)$. Wtedy z warunku (3.42) mamy: $\varphi_u(v) = \varphi_u(x)$, $\overline{\varphi}_u(v) = \overline{\varphi}_u(y), \varphi_u(w) = \varphi_u(y)$ oraz $\overline{\varphi}_u(w) = \overline{\varphi}_u(x)$. Korzystając z 1. mamy: $x = h(\varphi_u(x),\overline{\varphi}_u(x)) = h(\varphi_u(v),\overline{\varphi}_u(w)) = C_u\left(C_u(x,y),\overline{C}_u(x,y)\right)$ i podobnie dla y.
3. Wynika z 2.
4. Wynika z 3.

□

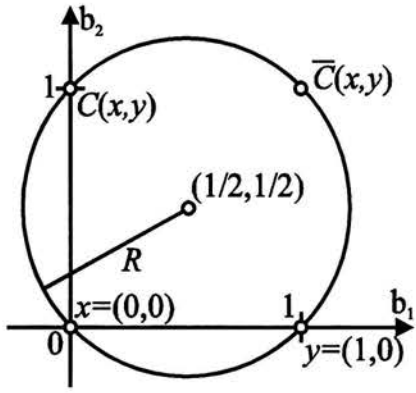
3.6.2 Uogólniony operator krzyżowania w $\mathcal{BAG}$

Dla $\mathcal{BAG}$ mamy $I := \mathcal{B}$ oraz przyjmujemy, że przestrzeń parametrów krzyżowania $U := \mathcal{B}$. Przyjmijmy, że $\oplus$ oznacza binarną operacją dodawania modulo 2 (XOR), a $\otimes$ oznacza binarny iloczyn (AND). Możemy wtedy przypisać: $\varphi_u(x) := u \otimes x$, $\overline{\varphi}_u(x) := (\mathbf{1} \oplus u) \otimes x$ oraz $h(x,y) := x \oplus y$. Zgodnie z (3.41) otrzymujemy:

$$C_u(x,y) := (u \otimes x) \oplus ((\mathbf{1} \oplus u) \otimes y) \quad \text{oraz} \quad \overline{C}_u(x,y) := (u \otimes y) \oplus ((\mathbf{1} \oplus u) \otimes x) \tag{3.43}$$

Zależnie od sposobu wyboru łańcucha u możemy uzyskać różne rodzaje operatorów krzyżowania, najważniejsze z nich wymieniamy poniżej.

- Jeżeli będziemy wybierali łańcuch u w sposób losowy, to otrzymamy operator jednorodny krzyżowania zdefiniowany w pracy Qi i Palmieriego [24].
- Jeżeli ograniczymy postać łańcucha u do binarnej reprezentacji liczby $2^k - 1$, dla wybieranego losowo $k \in \{1,\ldots,N\}$ (czyli $u = 0\ldots01\ldots1$), to otrzymamy standardowe krzyżowanie jednopunktowe zdefiniowane przez Hollanda w [5].
- Jeżeli ograniczymy postać łańcucha u do binarnej reprezentacji liczby $(2^{k_1-k_2}-1)2^{k_2}$, dla wybieranych losowo $k_1, k_2 \in \{0,\ldots,N\}$ przy założeniu, że $k_1 \geq k_2$ (czyli $u = 0\ldots01\ldots10\ldots0$), to otrzymamy krzyżowanie dwupunktowe.



Rys. 3.3: Rozmieszczenie na kuli rodziców i potomków powstałych w wyniku krzyżowania dla przypadku dwuwymiarowego.

Gdy założymy, że każdy wektor dwójkowy określa współrzędne punktu w N-wymiarowej przestrzeni euklidesowskiej, to zauważymy, że zarówno rodzice, jak i dzieci powstałe w wyniku krzyżowania leżą na powierzchni d-wymiarowej kuli o środku w punkcie $S = (\underline{b}_1 + \underline{b}_2)/2$ o promieniu $R = 1/2\sqrt{d}$ gdzie $d = d_H(\underline{b}_1, \underline{b}_2)$ jest odległością Hamminga między obojgiem rodziców $\underline{b}_1, \underline{b}_2$. Przypadek $d = 2$ pokazano na rys. 3.3.

W przypadku $\mathcal{BAG}$ wykorzystującego kodowanie za pomocą łańcuchów o długości N możemy zdefiniować maksymalnie 2^{N-1} różnych krzyżowań. Łatwo to uzasadnić zauważając, że każde dwa łańcuchy $u, u' \in U$ takie, że kolejne bity u' są negacjami odpowiednich bitów u, definiują takie same krzyżowania, w identyczny sposób rozdzielające bity rodziców pomiędzy potomków. Jedyną różnicą jest zamiana potomków miejscami.

Poniższe twierdzenie dotyczy rozkładu prawdopodobieństwa operatora uogólnionego krzyżowania dla przypadku krzyżowania jednorodnego – będzie ono wykorzystane w następnych rozdziałach.

Twierdzenie 3.2 *Niech $u \in \mathcal{B}$ będzie operatorem uogólnionego krzyżowania wybranym ze swej dziedziny losowo z rozkładem równomiernym. Wtedy:*

1. *prawdopodobieństwo że operator u będzie miał dokładnie k jedynek jest równe:*

$$\mathbf{P}(ones(u) = k) = \frac{\binom{N}{k}}{2^N}; \tag{3.44}$$

2. *prawdopodobieństwo że operator u będzie miał postać pewnego łańcucha $\underline{b} \in \mathcal{B}$ jest równe:*

$$\mathbf{P}(u = \underline{b}) = \frac{1}{2^N}. \tag{3.45}$$

□

3.6.3 Właściwości rozkładu prawdopodobieństwa potomstwa

Zbadamy teraz rozkład prawdopodobieństwa potomstwa $c_\chi(X, Y)$ uzyskanego w wyniku działania losowego krzyżowania χ na losowo wybranych rodzicach (X, Y).

Twierdzenie 3.3

1. *Dla każdego podzbioru $F \subset I \times I$ mamy* $\mathbf{P}\left(c_\chi(X, Y) \in F\right) = \mathbf{P}\left((X, Y) \in c_\chi(F)\right)$.

2. *Załóżmy, że* (X,Y) *i* χ *są niezależne, oraz* (X,Y) *ma gęstość* p *względem miary* ρ *na* $I \times I$. *Dodatkowo niech:*

$$\bigwedge_{u \in U} \bigwedge_{F \subset I \times I} \rho(c_u(F)) = \rho(F).$$

Wtedy prawdziwe są następujące trzy twierdzenia:

(a) $c_\chi(X,Y)$ *ma warunkową* ρ*-gęstość* $(v,w) \mapsto p(c_u(v,w))$ *dla* $\chi = u$, *taką, że:*

$$\bigwedge_{F \subset I \times I} \mathbf{P}(c_u(X,Y) \in F) = \int_F \rho(dv,dw)\, p(c_u(v,w));$$

(b) $c_\chi(X,Y)$ *ma bezwarunkową* ρ*-gęstość*

$$(v,w) \mapsto \int_F \mathbf{P}_\chi(du)\, p(c_u(v,w));$$

(c) jeśli dodatkowo $\rho = \mu \times \mu$ *dla pewnej miary* μ *na* I, *wtedy* $c_\chi(X,Y)$ *ma warunkową i bezwarunkową* μ*-gęstość*

$$v \mapsto \int \mu(dw)\, p(c_u(v,w)),$$

oraz odpowiednio

$$v \mapsto \int \mu(dw) \int_U \mathbf{P}_\chi(du)\, p(c_u(v,w)).$$

□

Twierdzenie powyższe zostało udowodnione w pracy [29]. Potwierdza ono ważne cechy klasycznej operacji krzyżowania: (1) potomstwo zachowuje jednostajny rozkład rodziców, oraz (2) losowe krzyżowanie powoduje wzrost entropii. Cechy te są sfomułowane w kolejnym wniosku.

Wniosek 3.3

1. *Jeśli* (X,Y) *ma jednostajny rozkład na pewnym podzbiorze* $F \subset I \times I$ *wtedy, mając* $\chi = u$, *rozkład warunkowy pary potomków* $c_\chi(X,Y)$ *jest rozkładem jednostajnym na* $c_u(F)$. *W szczególności, jeśli* X,Y *są niezależne i mają w* I *rozkład jednostajny, wtedy potomkowie* $C_\chi(X,Y)$ *i* $\overline{C}_\chi(X,Y)$ *są także niezależni i mają rozkład jednostajny.*

2. *Jeśli entropię zdefiniujemy jako* $H(X,Y) := -\mathbf{E}(\log p(X,Y))$ *to zachodzą:*

   (a) $\bigwedge_{u \in U} H(X,Y) = H(c_u(X,Y))$,

   (b) $H(X,Y) < H(c_\chi(X,Y))$.

□

Dokładny dowód punktu 2 może czytelnik znaleźć w pracy [29]. Punkt 2a dotyczy właściwości krzyżowania deterministycznego, natomiast 2b – krzyżowania losowego. Widać, że nierówność 2b oznacza zdolność operatora krzyżowania do eksploracji przestrzeni poszukiwań. Sugeruje to możliwość wykorzystania różnicy

$$H(c_\chi(X,Y)) - H(X,Y) \tag{3.46}$$

jako miary zdolności eksploracyjnych danego operatora krzyżowania.

3.6.4 Uogólnione pojęcie schematów

Stosując wprowadzone dotychczas pojęcia dotyczące uogólnionego pojęcia krzyżowania, można w przejrzysty sposób zdefiniowac pojęcie schematu. Należy zauważyć, że pojęcie schematu jest pojęciem ogólnym, podobnie jak krzyżowanie, nie dotyczącym wyłącznie binarnych chromosomów, oraz że oba pojęcia są bardzo spokrewnione.

Definicja 3.28 *Niech I będzie zbiorem osobników, U – przestrzenią rodzajów krzyżowania jak powyżej. Definiujemy zbiory:*

$$\Phi_u(a) := \varphi_u^{-1} \circ \varphi_u(a) = \{z \in I \mid \varphi_u(z) = \varphi_u(a)\}$$

i podobnie

$$\overline{\Phi}_u(a) := \overline{\varphi}_u^{-1} \circ \overline{\varphi}_u(a).$$

Zbiór $\Phi_u(a)$ nazwiemy schematem definiowanym przez u i a. □

Przykład 3.12 *W przypadku chromosomów binarnych przyjmujemy φ_u jak w przykładzie 1. Wtedy $\Phi_u(a)$ jest klasycznym schematem zawierającym na pozycji k wartość ustaloną a_k jeśli $u_k = 1$, albo gwiazdkę '*' gdy $u_k = 0$.* □

Pokażemy w poniższym lemacie, w jaki sposób krzyżowanie operuje na schematach, a w szczególności w jakiej sytuacji schemat pozostaje nieuszkodzony w wyniku krzyżowania. Pozwoli to nam następnie na uogólnienie klasycznego Twierdzenia o Schematach.

Lemat 3.3 *Niech $u, u' \in U$ oraz $y, a \in I$. Wtedy:*

1. *jeżeli $\varphi_u \circ \varphi_{u'} = \varphi_u$ to $C_{u'}(\cdot, y)$ przekształca $\Phi_u(a)$ na siebie.*
2. *jeżeli $\varphi_u \circ \overline{\varphi}_{u'} = \varphi_u$ to $\overline{C}_{u'}(\cdot, y)$ przekształca $\Phi_u(a)$ na siebie.*

□

Dowód Niech $x \in \Phi_u(a)$ i $\varphi_u \circ \varphi_{u'} = \varphi_u$, wtedy z warunku (3.42)

$$\begin{aligned} &\varphi_u(C_{u'}(x,y)) = \varphi_u(h(\varphi_{u'}(x), \overline{\varphi}_{u'}(y))) = \\ &\varphi_u \circ \varphi_{u'}(h(\varphi_{u'}(x), \overline{\varphi}_{u'}(y)) = \varphi_u \circ \varphi_{u'}(x) = \varphi_u(x) = \varphi_u(a) \end{aligned} \tag{3.47}$$

i podobnie dla 2. □

Przykład 3.13 *W przypadku $\mathcal{BAG}$ z jednorodnym krzyżowaniem spełnienie jednego z powyższych warunków oznacza, że przynajmniej jeden z utworzonych potomków jest elementem $\Phi_u(a)$, czyli że wszystkie bity będące na pozycjach ustalonych w $\Phi_u(a)$, w wyniku krzyżowania weszły w skład tego samego potomka. W $\mathcal{BAG}$ z krzyżowaniem jednopunktowym oznacza to, że pozycja cięcia nie rozdziela w łańcuchu żadnych pozycji ustalonych.* □

Twierdzenie 3.4 *Niech zmienna losowa X o wartościach z I i μ-gęstości q na I reprezentuje aktualną populację. Niech $f : I \to \mathbb{R}$ będzie funkcją przystosowania z $\mathbf{E}(f(X)) = \int_I \mu(dx)\, q(x)\, f(x) < \infty$. Wtedy rodzice wybrani z prawdopodobieństwem proporcjonalnym do ich przystosowania będą niezależnymi zmiennymi losowymi $\hat{X}, \hat{Y}$ o identycznym rozkładzie i μ-gęstości $\hat{q}(x) := q(x)\, f(x)/\mathbf{E}(f(X))$. Potomkowie powstali w wyniku skrzyżowania rodziców losowo wybranym operatorem krzyżowania χ są oznaczeni przez $C_\chi(\hat{X}, \hat{Y})$ i $\overline{C}_\chi(\hat{X}, \hat{Y})$.*

Dla ustalonego $u \in U$ *definiujemy* $\hat{U} := \{u' \in U \mid \varphi_u \circ \varphi_{u'} = \varphi_u \vee \varphi_u \circ \overline{\varphi}_{u'} = \varphi_u\}$. *Wtedy dla każdego* $a \in I$ *mamy:*

$$\mathbf{P}\left(C_\chi(\hat{X},\hat{Y}) \in \Phi_u(a) \vee \overline{C}_\chi(\hat{X},\hat{Y}) \in \Phi_u(a)\right) \geq$$
$$\geq \mathbf{P}\left(X \in \Phi_u(a)\right) \mathbf{P}\left(\chi \in \hat{U}\right) \frac{\mathbf{E}\left(f(X) \mid X \in \Phi_u(a)\right)}{\mathbf{E}\left(f(X)\right)}.$$

□

Twierdzenie 3.4 zostało udowodnione w pracy [29]. Jest ono dla $\mathcal{BAG}$ uogólnieniem klasycznego Twierdzenia o Schematach (Tw. 3.1) i pozwala na obliczenie dolnego ograniczenia prawdopodobieństwa przeżycia schematu $\Phi_u(a)$ przy selekcji dokonywanej standardową metodą ruletki i przy jednorodnym krzyżowaniu $\chi \in \hat{U}$.

Należy zauważyć, że wyrażenie $\mathbf{E}\left(f(X) \mid X \in \Phi_u(a)\right)$ odpowiada zdefiniowanemu wzorem (3.14) pojęciu przystosowania schematu w populacji. Podobnie możemy przyjąć, że

$$\mathbf{P}\left(X \in \Phi_u(a)\right) \approx \frac{m(\underline{s}, G)}{|\underline{s}|}, \tag{3.48}$$

gdzie $m(\cdot,\cdot)$ zdefiniowano jak w (3.13).

Rozdział 4

Adaptacja binarnego algorytmu genetycznego

4.1 Wprowadzenie do adaptacji algorytmów optymalizacji

Jak stwierdzono w pracy [31], nie istnieje uniwersalny algorytm optymalizacji jednakowo dobry dla wszystkich możliwych problemów. Stąd też przypuszczalnie jedyną metodą zwiększenia wydajności optymalizacji jest dobranie właściwego algorytmu oraz jego parametrów do danego zadania. Z praktycznego punktu widzenia wygodniej jest wykorzystywać jeden algorytm do wszystkich problemów dobierając jedynie jego parametry do danego zadania.

Dobrym kandydatem na algorytm uniwersalny jest Binarny Algorytm Genetyczny ($\mathcal{BAG}$). Algorytm ten w swej standardowej postaci jest wystarczająco skuteczny dla dużej liczby rzeczywistych problemów, ale wydaje się, że można jeszcze polepszyć jego działanie i rozszerzyć zakres zastosowań poprzez adaptację jego parametrów odpowiednio do rozwiązywanego problemu. Z adaptacją jest ściśle związane zagadnienie analizy rozwiązywanego problemu oraz zagadnienie oceny działania adaptowanego algorytmu. W niniejszym rozdziale postaramy się omówić powyższe zagadnienia oraz nakreślimy wiążące je relacje.

4.1.1 Podstawowe pojęcia

Ogólnie metody adaptacji parametrów dowolnego algorytmu optymalizacji możemy podzielić na dwie grupy:

1. **metody off-line** – porządane wartości parametrów są wyznaczane na podstawie analizy (pomiarów) problemu zdefiniowanego przy pomocy funkcji przystosowania przed użyciem algorytmu optymalizacji. Ze względu na rodzaj metody analizy możemy wyróżnić metody w których:

   (a) sam algorytm optymalizacji jest wykorzystywany jako metoda analizy problemu,

   (b) wykorzystana jest odrębna metoda analizy problemu.

2. **metody on-line** – modyfikacja parametrów jest przeprowadzana w trakcie pracy algorytmu na podstawie analizy (pomiarów) dotychczasowego działania algorytmu. Ze względu na stopień powiązania mechanizmu adaptacyjnego z samym algorytmem genetycznym (patrz praca [26]) możemy wyróżnić metody adaptacji w których:

   (a) algorytm genetyczny sam siebie adaptuje w trakcie rozwiązywania problemu *(ang. tightly coupled)*;

   (b) pewne elementy algorytmu genetycznego (np. operatory genetyczne) są wykorzystywane przez mechanizm adaptacji *(ang. loosely coupled)*;

(c) jest wykorzystywany całkiem odrębny mechanizm adaptacji *(ang. uncoupled)*.

Można przyjąć założenie, że każda metoda adaptacji wykorzystuje pewną metodę analizy problemu lub działania algorytmu, przy czym adaptacja parametrów algorytmu optymalizacji polega zwykle na "połączeniu" wyników analizy z wartościami parametrów pewnego rodzaju sprzężeniem zwrotnym.

Podsumowując należy zauważyć, że we wszystkich metodach adaptacji mamy do czynienia z czterema najważniejszymi zbiorami:

- **przestrzeń rozwiązań** X – omówiona we wprowadzeniu;
- **przestrzeń kodowa** – w przypadku $\mathcal{BAG}$ jest nią przestrzeń N-bitowych łańcuchów $\mathcal{B}$ (przy ustalonej funkcji kodującej, może być używana równoważnie z przestrzenią rozwiązań X podczas analizy działania $\mathcal{BAG}$);
- **przestrzeń funkcji przystosowania** $\mathcal{F}$ – ogólnie rodzina wszystkich odwzorowań typu $X \to \mathbb{R}$, w przypadku $\mathcal{BAG}$ zgodna z Definicją 3.2;
- **przestrzeń parametrów** P algorytmu optymalizacji;
- **przestrzeń pomiarów** M zawierająca wszystkie dopuszczalne wartości wyników analizy (pomiarów) problemu lub działania algorytmu;
- **przestrzeń porównań** – dokonujemy w niej wszelkich porównań: zarówno jakości wygenerowanych rozwiązań, jak jakości działania algorytmu optymalizacji; zwykle jest nią zbiór liczb rzeczywistych $\mathbb{R}$ (w pewnych zagadnieniach nie da się w tym celu użyć zbioru $\mathbb{R}$ – stosuje się wtedy optymalizację wielokryterialną).

Podobnie w wiekszości metod adaptacji można wyodrębnić elementy składowe, które w skrócie omówiamy poniżej.

- **metoda oceny sprawności algorytmu optymalizacji** – stosowana do porównywania działania algorytmu z różnymi wartościami parametrów; sprawność może być rozumiana np. jako oczekiwana szybkość zbieżności i/lub oczekiwana jakość znajdywanych rozwiązań – ogólnie mówiąc sprawność może oznaczać dowolne pożądane cechy działania algorytmu. Sprawność możemy zwykle przedstawiać jako funkcję typu $\mathcal{F} \times P \to \mathbb{R}$, której większa wartość odpowiada wyższej sprawności algorytmu optymalizacji. Funkcja ta jest zależna od rozwiązywanego problemu oraz od wartości adaptowanego parametru.
- **metoda analizy problemu off-line lub działania algorytmu on-line** – powinna umożliwiać ocenę czynników wpływających na sprawność algorytmu – może być przedstawiona jako funkcja typu $\mathcal{F} \to M$ dla metod off-line, oraz jako $D \to M$ dla metod on-line, gdzie D jest zbiorem możliwych opisów działania algorytmu optymalizacji;
- **algorytm optymalizacji**, którego parametry będą adaptowane – parametry adaptowalne powinny być tak dobrane, aby pozwalały zmieniać w możliwie szerokim zakresie sposób działania algorytmu, przy czym liczba parametrów i ich możliwych wartości nie powinna być zbyt wielka, gdyż w przeciwnym wypadku samo przeszukiwanie przestrzeni parametrów staje się zadaniem dla algorytmu optymalizacji;
- **metoda wyznaczania wartości adaptowanych parametrów** na podstawie wyników analiz, czyli odwzorowanie typu $M \to P$ – znalezienie odpowiedniej zależności jest zwykle bardzo trudnym zadaniem, trudniejszym niż samo przeszukiwanie przestrzeni P, dlatego często stosuje się tutaj inny – pomocniczy – algorytm optymalizacji, albo stosuje się adaptację typu 2a;

Najistotniejsze, spośród wyżej wymienionych, elementy składowe metod adaptacji omówimy szerzej w kolejnych podrozdziałach.

4.1.2 Cele adaptacji $\mathcal{BAG}$

Celem adaptacji każdego algorytmu optymalizacji jest poprawa jego sprawności działania. W przypadku $\mathcal{BAG}$, jego sprawność działania może być rozumiana na różne sposoby. Zwykle metody oceny sprawności można podzielić na dwie główne grupy:

1. Znajdujemy minimalną liczbę pokoleń $\mathcal{BAG}$, przy której wygenerowane rozwiązania mają pewne pożądane właściwości. Przy czym, oczywiście, algorytm, dla ktorego ta liczba jest mniejsza uznajemy za sprawniejszy.

2. Przyjmujemy, że $\mathcal{BAG}$ generuje pewną ustaloną liczbę pokoleń. Uznajemy za sprawniejszy ten algorytm, który wygenerował rozwiązania posiadające pożądane właściwości "w wyższym stopniu".

Spotykane w literaturze opisy algorytmów genetycznych pozwalają zauważyć, że przez "pożądane właściwości rozwiązań" zwykle rozumie się:

- maksymalizację wartość przystosowania najlepszego z wygenerowanych osobników z danego pokolenia (zwykle dla metod z 2. grupy);

- minimalizację różnicy przystosowania między najlepszym osobnikiem z danego pokolenia a globalnym optimum (tylko dla problemów testowych o znanych rozwiązaniach);

- minimalizacja wzrostu przystosowania najlepszego osobnika w ustalonej liczbie kolejnych pokoleń (zwykle dla metod z 1. grupy).

Ogólnie celem każdego algorytmu optymalizacji jest znalezienie optimum globalnego, ale nietety większość algorytmów nie gwarantuje, że ten cel zawsze będzie osiągnięty, a dodatkowo zwykle nie wiemy, czy już znaleźliśmy optimum, ani nawet jak blisko jego jesteśmy. Dlatego ocena sprawności algorytmu zwykle ma charakter względny i pod tym kątem definiuje się pożądane właściwości wygenerowanych przez algorytm rozwiązań.

Sprawność algorytmu optymalizacji zwykle oceniamy nie na podstawie jego jednokrotnego uruchomienia, ale wielokrotnego, dla losowo wybranych danych początkowych. Do oceny sprawności bierzemy wyniki średnie ze wszystkich uruchomień.

4.1.3 Analiza funkcji przystosowania

Sprawność $\mathcal{BAG}$ może być wyznaczona jedynie na podstawie przebiegu wielu jego uruchomień. Jest to zadanie bardzo kosztowne obliczeniowo i zwykle czasochłonne. Dlatego podejmowane są próby oceny przewidywanej sprawności $\mathcal{BAG}$ bez jego uruchamiania. Ocena taka może być dokonywana wyłącznie na podstawie analizy właściwości rozwiązywanego problemu, który jest definiowany przez funkcję przystosowania. Oczywiście jest celowe wykorzystanie wyłącznie takich metod analizy funkcji przystosowania, które są mniej złożone obliczeniowo od $\mathcal{BAG}$.

Ze względu na charakter zastosowań możemy wszystkie spotykane w literaturze metody analizy funkcji przystosowania podzielić na dwie grupy:

A. metody wymagające znajomości optimum globalnego – mające znaczenie poznawcze i pomocnicze – w rzeczywistych problemach optimum nie jest znane, czasami jest znane jedynie jego oszacowanie lub tylko oszacowanie maksymalnej wartości funkcji przystosowania;

B. metody nie wymagające znajomości optimum globalnego – nadające się do wykorzystania w problemach praktycznych.

Jak wspomniano wcześniej, wprawdzie analizę funkcji przystosowania można najłatwiej wyodrębnić w metodach adaptacji off-line, jednakże należy zauważyć, że sam $\mathcal{BAG}$ może być wykorzystywany jako metoda analizy funkcji przystosowania. Przykładem mogą być metody adaptacji on-line, w których można przyjąć, że mamy do czynienia z lokalną analizą funkcji przystosowania, która odbywa się na bieżąco na podstawie przebiegu działania algorytmu.

Ograniczymy od tego momentu zakres naszych rozważań do Binarnego Algorytmu Genetycznego i dokonamy przeglądu spotykanych w literaturze metod analizy funkcji przystosowania oraz metod adaptacji jego parametrów.

