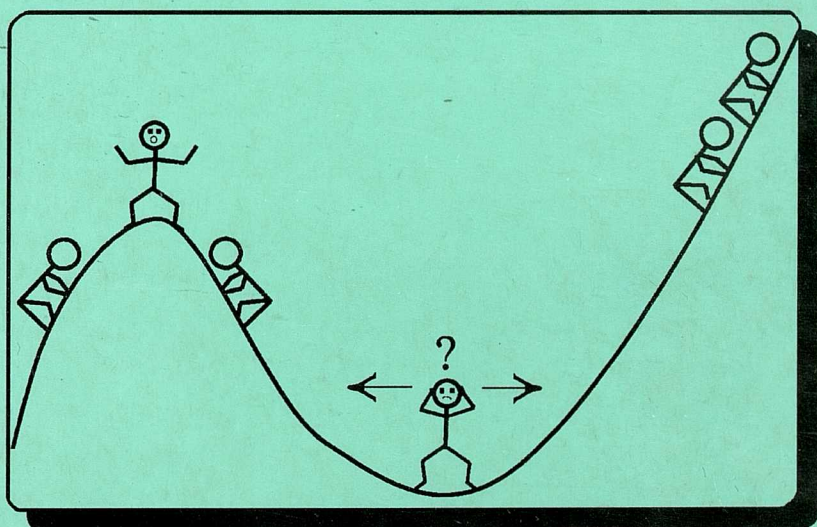


HALINA KWAŚNICKA

OBLICZENIA EWOLUCYJNE W SZTUCZNEJ INTELIGENCJI





HALINA KWAŚNICKA

OBLICZENIA EWOLUCYJNE W SZTUCZNEJ INTELIGENCJI



**Oficyna Wydawnicza Politechniki Wrocławskiej
Wrocław 1999**

OPINIODAWCY
Leonard Bolc
Ryszard Tadeusiewicz

REDAKTOR
Alicja Kordas

KOREKTA
Zofia Jasina, Alina Kaczak

Biblioteka Główna i OINT
Politechniki Wrocławskiej



001855903



3025 21 L / 1

© Copyright by Halina Kwaśnicka, Wrocław 1999

ISBN 83-7085-421-4

OFICyna WYDAWNICZA POLITECHNIKI WROCLAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

Ark. wyd. 13,75. Ark. druk. 11. Papier offset kl. III, 80 g, B1.
Drukarnia Oficyny Wydawniczej Politechniki Wrocławskiej. Zam. nr 409/99.

SPIS RZECZY

Wstęp.....	5
1. Algorytmy ewolucyjne.....	11
1.1. Krótko o ewolucji.....	11
1.2. Algorytmy genetyczne.....	13
1.3. Inne techniki ewolucyjne.....	16
1.4. Równoległe przetwarzanie algorytmów genetycznych.....	19
2. Przykłady systemów z wbudowanymi EA.....	22
2.1. GA w systemach informacyjnych.....	22
2.2. Łączenie algorytmów genetycznych z logiką rozmytą.....	24
2.2.1. Systemy z logiką rozmytą – wprowadzenie.....	25
2.2.2. Połączenie GA i FLS.....	26
2.2.3. Projektowanie rozmytego systemu sterowania – przykład.....	27
2.3. Zastosowanie GAs do sieci neuronowych.....	31
2.3.1. Sieci neuronowe.....	31
2.3.2. Łączenie GAs i NNs.....	33
3. <i>K</i> -model – rozszerzony model ewolucyjny.....	41
3.1. Biologiczne inspiracje <i>K</i> -modelu.....	41
3.2. Rola plejotropowości i poligeniczności.....	44
3.2.1. Model Wersja_1: klasyczny algorytm genetyczny.....	45
3.2.2. Model Wersja_2: GA z minimalną plejotropowością i poligenicznością.....	46
3.2.3. Ewolucja bez poziomu genotypowego (Wersja_3).....	46
3.2.4. Badania symulacyjne.....	47
3.3. <i>K</i> -model.....	51
3.3.1. Formalny opis modelu.....	52
3.3.2. Werbalny opis modelu.....	55
3.3.3. Opis programu komputerowego <i>K</i> -model.....	57
4. Własności <i>K</i> -modelu – badania symulacyjne.....	61
4.1. Funkcje testowe.....	62
4.2. Rekombinacje i mutacje w ewolucji.....	70
4.2.1. Rola rekombinacji w <i>K</i> -modelu.....	71
4.2.2. Mutacje jako czynnik ewolucji.....	78
4.3. Makromutacje w poszukiwaniu optymalnego rozwiązania.....	90
4.3.1. Geny nadmiarowe.....	90

4.3.2. Rekrudescencja – radykalne zmiany u części populacji.....	92
4.3.3. Kryzys – reorganizacja wszystkich osobników.....	93
4.3.4. Łączny efekt nadmiarowości, rekrudescencji i kryzysu.....	94
4.4. Liczebność populacji i ewolucja w K -modelu.....	96
4.5. Inne parametry modelu.....	103
4.6. Podział populacji na prawie izolowane demy.....	112
4.6.1. Symulacja z H -modelem.....	112
4.6.2. Symulacja z K -modelem.....	113
4.6.3. Symulacja ze zmodyfikowanym K -modelem.....	115
5. Ewolucja z wielowymiarowymi i wielomodalnymi funkcjami.....	119
5.1. Plejotropowość i poligeniczność w ukierunkowywaniu ewolucji.....	119
5.2. Wzrost wymiaru funkcji a ewolucja.....	128
5.3. Ewolucja z funkcjami testowymi de Jonga.....	134
5.4. Szukanie optimów pozostałych funkcji testowych.....	140
6. Przykład aplikacji K -modelu – modelowanie ewolucyjne w ekonomii.....	144
6.1. Ewolucyjny model przemysłu (EMP).....	144
6.1.1. Reprezentacja firm w EMP.....	147
6.1.2. Decyzje firm.....	148
6.1.3. Zróżnicowanie produktów i konkurencja.....	150
6.2. Konkurencyjna gra ekonomiczna oparta na EMP.....	152
7. Uwagi końcowe.....	156
Literatura.....	164
Załącznik 1. Słownik ważniejszych terminów.....	172
Skorowidz.....	175

Wiedza nigdy nie rozstrzygnie jednego problemu bez stworzenia dziesięciu innych.

Bernard SHAW (1856–1950)

WSTĘP

Myślące maszyny – marzenie czy rzeczywistość? Czy można jednoznacznie odpowiedzieć na tak postawione pytanie? Zarówno odpowiedź twierdząca, jak i przecząca ma swoich zwolenników jak i przeciwników¹. Zaproponowany w 1950 roku przez A. Turinga (1912–1954) test, zwany *testem Turinga* nie rozwiązał istniejących kontrowersji. W zamierzeniu test ten miał udzielać odpowiedzi, czy dany program komputerowy jest inteligentny. Test polega na tym, że przy jednym komputerze jest osoba, która prowadzi dyskusję (na ogólny temat) z innym dyskutantem, tyle, że może nim być program komputerowy lub człowiek, zajmujący miejsce przy klawiaturze innego komputera. Jeśli osoba nie rozpozna, czy dialog prowadzi z programem czy z człowiekiem, program spełnia test, w przeciwnym razie – nie spełnia. Jest wiele programów komputerowych spełniających taki test, ale przy ograniczeniu dyskusji do wąskiej tematyki. Sformułowanie tego testu nie zakończyło dyskusji na temat „czy maszyna może myśleć?” W latach siedemdziesiątych H.L. Dreyfus opublikował krytyczną książkę, w której dowodzi nieadekwatności i niepowodzeń symulacji w stosunku do rzeczywistego intelektu². Książka ta wywołała ożywioną polemikę wśród osób zajmujących się sztuczną inteligencją. Echo tych dyskusji widoczne jest również w literaturze polskojęzycznej (Tadeusiewicz 1995, Kloch 1996, Traczyk 1995).

Niezależnie od tych filozoficznych dyskusji, w ostatnich latach obserwuje się wzrost zastosowań takich systemów informatycznych, w których zawarta jest inteligencja wzorowana na inteligencji ludzkiej. W odróżnieniu od tej ostatniej, nazwano tę dziedzinę **sztuczną inteligencją** (**AI** – *Artificial Intelligence*) – obejmuje ona rozwiązywanie problemów sposobami wzorowanymi na naturalnych działaniach i procesach

¹ Patrz: J.R. Searle, *Umysł, mózg i nauka*, Warszawa, Wydawnictwo Naukowe PWN, 1995, czy *Trzecia kultura. Nauka u progu trzeciego tysiąclecia*, pod red. J. Brockmana, Warszawa, Wydawnictwo CIS, 1996.

² Dreyfus H.L., *What Computers Can't Do: The Limits of Artificial Intelligence*, wyd. II, New York, Harper and Row. Tematyka możliwości sztucznej inteligencji jest ciekawie przedstawiona w *Artificial Intelligence and Creativity*, T. Dartnall ed., ser. *Studies in Cognitive Systems*, Kluwer A. Publ., 1994.

poznawczych człowieka, za pomocą symulujących je programów komputerowych (Schalkoff 1990). W ramach tej nowej dziedziny wiedzy możemy wyróżnić różne techniki rozwiązywania problemów:

Systemy ekspertowe (ESs – Expert Systems), tzn. takie komputerowe programy konsultacyjne, które wspomagają decyzje i zastępują eksperta w danej, zazwyczaj wąskiej dziedzinie. System ekspertowy składa się z bazy wiedzy, modułu wnioskującego – do przetwarzania posiadanej wiedzy, modułu wyjaśniającego użytkownikowi sposób dochodzenia do konkluzji oraz modułu pozyskiwania wiedzy. Dobry interfejs oraz moduł wyjaśniający sposób dochodzenia do konkluzji są bardzo ważne, zwłaszcza z punktu widzenia testowania systemu, są też niezbędne do nabrania przez użytkownika (zwykle nie jest nim informatyk) zaufania do systemu. Ekspertyzy systemu powinny być co najmniej tak dobre, jak te, które dałby ekspert. Aby to było możliwe, musi być spełnionych kilka warunków, ale najważniejszym (i najtrudniejszym do spełnienia) jest jakość dostarczonej systemowi wiedzy. System ekspertowy może być na tyle dobry, na ile dostarczymy mu dobrą wiedzę. „Dobra” to znaczy prawdziwa, wystarczająca dla danej klasy problemów, dobrze sformalizowana, niesprzeczna itp. Pożądana jest również umiejętność przetwarzania wiedzy niepełnej i niepewnej. Wprowadza się wnioskowanie z logiką rozmytą czy z czynnikiem pewności, określającym stopień zaufania do generowanej wiedzy. Pozyskanie wiedzy dla systemów ekspertowych jest „wąskim gardłem” ich rozwoju. Pojawiły się narzędzia do tworzenia systemów ekspertowych, ułatwiające pozyskiwanie wiedzy, kodowanie jej i przetwarzanie³. W krajach wysoko rozwiniętych systemy ekspertowe stosuje się w wielu dziedzinach, między innymi w medycynie, ubezpieczeniach, marketingu itp. (Mulawka 1996, Yager i Filev 1995, Traczyk 1995, Medsker 1995, Heider i in. 1994, Herrera i Lozano 1995, Kwaśnicka i Markowska-Kaczmarska 1994, 1994a, Kwaśnicka 1996).

Sieci neuronowe (NNs – Neural Networks). Próba zbudowania sztucznego mózgu to początek rozwoju tej gałęzi sztucznej inteligencji. NNs są wzorowane na sieciach neuronów biologicznych w mózgu. Głównym podobieństwem biologicznej i sztucznej sieci neuronowej jest jej zdolność uczenia się odwzorowań zbiorów wejść na zbiór wyjść na podstawie uczących przykładów oraz jej zdolność uogólniania. W ostatnich latach rozwinięto nowe algorytmy uczenia sieci, głębiej poznano i sformułowano matematyczne podstawy sieci neuronowych, modele sieci zostały nauczone i zastosowane do rozwiązywania wielu problemów w różnych dziedzinach, np. do rozpoznawania podpisów w bankach, klasyfikacji, przewidywania kursów na giełdach itp. (Yager i Zadeh 1994, Tadeusiewicz 1992, 1993, Korbicz i Obuchowicz 1994, Ossowski 1994, Medsker 1995, Kacprzak i Ślot 1995).

Algorytmy genetyczne (GAs – Genetic Algorithms) są wzorowane na ewolucji biologicznej, która jest bardzo skutecznym, naturalnym systemem adaptacyjnym (Holland 1975, Michalewicz 1996, Goldberg 1989, Gwiazda 1995, Kwaśnicka 1994, Kwaśnicka 1998). Główne różnice między GAs a konwencjonalnymi metodami optymalizacji to:

³ Np. XpertRule, <http://www.attar.com>, EMYCIN (system ekspertowy MYCIN bez bazy wiedzy) itp.

– **GA**s pracują z zakodowanymi parametrami w postaci genów stanowiących chromosom, najczęściej (choć niekoniecznie) parametry są kodowane w alfabecie binarnym, jako ciąg zer i jedynek, chromosom koduje punkt w przestrzeni poszukiwań (rozwiązań),

– poszukują rozwiązania, pracując jednocześnie z populacją punktów, a nie z pojedynczym punktem w przestrzeni parametrów,

– nie posługują się pochodnymi funkcji ani inną wiedzą o funkcji celu,

– stosują probabilistyczne reguły przeszukiwania przestrzeni parametrów optymalizowanej funkcji, wykorzystując obszary o wysokim stopniu przystosowania (dobre rozwiązania).

Automatyczne uczenie się (ML – Machine Learning). Dziedzina ta zajmuje się szukaniem odpowiedzi na pytanie, w jaki sposób konstruować programy komputerowe, które potrafią automatycznie polepszać swoje działanie w miarę zdobywania „doświadczenia”. Stosuje się tu pojęcia i wiedzę z różnych obszarów badań, np. statystyki, sztucznej inteligencji, filozofii, teorii informacji, biologii i in. Ludzkość od początków istnienia komputerów marzyła, by mogły się one uczyć, ale jak je zaprogramować, by to potrafiły? Satysfakcjonującej odpowiedzi na to pytanie nadal nie znamy, choć rozwijane są nowe algorytmy. Dobre sformułowanie problemu, wybór zbioru uczącego (doświadczeń) mają bardzo duże znaczenie dla zadania automatycznego uczenia się (Mitchell 1997, Hutchinson 1994, Kwaśnicka 1998b).

Wymienione techniki znajdują zastosowanie w różnych dziedzinach. Wymienienie ich wszystkich wydaje się niemożliwe. W literaturze przedmiotu wyróżnia się najczęściej następujące zadania (Schalkoff 1990):

- Robotyka – planowanie drogi ruchu robotów (*path and motion planning*), rozpoznawanie obrazów – dokładniej, rozpoznawanie wzorców (*pattern recognition* i *computer vision*), sterowanie, tworzenie harmonogramów i diagnostyka produkcji (*manufacturing control, diagnostic systems, scheduling*).

- Systemy ekspertowe – diagnostyka medyczna, wojskowość, sterowanie podstacjami energetycznymi, geologia, projektowanie, konfiguracja komputerów.

- Gry – logiczne (najbardziej spektakularny sukces to Deep Blue, zwycięzca nad szachowym mistrzem świata A. Kasparowem, dobre osiągnięcia programu TD-GAMMON grającego w trik-traka (Mitchell 1997), menedżerskie i wojskowe.

- Rozumienie języka naturalnego (rozpoznawanie mowy, tłumaczenie tekstów).

- Automatyczne dowodzenie twierdzeń.

- Komputerowe wspomaganie nauczania(uczenia).

- Automatyczne podejmowanie decyzji.

- Automatyczne generowanie programów komputerowych.

Zaprezentowane wprowadzenie w problematykę sztucznej inteligencji, mimo iż bardzo krótkie, pokazuje jak szeroka i interdyscyplinarna jest ta dziedzina. Zgodnie z cytowaną już jej definicją, sztuczna inteligencja naśladuje zachowania, jakie są obserwowane u człowieka. Wydaje się jednak, że coraz bardziej upowszechnia się rozu-

mienie sztucznej inteligencji w sposób sformułowany przez D.B. Fogla w przedmowie do jego książki zatytułowanej *Evolutionary Computation Toward a New Philosophy of Machine Intelligence* (Fogel 1995): sztuczna inteligencja to zdolność systemu do dostosowania swojego działania tak, aby osiągnąć założony cel w środowisku, w którym się znajduje. Inteligentne stworzenia powstały w wyniku ewolucji biologicznej. Obserwując i modelując ewolucję, możemy uzyskać wiele inteligentnych zachowań. Wszelkie metody symulacji ewolucji za pomocą komputera noszą nazwę **obliczeń ewolucyjnych** (EC – *Evolutionary Computation*). Wszelkie algorytmy stosowane w takich symulacjach to **algorytmy ewolucyjne** (EAs – *Evolutionary Algorithms*).

W ostatnich latach paradygmat obliczeń ewolucyjnych stał się bardzo popularny. Obserwowane jest zwiększenie liczby prac na ten temat, czasopism, konferencji, książek, powstają listy dyskusyjne, strony internetowe. Podobnie zwiększa się liczba różnych dziedzin, w których algorytmy ewolucyjne znajdują zastosowanie: od naturalnych dla nich zadań modelowania dynamiki populacji, przez zastosowania czysto techniczne (np. projektowanie samolotów, gazociągu), zadania szeregowania, gry logiczne, nauki chemiczne i fizyczne, po nauki ekonomiczne – popularny ostatnio paradygmat ekonomii ewolucyjnej.

