Fragment 3 rozdziału z książki: S.T. Wierzchoń, *Sztuczne systemy immunologiczne. Teoria i zastosowania*. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001

Rozdział 3
Algorytmy genetyczne

Celem tego rozdziału jest prezentacja podstawowych pojęć dotyczących algorytmów genetycznych, które będą wykorzystywane w dalszych partiach książki. Czytelnika zainteresowanego szczegółami technicznymi odsyłamy do literatury źródłowej, np. [71], [135]. Tu natomiast skoncentrujemy się na omówieniu istoty algorytmów genetycznych ze wskazaniem własności genetycznych operatorów selekcji, krzyżowania i mutacji. Omówimy podstawowe dla teorii algorytmów genetycznych twierdzenie o schematach i wynikające zeń konsekwencje. W szczególności zajmiemy się problemem podtrzymywania różnorodności populacji przedstawiając tzw. techniki niszowe. Zdolność podtrzymywania takiej różnorodności zarówno w układzie odpornościowym jak i eko-systemach jest faktem oczywistym. Natomiast algorytmy genetyczne wymagają sporej dozy inwencji do uzyskania zbliżonego efektu. Wreszcie przedstawimy dwa przykładowe zastosowania algorytmów genetycznych do zadań uczenia maszynowego. Pierwszy przykład dotyczy ekstrakcji wiedzy z baz danych i jej reprezentacji w postaci pewnego skierowanego grafu reprezentującego zależności przyczynowo-skutkowe między atrybutami definiującymi schemat bazy danych. Omówimy tu własne rozwiązanie tego problemu. Drugi przykład to tzw. systemy klasyfikujące uznawane przez wielu autorów za najbardziej nowatorską propozycję Johna Hollanda. Wykazują one intrygującą zbieżność z własnościami układu odpornościowego – do problemu tego powracamy pod koniec rozdziału 4.

3.1. Uwagi wstępne

Istotą sformułowanej przez Karola Darwina w 1859 roku teorii ewolucji jest adaptacja rozumiana jako proces przebiegający bez wyznaczonego celu, a polegający na zmianach organizmów wskutek oddziaływania z otoczeniem. Rolą tych zmian jest poprawa sprawności czy też skuteczności danego organizmu. Podkreślmy jednak wyraźnie: sedno darwinowskiej adaptacji polega na znajdowaniu ulepszeń, a nie rozwiązań optymalnych. Ewolucja większości organizmów uwarunkowana jest dwoma mechanizmami: doborem naturalnym i rozmnażaniem płciowym. Pierwszy z tych mechanizmów wyznacza te jednostki, które przeżyją i wydadzą potomstwo, drugi zaś zapewnia wymieszanie i rekombinację genów potomstwa. Na dobór naturalny można spoglądać jak na „arenę krwawych walk na kły i pazury", ale można go też traktować jako przejaw ogólniejszego prawa „przetrwania najstabilniejszych", jak pisze o tym Dawkins, [40].

Pierwsze próby połączenia teorii ewolucji z informatyką prowadzono na przełomie lat pięćdziesiątych i sześćdziesiątych. Jak stwierdza Holland, [101], brak sukcesów wynikał z naśladownictwa ówczesnych podręczników biologii, które kładły większy nacisk na rolę mutacji jako źródła zmienności genetycznej w porównaniu z reprodukcją płciową. Widocznym przełomem była zaproponowana przez Hollanda w połowie lat sześćdziesiątych technika programowania uwzględniająca ewolucję zarówno przez mutację, jak i krzyżowanie się. W następnym dziesięcioleciu zakres stosowania tego algorytmu został poszerzony o kod genetyczny pozwalający reprezentować strukturę każdego problemu. W ten sposób powstał uniwersalny algorytm znany pod nazwą **algorytmu genetycznego**.

Odnotujmy na marginesie, że niemal równolegle z pracami Hollanda, w Niemczech prowadzono prace nad tak zwanymi strategiami ewolucyjnymi wykorzystującymi mechanizmy selekcji i mutacji naturalnej. O ile algorytmy genetyczne można traktować jako algorytmy ogólnego przeznaczenia, o tyle strategie ewolucyjne zorientowane są na rozwiązywanie zadań optymalizacji parametrycznej. Postęp w obu dziedzinach zatarł wiele z dzielących je różnic. Obecnie często mówi się o algorytmach ewolucyjnych rozumiejąc pod tym pojęciem dowolny algorytm przetwarzający populację rozwiązań, która to populacja modyfikowana jest w sposób naśladujący biologiczne narodziny i śmierć, a reprodukowane osobniki nie są dokładnymi kopiami swoich rodziców.

Prostym przykładem algorytmu ewolucyjnego jest algorytm dopasowywania do wzorca. Algorytm ten przetwarza populację łańcuchów znaków inicjowanych losowo, a celem ewolucji jest uzyskanie łańcucha o zadanej postaci, powiedzmy „*Dobór naturalny to krwawa walka na kły i pazury*". Dopasowanie dowolnego łańcucha z populacji do wzorca określa liczba pozycji, na których w obu łańcuchach występują identyczne litery. W każdej iteracji, po wymieszaniu osobników, populację dzieli się na małe grupy. Najlepszy (w

opisanym wyżej sensie) osobnik w każdej grupie jest kopiowany i zastępuje najgorszego osobnika w tej grupie, przy czym losowo wybrana pozycja kopii poddawana jest mutacji, tzn. występujący tam znak zmieniany jest na inny, wybrany losowo. Opisana tu operacja odzwierciedla „niedokładną reprodukcję" charakterystyczną dla naturalnej ewolucji. Nietrudno zauważyć, że populacja łańcuchów będzie z upływem czasu (mierzonego liczbą „pokoleń", czyli iteracji) dążyć do populacji zawierającej wiele łańcuchów zgodnych ze wzorcem. Przeszukiwanie realizowane jest tu wyłącznie drogą wprowadzania „niewielkich zmian", czyli mutacji. Wprowadzając dodatkowy operator stanowiący odpowiednik reprodukcji płciowej, czyli operator krzyżowania (por. punkt 3.2.5) uzyskujemy algorytm genetyczny. W rozważanym tu przykładzie operator ten można zrealizować w sposób następujący. Z każdej grupy wybiera się dwa najlepsze osobniki, nazywane rodzicami, a następnie wybiera się losowo numer pozycji w łańcuchu. Pierwszy potomek tworzony jest z liter pierwszego łańcucha kopiowanych od początku do ustalonej wcześniej pozycji, a resztę liter pobiera się od drugiego rodzica poczynając od kolejnej pozycji. Drugi potomek tworzony jest analogicznie, z tym, że jego początkowe litery pobierane są od drugiego rodzica, a końcowe od pierwszego.