4.2 Przegląd metod analizy funkcji przystosowania i adaptacji $\mathcal{BAG}$

4.2.1 Metody analizy funkcji przystosowania

Obliczanie korelacji odległości do globalnego maksimum i przystosowania

Metoda typu A opisana została przez Jonesa i Forresta (1996) [32]. Zaproponowali oni, aby obliczać współczynnik korelacji r_{FD} (*fitness to distance*) miedzy odległością Hamminga danego rozwiązania od globalnego maksimum, a jego przystosowaniem, dla pewnego losowego n-elementowego podzbioru rozwiązań $\{\underline{b}_1, \ldots, \underline{b}_n\}$:

$$r_{FD} = \frac{c_{FD}}{s_F\, s_D} \tag{4.1}$$

gdzie $F = (f_1, \ldots, f_n)$, $D = (d_1, \ldots, d_n)$, oraz $f_i = f(\underline{b}_i)$, $d_i = d_H(\underline{b}_i, \underline{b}_{opt})$ dla $i = 1, \ldots, n$; c_{FD} jest kowariancją, a s_F i s_D – odchyleniami standardowymi ciągów F i D.

Mając obliczoną w ten sposób wartość r_{FD}, możemy dokonać klasyfikacji funkcji przystosowania:

- zwodnicza dla $0.15 \leq r_{FD} \leq 1$;
- trudna dla $-0.15 \leq r_{FD} \leq 0.15$;
- łatwa dla $-1 \leq r_{FD} \leq -0.15$.

Istnieją jednakże problemy, w których metoda ta daje błędne oszacowania trudności funkcji przystosowania. Altenberg w pracy [40] podaje przykład takiego problemu. Proponowany problem jest łatwy dla $\mathcal{BAG}$ gdyż z każdego punktu przestrzeni $\mathcal{B}$ istnieje ścieżka do optimum globalnego o rosnącym przystosowaniu, przy czym nie występuje zależność przystosowania od odległości Hamminga od optimum globalnego. Altenberg proponuje również kilka sposobów ulepszenia omawianej metody oceny funkcji przystosowania.

Statystyczne pomiary losowych ścieżek w przestrzeni poszukiwań

Przykładem metody typu B jest metoda zaproponowana przez Hordijka w pracy [34]. Opiera się ona na pomiarach losowych ścieżek *(ang. random walks)* w krajobrazie przystosowania *(ang. fitness landscape)*. Ścieżki te są tworzone iteracyjnie poprzez wielokrotne użycie wybranego operatora genetycznego (np. krzyżowania lub mutacji) zaczynając od losowo wybranego łańcucha początkowego. Uzyskany ciąg kolejnych wartości funkcji przystosowania może być traktowny jak dyskretny sygnał $F = (f_1, \ldots, f_n)$. Autor stosuje do niego typowe metody analizy sygnałów, w szczególności znajduje funkcję autokorelacji, estymowaną przez

$$r_s = \frac{\sum_{i=1}^{n-s}(f_i - \overline{f})(f_{i+s} - \overline{f})}{\sum_{i=1}^{n-s}(f_i - \overline{f})^2}, \quad n \gg 0, \tag{4.2}$$

oraz identyfikację modelu ARMA *(ang. AutoRegresive Moving-Average)* tego sygnału. Korzystając z funkcji autokorelacji autor definiuje parametr nazywany *długością korelacji*, którego większa wartość oznacza że badana funkcja przystosowania jest gładsza, i powinna być łatwiejsza do optymalizacji. Do testowania swojej metody autor wykorzystuje funkcje przystosowania wygenerowane za pomocą *NK-modelu Kauffmana* wprowadzonego w pracy [17].

Analiza fourierowska krajobrazów przystosowania

Bardzo ciekawą metodą analizy krajobrazów przystosowania, bazującym na ich reprezentacji przy pomocy grafów, jest metoda analizy fourierowskiej, czyli widmowej. Podejście to zostało pierwotnie zaproponowane jako jedno z narzędzi w teorii grafów, natomiast jego szczególne zastosowanie do badania krajobrazów przystosowania omawia szeroko Stadler w pracach [30] i [38] oraz Hordijk i Stadler w pracy [41].

Podstawowym pojęciem w analizie fourierowskiej jest rozwinięcie krajobrazu przystosowania w szereg Fouriera. Mając dowolny krajobraz przystosowania $L = (f, G)$, gdzie $f \in \mathcal{F}$ jest funkcją przystosowania a $G = (\mathcal{B}, E)$ i $E \subset \mathcal{B} \times \mathcal{B}$, możemy zapisać **rozwinięcie L w szereg Fouriera** jako

$$f(\underline{b}) = \sum_{k=0}^{|\mathcal{B}|-1} a_k \varphi_k(\underline{b}), \qquad \underline{b} \in \mathcal{B}, \tag{4.3}$$

gdzie a_k są **współczynnikami Fouriera**, a φ_k są ortonormalnymi funkcjami własnymi laplasjanu $-\Delta$ grafu G (patrz Def. 3.22).

4.2.2 Metody adaptacji off-line

Autorowi nie udało się znaleźć w literaturze opisów tego typu metod za wyjątkiem prac [43], [42], [44] i [45], których jest autorem lub współautorem.

4.2.3 Metody adaptacji on-line

Adaptacja pozycji krzyżowania w krzyżowaniu wielopunktowym

W 1987 Schaffer i Morishima [14] zaproponowali $\mathcal{BAG}z$ adaptującym się operatorem krzyżowania wielopunktowego (metoda typu 2a). Powiększyli oni dwukrotnie długość łańcucha w ten sposób, że w pierwszej połowie łańcucha, zwanej *łańcuchem reprezentacji*, był zakodowany osobnik, natomiast druga połowa, nazywana *łańcuchem pozycji krzyżowania*, opisywała pozycje krzyżowania: "1" oznaczała, że odpowiednia pozycja w łańcuchu reprezentacji będzie pozycją krzyżowania, a "0" – brak krzyżowania. Na przykład chromosom

$$\underbrace{0\,1\,1\,0\,0\,0\,1\,1}_{\textit{reprezentacja}}\;\underbrace{0\,1\,0\,0\,1\,0\,0\,0}_{\textit{pozycje}}$$

będzie interpretowany jako

$$0\,1!1\,0\,0!0\,1\,1.$$

gdzie znak ! oznacza pozycję krzyżowania, tutaj po 2. i 5. pozycji.

Działanie tak zakodowanego krzyżowania pokazuje poniższy przykład dla reprezentacji 12-bitowej. Dla pary rodziców

a a a a a a a!a a a a a
b b b b!b b b b b!b b

otrzymujemy parę potomków

a a a a b b b a a a b b
b b b b!a a a!b b b!a a.

Widać z przykładu, że bity z łańcucha pozycji są również przemieszczane podczas krzyżowania wraz z bitami reprezentacji. Jeżeli dany chromosom w wyniku selekcji nie przeżyje, to wraz z nim giną jego pozycje krzyżowania. Odpowiedni dobór zasad krzyżowania sprawia, że dynamika zmian łańcucha pozycji odzwierciedla gromadzenie doświedczeń o tym które pozycje krzyżowania są dobre, a które nie dla aktualnej populacji.

Adaptacja przez wybór między krzyżowaniem dwupunktowym a jednorodnym

Interesującą analizę właściwości wybranych rodzajów krzyżowania wraz z wynikającym z niej algorytmem adaptacji (typu 2a) przedstawił Spears w pracy [26]. Dokonał on porównania krzyżowań n-punktowych dla $n = 1, \ldots, 6$ i krzyżowania jednorodnego ze względu na prawdopodobieństwa przeżycia schematu rzędu $o(\underline{s}) = 3$, w funkcji długości łańcucha (wyniki dla schematów dowolnego rzędu k oraz teoretyczną analizę krzyżowania n-punktowego i jednorodnego można znaleźć w pracy [18] Spearsa i De Jonga). Uzyskane wyniki pokazują, że największe prawdopodobieństwo przeżycia powyższego schematu jest dla krzyżowania dwupunktowego, a najmniejsze – dla jednorodnego.

Na podstawie tych rezultatów autor proponuje metodę adaptacji on-line, w której adaptowanym parametrem jest prawdopodobieństwo użycia jednego z dwóch rodzajów krzyżowania: dwupunktowego i jednorodnego. W celu zaimplementowania tej metody, autor dołączył do każdego chromosomu jeden bit, którego wartość '1' odpowiada krzyżowaniu jednorodnemu, a '0' – dwupunktowemu. Wartość tego bitu w chromosomach należących do danej populacji jest interpretowana w następujący sposób: więcej '1' oznacza że lepiej jest użyć krzyżowanie jednorodne, a więcej '0' – że lepiej użyć krzyżowanie dwupunktowe. W swojej pracy Spears proponuje zarówno lokalną jak i globalną metodę adaptacji wykorzystującą dodatkowy bit i analizuje konsekwencje wyboru każdej z nich.

Wykorzystanie metaalgorytmu genetycznego

W literaturze można spotkać opisy badania wpływu zmian podstawowych parametrów $\mathcal{BAG}$, takich jak liczność populacji, prawdopodobieństwa mutacji i krzyżowania, na sprawność działania algorytmu, oraz prób wykorzystania tych informacji do bieżącej modyfikacji parametrów $\mathcal{BAG}$. Przykładem sa prace: Mercer'a [8] oraz Grefenstette'a [12], w których zaproponowano użycie metaalgorytmu genetycznego do sterowania parametrami innego $\mathcal{BAG}$.

Rozdział 5

Elementy teorii ciągłego łącza informacyjnego

5.1 Wprowadzenie

W eksperymentalnych badaniach naukowych niekiedy nie można dokonać bezpośredniego pomiaru interesującej nas cechy badanego obiektu – **cechy szukanej**. W takiej sytuacji można próbować mierzyć inną cechę tego obiektu, tzn. **cechę mierzoną**, o której wiemy (albo przypuszczamy), że jest w jakiś sposób zależna od cechy szukanej. Następnie możemy próbować pośrednio ocenić cechę szukaną na podstawie dokonanych pomiarów cechy mierzonej.

W opisanej sytuacji mamy do czynienia z dwoma najważniejszymi problemami.

- po pierwsze, konieczna jest znajomość sposobu wnioskowania o cesze szukanej na podstawie cechy mierzonej;
- po drugie, należy oszacować poprawnoć takiego wnioskowania, to znaczy, że konieczna jest znajomość sposobu oceny, jak dużo informacji o cesze szukanej jest zawarte w cesze mierzonej, albo jak bardzo są one wzajemnie zależne.

Z tego rodzaju zagadnieniem będziemy mieli do czynienia w korelacyjnej metodzie adaptacji $\mathcal{BAG}$, która zostanie wprowadzona w Rozdziale 6, dlatego niniejszym rozdziale wprowadzimy teoretyczne podstawy wnioskowania, na których następnie oprzemy się w Rozdziale 6. Proponowane tutaj teoretyczne podejście, w pierwszej chwili wydaje się odległe temetycznie od zagadnienia adaptacji $\mathcal{BAG}$, ale w rzeczywistości doskonale się do naszych celów nadaje.

Przedstawione podejście jest nazywane **teorią łącza informacyjnego** (por. [16], Rozdział 36), (zależnie od sytuacji – ciągłego lub dyskretnego), będącą działem teorii informacji oraz statystyki. Teorię łącza stosujemy po przyjęciu założenia, że sygnałem źródłowym są wartości cechy szukanej, sygnałem odebranym – wartości cechy mierzonej, a powodem różnic występujących pomiędzy obydwoma sygnałami są losowe zakłócenia. Dodatkowo zakładamy, że wartości obu cech są sygnałami losowymi. Tak przyjęty model zależności cech badanego obiektu będziemy wykorzystywali w dalszych częściach niniejszej rozprawy.

Teoria łącza została wprowadzoną po raz pierwszy przez Shannona i Weavera w 1949 roku w pracy [1]. W samej pracy opisano wykorzystanie pojęcia entropii informacji do badania jakości łącza. Omówienie powyższych zagadnień można także znaleźć w książce Dymowskiego [3].

5.2 Statystyczny opis ciągłego łącza informacyjnego

Zakładamy, że ciągłe źródło informacji wytwarza przypadkowy sygnał ξ (**sygnał źródłowy**) o realizacjach x w zbiorze $\Omega_\xi = \{x\}$ będącym podzbiorem zbioru liczb rzeczywistych $\mathbb{R}$. Właściwości

ξ opisuje gęstość prawdopodobieństwa $f_\xi(x)$ określona na $\Omega_\xi = \{x\}$. Podobnie przyjmujemy, że na wyjściu łącza informacyjnego odbieramy przypadkowy sygnał η (**sygnał odbierany**) o realizacjach y w zbiorze $\Omega_\eta = \{y\}$, przy czym $\Omega_\eta \subset \mathbb{R}$, o właściwościach opisywanych przez gęstość prawdopodobieństwa $f_\eta(y)$ określoną na Ω_η.

Dla ustalonej realizacji x sygnału źródłowego ξ, właściwości łącza opisuje funkcja gęstości prawdopodobieństwa warunkowego $f_{\eta|\xi}(y|x)$ określona na Ω_η. Zatem właściwości łącza są opisywane przez funkcję $f_{\eta|\xi}(y|x)$ określoną na iloczynie kartezjańskim $\Omega_\xi \times \Omega_\eta$. Stąd wyznaczamy łączną gęstość prawdopodobieństwa $f_{\xi\eta}(x,y) = f_{\eta|\xi}(y|x)f_\xi(x)$ oraz drugą gęstość warunkową $f_{\xi|\eta}(x|y) = f_{\xi\eta}(x,y)/f_\eta(y)$. O sygnałach ξ i η będziemy także mówić, że tworzą one razem sygnał dwuwymiarowy (ξ,η).

Dla uproszczenia będziemy dalej przyjmowali, że dziedziną obu sygnałów jest zbiór liczb rzeczywistych, czyli $\Omega_\xi = \Omega_\eta = \mathbb{R}$.

5.3 Wykorzystanie entropii do oceny ciągłego łącza informacyjnego

Pierwszą z powszechnie stosowanych metod oceny zależności między między dwiema cechami badanego obiektu, odpowiadającej w naszym modelu jakości transmisji w ciągłym łączu informacyjnym, jest metoda polegająca na wyznaczeniu wielkości zwanej transinformacją lub informacją wzajemną. Transinformacja jak i wielkość pomocnicza – entropia, są funkcjonałami funkcji gęstości prawdopodobieństwa. Definiuje się je następująco:

- **entropia sygnału źródłowego**, czyli oczekiwana ilość informacji zawarta w przypadkowym sygnale ξ:

$$H(\xi) = -\int_{-\infty}^{\infty} f_\xi(x)\log_2 f_\xi(x)dx; \tag{5.1}$$

- **entropia sygnału odebranego**, czyli oczekiwana ilość informacji zawarta w przypadkowym sygnale η:

$$H(\eta) = -\int_{-\infty}^{\infty} f_\eta(y)\log_2 f_\eta(y)dy; \tag{5.2}$$

- **entropia warunkowa a posteriori**, czyli oczekiwana ilość informacji tracona w łączu na skutek zakłóceń:

$$H(\xi|\eta) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{\xi\eta}(x,y)\log_2 f_{\xi|\eta}(x|y)dx\,dy; \tag{5.3}$$

- **entropia warunkowa**:

$$H(\eta|\xi) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{\xi\eta}(x,y)\log_2 f_{\eta|\xi}(y|x)dx\,dy; \tag{5.4}$$

- **entropia łączna**:

$$\begin{aligned} H(\xi,\eta) &= H(\xi|\eta) + H(\eta) = H(\eta|\xi) + H(\xi) = \\ &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{\xi\eta}(x,y)\log_2 f_{\xi\eta}(x,y)dx\,dy; \end{aligned} \tag{5.5}$$

- **transinformacja lub informacja wzajemna**, czyli miara oczekiwanej ilości informacji przesyłanej przez łącze z zakłóceniami:

$$I(\xi;\eta) = H(\xi) + H(\eta) - H(\xi,\eta) = H(\xi) - H(\xi|\eta) = H(\eta) - H(\eta|\xi) = \\ = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{\xi\eta}(x,y)\log_2\frac{f_{\xi\eta}(x,y)}{f_\xi(x)f_\eta(y)}dx\,dy; \tag{5.6}$$

Wszystkie, zdefiniowane powyżej wielkości mierzymy w jednostkach zwanych **bitami**. Dla ostatniej z powyższych wielkości prawdziwe jest następujące twierdzenie, udowodnione w [16] w Rozdziale 36.

Twierdzenie 5.1 (o transinformacji) *Transinformacja jest wielkością nieujemną dla ciągłego łącza informacyjnego i równa się zeru wtedy i tylko wtedy, gdy łącze jest przerwane, to znaczy gdy zmienne losowe ξ i η są statystycznie niezależne, czyli jest spełniony warunek*

$$\bigwedge_{(x,y)\in\Omega_\xi\times\Omega_\eta}\frac{f_{\xi\eta}(x,y)}{f_\xi(x)f_\eta(y)} = 1. \tag{5.7}$$

□

Należy zauważyć, że praktyczne zastosowanie transinformacji może w pewnych sytuacjach być utrudnione ze względu na konieczność znajomości odpowiednich funkcji gęstości, których wyznaczenie może być bardzo złożone obliczeniowo w przypadku badania tą metodą algorytmów genetycznych. Dlatego podejście to jedynie sygnalizujemy i nie będziemy go dalej wykorzystywać.

5.4 Wykorzystanie korelacji do oceny ciągłego łącza informacyjnego

Druga z powszechnie stosowanych metod oceny zależności między dwiema cechami badanego obiektu jest metoda polegająca na wyznaczeniu korelacji między tymi cechami.

Załóżmy, że sygnały przypadkowe ξ i η są zdefiniowane, wraz z opisującymi je funkcjami gęstości, jak w podrozdziale 5.2. Zdefiniujemy teraz ich parametry statystyczne, które posłużą nam do wyliczenia ich korelacji:

- **wartość oczekiwana sygnału źródlowego ξ:**

$$\mathbf{E}(\xi) = \int_{-\infty}^{\infty} x f_\xi(x)dx; \tag{5.8}$$

- **wartość oczekiwana sygnału odebranego η**: j.w. ze zmianą f_ξ na f_η;
- **odchylenie standardowe sygnału źródlowego ξ:**

$$\sigma(\xi) = \sqrt{\mathbf{E}(\xi^2) - \mathbf{E}(\xi)^2} = \sqrt{\int_{-\infty}^{\infty}(x - \mathbf{E}(\xi))^2 f_\xi(x)dx}; \tag{5.9}$$

- **odchylenie standardowe sygnału odebranego η**: j.w. ze zmianą f_ξ na f_η;
- **wartość oczekiwana iloczynu sygnału źródłowego i odebranego**

$$\mathbf{E}(\xi\eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy f_\xi(x)f_\eta(y)dx\,dy; \tag{5.10}$$

- korelacja sygnałow ξ i η:

$$\varrho(\xi,\eta) = \frac{\mathbf{E}(\xi\eta) - \mathbf{E}(\xi)\mathbf{E}(\eta)}{\sigma(\xi)\sigma(\eta)}, \tag{5.11}$$

Włściwości korelacji dla przypadku granicznego, tzn. gdy sygnały ξ i η są zależne liniowo określa kolejne twierdzenie.

Twierdzenie 5.2 (o korelacji) *Dla dwóch dowolnych zmiennych losowych $X, Y \in \mathbb{R}$ zachodzi*

$$\mathbf{P}(Y = aX + b) \iff \varrho(X,Y) = \pm 1, \tag{5.12}$$

gdzie $a, b \in \mathbb{R}$ oraz $a \neq 0$. □

Należy zauważyć, że zarówno metoda bazująca na entropii, jak i bazująca na korelacji opierają się na pomiarze zależności zdarzeń losowych (por. wyrażenie 5.7 oraz licznik ułamka w 5.11). Obie metody jednak mają tę samą wadę: wymagają uprzedniej znajomości odpowiednich funkcji gęstości, dlatego ich praktyczne zastosowanie w dziedzinie algorytmów genetycznych jest poważnie utrudnione; złożoność obliczeniowa algorytmu znajdującego potrzebne funkcje gęstości dla sygnałów ξ i η powoduje, że stosowanie opisanych metod do oszacowania cechy szukanej jest nieopłacalne. Dlatego w zastosowaniach praktycznych stosuje się metodę opisaną w następnym podrozdziale.

5.5 Wyznaczanie korelacji jako statystyki z próby

5.5.1 Reprezentacja sygnału przez próbę losową

Zagadnienia wykorzystne w tym punkcie są szeroko opisane w pracach dotyczących statystyk z próby oraz wnioskowania statystycznego, między innymi w [16] w Rozdziale 36 oraz w [27].

Statystyki z próby stosujemy z reguły wtedy, gdy zależy nam jedynie na ocenie pewnych właściwości nieznanego rozkładu prawdopodobieństwa badanego sygnału. Zbiór realizacji tego sygnału nazywamy **populacją statystyczną** albo **populacją generalną**. W sytuacji, gdy badany sygnał ma bardzo dużą albo nieskończoą liczbę realizacji, wybieramy z niego próbę losową, którą badamy, a następnie wyniki badań uogólniamy dla całego sygnału. Ten rodzaj badań jest nazywany **badaniami reprezentacyjnymi**.

Definicja 5.1 *Dla dowolnego sygnału losowego χ o wartościach z Ω_χ, n-elementówą* **próbą losową prostą** *z χ nazywamy wektor losowy*

$$\mathbf{X} = (X_1, \ldots, X_n) \in \mathcal{X}, \tag{5.13}$$

gdzie $X_1, \ldots, X_n \in \Omega_\chi$ są niezależnymi zmiennymi losowymi, z których każda ma rozkład określony funkcją gestości badanego sygnału, czyli $f_\chi(x)$, a zbiór $\mathcal{X}$ jest nazywany **przestrzenią prób**. □

Definicja 5.2 Realizacją próby $\mathbf{X}$ *nazywamy wektor $\mathbf{x} = (x_1, \ldots, x_n)$ o składowych z Ω_χ, będących realizacjami zmiennych losowych $X_1, \ldots, X_n$. Wszystkie możliwe wektory $\mathbf{x}$ są punktami w przestrzeni prób.* □

Definicja 5.3 Statystyką z próby *n-elementowej nazywamy dowolną funkcję $Z_n = g(\mathbf{X})$ określoną na przestrzeni prób $\mathcal{X}$. Statystyka Z_n jest zmienną losową o wartościach $z_n = g(\mathbf{x})$.* □

Szczególnie interesujące są statystyki będące estymatorami paramterów rozkładu prawdopodobieństwa badanego sygnału. Najczęściej stosowane z nich to średnia arytmetyczna z próby, wariancja z próby oraz współczynnik korelacji z próby.

Aby móc na podstawie statystyki wnioskować o badanej populacji generalnej, konieczne jest wyznaczenie jej rozkładu prawdopodobieństwa oraz przedziału ufności. Należy zwrócić baczną uwagę na to, że statystyki z próby mogą poprawnie estymować parametry rozkładu populacji generalnej jedynie wtedy, gdy badana próba jest rzeczywiście losowa, to znaczy prawdopodobieństwo znalezienia się w próbie było identyczne dla każdego elementu populacji generalnej (patrz [16]).

5.5.2 Estymacja współczynnika korelacji sygnałów

Mając dwa sygnały, czyli sygnał dwuwymiarowy, tworzący dwuwymiarową populację generalną, możemy w poniższy sposób znaleźć estymację ich współczynnika korelacji.

Twierdzenie 5.3 *Niech wektory losowe* $\mathbf{X} = (X_1, \ldots, X_n) \in \mathcal{X}$ *i* $\mathbf{Y} = (Y_1, \ldots, Y_n) \in \mathcal{Y}$ *będą prostymi próbami losowymi sygnałow losowych* ξ *i* η*, zdefiniowanych jak w Podrozdziale 5.2. Wtedy statystyka*

$$R(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{k=1}^{n} (X_k - \overline{X})(Y_k - \overline{Y})}{\sqrt{\sum_{k=1}^{n} (X_k - \overline{X})^2} \sqrt{\sum_{k=1}^{n} (Y_k - \overline{Y})^2}} \tag{5.14}$$

jest nieobciążonym i zgodnym estymatorem współczynnika korelacji $\varrho(\xi, \eta)$ *zdefiniowanego przez (5.11), gdzie* $\overline{X} = \frac{1}{n} \sum_{k=1}^{n} X_k$ *i* $\overline{Y} = \frac{1}{n} \sum_{k=1}^{n} Y_k$. □

Wartość statystyki $R(\mathbf{X}, \mathbf{Y})$ na podstawie realizacji próby $\mathbf{x} = (x_1, \ldots, x_n)$ i $\mathbf{y} = (y_1, \ldots, y_n)$ obliczamy ze wzoru

$$r(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{n} (x_k - \overline{x})(y_k - \overline{y})}{\sqrt{\sum_{k=1}^{n} (x_k - \overline{x})^2} \sqrt{\sum_{k=1}^{n} (y_k - \overline{y})^2}}. \tag{5.15}$$

W przypadku, gdy dysponujemy dostatecznie liczną próbą losową (przyjmuje się, że $n \geq 50$), możemy wykorzystać poniższe twierdzenie do utworzenia przedziału ufności dla statystyki R.