Niniejsza praca dotyczy **obliczeń ewolucyjnych**, można powiedzieć – modnego ostatnio tematu. Intencją autorki jest, by przy okazji własnych prac pokazać, jak można wykorzystać podejście ewolucyjne do rozwiązywania różnych klas zadań – automatyczne generowanie reguł, projektowanie sieci neuronowych czy systemów z logiką rozmytą, modelowanie w ekonomii. Powinno to pomóc czytelnikowi w zdefiniowaniu własnego problemu w terminach algorytmów ewolucyjnych. Ten pierwszy etap ich stosowania jest zwykle bardzo trudny. Czasami nasze próby z EA kończą się fiaskiem, z braku „wyczucia” działania tych algorytmów, ich właściwości i wrażliwości na parametry, stosowane funkcje przystosowania itp. Autorka w tej pracy pominęła analizę matematyczną właściwości algorytmów ewolucyjnych; próby takich analiz są dostępne, głównie w czasopismach i materiałach konferencyjnych, można je znaleźć również w internecie⁴.

Początek pracy – rozdział pierwszy – to wprowadzenie do tematyki algorytmów ewolucyjnych. Odrobina informacji o teorii ewolucji, która była inspiracją rozwoju algorytmów genetycznych, ułatwi czytelnikowi zrozumienie ich idei (Gould 1991, Fogel 1992, Hoffman 1983, Dawkins 1995, 1996, Szarski 1986, Mayr 1982, 1991). Następnie omówiono działanie algorytmów genetycznych oraz krótko przedstawiono pozostałe, równie popularne klasy algorytmów ewolucyjnych, tzn. strategie ewolucyjne, programowanie ewolucyjne i programowanie genetyczne. W tej części znajduje się również krótki przegląd możliwych sposobów przetwarzania EA w środowisku roz-

⁴ Biblioteka prowadzonego przez D.E. Goldberga laboratorium *Illinois Genetic Algorithms Laboratory*, University of Illinois at Urbana-Champaign jest ogólnie dostępna w internecie pod adresem <http://GAL4.GE.UIUC.EDU/illigal.home.html>. Są też inne strony zawierające zarówno artykuły, jak i kody programów (patrz odsyłacze na stronie internetowej autorki: <http://www.ci.pwr.wroc.pl/~kwasnick>).

proszonym i równoległym (Holland 1975, Goldberg 1989, Michalewicz 1996, Kwaśnicka 1998, Cantú-Paz 1977a, Seredyński 1998).

W rozdziale drugim pokazano wybrane przykłady zastosowań algorytmów ewolucyjnych w hybrydowych systemach inteligentnych. Skupiono się głównie na pracach wykonanych przy współudziale autorki⁵. Włączenie **EA** w systemy wyszukiwania informacji może przyspieszyć działanie takich systemów. Wyniki naszej pracy zdają się potwierdzać opłacalność takiego podejścia. Bardzo użyteczne są algorytmy genetyczne zastosowane do projektowania systemów z logiką rozmytą. Ten sposób łączenia technik **AE** i **FL** wydaje się najciekawszy, choć nie jest jedyny. Poświęcono temu jeden z podrozdziałów. Pokazano, w jaki sposób definiować problem i jakie uzyskano przykładowe wyniki. Ostatnia część rozdziału to omówienie zastosowania **EA** do uczenia i projektowania topologii sieci neuronowych (Medsker 1995, Kwaśnicka i Nowostawski 1997, Heider i in. 1994, Heider i Drabe 1997, Herrera i Lozano 1996, Rutkowska i in. 1997, Montana i Devis 1989, Schafer i in. 1992, Porto i in. 1995, Kwaśnicka i in. 1998, Kwaśnicka i Szerszon 1997, Kwaśnicka 1997).

Kolejne trzy rozdziały pracy to oryginalny dorobek autorki. Na podstawie analogii biologicznych zaproponowano algorytm ewolucyjny, nazwany **K-modelem**. W **K-modelu** uwzględniono dodatkowe mechanizmy genetyczne, między innymi ułatwiające modelowanie znanych z biologii makromutacji. Uwzględnienie w reprezentacji osobnika genów nadmiarowych oraz efektów plejotropowego i poligenicznego istotnie wyróżnia zaproponowany model od innych algorytmów ewolucyjnych. Macierz plejotropowości, określająca zależności wartości fenów osobników od jego genów, pozwala zwiększyć efektywność algorytmu. Pierwotnym celem budowy tego modelu była chęć modelowania dynamiki populacji, analiza tempa i charakteru ewolucji. W trakcie doświadczeń z różnymi algorytmami ewolucyjnymi nabrano przekonania, że **K-model** ma duże możliwości optymalizacyjne i może być użyteczny do innych zadań.

Omówienie zaproponowanego **K-modelu** zawarte w rozdziale trzecim poprzedzono krótkim przedstawieniem jego przesłanek biologicznych. Wykonane badania symulacyjne, mające na celu sprawdzenie, czy model z plejotropowością jest efektywny, wskazują, że warto wykorzystać ten *pomysł* natury. Przedstawiono sposób reprezentacji osobnika i stosowane operatory, w tym zaproponowane operatory nazwane tranzycją, transpozycją, rekrudescencją i kryzysem.

Rozdział czwarty to seria eksperymentów z **K-modelem**, mająca na celu pokazanie właściwości zaproponowanego algorytmu. Ciekawe wyniki odnoszą się do charakteru i tempa ewolucji populacji liczących niewiele (dziesięciu) osobników w porównaniu do ewolucji dużo większych populacji (100 i 1000 osobników), dla funkcji o różnym wymiarze. Okazuje się, że jeśli nasze zadanie polega na szybkim znalezieniu akceptowalnego rozwiązania, a nie na wysokim średnim przystosowaniu całej populacji, to

⁵ Dokładniejsze omówienie zastosowań algorytmów ewolucyjnych, w tym w systemach hybrydowych, jest zawarte w przygotowywanej do druku pracy *Naśladowanie natury w sztucznej inteligencji*.

małe populacje są znacznie efektywniejsze, zwłaszcza dla funkcji o niewielkiej liczbie wymiarów. Interesujące wyniki uzyskano dzięki analizie wpływu podziału ewoluującej populacji na podpopulacje (demy) na tempo i charakter ewolucji. Wydaje się możliwe znalezienie optymalnego podziału: jeden dem większy i kilka małych.

W rozdziale piątym przedstawiono dalsze badania symulacyjne tego samego modelu. Skupiono się tu nie tyle na badaniach samego modelu, co na sprawdzaniu, czy model radzi sobie dobrze w przypadku funkcji wielomodalnych i wielowymiarowych, w tym dla funkcji testowych de Jonga. Testowano przy okazji, jak na efektywność algorytmu wpływa dobór macierzy plejotropowości.

W rozdziale szóstym pokazano możliwe i przetestowane zastosowanie zaproponowanego *K*-modelu. Z zastosowaniem ewolucyjnego modelu rozwoju firm (Kwaśnicki 1996) podjęto próbę wykonania gry menedżerskiej. Wyniki są bardzo obiecujące, z całą pewnością kierunek ten wart jest kontynuacji (Nowicki 1992, Kot 1997, Kwaśnicka 1998).

W rozdziale końcowym pracy podsumowano uzyskane wyniki oraz omówiono aktualne i planowane prace w tej dziedzinie. Jeden z głównych kierunków zainteresowań to stworzenie w miarę uniwersalnego narzędzia optymalizującego. Ambicją autorki jest, by był to algorytm ewolucyjny, działający w środowisku rozproszonym, inteligentnie zarządzany. Dokładne wycucie wrażliwości algorytmów na parametry, oraz ich naturalnych predyspozycji (w zależności od parametrów, sposobu kodowania itp.) pozwoli sformułować moduł sterujący całością algorytmu. Pierwsza próba – dwa kooperujące algorytmy genetyczne, z których jeden jest ukierunkowany na przeszukiwanie przestrzeni, a drugi na zastosowanie znalezionych dobrych obszarów – wskazuje, że warto jest realizować ten cel. Prace nad tym tematem są rozpoczęte. Drugi nurt zainteresowań to projektowanie sztucznych sieci neuronowych z wykorzystaniem algorytmów ewolucyjnych. W tej dziedzinie mamy spore doświadczenia i będziemy kontynuować prace. Obecnie podejmowane są próby włączenia pewnych elementów *K*-modelu do tego zadania. Opracowanie dydaktycznej gry menedżerskiej, z zastosowaniem paradygmatu ekonomii ewolucyjnej (i *K*-modelu jako podstawy), nadającej się do celów dydaktycznych i badawczych, to również jeden z planowanych kierunków pracy autorki.

Dotychczasowe prace wykazały zalety i użyteczność proponowanego w *K*-modelu podejścia. Jest to algorytm ewolucyjny z zaimplementowanymi genami nadmiarowymi, plejotropowością i poligenicznością oraz mechanizmami rekrudescencji i kryzysu. Kontynuacja badań w wymienionych kierunkach pozwoli zweryfikować walory aplikacyjne modelu.

1. ALGORYTMY EWOLUCYJNE

Uczenie się, zdolność adaptacji do nowych warunków to bardzo pożądane właściwości projektowanych przez człowieka systemów. Stworzenie takiego systemu wciąż jest trudnym zadaniem. Podglądanie i próba naśladowania natury może dać pozytywne wyniki. Jednym z naturalnych procesów, powszechnie naśladowanym, jest ewolucja biologiczna.

Zagadnienie optymalizacji można potraktować w sposób przypominający codzienne działanie człowieka: ocena decyzji człowieka będzie dobra, jeśli podejmuje on właściwą decyzję spośród możliwych, dostępnych mu w danym czasie. Zatem „dobroć” decyzji jest oceną relatywną w stosunku do innych, możliwych w tym czasie decyzji. Akceptując takie podejście do zadania optymalizacji, możemy wzorować się na ewolucji biologicznej.

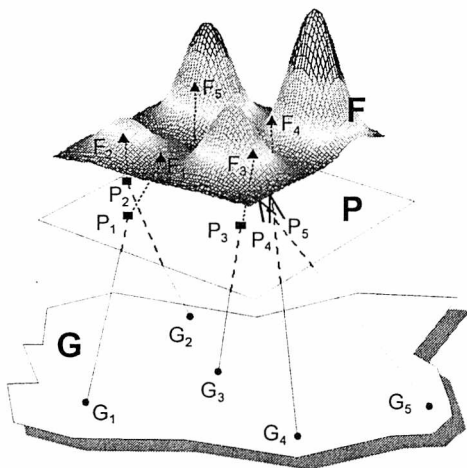
Na podstawie analogii do procesów obserwowanych w ewolucji biologicznej powstało wiele algorytmów, nazywanych wspólnym mianem **algorytmy ewolucyjne** (EAs). Według najbardziej popularnej klasyfikacji, wśród algorytmów ewolucyjnych wyróżniana się trzy główne klasy, wywodzące się z różnych korzeni i stosujące różne techniki (de Jong 1998): **algorytmy genetyczne** (GAs – *Genetics Algorithms*), **strategie ewolucyjne** (ESs – *Evolutionary Strategies*) i **programowanie ewolucyjne** (EP – *Evolutionary programming*). Popularne **programowanie genetyczne** (GP – *Genetic Programming*) zwykle bywa uważane za podgrupę algorytmów genetycznych.

1.1. KRÓTKO O EWOLUCJI

Do późnych lat XVIII wieku, zgodnie z *Biblią*, przyjmowano powszechnie teorię, iż wszystkie organizmy powstały w rezultacie jednorazowego, krótkotrwałego aktu stworzenia. Odkrycie gigantycznych kości w Ohio i w Europie zwróciło jednak uwagę Francuza Georges’a Cuvieara (1779–1832) na to, że wiele gatunków przetrwało do dziś jedynie w postaci skamieniałości – wymarły one przed milionami lat. Prawdziwa rewolucja w biologii nastąpiła 1 lipca 1858 roku, kiedy Charles Darwin i Alfred R. Wallace zaprezentowali swoje teorie. Zrozumienie mechanizmów ewolucji biologicznej jest ważne zarówno dla ewolucjonistów, jak i dla wszystkich, którzy chcieliby zaprojektować sztuczne systemy adaptacyjne, wzorując się na naturze.

Każdy organizm biologiczny rozwija się w określonym środowisku, w populacji osobników tego samego gatunku. Osobniki te różnią się między sobą, niektóre z nich są lepsze (np. szybsze, sprytniejsze, silniejsze), takie osobniki zazwyczaj przeżywają i dają potomstwo. Inne, te „gorsze”, też mogą przeżyć i dać potomstwo, ale ich szanse są zdecydowanie mniejsze. W ten sposób populacja dąży do coraz lepszego przystosowania się do środowiska, które jest w rzeczywistości zmienne. Można powiedzieć, że środowisko „ocenia” poszczególne osobniki populacji danego gatunku, biorąc pod uwagę ich zewnętrzne cechy. Zestaw zewnętrznych cech osobnika nazywany jest jego **fenotypem**. Zewnętrzne cechy osobnika zależą od zestawu jego genów (*genomu*). Mimo iż presja selekcyjna działa bezpośrednio na fenotyp osobnika, to jego *przystosowanie* jest funkcją spójnego zestawu jego genów. Preferowanie lepszych osobników jest w pewnym sensie procesem optymalizacji – optymalizacji struktur genotypowych, by kodowane fenotypy były możliwie najlepiej przystosowane. W ten sposób cała populacja poprawia swoje przystosowanie do aktualnego środowiska (Dawkins 1995, Hoffman 1983).

Sewall Wright w 1932 r. zaproponował pojęcie **adaptacyjnej topografii** do opisu przystosowania osobników i ilustracji, w jaki sposób ewolucja może przesuwać się w nowe strefy adaptacyjne (Fogel 1992). Osobnik w populacji jest definiowany przez zespół potencjalnych genów, tzw. **genotyp**. Genotypy osobników (populacji) są odwzorowywane na ich fenotypy, które są dalej odwzorowywane na adaptacyjną topografię (*krajobraz przystosowawczy*). Każdy szczyt przystosowawczy odpowiada optymalnemu fenotypowi, a ten z kolei jednemu lub więcej genotypowi. Ewolucja statystycznie przesuwa populację wzdłuż nachylenia krajobrazu, w kierunku szczytu przystosowawczego, w miarę usuwania przez selekcję niedostosowanych osobników.



Rys. 1.1. Każdemu punktowi w przestrzeni **G** odpowiada jeden punkt w **P**, jednemu fenotypowi może odpowiadać kilka genotypów, ale tylko jedna wartość funkcji przystosowania **F**

Podsumowując, żywy organizm może być postrzegany dualnie: jako jego **genotyp** (ukryty w kodowaniu genów) i jego **fenotyp** (sposób odpowiedzi organizmu na genotyp) (rys. 1.1). Można wyspecyfikować dwie przestrzenie: populacyjną **genotypową** (informacyjną), N -wymiarową, przestrzeń **G** i populacyjną **fenotypową** (behawioralną), M -wymiarową przestrzeń **P** (Atmar 1986, Lewontin 1974, za: Fogel 1992). W rzeczywistości N jest dużo większe od M . Można sobie wyobrazić, że nad przestrzenią fenotypową rozpościera się powierzchnia **wartości przystosowania** poszczególnych elementów **F** – jest to funkcja, której

argumentami są poszczególne feny (cechy) osobników, określa ona, na ile dany osobnik jest preferowany przez środowisko, przy czym preferencje te wyrażają się zwiększoną szansą posiadania potomstwa.

Niezależnie od chęci stworzenia sztucznego systemu o dużych zdolnościach adaptacyjnych, modelowanie ewolucji służy ewolucjonistom do lepszego poznania jej mechanizmów. Mylenie aspektów ewolucji prowadziło często do niewłaściwych symulacji, które dawały niezadowalające rezultaty, np. odnośnie szybkości ewolucji¹. Nadal ważne pozostaje pytanie o przedmiot optymalizacji: kto zyskuje w ewolucji: pojedynczy gen, genom, cała populacja? (Dawkins 1995, Fogel 1992).

1.2. ALGORYTMY GENETYCZNE

Szybki (zwłaszcza w ostatnich latach) wzrost zainteresowania algorytmami genetycznymi obserwuje się od czasu opublikowania pracy J. Hollanda (1975). Stosowane są one jako algorytmy przeszukujące (*Search*), optymalizujące (*Optimization*) i uczące się (*Learning*).