3.2. Operatory genetyczne

Algorytm genetyczny ma dobrze znaną postać – por. [71], [135]. Po wybraniu odpowiedniej dla rozwiązywanego problemu reprezentacji osobników (por. punkt 3.2.1) inicjuje się początkową ich populację (punkt 3.2.2), a następnie powtarza określoną (poprzez wybranie warunku zatrzymania iteracji) ilość razy następujące operacje:

- po pierwsze ocenia się jakość osobników z aktualnej populacji (stosując tzw. funkcję dostosowania – por. punkt 3.2.3),
- po drugie tworzy się nową populację stosując operatory selekcji, krzyżowania i mutacji omówione odpowiednio w punktach 3.2.4, 3.2.5 i 3.2.6.

3.2.1. Definicja osobników

Pierwszym krokiem implementacji algorytmu genetycznego jest określenie struktury danych dla osobników tworzących populację podlegającą ewolucji. Struktura ta określana jest często terminem **chromosom**. Jak stwierdza Goldberg, wybór takiej struktury „nie stanowi żadnego problemu, gdyż programista jest ograniczony głównie swoją własną wyobraźnią" ([71] str. 95). Tym niemniej przy wyborze struktury należy kierować się z jednej strony jej uniwersalnością, a z drugiej jej prostotą. Przez uniwersalność struktury rozumiemy to, że powinna ona umożliwiać reprezentację potencjalnych rozwiązań nie tylko pojedynczego zadania, ale przynajmniej dowolnego zadania należącego do określonej klasy problemów. Na przykład struktura powinna umożliwiać reprezentację rozwiązań dowolnego zadania optymalizacji funkcji rzeczywistej n zmiennych. Przez jej prostotę rozumiemy łatwość posługiwania się taką strukturą, przede wszystkim łatwość wyznaczania dostosowania konkretnego osobnika.

Jedną z zasad wyboru struktury jest zasada minimalnego alfabetu, tzn. należy wybrać najmniejszy alfabet, w którym dane zadanie wyraża się w sposób naturalny, [71]. Najbardziej popularną strukturą jest łańcuch binarny o długości l. Długość ta – dla każdego osobnika populacji – może być albo stała, albo zmienna. Chromosomy o stałej długości są znacznie prostsze, ale chromosomy o zmiennej długości są bardziej elastyczne, gdyż umożliwiają bardziej naturalną reprezentację konkretnych rozwiązań. Reprezentacją wygodną do rozwiązywania zadań optymalizacji funkcji n zmiennych jest przyjęcie za chromosom n wymiarowego wektora o składowych rzeczywistych. Z kolei w zadaniach uczenia maszynowego chromosomy reprezentują albo pojedyncze reguły (jest to tzw. **podejście Michigan**, które można traktować jako komputerowy model świadomości), albo też cały zestaw reguł, czyli oddzielną świadomą jednostkę

(tzw. **podejście Pitt)**[1]. Jeżeli natomiast każdy chromosom reprezentuje pojedynczy program – mamy do czynienia z tzw. **programowaniem genetycznym**, [126].

Bez względu na to, jak złożony jest chromosom przyjmuje się, że składa się on z liniowo uszeregowanych **genów** (tzn. *znaków*, *cech* lub *dekoderów*), które mogą występować w pewnej liczbie odmian nazywanych **allelami**[2]. Pozycję genu w chromosomie określa się terminem **locus**.

Zestaw genów charakteryzujący pojedynczego osobnika określany jest w biologii **genotypem**. W algorytmach genetycznych przyjmuje się zazwyczaj, że mamy do czynienia z osobnikami *haploidalnymi*[3], tzn. osobnikami wyposażonymi w pojedynczy chromosom. Jak pisze Dawkins „(...) dobór darwinowski nie może działać bezpośrednio na geny. Wszystkie geny wyglądają podobnie, tak samo jak podobne do siebie są taśmy magnetofonowe. Istotne różnice między genami ujawniają się w skutkach ich działania. (...) Termin '**fenotyp**' używany jest dla określenia cielesnej manifestacji genu – jego wpływu na rozwój osobnika, w porównaniu do wpływów wywieranych przez jego allele."[4] W rozpatrywanym w tym rozdziale „cyfrowym" świecie, jeżeli genotypem jest na przykład łańcuch binarny, to fenotypem jest jego reprezentacja dziesiętna.

3.2.2. Populacja początkowa

Populacja początkowa stanowi zbiór chromosomów, które będą poddawane ewolucyjnym modyfikacjom. Przy jej tworzeniu pożądane jest zachowanie jak największej różnorodności osobników, co sprzyja szybkiej ocenie możliwie dużej liczby potencjalnych rozwiązań, bez nadmiernego zwiększania rozmiaru populacji.

Najpopularniejszą i najprostszą metodą tworzenia populacji początkowej jest losowa generacja chromosomów. W pewnych sytuacjach metoda taka nie jest efektywna, gdyż wiele osobników może nie spełniać ograniczeń występujących w rozwiązywanym zadaniu (przykładem może być zadanie programowania liniowego lub nieliniowego polegające na wyznaczeniu optimum funkcji przy dodatkowych ograniczeniach równościowych i nierównościowych – por. rozdział 7 w [135]). W takich przypadkach dobrze jest odwołać się do wiedzy o rozwiązywanym problemie i „zaprojektować" zestaw chromosomów[5]. Nawet jeżeli populacja tworzona jest losowo, wiedzę o problemie można wykorzystać do wyboru rozkładu prawdopodobieństwa, z którym będą wybierane poszczególne allele[6]. Inna metoda polega na wykorzystaniu wyników poprzedniego przebiegu algorytmu genetycznego do zainicjowania populacji. Wreszcie, czasem wystarczy wprowadzenie do populacji kilku specjalnie zaprojektowanych osobników i zapełnienie reszty populacji osobnikami losowymi.

Ważnym parametrem algorytmu jest liczebność populacji. Jeżeli będzie ona zbyt mała, algorytm może zbiegać zbyt szybko; jeżeli będzie zbyt duża – niepotrzebnie będzie zużywać się czas i zasoby komputerowe. Liczebność populacji może być stała we wszystkich iteracjach, albo też może ulegać zmianom – problemy te omówiono w punkcie 4.4 książki [135].