Twierdzenie 5.4 *Dla dowolnego rozkładu sygnałów* ξ *i* η *statystyka* $R(\mathbf{X}, \mathbf{Y})$ *ma aymptotyczny rozkład normalny* $N(\varrho, \frac{1-\varrho^2}{\sqrt{n}})$*, natomiast przedział ufności dla* ϱ *jest następujący:*

$$\mathbf{P}\left(R + u_{\frac{\alpha}{2}} \frac{1 - R^2}{\sqrt{n}} < \varrho < R + u_{1-\frac{\alpha}{2}} \frac{1 - R^2}{\sqrt{n}}\right) = 1 - \alpha, \tag{5.16}$$

gdzie u_t *oznacza, dla dowolnego* t*, liczbę odczytaną z tablicy dystrybuanty rozkładu normalnego* $N(0, 1)$ *taką, że* $F(u_t) = t$*, gdzie* $F(\cdot)$ *oznacza dystrybuantę rozkładu normalnego.* □

Uwaga 5.1 (o korelacji) *Niezależnie od statystycznej interpretacji wzoru (5.15), dla dowolnych wektorów* $\mathbf{x}$, $\mathbf{y}$ *może być on wykorzystywany również jako miara podobieństwa tych wektorów. Można udowodnić, że miara ta jest niezmiennicza względem odwzorowania liniowego, tzn.*

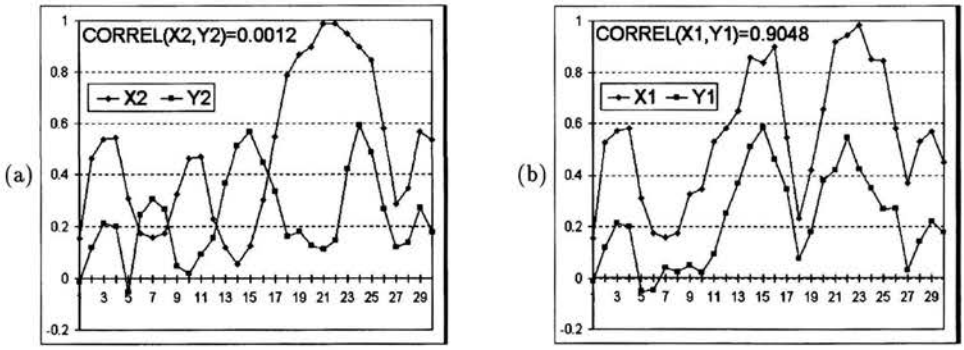
$$r(\mathbf{x}, \mathbf{y}) = r(a_x \mathbf{x} + b_x, a_y \mathbf{y} + b_y), \tag{5.17}$$

dla $a_x, a_y, b_x, b_y \in \mathbb{R}$*, przy czym* $a_x a_y > 0$. □

5.6 Wnioskowanie o sygnale źródlowym na podstawie sygnału odebranego

Dotychczas omówiliśmy sposoby oceniania zależności między sygnałem źródlowym a odebranym. W tym podrozdziale zajmiemy się wnioskowaniem o sygnale źródlowym na podstawie sygnału odebranego. Będziemy wykorzystywali podstawowe zależności rachunku prawdopodobieństwa, które czytelnik może znaleść na przykład w pracy [16] w Rozdziale 36.

Załóżmy że sygnały losowe ξ i η są zdefiniowane jak w Podrozdziale 5.2. Niech X_1, X_2 będą niezależnymi zmiennymi losowymi o gęstości $f_\xi(x)$, Y – zmienną losową o gęstości $f_\eta(y)$, a $Y_1 =$



Rys. 5.1: Przykład przebiegów nieskorelowanych (a) i przebiegów silnie skorelowanych (b).

$Y|x_1$ i $Y_2 = Y|x_2$ – niezależnymi zmiennymi o gęstościach warunkowych odpowiednio $f_{\eta|\xi}(y_1|x_1)$ i $f_{\eta|\xi}(y_2|x_2)$. Znajdźmy prawdopodobieństwo, że następujące dwa wnioski (implikacje) są prawdziwe:

$$X_1 \le X_2 \Rightarrow Y_1 \le Y_2 \quad \text{oraz} \quad X_1 > X_2 \Rightarrow Y_1 > Y_2. \tag{5.18}$$

Otrzymamy następujący iloczyn prawdopodobieństw warunkowych:

$$\begin{aligned} \mathbf{P}(A) &= \mathbf{P}\left((Y_1 \le Y_2 | X_1 \le X_2) \wedge (Y_1 > Y_2 | X_1 > X_2)\right) = \\ &= \mathbf{P}\left((Y_1 \le Y_2 \Leftarrow X_1 \le X_2) \wedge (Y_1 > Y_2 \Leftarrow X_1 > X_2)\right), \end{aligned} \tag{5.19}$$

skąd, po zastosowaniu tożsamości boolowskiej $(p \Leftarrow q) \wedge (\neg p \Leftarrow \neg q) \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$, otrzymamy

$$\begin{aligned} \mathbf{P}(A) &= \mathbf{P}\left((Y_1 \le Y_2 \wedge X_1 \le X_2) \vee (Y_1 > Y_2 \wedge X_1 > X_2)\right) = \\ &= \mathbf{P}(Y_1 \le Y_2 \wedge X_1 \le X_2) + \mathbf{P}(Y_1 > Y_2 \wedge X_1 > X_2), \end{aligned} \tag{5.20}$$

gdzie zajście zdarzenia $A = (Y_1 \le Y_2 \wedge X_1 \le X_2) \vee (Y_1 > Y_2 \wedge X_1 > X_2)$ odpowiada wnioskowaniu poprawnemu, a zajście zdarzenia przeciwnego $\neg A$ – wnioskowaniu błędnemu. Zauważmy, że skoro X_1, X_2 są niezależne, ciągłe i mają taki sam rozkład, to

$$\mathbf{P}(X_1 \le X_2) = \mathbf{P}(X_1 > X_2) = \frac{1}{2}. \tag{5.21}$$

Zakładając, że łączna gęstość prawdopodobieństwa przy niezależności odpowiednio X_1 i X_2 oraz Y_1 i Y_2 jest równa

$$f(x_1, x_2, y_1, y_2) = f_{\xi\eta}(x_1, y_1)\, f_{\xi\eta}(x_2, y_2) = f_{\eta|\xi}(y_1|x_1)\, f_{\xi}(x_1)\, f_{\eta|\xi}(y_2|x_2)\, f_{\xi}(x_2), \tag{5.22}$$

otrzymujemy:

$$\mathbf{P}(A) = \underset{A}{\iiint\!\!\iint} f(x_1, x_2, y_1, y_2)\, dx_1\, dx_2\, dy_1\, dy_2. \tag{5.23}$$

W celu uproszczenia obliczeń wprowadzamy nowe zmienne losowe $Z_X = X_2 - X_1$ oraz $Z_Y = Y_2 - Y_1$, zauważając, że $\mathbf{E}(Z_X) = 0$ oraz funkcja gęstości Z_X jest zawsze symetryczna względem 0, czyli dystrybuanta $F_{Z_X}(0) = \frac{1}{2}$. Nowe zmienne pozwalają zdefiniować obszar A w prostszy sposób:

$$A = (Z_X \ge 0 \wedge Z_Y \ge 0) \vee (Z_X < 0 \wedge Z_Y < 0) = Z_X Z_Y \ge 0. \tag{5.24}$$

Następnie wyznaczamy nowe funkcje gęstości:

$$f_{Z_X}(z_x) = \int_{-\infty}^{+\infty} f_\xi(x_2)\, f_\xi(x_2 - z_x)\, dx_2, \tag{5.25}$$

$$f_{Z_Y|X_1X_2}(z_y|x_1, x_2) = \int_{-\infty}^{+\infty} f_{\eta|\xi}(y_2|x_2)\, f_{\eta|\xi}(y_2 - z_y|x_1)\, dy_2, \tag{5.26}$$

$$f_{X_1X_2Z_Y}(x_1, x_2, z_y) = f_{Z_Y|X_1X_2}(z_y|x_1, x_2)\, f_\xi(x_1)\, f_\xi(x_2), \tag{5.27}$$

$$\begin{aligned} f_{Z_XZ_Y}(z_x, z_y) &= f_{Z_Y|Z_X}(z_y|z_x)\, f_{Z_X}(z_x) = \\ &= \int_{-\infty}^{+\infty} f_{X_1X_2Z_Y}(x_2 - z_x, x_2, z_y)\, dx_2 = \\ &= \int_{-\infty}^{+\infty} f_{Z_Y|X_1X_2}(z_y|x_2 - z_x, x_2)\, f_\xi(x_2 - z_x)\, f_\xi(x_2)\, dx_2, \end{aligned} \tag{5.28}$$

stąd mamy:

$$\begin{aligned} \mathbf{P}(Z_X \geq 0 \wedge Z_Y \geq 0) &= \int_0^{+\infty}\int_0^{+\infty} f_{Z_XZ_Y}(z_x, z_y)\, dz_x\, dz_y = \\ &= 1 - F_{Z_XZ_Y}(0, +\infty) - F_{Z_XZ_Y}(+\infty, 0) + F_{Z_XZ_Y}(0, 0), \end{aligned} \tag{5.29}$$

oraz

$$\mathbf{P}(Z_X < 0 \wedge Z_Y < 0) = \int_{-\infty}^{0}\int_{-\infty}^{0} f_{Z_XZ_Y}(z_x, z_y)\, dz_x\, dz_y = F_{Z_XZ_Y}(0, 0). \tag{5.30}$$

Na tej podstawie wyznaczamy nową postać wzoru (5.23):

$$\begin{aligned} \mathbf{P}(A) &= \mathbf{P}(Z_X \geq 0 \wedge Z_Y \geq 0) + \mathbf{P}(Z_X < 0 \wedge Z_Y < 0) = \\ &= 1 - F_{Z_XZ_Y}(0, +\infty) - F_{Z_XZ_Y}(+\infty, 0) + 2F_{Z_XZ_Y}(0, 0) \end{aligned} \tag{5.31}$$

Rozpatrzmy następujące przypadki szczególne:

- Sygnały ξ i η są niezależne, stąd Z_X i Z_Y są również niezależne, czyli $F_{Z_XZ_Y}(z_x, z_y) = F_{Z_X}(z_x)\, F_{Z_Y}(z_y)$, a także $\varrho(\xi, \eta) = 0$; z wyrażenia (5.24) otrzymujemy wtedy

$$\begin{aligned} \mathbf{P}(A) &= 1 - F_{Z_X}(0)\, F_{Z_Y}(+\infty) - F_{Z_X}(+\infty)\, F_{Z_Y}(0) + 2F_{Z_X}(0)\, F_{Z_Y}(0) = \\ &= 1 - \frac{1}{2} - \frac{1}{2} + \frac{1}{2} = \frac{1}{2}. \end{aligned} \tag{5.32}$$

  Widać, że obie możliwości są jednakowo prawdopodobne, czyli nie możemy wnioskować o η znając ξ.

- Sygnały ξ i η są zależne liniowo, stąd mamy $Y_1 = aX_1 + b$, $Y_2 = aX_2 + b$ oraz $Z_Y = aZ_X$, a także $\varrho(\xi, \eta) = \pm 1$; analizując wyrażenie (5.24) otrzymujemy wtedy

$$\mathbf{P}(A) = \begin{cases} 1, & \text{gdy } a \geq 0; \\ 0, & \text{wpp.} \end{cases} \tag{5.33}$$

  Stąd wynika, że nasze wnioskowanie o η będzie jednoznaczne, przy założeniu znajomości ξ.

Za wyjątkiem przedstawionych przypadków szczególnych wyliczenie wartości wyrażenia (5.31) jest trudne, gdyż wymaga wyznaczenia odpowiednich dystrybuant, dlatego może być ono wykorzystywane tylko do analiz teoretycznych. W praktycznych zastosowaniach prawdziwość omawianego wnioskowania będziemy oceniali pośrednio, na podstawie współczynika korelacji obu badanych sygnałów, wyznaczonego z realizacji próby losowej, zgodnie z (5.15).

Rozdział 6

Nowa metoda adaptacji off-line $\mathcal{BAG}$

6.1 Korelacyjne metody adaptacji off-line

6.1.1 Zagadnienie adaptacji off-line

W poprzednim rozdziale omówiono grupę metod adaptacji off-line wykorzystujących odrębną metodę analizy funkcji przystosowania. Celem adaptacji jest w tym przypadku znalezienie dla danego problemu, na podstawie jego pomiarów, takich wartości parametrów $\mathcal{BAG}$, dla których sprawność $\mathcal{BAG}$przy rozwiązywaniu tego problemu będzie możliwie najwyższa. Podczas adaptacji zwykle nie modyfikujemy wszystkich parametrów $\mathcal{BAG}$, ale jedynie pewien ich podzbiów, natomiast pozostałe parametry pozostawiamy niezmienne. Dlatego istotnym problemem jest optymalny wybór adaptowanych parametrów. Dla zachowania ogólności, w niniejszym rozdziale będziemy posługiwali się uogólnionym adaptowanym parametrem o wartościach ze zbioru P – bez wnikania w jego postać i właściwości.

W każdej metodzie adaptacji jednym z ważniejszych zagadnień jest wybór metody oceny sprawności $\mathcal{BAG}$dla danego zestawu wartości adaptowanych parametrów. Zakładając, że zdefiniowalismy pojęcie sprawności oraz, że sprawność daje się wyrazić w postaci skalarnej wartości należącej do zbioru liczb rzeczywistych, możemy zdefiniować pojęcie funkcji sprawności.

Definicja 6.1 *Niech P będzie skończonym zbiorem wartości adaptowanych parametrów $\mathcal{BAG}$. Niech $f \in \mathcal{F}$ oznacza rozwiązywany problem.* **Funkcją sprawności** *$\mathcal{BAG}$nazwiemy taką funkcję*

$$\mathcal{E}_f : P \to \mathbb{R},$$

dla której ze spełnienia warunku $\mathcal{E}_f(p_1) > \mathcal{E}_f(p_2)$, gdzie $p_1, p_2 \in P$, wynika, że algorytm optymalizacji rozwiązuje problem f statystycznie sprawniej dla wartości parametru p_1 niż dla p_2. □

Przykład 6.1 *W niniejszej pracy przyjęto, że sprawność algorytmu optymalizacji jest określona poprzez wartość wyrażenia:*

$$\mathcal{E}_f(p) \equiv -\mathbf{E}(G_f(p)) \equiv -\mathcal{G}_f(p) \tag{6.1}$$

gdzie $\mathbf{E}(\cdot)$ jest wartością oczekiwaną, a $G_f(p) \in \mathbb{R}_+$ jest zmienną losową której wartościami są liczby generacji potrzebnych do znalezienia optimum globalnego przez algorytm optymalizacji dla wartości parametru $p \in P$, przy czym zdarzeniami losowymi są uruchomienia algorytmu optymalizacji. Wartość $\mathbf{E}(G_f(p)) = \mathcal{G}_f(p)$ jest przybliżana poprzez średnią liczbę generacji dla wielokrotnego uruchomienia algorytmu. □

Zapiszmy problem adaptacji off-line w bardziej formalny sposób. Jak wspomniano wcześniej celem adaptacji jest dobór takiej metody pomiarowej, zapisywanej jako funkcja

$$\mathcal{M} : \mathcal{F} \to M, \tag{6.2}$$

i takiej metody wyznaczania wartości adaptowanych parametrów, zapisywanej jako funkcja

$$\mathcal{P} : M \to P, \tag{6.3}$$

że spełniona jest implikacja

$$p^* = \mathcal{P}(\mathcal{M}(f)) \Longrightarrow \mathcal{E}_f(p^*) = \max_{p \in P} \mathcal{E}_f(p), \tag{6.4}$$

gdzie M jest przestrzenią wyników pomiarów, a P jest zbiorem wartości adaptowanego parametru. Oczywiście akceptujemy również sytuację, gdy p^* oznacza pewien podzbiór wartości optymalnych, a nie pojedyńczą wartość.

Odwzorowania $\mathcal{M}$ i $\mathcal{P}$ powinny być uniwersalne, czyli powinny pozwalać na wyznaczenie prawidłowych wartości $p^* \in P$ dla możliwie dużej liczby klas problemów $f \in \mathcal{F}$.

W ogólnym przypadku rozwiązanie tak postawionego problemu jest niezwykle trudne. Na tą trudność składają się następujące powody:

- istnieje ogromna liczba możliwości pomiaru funkcji przystosowania, czyli zarówno zdefiniowania odwzorowania $\mathcal{M}$, jak i semantyki przestrzeni M;
- w przypadku problemów rzeczywistych nie jest znane optimum globalne, w związku z tym stosowane metody pomiarowe mogą wykorzystywać wyłącznie statystyczne testy całej przestrzeni rozwiązań $\mathcal{B}$;
- trudności ze znalezieniem odwzorowania $\mathcal{P}$, czyli z określeniem charakteru powiązania między wynikami pomiarów a odpowiednimi dla danego problemu wartościami parametrów.

Wydaje się, ze względu na wymienione trudności, że znalezienie metody analitycznego wyznaczania optymalnej wartości adaptowanego parametru na podstawie pomiarów funkcji przystosowania długo jeszcze nie będzie możliwe, dlatego w tej dziedzinie duże nadzieje pokłada się w metodach adaptacji.

W celu ułatwienia implementacji metody adaptacji przyjmujemy pewne założenia upraszczające. Zakładamy mianowicie, że dysponujemy stałym i niezależnym od problemu zestawem metod pomiarowych opisywanych przez odwzorowanie $\mathcal{M}$ i na podstawie otrzymanych wyników pomiarów adaptujemy jedynie odwzorowanie $\mathcal{P}$.

Przykładem zastosowania powyższego rozumowania w praktyce jest, wprowadzona w następnych podrozdziałach, nowa klasa metod adaptacji off-line parametrów $\mathcal{BAG}$, nazywana korelacyjnymi metodami adaptacji.

6.1.2 Idea korelacyjnej metody adaptacji

Załóżmy, że sprawność algorytmu optymalizacji $\mathcal{E}_f$ jest określona jak w Definicji 6.1. Zdefiniujmy nową metodę pomiarową zgodną ze wzorem (6.2). Załóżmy, że przestrzenią wyników pomiarów jest wektor o długości równej liczbie elementów P, czyli $M = \mathbb{R}^{|P|}$. Kolejne składowe tego wektora niech będą wyliczane następująco:

$$\mathcal{M}(f) \equiv \Big(m(f, p_1), \ldots, m(f, p_{|P|})\Big) = \Big(m_f(p_1), \ldots, m_f(p_{|P|})\Big), \tag{6.5}$$

przy czym p_k dla $k = 1, \ldots, |P|$ przyjmuje kolejno wartości wszystkich elementów P, czyli $p_i \neq p_j$ dla $i \neq j$, a funkcja m jest typu $\mathcal{F} \times P \to \mathbb{R}$. Dla ustalonego $f \in \mathcal{F}$ będziemy funkcję m zapisywali w postaci skróconej jako

$$m_f : P \to \mathbb{R} \tag{6.6}$$

i będziemy nazywali **funkcją pomiarową**. Przykłady doświadczalnie sprawdzonych funkcji pomiarowych zostaną przedstawione w Rozdziale 7.

Widać, że typ funkcji (6.6) jest identyczny z typem funkcji sprawności (6.1). Załóżmy, że $m_f \equiv \mathcal{E}_f$, to znaczy, ze wartością funkcji m_f jest rzeczywista sprawność algorytmu. Aby znaleźć optymalną wartość parametru p^* zgodną ze wzorem (6.4), wystarczyłoby wziąść metodę $\mathcal{P}$ zgodną ze wzorem (6.3) i zdefiniowaną następująco:

$$\mathcal{P}\left(m_1, \ldots, m_{|P|}\right) = m_f^{-1}\left(\max_i m_i\right), \tag{6.7}$$

dzięki czemu moglibyśmy lewą stronę implikacji (6.4) zapisać w równoważnej postaci jako

$$p^* = \mathcal{P}\left(m_f(p_1), \ldots, m_f(p_{|P|})\right) = m_f^{-1}\left(\max_{p \in P} \mathcal{E}_f(p)\right). \tag{6.8}$$

Podobnie jak we wzorze (6.4), w przypadku gdy funkcja m_f nie jest bijekcją możemy w rezultacie otrzymać zamiast jednej wartości, zbiór optymalnych wartości parametru.

Widać, że przyjęcie zaproponowanych założeń pozwala interpretować odwzorowanie $\mathcal{P}$ jako metodę adaptacji (w tym przypadku polegającą na sprawdzeniu wszystkich możliwych wartości parametru), natomiast funkcję pomiarową m_f można traktować jako metodę oszacowania sprawności $\mathcal{BAG}$. W tej sytuacji nasuwa się pytanie, czy możliwe jest znalezienie funkcji m_f, której obliczenie nie wymaga uruchomienia algorytmu optymalizacji, i którą możnaby wykorzystać do oszacowania sprawności algorytmu optymalizacji $\mathcal{E}_f$. Okazuje się, że w pewnych sytuacjach odpowiedź na to pytanie jest pozytywna. Przykładem metod adaptacji wykorzystujących takie oszacowanie są korelacyjne metody adaptacji.

6.1.3 Funkcja pomiarowa jako oszacowanie funkcji sprawności

Zauważmy, że możemy, korzystając z funkcji sprawności $\mathcal{E}_f$, zdefiniować odwzorowanie $\mathcal{E}$ typu $\mathcal{F} \to \mathbb{R}^{|P|}$, analogiczne do (6.5), w następujący sposób:

$$\mathcal{E}(f) \equiv \left(\mathcal{E}_f(p_1), \ldots, \mathcal{E}_f(p_{|P|})\right), \tag{6.9}$$

przy czym wartości p_k są identyczne jak w (6.2).

Widać, że wektory $\mathcal{M}(f)$ i $\mathcal{E}(f)$ są wykresami odpowiednio m_f i $\mathcal{E}_f$ w funkcji $k \in \{1, \ldots, |P|\}$. Jeżeli teraz znajdziemy taką funkcję m_f, której wykres będzie wystarczająco podobny do wykresu $\mathcal{E}_f$, to będziemy mogli ją wykorzystać do przybliżonego wnioskowania o $\mathcal{E}_f$ przy użyciu następujących wnioskowań:

$$m_f(p_1) \leq m_f(p_2) \longrightarrow \mathcal{E}_f(p_1) \leq \mathcal{E}_f(p_2) \quad \text{oraz} \quad m_f(p_1) > m_f(p_2) \longrightarrow \mathcal{E}_f(p_1) > \mathcal{E}_f(p_2), \tag{6.10}$$

gdzie $p_1, p_2 \in P$. Oczywiście im podobieństwo to będzie większe, tym większe będzie prawdopodobieństwo poprawności takiego wnioskowania.

Aby móc porównywać różne funkcji m_f musimy mieć możliwość zmierzenia, w jakim stopniu dwa wykresy są do siebie podobne. Możemy w tym celu wykorzystać narzędzia, które dostarcza nam statystyka matematyczna. Jeżeli przyjmiemy, że wektory $\mathcal{M}(f)$ oraz $\mathcal{E}(f)$ są zbiorami realizacji dwóch losowych sygnałów (patrz Podrozdział 5.2), tworzących razem dwuwymiarową populację generalną, to po wyznaczeniu odpowiednich funkcji gęstości, jako miarę ich podobieństwa możemy przjąć korelację obu sygnałów (patrz Podrozdział 5.4).

Wiedząc, że oba wektory są skorelowane w wystarczającym stopniu możemy, traktując wartość funkcji sprawności jako sygnał źródłowy, a wartość funkcji pomiarowej – jako sygnał odebrany, zastosować wnioskowanie opisane w Podrozdziale 5.6. W tym przypadku implikacje (5.18) przybiorą postać (6.10).

6.1.4 Implementacja korelacyjnej metody adaptacji

Obliczenie korelacji wektorów $\mathcal{M}(f)$ i $\mathcal{E}(f)$ dla wszystkich elementów zbioru P zwykle nie jest praktycznie możliwe, ze względu na dużą liczebność tego zbioru. Dlatego implementując metodę wyznaczamy korelację wektorów złozonych z elementów pewnego podzbioru Q zbioru P takiego, że $|Q| \ll |P|$, czyli, inaczej mówiąc, wyznaczamy korelację jako statystykę z próby losowej (patrz Podrozdział 5.5).

Definicja 6.2 (reprezentatywności) *Podzbioru $Q \subset P$ jest reprezentatywnym podzbiorem P, co oznacza, że charakter zależności między wartościami funkcji $m_f(q)$ i $\mathcal{E}_f(q)$ jest dla $q \in Q$ taki sam jak dla całego zbioru P.* □

Reprezentatywność Q dla P oznacza w naszym przypadku, że korelacja wyznaczona jako statystyka z próby losowej Q jest wystarczająco dobrym oszacowaniem korelacji dwuwymiarowej populacji generalnej P. Oczywiście nie każdy Q będzie reprezentatywny dla P. Sytuacja taka może zajść wtedy, gdy podzbiór Q zostanie wybrany niezupełnie losowo, albo wręcz tendencyjnie (patrz Podrozdział 5.5).

Definicja 6.3 Korelacyjną metodą adaptacji *będziemy nazywali każdą taką metodę adaptacji off-line, która zamiast bezpośredniego porównywania sprawności algorytmu optymalizacji dla różnych wartości parametru $p \in P$, wykorzystuje implikacje (6.10), przy czym funkcja pomiarowa m_f jest tak dobrana, że wektory $\left(m_f(q_1), \ldots, m_f(q_{|Q|})\right)$ i $\left(\mathcal{E}_f(q_1), \ldots, \mathcal{E}_f(q_{|Q|})\right)$ są (dodatnio lub ujemnie) skorelowane, gdzie $q_l \in Q \subset P$ dla $l = 1, \ldots, |Q|$, oraz $q_i \neq q_j$ dla $i \neq j$.* □

Korelacyjną metodę adaptacji możemy stosować wtedy, gdy stwierdzimy, że wartości funkcji m_f i $\mathcal{E}_f$ są "wystarczająco" silnie skorelowane. Odpowiednie kryterium, polegające na przyjęciu dolnego progu korelacji, wyznaczonej jako statystyka z próby, zaproponowano i omówiono w Podrozdziale 5.6.