Algorytm genetyczny działa w dyskretnym czasie. W każdej jednostce czasu t , w pewnym środowisku Q istnieje populacja osobników tego samego gatunku $P(t)$. Z założenia, że osobniki należą do tego samego gatunku wynika, że konkurują one o zasoby w tym środowisku oraz mogą dowolnie krzyżować się w obrębie całej populacji (tzw. populacja panmiktyczna²). Osobniki „lepsze” od innych mają duże szanse przeżyć i wydać stosunkowo liczne potomstwo. Inne, te „gorsze”, mogą przeżyć przy odrobinie szczęścia, ale ich szanse są zdecydowanie mniejsze niż tych „lepszyc”. W populacji zachodzi proces reprodukcji, tzn. osobniki wydają potomstwo. W czasie reprodukcji następuje wymiana materiału genetycznego pomiędzy osobnikami jako wynik krzyżowania (potomek dziedziczy część materiału genetycznego od jednego rodzica, część od drugiego) i mutacji (losowa zmiana genu). Powstaje popula-

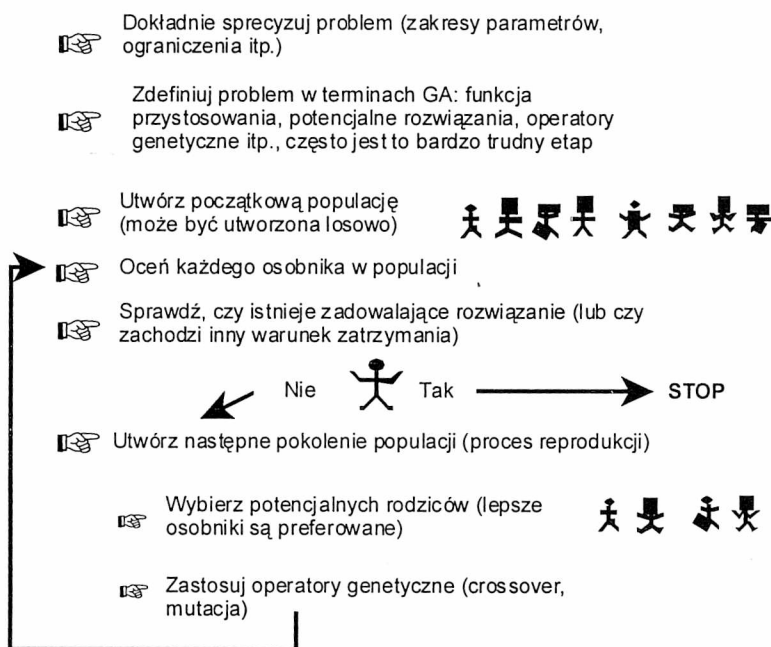
¹ Opozycjoniści Darwina uważają, że tak złożone struktury, jak np. oko, muszą się rozwijać jako całkowicie ukształtowane, inaczej będą nieużyteczne. Tymczasem przypadkowa zmiana – mutacja dająca tak złożone i spójne zmiany – jest wręcz nieprawdopodobna. Prawdą jest, że połowa oka nie jest użyteczna, ale oko w połowie rozwinięte – tak, np. oko z siatkówką, ale bez soczewki gromadziłoby światło i wykrywało ruch, co ułatwiłoby jego posiadaczom np. wykrywać drapieżniki. D. Nilsson i S. Pelger w 1994 r. dokonali symulacji komputerowej ewolucji oka. Symulację rozpoczęli od matematycznego modelu płaskiego obszaru komórek, dopuszczając rozmaite rodzaje „mutacji” (np. zakrzywienie kształtu obszaru komórek, zmiana wrażliwości komórki na światło itp.). Wykrywanie światła i analizowanie widzianych wzorów jest podstawą selekcji. W symulowanym czasie (odpowiadał on 400 000 lat), obszar komórek związał się w głęboką kulistą jamę z małym otworem podobnym do tęczy i – co najdziwniejsze – z soczewką, w której współczynnik załamania światła zmienia się w zależności od miejsca, jak u człowieka (J. Stewart, *Liczyby natury*, Warszawa, Wydawnictwo CIS, 1996, s. 33).

² Panmiksja – losowe kojarzenie się w pary w obrębie populacji bez objawów preferowania jakichkolwiek cech.

cja potomna $P(t+1)$, która jest zazwyczaj lepsza od populacji rodzicielskiej. Aktualna populacja ($P(t+1)$) podlega procesowi reprodukcji, dając kolejne pokolenie, $t+2$ itd. Proces „ulepszania” populacji jest kontynuowany. Jest to model prostego (klasycznego) algorytmu genetycznego.

Ewolucja populacji jest procesem przeszukiwania przestrzeni potencjalnych rozwiązań. W procesach takich jest ważne zachowanie równowagi pomiędzy przekazywaniem najlepszych cech do następnego pokolenia, czyli wykorzystaniem dotychczas znalezionych „obiecujących” rozwiązań (*exploiting*) z jednej strony, a szerokim przeszukiwaniem (*exploring*) z drugiej (Eibe i Schippers 1998). Algorytm genetyczny umożliwia zachowanie takiej równowagi.

Aby zastosować GA do rozwiązania rzeczywistego problemu należy wykonać czynności pokazane na rysunku 1.2.

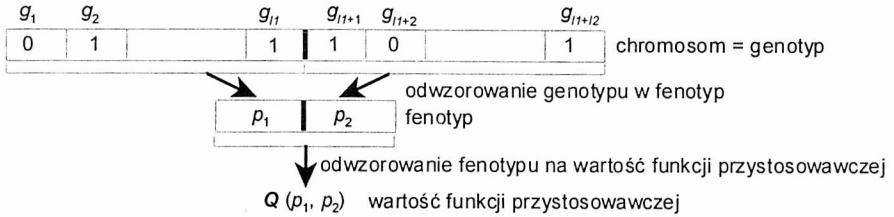


Rys. 1.2. Etapy w stosowaniu algorytmów genetycznych – ogólny schemat

Na poziomie fenotypowym osobnik jest opisany za pomocą wektora m fenów $[p_1, \dots, p_m]$, są to argumenty optymalizowanej funkcji. Na poziomie genotypowym osobnik najczęściej jest opisany przez łańcuch bitów³ $[011\dots01]$ (jego genotyp). Liczba bitów wymagana do reprezentacji pojedynczego osobnika wynika z zakresów fenów oraz

³ Czasem lepsze wyniki daje kod Graya – kolejne liczby całkowite różnią się od siebie wartością jednego bitu, np.: 0000 $\rightarrow$ 0; 0001 $\rightarrow$ 1; 0011 $\rightarrow$ 2; 0010 $\rightarrow$ 3; 0110 $\rightarrow$ 4; 0111 $\rightarrow$ 5; 0101 $\rightarrow$ 6; 0100 $\rightarrow$ 7; 1100 $\rightarrow$ 8 itd.

z przyjętej dokładności (Goldberg 1989, Michalewicz 1996). Przykład reprezentacji osobnika (dla funkcji dwuwymiarowej) w prostym algorytmie genetycznym pokazano na rysunku 1.3.



Rys. 1.3. Reprezentacja binarna osobnika w prostym algorytmie genetycznym dla dwuwymiarowej funkcji

Populację początkową tworzy najczęściej zbiór N losowo wygenerowanych chromosomów (N – zadany rozmiar populacji). Każdy osobnik w bieżącej populacji jest oceniany: liczona jest wartość przystosowawcza zakodowanego fenotypu. Następnie zachodzi proces reprodukcji. Wybierany jest osobnik rodzicielski (w zależności od jego *fitness*). Znane są różne metody selekcji, można opracować też własną metodę, niemniej wszystkie one muszą mieć jedną wspólną cechę: lepszy osobnik musi mieć większe szanse na posiadanie potomstwa, zgodnie z zasadą naturalnej selekcji *przeżywa najlepszy*. Do najbardziej popularnych metod selekcji należą:

Metoda ruletki (nazywana też *stochastyczną z powtórzeniami*) – wartości przystosowania wszystkich osobników w populacji są sumowane, suma stanowi całe koło ruletki. Następnie każdemu osobnikowi przypisywany jest wycinek koła proporcjonalny do jego *fitness*. Koło ruletki jest „obracane” (wybierana jest losowo liczba) i wybierany jest osobnik odpowiadający temu sektorowi na ruletce, w którym mieści się wylosowana liczba. Czynność losowania powtarza się N razy (N – rozmiar populacji).

Metoda próbkowania deterministycznego – wartość oczekiwana liczby potomków dla każdego osobnika w bieżącej populacji liczona jest ze wzoru

$$E_i = N P_i,$$

gdzie:

E_i – oczekiwana liczba potomków i -tego osobnika,

N – liczba osobników w populacji,

P_i – prawdopodobieństwo wybrania i -tego osobnika do reprodukcji, wyrażone wzorem

$$P_i = \frac{f_i}{\sum_{j=1}^N f_j},$$

w którym f_j jest wartością funkcji przystosowania (*fitness*) j -tego osobnika.

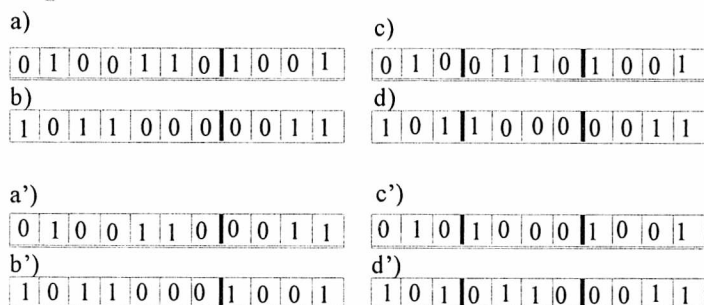
Każdy osobnik z populacji ma tylu potomków, ile wynosi część całkowita wartości oczekiwanej E_i . Następnie osobniki są szeregowane według ułamkowych części E_i (malejąco). Do reprodukcji dobierane są osobniki z początku tej listy, w liczbie potrzebnej do zachowania stałego rozmiaru populacji.

Metoda stochastyczna według reszt z powtórzeniami jest podobna do poprzedniej, opiera się na liczeniu wartości oczekiwanej liczby potomków dla każdego osobnika z populacji i przydzieleniu mu całkowitej części E_i . Części ułamkowe E_i są wykorzystywane do stworzenia koła ruletki, za pomocą którego wybierane są pozostałe osobniki do reprodukcji.

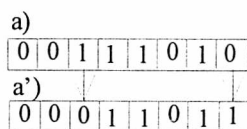
Metoda stochastyczna według reszt bez powtórzeń – analogicznie jak w poprzednich dwóch metodach, część całkowita E_i stanowi liczbę potomków i -tego osobnika, natomiast część ułamkowa jest traktowana jako prawdopodobieństwo dla rozkładu Bernoulliego, brakujące osobniki są losowane zgodnie z tymi prawdopodobieństwami.

Metoda turniejowa – wybierane są dwa osobniki (można stosować turniej więcej niż dwóch osobników) za pomocą metody ruletki, do reprodukcji wybierany jest najlepszy osobnik spośród biorących udział w turnieju.

Wyselekcjonowany osobnik podlega reprodukcji. Reprodukowany osobnik może być krzyżowany z innym osobnikiem (działanie operatora *crosssower*). Operator ten działa z zadaniem prawdopodobieństwem, drugi osobnik do krzyżowania wybierany jest losowo. W czasie krzyżowania, dwa osobniki wymieniają między sobą części genotypu. Powstały po krzyżowaniu potomek podlega *mutacji*. Mutacja zachodzi z zadaniem prawdopodobieństwem, polega ona na losowej zmianie wartości bitu z zera na *jedynkę* lub odwrotnie.



Rys. 1.4. Krzyżowanie jednopunktowe (osobniki a i b, ich potomstwo a' i b') oraz dwupunktowe (c i d, potomstwo c', d')



Rys. 1.5. Przykład działania mutacji (a – osobnik przed mutacją, a' – osobnik z dwoma zmutowanymi bitami)

Po zakończeniu procesu reprodukcji powstaje N nowych osobników, tworzących następne pokolenie populacji – populację potomków. W wielu zastosowaniach algorytmów genetycznych następne pokolenie tworzy się, wybierając N najlepszych osobników spośród rodziców i potomstwa lub też stosuje się tzw. opcję *zachowaj najlepszego*, to znaczy najlepszy osobnik jest kopiowany bez żadnych zmian do następnego pokolenia. Osobniki w powstałym pokoleniu są oceniane – liczona jest ich wartość

przystosowawcza. Kończy to jeden cykl życia populacji. Jeśli w nowo powstałym pokoleniu warunek zatrzymania **GA** nie jest spełniony (np. nie jest znalezione satysfakcjonujące rozwiązanie), to rozpoczyna się kolejny cykl życia populacji – selekcja i reprodukcja osobników. W wielu pracach rozważane są teoretyczne aspekty **GA** (np. Holland 1975, Goldberg i in. 1995, Fogel 1996).

1.3. INNE TECHNIKI EWOLUCYJNE

Strategie ewolucyjne (ESs) zostały rozwinięte w Niemczech przez H.P. Schwefela i I. Rechenberga, na początku lat sześćdziesiątych (Michalewicz 1992, Schwefel 1989). W **ESs** nie stosuje się binarnego kodowania, nie ma też poziomu genotypowego. Ewoluuje populacja chromosomów, gdzie każdy chromosom jest łańcuchem liczb rzeczywistych: bezpośrednio wartości argumentów funkcji celu oraz prawdopodobieństwa mutacji i rekombinacji (w odróżnieniu od **GA**s są one włączone do chromosomu) (Schwefel 1989). We wczesnych aplikacjach populacja składała się z pojedynczego osobnika i jedynie mutacja była wykorzystywana jako czynnik różnicujący potomka od rodzica. Mutacja była tak zaimplementowana, że małe zmiany były bardziej prawdopodobne niż duże. Potomek mógł zastąpić rodzica tylko wtedy, gdy był od niego lepszy. Takie podejście, w którym potomek konkuruje tylko ze swoim rodzicem nazywane jest **strategią dwuelementową** (*two-membered evolution strategy*) i jest oznaczane przez symbol **(1+1)-ES**. Ewolucja populacji μ -elementowej (gdzie $\mu > 1$) jest nazywana **strategią wieloelementową** (*multi-membered evolution strategy*). W tym podejściu stosowana jest już rekombinacja, każdy osobnik może być wybrany z jednakowym prawdopodobieństwem. Podobnie jak w strategii **(1+1)-ES**, w jednym pokoleniu produkowany jest tylko jeden potomek, który zastępuje najgorszego osobnika w populacji. Jeśli wygenerowany potomek jest gorszy od wszystkich w populacji, jest on usuwany i populacja nie zmienia się w tym pokoleniu. Taka strategia (produkcja pojedynczego osobnika w jednym pokoleniu) jest nazywana **($\mu+1$)-ES**. W późniejszym czasie **ES** zostały rozwinięte do postaci znanych jako **($\mu+\lambda$)-ES** i **(μ, λ)-ES**. W strategii **($\mu+\lambda$)-ES** populacja liczy μ osobników, które w jednym pokoleniu produkują λ potomków. Do populacji tworzącej następne pokolenie wybierane są najlepsze osobniki zarówno spośród rodziców, jak i potomków. Różnica pomiędzy strategiami **($\mu+\lambda$)-ES** i **(μ, λ)-ES** polega na tym, że w tej ostatniej do następnego pokolenia wybierane są osobniki tylko spośród potomków (populacja jednopokoleniowa).

Strategie ewolucyjne były rozwijane jako metoda optymalizacji numerycznej, algorytmom genetycznym stawiano natomiast wymagania zdolności przeszukiwania. W obu podejściach służyła darwinowska metoda selekcji oraz populację osobników będących potencjalnymi rozwiązaniami. Podstawowe różnice to sposób reprezentacji osobnika oraz selekcji i reprodukcji. Selekcja, poza tym, że w **ES** jest deterministyczna, zachodzi po procesie reprodukcji, w **GA** natomiast zachodzi przed re-

produkcją (osobniki są wybierane do reprodukcji) i nie jest deterministyczna (większe szanse być wybranym ma lepszy osobnik). W klasycznym **GA** prawdopodobieństwa mutacji i rekombinacji są z reguły stałe dla wszystkich osobników i w czasie całej ewolucji, w **ES** natomiast są one różne dla różnych osobników i ulegają zmianie w czasie ewolucji (same podlegają ewolucji, stanowiąc część chromosomu). W strategiach ewolucyjnych potomek nie spełniający wymaganych warunków jest eliminowany z populacji, nie ma mechanizmów typu nakładanie kar.

Lawrence Fogel (1966) zaproponował ewolucję populacji automatów skończonych w celu predykcji zmian środowiska i nazwał to podejście **programowaniem ewolucyjnym (EP – Evolutionary Programming)**. Opis środowiska to sekwencja symboli ze skończonego alfabetu. Zadanie polega na przewidywaniu kolejnego symbolu na podstawie znanej pewnej ich sekwencji. Miara przystosowania ewoluujących automatów skończonych jest dokładność przewidywania. Każdy osobnik w populacji produkuje jednego potomka. Potomek jest zmutowanym rodzicem. Mutacja jest to losowy proces, polegający na: (1) zmianie symbolu wejściowego, (2) dodaniu nowego stanu, (3) usunięciu jednego z istniejących stanów lub (4) zmianie ścieżki pomiędzy stanami (tranzycji). Podczas tworzenia potomka przez mutację osobnika z istniejącej populacji podejmowana jest decyzja, która z wymienionych mutacji zachodzi. Decyzja ta zapada na podstawie zadanego rozkładu prawdopodobieństwa. Po reprodukcji każdego osobnika populacja chwilowo składa się z podwojonej liczby osobników: rodziców i potomków. Jako następne pokolenie wybierane są najlepsze osobniki z podwojonej populacji, tak że zachowany jest stały rozmiar populacji w kolejnych pokoleniach. W pracach Fogela można znaleźć opisy wielu eksperymentów, w których automatom skończonym stawiano coraz trudniejsze zadania predykcji. Eksperymentowano z zastosowaniem programowania ewolucyjnego do gier. Eksperymenty z dwuosobowymi grami o sumie zerowej pokazały, że **EP** jest w stanie odnaleźć globalnie najlepszą strategię dla prostych gier i niewielkiej liczby graczy (czterech). Wielu badaczy **EP** wiąże nadzieje z ich użytecznością w technice do sterowania (*control applications*). Prace nad stosowaniem programowania ewolucyjnego są dość liczne, ich krótki przegląd można znaleźć w pracy Fogela (1966)⁴.