3.2.3. Ocena dostosowania

Zasadniczym etapem każdej iteracji algorytmu jest obliczenie tzw. **dostosowania** osobników wchodzących w skład modyfikowanej populacji. Dostosowanie jest pewną miarą obrazującą, jak dobrze określony osobnik „radzi sobie" z rozwiązywanym problemem. Właściwy dobór dostosowania jest kluczowym elementem algorytmu genetycznego. Jest to łącznik między abstrakcyjnym (uniwersalnym) algorytmem powielającym

[1] Krótki przegląd tych zagadnień można znaleźć w rozdziale 12 książki [135].
[2] Są to w gruncie rzeczy wartości cech. W przypadku łańcuchów binarnych allele to symbole **0**, **1**.
[3] Informację na temat *diploidalności*, czyli podwójnego zestawu chromosomów i dominowania, czyli ważonych zależności między genotypem a fenotypem, można znaleźć w rozdziale 5 książki [71].
[4] Por. [40], str. 321.
[5] W literaturze anglosaskiej używa się terminu *seeding the run* na określenie tak inicjowanej populacji.
[6] Por. punkt 7.3.3.1

opisane wcześniej kroki a rzeczywistym problemem. W przypadku optymalizacji funkcji za miarę dostosowania można przyjąć wartość optymalizowanej funkcji. W wielu jednak problemach, np. uczenia maszynowego, dobór miary dostosowania nie jest sprawą trywialną. Z uwagi na fakt, że dostosowanie obliczane jest dla każdego osobnika i w każdej iteracji algorytmu, funkcja czy też procedura realizująca pomiar dostosowania powinna być możliwie prosta. Powinna jednocześnie odzwierciedlać charakterystyki istotne dla poszukiwanego rozwiązania. W pewnych, szczególnie złożonych, sytuacjach można wyobrazić sobie dwuetapowy pomiar dostosowania osobników. W początkowych iteracjach dostosowanie to można liczyć w sposób przybliżony, stosując uproszczoną procedurę, aby możliwie szybko zlokalizować obszary występowania potencjalnych rozwiązań. Dopiero w dalszych iteracjach, gdy próbne rozwiązania poddawane są kolejnym modyfikacjom genetycznym, można obliczać precyzyjnie dostosowanie proponowanych rozwiązań.

Istnieją dwa typy modyfikacji, którym mogą podlegać osobniki początkowych populacji. Pierwszy z tych typów, nazywany po prostu **ewolucją**, polega na niezależnym pomiarze dostosowania każdego osobnika. Jako miarę jakości przyjmuje się tu jego bezwzględne dostosowanie do wymogów stawianych poszukiwanemu rozwiązaniu. W drugim typie, nazywanym **koewolucją**, dostosowanie pojedynczego osobnika uwzględnia własności pozostałych członków populacji. Omawiane w punkcie 3.5 metody niszowe, stosowane w optymalizacji wielomodalnej, są najprostszym przykładem koewolucji. Obecność dużej liczby osobników o dużym dostosowaniu skoncentrowanych wokół pewnego punktu wpływa na obniżenie ich dostosowania „postrzeganego" przez algorytm. W ten sposób populacja koewoluuje ze sobą, co prowadzi (przynajmniej w zamyśle) do oczekiwanego rezultatu. Podany przykład jest tylko niewielką modyfikacją „standardowego" algorytmu ewolucyjnego. Niejako klasycznym już przykładem bardziej wyrafinowanego traktowania koewolucji jest praca [93] poświecona symulowaniu zachowań pasożytów, omówiona w punkcie 3.5.3.

Przypisując każdemu osobnikowi populacji jego dostosowanie możemy mówić o **pejzażu dostosowań** – przestrzeni, w której istnieje i ewoluuje dana populacja. Aby prawidłowo przedstawiać taką przestrzeń, należy określić najpierw relację podobieństwa na zbiorze wszystkich struktur. Pojęcie pejzażu dostosowań używane jest przez biologów od lat 30-tych dwudziestego wieku do opisu ewolucji genotypów jako procesu poszukiwania, w którym zmiany genetyczne gatunków powodują ich przemieszczanie się między „pikami" w przestrzeni genotypów.

3.2.4. Modele ewolucji

Pod pojęciem „metoda ewolucji" rozumiemy metodę wyboru osobników do tzw. **puli rodzicielskiej**. Osobniki tworzące tę pulę poddawane są następnie operacjom krzyżowania i mutacji prowadzącym do utworzenia potomnej populacji osobników przetwarzanej w kolejnej iteracji algorytmu. Wybór właściwego modelu ewolucji znacząco wpływa na jakość algorytmu.

Najprostszym typem selekcji jest **selekcja proporcjonalna** wykorzystująca **regułę ruletki**. Rodzice wybierani są z prawdopodobieństwem proporcjonalnym do ich dostosowania. Jeżeli dostosowanie i-tego osobnika oznaczymy przez f_i, to prawdopodobieństwo jego selekcji jest równe $p_i = f_i/F$, gdzie F oznacza sumę dostosowań wszystkich osobników populacji. Oczywiście taki sposób określania prawdopodobieństwa jest poprawny tylko wówczas, jeżeli dostosowanie każdego osobnika jest liczbą nieujemną, a przynajmniej jednego osobnika – liczbą dodatnią. W sytuacjach, gdy dostosowanie pojedynczego osobnika wyraża się liczbą rzeczywistą, konieczne jest jego **skalowanie**. Problem ten omówiono wyczerpująco w [71]. Wadą selekcji proporcjonalnej jest towarzyszący jej stosunkowo duży błąd stochastyczny selekcji. Brindle, [22], obliczyła wariancję tego błędu: $\sigma_i^2 = n \cdot p_i \cdot (1-p_i)$, gdzie n oznacza rozmiar populacji. Aby zmniejszyć tę wariancję, zaproponowała ona metodę nazwaną wyborem losowym wg reszt z powtórzeniami[7]; w tym wypadku wariancja jest równa $\sigma_i^2 = r_i \cdot (1-r_i)/(\sum_{i=1,\ldots,n} r_i)$, gdzie $r_i = n \cdot p_i \cdot (1- n \cdot p_i)$, jest prawdopodobieństwem

[7] Por. [71], str. 136-137.

resztowego wyboru osobników. Inną metodą redukcji wariancji jest **uniwersalna selekcja stochastyczna**, w której tarcza ruletki dzielona jest na m równych części, gdzie m oznacza liczbę osobników przeznaczonych do wylosowania, [9]. W przypadku tej startegii wariancja jest równa zero, a liczba kopii danego osobnika jest nie mniejsza niż $\lfloor n \cdot p_i \rfloor$ i nie większa niż $\lceil n \cdot p_i \rceil$.