Korelacja między m_f i $\mathcal{E}_f$ zwykle silnie zależy od rodzaju rozwiązywanego problemu, od rodzaju parametru adaptowanego P, a także od przyjętych wartosci pozostałych (nieadaptowanych) parametrów algorytmu optymalizacji (w przypadku $\mathcal{BAG}$mogą to byŋp. prawdopodobieństwo mutacji i rozmiar populacji). Dlatego praktyczne zastosowanie korelacyjnych metod adaptacji jest uzależnione od znalezienia wystarczająco uniwersalnych metod pomiarowych, silnie skorelowanych ze sprawnością $\mathcal{BAG}$dla wielu różnych klas funkcji przystosowania. Dodatkowo, aby stosowanie tych metod miało sens, muszą one mieć mniejszą złożoność obliczeniową niż sam algorytm optymalizacji, gdyż w przeciwnym wypadku w miejsce metody pomiarowej lepiej po prostu uruchomić algorytm optymalizacji i wyznaczyć jego sprawność bezpośrednio.

6.1.5 Przykłady zastosowań korelacyjnych metod adaptacji

Metoda "prób i błędów"

Najprostszą metodą adaptacji której wersję korelacyjną można łatwo uzyskać, jest metoda "prób i błędów" przedstawiona poglądowo w algorytmie na Rysunku 6.1. Oryginalna metoda polega na uruchamianiu algorytmu optymalizacji dla kilku losowo wybranych wartości parametru i wybraniu tej wartości, dla której uzyskano najlepsze rezultaty. W wersji korelacyjnej zamiast każdorazowego uruchamiania algorytmu optymalizacji oblicza się jedynie wartość funkcji pomiarowej dla danej wartości parametru p.

Korzyści jakie możemy odnieść w wyniku dokonanej modyfikacji sprowadzają się do tego, że próby dokonywane przez algorytm są teraz mniej złożone obliczeniowo i, w związku z czym, może być ich więcej w tym samym czasie.

start
wybierz początkową wartość p **z** P.
niech $M_{max} := 0$
while M_{max} **jest niewystarczająco duże do**
  oblicz $M := m_f(p)$
  if $M > M_{max}$ **then niech** $M_{max} := M$ **i** $p_{max} := p$
  wybierz następną wartość p **z** P
end
wykonaj $\mathcal{BAG}$ **z parametrem** p_{max}
stop

Rys. 6.1: Korelacyjna wersja metody "prób i błędów".

Zalety metody "prób i błędów" objawiają się zwykle w sytuacji gdy istnieją pewne reguły heurystyczne doboru właściwej wartości adaptowanego parametru. Wtedy dobór kolejnych wartości parametrów odbywa się nie "na ślepo" ale zgodnie z tymi regułami. Reguły takie mogą przykładowo preferować albo wykluczać pewne obszary przestrzeni P, ale mogą także opisywać algorytm wyboru kolejnej wartości parametru na podstawie dotychczasowych doświadczeń.

Metaalgorytm genetyczny

Opisana powyżej prosta metoda "prób i błędów" nie pozwala na sprawne przeszukiwanie przestrzeni parametrów P w przypadku ogólnym, to znaczy gdy nie są znane żadne reguły poszukiwań. Ze względu na fakt, że przestrzeń parametrów P ma zwykle bardzo duże rozmiary porównywalne albo nawet większe niż przestrzeń kodowań $\mathcal{B}$, jest rzeczą oczywistą, że do przeszukiwaną przestrzeni parametrów nadaje się doskonale binarny algorytm genetyczny, który będzie dla uniknięcia niejednoznaczności nazywany metaalgorytmem, w skrócie meta$\mathcal{BAG}$.

Aby meta$\mathcal{BAG}$przeszukujący przestrzeń P mógł działać wystarczająco sprawnie, ograniczamy się zwykle do pewnego odpowiednio dużego podzbioru P, przyjmując, że będziemy poszukiwali optymalnej wartości parametru wyłącznie wśród elementów tego podzbioru. Tego typu ograniczenie może być również wymuszone przez sam sposób kodowania elementów P.

Szkielet korelacyjnej metody adaptacji wykorzystującej metaalgorytm genetyczny pokazano na Rysunku 6.2. Należy zwrócić uwagę, że rezygnujemy tutaj ze znalezienia optymalnej wartości parametru i zadowalamy się poprawą uzyskaną w wyniku wygenerowania zadanej liczby pokoleń przez meta$\mathcal{BAG}$. W rzeczywistości nie zależy nam na znalezieniu optymalnej wartości parametru, a jedynie na poprawie działania głównego algorytmu genetycznego. Stąd zakładamy, że jeżeli nie udało

start
wylosuj początkową populację wartości parametru z P
wykonaj W **pokoleń meta**$\mathcal{BAG}$ **dla funkcji przystosowania** $f(p) = m_f(p)$
niech p_{max} **będzie najlepszym osobnikiem w** W**-tym pokoleniu**
wykonaj $\mathcal{BAG}$ **z parametrem** p_{max}
stop

Rys. 6.2: Korelacyjna metoda adaptacji wykorzystująca metaalgorytm genetyczny (W – ustalony parametr).

się uzyskać poprawy działania w z góry zadanej liczbie W pokoleń, to jest mało prawdopodobne, że taka poprawa byłaby możliwa w następnych pokoleniach.

6.2 Kodowanie z permutacją

6.2.1 Wprowadzenie pojęcia permutacji

Pojęcie permutacji jest wprowadzane niezależnie w dwóch działach matematyki: w kombinatoryce oraz w algebrze abstrakcyjnej, a zwłaszcza w teorii grup permutacji, dlatego przytoczymy poniżej obie definicje. W niniejszej rozprawie będziemy się zajmowali wyłącznie teorią skończonych grup permutacji. Szczegółowe omówienie powyższej teorii może czytelnik znaleźć w pracach [2] oraz [33]. Ponadto w pracy Stadlera [30] omówiono teorię grup permutacji w kontekście wykorzystania jej w badaniu krajobrazów przystosowania.

Definicja 6.4 (kombinatoryczna) *Niech A będzie dowolnym zbiorem składającym się z N elementów.* **Permutacją (bez powtórzeń)** *elementów zbioru A nazywamy każdy ciąg N-wyrazowy $(a_1, \ldots, a_N)$, którego wyrazy są różnymi elementami zbioru A.* □

Definicja 6.5 (z algebry abstrakcyjnej)

1. **Permutacją** *skończonego zbioru A nazywamy każde jednoznaczne odwzowrowanie (bijekcję) zbioru A na siebie.*
2. *Zbiór wszystkich permutacji skończonego zbioru A tworzy grupę ze względu na składanie odwzorowań. Nazywamy ją* **grupą symetrii zbioru** *A i oznaczamy przez $Sym(A)$. Elementem jednostkowym tej grupy jest permutacja odwzorowująca każdy element A na siebie. Dodatkowo mamy takie $\pi, \pi^{-1} \in Sym(A)$, że dla każdego $a, b \in A$ zachodzi $b = \pi(a) \iff a = \pi^{-1}(b)$.*

□

Każda permutacja $\pi : A \to A$ może być zapisana na dwa sposoby. Pierwszy sposób, wynikający wprost z definicji, jest następujący:

$$\pi = \begin{pmatrix} a_1 & a_2 & \ldots & a_N \\ x(a_1) & x(a_2) & \ldots & x(a_N) \end{pmatrix}. \tag{6.11}$$

Drugi sposób nazywany jest **postacią cykliczną**. Jego idea wynika stąd, że każdą permutację można jednoznacznie przedstawić w postaci złożenia rozłącznych cykli (podgrup cyklicznych).

Definicja 6.6 *Permutację $\pi \in Sym(A)$ nazywamy* **cyklem** *rzędu s jeśli dla s różnych elementów $\alpha_1, \alpha_2, \ldots, \alpha_s \in A$ permutacja π odwzorowuje α_i w α_{i+1} dla $i = 1, \ldots, s-1$ oraz α_s w α_1, natomiast pozostałe elementy A pozostawia niezmienione.* **Rozłączność** *cykli oznacza, że żadne dwa cykle nie oddziaływują na te same elementy A.* □

Porównanie obu sposobów zapisu pokazuje następny przykład.

Przykład 6.2 *(z książki [33])*
Niech $A = \{0, 1, 2, 3, 4, 5, 6\}$ wtedy odwzorowanie $x \mapsto (4x+1) \bmod 7$ definiuje permutację π zbioru A, która może być zapisana w postaci:

$$\pi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 5 & 2 & 6 & 3 & 0 & 4 \end{pmatrix}, \tag{6.12}$$

albo w postaci rozłącznych cykli

$$\pi = (015)(2)(364) = (2)(015)(643) = \ldots = (015)(364). \tag{6.13}$$

Ostatnie wyrażenie pokazuje postać uproszczoną, w której pominięto cykle jednoelementowe. □

Definicja 6.7 (grupy permutacji) *Niech będzie $G \subseteq Sym(A)$. Jeżeli $A_0 \subseteq A$, to zbiór $G^{A_0} \subseteq G$ takich permutacji zbioru A, które zachowują podzbiór A_0 (tzn. odwzorowują każdy element zbioru A_0 w inny jego element) nazywamy* **podgrupą grupy symetrii** *A albo* **grupą permutacji**. *Mówimy, że zbiór A_0 jest niezmiennikiem grupy G^{A_0}.* □

Z powyższej definicji wynika także, że zbiór $A \setminus A_0$ także jest niezmiennikiem grupy G^{A_0}. Ilustruje to poniższy przykład.

Przykład 6.3 *Niech $A = \{1,2,3,4,5\}$ oraz $A \supset A_0 = \{3,5\}$. Niech grupa permutacji $Sym(A) \supset G = \{\pi_1, \pi_2\}$ składa się z*

$$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 2 & 3 & 4 \end{pmatrix} \quad oraz \quad \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 5 & 2 & 3 \end{pmatrix}. \tag{6.14}$$

Otrzymujemy dwie podgrupy $G^{A_0} = G^{A \setminus A_0} = \{\pi_2\}$, których niezmiennikami są zbiory A i $A \setminus A_0$. □

Aby ujednolicić opis permutacji i uniezależnić się od postaci konkretnego zbioru A, zamiast na jego elementach operujemy na numerach pozycji, czyli dokonujemy permutacji zbioru $I = \{1,\ldots,N\}$. Możemy teraz zdefiniować permutację jako odwzorowanie operujące na dowolnych ciągach N-wyrazowych.

Definicja 6.8 *Niech $I = \{1,\ldots,N\}$ będzie zbiorem numerów pozycji w N-wyrazowym ciągu. Niech $\Omega \subset I^N$ będzie podzbiorem wszystkich takich N-wyrazowych ciągów, których wyrazy są różnymi elementami zbioru I (czyli $\Omega = Sym(I)$). Wtedy każdy element $\omega \in \Omega$ definiuje pewną permurację N-wyrazowego ciągu elementów dowolnego zbioru A, czyli ω jest odwzorowaniem z A w A, a w szczególności z I w I, natomiast zbiór Ω jest rodziną bijekcji na A (lub na I).* □

Wniosek 6.1

1. *Każda permutacja ω może być jednoznacznie zapisana jako ciąg N różnych liczb od 1 do N. Liczba k na i-tej pozycji oznacza, że $k = \omega(i)$.*
2. *Jedyne odwzorowanie tożsamościowe zapisywane jest jako $\omega_e = (1,2,\ldots,N) \in \Omega$.*
3. *Liczba różnych permutacji elementów zbioru N-elementowego jest równa $|\Omega| = N!$.*

□

Przykład 6.4 *Zgodnie z regułą podaną we Wniosku 6.1.1, permutacja π z przykładu 6.2 może być zapisana następującym ciągiem*

$$\omega = (2,6,3,7,4,1,5). \tag{6.15}$$

□

Podane właściwości permutacji są wystarczające aby pokazać sposób jej wykorzystania w funkcji kodującej. Inne własności zostaną omówione przy wprowadzeniu metod doboru właściwej permutacji dla zadanego problemu.

6.2.2 Nowa funkcja kodująca

W poprzednim rozdziale, w definicji 3.1 wprowadziliśmy standardową definicję funkcji kodującej Ψ. W tym rozdziale wprowadzimy, wraz z koniecznymi definicjami, nową funkcję kodującą realizującą kodowanie z permutacją.

Jezeli przyjmiemy, że zbior $I = \{1, \ldots, N\}$ jest zbiorem numerów pozycji chromosomu (schematu) o długości N, to możemy przy użyciu odwzorowań ze zbioru Ω dokonywać permutacji bitów chromosomu albo pozycji schematu, czyli możemy traktować dowolną permutację $\omega \in \Omega$ jako odwzorowanie typu $\mathcal{B} \to \mathcal{B}$ albo odpowiednio $\mathcal{S} \to \mathcal{S}$, z tym, że odwzorowanie te nie będą już bijekcjami.

Przykład 6.5 *Weźmy zbiór łańcuchów o długości $N = 8$. Zdefiniujmy permutację*

$$\omega = (2,4,6,8,1,3,5,7).$$

Permutacja odwrotna jest następująca:

$$\omega^{-1} = (5,1,6,2,7,3,8,4).$$

Działając ω na przykładowe łańcuchy i schematy otrzymamy:

1. $\omega(10101010) = 00001111$;
2. $\omega(11001100) = 10101010$;
3. $\omega(*0*01*1*) = 00****11$;
4. $\omega(***01***) = *0****1*$. □

Definicja 6.9 *Funkcją kodującą z permutacją (*kodowaniem z permutacją*) nazwiemy odwzorowanie*

$$\Psi^* : X \times \Omega \to \mathcal{B}$$

przyporządkowujące każdemu elementowi $x \in X$ chromosom $\Psi^*(x,\omega) \in \mathcal{B}$ dla pewnego $\omega \in \Omega$. Odwzorowanie Ψ^* jest bijekcją ze względu na swój pierwszy argument. □

Istnieje ścisłe powiązanie między standardowymi funkcjami kodującymi a funkcjami kodującymi z permutacją. Widać, że jeżeli ustalimy wartość drugiego argumentu Ψ^*, to otrzymamy pewną standardową funkcję kodującą $\Psi^*(\cdot,\omega)$ typu $X \to \mathcal{B}$. Podobnie jest oczywiste, że prawdziwy jest

Lemat 6.1 *Każdą standardową funkcję kodującą w postaci $\Psi^*(\cdot,\omega) : X \to \mathcal{B}$ możemy także przedstawić jako złożenie:*

$$\Psi^* = \omega \circ \Psi,$$

gdzie $\Psi : X \to \mathcal{B}$ jest pewną standardową funkcją kodującą. □

Z lematu 6.1 wynika, dla ustalonego ω zastąpienie standardowej funkcji kodującej przez funkcję kodującą z permutacją nie wpływa w żaden sposób na strukturę i działanie $\mathcal{BAG}$, natomiast wpływa jedynie na wynik operacji krzyżowania. Można także łatwo wykazać, że prawdziwe jest

Twierdzenie 6.1 *Odległość Hamminga dwóch dowolnych łańcuchów jest niezależna od użytej permutacji, czyli*

$$\bigwedge_{\underline{b},\underline{b}' \in \mathcal{B}} \bigwedge_{\omega \in \Omega} d_H(\underline{b},\underline{b}') = d_H(\omega(\underline{b}),\omega(\underline{b}')).$$

□

Dowód tego twierdzenia jest oczywisty: ponieważ odległość Hamminga jest równa liczbie różnic między bitami na tych samych pozycjach w $\underline{b}$ i $\underline{b}'$, więc jakakolwiek *jednakowa* zmiana kolejności bitów w obu tych łańcuchach nie wpływa na wartość tej odległości.

Bardzo interesujący wydaje się być wpływ permutacji na zwodniczość problemu pojmowaną jak w Definicji 3.9. Wpływ ten jest opisywany przez kolejne

Twierdzenie 6.2 *Permutacja nie wpływa na zwodniczości funkcji przystosowania.* □

Dowód tego twierdzenia wynika z definicji zwodniczości (Def. 3.9). Istotnie, jeżeli poddamy schematy $\underline{s}_1, \underline{s}_2$ oraz chromosom $\underline{b}_{opt}$ działaniu tej samej permutacji $\omega \in \Omega$, to nie wpłynie to na prawdziwość dwóch pierwszych czynników koniunkcji występującej w (3.16). Natomiast brak wpływu na prawdziwość czynnika trzeciego można uzasadnić tym, że występująca w nim funkcja $f \in \mathcal{F}$ jest w istocie następującym złożeniem

$$f = \hat{f} \circ \Psi^{*-1}(\cdot, \omega) = \hat{f} \circ \Psi^{-1} \circ \omega^{-1}. \tag{6.16}$$

na mocy Lematu 6.1 i wzoru (3.3). Wynika stąd, że dla dowolnego permutacji $\omega \in \Omega$ użytej do kodowania rozwiązań w $\mathcal{BAG}$, mamy

$$f(\omega(\underline{b})) = \hat{f}\left(\Psi^{-1}\left(\omega^{-1}\left(\omega(\underline{b})\right)\right)\right) = \hat{f}\left(\Psi^{-1}(\underline{b})\right) = f(\underline{b}), \tag{6.17}$$

co kończy dowód. □

6.2.3 Permutacje a uogólniony operator krzyżowania

W tym podrozdziale pokażemy, że kodowanie z losowo wybraną permutacją użyte w $\mathcal{BAG}$wraz z jednopunktowym krzyżowaniem jest równoważne (z pominięciem rozkładu prawdopodobieństawa możliwych rodzajów krzyżowania) krzyżowaniu jednorodnemu, które wprowadzono w Podrozdziale 3.6.

Definicja 6.10 (zachowywania krzyżowania przez permutację) *Mówimy, że permutacja $\omega \in \Omega$ zachowuje krzyżowanie uogólnione określone przez $u \in \mathcal{B}$, jeżeli dla wszystkich $\underline{b}_1, \underline{b}_2 \in \mathcal{B}$ są spełnione warunki*

$$C_u(\underline{b}_1, \underline{b}_2) = \omega^{-1}(C_u(\omega(\underline{b}_1), \omega(\underline{b}_2))) \quad \textit{oraz} \quad \overline{C}_u(\underline{b}_1, \underline{b}_2) = \omega^{-1}\left(\overline{C}_u(\omega(\underline{b}_1), \omega(\underline{b}_2))\right), \tag{6.18}$$

gdzie odzwzorowania $C_u(\underline{b}_1, \underline{b}_2)$ i $\overline{C}_u(\underline{b}_1, \underline{b}_2)$ oznaczają potomków $\underline{b}_1$ i $\underline{b}_2$, zgodnie z Definicją 3.27. □

Wprowadzimy następujące pomocnicze twierdzenie.

Twierdzenie 6.3 *Niech $\underline{b} \in \mathcal{B}$ będzie dowolnym łańcuchem. Niech $I_{\underline{b}} \subseteq I$ będzie zbiorem tych pozycji $\underline{b}$, na których są jedynki (lub zera). Grupą permutacji zachowujących łańcuch $\underline{b}$ będzie podgrupa $\Omega^{I_{\underline{b}}} \subseteq \Omega$, której niezmiennikiem jest zbiór $I_{\underline{b}}$, czyli dla dowolnego $\underline{b} \in \mathcal{B}$ zachodzi*

$$\omega(\underline{b}) = \underline{b} \iff \omega \in \Omega^{I_{\underline{b}}}. \tag{6.19}$$

□

Dowód tego twierdzenia wynika z definicji podgrupy grupy permutacji (patrz prace [16] rozdział 3 oraz [33]).

Ponieważ każde krzyżowanie uogólnione jest jednoznacznie definiowane przez odpowiedni chromosom $u \in \mathcal{B}$, więc oczywistym jest następujące twierdzenie.

Twierdzenie 6.4 *Permutacja $\omega \in \Omega$ zachowuje krzyżowane uogólnione określone przez łańcuch $u \in \mathcal{B}$, jeżeli zachowuje ona łańcuch u, czyli jeśli $\omega(u) = u$.* □

Powyższe twierdzenie łatwo można zastosować do innych rodzajów krzyżowania, np. jednopunktowego lub dwupunktowego. Wystarczy w tym celu utworzyć odpowiedni łańcuch u zgodnie z zasadami podanymi w Podrozdziale 3.6.2. Z powyższego twierdzenia wynika, że to samo krzyżowanie uogólnione możemy uzyskać w wyniku działania różnych permutacji na operator u. Liczba tych permutacji jest równa $|\Omega^{I_{\underline{b}}}|$ i zależna od liczby jedynek w u, jak to pokażemy we wzorach kombinatorycznych.

Omówienie rozkładu prawdopodobieństwa poprzedzimy wprowadzeniem podstawowych wzorów kombinatorycznych odnoszących się do kodowania z permutacją i związanych z Twierdzeniem 6.3. I tak dla dowolnego łańcuch $\underline{b} \in \mathcal{B}$ o długości N i liczbie jedynek $k = ones(\underline{b})$ mamy:

- $\Pi_{k,N} = k!(N-k)!$ – liczba wszystkich permutacji ω takich, że $\omega(\underline{b}) = \underline{b}$, czyli zachowujących krzyżowanie uogólnione określone przez $\underline{b}$;
- $C_{k,N} = \frac{N!}{k!(N-k)!} = \binom{N}{k}$ – liczba możliwych permutacji łancucha zawierającego k jedynek;
- $|\mathcal{B}| = 2^N = \sum_{i=0}^{N} \binom{N}{k}$ – liczba wszystkich łańcuchów o długości N.

Korzystając z powyższych wzorów kombinatorycznych możemy podać następujące twierdzenie o rozkładzie prawdopodobieństwa operatorem uogólnionego krzyżowania.

Twierdzenie 6.5 *Niech $u \in \mathcal{B}$ będzie operatorem uogólnionego krzyżowania. Permutacja $\omega \in \Omega$ oraz pozycja krzyżowania $k \in \langle 1, N \rangle$ niech będą wybrane ze swej dziedziny losowo z rozkładem równomiernym. Wtedy:*

1. *prawdopodobieństwo że otrzymany operator u będzie miał dokładnie k jedynek jest równe:*

$$\mathbf{P}(ones(u) = k) = \frac{1}{N}; \tag{6.20}$$

2. *prawdopodobieństwo że otrzymany operator u będzie miał postać pewnego łańcucha $\underline{b} \in \mathcal{B}$ jest równe:*

$$\mathbf{P}(u = \underline{b}) = \frac{1}{N\binom{N}{ones(\underline{b})}}. \tag{6.21}$$

□

Wprowadzimy teraz algorytm znajdywania operatora krzyżowania uogólnionego mając daną permutację i pozycję krzyżowania jednopunktowego oraz algorytm odwrotny.

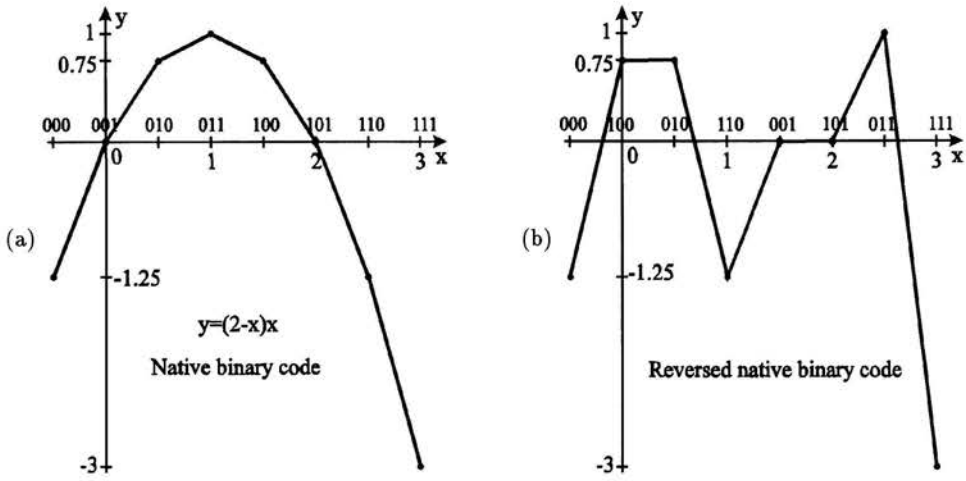
Algorytm 6.1

Mając daną permutację $\omega \in \Omega$ oraz pozycję krzyżowania jednopunktowego $k \in \langle 0, N \rangle$ możemy otrzymać równoważny im operator krzyżowania uogólnionego $u \in \mathcal{B}$ stosując następujący algorytm:

1. *Tworzymy łańcuch $\underline{b} \in \mathcal{B}$ w ten sposób, że na lewo od pozycji k-tej są same jedynki, a na prawo – same zera, czyli u jest binarną reprezentacją liczby $2^k - 1$;*
2. *Znajdujemy szukany operator jako łańcuch $u = \omega(\underline{b})$.* □

Algorytm 6.2

Mając dany operator krzyżowania uogólnionego $u \in \mathcal{B}$ możemy otrzymać równoważną mu permutację $\omega \in \Omega$ wraz z pozycją krzyżowania jednopunktowego $k \in \langle 0, N \rangle$ stosując następujący algorytm:



Rys. 6.3: Wykres funkcji $y = (2 - x)x$ dla $x \in \langle -0.5, 3 \rangle$ przekształcony poprzez odwócenie kolejności bitów kodu dziedziny. Należy zauważyc, że z punktu widzenia algorytmu genetycznego z binarnym kodowaniem, obie funkcje są nierozróżnialne.