Rzeczony rozwój metod automatycznego programowania budzi nadzieje na powstawanie poprawnych, dobrze wykonujących swoje zadanie programów komputerowych. John Koza (1990) zaproponował automatyczne tworzenie programu komputerowego, przez jego stopniową ewolucję. Podejście zaproponowane przez Kozę nazywane jest **programowaniem genetycznym (Genetic Programming – GP)**. Populacja programów komputerowych ewoluuje, wytwarzając programy coraz lepiej rozwiązujące określone zadanie. Osobniki (programy) podlegają reprodukcji z udziałem mechanizmów losowych, takich jak mutacja, krzyżowanie czy inne możliwe operatory (np. permutacje).

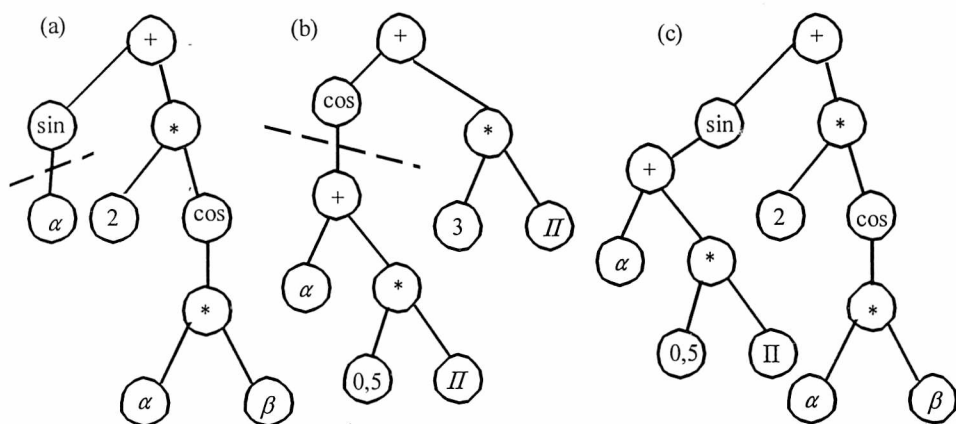
⁴ Można też znaleźć literaturę pod adresem: <http://www.natural-selection.com/people/dbf.html> lub strona *Working papers of Santa Fe Institute* <http://www.santafe.edu/sfi/publications/working-papers.html>.

Przed przystąpieniem do tworzenia populacji początkowej i jej ewolucji należy określić zbiór możliwych funkcji, które mogą wchodzić w skład programów (mogą to być funkcje matematyczne, typu $\sin$, x^3 i inne, operatory, np. $+$, $/$, $-$ itp.), oraz tzw. końcówki (*endings*, np. wartości stałe, jak 5, 12,5). Funkcją przystosowania może być suma odległości pomiędzy rzeczywistymi wynikami a wynikami uzyskiwanymi przez program dla wszystkich danych testowych. Na rysunku 1.6 pokazano dwa programy rodzicielskie (a) i (b), oraz przykładowy program potomny (c), powstały wskutek rekombinacji. Poszczególne drzewa reprezentują wyrażenia:

(a) $\sin(\alpha) + 2\cos(\alpha\beta)$,

(b) $\cos(\alpha + 0,5\Pi) + 3\Pi$,

(c) $\sin(\alpha + 0,5\Pi) + 2\cos(\alpha\beta)$.



Rys. 1.6. Przykład osobników i krzyżowania w GP: (a) pierwszy rodzic, (b) drugi rodzic, (c) jeden z możliwych potomków powstały wskutek krzyżowania (linia przerywana pokazuje punkt krzyżowania)

Ewoluuje programy mają różną długość, reprezentowane w postaci drzewa mają różną głębokość i różną liczbę rozgałęzień, dlatego wymagają reprezentacji w postaci chromosomów o zmiennej długości i dostosowania odpowiednio operatorów genetycznych (np. punkt krzyżowania może być wybierany na krótszym osobniku). Wiele obiecujących obszarów zastosowań GP można znaleźć w pracy (Koza 1996).

1.4. RÓWNOLEGŁE PRZETWARZANIE ALGORYTMÓW GENETYCZNYCH

Doświadczenie wskazuje, że główne trudności w stosowaniu algorytmów genetycznych to:

- znalezienie odpowiedniej reprezentacji rozwiązania,

- odpowiedni dobór operatorów i parametrów algorytmu (wielkość populacji, prawdopodobieństwa działania operatorów, np. mutacji, crossingover),
- czas uzyskiwania satysfakcjonującego rozwiązania, zwłaszcza dla trudnych problemów, wymagających stosowania dość dużych populacji.

Wiele prac jest poświęconych reprezentacji i specjalizowanym operatorom genetycznym, ich celem jest przede wszystkim zwiększenie skuteczności **GA**. Coraz częściej problemy związane z długim czasem przetwarzania próbuje się rozwiązywać przez przetwarzanie **GA** na komputerach równoległych lub w środowisku rozproszonym.

W jednej z najwcześniejszych prac teoretycznych J. Holland (1995) rozpoznał współbieżną naturę algorytmów genetycznych. Zaproponował nawet klasę maszyn równoległych, ale nie zostały one zbudowane. Za jedną z pierwszych implementacji **GA** na architekturach równoległych uznać można pracę J. Grefenstette'a (1981), algorytm synchroniczny, scentralizowany (za: Cantu-Paz 1997a). Później pojawiały się kolejne implementacje, z asynchronicznym obliczaniem funkcji przystosowania elementów populacji, z pamięcią dzieloną itp. D.B. Goldberg (1989) rozważał kilka współbieżnych implementacji **GA**, przedstawił ideę czterech sposobów implementacji:

- algorytm **synchroniczny scentralizowany**,
- algorytm **pólsynchroniczny scentralizowany**,
- algorytm **asynchroniczny rozproszony**,
- algorytm **sieciowy**.

Poszczególne implementacje różnią się sposobem komunikacji procesów między sobą.

Historia równoległych algorytmów genetycznych (**PGA** – *Parallel Genetic Algorithms*) nie jest zbyt długa (sięga początku lat osiemdziesiątych). Ostatnio są one przedmiotem wielu prac, głównie jednak empirycznych, brakuje analizy teoretycznej (Seredyński 1998). Opisywane eksperymenty dotyczą różnych zadań, stosują różne operatory genetyczne, wykonywane są w różnych środowiskach, korzystają z różnej architektury, co powoduje, że jakiegokolwiek porównanie ich staje się praktycznie niemożliwe. Erick Cantu-Paz (1995, 1997a, 1999) zaproponował pewien sposób usystematyzowania znanych z literatury podejść.

Pierwsze podejście do zrównoleglania **GA** to **równoległość globalna** (*global parallelization*). Oznacza to, że zarówno operatory genetyczne, jak i liczenie funkcji przystosowania są wykonywane równolegle. Wszystkie osobniki w populacji konkurują między sobą i mogą dowolnie krzyżować się między sobą. Wynika z tego, że taka implementacja nie wymaga wprowadzania dodatkowych operatorów w stosunku do klasycznego algorytmu i jest stosunkowo prosta w implementacji. Równoległość globalna nie wymaga określonej architektury, może być efektywnie implementowana zarówno na komputerach z pamięcią dzieloną, jak i rozproszoną.

Inne podejście to **gruboziarniste PGA** (*coarse grained*). Proces przetwarzania równoległego nazywany gruboziarnistym, jeśli czas przeznaczony na obliczenia jest dużo większy od czasu potrzebnego na komunikację pomiędzy procesami. Czasami

ten rodzaj **PGA** nazywany jest **rozproszonym GA**, ponieważ najczęściej implementowany jest na maszynach MIMD z rozproszoną pamięcią. Cała populacja jest podzielona na kilka stosunkowo licznych podpopulacji. W analogii do biologii, podpopulacje takie są nazywane **demami** (Kwaśnicka i Markowska-Kaczmarska 1996, Cantú-Paz 1999). Każdy dem ewoluuje prawie niezależnie od siebie. Osobniki konkurują między sobą w obrębie demów, jednakże od czasu do czasu osobniki z jednego demu mogą przechodzić do innego i wносить do niego swój materiał genetyczny. Dla takiego **GA** należy zdefiniować dodatkowy operator, odpowiedzialny za wymianę materiału genetycznego pomiędzy demami. Najczęściej dodatkowy operator to **migracja**: z zadaniem prawdopodobieństwem, niektóre osobniki są przesuwane z jednego demu do innego. Podczas definiowania migracji należy określić jej prawdopodobieństwo oraz dwa inne czynniki: po pierwsze, który osobnik z danego demu ma migrować – losowy, najlepszy, najgorszy? Po drugie, do którego demu ma przejść migrujący osobnik? Najczęściej migruje najlepszy osobnik z danego demu, choć nie zawsze daje to wyraźnie lepsze rezultaty niż przesuwanie losowo wybranego osobnika (Kwaśnicka i Markowska-Kaczmarska 1996).

Ostatnie podejście do **PGA** to **drobnoziarniste** zrównoleganie (*fine grained*). Cała ewoluująca populacja jest podzielona na dużą liczbę stosunkowo mało licznych demów, najlepiej, kiedy dysponuje się oddzielnym procesorem dla każdego osobnika populacji. Wymaga to komputera o dużej liczbie procesorów (*massively parallel computers*). W tym podejściu, podobnie jak w poprzednim, osobniki konkurują i krzyżują się tylko w obrębie swoich demów. Liczebność demów jest zwykle znacznie mniejsza niż algorytmów genetycznych przetwarzanych szeregowo, istnieje zatem niebezpieczeństwo przedwczesnej zbieżności. Zapobiega temu, przynajmniej częściowo, istnienie migracji. W skrajnych przypadkach na pojedynczym procesorze przetwarzany jest pojedynczy osobnik. Wtedy nie ma wydzielonych demów, podpopulację stanowią sąsiadujące osobniki (z sąsiednich procesorów), obszary sąsiedztwa zachodzą na siebie.

W literaturze przedmiotu można znaleźć inne podejścia, będące np. kombinacją wymienionych wyżej. Nazywane są one **podejściem hybrydowym**. Określenia tego nie należy kojarzyć z *technikami hybrydowymi*, które oznaczają połączenie różnych technik rozwiązywania problemu, np. połączenie sieci neuronowych z algorytmami genetycznymi.

2. PRZYKŁADY SYSTEMÓW Z WBUDOWANYMI EA

W wielu praktycznych zastosowaniach pojedyncza technika sztucznej inteligencji nie daje zadowalającego rezultatu. Poszczególne techniki (logika rozmyta, sieci neuronowe, systemy ekspertowe) mają swoje zalety, ale także wykazują pewne niedociągnięcia. Możliwe jest zatem, że systemy łączące w sobie różne techniki mogą okazać się bardziej odpowiednie dla wielu trudnych zadań, zarówno ze względu na jakość, jak i szybkość dawanego rozwiązania. Systemy takie to **hybrydowe systemy inteligentne** (Medsker 1995). Kombinacje **GA** i **logiki rozmytej (FL – Fuzzy Logic)** są stosowane na przykład w systemach sterowania. Dla systemów opartych na sieciach neuronowych **GA** są użyteczne do poprawy działania sieci, wstępnego przygotowywania danych dla **NN (Neural Networks)**, uczenia i(lub) projektowania struktury sieci, w systemach informacyjnych – do przyspieszenia i polepszenia odpowiedzi na zadane systemowi pytanie. Systemy hybrydowe są postrzegane jako nowy, szybko rozwijający się kierunek sztucznej inteligencji. Algorytmy genetyczne często są jednym z elementów takich systemów.

2.1. GA W SYSTEMACH INFORMACYJNYCH

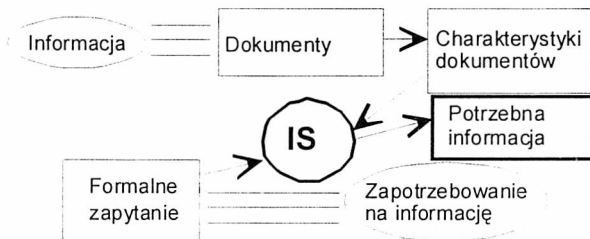
Ogromna ilość zapamiętywanej informacji, zajmująca tysiące gigabajtów, musi być dostępna dla człowieka (lub innego systemu komputerowego), gdy jest mu potrzebna. Tradycyjne metody nie są w stanie sprostać potrzebom ze względu na rozmiar przetwarzanej i przeszukiwanej informacji. W związku z tym wiele osób szuka nowych rozwiązań w dziedzinie przetwarzania informacji.

Prosty model systemu informacyjnego (**IS – Information System**) jest pokazany na rysunku 2.1 (Kwaśnicka i Nowostawski 1997). W systemach informacyjnych informacja jest reprezentowana w postaci dokumentów. Są metody tworzenia charakterystyk dokumentów, oceny i wyszukiwania odpowiednich dokumentów przez przeglądanie ich charakterystyk. Odległość między dwoma wektorami, wektorem zapytania i wektorem charakterystyki dokumentu, jest liczona jako odległość kartezjańska lub wykorzystuje się do tego celu metodę kosinusów, która jest stosunkowo prosta. Klu-

czowym problemem jest szybkość pracy takich systemów. Obliczanie odległości pomiędzy zapytaniem a charakterystyką każdego dokumentu zajmuje dużo czasu, zwłaszcza, że bazy zawierają dużą liczbę dokumentów. Poszukuje się zatem sposobów na skrócenie czasu oczekiwania na odpowiedź systemu. Jedną z takich możliwości jest opracowanie metody dającej satysfakcjonującą odpowiedź bez konieczności przeszukiwania wszystkich charakterystyk.

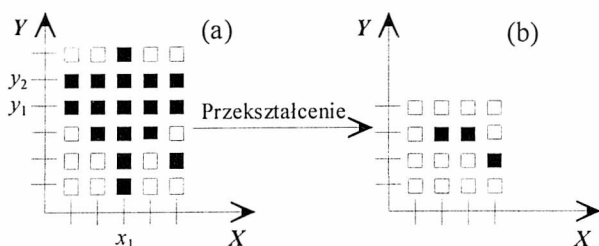
Zastosowanie algorytmów genetycznych w systemach informacyjnych powinno pozwolić na znajdowanie satysfakcjonującej odpowiedzi bez sprawdzania wszystkich dokumentów. **GA** umożliwiają szukanie satysfakcjonującego rozwiązania przy analizie tylko części (najbardziej obiecujących) przestrzeni potencjalnych rozwiązań. Stosując zatem **GA** można obliczać tylko odległości pomiędzy wektorem zapytania a podzbiorem charakterystyk dokumentów. Dodatkowo, działanie systemu informacyjnego można przyspieszyć przez zastosowanie obliczeń równoległych. Wadą takiego podejścia jest brak pewności, że odpowiedź systemu jest najlepsza z możliwych.

Są jednak problemy z włączeniem **GA** do zadania przeszukiwania przestrzeni charakterystyk dokumentów. Wśród przyczyn problemów można wymienić dyskretny charakter przestrzeni charakterystyk dokumentów, wiele *optimów* lokalnych, złożoność zapytań. Odpowiedź systemu powinna zawierać zbiór dokumentów odpowiadających zapytaniu. Na potrzeby przetestowania zaproponowanego podejścia opracowano system nazwany *FUSION* (Kwaśnicka i Nowostawski 1997) i przeprowadzono wiele symulacji. Jako model przestrzeni charakterystyk dokumentów można przyjąć wielowymiarową hiperkostkę, dyskretną, aczkolwiek w pewnym sensie ciągłą (tzn. że każda dyskretna wartość pomiędzy $[a_i, b_i]$, gdzie a_i i b_i są granicznymi wartościami hiperkostki w i -tym wymiarze, jest dopuszczalną wartością dla tego rozmiaru). W rzeczywistości, przestrzeń charakterystyk dokumentów jest pełna „dziur”, tzn. że istnieją punkty w tak skonstruowanej przestrzeni, dla których nie istnieją odpowiednie dokumenty w bazie. Utrudnia to implementację **GA**, dlatego też rzeczywista przestrzeń charakterystyk jest przekształcana jak na rysunku 2.2. Przekształcona zawiera na ogół znacznie mniejszą liczbę „dziur”. Wskutek działania operatorów genetycznych powstają niepo-



Rys. 2.1. Prosty model systemu informacyjnego

Są jednak problemy z włączeniem **GA** do zadania przeszukiwania przestrzeni charakterystyk dokumentów. Wśród przyczyn problemów można wymienić dyskretny charakter przestrzeni charakterystyk dokumentów, wiele *optimów* lokalnych, złożoność zapytań. Odpowiedź systemu powinna zawierać zbiór dokumentów odpowiadających zapytaniu. Na potrzeby przetestowania zaproponowanego podejścia opracowano system nazwany *FUSION* (Kwaśnicka i Nowostawski 1997) i przeprowadzono wiele symulacji. Jako model przestrzeni charakterystyk dokumentów można przyjąć wielowymiarową hiperkostkę, dyskretną, aczkolwiek w pewnym sensie ciągłą (tzn. że każda dyskretna wartość pomiędzy $[a_i, b_i]$, gdzie a_i i b_i są granicznymi wartościami hiperkostki w i -tym wymiarze, jest dopuszczalną wartością dla tego rozmiaru). W rzeczywistości, przestrzeń charakterystyk dokumentów jest pełna „dziur”, tzn. że istnieją punkty w tak skonstruowanej przestrzeni, dla których nie istnieją odpowiednie dokumenty w bazie. Utrudnia to implementację **GA**, dlatego też rzeczywista przestrzeń charakterystyk jest przekształcana jak na rysunku 2.2. Przekształcona zawiera na ogół znacznie mniejszą liczbę „dziur”. Wskutek działania operatorów genetycznych powstają niepo-



Rys. 2.2. Przekształcanie przestrzeni charakterystyk dokumentów do przestrzeni, na której operuje **GA**

prawne osobniki w populacji (punkty będące „dziurami”). W *FUSSION* osobniki takie są usuwane z populacji (nie stosuje się funkcji kar ani naprawiania).