Pojedyncza selekcja turniejowa polega na „wymieszaniu" populacji i podzieleniu jej na małe grupy najczęściej złożone z czterech osobników. Z każdej grupy wybiera się dwóch osobników o najwyższym dostosowaniu. Pary wybranych w ten sposób osobników stanowią rodziców przyszłego potomstwa. Taki mechanizm wyboru rodziców posiada szereg zalet. Po pierwsze, w przypadku małych grup o rozmiarze k zapewnia to szansę przetrwania k-2 najlepszym osobnikom, co z kolei gwarantuje, że dostosowanie populacji nie pogarsza się w trakcie kolejnych iteracji. Po drugie, bez względu na to, jak dobrze dostosowany jest osobnik w porównaniu z resztą populacji, produkuje on co najwyżej jednego potomka w każdej iteracji. Własność ta zapobiega przedwczesnej zbieżności populacji – problemowi, z którym boryka się większość modeli ewolucji. W selekcji proporcjonalnej osobnik o wysokim dostosowaniu dominuje początkowe etapy ewolucji, co skutecznie ogranicza obszar poszukiwań i powoduje zbieżność populacji do rozwiązań suboptymalnych.

Jeżeli przy wyborze rodziców dba się o to, aby najlepiej dostosowane osobniki, nazywane **elitą**, były zawsze członkami puli rodzicielskiej, strategię taką określa się mianem **elitarnej**. Strategia ta powoduje, że w dalszych iteracjach maksymalne dostosowanie w populacji nie będzie zbyt wysokie, ale potomstwo elity będzie dominować przyszłe populacje. Może to ograniczać zdolności eksploracyjne algorytmu, gdyż początkowa elita niekoniecznie zawiera geny charakteryzujące najlepsze osobniki. Rozwiązaniem jest utrzymywanie elity o niewielkim rozmiarze, podobnie jak ma to miejsce w pojedynczej selekcji turniejowej, gdzie elita składa się z dwóch osobników.

W **podwójnej selekcji turniejowej** wybiera się najpierw grupę złożoną z k osobników, a następnie z grupy tej wybiera się najlepiej dostosowanego osobnika i traktuje go jako pierwszego rodzica. W celu wyboru drugiego rodzica procedura jest powtarzana. Selekcja rodziców może albo dopuszczać powtórzenia (ten sam osobnik jest pierwszym i drugim rodzicem), albo nie. W drugim przypadku mówimy o selekcji **bez powtórzeń**.

Selekcja rangowa (porządkowa) jest podobna do selekcji według reguły ruletki z tym, że osobniki są sortowane według ich dostosowania, a ich rangi pełnią rolę funkcji dostosowania.

Dysponując pulą rodzicielską należy opracować strategię wprowadzania potomków do nowej populacji. Można to czynić zwiększając rozmiar populacji (np. w ciągu kilku iteracji dołączać potomków do starej populacji, a po pewnej ich liczbie usunąć „nadmiarowych" osobników wracając do początkowego rozmiaru populacji), albo też w każdej iteracji zachowywać stały rozmiar populacji. W tym drugim przypadku najprostszą, **losową**, strategią jest wstawianie potomka w losowo wybrane miejsce. Jeżeli wybieranie osobników do zastąpienia odbywa się z prawdopodobieństwem odwrotnym do ich dostosowania, mamy do czynienia z **zastępowaniem według reguły ruletki**. Odpowiednikiem **rangowej strategii zastępowania** jest uszeregowanie osobników w kolejności odwrotnej niż w selekcji rangowej i wybieranie ich z prawdopodobieństwem proporcjonalnym do przydzielonej rangi. Inna metoda to **zastępowanie według absolutnego dostosowania**: najsłabiej dostosowane osobniki starej populacji zastępowane są przez nowe osobniki. Kolejna metoda polega na zastępowaniu starych osobników nowymi tylko wówczas, gdy nowe osobniki mają wyższe dostosowanie niż stare. Mianowicie, dwóch rodziców porównuje się z ich potomstwem; potomek zastępuje rodzica tylko wtedy, gdy jest od niego lepszy. Jest to tzw. **lokalne zastępowanie elitarne**. W **losowym zastępowaniu elitarnym** potomek porównywany jest z losowo wybranym rodzicem.

W każdej z omówionych strategii na początku podejmuje się decyzję, jak wiele par rodziców należy wybrać do stosownego porównania. Krańcowe rozwiązania polegają na zastąpieniu wszystkich rodziców, albo tylko

jednej pary rodziców czy nawet jednego rodzica. W pierwszym przypadku mówimy o **pokoleniowym algorytmie gene-tycznym**, a w drugim – o **algorytmie ze stanem ustalonym**.

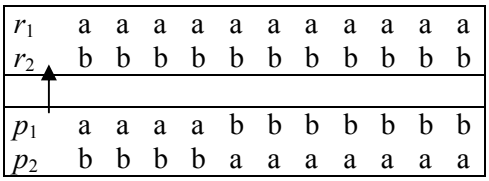
3.2.5. Krzyżowanie

Operacja krzyżowania jest najbardziej charakterystyczną cechą algorytmów genetycznych. Pod pojęciem krzyżowania osobników z populacji G rozumiemy odwzorowanie

$$\kappa: G \times G \to G \times G$$

Parę punktów z dziedziny odwzorowania κ nazywamy rodzicami, zaś parę punktów z jego przeciwdziedziny nazywamy potomkami. Rolą operatora krzyżowania jest wymiana informacji między rodzicami realizowana z nadzieją otrzymania potomstwa o wyższej jakości.

Najczęściej stosowanym operatorem krzyżowania jest krzyżowanie jednopunktowe przedstawione na rysunku 3.1.