1. *Obliczamy* $k = ones(u)$;
2. *Tworzymy łańcuch* $\underline{b} \in \mathcal{B}$ *w ten sposób, że na lewo od pozycji k-tej są same jedynki, a na prawo - same zera, czyli u jest binarną reprezentacją liczby* $2^k - 1$;
3. *Znajdujemy dowolną permutację* $\omega \in \Omega$ *taką, że* $u = \omega(\underline{b})$. □

Ponieważ nie ma żadnych ograniczeń stosowalności powyższych algorytmów, uzasadniają one w sposób konstruktywny wzajemną odpowiedniość między krzyżowaniem uogólnionym a krzyżowaniem jednopunktowym z permutacją kodu. Analogiczne algorytmy można sformułować także dla krzyżowania dwupunktowego, więc także w tym wypadku zachodzi odpowiedniość.

6.2.4 Znaczenie kodowania z permutacją

Głównym celem wprowadzenia funkcji kodującej z permutacją jest rozszerzenie możliwości standardowego operatora krzyżowania oraz umożliwienie sterowania jego działaniem bez ingerencji w strukturę i działanie $\mathcal{BAG}$. W wyniku przeprowadzonych eksperymentów obliczniowych stwierdzono, że szybłość znajdowania maksimum globalnego przez $\mathcal{BAG}$dla danego problemu jest zależna od permutacji użytej przy kodowaniu.

Poglądowy przykład przekształcenia, które może zmienić funkcję "łatwą" dla algorytmu gradientowego w funkcję "trudną", pokazano na rys. 6.3. Jest to funkcja typu $f : \mathbb{R} \to \mathbb{R}$ kodowana na trzech bitach przy użyciu dwóch różnych funkcji kodujących. Funkcja (a) ma jedno ekstremum będące globalnym maksimum. Widać, że algorytm gradientowy startujący z dowolnego punktu dziedziny powinien bez problemu znaleźć maksimum globalne. Funkcja (b) ma dwa maksima o podobnej wysokości. W tym przypadku algorytm gradientowy, w zależności od miejsca startu, znajduje albo maksimum globalne albo lokalne. Można stąd wnioskować, że funkcja trudna w znaczeniu tradycyjnym nie musi być trudna dla $\mathcal{BAG}$.

Wpływ permutacji na działanie $\mathcal{BAG}$można łatwo uzasadnić biorąc pod uwagę twierdzenie 3.1 o schematach. Zuważmy że dowolny schemat spełniający warunek małej rozpiętości może być, przy użyciu odpowiedniej permutacji, zamieniony w schemat nie spełniający tego warunku.

Przykład 6.6 *Weźmy zbiór łańcuchów o długości* $N = 8$ *oraz permutację*

$$\omega = (2, 4, 6, 8, 1, 3, 5, 7)$$

jak w Przykładzie 6.5. Działając ω *na schemat* $\underline{s} = **101***$ *rzędu* $o(\underline{s}) = 3$ *o rozpiętości* $\delta(\underline{s}) = 2$ *otrzymamy:*

$$\underline{s}' = \omega(\underline{s}) = 1 * * * * 1 * 0$$

Widać że w wyniku przekształcenia istotnie zmieniła się rozpiętości schematu i teraz $\delta(\underline{s}') = 7$. □

Jeśli założymy prawdziwość hipotezy 3.1 cegiełek, to widać, że zdolność algorytmu do zestawiania rozwiązania z cegiełek silnie zależy od wybranej permutacji. W przypadku gdy wybrana permutacja jest dla danego problemu nieodpowiednia, każde krzyżowanie będzie niszczyło pożyteczne schematy.

Rozdział 7

Nowe funkcje pomiarowe

7.1 Wprowadzenie

W tym rozdziale zdefiniujemy nowe przykładowe funkcje pomiarowe, wprowadzone w Podrozdziale 6.1.2 i służące do pomiaru (analizy) funkcji przystosowania w korelacyjnych metodach adaptacji off-line binarnego algorytmu genetycznego. Wszystkie przedstawione funkcje pomiarowe wykorzystują permutacje kodu ze zbioru Ω (patrz Podrozdział 6.2.1) jako adaptowany parametr. Są one funkcjami typu $\mathcal{F} \times \Omega \rightarrow \mathbb{R}$ lub $\Omega \rightarrow \mathbb{R}$ dla ustalonej funkcji przystosowania, (patrz wzory (6.5) i (6.6)). Przedstawione funkcje pomiarowe mogą być wykorzystywane w rozwiązywaniu rzeczywistych problemów, ponieważ nie wymagają znajomości globalnego optimum.

Oprócz funkcji pomiarowych stosowanych w adaptacji, wprowadzamy także kilka pomocniczych metod pomiaru funkcji przystosowania. Mogą one nam dostarczyć interesujących informacji na temat cech funkcji przystosowania, pomagających zrozumieć działanie właściwej metody pomiarowej i wyjaśnić otrzymane rezultaty eksperymentów. Niektóre z metod pomocniczych nie mogą być wykorzystywane do badania rzeczywistych problemów, ponieważ ich użycie wymaga uprzedniej znajomości optimum globalnego.

Należy zauważyć, że w żadnej z przedstawionych w tym rozdziale funkcji lub metod pomiarowych nie jest wykorzystywane zliczanie zer lub jedynek, czy też obliczanie odległości Hamminga, ponieważ wszystkie takie pomiary są niezależne od permutacji kodu (patrz Tw. 6.1) i dlatego nie ma sensu używanie ich w sytuacji gdy adaptowanym parametrem jest permutacja kodu. Przykładem takiej metody pomiarowej jest metoda *Fitness Distance Correlation* opisana w pracy [32].

Wszystkie przedstawione w tym rozdziale funkcje i metody pomiarowe są algorytmami czysto heurystycznymi, to znaczy, że powstały one w całości na podstawie intuicyjnego rozumienia i analizy działania $\mathcal{BAG}$. Z tego względu autor nie jest w stanie udowodnić ich matematycznej poprawności, natomiast tam gdzie było to możliwe, podane zostały analogie i odniesienia do ogólnie przyjętej teorii algorytmów genetycznych. Wartość przedstawionych funkcji i metod jest potwierdzona wyłącznie na drodze eksperymentalnej (patrz Rozdział 8). Podobnie celem przytoczonych poniżej definicji jest jedynie precyzyjne i jednoznaczne oddanie algorytmu służącego do przeprowadzania opisywanego pomiaru. Ponieważ podejście heurystyczne odgrywało tak istotną rolę, przy każdym z poniższych pomiarów przedstawiono objaśnienie kluczowych idei z nim związanych.

7.2 Pomiar histogramu zakłóceniowego przedziału roboczego

Pierwszą z proponowanych metod pomiaru funkcji przystosowania jest metoda histogramu zakłóceniowego przedziału roboczego (*ang. working interval disturbance histogram*) (WIDH). Do wprowadzenia metody niezbędne jest uprzednie zdefiniowanie kilku pomocniczych pojęć i pomocniczej metody pomiarowej mierzącej wpływ wartości bitów chromosomu na wartość funkcji przystosowa-

nia.

7.2.1 Pomiar wpływu wartości bitów chromosomu na wartość funkcji przystosowania

Definicja 7.1 *Niech $f \in \mathcal{F}$ będzie funkcją przystosowania. Definiujemy absolutny przyrost funkcji f w kierunku i-tego bitu jako*

$$\Delta_i f(\underline{b}) = |f(b_1, \ldots, 1, \ldots, b_N) - f(b_1, \ldots, 0, \ldots, b_N)| \quad dla\ i = 1, \ldots, N, \tag{7.1}$$

gdzie chromosom $\underline{b} \in \mathcal{B}$ ma wszystkie swoje bity ustalone z wyjątkiem i-tego. □

Dla ustalonej funkcji przystosowania f i ustalonego i, wartość $\Delta_i f(\underline{b})$ zależy wyłącznie od bitów $b_1, \ldots, b_{i-1}$ oraz $b_{i+1}, \ldots, b_N$. Bity te wybramy losowo z rozkładem równomiernym, a zbiór ich wartości traktujemy jako zbiór zdarzeń elementarnych. W takim przypadku możemy przyjąć, że wartości $\Delta_i f(\underline{b})$ są realizacją zmiennej losowej $\Delta_i f : \{0,1\}^{N-1} \to \mathbb{R}$.

Definicja 7.2 *Definiujemy* **średni wpływ** *i-tego bitu na wartość funkcji przystosowania f jako:*

$$P_f(i) = \mathbf{E}(\Delta_i f) = \frac{1}{|\mathcal{B}|} \sum_{\underline{b} \in \mathcal{B}} \Delta_i f(\underline{b}) \quad \text{dla } i = 1, \ldots, N, \tag{7.2}$$

gdzie $\mathbf{E}(\cdot)$ jest wartością oczekiwaną. □

Tak zdefiniowany pomiar daje nam pewne interesujące informacje o cechach funkcji przystosowania i użytym kodowaniu. Można zauważyć, że permutacja kodu nie wpływa na same wartości $P_f(i)$, a jedynie zmienia ich porządek, czyli przypisanie do kolejnych numerów bitów i.

Przy spełnieniu pewnych warunków wartość $P_f(i)$ może być wykorzystana do oszacowania wartości oczekiwanej wyrażenia $\frac{\partial}{\partial b_i} f(\underline{b})$ traktowanego jako zmienna losowa dla $\underline{b}$ wybieranego losowo (jeśli f byłaby traktowana jako funkcja N ciągłych zmiennych, z których każda należy do przedziału $\langle 0, 1 \rangle$).

Definicja 7.3 *Definiujemy* **roboczy przedział** *$\langle P_{fmin}, P_{fmax} \rangle$ zmian funkcji przystosowania w którym wartości graniczne są następujące:*

$$P_{fmin} = \min_i P_f(i), \quad i \quad P_{fmax} = \max_i P_f(i). \tag{7.3}$$

□

Roboczy przedział jest podprzedziałem pełnego zakresu zmian funkcji przystosowania, który jest rozumiany jako zakres wszystkich wartości przyjmowanych przez wyrażenie

$$|f(\underline{b}_1) - f(\underline{b}_2)|$$

wektory $\underline{b}_1, \underline{b}_2$ przyjmują wszystkie możliwe wartości z $\mathcal{B}$.

7.2.2 Histogram zakłóceniowy przedziału roboczego

Operator zakłócenia

Definicja 7.4 *Niech $\mathcal{B}$ będzie jak wyżej. Niech $\mathcal{D}_w = \{0,1\}^w \setminus \{(0,0,\ldots,0)\}$ będzie zbiorem* **wektorów zakłóceń**. *Definiujemy* **oprerator zakłócenia** *jako funkcję $D_{i,w} : \mathcal{B} \times \mathcal{D}_w \to \mathcal{B}$, której wartością jest jej argument ze zmienionymi pewnymi bitami zgodnie z poniższym warunkiem.*

Niech $\underline{b}, \hat{\underline{b}} \in \mathcal{B}$, $\underline{d} \in \mathcal{D}_w$ *i* $\hat{\underline{b}} = D_{i,w}(\underline{b}, \underline{d})$, *wtedy bity łańcuchów* $\underline{b}$ *i* $\hat{\underline{b}}$ *spełniają następujący warunek:*

$$\bigwedge_{\substack{k\in\langle 1,w\rangle \\ l=(i+k-1) \bmod N}} \hat{b}_l = \begin{cases} b_l, & gdy\ d_k = 0 \\ \bar{b}_l, & gdy\ d_k = 1 \end{cases} \tag{7.4}$$

gdzie $w, i \in \langle 1, N\rangle$ *są parametrami operatora zakłócenia: w jest* **szerokością zakłócenia** *a i jest* **pozycją zakłócenia**. *Zakładamy, że pozycje bitów w chromosomie są numerowane cyklicznie, to znaczy, że następną pozycją po N-tej jest pozycja pierwsza. Ponadto będziemy mówili, że operator zakłócenia generuje* **kroki** *w przestrzeni chromosomów* $\mathcal{B}$. □

Innymi słowy operator $D_{i,w}$ zawsze zwraca swój argument z zakłóceniem na bitach o numerach od i do $i + w - 1$ włącznie, zgodnie z wartością drugiego argumentu. Definicja powyższa nas zapewnia, że operator zakłocenia jest odwzorowaniem przeciwzwrotnym, to znaczy że

$$\bigwedge_{\substack{\underline{b}\in\mathcal{B} \\ \underline{d}\in\mathcal{D}_w}} \underline{b} \neq D_{i,w}(\underline{b}, \underline{d}).$$

Łatwo zauważyć, że operator zakłócenia jest przykładem operatora przemieszczenia w myśl Definicji 3.10. Sposób jego działania jest w pewien sposób podobny do wymiany ciągów bitów między dwoma chromosomami realizowanej przez operator krzyżowania dwupunktowego. Można zauważyć, że podobnie jak w przypadku krzyżowania, działanie operatora zakłócenia zależy od użytej permutacji kodu.

Jeśli pozycję zakłócenia i i wektor zakłóceń $\underline{d}$ wybieramy w losowo z ich dziedzin, to zapisujemy operator zakłocenia w sposób skrócony: D_w, oraz pomijamy drugi argument. Dodatkowo będziemy zakładali, że parametr w jest ustalony – przyjmujemy, że $w = round(\sqrt{N})$. Wartość ta może być interpretowana jako średnia długość cegiełki rozumianej zgodnie z teorią algorytmów genetycznych (patrz prace [37] i [28]).

W literaturze można znaleźć operator bardzo podobny do operatora zakłócenia, na przykład w pracy Jones'a [36]. Jest on nazywany makromutacją lub krzyżowaniem losowym. Są dwie podstawowe różnice między makromutacją, a operatorem zakłócenia. Po pierwsze – operator zakłocenia jest przeciwzwrotny, po drugie – maksymalna liczba modyfikowanych przez niego bitów jest równa szerokości zakłocenia w a nie długości łańcucha N jak w przypadku makromutacji.

Bezwzględny przyrost zakłóceniowy

Definicja 7.5 (bezwzględnego przyrostu zakłóceniowego) *Niech* $f, w, \underline{d}, \underline{b}$ *będą jak wyżej. Definiujemy* **bezwzględny przyrost zakłóceniowy** *następującym wyrażeniem:*

$$\Delta_{i,w,\underline{d}}(f, \underline{b}) = |f(\underline{b}) - f(D_{i,w}(\underline{b}, \underline{d}))| \tag{7.5}$$

W niniejszej rozprawie przyrost będziemy nazywać alternatywnie **długością kroku**. □

Podobnie jak w przypadku skróconego zapisu operatora zakłócenia, używamy skróconego zapisu przyrostu zakłoceniowego: $\Delta_w(f, \underline{b})$.

Jeśli wartości $i, \underline{b}, \underline{d}$ będziemy wybierali losowo ze swej dziedziny, to możemy przyjąć, że wartość $\Delta_{i,w,\underline{d}}(f, \underline{b})$ dla ustalonego w jest realizacją zmiennej losowej $Y : \mathbb{N} \times \mathcal{B} \times \mathcal{D}_w \to \mathbb{R}$.

Zdefiniujmy nową zmienną losową $Y_L = \ln Y$ (zakładamy, że zawsze $Y > 0$). Naszym celem jest znalezienie rozkładu losowego Y_L, który może być przybliżony histogramem Y_L. Aby go uzyskać obliczamy R-krotnie Y_L i otrzymujemy ciąg liczb. Przedział zawierający wszystkie te liczby dzielimy na M podprzedziałów, zliczając ilość liczb wpadającą do każdego podprzedziału. Liczba obliczeń Y_L, R powinna być większa większej długości chromosomu i, jak przypuszczamy, dla bardziej nieregularnych funkcji przystosowania. Liczba podprzedziałów histogramu M powinna być tak dobrana do R, aby uzyskać w każdym przedziale *wystarczająco* dużo liczb.

Aby otrzymać WIDH, histogram tworzymy wyłącznie na podstawie takich wartości Y_L, dla których odpowiadające im wartoći Y należą do roboczego przedziału, zdefiniowanego powyżej, co oznacza, że spełniony jest warunek $Y_L \in \langle \ln P_{fmin}, \ln P_{fmax} \rangle$. Stworzony histogram jest dyskretną funkcją $H_{WIDH} : \mathbb{N} \to \mathbb{N}$, której argument $m \in \langle 1, M \rangle$ jest numerem podprzedziału przedziału roboczego, a jej wartość określa, ile liczb wpadło do danego podprzedziału.

Odchylenie standardowe funkcji H_{WIDH} zostanie przez nas użyte jako wynik pomiaru funkcji przystosowania metodą WIDH. Dla ustalonej funkcji przystosowania pomiar ten zapisujemy jako funkcję permutacji kodu: $\sigma_{WIDH} : \Omega \to \mathbb{R}$.

7.2.3 Objaśnienie idei pomiaru

Jak wcześniej wspomniano, sposób, w jaki działa operator zakłócenia, jest podobny do wymiany łańcuchów binarnych między dwoma chromosomami przez operator krzyżowania. Są dwie główne różnice między tymi operatorami. Po pierwsze operator krzyżowania jest zdefiniowany jako odwzorowanie typu $c : \mathcal{B}^2 \to \mathcal{B}^2$, stąd można powiedzieć, że działanie operatora krzyżowania na pewną parę chromosomów jest w pewien sposób równoważne działaniu operatora zakłócenia na każdy z tych chromosomów. Widać, że tak zdefiniowany operator krzyżowania generuje jednocześnie dwie ścieżki w przestrzeni posukiwań. Po drugie, maksymalna liczba zmienianych przez operator zakłócenia bitów jest równa szerokości zakłocenia w. Natomiast w przypadku operatora krzyżowania można przyjąć, że jest ona równa połowie długości chromosomu, czyli $\frac{N}{2}$. Można stąd przypuszczać, że zależność działania obu operatorów od użytej permutacji kodu jest podobna. Stąd wywnioskowaliśmy, że można przewidzieć właściwości operatora krzyżowania dla danej permutacji kodu, badając wyłącznie właściwości operatora zakłócenia dla tej zamej permutacji. Może być to użyteczne zwłaszcz dlatego, że nie musimy operować na całej populacji chromosomów.

Proces poszukiwania maksimum funkcji przystosowania może być podzielony na dwie fazy:

- globalne szukanie – szukanie "obiecujących" oszarów przestrzeni poszukiwań, oraz
- lokalne szukanie – "wspinanie się" na znalezione "wzgórze" lokalnego lub globalnego maksimum.

Zależnie od kształtu funkcji przystosowania, obie fazy mogą mieć różny udział w pracy $\mathcal{BAG}$. W globalnym szukaniu długie kroki mają znaczenie najistotniejsze, w szukaniu lokalnym – kroki krótkie. W przypdku gdy operator krzyżowania słabo generuje długie kroki, pewne obszary przestrzeni poszukiwań mogą być trudno dostępne. Gdy krótkie kroki są generowane słabo, to globalne optimum może być przypadkowo przeskoczone. Wydaje się, że dobry operator krzyżowania powinien posiadać wysoką zdolność do generowania obu rodzajów kroków. Jeśli założymy, że oceniamy właściwości operatrora krzyżowania poprzez ocenę właściwości operatora zakłócenia, to powyższy wymaganie oznacza, że histogram długości kroków wygenerowanych przez operator zakłócenia, czyli WIDH, powinien być płaski, czyli jego odchylenie standardowe σ_{WIDH} – niskie.

Teraz jedynie pojęcie przedziału roboczego wymaga pewnych wyjaśnień. Bezwzględne przyrosty zakłóceniowe generowane przez operator zakłóceniowy mogą się zmieniać w bardzo szerokim zakresie. Bardzo małe i bardzo duże przyrosty są spowodowane niepowtarzalnymi lokalnymi interakcjami pomiędzy zmienianymi bitami, dlatego nie mają one wpływu na globalne właściwości $\mathcal{BAG}$. Dlatego tworzymy histogram zakłóceniowy wyłącznie w zakresie przedziału roboczego $\langle P_{fmin}, P_{fmax} \rangle$.

7.3 Pomiar korelacji między przystosowaniem i jego przyrostem

7.3.1 Opis algorytmu pomiaru

Drugą z proponowanych funkcji pomiarowych polega na wyznaczeniu korelacji między przystosowaniem a przyrostem przystosowania (*ang. fitness increment correlation*) (FIC). Wykorzystuje ona

zdefiniowane uprzednio pojęcie operatora zakłócenia (Definicja 7.4), pojęcie bezwzględnego przyrostu zakłóceniowego (Definicja 7.5) oraz inne pojęcia związane z metodoą WIDH. Zdefiniujemy teraz nowe pojęcie wykorzystywane w opisywanej metodzie.

Definicja 7.6 (średniego przystosowania zakłóceniowego) *Niech $f, w, \underline{d}, \underline{b}$ będą jak w Definicji 7.4. Definiujemy* średnie przystosowanie zakłóceniowe *następującym wyrażeniem:*

$$F_{i,w,\underline{d}}(f,\underline{b}) = \frac{f(\underline{b}) + f(D_{i,w}(\underline{b},\underline{d}))}{2}. \tag{7.6}$$

□

Zgodnie z wcześniejszymi rozważaniami przyjmujemy, że szerokość zakłócenia w jest stała. Aby teraz dokonać pomiaru należy utworzyć trzy losowe ciągi o długości W (dla $k = 1, \ldots, W$):

1. ciąg chromosomów $\underline{b}_k \in \mathcal{B}$;
2. ciąg wektorów zakłóceniowych $\underline{d}_k \in \mathcal{D}_w$;
3. ciąg pozycji zakłócenia $i_k \in \langle 1, N \rangle$.

Na podstawie tych ciągów tworzymy dwa nowe ciągi:

$$\Delta_f = \{\Delta_{i_k,w,\underline{d}_k}(f,\underline{b}_k)\} \quad \text{oraz} \quad F_f = \{F_{i_k,w,\underline{d}_k}(f,\underline{b}_k)\}, \tag{7.7}$$

gdzie $\Delta_{i,w,\underline{d}_k}$ jest bezwzględnym przyrostem zakłóceniowym (patrz Definicja 7.5). Następnie, zgodnie z Definicją 5.15, obliczamy korelację między tymi dwoma ciągami (patrz Uwaga 5.1)

$$r_{FI}(\omega) = \varrho(\Delta_f, F_f), \tag{7.8}$$

która będzie stanowiła wynik pomiaru metodą FIC. Należy zwrócić uwagę, że r_{FI} jest zależna od wartości parametru, czyli od permutacji kodu $\omega \in \Omega$, ze względu na sposób obliczania przystosowania chromosomu (patrz Wniosek 3.1).

7.3.2 Objaśnienie idei pomiaru

Pomysł metody FIC oparty jest na prostym heurystycznym wniosku sformułowanym na podstawie analizy działania $\mathcal{BAG}$. Przypuśćmy, że mamy funkcję przystosowania o następujęcej właściwości: im wyższe jest przystosowanie w pewnym punkcie jej dziedziny, tym mniejsze są zmiany przystosowania w sąsiedztwie tego punktu. Możemy podejrzewać, że taka funkcja może być łatwiej optymalizowana niż inne, ponieważ jej kształt odpowiada najczęstszemu przebiegowi optymalizacji, to znaczy, na początku mamy szukanie globalne z dużymi przemieszczeniami w dziedzinie funkcji przystosowania, a następnie przemieszczenia maleją w miarę zbliżania się do coraz lepszych rozwiązań w fazie szukania lokalnego.

W wyniku takiego rozumowania otrzymujemy kryterium, które może być wykorzystywane do pomiaru funkcji przystosowania, jeżeli znajdziemy sposób oceny, w jakim stopniu badana funkcja spełnia to kryterium. Wydaje się że najwłaściwszą metodą jego zaimplementowania tego kryterium, jest wyznaczenie korelacji między średnim przystosowaniem osobników przed i po krzyżowaniu a przyrostem przystosowania w wyniku krzyżowania. Obliczenia te należałoby przeprowadzić dla odpowiednio długiego, losowo wybranego ciągu chromosomów, a następnie poddaniu ich operacji krzyżowania albo operacji podobnej, na przykład operacji zakłócenia zgodnej z Definicją 7.4. Szczegółowy opis metody pomiaru funkcji przystosowania bazujący na przedstawionym pomyśle został przedstawiony w poprzednim podrozdziale.

7.4 Pomiar korelacji między rodzicami a potomstwem

7.4.1 Opis algorytmu pomiaru

Trzecia z proponowanych funkcji pomiarowych w przybliżeniu odpowiada wyznaczeniu korelacji między rodzicami a potomstwem (*ang. parent offspring correlation*) (POC), tworzonym z rodziców w $\mathcal{BAG}$ w trakcie krzyżowania. W celu uproszczenia pomiaru, zamiast krzyżowania wykorzystujemy zdefiniowane uprzednio pojęcie operatora zakłócenia (Definicja 7.4).

Zgodnie z wcześniejszymi rozważaniami przyjmujemy, że szerokość zakłócenia w jest stała. Aby teraz dokonać pomiaru należy utworzyć trzy losowe ciągi o długości W (dla $k = 1, \ldots, W$):

1. ciąg chromosomów $\underline{b}_k \in \mathcal{B}$;
2. ciąg wektorów zakłóceniowych $\underline{d}_k \in \mathcal{D}_w$;
3. ciąg pozycji zakłócenia $i_k \in \langle 1, N \rangle$.