Do liczenia odległości (oceny rozwiązań generowanych przez **GA**) brana jest pod uwagę rzeczywista przestrzeń charakterystyk (nie ta przekształcona). W terminach **GA** przekształcona przestrzeń odpowiada poziomowi genotypowemu, rzeczywista zaś – poziomowi fenotypowemu. W *FUSSION* genotyp jest wektorem stanowiącym charakterystykę dokumentu. Selekcja jest dokonywana metodą próbkowania deterministycznego, stosowana jest mutacja (odrzuca, jeśli w efekcie daje „dziurę”), krzyżowanie jednopunktowe. Przystosowanie osobnika stanowi odległość pomiędzy zakodowanym w chromosomie punktem w przestrzeni charakterystyk dokumentów a zadanym systemowi pytaniem.

System został zaimplementowany w środowisku **PVM** (*Parallel Virtual Machine*), przeszukiwania są zatem rozłożone między kilka procesorów. W systemie proces inicjujący zajmuje dość dużo czasu, niemniej po zainicjowaniu czas potrzebny na uzyskanie odpowiedzi jest stosunkowo krótki. Dla 300 000 dokumentów metoda klasyczna jest trochę wolniejsza niż z **GA**. Dla 600 000 lub więcej działanie zaproponowanego systemu jest dużo szybsze, jakość uzyskiwanych odpowiedzi jest natomiast lepsza w klasycznym podejściu. Być może wprowadzenie elementów pozwalających populacji zajmować różne (lokalne) szczyty przystosowawcze polepszyłyby jakość odpowiedzi.

Inny sposób zastosowania **GA** w systemach informacyjnych jest omówiony w pracy Petry i in. (1993). Autorzy zastosowali binarną macierz odwzorowującą dokumenty na binarne symbole. Zapytanie jest wektorem **termów** (zbiorem indeksów). Przystosowaniem może być miara podobieństwa *l*-tego dokumentu do wektora zapytania, autorzy stosują metodę kosinusową miary odległości. Prezentowane przez autorów wyniki są wynikami wstępnymi, autorzy zapowiadają modyfikację ich modelu w celu uwzględnienia logiki rozmytej.

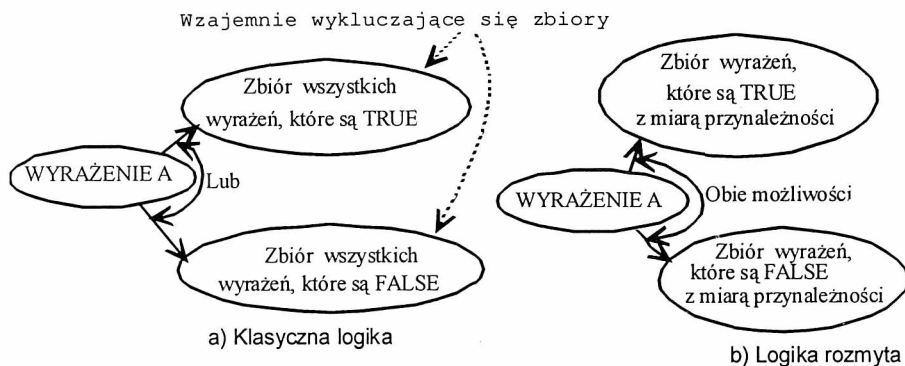
2.2. ŁĄCZENIE ALGORYTMÓW GENETYCZNYCH Z LOGIKĄ ROZMYTĄ

Rozumowanie komputera zastosowane w systemach ekspertowych różni się od rozumowania mózgu ludzkiego. „Wnioskowanie” komputera polega na przetwarzaniu ściśle zdefiniowanych faktów, które mogą być tylko prawdziwe lub tylko fałszywe. Ludzie posługują się zdrowym rozsądkiem, który pozwala rozumieć i analizować fakty, które mogą być częściowo prawdziwe¹.

¹ Na przykład przedział temperatury 0÷40 °C można podzielić na trzy przedziały wartości: *zimno*, *ciepło*, *gorąco*. Jeśli mamy temperaturę 15 °C, to czy to jest jeszcze zimno, czy już ciepło? Dwuwartościowa logika pozwala, by określenie *zimno* było albo prawdziwe albo fałszywe. W rzeczywistości temperatura 15 °C może oznaczać, że w pewnym stopniu jest jeszcze *zimno*, a w pewnym już *ciepło*.

2.2.1. SYSTEMY Z LOGIKĄ ROZMYTĄ – WPROWADZENIE

W logice klasycznej wartość logiczna wyrażenia może być odwzorowana w dwa wzajemnie wykluczające się zbiory (rys. 2.3a), obiekt nie może jednocześnie należeć do zbioru i jego dopełnienia. W logice rozmytej zbiory wyrażeń prawdziwych (TRUE) i fałszywych (FALSE) nie są rozłączne, wartość logiczna danego wyrażenia może należeć w części do zbioru zdań prawdziwych (z pewną miarą przynależności) i w części do zbioru zdań fałszywych (rys. 2.3b).



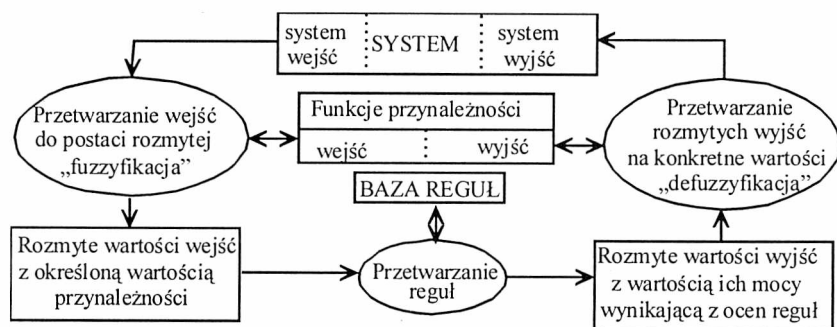
Rys. 2.3. Różnica między logiką klasyczną (a) a logiką rozmytą (b)

Logika rozmyta jest gałęzią informatyki, której celem jest umożliwienie komputerom tworzenia zdroworozsądkowego, zróżnicowanego obrazu niepewnej rzeczywistości. Pierwsze badania nad logiką rozmytą i jej paradoksami zaczęły się w początkach wieku, kiedy Bertrand Russel doszedł do wniosku, że stary, grecki paradoks jest kluczem do zrozumienia teorii zbiorów i logiki matematycznej². Logika rozmyta nakłada tylko jedno ograniczenie: suma przynależności danego obiektu do rozłącznych zbiorów musi wynosić jeden (np. jeśli temperatura jest w 30% *ciepła*, to musi być w 70% *nieciepła*).

W latach sześćdziesiątych przełomową pracę zatytułowaną *Zbiory rozmyte* opublikował Lotfi Zadeh. Zastosował on logikę Jana Łukasiewicza do każdego elementu zbioru i wyprowadził pełną algebrę zbiorów rozmytych. Pierwsze próby zastosowania nastąpiły dopiero w połowie lat siedemdziesiątych: Ebrahim H. Mamdani z Queen Mary College w Londynie opracował program sterujący maszyną parową, oparty na logice rozmytej. Logika rozmyta może znajdować zastosowanie nie tylko w sterowaniu, powstała raczej z potrzeb nauk nazywanych czasami „miękkimi naukami”, takich

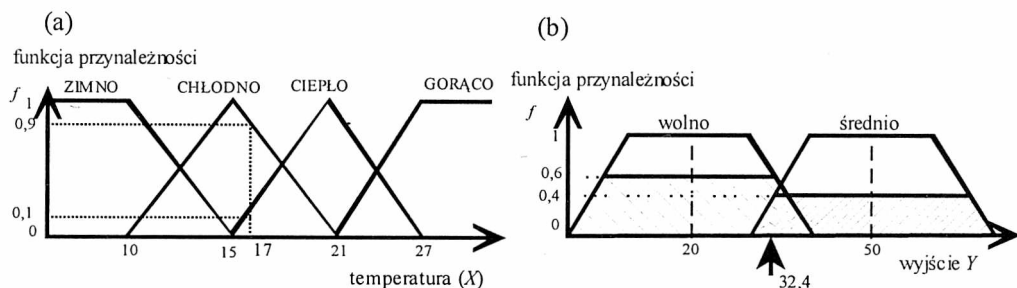
² W starej łamigłówce pewien Kreteńczyk stwierdza, że „wszyscy Kreteńczycy kłamią”. Nasuwa się pytanie, czy wobec powyższego on też kłamie? Jeśli odpowiedź będzie „tak”, to znaczy, że mówi prawdę, a zatem kłamie. Jeśli ten Kreteńczyk mówi prawdę (odpowiedź „nie”), to znaczy, że jest kłamcą mówiąc, że wszyscy kłamią. Oba przypadki prowadzą do sprzeczności według logiki klasycznej, ponieważ zdanie jest jednocześnie prawdziwe i fałszywe.

jak psychologia, biologia. W technice systemy z logiką rozmytą (będziemy je nazywać **systemami rozmytymi – FLS**) są dość popularne jako systemy sterowania. Schemat przykładowego systemu **FLS** pokazano na rysunku 2.4.



Rys. 2.4. Przepływ informacji w typowym systemie sterowania z logiką rozmytą

Przykładowe zbiory rozmyte i ich funkcje przynależności dla sterowania wentylatorkiem pokazano na rysunku 2.5: a – dla wejść (temperatura otoczenia), b – dla wyjść (obroty silnika).



Rys. 2.5. Przykładowe zbiory rozmyte dla wejść i wyjść systemu **FLS**

Za pierwsze zastosowanie logiki rozmytej do sterowania można uznać system sterowania piecem w cementowni, wprowadzony przez firmę z Kopenhagi F.L. Smidth & Company. Najczęściej podawanym przykładem rozmytego systemu sterowania jest sterownik wagonów metra w Sendai, w Japonii. Sterownik ten jest stosowany od końca lat osiemdziesiątych przez firmę Hitachi. Według zgodnych ocen firmy i pasażerów, sterownik ten przewyższa nie tylko standardowe sterowniki, ale i ludzką obsługę.

2.2.2. POŁĄCZENIE GA I FLS

Algorytmy genetyczne są bardzo użyteczne jako narzędzie do optymalizacji funkcji przynależności. Systemy z logiką rozmytą są wygodne w stosowaniu i efektywne,

lecz duże systemy są trudne do zaprojektowania. **FLS** są bardzo wrażliwe na niektóre ich parametry, zwłaszcza zaś na parametry funkcji przynależności. Z reguły projektowanie zbiorów rozmytych (ich liczba, kształt, zakresy itp.) odbywa się ręcznie, metodą prób i błędów, z udziałem ekspertów. Ten etap projektowania **FLS** jest bardzo trudny i czasochłonny. **GA** mogą być stosowane do doboru parametrów systemu rozmytego lub do dostrojenia ich wartości (Medsker 1995). Populacja parametrów systemu jest oceniana z uwzględnieniem ogólnego działania projektowanego systemu. W niektórych przypadkach **GA** mogą być użyte do uczenia systemu rozmytych reguł. Wtedy chromosomy zawierają informacje o warunkach reguły rozmytej (część IF reguły) i o odpowiadającej tym warunkom akcji (część THEN reguły) (Medsker 1995). System może nauczyć się odpowiednich reguł lub dostroić zbiór zadanych rozmytych reguł. Powstało wiele prac poświęconych automatyzacji projektowania takich systemów z zastosowaniem algorytmów genetycznych (np. Heider i in. 1994, Ch.L. Karr w Medsker 1995, s. 187–202).

2.2.3. PROJEKTOWANIE ROZMYTEGO SYSTEMU STEROWANIA – PRZYKŁAD

Algorytm genetyczny zastosowano tu do generowania zestawu rozmytych reguł dla **FLC-GA** (*Fuzzy Logic Controller*), wraz z liczbą i granicami zbiorów rozmytych zarówno dla zmiennych wejściowych systemu sterującego układem odwróconego wahadła, jak i zmiennych wyjściowych (Kwaśnicka i in. 1998). Ponieważ celem jest tu pokazanie metody i sprawdzenie jej efektywności, sam układ odwróconego wahadła jest uproszczony.

2.2.3.1. OPIS FIZYCZNEGO SYSTEMU

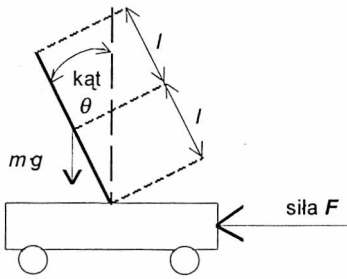
Pełna wersja układu to odwrócone wahadło zaczepione na wózku, który może się poruszać po ograniczonej trasie. Zadaniem systemu sterującego jest utrzymanie wahadła w pozycji pionowej (prostopadle do wózka, rys. 2.6).

Jeśli wahadło jest pionowo, mówimy, że kąt jego odchylenia θ jest równy zeru. Jeśli θ jest różne od zera, należy przyłożyć siłę F do wózka skierowaną w lewo (ujemną) lub w prawo (dodatnią).

Zmienne stanu systemu wahadła to:

- θ – kąt odchylenia od pionu,
- ω – prędkość kątowna wahadła,
- poz – pozycja wózka na trasie,
- v – prędkość wózka.

Zmienną sterującą (czyli zmienną wyjściową systemu sterującego) jest F – siła przyłożona do wózka przez krótki okres, np. 0,02 s po zidentyfikowaniu odchylenia wahadła.



Rys. 2.6. Schemat układu odwróconego wahadła

Jak już wspomniano, do celów testowania metody system ten został uproszczony do dwóch zmiennych wejściowych i jednej wyjściowej. Uproszczenie polega na pominięciu ograniczonej trasy dla wózka. Eliminujemy w ten sposób pozycję wózka i jego prędkość. Pozostają dwie zmienne wejściowe systemu sterującego – są nimi θ i ω oraz jedna zmienna wyjściowa – F . System odwróconego wahadła określają znane równania matematyczne (Kwaśnicka i in. 1998). Przyjęte dziedziny zmiennych stanu i zmiennych sterujących to: $\theta \in [-12,0, 12,0]$ (w stopniach), $\omega \in [-500, 500]$ (w stopniach na sekundę), $F \in [-10, 10]$ (w niutonach). Równania te (oraz przyjęte wartości stałych występujących w tych równaniach) są stosowane do sprawdzania, na ile generowany **FLC-GA** dobrze steruje układem wahadła. Aby projektowany system sterujący był bardziej uniwersalny, początkowy stan układu jest ustalany losowo i cykl sterowania jest powtarzany dla zadanej liczby początkowych stanów. Dopiero na tej podstawie osobniki w populacji są oceniane.

2.2.3.2. OPIS SYSTEMU STERUJĄCEGO – FLC-GA

Projektowany system to zestaw reguł i zestawy zbiorów rozmytych dla dwóch zmiennych wejściowych – prędkości kątowej wahadła i jego kąta odchylenia od pionu, oraz zmiennej wyjściowej – przykładanej siły. Założono, że każda zmienna wejściowa może mieć maksymalnie siedem zbiorów rozmytych, zmienna wyjściowa natomiast może przyjmować dwie wartości, tzn. nie jest zmienną rozmytą, a więc nasz system nie wymaga procesu „defuzyfikacji” (wyostrzania). Reguły mają typową dla systemów ekspertowych postać

JEŚLI przesłanka, TO akcja,

gdzie przesłanki odnoszą się do zmiennych stanu (zmiennych wejściowych **FLC-GA**), a akcje odnoszą się do zmiennej wyjściowej – czy i jaką siłą należy zadziałać na wózek. Przy założeniu maksymalnie siedmiu zbiorów rozmytych dla każdego wejścia, do opisu możliwych kombinacji wejść, zakładając, że na wejściu systemu może również wystąpić pojedyncza zmienna, potrzebujemy $(7 + 1)^2 - 1 = 63$ reguły (tab. 2.1). Zadaniem **GA** projektującego system jest zarówno znalezienie systemu dobrze sterującego układem wahadła, jak i minimalizacja liczby używanych reguł.