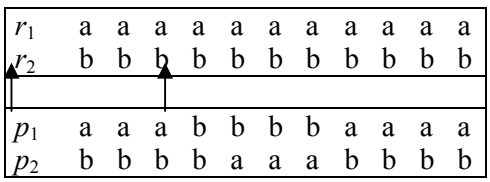


Rysunek 3.1. Krzyżowanie jednopunktowe

W górnej części rysunku przedstawiono parę rodziców (r_1, r_2). Strzałką zaznaczono losowo wybrany punkt rozcięcia łańcuchów. Pierwszy potomek powstaje przez kopiowanie genów pierwszego rodzica do punktu rozcięcia; resztę genów pobiera się od drugiego rodzica. Drugi potomek tworzony jest w analogiczny sposób z tym, że początkowy segment tworzą geny drugiego rodzica, a końcowy – geny pierwszego rodzica.

Stosując tę procedurę zauważamy bez trudu, że pozycje bliskie siebie mają mniejszą szansę rozerwania niż pozycje oddalone. W celu redukcji tego zjawiska wprowadza się krzyżowanie n-punktowe, w którym wybiera się n punktów, a potomków tworzy się przez naprzemienne wybieranie sekwencji genów od obu rodziców. Na rysunku 3.2 pokazano realizację krzyżowania 2-punktowego.

Aby zapewnić jednakową szansę wymiany wszystkich pozycji łańcucha, stosuje się krzyżowanie jednostajne, w którym każdy gen jest z jednakowym prawdopodobień-stwem wybierany od obu rodziców, tzn. przed wypełnieniem kolejnej pozycji rzucamy monetą: „orzeł” nakazuje pobrać gen od pierwszego rodzica, „reszka” – od drugiego. Metoda taka jest kosztowna z obliczeniowego punktu widzenia, gdyż wymaga wielokrotnego stosowania generatora liczb pseudolosowych.



Rysunek 3.2. Krzyżowanie dwupunktowe

W sytuacjach praktycznych może się okazać się, że różne typy krzyżowania są skuteczne w różnych etapach procesu obliczeniowego. W takich sytuacjach wygodnie jest stosować **krzyżowanie adaptacyjne**. Każdy osobnik wyposażony jest we wzorzec, tzn. binarny łańcuch wskazujący, od którego rodzica należy pobierać określony gen. Np. zero na i-tej pozycji wzorca oznacza, że i-tą pozycję potomka należy zapełnić genem pochodzącym od pierwszego rodzica, a jedynka – od drugiego rodzica. Wzorce podlegają krzyżowaniu i

mutacji w celu dostosowania strategii krzyżowania rodziców. Zatem wzorce koewolują wraz z osobnikami populacji. Operacja taka powinna skierować ewolucję w te obszary, które przyspieszają rozwiązanie właściwego problemu. Modyfikowane wzorce mogą zawierać użyteczną i nietrywialną informację o rozwiązywanym problemie.

Na zakończenie warto wspomnieć o **pustym krzyżowaniu**, kiedy to potomstwo jest kopiami rodziców. Puste krzyżowanie może być użyteczne w mieszanych strategiach krzyżowania i badaniu poprawności programu. Wreszcie pod pojęciem **krzyżowania konserwatywnego** rozumiemy takie, w którym krzyżowanie identycznych rodziców produkuje identyczne potomstwo.

Przedstawione tu operatory krzyżowania są operatorami „ogólnego przeznaczenia". Przy rozwiązywaniu specjalizowanych zadań, np. zadań optymalizacji dyskretnej, należy korzystać z operatorów uwzględniających specyfikę konkretnego zadania – por. [135], rozdziały 9 i 10.

3.2.6. Mutacja

Przez mutację osobników populacji G rozumiemy odwzorowanie

$$\mu: G \to G$$

przekształcające chromosom w inny chromosom.

O ile rolą krzyżowania jest wymiana informacji między osobnikami, to mutacja powoduje niewielkie modyfikacje pojedynczych osobników. W algorytmach genetycznych modyfikacje te wprowadzane są z bardzo małym prawdopodobieństwem.
Przez **punktową mutację** rozumiemy zmianę pojedynczego (wybranego losowo) bitu. W mutacji **wielopunktowej** losujemy kilka pozycji i zamieniamy na przeciwne występujące na nich bity. **Probabilistyczna** mutacja polega na mutowaniu każdego bitu z prawdopodobieństwem α. Przez mutację **użyteczną** rozumiemy operację polegającą na sprawdzeniu jakości chromosomu przed wykonaniem mutacji jednego z omówionych wyżej rodzajów oraz po wykonaniu tejże mutacji i wybraniu chromosomu o wyższym dostosowaniu. Mutacja **lamarckowska** rzędu k polega na sprawdzeniu wyników mutacji m-punktowych, $0 \le m \le k$ i wybraniu tej, która zapewnia najwyższy stopień dostosowania. Wreszcie **pusta** mutacja to taka, która nie zmienia chromosomu.

Podobnie jak w przypadku operatorów krzyżowania, tak i tutaj opracowano szereg specjalizowanych operatorów dostosowanych do konkretnych zadań. Przykładem może być mutacja lamarckowska stosowana w optymalizacji różniczkowalnych funkcji n zmiennych. Do każdego elementu populacji dodaje się wektor

$$r \cdot \frac{\nabla f}{\| \nabla f \|}$$

gdzie r jest liczbą losową z przedziału [-1, 1], symbol ∇f oznacza gradient funkcji f, a symbolem $\|\mathbf{a}\|$ oznaczono normę, czyli długość wektora **a**.

3.2.7. Krzyżowanie czy mutacja?

Operator krzyżowania jest operatorem odróżniającym algorytmy genetyczne od innych algorytmów ewolucyjnych. Jego rola jest jednak krytykowana przez wielu badaczy[8]. „Przedwczesna zbieżność" algorytmu, jak określa się eufemistycznie zbyt szybką utratę różnorodności populacji, zdaniem wielu autorów, np. [171], jest konsekwencją krzyżowania. Środkiem służącym podtrzymaniu właściwej dywersyfikacji populacji, a w konsekwencji zapewniającym dostateczną eksplorację przestrzeni rozwiązań jest mutacja. Jednakże mutowanie łańcuchów bitowych z dużym prawdopodobieństwem jest często

[8] Jak nietrudno zgadnąć, szczególnie badaczy zajmujących się algorytmami ewolucyjnymi, por. np. [62].

destrukcyjne. Dobre rozwiązania tracone są w kolejnych pokoleniach. Oczywiście dywersyfikacja dla samej dywersyfikacji nie jest celem. Należy raczej dążyć do podtrzymywania właściwej różnorodności populacji, aby móc eksplorować nowe obszary przestrzeni rozwiązań bez gubienia dotychczas zdobytej informacji.