Na podstawie tych ciągów tworzymy dwa nowe ciągi:

$$F_{pf} = \{f(\underline{b}_k)\} \quad \text{oraz} \quad F_{of} = \{f(D_{i_k,w}(\underline{b}_k, \underline{d}_k))\}, \tag{7.9}$$

gdzie $D_{i_k,w}(\cdot, \underline{d}_k)$ jest operatorem zakłócenia (patrz Definicja 7.4). Następnie, zgodnie z Definicją 5.15, obliczamy korelację między tymi dwoma ciągami (patrz Uwaga 5.1)

$$r_{PO}(\omega) = \varrho(F_{pf}, F_{of}), \tag{7.10}$$

która będzie stanowiła wynik pomiaru metodą POC. Należy zwrócić uwagę, że r_{PO} jest zależna od wartości parametru, czyli od permutacji kodu $\omega \in \Omega$, ze względu na sposób obliczania przystosowania chromosomu (patrz Wniosek 3.1).

7.4.2 Objaśnienie idei pomiaru

Pomysł metody POC został został zainspirowany pracą Altenberga [40]. Załóżmy, że mamy taki $\mathcal{BAG}$ i taką funkcję przystosowania, że przystosowania rodziców i potomstwa nie są skorelowane, albo nawet są skorelowane ujemnie, czyli $r_{PO} \leq 0$. Podobnie jak w poprzednich dwóch pomiarach, może to powodować istotną trudność w optymalizacji, gdyż nawet jeśli rodzice mają wysokie przystosowanie, to jest mniejsza szansa na to, że podobne przystosowanie będzie miało ich potomstwo. Możemy się także liczyć z dużymi skokami przy zbliżaniu się do optimum globalnego, co może powodować trudności w "trafieniu" w to optimum. Stąd wnioskujemy, że $\mathcal{BAG}$ prawdopodobnie będzie działał sprawniej dla danego problemu przy takim doborze adaptowanego parametru, przy którym wartość r_{PO} będzie większa dla losowo wybranego i zakłóconego losowo ciągu chromosomów. W wyniku takiego rozumowania otrzymujemy kryterium, które może być wykorzystywane do pomiaru funkcji przystosowania.

7.5 Pomocnicze metody pomiarowe

7.5.1 Metoda średniej długości ścieżki w sąsiedztwie optimum globalnego

Opis metody

W trakcie eksperymentów z funkcjami WIDH i FIC zauważyliśmy, że potrzebna jest pomocnicza metoda pomiaru, która by mierzyła wprost zdolność algorytmu do lokalnego przeszukiwania *(ang. hillclimbing)*. Metodą taką jest metoda prezentowna w niniejszym podrozdziale, którą będziemy nazywali **metodą średniej długości w sąsiedztwie optimum globalnego** *(ang. Mean Path Length in*

neighbourhood of global optimum) (MPL). Nie może ona być wykorzystywana do pomiaru rzeczywistych problemów, ponieważ wymaga znajomości optimum globalnego, dlatego zaliczamy ją do grupy metod pomocniczych.

W celu zdefiniowania metody MPLposłużymy się pojęciem nieskończonej ścieżki losowej zdefiniowanym w Def. 3.26. Przyjmijmy następujące oznaczenie **ścieżki z globalnego optimum**

$$\Pi = \Pi_{\underline{b}_{opt},D_w} = \left(\underline{b}_{opt}, D_w^1(\underline{b}_{opt}), D_w^2(\underline{b}_{opt}), \ldots\right) \in \mathcal{B}^\infty, \tag{7.11}$$

gdzie D_w – operator zakłócenia zdefiniowany w Def. 7.4, a $\underline{b}_{opt}$ – optimum globalne funkcji przystosowania $f \in \mathcal{F}$. Niech $\Pi(i) \in \mathcal{B}$ oznacza i-ty element ścieżki Π, Przykłady takich ścieżek pokazano na Rys. 7.1.

Definicja 7.7 *Weźmy f, $\mathcal{B}$, and D_w jak powyżej, i niech wartości f_{min} oraz $f_{max} \equiv f(\underline{b}_{opt})$ oznaczają odpowiednio minimalną i maksymalną wartość funkcji przystosowania f. Definiujemy* **długość ścieżki w sąsiedztwie globalnego optimum** *(ang. Path Length in a neighbourhood of global optimum) jako:*

$$\widehat{\mathsf{PL}}(f_{low}, w, \Pi) = \min\{i : i \geq 1 \wedge f(\Pi(i)) < f_{low}\} \tag{7.12}$$

gdzie parametr f_{low} definiuje rozmiar sąsiedztwa, i spełnia warunek $f_{low} \in (f_{min}, f_{max}\rangle$. □

Znacznie wygodniej jest używać parametru znormalizowanego zamiast f_{low}. Dlatego będziemy obliczać f_{low} w następujący sposób:

$$f_{low}(\beta) = f_{max} - \beta\left(f_{max} - \overline{f}\right) \tag{7.13}$$

gdzie $\overline{f}$ jest średnią wartością funkcji przystosowania, a $\beta \geq 0$ jest nowym parametrem definiującym sąsiedztwo, takim, że:

$$f_{low}(\beta) > f_{min}. \tag{7.14}$$

Korzystając z parametru β wprowadzamy nowe oznaczenie:

$$\mathsf{PL}(\beta, w, \Pi) \equiv \widehat{\mathsf{PL}}(f_{low}(\beta), w, \Pi). \tag{7.15}$$

Ponieważ założyliśmy wcześniej, że parametr w operatora zakłócenia jest ustalony, możemy traktować PL jako funkcję jedynie β oraz Π, czyli $\mathsf{PL} : \mathbb{R} \times \mathcal{B}^\infty \to \mathbb{R}$.

Definicja 7.8 *Niech β spełnia warunek 7.14. Definiujemy* **średnią długość ścieżki w sąsiedztwie globalnego optimum** *(ang. Mean Path Length in a neighbourhood of global optimum) jako funkcję typu $\mathbb{R} \to \mathbb{R}$ określoną poprzez formułę:*

$$\mathsf{MPL}(\beta) = \frac{1}{|\mathcal{B}^\infty|} \sum_{\Pi \in \mathcal{B}^\infty} \mathsf{PL}(\beta, \Pi) \tag{7.16}$$

□

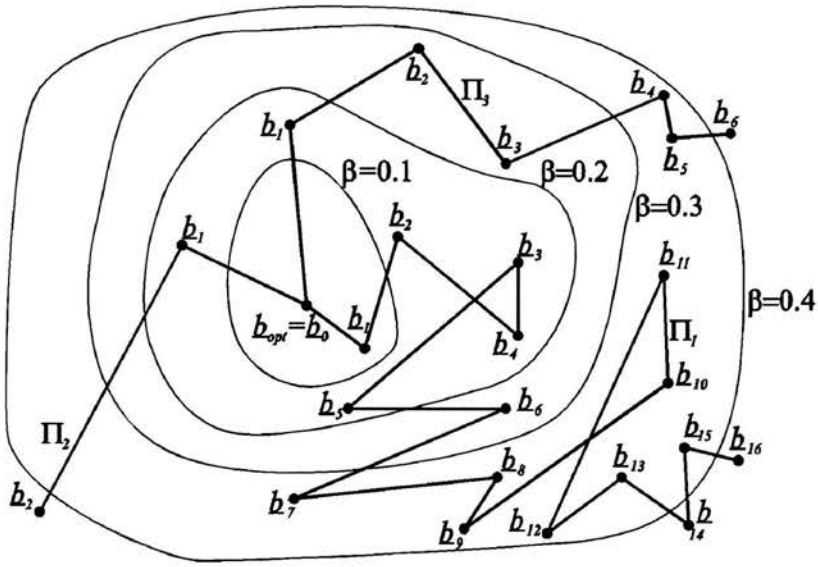
W niniejszej rozprawie będziemy obliczali MPL(β) dla wartości z przedziału $\beta \in \langle 0, \beta_{max}\rangle$ aby otrzymać w wyniku krzywą MPL. Wartość β_{max} nie powinna być większa od 1 – określa ona rozmiar badanego sąsiedztwa globalnego optimum czyli umowny obszar szukania lokalnego.

Ponieważ obliczenie funkcji MPL bezpośrednio z Def. 7.8 jest niemożliwe, więc w eksperymentach obliczeniowych będziemy przybliżali kształt krzywej na podstawie skończonej liczby ścieżek, oraz będziemy generować jedynie odpowiednią liczbę początkowych wyrazów ścieżki – właściwą dla wybranego β_{max}.

Jak można zauważyć wartości funkcji MPL zależą w pośredni sposób od użytej permutacji kodu. Jest to spowodowane wykorzystaniem zależnego od permutacji operatora zakłócenia. Dlatego ostatecznie będziemy traktować MPL jako funkcję typu:

$$\mathsf{MPL} : \mathbb{R} \times \Omega \to \mathbb{R} \tag{7.17}$$

gdzie drugi argument oznacza permutację kodu.



Rys. 7.1: Przykładowe ścieżki losowe z globalnego optimum w przestrzeni poszukiwań $\mathcal{B}$ wygenerowane zgodnie z Def. 3.26. Można wyczytać z rysunku, że $\mathsf{PL}(0.1, \Pi_2) = 1$, $\mathsf{PL}(0.4, \Pi_1) = 14$, $\mathsf{PL}(0.4, \Pi_3) = 6$, itd.

Objaśnienie idei metody

Główna idea MPL polega na analizie w przeciwnej kolejności rozwiązań wygenerowanych przez $\mathcal{BAG}$, co oznacza, że zaczynamy ze znanego optimum globalnego i analizujemy wszystkie ścieżki dojścia do niego. Znalezienie takich sieżek byłoby trudne przy analizie zgodnie z kierunkiem generowania ścieżek przez $\mathcal{BAG}$. Celem metody pomiarowej jest dowiedzenie się, ile średnio końcowych rozwiązań z analizowanej ścieżki leży w pewnnym zadanym sąsiedztwie optimum globalnego. W metodzie MPL nie używamy do tego celu $\mathcal{BAG}$, ale symulujemy jego pracę wprowadzając wielokrotnie losowe zakłocenia odcinków chromosomu o ustalonej długości. Wykorzystujemy do tego celu operator zakłócenia opisany w Def. 7.4. Tak więc w trakcie pomiaru będziemy zliczać, ilu zakłoceniom musimy poddać optymalny chromosom, aby opuścić dane sąsiedztwo globalnego optimum.

Rozdział 8

Omówienie przeprowadzonych badań eksperymentalnych

8.1 Opis bazowego algorytmu genetycznego

Przyjęto, że wszystkie eksperymenty będą przeprowadzane z wykorzystaniem tej samej, bazowej wersji $\mathcal{BAG}$, tak aby było możliwe porównywanie otrzymywanych wyników. Wprowadzona modyfikacja polegała wyłącznie na zamianie standardowej funkcji kodującej na funkcję kodującą z permutacją. Zmiana taka jest "przeźroczysta" dla $\mathcal{BAG}$, to znaczy nie powoduje zmian jakiegokolwiek elementu algorytmu rozumianego jako sekwencja operacji. Jak wspomniano wcześniej, jedynym elementem $\mathcal{BAG}$, na który zmiana sposobu kodowania wywiera pośredni wpływ jest operacja krzyżowania.

Wybór bazowej wersji $\mathcal{BAG}$ był dokonany na podstawie wyników badań różnych wersji $\mathcal{BAG}$ opisanych w literaturze oraz na podstawie eksperymentów przeprowadzonych przez autora. Kierowano się przy tym następującymi kryteriami:

- algorytm powinien sprawnie znajdować rozwiązanie każdego z testowych problemów, w czasie na tyle krótkim, aby było możliwe wykonanie w rozsądnym czasie odpowiedniej liczby eksperymentów, wystarczającej do uzyskania wiarygodnych wyników;
- wybrana wersja algorytmu powinna być w największym możliwym stopniu zbliżona do standardowego algorytmu genetycznego zaproponowanego przez Hollanda w pracy [5], tak aby w maksymalnym stopniu zachować wszystkie jego właściwości i leżące u ich podstaw idee.

Na tej podstawie wybrano wersję $\mathcal{BAG}$ której elementy składowe skomentowano i scharakteryzowano w skrócie poniżej:

- **krzyżowanie dwupunktowe** – jak wykazali Spears i DeJong w pracy [18], ten rodzaj krzyżowania zapewnia najwyższe średnie prawdopodobieństwo przeżycia schematów o różnych długościach. Ci sami autorzy wykazali, że najniższe prawdopodobieństwo przeżycia występuje w przypadku krzyżowania jednorodnego, stąd można wnioskować, że modyfikacja permutacji kodu pozwala na efektywne korzystanie z właściwości obu powyższych rodzajów krzyżowania.
- **elitaryzm** polegający na zapewnieniu przejścia podczas selekcji najlepszego osobnika z pokolenia aktualnego do pokolenia następnego – taki sposób modyfikacji metody selekcji został zaproponowany przez De Jonga w pracy [6], natomiast Rudolph w pracy [25] udowodnił, że elitaryzm w $\mathcal{BAG}$ gwarantuje zbieżność do globalnego optimum, w przeciwieństwie do $\mathcal{BAG}$ bez elitaryzmu. Istotnie, na podstawie doświadczeń stwierdzono że poprawia on radykalnie sprawność $\mathcal{BAG}$ pozwalając na uzyskanie lepszych wyników przy mniejszej liczbie generacji.

- **selekcja metodą ruletki** – w wyniku przeprowadzonych eksperymentów zrezygnowano z ograniczania rozrzutu metody ruletki poprzez np. zastosowanie selekcji deterministycznej lub według reszt (patrz praca Brindle [10]). Wydaje się, że najwłaściwsze jest połączenie elitaryzmu dającego gwarancję bieżności ze standardową metodą ruletki zapewniającą odpowiednio intensywną eksplorację przestrzeni poszukiwań.

- **warunek stopu** – algorytm zatrzymuje się wtedy, gdy przystosowanie najlepszego chromosomu w aktualnej populacji jest mniejsze od globalnego maksimum nie więcej niż o maksymalny błąd ϵ, zdefiniowany osobno dla każdego z badanych problemów. Zastosowany warunek pozwala uwzględnić w pomiarach także fazę szukania lokalnego w otoczeniu maksimum globalnego. Przeprowadzone badania wykazały, że przyjęty rodzaj parametryzacji $\mathcal{BAG}$ (permutacja kodu) ma największy wpływ na działanie algorytmu właśnie w końcowej jego fazie. Dlatego zrezygnowano z przyjętej przez innych badaczy metody oceny sprawności algorytmu, polegającej na wzięciu najlepszego przystosowania osiągniętego w zadanej stałej liczbie pokoleń.

- **funkcja kodująca** – zastosowano funkcję kodującą z permutacją, zgodną z Definicją 6.9, umożliwiającą testowanie działania $\mathcal{BAG}$ dla różnych permutcji kodu.

8.2 Opis użytych testowych funkcji przystosowania

8.2.1 Wprowadzenie

Zaproponowane poniżej testowe problemy są funkcjami bardzo prostymi – są uproszczonymi ale nie trywialnymi wersjami typowych zadań rozwiązywanych zwykle przy użyciu $\mathcal{BAG}$. Aby oszacować w sposób wiarygodny efektywność $\mathcal{BAG}$ dla zadanego zestawu parametrów, konieczne jest wielokrotne jego uruchamianie, gdyż uzyskiwane w ten sposób wyniki zwykle mają duży rozrzut. Z tego samego powodu proponowane w niniejszej rozprawie funkcje pomiarowe powinny próbkować badane funkcje przystosowania w jak największej liczbie punktów. W związku z tym użycie bardziej złożonych, rzeczywistych problemów testowych było niemożliwe przy wykorzystaniu dostępnego sprzętu komputerowego.

Zauważmy, że, podobnie jak w przypadku funkcji pomiarowych (patrz Podrozdział 7.1), nie ma sensu wykorzystywanie $\mathcal{BAG}$, w którym adaptowanym parametrem jest permutacja kodu, do rozwiązywania problemów bazujących na zliczaniu zer, jedynek lub na odległości Hamminga (patrz Tw. 6.1). Przykładem bardzo popularnego problemu testowego tego rodzaju, stosowanego przez wielu badaczy (patrz praca [22]), jest funkcja ONEMAX, której wartością jest liczba jedynek w chromosomie.

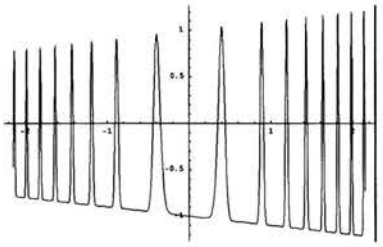
Wszystkie badane funkcje przystosowania były kodowane za pomocą 32-bitowych chromosomów. Optymalne rozwiązania (maksima) testowych problemów wraz z reprezentującymi je chromosomami zostały wyznaczane analitycznie tam, gdzie było to możliwe. W pozostałych wypadkach zostały one wyznaczone w sposób eksperymentalny za pomocą algorytmu genetycznego.

8.2.2 Problem 1: maksymalizacja funkcji argumentu skalarnego

Maksymalizujemy następującą funkcję (patrz Rys. 8.1) typu $\mathbb{R} \to \mathbb{R}$:

$$f(x) = (0.1x + 1)\left(2\sin^{40}(5x^2 + \pi/4) - 1\right) \tag{8.1}$$

dla $x \in \langle -2^{31}/10^9, (2^{31} - 1)/10^9 \rangle$ kodowanego w standardowy sposób. Globalne maksimum jest dla $f(\sqrt{1.45\pi}) = 1.2134317068$, maksymalny błąd ϵ jest równy 10^{-6}.



Rys. 8.1: Wykres funkcji przystosowania z Problemu 1.

8.2.3 Problem 2: problem plecakowy

Pakujemy do plecaka 32 przedmioty o ciężarach w_i i wartościach v_i. Maksymalizujemy całkowitą wartość V zapakowanych przedmiotów pod warunkiem, że całkowity ciężar W nie przekracza zadanego ciężaru maksymalnego W_{max}.

Możliwe rozwiązania są kodowane przez binarne chromosomy w ten sposób, że 1 oznacza przedmiot zapakowany, a 0 – niezapakowany, tak więc dla każdego $\underline{b} \in \mathcal{B}$ całkowity ciężar i całkowita wartość są zdefiniowane następująco:

$$W(\underline{b}) = \sum_{i=1}^{32} w_i b_i \quad \text{oraz} \quad V(\underline{b}) = \sum_{i=1}^{32} v_i b_i. \tag{8.2}$$

gdzie ciężary i wartości pakowanych przedmiotów zostały wyliczone następująco:

$$w_i = 33 - i \quad \text{oraz} \quad v_i = i^{1.5}/\sqrt{32}. \tag{8.3}$$

Maksymalizujemy następującą funkcję przystosowania:

$$f(\underline{b}) = V(\underline{b}) - g(\underline{b}). \tag{8.4}$$

Funkcja kary $g(\underline{b})$ jest zdefiniowana jako

$$g(\underline{b}) = \begin{cases} \tau(W(\underline{b}) - W_{max}) & \text{dla} \quad W(\underline{b}) > W_{max} \\ 0 & \text{wpp} \end{cases} \tag{8.5}$$

gdzie $\tau = 5$ jest współczynnikiem kary, ponadto przyjmujemy że $W_{max} = 320$, oraz że maksymalny błąd ϵ jest równy 10^{-6}.

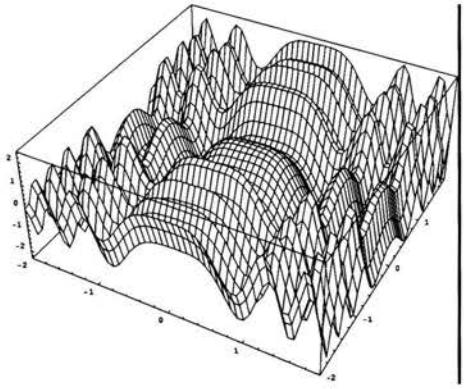
Uwaga: Użycie funkcji kary powoduje, że globalne maksimum zwykle nie jest poprawnym rozwiązaniem problemu plecakowego. Lepsze metody kodowania dla tego problemu zastały opisane przez Michalewicza w książce [37].

8.2.4 Problem 3: maksymalizacja funkcji wielu zmiennych

Maksymalizujemy następującą funkcję (patrz Rys. 8.2) typu $\mathbb{R}^n \to \mathbb{R}$ dla $n = 4$:

$$f(\underline{x}) = \sum_{i=1}^{n} (0.1x_i + 1)\left(2\sin^2\left((i+1)(x_i^2 + \pi/4)\right) - 1\right) \tag{8.6}$$

dla $\underline{x} = (x_1, \ldots, x_n)$, gdzie $x_1, \ldots, x_n \in \langle -2^7/64, (2^7 - 1)/64\rangle$. Składowe x_1, x_2, x_3, x_4 są kodowane w standardowy sposób (patrz Przykład 3.1), każda na 8 bitach. Globalne maksimum jest dla $f(\sqrt{1.45\pi}) = 1.2134317068$, a maksymalny błąd ϵ jest równy 10^{-2}.



Rys. 8.2: Uproszczony wykres funkcji przystosowania z Problemu 3 dla $n = 2$.

8.2.5 Problem 4: prosty klasyfikator

Zdefiniujmy wskaźnik jakości grupowania J korzystając z następującego wyrażenia:

$$J = \frac{\sum_{j=1}^{P} \|\underline{x}_j - \underline{m}\|^2}{\sum_{i=1}^{c} \left(\sum_{j=1}^{N_i} \|\underline{x}_{ij} - \underline{m}_i\|^2 + \|\underline{m}_i - \underline{M}\|^2 \right)} - 1 \tag{8.7}$$

gdzie P jest liczbą wszystkich wektorów wejściowych, $\underline{x}_j$ oznacza j-ty wektor wejściowy, $\underline{m}$ – średni wektor, $\underline{x}_{ij}$ – j-ty wektor z i-tego klastra, c – liczbę klastrów, N_i – liczbę wektorów w i-tym klastrze, $\underline{m}_i$ – średni wektor (centroid) i-tego klastra, $\underline{M}$ – średni centroid wszystkich klastrów.

Zadaniem $\mathcal{BAG}$ jest znalezienie takiego sposobu grupowania wektorów w klastry, dla którego wskaźnik jakości J ma maksymalną wartość. W zadaniu testowym grupujemy 16 dwuwymiarowych wektorów pomiędzy maksymalnie 4 klastry. Każde rozwiązanie składa się 16 numerów klastrów, przy czym numer klastru jest kodowany na 2 bitach, co razem daje 32 bity na jedno rozwiązanie.

8.3 Opis testowego ciągu permutacji

8.4 Badanie wpływu permutacji kodu na sprawność $\mathcal{BAG}$

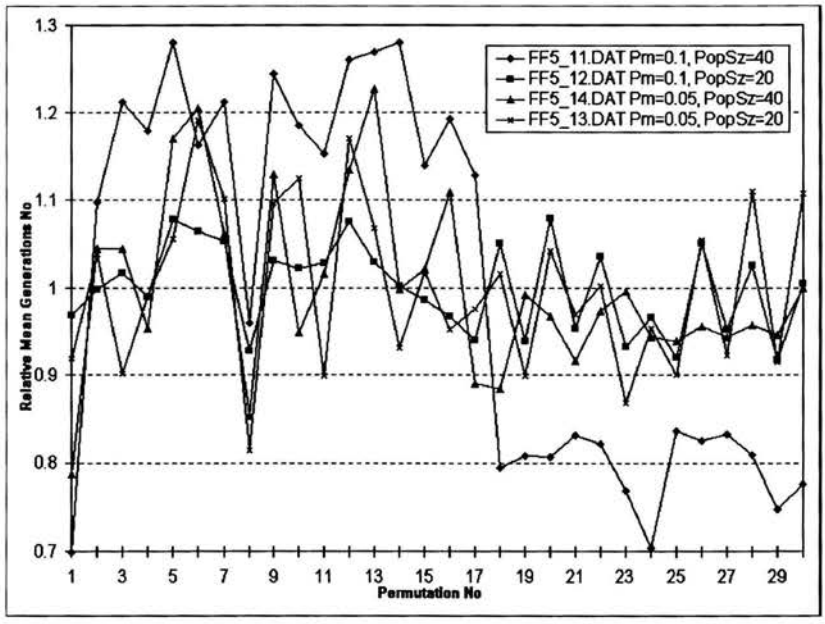
8.4.1 Eksperyment podstawowy

Wyniki eksperymentów pokazujące wpływ doboru permutacji kodu na sprawność działania $\mathcal{BAG}$ były tymi, które zachęciły autora do szerszego zajęcia się tematyką poruszaną w niniejszej rozprawie. W tym punkcie przedstawiamy wykresy pokazujące względną oczekiwaną liczbę generacji $\mathcal{G}_{fr}$ dla $\mathcal{BAG}$ rozwiązującego testowe problemy dla różnych permutacji kodu. Wartość ta dla permutacji p_k była liczona następująco:

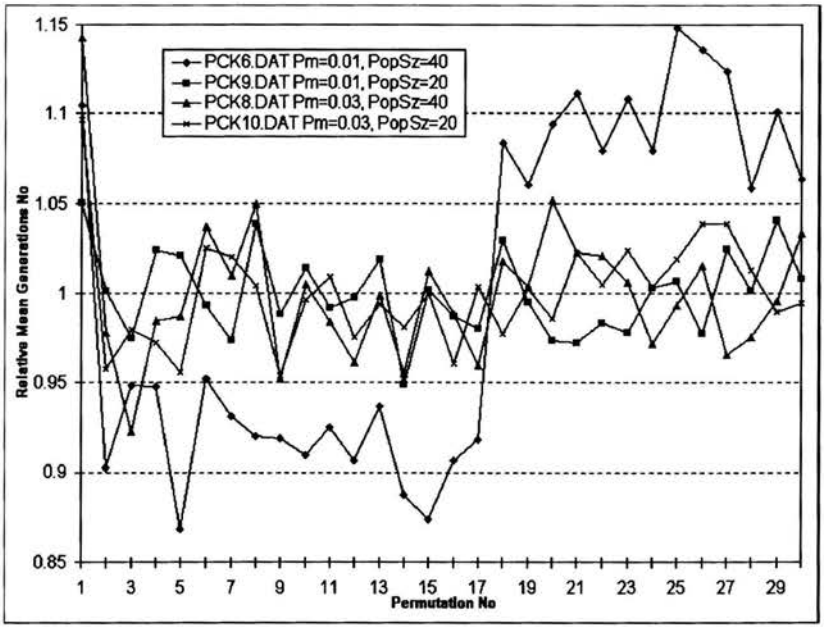
$$\mathcal{G}_{fr}(p_k) = \frac{\mathcal{G}_f(p_k)}{\overline{\mathcal{G}}_f} \tag{8.8}$$

gdzie $\overline{\mathcal{G}}_f = \frac{1}{K} \sum_{i=1}^{K} \mathcal{G}_f(p_i)$. Wartości $\mathcal{G}_f(p_k)$ były przybliżane średnią z 1000-krotnego wykonania $\mathcal{BAG}$ dla każdej z $K = 30$ różnych permutacji kodu, tworzących ciąg testowy $\{p_k\}$.