Schemat kodowania (definicja chromosomu)

Chromosom to kompletny zestaw reguł i zbiorów rozmytych, jest łańcuchem bitów. Do reprezentacji pojedynczego zbioru potrzebne są cztery punkty i tzw. flaga, oznaczająca czy dany zbiór bierze udział w regułach, czy nie. Na kodowanie każdego punktu przeznaczamy osiem bitów, na flagę – jeden. Dla dwóch zmiennych wejścio-

wych i siedmiu zbiorów dla każdej z nich potrzebne jest $7 \cdot 2 + 4 \cdot 8 \cdot 7 \cdot 2 = 462$ bity. Założenie, że trzeci i czwarty punkt jednego zbioru jest jednocześnie pierwszym i drugim punktem następnego zbioru oraz dwa pierwsze punkty pierwszego zbioru to dolna granica dziedziny zmiennej, dwa ostatnie punkty ostatniego zbioru są górną granicą zmiennej (to zapewnia pokrycie zbiorami rozmytymi całej dziedziny zmiennej wejściowej) pozwala znacznie zredukować długość chromosomu potrzebną do zakodowania zbiorów: wystarczy $7 \cdot 2 + 2 \cdot 8 \cdot (7 - 1) \cdot 2 = 206$ bitów. Bit flagi równy zeru oznacza, że dany zbiór nie jest używany w regułach. Po zakodowaniu zbiorów rozmytych dla obu zmiennych wejściowych, kodujemy wszystkie możliwe 63 reguły. Każda reguła może mieć tylko dwie możliwe akcje, może też nie być w ogóle używana przez system. Musimy zatem zakodować trzy potencjalne wartości dla każdej reguły (dwa bity), czyli długość chromosomu zwiększa się do $206 + 63 \cdot 2 = 332$ bity. Postać chromosomu to

$$f_1 \text{ zbiór}_1 f_2 \text{ zbiór}_2 \dots f_n \text{ zbiór}_n R_1 R_2 \dots R_m,$$

gdzie:

f_i – flaga i -tego zbioru rozmytego,

zbiór_i – i -ty zbiór rozmyty,

R_i – i -ta reguła według numeracji jak w tabeli 2.1.

Jako ciąg uczący użyto 57 przykładów (par wejścia i odpowiadającego im wyjścia) otrzymanych z odpowiednich równań matematycznych (Nuno 1997). Zastosowano proste krzyżowanie jednopunktowe i mutację, selekcji osobników dokonywano metoda turniejową.

Funkcja przystosowania

Projektowanie systemu **FLC-GA** odbywało się w dwóch fazach: faza uczenia z przykładów oraz faza oceny i dostrojenia. Wykorzystywano dwie funkcje przystosowania – inne w poszczególnych fazach projektowania systemu. Pierwsza funkcja oceniająca to ważona suma zakumulowanej różnicy pomiędzy oczekiwanym wyjściem (z ciągu uczącego) a wyjściem dawanym przez zdekodowany chromosom dla wszystkich 57 przykładów oraz liczby stosowanych reguł

Tabela 2.1. Zestaw wszystkich możliwych reguł

	WE 1	WE 2
R ₁ :	JEŚLI zbiór_{1i}	TO akcja
R ₂ :	JEŚLI zbiór_{1i}	zbiór_1 TO akcja
R ₃ :	JEŚLI zbiór_{1i}	zbiór_2 TO akcja
...	...	
R ₈ :	JEŚLI zbiór_{1i}	zbiór_7 TO akcja
R ₉ :	JEŚLI zbiór_{2i}	TO akcja
R ₁₀ :	JEŚLI zbiór_{2i}	zbiór_1 TO akcja
R ₁₁ :	JEŚLI zbiór_{2i}	zbiór_2 TO akcja
...	...	
R ₁₆ :	JEŚLI zbiór_{2i}	zbiór_7 TO akcja
...	...	
R ₄₉ :	JEŚLI zbiór_{7i}	TO akcja
R ₅₀ :	JEŚLI zbiór_{7i}	zbiór_1 TO akcja
R ₅₁ :	JEŚLI zbiór_{7i}	zbiór_2 TO akcja
...	...	
R ₅₆ :	JEŚLI zbiór_{7i}	zbiór_7 TO akcja
R ₅₇ :	JEŚLI i	zbiór_1 TO akcja
R ₅₈ :	JEŚLI i	zbiór_2 TO akcja
...	...	
R ₆₃ :	JEŚLI i	zbiór_7 TO akcja

$$E = \sum_{i=1}^n |EV(i) - CV(i)|,$$

gdzie:

- E – błąd systemu zakodowanego w chromosomie,
- $EV(i)$ – oczekiwana wartość na wyjściu w i -tym przykładzie,
- $CV(i)$ – obliczona wartość wyjścia dla i -tego przykładu,
- n – liczba przykładów ($n = 57$).

Dla większej czytelności otrzymany wynik jest podawany w procentach (100% oznacza osobnika o najlepszym możliwym przystosowaniu). Faza uczenia kończy się, gdy średnie przystosowanie populacji wynosi 90%.

Druga funkcja przystosowania jest inna. Ewolucja rozpoczyna się od losowania wartości zmiennych stanu z przedziałów $\theta \in [-5, 5]$ i $\omega \in [-150, 150]$. Z równań liczona jest wartość nowych zmiennych stanu (po zadziałaniu siłą na wózek). Akumulowana jest wartość bezwzględna otrzymanego kąta, jest to wartość odchylenia wahadła. Jeśli żadnej reguły nie można zastosować, do akumulatora dodaje się kąt wartości 12 stopni w radianach ($12\pi/180$). W takiej sytuacji, jak również wówczas, kiedy zmienne stanu wychodzą poza ich zakresy, zakodowany w chromosomie system nie jest w stanie sterować wahadłem, stan systemu jest ponownie inicjowany losowo z zadanego przedziału zmiennych (pozwala to stwierdzić, czy ten chromosom jest równie zły dla innych warunków początkowych). Gdy sterowanie się utrzymuje, podana procedura jest powtarzana 1000 razy, po czym zmienne stanu są ponownie losowo inicjowane (aby stwierdzić, czy chromosom będzie równie dobry przy innych warunkach początkowych). W ostatnim przypadku czynność ta jest powtarzana cztery razy. Błąd tym razem jest liczony według następującego wzoru

$$E = \sum_{i=1}^n |\theta| + r w_1 + s w_2,$$

gdzie:

- θ – wartość kąta odchylenia,
- r – liczba reguł stosowanych przez zakodowany w chromosomie system,
- s – liczba zbiorów wykorzystywanych przez zakodowany w chromosomie system,
- w_1, w_2 – odpowiednio: waga dla liczby reguł i liczby zbiorów rozmytych,
- n – liczba iteracji.

Ostateczna wartość przystosowania jest liczona tak samo jak w fazie pierwszej.

Rezultaty

Przedstawione dalej rezultaty otrzymano przy ewolucji populacji liczącej 30 osobników, krzyżowaniu z prawdopodobieństwem 0,95 i mutacjami z prawdopodobieństwem 0,001 (0,005 dla drugiej fazy). Wyniki eksperymentów wskazują, że te wartości dawały najlepsze wyniki. Waga dla liczby reguł wynosiła $w_1 = 0,5$, dla liczby zbiorów $w_2 = 8$. Po pierwszej fazie uczenia najlepszy osobnik wykorzystywał 10 zbiorów rozmytych (4 dla pierwszej zmiennej wejściowej, 6 dla drugiej) i 18 reguł, jego przy-

stosowanie wynosiło 82,4% (średnie przystosowanie populacji – 51,3%). Ta sama populacja osiągnęła dużo lepsze wyniki w fazie dostrajania: średnie przystosowanie było równe 90,2%, najlepsze – 92,7%. Teraz najlepszy osobnik to system z trzema zbiorami rozmytymi dla każdego wejścia i pięcioma regułami. Wszystkie osobniki w populacji były w stanie utrzymać wahadło w równowadze przez 4000 iteracji.

W innym eksperymencie populacja zaledwie w ciągu 12 pokoleń w fazie uczenia i 19 w fazie dostrajania uzyskała całkiem dobre rezultaty. Najlepszy osobnik po fazie uczenia, z przystosowaniem 90,1%, stosował 7 zbiorów rozmytych (4 dla pierwszej zmiennej i 3 dla drugiej) oraz 12 reguł. Średnie przystosowanie populacji wyniosło 86,1%. Po fazie dostrajania najlepszy osobnik wykorzystywał jedynie 5 zbiorów rozmytych (3 dla pierwszego wejścia i 2 dla drugiego) i 7 reguł. 28 spośród trzydziestu osobników było w stanie sterować wahadłem przez 4000 iteracji.

Przedstawione pokrótce wyniki wskazują, że takie podejście może być konkurencyjne w stosunku do „ręcznego” projektowania rozmytych systemów sterowania.

W omawianych aplikacjach właściwe zadanie rozwiązywał system rozmyty, **GA** był stosowany do zaprojektowania odpowiedniego **FLS**. Można sobie wyobrazić sytuację odwrotną: właściwe zadanie wykonuje **GA**, natomiast **FLS** jest stosowany do ustalania parametrów **GA**³. Lee i Takagi (Medsker, 1995) w swojej propozycji poszli jeszcze dalej, zastosowali mianowicie **FLS** do dostrajania parametrów **GA** wykonującego określone zadanie, natomiast do dostrajania samego **FLS** wykorzystali inny **GA**. Podejście to nie jest często opisywane, wydaje się dość interesujące i warto dalszych badań.

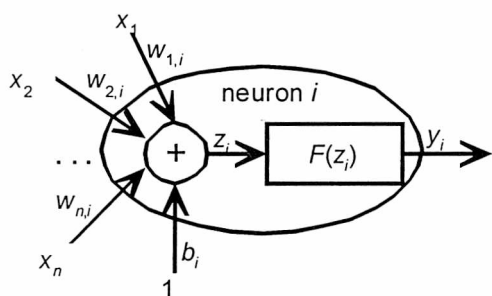
2.3. ZASTOSOWANIE GAs DO SIECI NEURONOWYCH

Jedną z technik naśladowujących naturę są **sztuczne sieci neuronowe** (NNs – *Artificial Neural Networks*). Inspiracją do ich budowy jest mózg żywych organizmów. Obie techniki – **GAs** i **NNs** – bywają w różny sposób łączone.

2.3.1. SIECI NEURONOWE

Sieć neuronowa składa się z pojedynczych elementów, zwanych neuronami. Są one ze sobą połączone, a każdemu połączeniu przypisana jest waga określająca moc tego połączenia. Każdy neuron sumuje przychodzące do niego sygnały pomnożone przez wagę połączeń. Ten sumaryczny sygnał jest następnie przetwarzany wewnątrz neuronu przez funkcję aktywacji. Tak powstały sygnał jest dalej przekazywany do innego neuronu lub – jeśli jest to neuron wyjściowy – jest sygnałem wyjściowym z sieci (rys. 2.7).

³ Jak wiadomo, algorytmy genetyczne są bardzo wrażliwe na wartości takich parametrów, jak częstość mutacji, rodzaj i częstość krzyżowania itp. Ustalenie odpowiednich dla danego zadania parametrów wymaga czasami dużej liczby prób.



Rys. 2.7. Schemat pojedynczego neuronu: x_j – j -te wejście neuronu, $w_{j,i}$ waga połączenia na j -tym wejściu i -tego neuronu, b_i – polaryzacja (ang. *bias*) i -tego neuronu, z_i – suma ważonych wejść i -tego neuronu, $F(z_i)$ – funkcja aktywacji, y_i – wartość na wyjściu i -tego neuronu

Neurony w sieci mogą być połączone w różny sposób. Często są one pogrupowane i tworzą warstwy. W przypadku takich sieci neurony jednej warstwy są połączone z neuronami kolejnej warstwy. Wyróżnia się warstwę neuronów wejściowych, do których podawana jest informacja wejściowa, i warstwę neuronów wyjściowych, z których informacja jest odczytywana jako odpowiedź sieci. Wewnątrz istnieje jedna lub kilka warstw ukrytych. Informacja przepływa od warstwy wejściowej, przez warstwy ukryte, do warstwy wyjściowej. Jest to sieć jednokierunkowa (*feedforward network*). In-

ny rodzaj architektury sieci to taki, w którym występują połączenia zwrotne między neuronami (sieć ze sprzężeniami zwrotnymi). Można też sobie wyobrazić całkiem inną architekturę sieci, w której wyróżnia się grupę neuronów wejściowych i grupę neuronów wyjściowych, sposób połączenia neuronów ukrytych natomiast jest dowolny.

Po to, by sieć realizowała stawiane jej zadanie, należy ją tego zadania wyuczyć. Można wyróżnić dwa tryby pracy sieci. Pierwszy – tryb uczenia i drugi – tryb odtwarzania. Uczenie sieci polega na takiej modyfikacji wag połączeń, aby sieć realizowała dane zadanie. Wyróżnia się uczenie nadzorowane (inaczej zwane z nauczycielem), kiedy znany jest zbiór par – wektor wejściowy i odpowiadający mu wektor wyjściowy, oraz uczenie nienadzorowane, kiedy znany jest jedynie zbiór wektorów wejściowych. Wówczas sieć, wykorzystując odpowiednie reguły uczenia, sama potrafi określić odpowiednie odwzorowanie.

Większość zastosowań korzysta z takich zdolności sieci, jak: kojarzenia (sieć spełnia rolę pamięci asocjacyjnej odwzorowującej wejście w wyjście), klasyfikacji, uogólniania.

W literaturze jest opisanych wiele algorytmów uczenia sieci i różnych architektur, np.: sieci z propagacją wsteczną, sieci Kohonena, sieci typu ART, BAM czy sieci hierarchiczne (Freeman i Skapura 1992, Ossowski 1994).

Problem projektowania sieci neuronowych

Proces rozwoju modelu sieci neuronowej dla danego zadania obejmuje zwykle cztery etapy (Miller i in. 1989):

1. Dobór problemu (np. rozpoznawanie obrazów, przetwarzanie języka lub inne).

2. Projektowanie architektury sieci do uczenia jej danego zadania, architektura ta jest szkieletem struktury sieci: określa się liczbę użytych neuronów, ich organizację w warstwy, połączenia między nimi.

3. Uczenie sieci wybranym algorytmem w zależności od tego, czy sieć ma być uczona z nauczycielem czy bez.

4. Ocena nauczonej sieci – sprawdzanie, na ile wykonuje ona swoje zadanie oraz czy szybkość uczenia i zdolność uogólniania są zadowalające.

Cały ten proces może być powtarzany, aż osiągnięte zostaną zadowalające rezultaty.

Mimo że istnieją sensowne metody dla trzech etapów, etap projektowania sieci pozostaje bardziej sztuką niż techniką (Miller i in. 1989). Badacz powinien polegać na własnym doświadczeniu w projektowaniu sieci i nieformalnych heurystykach. Istnieje kilka zasad projektowania (np.: im masz poważniejszy problem, tym więcej jednostek ukrytych potrzebujesz), ale brak jest powszechnie uznanej metody.

Sieci neuronowe są budowane jako analogia biologicznego mózgu. Mózg powstał w wyniku wieloletniej ewolucji biologicznej. Wydaje się naturalne podejście, w którym sztucznym algorytmom genetycznym pozwala się budować sztuczne sieci neuronowe, rozwiązujące określone zadania.

2.3.2. ŁĄCZENIE GAs I NNs

Przez długi czas algorytmy genetyczne i sieci neuronowe były rozwijane niezależnie od siebie. Prace opisujące łączenie GAs i NNs zaczęły pojawiać się w latach osiemdziesiątych. Wydaje się jednak, że potrzebne są dalsze badania w celu określenia zarówno sposobu łączenia obu technik, jak i zadań, w których takie połączenie jest opłacalne.

Algorytmy genetyczne mogą być stosowane do projektowania sieci neuronowych na kilka różnych sposobów, m.in.:

- optymalizacja wag w sieci o ustalonej strukturze (genetyczne lub ewolucyjne uczenie) (np. Montana i Davis 1989),
- optymalizacja parametrów algorytmu uczącego lub reguł uczących (np. Nadel i Stein 1993, *Brain Maker*),
- wybór (przygotowanie) ciągu uczącego (np. Nadel i Stein 1993, Schafer i in. 1992),
- optymalizacja struktury sieci neuronowych (np. Bornholdt i Graudenz 1992, Murray 1994):
 - optymalizacja liczby neuronów i(lub) połączeń, i(lub) liczby warstw,
 - projektowanie topologii sieci (struktury razem z wagami i(lub) funkcjami aktywacji).

W niniejszej pracy ograniczymy się do omówienia własnych doświadczeń w tym zakresie⁴.

⁴ Dokładniejsze omówienie tej problematyki znajdzie się w przygotowywanej do druku pracy *Naśladowanie natury w sztucznej inteligencji*.

Montana i Davis (1989) oraz Porto i in. (1995) przedstawiają w publikacjach eksperymenty wskazujące na przewagę uczenia genetycznego w stosunku do prostej metody wstecznej propagacji błędów.

W ramach zajęć dydaktycznych wykonaliśmy wiele prób genetycznego uczenia sieci neuronowych. Wiele z nich dotyczyło prostych zadań klasyfikacji, np. rozpoznawanie ręcznie pisanych cyfr lub liter. Eksperymenty nie dają podstaw do jednoznacznego stwierdzenia, że genetyczne uczenie daje lepsze efekty od tradycyjnej metody wstecznej propagacji błędów, zarówno jeśli wziąć pod uwagę jakość nauczonej sieci, jak i czas uczenia. Zaimplementowaliśmy uczenie sieci neuronowej metodą kombinowaną, genetyczne uczenie na przemian z metodą *Instar* Grossberga (Ossowski 1994). Sieci w ewoluującej populacji były uczone metodą *Instar*, wynik uczenia był traktowany jako przystosowanie sieci dla operatorów genetycznych. Taka kombinacja dała całkiem dobre rezultaty dla zadania rozpoznawania cyfr pisanych ręcznie na ekranie monitora, za pomocą myszki (rys. 2.8). Zarówno czas obliczeń, jak i efekty działania sieci uczonej metodą kombinowaną oraz metodą *Instar* są porównywalne (Kwaśnicka 1995).