Park i Carter, [147], twierdzą, że w rzeczywistych złożonych zadaniach – choćby zadaniu wyznaczenia maksymalnej kliki grafu[9] – rola operatora krzyżowania jest marginalna, a jedynym jej skutkiem jest tracenie zasobów obliczeniowych.

Przeprowadzone przez Schafera i Eshelmana, [163], testy wskazują, że krzyżowanie może wykorzystywać **epistazę** (tzn. silne uzależnienie między genami chromosomu), natomiast czysta mutacja - nie. Próbą załagodzenia tego sporu jest praca [172], w której twierdzi się, że użyteczność mutacji zależy od rozmiaru populacji. W populacjach o małym rozmiarze mutacja okazuje się bardziej skuteczna, natomiast ze wzrostem rozmiaru populacji rośnie rola krzyżowania.

Natomiast Spears, [170] – traktując mutację i krzyżowanie jako operatory „destrukcyjne" – stwierdza, że mutacja pozwala na uzyskanie zniekształceń chromosomów[10] w stopniu porównywalnym ze zniekształceniami wywołanymi ich krzyżowaniem. Co więcej, zakresem eksploracji przestrzeni rozwiązań mutacja przewyższa odpowiednie zdolności operatora krzyżowania. Konkludując, autor ten stwierdza, że oba operatory można traktować jako dwie formy pewnego operatora odpowiadającego za eksplorację przestrzeni rozwiązań, który może modyfikować allele na podstawie dostępnej informacji (por. też [179]). Zdaniem Spears'a nie jest jasne czy rozróżnianie między krzyżowaniem a mutacją jest konieczne ani nawet wskazane, chociaż jest niewątpliwie wygodne.

3.2.8. Ewolucja lamarckowska czy darwinowska?

Twórca terminów „bezkręgowce" i „biologia", Jean-Baptiste Lamarck (1744-1829), był pierwszym, który zasugerował, że proste gatunki przekształcają się w bardziej złożone na skutek dziedziczenia zdobytych zdolności i ukierunkowanej ewolucji. A zatem, spontanicznie powstałe mikroby w procesie powolnych zmian przekształcają się w rośliny, zwierzęta niskiego rzędu, kręgowce a wreszcie w ludzi. Kwintesencją teorii Lamarcka są cztery prawa:

1. Wszystkie żyjące formy nieustannie zwiększają swój rozmiar, złożoność i umiejętności do momentu przekroczenia granic danego gatunku.
2. Powstawanie nowych organów jest skutkiem nowych potrzeb lub konsekwentnej dążności danego organizmu (tzw. psychologiczny czynnik rozwoju).
3. Te części ciała, które używane są bardziej intensywnie w trakcie życia danego osobnika wzrastają szybciej i stają się coraz bardziej złożone, natomiast rzadko używane części ciała stopniowo zanikają.
4. Zdolności pozyskane przez osobnika są dziedziczone przez jego potomków.

Chociaż stosunkowo szybko wykazano naiwność tej teorii, to czwarte prawo Lamarcka odnajdujemy w części prac dotyczących algorytmów ewolucyjnych. Możliwość kodowania pewnych charakterystyk fenotypowych – pozyskanych w trakcie uczenia – w struktury genowe wydaje się przyspieszać proces poszukiwania rozwiązań. Z drugiej strony, J.M. Baldwin zasugerował w 1896 roku nie-lamarckowską drogę ewolucji, zgodnie z którą zdolności pozyskane w trakcie życia jednostki przenoszą się pośrednio na potomstwo. Zgodnie z jego propozycją, jeżeli zdolność uczenia się zwiększa szanse przeżycia, to osobniki charakteryzujące się takimi zdolnościami będą mieć większą liczbę potomstwa dziedziczącego ich geny odpowiedzialne za łatwość uczenia się. Mechanizm ten, nazwany przez samego Baldwina „selekcją organiczną", znany jest w literaturze jako **efekt Baldwina**. Wielu ewolucyjnych biologów pozostaje sceptycznymi wobec tego efektu twierdząc, że mechanizmy korelujące fenotypową plastyczność z genetycznym zróżnicowaniem nie są dostatecznie jasne. Pomimo tych wątpliwości, efekt Baldwina wykorzystywany jest w wielu pracach poświęconych algorytmom ewolucyjnym. Bogatą bibliografię można znaleźć na stronie http://extractor.iit.nrc.ca/baldwin/bibliography.html.

[9] Por. [4].
[10] Bardziej precyzyjnie: pozwala rozrywać (niszczyć) schematy. Pojęcie schematu wprowadzamy w punkcie 3.3.

Hinton i Nowlan, [94], jako jedni z pierwszych zbadali wpływ tego efektu na przyspieszenie ewolucji. W swoim eksperymencie wykorzystali oni algorytm genetyczny do modelowania ewolucji populacji złożonej z 1000 osobników, których chromosomy składały się z 20 genów, a każdy gen przyjmował jedną z wartości **0**, **1** lub **?**. Pojedynczy osobnik reprezentował połączenia prostej sieci neuronowej. Symbol **0** odpowiadał brakowi połączenia, symbol **1** – połączeniu odpowiednich jednostek a **?** – decyzję dotyczącą braku lub konieczności wystąpienia odpowiedniego połączenia. Celem uczenia było ustalenie odpowiednich połączeń między neuronami sieci, tzn. właściwe zastąpienie symboli **?** przez **0** lub **1**. Początkową populację utworzono losowo, przy czym prawdopodobieństwo wystąpienia symboli **0** i **1** było równe 0,25, a symbolu **?** – 0,5. W każdej iteracji pojedynczy osobnik podejmował 1000 prób prawidłowego zastąpienia symboli **?** przez symbole ze zbioru {**0**, **1**}. Jego dopasowanie obliczano zgodnie ze wzorem

$$f = 1 + \frac{(1000 - i) \cdot 19}{1000}$$

gdzie i oznacza liczbę prób, po której uzyskano prawidłową konfigurację połączeń. Jeżeli po wykonaniu 1000 prób konfiguracja połączeń była nadal nieprawidłowa, wartość dopasowania osobnika pozostawała równa 1.