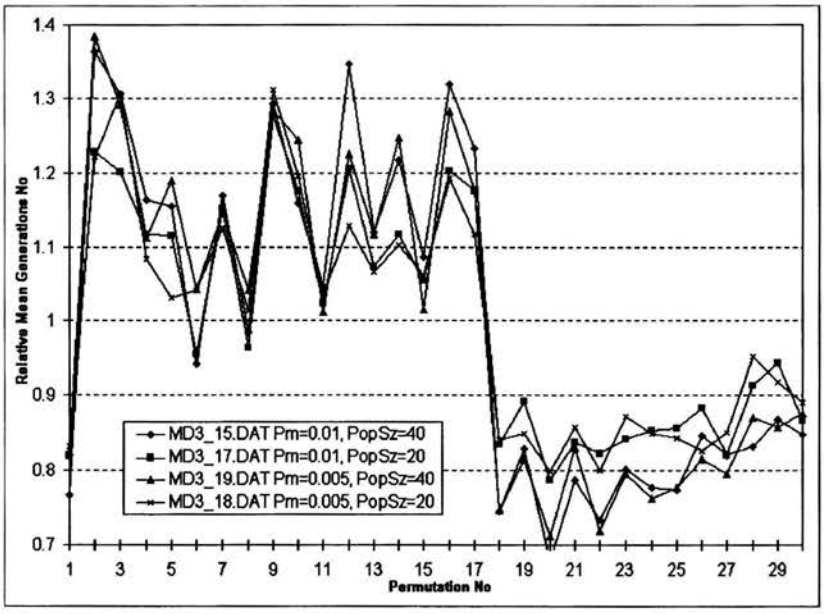
Parametry $\mathcal{BAG}$ dla wykresów z rysunków (8.3)–(8.6) odpowiadają wartościom p_m i $PopSz$ podanym w Tabeli 8.1. Najbardziej istotną cechą każdego z wykresów jest względny zakres przyjmowanych wartości, który może być oszacowany przez obliczenie statystycznego współczynnika



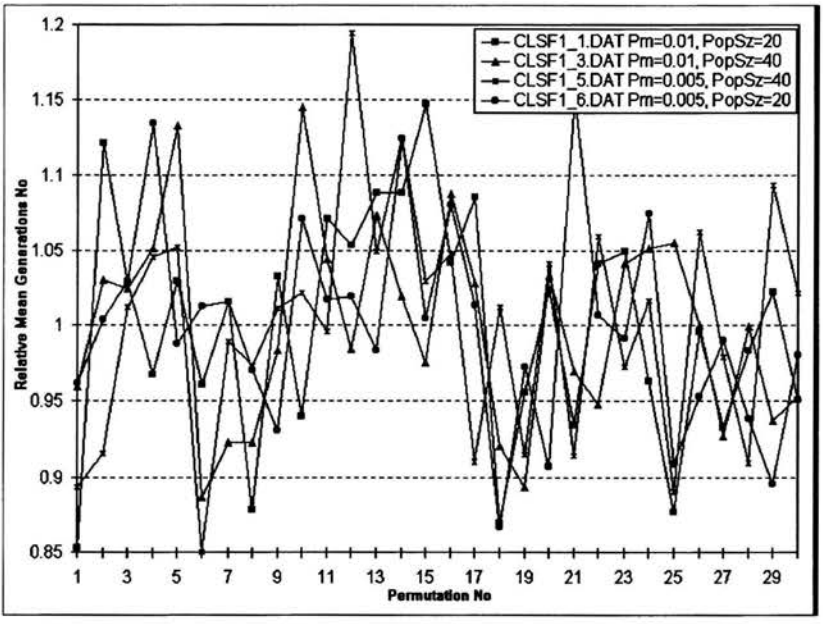
Rys. 8.3: Wpływ permutacji kodu na sprawność *BAG* dla problemu 1.



Rys. 8.4: Wpływ permutacji kodu na sprawność *BAG* dla problemu 2.



Rys. 8.5: Wpływ permutacji kodu na sprawność $\mathcal{BAG}$ dla problemu 3.



Rys. 8.6: Wpływ permutacji kodu na sprawność $\mathcal{BAG}$ dla problemu 4.

Numer problemu	Parametry $\mathcal{BAG}$		$\overline{\mathcal{G}_f}$	$\gamma(\mathcal{G}_f)$	r_{FD}
	p_m	$PopSz$			
1	0.1	40	318	0.208	0.041
	0.05	40	501	0.100	
	0.1	20	308	0.050	
	0.05	20	1196	0.093	
2	0.03	40	305	0.040	-0.433
	0.01	40	226	0.094	
	0.03	20	246	0.029	
	0.01	20	176	0.023	
3	0.01	40	748	0.216	-0.017
	0.005	40	728	0.206	
	0.01	20	983	0.154	
	0.005	20	987	0.154	
4	0.01	40	119	0.065	0.0003
	0.005	40	116	0.075	
	0.01	20	214	0.074	
	0.005	20	202	0.067	

Tablica 8.1: Wpływ permutacji kodu na sprawność $\mathcal{BAG}$ ocenianą metodą standardową. Im większe $\gamma(\mathcal{G}_f)$ tym permutacja kodu ma większy wpływ na sprawność $\mathcal{BAG}$. Dodatkowo podano wartość r_{FD} *(Fitness Distance Correlation)* dla każdego z problemów.

zmienności $\gamma(\mathcal{G}_f)$ ciągu $\{\mathcal{G}_f(p_k)\}$, definiowanego w literaturze (patrz [16]) jako:

$$\gamma(\mathcal{G}_f) = \frac{\sigma(\mathcal{G}_f)}{\mathbf{E}(\mathcal{G}_f)} \tag{8.9}$$

przy czym wartość oczekiwaną $\mathbf{E}(\mathcal{G}_f)$ estymowano średnią $\overline{\mathcal{G}_f}$. Większa wartość współczynnika oznacza, że badany sposób parametryzacji może być bardziej użyteczny w adaptacji $\mathcal{BAG}$ dla danego problemu, gdyż zmiana wartości parametru bardziej wpływa na sprawność algorytmu. W Tabeli 8.1 zamieszczono wartości $\gamma(\mathcal{G}_f)$ odpowiadające przedstawionym wykresom. Widać z nich, że wpływ permutacji kodu na sprawność $\mathcal{BAG}$ jest zależny zarówno od rozwiązywanego problemu, jak i od wartości parametrów p_m i $PopSz$.

8.4.2 Alternatywna metoda oceny sprawności $\mathcal{BAG}$

Zgodnie z opisem bazowego $\mathcal{BAG}$ w Podrozdziale 8.1, przyjęto we wszystkich eksperymentach, że sprawność $\mathcal{BAG}$ jest oceniana przez liczbę generacji koniecznych do osiągnięcia optimum globalnego. W celach porównawczych wykonano dodatkowo kilka eksperymentów z inną, bardzo popularną oceną sprawności $\mathcal{BAG}$, polegającą na pomiarze postępów algorytmu osiągniętych w zadanej liczbie generacji.

W trakcie eksperymentów obliczano, zgodnie ze wzorem (8.9), współczynnik zmienności ciągu $\{\mathcal{G}_f(p_k)\}$, którego wartości pokazano w Tabeli 8.2. Zwraca uwagę przede wszystkim niska wartość tego współczynnika dla wszystkich badanych problemów, niższa o rząd, a niekiedy o dwa rzędy wielkości w stosunku do odpowiednich wartości w Tabeli 8.1. Oznacza to, że permutacja kodu ma zbyt mały wpływ na tak mierzoną sprawność $\mathcal{BAG}$. Można to zjawisko interpretować następująco. Zatrzymanie $\mathcal{BAG}$ na długo przed znalezieniem optimum globalnego powoduje, że badana alternatywna metoda oceny sprawności nie uwzględnia końcowej fazy działania algorytmu, czyli szukania lokalnego.

Z porównania obu metod oceny sprawności wynika, że wpływ wyboru permutacji kodu na działanie $\mathcal{BAG}$ jast największy w fazie szukania lokalnego, natomiast nie wpływa on istotnie na fazę szukania globalnego.

Numer problemu	Parametry $\mathcal{BAG}$		$\gamma(\mathcal{G}_f)$	$\varrho(\mathcal{G}_f, \sigma_{WIDH})$	$\varrho(\mathcal{G}_f, r_{FI})$	$\varrho(\mathcal{G}_f, r_{PO})$
	p_m	$PopSz$				
1	0.1	20	0.0008	-0.065	-0.259	-0.298
2	0.01	20	0.0002	-0.599	-0.862	0.817
3	0.01	40	0.004	-0.254	-0.594	-0.939
4	0.01	20	0.0013	-0.083	-0.274	-0.179

Tablica 8.2: Wartości współczynników korelacji $\varrho(\mathcal{G}_f, \sigma_{WIDH})$ $\varrho(\mathcal{G}_f, r_{FI})$ oraz $\varrho(\mathcal{G}_f, r_{PO})$ dla alternatywnej metody oceny sprawności $\mathcal{BAG}$. Podano dodatkowo współczynnik zmienności $\gamma(\mathcal{G}_f)$ dla ciągu $\{\mathcal{G}_f(p_k)\}$.

8.5 Badanie korelacji funkcji pomiarowych i funkcji sprawności

8.5.1 Eksperyment podstawowy

Badanie korelacji między wynikami pomiaru funkcji przystosowania a sprawnością $\mathcal{BAG}$ przeprowadzono dla trzech funkcji pomiarowych:

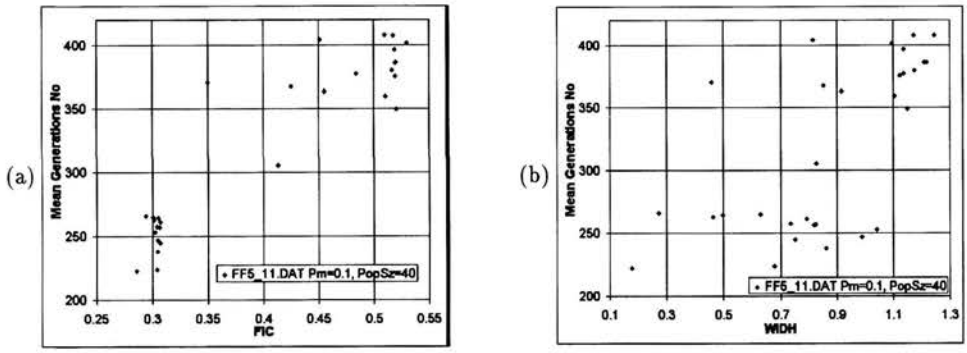
1. σ_{WIDH} – opisanej w Podrozdziale 7.2;
2. r_{FI} – opisanej w Podrozdziale 7.3;
3. r_{PO} – opisanej w Podrozdziale 7.4.

Podczas obliczania σ_{WIDH} przedziały robocze były obliczane na podstawie 500 próbek na każdy bit w chromosomie. Do obliczenia σ_{WIDH} użyto 100 podprzedziałów przedziału roboczego. Został on stworzony na podstawie 500 próbek na bit czyli łącznie 16000 próbek. Oczekiwana liczba $\mathcal{G}_f$ generacji $\mathcal{BAG}$ była przybliżana średnią z 1000-krotnego wykonania $\mathcal{BAG}$. Wartości σ_{WIDH} oraz $\mathcal{G}_f$ obliczano dla każdej z $K = 30$ badanych permutacji kodu (ciągu testowego $\{p_k\}$) i na tej podstawie liczono korelację między nimi, zgodnie ze wzorem (5.15), jako statystykę z próby.

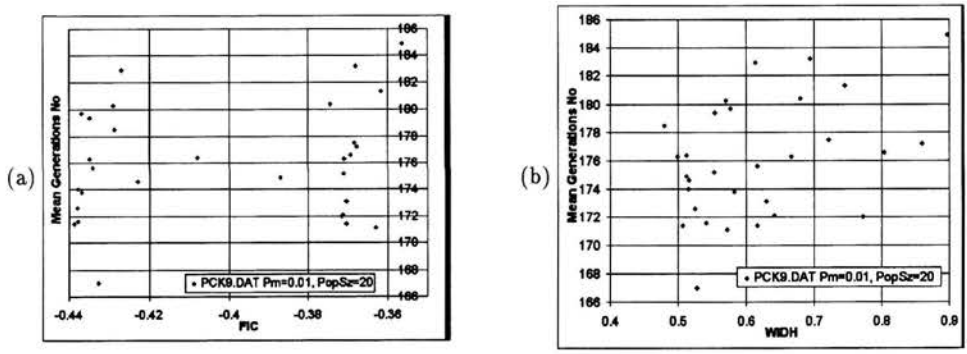
Dla każdego badanego problemu uruchamialiśmy $\mathcal{BAG}$ z kilkoma wartościami prawdopodobieństwa mutacji p_m i rozmiaru populacji $PopSz$. Następnie dla każdego problemu i każdego zestawu parametrów $\mathcal{BAG}$ obliczyliśmy średnią $\overline{\mathcal{G}}_f$ ciągu $\{\mathcal{G}_f(p_k)\}$, korelację $\varrho(\mathcal{G}_f, \sigma_{WIDH})$ pomiędzy ciągami $\{\mathcal{G}_f(p_k)\}$ i $\{\sigma_{WIDH}(p_k)\}$, korelację $\varrho(\mathcal{G}_f, r_{FI})$ między ciągami $\{\mathcal{G}_f(p_k)\}$ i $\{r_{FI}(p_k)\}$ oraz korelację $\varrho(\mathcal{G}_f, r_{PO})$ między ciągami $\{\mathcal{G}_f(p_k)\}$ i $\{r_{PO}(p_k)\}$. Otrzymane wyniki są przedstawione w Tabeli 8.3.

Na wykresach przedstawionych na rysunkach (8.7), (8.8), (8.9) i (8.10) pokazano graficznie wyniki dla wybranego z Tabeli 8.3 jednego zestawu parametrów $\mathcal{BAG}$ dla każdego z problemów. Na pionowych osiach tych wykresów oznaczono wartości $\mathcal{G}_f$ a na poziomych – odpowiednio: σ_{WIDH} – wykresy (a) i r_{FI} – wykresy (b). Każdy znacznik odpowiada wynikom dla jednej permutacji. Im ściślej znaczniki są skupione wokół pewnej prostej, tym wyższe są wartości współczynników korelacji odpowiednio $\varrho(\mathcal{G}_f, \sigma_{WIDH})$ i $\varrho(\mathcal{G}_f, r_{FI})$.

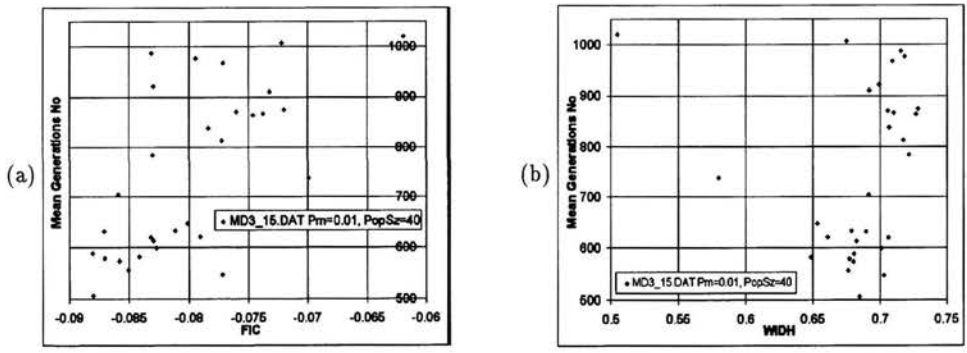
Uzyskane wyniki świadczą o innym zakresie zastosowań każdej badanych funkcji pomiarowych. Funkcja σ_{WIDH} sprawdza się dobrze dla Problemów 1 i 2, średnio – dla 4, a nie nadaje się zupełnie do Problemu 3. Natomiast funkcja r_{FI} sprawdza się dobrze dla Problemów 1, 2, 3, a dla Problemu 4 sprawdza się średnio. Funkcja r_{PO} daje ogólnie najlepsze wyniki dla Problemów 1,3,4, natomiast dla Problemu 2 daje wyniki fałszywe, co oznacza, że nie jest ona uniwersalna.



Rys. 8.7: Rozmieszczenie na płaszczyźnie $(\mathcal{G}_f, \sigma_{WIDH})$ i $(\mathcal{G}_f, r_{FI})$ znaczników odpowiadających badanym permutacjom dla Problemu 1.



Rys. 8.8: Rozmieszczenie na płaszczyźnie $(\mathcal{G}_f, \sigma_{WIDH})$ i $(\mathcal{G}_f, r_{FI})$ znaczników odpowiadających badanym permutacjom dla Problemu 2.



Rys. 8.9: Rozmieszczenie na płaszczyźnie $(\mathcal{G}_f, \sigma_{WIDH})$ i $(\mathcal{G}_f, r_{FI})$ znaczników odpowiadających badanym permutacjom dla Problemu 3.

Numer problemu	Parametry $\mathcal{BAG}$		$\varrho(\mathcal{G}_f, \sigma_{WIDH})$	$\varrho(\mathcal{G}_f, r_{FI})$	$\varrho(\mathcal{G}_f, r_{PO})$
	p_m	$PopSz$			
	0.1	40	0.651	0.927	0.940
	0.05	40	0.370	0.488	0.522
1	0.1	20	0.296	0.269	0.305
	0.05	20	0.209	0.226	0.254
	0.03	40	0.411	0.498	-0.562
	0.01	40	0.758	0.922	-0.756
2	0.03	20	0.506	0.574	-0.577
	0.01	20	0.410	0.177	-0.293
	0.01	40	0.013	0.657	0.911
	0.005	40	-0.037	0.673	0.910
3	0.01	20	0.059	0.644	0.925
	0.005	20	-0.030	0.607	0.894
	0.01	40	0.218	-0.025	0.281
	0.005	40	-0.037	0.458	0.319
4	0.01	20	0.251	0.292	0.453
	0.005	20	0.355	0.196	0.393

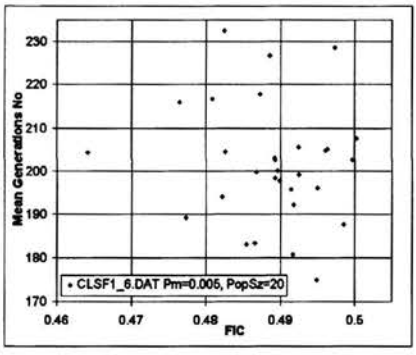
Tablica 8.3: Wartości współczynników korelacji $\varrho(\mathcal{G}_f, \sigma_{WIDH})$ $\varrho(\mathcal{G}_f, r_{FI})$ oraz $\varrho(\mathcal{G}_f, r_{PO})$ dla standardowej metody oceny sprawności $\mathcal{BAG}$. Większa wartość współczynnika korelacji oznacza dana funkcja pomiarowa powinna lepiej oszacować sprawność $\mathcal{BAG}$.

Należy zauważyć, że w żadnym z testowych przypadków dla funkcji pomiarowych σ_{WIDH} i r_{FI} nie stwierdzono znaczącej ujemnej korelacji. Jest to bardzo ważne spostrzeżenie i zaleta, gdyż pozwala mieć nadzieję, że obie funkcje pomiarowe albo dają wyniki pozytywne, albo nie dają żadnych, to jest, nie dają wyników fałszywych.

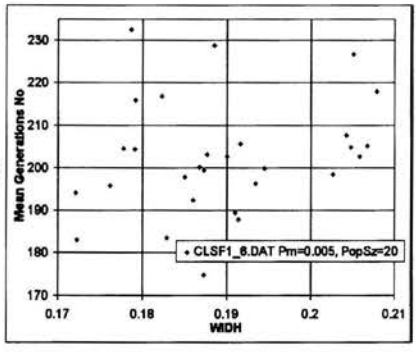
8.5.2 Alternatywna metoda oceny sprawności $\mathcal{BAG}$

Podobnie jak Podrozdziale 8.4.2, przeprowadzono także eksperymenty z alternatywną metodą oceny sprawności $\mathcal{BAG}$. Wyniki przedstawiono w Tabeli 8.2. W większości przypadków okazało się, że występuje przeciwna korelacja między między $\mathcal{G}_f$ a trzema badanymi funkcjami pomiarowymi, co, na podstawie rozumowania przeprowadzonego w Podozdziale 8.4.2, oznacza, że permutacje "lepsze"

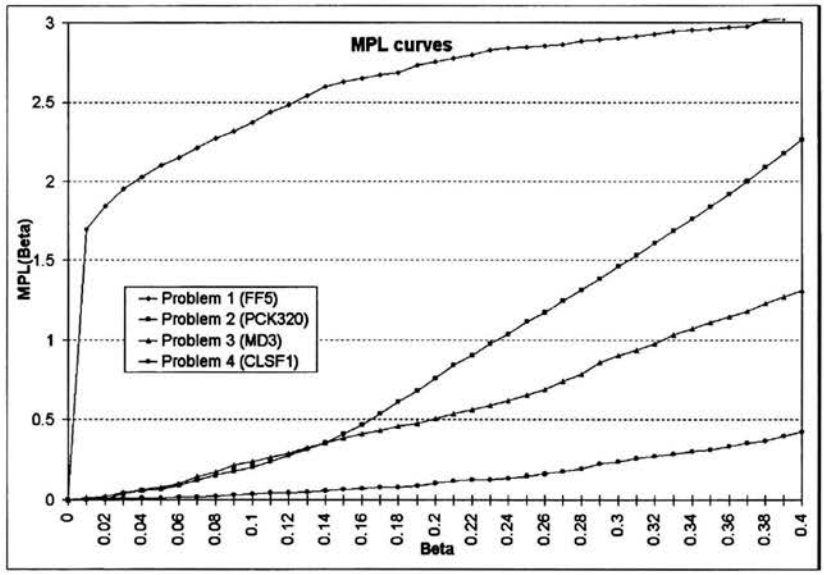
(a)



(b)



Rys. 8.10: Rozmieszczenie na płaszczyźnie $(\mathcal{G}_f, \sigma_{WIDH})$ i $(\mathcal{G}_f, r_{FI})$ znaczników odpowiadających badanym permutacjom dla Problemu 4.



Rys. 8.11: Krzywe MPL dla testowanych problemów. Szybszy wzrost krzywej oznacza większy udział fazy przeszukiwania lokalnego w pracy $\mathcal{BAG}$.

w fazie szukania lokalnego są zwykle "gorsze" w fazie szukania globalnego. Jednakże na podstawie wartości współczynnika zmienności $\gamma(\mathcal{G}_f)$, należy przypuszczać, że to pogorszenie jest znikome.

Ze względu na stwierdzony mały wpływ permutacji na sprawność $\mathcal{BAG}$ mierzoną metodą alternatywną, przeprowadzone w tym zakresie eksperymenty miały ograniczony charakter. Przypuszczalnie taka metoda oceny sprawności nie nadaje się do wykorzystania razem z permutacją kodu jako adaptowanym parametrem, gdyż pomija ona końcową fazę działania $\mathcal{BAG}$, w której wpływ permutacji jest największy.

8.5.3 Pomocnicze metody pomiarowe

Dodatkowo, w celu lepszego zrozumienia otrzymanych wyników, dla każdego z badanych problemów zastosowaliśmy dwie inne, pomocnicze metody pomiarowe. Pierwsza z nich nazywa się *Fitness Distance Correlation*, jest oznaczana przez r_{FD} i została zdefiniowana w pracy Jonesa i Forresta [32].

Obliczenie wartości r_{FD}, pokazanych w Tablicy 8.1, pozwala zaklasyfikować badane problemy do różnych grup trudności, i tak, zgodnie z [32], Problemy 1, 3 i 4 są klasyfikowane jako trudne, a Problem 2 – jako bardzo prosty. Niestety wyniki te nie przybliżają nas w istotny sposób do wyjaśnienia sposobu działania badanych funkcji pomiarowych. Można jedynie zauważyć, że w przypadku Problemu 4, z którym obie funkcje pomiarowe miały trudności, wartość r_{FD} jest najbliższa zeru.

Drugą z zastosowanych metod pomiarowych jest metoda krzywej MPL, zdefiniowana w Rozdziale 7.5.1. Krzywe MPL, wyznaczone dla każdego z badanych problemów, przedstawiono na Rysunku 8.11.