W projektach METEO i WOCC⁵ testowaliśmy genetyczne uczenie sieci neuronowych dla zadania przewidywania pogody oraz do klasyfikacji typów charakterologicznych osób na podstawie cech ich pisma odręcznego oraz oddzielnie, na podstawie cech ich wyglądu. We wszystkich trzech przypadkach uczenie za pomocą wstecznej propagacji błędów dało wyraźnie lepsze rezultaty. Zwykle we wspomnianych projektach uczyliśmy sieć neuronową również zadania XOR. Co jest interesujące, prawie w każdym eksperymencie sieć była w stanie genetycznie nauczyć się tego zadania. Dla innych zadań, wyniki genetycznego uczenia nie były lepsze, a zwykle gorsze od uczenia klasycznymi metodami (wstecznej propagacji błędów i *Instar*).

Murray (1994) stosował **GA** do optymalizacji liczby ukrytych neuronów i współczynnika uczenia dla 3-warstwowej sieci uczonej metodą wstecznej propagacji błędów. Problem optymalizacji struktury jednokierunkowej sieci neuronowej (liczba neuronów, warstw, połączeń, parametry algorytmu uczącego) jest rozpatrywany w pracy Harpa i in. (1989). *NeuroForecaster* – program do budowy sieci neuronowych dla zadań przewidywania⁶ – ma wbudowany moduł *GENETICA Net Builder Option* do ewolucyjnego tworzenia i optymalizowania sieci neuronowych. Miller i inni (1989) pokazują, że **GA** jest obiecującą metodą automatyzacji projektowania sieci neuronowych. **GA**s są stosowane do optymalizacji struktury sieci neuronowych do rozwiązy-

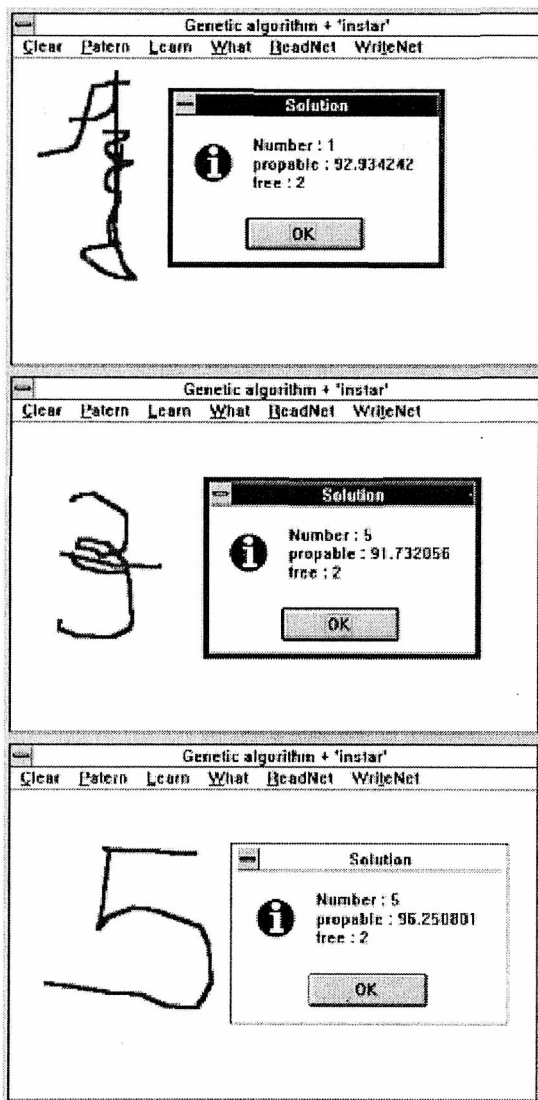
⁵ Kwaśnicka H., Micór A., *Neural Networks Supported a Short-Time Weather Forecasting* oraz Kwaśnicka H., Przygoda W., *WOCC – Hybrid Expert System for Characterological Type Determining*, obie prace opublikowane w materiałach konferencyjnych III Konferencji Inżynierii Wiedzy i Systemów Ekspertowych, Wrocław, czerwiec 10–13, 1997.

⁶ *NeuroForecaster – State of the Art Business Forecasting System, Information About the System, Futura*, Singapore, 1994.

wania trudnych zadań⁷, np. rozpoznawanie obrazu satelitarnego, składającego się z 36 milionów pikseli, prognozowanie giełdy przy danych pochodzących z 12 lat, obserwacje prowadzone raz dziennie, a czasami co jedną minutę, ustawiania anteny satelitarnej itp. Najnowsze oprogramowanie pracuje na maszynach równoległych, czas generowania optymalnej sieci neuronowej sięga tygodnia lub dłużej.

GA mogą być użyteczne do projektowania NNs w odmienny sposób. Bornholdt i Graundez (1992) zaproponowali projektowanie asymetrycznych sieci neuronowych za pomocą GA. Wyróżnione były neurony wejściowe, wyjściowe oraz pozostałe neurony, autorzy nazywają je korą (*cortex region*). Pomiędzy neuronami kory dopuszczalne są dowolne połączenia. Funkcja przystosowawcza uwzględnia liczbę poprawnych bitów na wyjściu i średnią liczbę cykli, po których sieć osiąga stan stabilny. Chromosom reprezentuje kompletną topologię sieci, razem z wagami. Wyniki eksperymentów dla funkcji logicznych (XOR, 5-bitowa funkcja parzystości) są bardzo obiecujące.

Dasgupta i McGregor zaimplementowali strukturalny GA do projektowania sieci neuronowych. Chromosom jest łańcuchem bitów, składającym się z dwóch poziomów: poziom wyższy reprezentuje połączenia pomiędzy neuronami, poziom niższy – wagi tych połączeń. Funkcja przystosowania uwzględnia



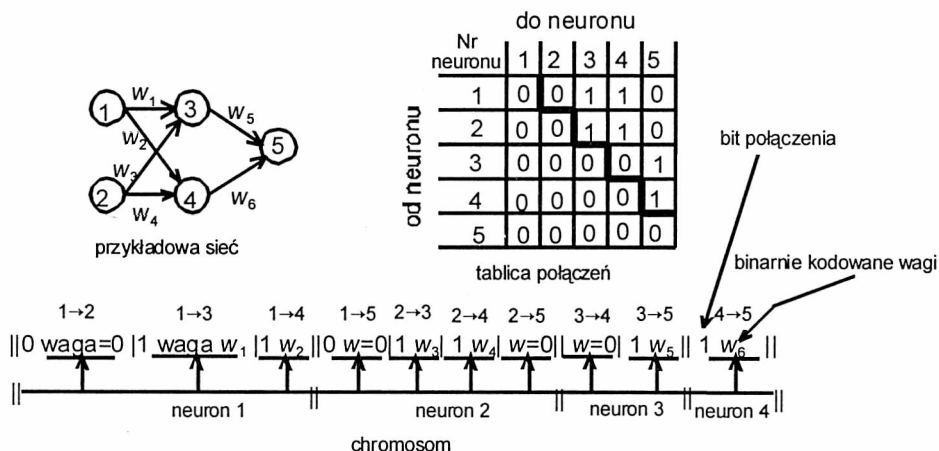
Rys. 2.8. Przykład rozpoznawania cyfr przez NN uczoną metodą kombinowaną *Instar* i uczenie genetyczne

⁷ *Genetic Search in NNs Optimizations*, wykład profesora J. Korczaka ze Strassburga, Francja, na konferencji „Sieci neuronowe i ich zastosowania”, Kule k.Częstochowy, 14–18.10.1997. Część oprogramowania jest dostępna pod adresem <http://dpt-info.u-strasbg.fr/lsiit/orii/gria>.

sumę średniokwadratowego błędu, poprawność i złożoność struktury sieci. Projektowanie większych sieci stwarzało systemowi problemy (Dasgupta i McGregor 1992).

Wyniki naszych prac dotyczących projektowania sieci neuronowych za pomocą algorytmów genetycznych są zaprezentowane m.in. w kilku publikacjach (Kilińska i Kwaśnicka 1995, Kwaśnicka 1995, Kwaśnicka i Szerszon 1997, Cińcio i in. 1999).

W pierwszym podejściu do projektowania sieci neuronowych za pomocą GA chodziło o zaprojektowanie sieci, która nauczy się grać w grę podobną do gry w kółko i krzyżyk (*tic-tac-toe*) (Rusoń 1995). Sieć była kodowana za pomocą binarnego łańcucha, stanowiącego chromosom. Otrzymane rezultaty nie były zadowalające, uczenie sieci metodą wstecznej propagacji błędu dawało lepsze rezultaty. Do kodowania sieci stosowano tablicę połączeń z dodanymi binarnie kodowanymi wagami (rys. 2.9). Być może zastosowanie chromosomu o zmiennej długości dałoby lepsze rezultaty.



Rys. 2.9. Reprezentacja sieci dla zadania gry w kółko i krzyżyk

Drugie podejście dotyczyło jednokierunkowych sieci neuronowych, gdzie sieć neuronowa była reprezentowana jako formuła gramatyczna (Lieske 1995). Reprezentacja ta wydaje się odpowiednia, ponieważ GA produkuje tylko poprawne sieci neuronowe. Aby zdefiniować gramatykę, należy zdefiniować cztery wielkości:

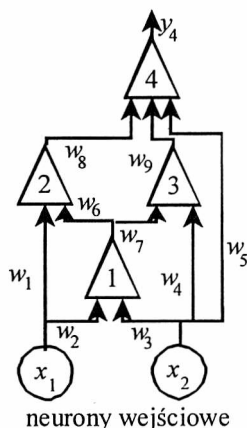
- Zbiór pierwotnych symboli (*primitives*) V_T (mogą nimi być: wagi w_i , zbiór sygnałów wejściowych x_i , zbiór sygnałów wyjściowych o_i oraz zbiór funkcji aktywacji T_i).
- Zbiór zmiennych (wielkości pośrednich) (*non-terminal symbols*) V_N (np. *ważone_wejście*, *suma_ważonych_wejść*, *potencjał_neuronu*, *sieć*).
- Zbiór produkcji P (*productions*) są to reguły produkcji, które pozwalają na zastępowanie zmiennych przez symbole pierwotne.
- Korzeń – symbol zarodkowy (*root symbol*), musi to być symbol należący do V_N .

Przykładowa formuła gramatyczna opisująca sieć pokazaną na rysunku 2.10 (po zastąpieniu, zgodnie z regułami produkcji zmiennych y_1, y_2 i y_3) może mieć postać ($N \in V_N$) i oznacza zmienną sieć:

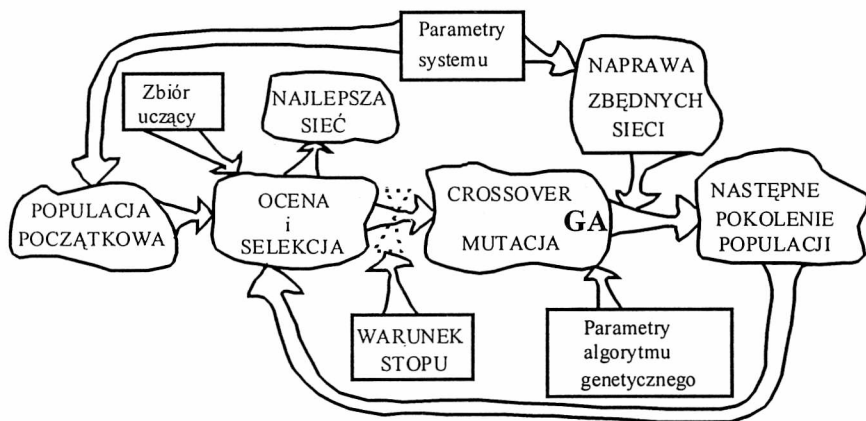
$$\begin{aligned} N &\leftarrow T_4(w_8 y_2 + w_9 y_3 + w_5 x_2) = \\ &= T_4(w_8 T_2(w_1 x_1 + w_6 T_1(w_2 x_1 + w_3 x_2)) + \\ &\quad + w_9 T_3(w_4 x_2 + w_7 T_1(w_2 x_2 + w_3 x_3)) + w_5 x_2). \end{aligned}$$

Taka formuła może być reprezentowana jako chromosom i poddana działaniu algorytmu genetycznego. Do wstępnych eksperymentów zastosowano binarne kodowanie struktury sieci, sieć była uczona metodą wstecznej propagacji błędów i dla uproszczenia kodowania założono maksymalną liczbę warstw. Zaproponowane podejście wymaga dużo pamięci, również szybkość działania nie jest zadowalająca. Podejście to wydaje się warte dalszych prac, głównie nad znalezieniem odpowiedniego sposobu kodowania formuł.

Trzecie podejście, kontynuowane obecnie, to modyfikacja wykonanego wcześniej systemu nazwanego *generator sieci neuronowych NNG* (Kilińska i Kwaśnicka 1995). Ogólny schemat tego systemu pokazano na rysunku 2.11.



Rys. 2.10. Przykładowa sieć neuronowa (jednokierunkowa)



Rys. 2.11. Projektowanie sieci neuronowych za pomocą algorytmu genetycznego

Sieć neuronowa jest reprezentowana jako lista neuronów, każdy neuron ma własną listę zawierającą informacje o wszystkich połączeniach dochodzących do niego. W systemie są dopuszczone wszystkie typy połączeń, co oznacza, że nie ma wymagań odnośnie do struktury sieci. Istnieje jednak możliwość nałożenia ograniczeń na strukturę sieci, np. takich, że sieć ma być jednokierunkowa. Funkcje aktywacji mogą być

wybrane z predefiniowanego zbioru funkcji. Przyjęto, że funkcja przystosowania będzie uwzględniać średni błąd oraz maksymalny błąd na pojedynczym wyjściu sieci. Jest nią średnia arytmetyczna wartości a_k , liczonych dla każdego wzorca w ciągu uczącym według wzoru

$$a_k = \frac{1}{1 + \sum_{i=1}^N |Y_i - y_i| + \max_N |Y_i - y_i|},$$

w którym:

Y_i – pożądane wyjście dla k -tego wzorca,

y_i – wyjście produkowane przez ocenianą sieć,

N – liczba wyjść sieci (wyjściowych neuronów).

GA stosuje operatory mutacji (umożliwiające zmianę wag, dodawanie i usuwanie neuronów i połączeń) oraz krzyżowania. System *NNG* testowano, projektując sieci neuronowe dla zadań XOR i kodowania 4×3 . Dla XOR system znalazł optymalne rozwiązanie już po 50 generacjach (sieć składała się z 4 neuronów), dla problemu kodowania system znalazł dobre rozwiązanie po 300 generacjach (sieć składała się z trzech perceptronów). System nie produkuje dużych sieci, ponieważ dla dwóch sieci o jednakowym przystosowaniu w czasie selekcji (elitarna metoda selekcji) wybierana jest ta sieć, która ma mniejszą liczbę neuronów. Preferowane są więc sieci mniejsze, ale tylko przy nie pogorszeniu działania sieci.

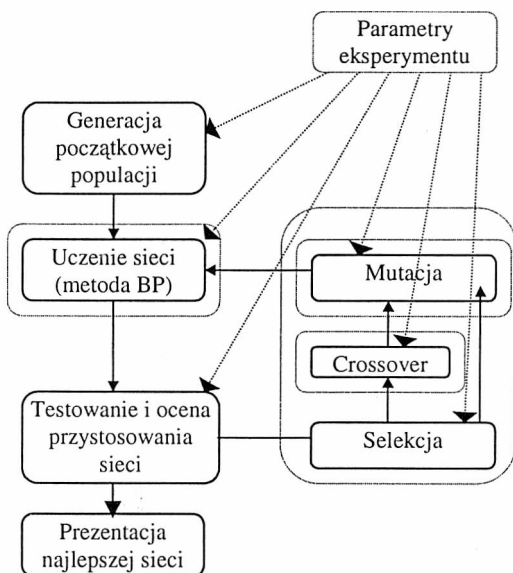
Działanie systemu to przede wszystkim generowanie sieci dobrze nauczonych. Jeśli populacji stosunkowo dobrze nauczonych sieci pozwolimy dalej ewoluować, to system będzie produkował sieci o mniejszym rozmiarze bez pogorszenia poprawności ich działania.