W eksperymencie wykorzystano prosty algorytm genetyczny z selekcją proporcjonalną i jednopunktowym krzyżowaniem (bez mutacji). Liczbę iteracji ustalono na 50. Docelową konfiguracją był chromosom, którego wszystkie geny mają wartość **1**. W trakcie eksperymentów zaobserwowano szybkie usuwanie symboli **0**, a końcowa populacja zawierała 55% symboli **1** i 45% symboli **?**. Przeciętny osobnik składał się wiec z 11 genów **1** i 9 genów **?**. Dopasowanie takiego osobnika jest równe 11,6. Bez uczenia natomiast przeciętne dopasowanie osobnika po 50 iteracjach jest równe 1,006 – por. [85]. Wysnuto stąd wniosek, że istotnie uczenie przyspiesza ewolucję.

Zastanawiając się nad uzyskanym średnim dopasowaniem można uznać je za wysokie w porównaniu z wartością 1,006, ale z drugiej strony 11,6 to zaledwie 58% maksymalnego dopasowania. Hinton i Nowlan tłumaczyli tę wartość zastosowaniem bardzo prostej, losowej strategii uczenia. Z kolei w cytowanej wyżej pracy [85] za przyczynę tak niskiego dopasowania uznano zbyt małą liczbę iteracji i **dryft genetyczny** spowodowany mutacją (problem w tym, że w zastosowanym w eksperymencie algorytmie genetycznym zrezygnowano z mutacji). Zwiększając liczbę iteracji do 500 uzyskano redukcję symboli **?**, ale liczba symboli **1** nadal pozostawała mniejsza niż 20.

O ile rola uczenia w przyspieszaniu działania algorytmów ewolucyjnych wciąż budzi kontrowersje, to mechanizm selekcji klonalnej polegający na klonowaniu najlepszych osobników, ich mutowaniu i wyborze nowych, skuteczniej zwalczających patogeny przypomina w pewnym stopniu omawiany tu efekt Baldwina. Ogólnie przyjmuje się, że jeżeli ewoluujące osobniki wykorzystują mechanizmy uczenia się i wywołane tymi procesami zmiany genetyczne wprowadzane są do populacji, mamy do czynienia z algorytmem lamarckowskim; jeżeli zdolność do uczenia odzwierciedlana jest wyłącznie w wartości dostosowania osobnika – z algorytmem stosujacym efekt Baldwina.

3.3. Twierdzenie o schematach

Sformułowane przez Hollanda twierdzenie o schematach udziela prostej i jednocześnie eleganckiej odpowiedzi na pytanie „dlaczego algorytm genetyczny działa?”. Twierdzenie to odnosi się do algorytmów używających binarnej reprezentacji chromosomów. Kluczową rolę przy jego formułowaniu odgrywa pojęcie **schematu**, pod którym to pojęciem rozumiemy szablon pozwalający określać ważne sekwencje genów w chromosomie. W tym celu do alfabetu genów wprowadza się dodatkowy symbol ***** (nieistotne). Na przykład schemat H_1: 0**10*** oznacza, że istotne dla osiągnięcia pożądanych własności są pozycje: czwarta, piąta i ósma, na których powinny wystąpić wartości 0, 1 i 0; na pozostałych pozycjach łańcucha mogą występować dowolne wartości. Schematy nazywane są także *zbiorami podobieństw*, gdyż każdy schemat reprezentuje zbiór łańcuchów binarnych o identycznych bitach na wyróżnionych pozycjach. Ważnymi charakterystykami schematu H są: jego **rząd** – oznaczany $o(H)$ – czyli liczba znaczących pozycji w H, oraz jego **rozpiętość**[11] – oznaczana $\delta(H)$ – czyli odległość między pierwszą a ostatnią ustaloną pozycją schematu. Na przykład, dla

[11] Inna nazwa rozpiętości schematu to długość definiująca schematu.

określonego wyżej schematu H_1 sprawdzamy, że $o(H_1) = 3$, $\delta(H) = 8 - 4 = 4$. W istocie rozpiętość schematu określa liczbę punktów rozcięcia grożących jego zniszczeniem. Każdy schemat reprezentuje 2^r łańcuchów, gdzie r jest liczbą znaczących pozycji schematu, a każdy łańcuch binarny o długości l jest reprezentantem 2^l schematów. Na przykład łańcuch 01 należy do czterech schematów: 01, 0*, 1*, **.

Schematy pozwalają śledzić wpływ selekcji, krzyżowania i mutacji na zawartość populacji łańcuchów binarnych w kolejnych iteracjach. Analizy te dotyczą tzw. *prostego* algorytmu genetycznego, tzn. takiego algorytmu, w którym kolejne populacje tworzone są z wykorzystaniem operatorów selekcji proporcjonalnej, krzyżowania jednopunktowego i mutacji probabilistycznej. Jeżeli dla konkretnej populacji oznaczymy przez:

- $M(H, t)$ liczbę łańcuchów pasujących do schematu H w chwili t,
- p_c, p_m – odpowiednio prawdopodobieństwo krzyżowania i mutacji,
- l – długość łańcucha,
- $<f>_H$ – wartość dostosowania schematu H,
- $<f>_N$ – średnią wartość dostosowania w populacji złożonej z N osobników, to oczekiwaną liczbę łańcuchów pasujących do tego schematu w następnym pokoleniu określa nierówność[12]:

$$M(H,t+1) \geq M(H,t) \cdot \frac{<f>_H}{<f>_N} \cdot \left[1 - p_c \cdot \frac{\delta(H)}{l-1} - p_m \cdot o(H)\right] \qquad (3.1)$$

Fakt, że w (3.1) zamiast równości mamy do czynienia z nierównością wynika z pesymistycznego założenia, że skrzyżowanie w punkcie leżącym między jego skrajnymi znaczącymi pozycjami prowadzi zawsze do utraty tego schematu W rzeczywistości to co jest stratą dla jednego schematu, równocześnie jest zyskiem dla innego schematu. Analizę utraty materiału genetycznego dla problemu dwubitowego zawiera punkt 2.6.1 książki [71], a jej uogólnienie dla problemu m-bitowego ($m \leq 15$) – praca [195].