Krzywa dla Problemu 1 pokazuje, że w tym przypadku faza szukiwania lokalnego ma największy udział. Istotnie, jest to zgodne z oczekiwaniami, gdyż pojedyńczy argument jest tutaj kodowany ze stosunkowo dużą dokładnością, bo aż na 32 bitach. Stąd dużo czasu musi zająć ustawianie bitów na mało znaczących pozycjach, co objawia się bardzo łagodnym dojściem do optimum. Krzywe dla Problemów 2 i 3 są bardzo do siebie podobne w zakresie małego sąsiedztwa (małe β) i leżą

pomiędzy krzywymi dla Peoblemów 1 i 4. Natomiast dla Problemu 4 udział szukania lokalnego jest najmniejszy. Może to oznaczać, że dojście do optimum globalnego ma w dużym stopniu charakter losowy, co oznacza, że $\mathcal{BAG}$ nie zbliża się do optimum stopniowo, ale "skacze" w okół niego.

8.6 Badanie skuteczności przykładowej korelacyjnej metody adaptacji

8.6.1 Wprowadzenie

Głównym celem adaptacji parametrów $\mathcal{BAG}$ jest poprawienie jego sprawności, dlatego najbardziej użyteczną metodą sprawdzenia skuteczności danej metody adaptacji jest porównanie sprawności badanego algorytmu użytego bez adaptacji (algorytm odniesienia) i tego samego algorytmu z adaptacją.

Jednakże istotnym problemem jest określenie, jakie powinny być przyjęte wartości adaptowanych parametrów dla algorytmu odniesienia. W przypadku wielu problemów eksperymentator może dosyć łatwo odkryć, jaka wartość adaptowanego parametru zapewnia wystarczającą efektywność algorytmu. W tej sytuacji jest znikoma szansa, że adaptacja spowoduje jakąś znaczącą poprawę działania algorytmu. Z drugiej strony dla pewnych problemów może wystąpić pozornie bardzo duża poprawa sprawności $\mathcal{BAG}$ w wyniku adaptacji. Jednakże prawdziwym powodem tej poprawy może nie być sama adaptacja, ale fakt dopuszczenia innych niż "zwyczajowa" wartość adaptowanego parametru. W przypadku gdy adaptujemy permutację kodu, "zwyczajową" wartością tego parametru jest rosnące uporządkowanie numerów pozycji bitów w chromosomie, które dla pewnych problemów powoduje znaczący spadek sprawności $\mathcal{BAG}$. Aby uniknąć ryzyka wyżej opisanych zafałszowań, proponujemy, aby wartość adaptowanego parametru w algorytmie odniesienia była wybierana losowo, a następnie była ona również wartością początkową w adaptacji.

W przypadku korelacyjnych metod adaptacji wydaje się celowe również porównanie *szacowanego* przyrostu sprawności algorytmu genetycznego z *rzeczywistym* przyrostem sprawności. Oszacowanie przyrostu sprawności uzyskujemy tutaj na podstawie różnicy wartości funkcji pomiarowej dla porównywanych wartości adaptowanego parametru, natomiast rzeczywisty przyrost sprawności – na podstawie wielokrotnego uruchmiania $\mathcal{BAG}$ dla obu wartości parametru. Porównując oba przyrosty jesteśmy w stanie ocenić, w jakim stopniu możemy przewidywać działanie $\mathcal{BAG}$ korzystając z funkcji pomiarowej.

8.6.2 Algorytm pomiarowy

Skuteczność korelacyjnej metody adaptacji będziemy mierzyć na przykładzie wykorzystującego ją metaalgorytmu genetycznego (meta$\mathcal{BAG}$) omówionego w Podrozdziale 6.1.5, którego strukturę przedstawiono na Rysunku 6.2. W wyniku modyfikacji tej struktury, otrzymujemy pokazany na Rysunku 8.12 algorytm pomiarowy.

W trakcie działania powyższego algorytmu wykonujemy V identycznych eksperymentów z różnymi losowymi populacjami początkowymi. Dla i-tego eksperymentu wyznaczamy następujące wskaźniki:

- względny przyrost efektywności rzeczywistej:

$$\zeta_{Ei} = \frac{2(\mathcal{E}_{fW} - \mathcal{E}_{f0})}{\mathcal{E}_{fW} + \mathcal{E}_{f0}}, \tag{8.10}$$

- względny przyrost oszacowania efektywności:

$$\zeta_{Mi} = \frac{2(m_{fW} - m_{f0})}{m_{fW} + m_{f0}}, \tag{8.11}$$

start

for i:=1 **to** V **do**

wylosuj początkową (zerową) populację wartości z P

niech $\mathcal{E}_{f0} = \mathcal{E}(p_{max0})$ i $m_{f0} = m_f(p_{max0})$

wykonaj W pokoleń meta$\mathcal{BAG}$ dla funkcji przystosowania $f(p) = m_f(p)$

niech $\mathcal{E}_{fW} = \mathcal{E}(p_{maxW})$ i $m_{fW} = m_f(p_{maxW})$

oblicz wartości ζ_{Ei}, ζ_{Mi} i κ_{Ei}

end for

stop

Rys. 8.12: Algorytm pomiarowy mający na celu pomiar skuteczności korelacyjnej metody adaptacji wykorzystującej metaalgorytm genetyczny (por. Rys. 6.2). Wartość p_{maxj} oznacza najlepszego osobnika w j-tym pokoleniu meta$\mathcal{BAG}$. Należy zauważyć, że każdorazowe wyliczenie funkcji $\mathcal{E}_f$ wymaga wielokrotnego wykonania badanego algorytmu genetycznego.

Numer problemu	Parametry $\mathcal{BAG}$		$\varrho(\zeta_E, \zeta_M)$	τ_f	$\overline{\zeta}_M$	$\overline{\zeta}_E$	$\overline{\zeta}_M/\overline{\zeta}_E$	$\overline{\kappa}_E$
	p_m	$PopSz$						
1	0.1	20	0.199	0.740	0.311	0.0465	0.150	0.152
	0.1	40	0.507	0.547		0.223	0.717	0.697
2	0.01	40	-0.0151	0.0164	0.0248	-0.0025	-0.101	-0.455
3	0.01	20	0.157	0.244	0.0796	0.0476	0.598	1.247
	0.005	20	0.122	0.201		0.0320	0.402	0.891
4	0.01	20	0.225	0.771	0.045	0.0019	0.0417	-1.619
	0.005	20	0.0039	0.0110		0.0074	0.163	0.0366

Tablica 8.4: Wyniki pomiaru skuteczności korelacyjnej metody adaptacji wykorzystującej meta$\mathcal{BAG}$ z funkcją pomiarową r_{FI} dla czterech problemów z przykładowymi parametrami $\mathcal{BAG}$. Pokazano dodatkowo wartość $\tau_f = \frac{\varrho(\zeta_E,\zeta_M)}{\varrho(\mathcal{G}_f,r_{FI})}$, przy czym wartości $\varrho(\mathcal{G}_f, r_{FI})$ wzięto z Tabeli 8.3.

- wskaźnik adaptowalności:

$$\kappa_{Ei} = \frac{\zeta_{Ei}}{\zeta_{Mi}}, \tag{8.12}$$

gdzie $\mathcal{E}_{fj}$ i m_{fj} są odpowiednio efektywnością i jej oszacowaniem osiągniętymi po W pokoleniach meta$\mathcal{BAG}$, W jest ustaloną liczbą pokoleń meta$\mathcal{BAG}$. Należy zauważyć, że wartości m_{fj} oraz ζ_M mogą być liczone jednorazowo dla każdej z badanych funkcji przystosowania, w przeciwieństwie do pozostałych wartości, które są zależne od przebiegu działania adaptowanego $\mathcal{BAG}$.

Wyznaczone wskaźniki tworzą ciągi V-elementowe, odpowiednio: $\{\zeta_{Ei}\}$, $\{\zeta_{Mi}\}$ i $\{\kappa_{Ei}\}$. Na ich podstawie wyznaczamy wartości średnie $\overline{\zeta}_E$, $\overline{\zeta}_M$ i $\overline{\kappa}_E$, a także korelację ciągów $\{\zeta_{Ei}\}$ i $\{\zeta_{Mi}\}$ oznaczaną przez $\varrho(\zeta_E, \zeta_M)$.

8.6.3 Omówienie uzyskanych wyników

Przedstawiony algorytm pomiaru skuteczności korelacyjnej metody adaptacji pozwala na szczególnie wiarygodną ocene tejże, gdyż pomiary są dokonywane w najbardziej dla niej typowym i naturalnym zastosowaniu.

Ponieważ badania opisane w Podrozdziale 8.5.1 wykazały pewną przewagę funkcji pomiarowej r_{FI} nad pozostałymi funkcjami, dlatego pomiary skuteczności przeprowadzimy jedynie dla r_{FI} jako

bardziej uniwersalnej i obiecującej. Stąd przyjmujemy, że funkcja pomiarowa m_f występująca w algorytmie pomiarowym na Rys. 8.12 jest tożsama z funkcją r_{FI}. Aby odnieść się do wcześniejszych badań właściwości funkcji pomiarowych, w Tabeli 8.4 podano dodatkowo wartości $\varrho(\mathcal{G}_f, r_{FI})$ przeniesione z Tabeli 8.3 oraz wartości ilorazu $\tau_f = \varrho(\zeta_E, \zeta_M)/\varrho(\mathcal{G}_f, r_{FI})$.

Spośród wartości przedstawionych w Tabeli 8.4 najbardziej znacząca z punktu widzenia korzyści związanych ze zwiększeniem sprawności $\mathcal{BAG}$ jest maksymalizacja wartości $\overline{\zeta}_E$ określającej względny wzrost sprawności spowodowany zastosowaniem adaptacji w stosunku do algorytmu odniesienia z losową wartością adaptowanego parametru. Natomiast z punktu widzenia zdolności użytej funkcji pomiarowej do oszacowania rzeczywistej sprawności $\mathcal{BAG}$ najistotniejsza jest maksymalizacja wartości $\varrho(\zeta_E, \zeta_M)$ oraz $\overline{\kappa}_E$.

Z kolei wartość τ_f dostarcza informacji innego rodzaju. Określa on w pewnym stopniu, jak bardzo ciąg testowy $\{p_k\}$ wartości adaptowanego parametru, wykorzystywany w eksperymentach opisanych w Podrozdziale 8.5.1, odbiega w swych właściwościach od próby losowej wziętej ze zbioru P wartości adaptowanego parametru. Można przy tym przyjąć, że $\tau_f = 1$ oznacza, że ma on właściwości zbliżone do próby losowej, zaś im bardziej wartość τ odbiega od 1 tym bardziej tendencyjnie był wybrany ciąg testowy.

Rozdział 9

Uwagi końcowe

9.1 Podsumowanie

W pracy zaproponowano nową metodę adaptacji parametrów $\mathcal{BAG}$ należącą do grupy metod off-line – jest nią korelacyjna metoda adaptacji. Zaproponowano także permutację kodu jako nowy adaptowalny parametr $\mathcal{BAG}$. Zbadano na drodze eksperymentalnej wpływ nowego parametru na sprawność $\mathcal{BAG}$. Następnie opracowano i przetestowano trzy funkcje pomiarowe przeznaczone do oszacowywania sprawności $\mathcal{BAG}$ w korelacyjnych metodach adaptacji. Aby pokazać skuteczność nowej metody adaptacji, zaimplementowano i poddano testom korelacyjną metodę adaptacji wykorzystującą meta$\mathcal{BAG}$ do przeszukiwania przestrzeni parametrów.

Wykonanie powyższych działań oznacza, że zostały zrealizowane wszystkie cele rozprawy. Biorąc także pod uwagę pozytywne wyniki przeprowadzonych eksperymentów obliczeniowych możemy jednoznacznie wykazać słuszność tezy rozprawy.

W trakcie realizacji pracy osiągnięto wiele oryginalnych rezultatów, rzucających nowe światło na sposób działania $\mathcal{BAG}$ oraz na problem adaptacji parametrów $\mathcal{BAG}$. Najważniejsze z nich są następujące:

1. Wykazanie, że permutacja kodu jest parametrem, który istotnie wpływa na sprawność $\mathcal{BAG}$ i który może być łatwo dołączony do dowolnej istniejącej implementacji $\mathcal{BAG}$.

2. Opracowanie sposobu konstrukcji funkcji pomiarowych umożliwiającego wykorzystanie ich do oszacowania sprawności $\mathcal{BAG}$ bez konieczności jego uruchamiania, jedynie na podstawie kształtu funkcji przystosowania. Opracowanie sbosobu łatwej oceny jakości tych funkcji poprzez wyznaczenie ich skorelowania z funkcją sprawności $\mathcal{BAG}$. Bez takich funkcji adaptacja off-line parametrów $\mathcal{BAG}$ byłaby niemożliwa.

3. Skonstruowanie trzech przykładowych funkcji pomiarowych: σ_{WIDH}, r_{FI} oraz r_{PO}, które mogą być skutecznie stosowane do oszacowania sprawności $\mathcal{BAG}$ w korelacyjnych metodach adaptacji.

4. Zaimplementowanie korelacyjnej metody adaptacji wykorzystującej meta$\mathcal{BAG}$ do przeszukiwania przestrzeni parametrów.

5. Pokazanie, że działanie $\mathcal{BAG}$ w trakcie rozwiązywania danego problemu zależy w istotnym stopniu od globalnych właściwości funkcji przystosowania, oraz że wpływ tych właściwości można w pewnym stopniu przewidzieć na podstawie zaproponowanych pomiarów funkcji przystosowania.

6. Pokazanie, iż zależność powyższa jest silna zwłaszcza w fazie szukania lokalnego, w otoczeniu globalnego optimum, natomiast w fazie szukania globalnego jest ona mało istotna.

Oprócz przeprowadzonych badań nad nowymi rozwiązaniami w dziedzinie adaptacji $\mathcal{BAG}$, w pracy podjęto również próbę uporządkowania i uściślenia istniejącej już wiedzy na temat $\mathcal{BAG}$, a zwłaszcza tych aspektów, które są związane z tematem rozprawy. Ze szczególną uwagą zajęto się następującymi zagadnieniami:

1. Teoria funkcji przystosowania oraz funkcji kodującej.
2. Właściwości populacji i schematów.
3. Modele operatorów genetycznych.
4. Teoria uogólnionego krzyżowania.
5. Klasyfikacja metod adaptacji $\mathcal{BAG}$ oraz uściślenie związanych z tym pojęć.

9.2 Kierunki dalszych badań

Na podstawie analizy literatury w dziedzinie algorytmów genetycznych można stwierdzić, iż badania przeprowadzone przez autora w ramach niniejszej rozprawy miały charakter prekursorski. Autorowi nie udało się znaleźć żadnych prac na temat adaptacji off-line parametrów $\mathcal{BAG}$, a ponadto autor spotkał się z sugestiami, że taka adaptacja jest w obecnym stanie wiedzy niemożliwa.

Ze względu na fakt, że prezentowana rozprawa jest pierwszą w swojej dziedzinie, naturalne jest, że nie porusza ona ogromnej listy zagadnień związanych z tą tematyką, ale raczej skupia się na pokazaniu, że prezentowane podejście jest skuteczne i może przynieść korzystne efekty poprzez poprawienie działania $\mathcal{BAG}$. Z drugiej strony, zarówn prezentowana realizacja korelacyjnej metody adaptacji jak i proponowane funkcje pomiarowe, na pewno nie są najlepsze ani najbardziej efektywne. Odnosi się to zwłaszcza do funkcji pomiarowych, które, jak wspominiano, zostały zprojektowane na drodze eksperymentalnej przy wykorzystaniu intuicji autora, natomiast nie korzystają one bezpośrednio z matematycznego opisudziałania $\mathcal{BAG}$. Dlatego konieczne jest prowadzenie dalszych intensywnych badań w tym temacie. Wśród najciekawszych i najbardziej wymagających rozwiązania problemów na uwagę zasługują następujące:

1. Przetestowanie zaproponowanych metod adaptacji na dobrze zdefiniowanych funkcjach przystosowania. Wydaje się, że może być tutaj użyteczny pomysł *NK-modelu* funkcji przystosowania zaproponowany przez Kauffmana w pracy [17], umożliwiający tworzenie funkcji przystosowania o zadanych właściwościach w zakresie gładkości.
2. Zbadanie, na drodze matematycznej analizy działania $\mathcal{BAG}$, od jakich cech kształtu funkcji przystosowania zależy w istocie sprawność $\mathcal{BAG}$. W ten sposób można by było łatwiej tworzyć funkcje pomiarowe, gdyż wystorczyłoby wtedy je tak zaprojektować, aby były wrażliwe na znane cechy.
3. Zastosowanie permutacji kodu w innych wersjach algorytmów genetycznych oraz w innych algorytmach optymalizacji opartych na kodowaniu rozwiązań za pomocą binarnych chromosomów. Dobrym przykładem potencjalnego zastosowania może być praca Jonesa [36] opisująca zmodyfikowany $\mathcal{BAG}$ wykorzystujący makromutację w miejsce krzyżowania.
4. Zbadanie możliwości wykorzystania w korelacyjnej metodzie adaptacji innych metod pomiaru funkcji przystosowania opisywanych w literaturze. Zwłaszcza może być obiecujące wykorzystanie metody polegającej na badaniu metodami analizy sygnałow losowych ścieżek w przestrzeni poszukiwań, zaproponowanej przez Hordijka w pracy [34].

5. Opracowanie, na podstawie teorii permutacji, algorytmu doboru permutacji do rozwiązywanego problemu metodą kolejnych przybliżeń albo metodą przeszukiwania drzewa permutacji. Algorytm taki umożliwiłby ograniczenie przeszukiwania w danym kroku do pewnego odpowiednio małego podzbioru przestrzeni permutacji i mógłby istotnie zmniejszyć złożoność obliczeniową tego procesu.

6. Niewątpliwie najbardziej interesujące byłoby podjęcie próby znalezienia konstruktywnej metody znajdywania optymalnej permutacji kodu do rozwiązywanego zadania, bez konieczności stosowania metod przeszukujących przestrzeń permutacji, takich jak meta$\mathcal{BAG}$.

7. Opracowanie sposobu wykorzystania idei korelacyjnej metody adaptacji w niebinarnych algorytmach genetycznych, na przykład wykorzystujących chromosomy będące wektorami liczb zmiennoprzecinkowych, i innych, opisanych przez Michalewicza w książce [37].

Bibliografia

[1] Shannon C., Weaver W.: *The mathematical theory of communication*, University of Illinois Press, 1949.

[2] Wielandt H.: *Finite Permutation Groups*, Academic Press, New York & Londyn, 1964.

[3] Dymowski S.: *Elementy teorii informacji*, Wyd. Politechniki Warszawskiej, Warszawa, 1968.

[4] Hollstien R.B.: *Artificial genetic adaptation in computer control systems*, Doctoral dissertation, University of Michigan, 1971.

[5] Holland J.H.: "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975.

[6] De Jong K.A.: "An analysis of the behavior of a class of genetic adaptive systems" (Doctoral dissertation, University of Michigan), *Dissertation Abstracts International 36(10)*, 5140B, University Microfilms No. 76-9381, 1975.

[7] Spitzer F.: *Principles of Random Walks*, Springer-Verlag, New York, 1976.

[8] Mercer R.E.: *Adaptive search using a reproducive meta-plan*, master's thesis, University of Alberta, Edmonton, 1977.

[9] Deo N.: *Teoria grafów i jej zastosowania w technice i informatyce*, PWN, Warszawa, 1980.

[10] Brindle A.: "Genetic algorithms for function optimization", Unpublished doctoral dissertation, University of Alberta, Edmonton, 1981.

[11] Czogała E., Pedrycz W.: *Elementy i metody teorii zbiorów rozmytych*, Politechnika Śląska, Gliwice, 1983.

[12] Grefenstette J.J.: "Optimization of control parameters for genetic algorithms", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 16, No. 1, pp. 122-128, 1986.

[13] Wiedermann J.: *Searching Algorithms*, BSD Teubner, Lipsk, 1987.

[14] Schaffer J.D., Morishima A.: "An adaptive crossover distribution mechanism for genetic algorithm", *Proc. of the 2nd International Conference on Genetic Algoritms*, pp. 36-40, Erlbaum Associates, Hillsdale, NJ, 1987.

[15] Goldberg D.E.: "Simple genetic algorithm and the minimal, deceptive problem", in Davies L. (Ed.) *Genetic Algoithms and Simulated Annealing* pp. 74-88, Kaufmann Publishers, San Mateo, CA, 1987.

[16] *Poradnik Inżyniera. Matematyka*, WNT, Warszawa, 1987.

[17] Kauffman S.A.: "Adaptation on Rugged Fitness Landscapes", in D. Stein, editor, *Lectures in the the Sciences of Complexity*, pp. 527-618, Addison-Wesley, 1989.

[18] Spears W.M., De Jong K.A.: "An Analysis of Multi-point Crossover", *Proc. of the Foundations of Genetic Algorithms Workshop*, Indiana, July, 1990.

[19] Hart W.E., Belew R.K.: "Optimizing an arbitrary function is hard for genetic algorithm", *Proc. of the 4th International Conference on Genetic Algoritms*, pp. 190-195, Morgan Kaufmann, Los Altos, CA, 1991.

[20] Harel D.: *Rzecz o istocie informatyki. Algorytmika.*, WNT, Warszawa, 1992.

[21] De Jong K.A., Spears W.M.: "A Formal Analysis of the Role of Multi-point Crossover in Genetic Algorithms", *Annals of Mathematics and Artificial Intelligence Journal*, Vol. 5, No. 1, pp. 1-26. J.C. Baltzer AG Scientific Publishing Company, 1992.

[22] Bäck T.: "The Interaction of Mutation Rate, Selection, and Self-Adaptation within a Genetic Algorithm", *Parallel Problem Solving from Nature* Vol. 2, pp. 85-94, Amsterdam, North Holland, 1992.

[23] Biggs N.L.: *Algebraic Graph Theory*, Cambridge University Press, Cambridge, UK, 1994.

[24] Qi T., Palmieri F.: "Theoretical Analysis of Evolutionary Algorithms With an Infinite Population Size in Continuous Space, Part II: Analysis of the Diversification Role of Crossover", *IEEE Trans. of Neural Networks*, Vol. 5, No. 1, 1994.

[25] Rudolph G.:"Convergence analysis of canonical genetic algorithms", *IEEE Trans. on Neural Networks*, Special Issue on Evolutionary Computation, Vol. 5, No. 1, 1994.

[26] Spears W.M.: "Adapting Crossover in a Genetic Algorithm", *Artificial Intelligence Center Internal Report #AIC-94-025*, Naval Research Laboratory, Washington, DC 20375. Evolutionary Programming IV, McDonnell J.R., Reynolds R.G., and Fogel D.B., editors, pp. 367-384, February, 1995.

[27] Bobrowski D., Łybacka K.: *Wybrane metody wnioskowania statystycznego*, Wyd. Politechniki Poznańskiej, Poznań, 1995.

[28] Goldberg D.E.: *Algorytmy genetyczne i ich zastosowania*, WNT, Warszawa, 1995.

[29] Kolonko M.: "A Generalized Crossover Operation for Genetic Algorithms", *Complex Systems*, pp. 177-191, Vol. 9, 1995.

[30] Stadler P.F.: "Torwards a theory of landscapes", *Complex Systems and Binary Networks*, pp. 77-163, Springer-Verlag, New York, 1995.

[31] Wolpert D., Macready W.: "No free lunch theorems for search", Santa Fe Institute technical report, SFI-TR-02-010, 1995.

[32] Jones T., Forrest S.: "Fitness distance correlation as a measure of problem difficulty for genetic algorithms", *Proceedings of the 6th International Conference on Genetic Algoritms*, Morgan Kaufmann, San Mateo, CA, 1995.

[33] Dixon J.D., Mortimer B.: *Permutation Groups*, Graduate Texts in Mathematics 163, Springer-Verlag, New York, 1996.

[34] Hordijk W.: "A Measure of Landscapes", *Evolutionary Computation*, pp. 335-360, Vol. 4, No. 4, 1996.

[35] Happel R., Stadler P.F.: "Cannonical approximation of fitness landscapes", *Complexity*, Vol. 2, pp. 53-58, 1996.

[36] Jones T.: "Crossover, macromutation, and population-based Search", *Proceedings of the 6th International Conference on Genetic Algoritms*, Morgan Kaufmann, San Mateo, CA, 1996.

[37] Michalewicz Z.: *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, WNT, Warszawa, 1996.

[38] Stadler P.F.: "Landscapes and their correlation functions", *J. Math. Chem.*, pp. 1-45, Vol. 20, 1996.

[39] Cytowski J.: *Algorytmy genetyczne: podstawy i zastosowania*, Seria: Problemy Współczesnej Nauki – Teoria i Zastosowania Nr 18, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1996.

[40] Altenberg L.: "Fitness distance correlation analysis: an instructive counterexample", *Proceedings of the 7th International Conference on Genetic Algoritms*, Morgan Kaufmann, San Mateo, CA, 1997.

[41] Hordijk W., Stadler P.F.: "Amplitude spectra of fitness landscapes", *Advances in Complex Systems*, pp. 39-66, Vol. 1, No. 1, 1998.

[42] Kieś P., Kosiński W.: "On the correlational method of an off-line adaptation", *Colloquia in Artificial Intelligence Theory and Application, CAI'98*, Łódź, Poland, September 1998.

[43] Kieś P.: "Correlational method of an off-line adaptation", *Procs. of the 2nd Asia-Pacific Conf. on Simulated Evolution And Learning*, Canberra, Australia, November 1998.

[44] Kieś P.: "Użycie korelacyjnej metody adaptacji off-line w metaalgorytmie genetycznym", III Krajowa Konferencja Algorytmy Ewolucyjne i Optymalizacja Globalna (KAEiOG'99), Złoty Potok k. Częstochowy, Maj 1999. (wysłana do druku)

[45] Kieś P.: "Correlational adaptation method and genetic meta-algorithm", VIII Midzynarodowe Sympozjum Inteligentne Systemy Informacyjne (IIS'99), Ustroń k. Cieszyna, Czerwiec 1999. (wysłana do druku)