Pomijając chwilowo wpływ krzyżowania i mutacji otrzymujemy z (3.1) równanie geometrycznego wzrostu schematów w kolejnych pokoleniach mówiące, że ponadprzeciętne schematy, czyli schematy, których dostosowanie przekracza średnie dostosowanie w populacji, będą uzyskiwać w kolejnych pokoleniach wykładniczo rosnącą liczbę łańcuchów. W podobnym tempie będą zanikać schematy o niskim dostosowaniu. Dodatkowo, jeżeli ponadprzeciętne schematy mają niewielką rozpiętość, ich szansa rozerwania – zgodnie z drugim składnikiem występującym w nawiasie kwadratowym wyrażenia (3.1) – jest niewielka. Podsumowaniem obu tych obserwacji jest tzw. **twierdzenie o schematach** (nazywane też *podstawowym twierdzeniem algorytmów genetycznych*) mówiące, że ponadprzeciętne schematy o niskiej rozpiętości uzyskują wykładniczo rosnącą liczbę łańcuchów w kolejnych pokoleniach.

Z twierdzeniem tym związana jest tzw. **hipoteza o blokach budujących** głosząca, że algorytm genetyczny poszukuje działania zbliżonego do optymalnego przez zestawianie krótkich, niskiego rzędu schematów o wysokim dostosowaniu, nazywanych blokami budującymi. Zestawiając, wybierając i powielając bloki budujące algorytm tworzy z nich nowe ciągi kodowe o potencjalnie wyższym dostosowaniu. W ten sposób zamiast przeglądać całą przestrzeń rozwiązań, algorytm koncentruje się jedynie na wybranych jej obszarach wskazywanych przez ponadprzeciętne schematy.

Elegancja i ekspresja twierdzenia o schematach są niewątpliwe. Ale nasz świat nie jest najlepszym z możliwych światów. Zastanawiając się nad konsekwencjami tego twierdzenia dochodzimy bowiem do wniosku, że przeczy ono przesłankom, na których opiera się idea algorytmów genetycznych. O ile w przyrodzie ewolucja doprowadziła do koegzystencji dużej liczby stabilnych gatunków, o tyle w algorytmie genetycznym schemat o wysokim dostosowaniu dominuje populację „wymiatając" z niej inne „gatunki". Zdaniem Gaspara i Collarda, [69], takie zachowanie jest typowe dla algorytmów optymalizacji, gdzie nacisk kładzie się na szybkość znalezienia optimum, a nie dla systemów adaptacyjnych. Zauważają oni także, że

[12] Jej wyprowadzenie zawiera niemal każdy podręcznik poświecony algorytmom genetycznym.

ze wzrostem iteracji maleje wpływ krzyżowania, a jedynym mechanizmem odpowiedzialnym za eksplorację przestrzeni rozwiązań staje się operator mutacji (por. dyskusję z punktu 3.2.7).

Działanie algorytmu genetycznego jest wypadkową dwóch mechanizmów: pierwszy mechanizm odpowiedzialny jest za zdolności eksploracyjne algorytmu polegające na analizie różnych obszarów przestrzeni rozwiązań, a drugi mechanizm – za stopień wykorzystania informacji o jakości danego obszaru. Drugi z wymienionych mechanizmów wiąże się z pojęciem **naporu selekcyjnego** rozumianego jako stosunek dostosowania najlepszego osobnika w populacji do średniego dostosowania tejże populacji, [11]. Wzrost prawdopodobieństwa krzyżowania bądź mutacji, ewentualnie obniżenie naporu selekcyjnego, zwiększa różnorodność populacji wpływając pozytywnie na zdolności eksploracyjne i jednocześnie zmniejszając jego zdolności eksploatacyjne dobrych schematów. Na odwrót, zwiększenie naporu selekcyjnego poprawia zdolności eksploatacyjne kosztem zdolności eksploracyjnych. W efekcie algorytm może znajdować rozwiązania suboptymalne pomijając regiony, w których lokują się rozwiązania optymalne. Istotnym problemem jest więc utrzymanie właściwej równowagi między zdolnościami eksploracyjnymi a eksploatacyjnymi algorytmu. Dążenie to tłumaczy między innymi różnorodność mechanizmów selekcji wprowadzonych w punkcie 3.2.4.

Algorytmy immunologiczne, stanowiące zasadniczy przedmiot książki, w zdecydowanie bardziej naturalny sposób równoważą eksplorację z eksploatacją, co prowadzi do powstawania i stabilnego podtrzymywania zróżnicowanych populacji osobników. Wykorzystanie dodatkowych mechanizmów, omawianych w punkcie 3.5, pozwala co prawda „wymuszać" zbliżone zachowanie się algorytmów genetycznych, ale jak to zwykle bywa z wymuszanymi zachowaniami – nie gwarantują one pełnego sukcesu.

3.5. Metody niszowe

Metody niszowe są przykładem wysiłków badaczy zajmujących się algorytmami genetycznymi w celu uzyskania stabilnych podpopulacji reprezentujących rozwiązania o różnych własnościach. Metody te można podzielić na **przestrzenne** i **temporalne** w zależności od tego, czy kolejne podpopulacje rozwiązań powstają równolegle czy też szeregowo[13] oraz na metody z **pojedynczym** lub **wielokrotnym środowiskiem**, gdzie modelem środowiska jest funkcja dostosowania, [134]. Ten drugi podział równoważny jest dychotomii sympatryczna-alopatryczna specjacja[14] stosowanej w ekologii populacyjnej. Termin **sympatryczna specjacja** odnosi się do gatunków koegzystujących geograficznie, które w trakcje ewolucji wykorzystują różne zasoby; natomiast **specjacja alopatryczna** dotyczy gatunków rozwijających się w odrębnych rejonach geograficznych. Oczywiście z odrębnymi rejonami geograficznymi wiążą się różne środowiska. Podsumowując tę krótka charakterystykę należy podkreślić, że metody przestrzenne z pojedynczym środowiskiem nadają się do rozwiązywania zadań optymalizacji wielomodalnej czy klasyfikacji; metody przestrzenne z wielokrotnym środowiskiem stosowane są w optymalizacji wielokryterialnej[15], a temporalne wielo-środowiskowe techniki – do symulacji adaptacyjnej. Przykładem wielośrodowiskowych metod przestrzennych są algorytmy immunologiczne. Poniżej omawiamy krótko wybrane techniki.

[13] Z tego powodu te ostatnie nazywane są również sekwencyjnymi metodami niszowymi.
[14] Por. [71], punkt 5.4
[15] Zadania optymalizacji wielomodalnej i wielokryterialnej zdefiniowano w punkcie 9.1