Po wykonaniu wielu eksperymentów można stwierdzić, że jest to droga bardzo obiecująca, ale wymaga jeszcze wielu badań. Na przykład system stosunkowo szybko znajdował niezłą sieć, miał natomiast problemy z jej dostrojeniem – można oczekiwać, że połączenie zaproponowanej metody projektowania sieci neuronowych z klasycznymi metodami uczenia może umożliwić uzyskiwanie odpowiednich sieci. Podobnie, dalszych prób wymaga określenie najbardziej właściwej funkcji przystosowania. Teoretyczne analizy niewiele mogą tu powiedzieć, dalsze badania wymagają bardziej elastycznego narzędzia i wielu prób. Rozpoczęto więc prace nad kolejnym projektem, w którym wykorzystane zostaną dotychczasowe wyniki (Kwaśnicka 1997). Celem projektu jest opracowanie komputerowego systemu projektowania sieci neuronowych z zastosowaniem algorytmów genetycznych, który będzie spełniał dwa główne warunki:

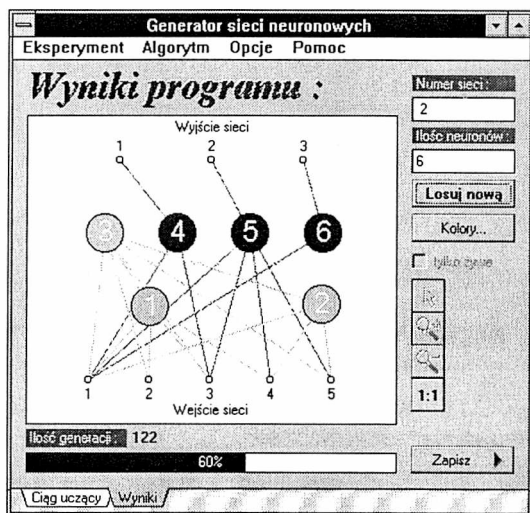
- umożliwi wszechstronne badania różnych podejść do projektowania sieci neuronowych z zastosowaniem (w różnym zakresie) algorytmów genetycznych,
- będzie spełniał wymagania dobrego programu dydaktycznego z dziedziny **GA** i **NN**, co oznacza, że będzie bardzo łatwy w obsłudze, będzie miał rozbudowany interfejs graficzny do ustalania parametrów systemu, jak i prezentowania wyników.

Zaimplementowano część systemu NETGEN umożliwiającą projektowanie jednokierunkowych sieci neuronowych (Szerszon 1997, Kwaśnicka i Szerszon 1997). Schemat systemu pokazano na rysunku 2.12.

W aktualnej wersji program umożliwia łączenie ewolucyjnego projektowania struktury z ingerencją użytkownika w strukturę, uczenie sieci metodą wstecznej propagacji błędów wraz z uczeniem genetycznym. W odróżnieniu od znanych nam prac, system pozwala na istnienie w sieci tzw. „martwych” neuronów. „Martwe” neurony to takie, które nie mają połączenia wyjściowego (ich sygnał wyjściowy nie jest doprowadzony do żadnego innego neuronu (np. neurony o numerach 1, 2 i 3 na rysunku 2.13). Istnienie takich neuronów nie przeszkadza sieci przetwarzać dochodzących do niej sygnałów, natomiast w toku dalszej ewolucji, wskutek działania operatorów genetycznych może powstać połączenie od „martwego” dotychczas neuronu, co może przyczynić się do lepszego działania sieci. Dopuszczając istnienie martwych neuronów w sieci, w mniejszym stopniu ingerujemy w ewolucyjny proces tworzenia struktury sieci. System pozwala użytkownikowi na modyfikację struktury sieci poprzez np. dodanie(usunięcie) neuronu, połączenia, zmianę wagi. Jest bardzo przyjazny, sam dba o to, aby wszystkie potrzebne parametry były ustalone (przez ich wybór lub wpisanie w odpowiednim polu okienka).



Rys. 2.12. Schemat systemu NETGEN



Rys. 2.13. Wizualizacja sieci neuronowej w systemie NETGEN (z możliwością jej modyfikacji)

Dotychczas zaimplementowano i przetestowano dwie funkcje przystosowawcze. Do oceny przystosowania lepsza okazała się następująca funkcja

$$f = \left(\frac{E}{N} \right)^R \left(\frac{L}{L_{\max}} \right)^S,$$

w której:

E – średniokwadratowy błąd odpowiedzi sieci,

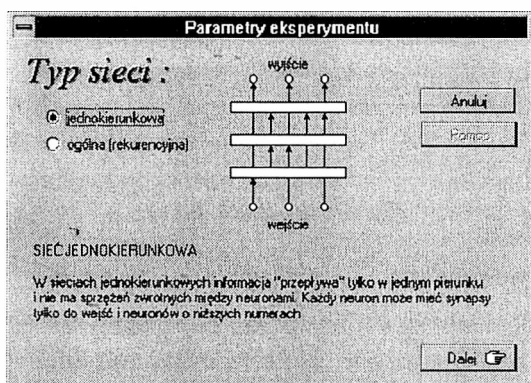
N – liczba wzorców,

L – liczba neuronów w sieci,

$L_{\max}$ – założona maksymalna liczba sieci w populacji,

R – parametr definiujący szybkość uczenia ($R \leq 1$),

S – parametr definiujący siłę, z jaką rozmiar sieci wpływa na jej wartość przystosowawczą ($S > 1$: twarda selekcja, $0 < S < 1$: miękka selekcja).



Rys. 2.14. Przykładowe okno do zmiany parametrów systemu NETGEN



Rys. 2.15. Ustalanie parametrów operatorów genetycznych w systemie NETGEN

Ponieważ zaimplementowany GA maksymalizuje wartość przystosowawczą, za funkcję fitness przyjęto

$$\text{fitness} = \frac{1}{f + 10^{20}}.$$

Funkcja f ma czynnik odpowiedzialny za jakość działania sieci i drugi czynnik odpowiedzialny za rozmiar sieci. Jak widać, w testowanej funkcji przystosowawczej spróbowano uwzględnić bezpośrednio zarówno błąd, jak i rozmiar sieci. Taka funkcja ma jednak tendencje do preferowania albo małych sieci ze stosunkowo dużym błędem na wyjściu, albo dobrze nauczonych, ale stosunkowo dużych sieci (Szczeszon 1997). Zastąpienie iloczynu sumą w funkcji f , jak wynika zarówno z analizy funkcji, jak i eksperymentów, poprawia działanie systemu.

Doświadczenia wskazują dalsze kierunki pracy, np. dynamiczne, inteligentne sterowanie parametrami GA. Wcześniej należy znaleźć odpowiednią funkcję lub ich zbiór, z których będzie korzystał system. Zadanie jest o tyle trudne, że w istocie mamy do czynienia z zadaniem wielokryterialnym: istotne jest

nauczenie sieci, zdolność jej uogólniania i możliwie mały rozmiar sieci.

3. K-MODEL

– ROZSZERZONY MODEL EWOLUCYJNY

Algorytmy genetyczne, mimo iż powstały z inspiracji ewolucji biologicznej, korzystają ze słownictwa zapożyczonego z biologii i genetyki, są jedynie ubogim modelem natury. Nie przeszkadza to być im bardzo użytecznymi w wielu dziedzinach, zwłaszcza w zadaniach optymalizacyjnych i automatycznego uczenia się. W wielu trudnych, praktycznych zadaniach skuteczność algorytmów genetycznych nie jest jeszcze zadowalająca. Przedstawiony tu rozszerzony model ewolucyjny, nazwany **K-modelem**, zawiera pewne mechanizmy, które pozwalają uzyskać większą skuteczność w trudnych zastosowaniach. Mechanizmy te są wzorowane na ewolucji biologicznej.

3.1. BIOLOGICZNE INSPIRACJE K-MODELU

Tempo ewolucji biologicznej nadal pozostaje tematem gorących dyskusji. Wiadomo również, że genotyp organizmu nie koduje jego cech fenotypowych w sposób jeden gen – jeden fen. Samo pojęcie genu też budzi jeszcze dyskusje. Mając na uwadze zarówno dyskusje ewolucjonistów, jak i zwiększenie skuteczności algorytmów genetycznych, zaproponowano rozszerzony algorytm ewolucyjny.

Makromutacje

W długiej perspektywie ewolucja gatunków przebiega z różnym tempem i w różnych kierunkach (Fogel 1992, Hoffman 1983), co jest zgodne z paradygmatem neodarwinizmu. Jak ujmuje to R. Dawkins (1995, s. 125): „Małymi kroczkami do celu – taka się wydaje naczelna zasada ewolucji. Potęga bowiem ewolucji tkwi w stopniowości zmian. ... Mogły istnieć momenty nagłych, skokowych zmian, jednorazowych *makromutacji*, które w sposób zasadniczy odróżniały dziecko od obojga rodziców” . Makromutacje są postrzegane jako proces dający radykalny efekt fenotypowy (Mayr 1963). Obecnie biolodzy są zgodni, że makromutacje występują w naturze.

Geny nadmiarowe (redundancyjne)

Obecnie jest rzeczą powszechnie znaną, że jedynie niewielka część genotypu organizmów koduje uwidocznione cechy organizmów (cechy fenotypowe) (Gould 1991,

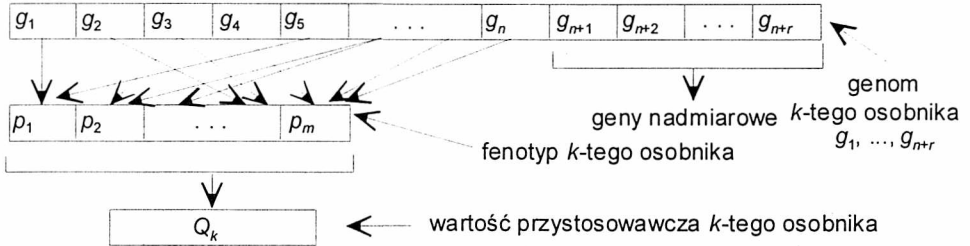
s. 172). Geny kodujące cechy fenotypowe będziemy nazywać **genami aktywnymi** lub **genami fenotypowymi**. Pozostałą część genotypu będziemy nazywać **genami nadmiarowymi** lub **genami ukrytymi**. Ewolucja wytwarza nie tylko perfekcyjne zestawy genów, znaczna część genotypów to geny nie mające wpływu na funkcjonowanie organizmu (Szarski 1986). Wśród nich mogą istnieć nawet geny śmiertelne, ale mogą one się „wieźć” przez pokolenia dzięki innym, pozytywnym, aktywnym genom, powodującym ewolucyjną preferencję danego organizmu. Geny nadmiarowe podlegają zmianom genetycznym podczas działania takich operatorów, jak mutacja czy krzyżowanie. Ze względu na rolę tych genów, efekty ich zmian nie są uwidocznione w organizmie, nie mają więc wpływu na jego funkcjonowanie, ale są kumulowane w organizmach. Wiadomo również, że podział genotypu na część ukrytą i aktywną nie jest stały w czasie trwania ewolucji. Zdarza się, że w pewnych okolicznościach zachodzi reorganizacja chromosomu, wskutek której gen dotychczas ukryty (nadmiarowy) uaktywnia się, zaczyna wpływać na fenotyp organizmu, a zatem na jego przystosowanie do środowiska (Gould 1991, Beasley i in. 1994a, Kwaśnicka 1997b). Obserwowana zmiana fenotypowa u takich organizmów może być bardzo duża, można ją więc traktować jako wynik makromutacji. Taka duża reorganizacja chromosomu jest nazywana przez Goldschmidta „utrata spójności genotypu” (*loosing the cohesion of genotype*) (Mayr 1982). Oczywiście, można się spodziewać, że większość powstałych organizmów będzie eliminowana z populacji, ale mogą powstawać wtedy osobniki nazywane „potworami z przyszłością” (*hopeful monsters*) – będą to osobniki, które przeżyją i mogą umożliwić populacji osiągnięcie nowej niszy ekologicznej (Goldschmidt 1940).

Pleiotropowość i poligeniczność organizmów

Algorytmy genetyczne operują na populacji osobników stanowiących potencjalne rozwiązania rozpatrywanego problemu. Najczęściej osobnik reprezentowany jest jako łańcuch symboli z zadanego alfabetu, zwykle stosowany jest alfabet binarny. W GA stosuje się proste odwzorowanie: *jeden gen – jedna cecha fenotypowa*. Oznacza to, że określony ciąg bitów (gen) koduje wprost jeden parametr optymalizowanej funkcji. Sposób kodowania cech fenotypowych w biologii jest zupełnie inny, proste odwzorowanie *jeden gen – jedna cecha* nie występuje u żywych organizmów. Pojedynczy gen może wpływać jednocześnie na wiele cech osobnika. Zjawisko to jest nazywane **efektem pleiotropowym**. Z drugiej strony, każda cecha fenotypowa może być determinowana przez pewną liczbę genów – to zjawisko nazywane jest **efektem poligenicznym**. Model stosowany w klasycznym algorytmie genetycznym wydaje się pod tym względem zbyt uproszczony. Trudno jest znaleźć aplikacje, w których te dwa efekty są brane pod uwagę (np. Altenberg 1994). Porównanie trzech modeli: klasycznego GA, GA z niewielkim efektem pleiotropowym i poligenicznym oraz GA działającego bezpośrednio na parametrach optymalizowanej funkcji (bez kodowania) można znaleźć w publikacjach autorki (Kwaśnicka 1997b i 1998a). Przedstawione tam wyniki symulacji wskazują, że charakter i tempo ewolucji (mierzone liczbą pokoleń niezbędnych do opuszczenia lokalnego szczytu, stanowiącego tzw. pułapkę ewolucyjną, i osiągnię-

cia globalnego szczytu) są różne w poszczególnych modelach, przy czym ewolucja jest skuteczniejsza w modelu z plejotropowością i poligenicznością.

Schemat reprezentacji osobnika przy uwzględnieniu plejotropowości, poligeniczności i genów nadmiarowych pokazano na rysunku 3.1.



Rys. 3.1. Reprezentacja osobnika z plejotropowością i poligenicznością oraz z genami nadmiarowymi

Alfabet genotypu

Watson i Crick odkryli, że geny są cyfrowymi łańcuchami informacji, ale nie jest to łańcuch binarny. W biologii geny mają różną długość i są łańcuchami czterech symboli (nukleotydów): A, G, C, T. Dawkins (1995, s. 29, 37) pisze: „Kod genetyczny można wyobrazić sobie jako słownik, w którym 64 słowa w jednym języku (64 możliwe kombinacje trójkowe – tryplety – z czteroliterowego alfabetu) są przekładane na 21 słów w innym języku (20 aminokwasów plus znak kropki). Prawdopodobieństwo wystąpienia dwukrotnie takiego samego przełożenia, 64 na 21, jest mniejsze niż jeden do miliona miliona miliona miliona milionów. ... Kod genetyczny nie jest dwójkowy, jak w komputerach, ani ósemkowy jak w niektórych systemach telefonicznych, tylko czwórkowy, operuje bowiem czterema symbolami”.

Przyjęte w klasycznym **GA** binarne kodowanie trudno jest uzasadnić przesłankami biologicznymi (Back 1996). Twierdzenie o schematach mówi, że liczba krótkich i niskiego rzędu schematów o średnim przystosowaniu większym od średniego przystosowania w populacji, tzw. *bloków budujących* zwiększa się eksponencjalnie w trakcie ewolucji. Dla prostego **GA** z mutacją i jednopunktowym krzyżowaniem oraz dużych populacji można wykazać, że alfabet składający się z możliwie małej liczby symboli, tzn. binarny, jest optymalny (Goldberg 1989). Ostatnio coraz częściej teza ta jest podważana i pojawia się wiele aplikacji, w których nie stosuje się kodowania binarnego, lecz inne, odpowiedniejsze dla danego problemu (Goldberg 1991, Beasley i in. 1994, Michalewicz 1992, Kwaśnicka 1997d).

W niektórych zadaniach, np. projektowanie sieci neuronowych, binarne kodowanie całej struktury sieci neuronowej może stwarzać trudności. Z drugiej strony, stosowanie większego alfabetu do kodowania potencjalnych rozwiązań przy jednoczesnym stosowaniu jedynie podstawowych operatorów genetycznych (jak w klasycznym **GA**) może powodować utratę efektywności. Algorytm genetyczny silnie zależy zarówno od

używanej reprezentacji, jak i od stosowanych operatorów, dlatego też należy zwracać uwagę na odpowiedni ich dobór.

Nie ma podstaw sądzić, że natura nie działa w sposób efektywny. Nasuwają się zatem pytania: dlaczego nie rozważać genów nadmiarowych, plejotropowości i poligeniczności w algorytmach ewolucyjnych? Jakie korzyści może dać uwzględnienie tych mechanizmów w modelu?

Aby sprawdzić, czy taki rozszerzony model może działać inaczej (efektywniej) niż klasyczny algorytm genetyczny wykonano specjalnie zaprojektowane eksperymenty.

3.2. ROLA PLEJOTROPOWOŚCI I POLIGENICZNOŚCI

Dla sprawdzenia ewentualnego działania plejotropowości i poligeniczności opracowano program komputerowy GENPOP, w którym zaimplementowano trzy wersje algorytmu ewolucyjnego:

- **Wersja 1** jest klasycznym algorytmem genetycznym, stosującym binarne kodowanie, mutacje i crossover.

- **Wersja 2** to algorytm oparty na K -modelu. Chromosom jest tu łańcuchem liczb rzeczywistych, włączony jest jedynie minimalny efekt plejotropowy i poligeniczny. Nie ma żadnego modelu makromutacji.

- **Wersja 3** to działanie operatorów genetycznych (krzyżowanie i mutacja) na chromosomie składającym się z liczb rzeczywistych, stanowiących bezpośrednio zestaw argumentów optymalizowanej funkcji. Jest to model bez poziomu genotypowego; z takiego podejścia często się korzysta przy stosowaniu GA do uczenia lub projektowania sieci neuronowych (Montana 1989, Kwaśnicka 1997).

Do badań wybrano funkcję, która jest łatwo modyfikowalna w sensie wymiaru funkcji, oraz liczby, położenia i wysokości szczytów¹. Ogólny wzór tej funkcji to

$$Q(x_1, x_2, \dots, x_m) = \sum_{i=1}^{l_s} h_i e^{-\sum_{j=1}^m n_j (x_j - x_j^i)^2}, \quad (3.1)$$

gdzie:

x_i – i -ta współrzędna funkcji,

x_j^i – wartość j -tej współrzędnej dla i -tego szczytu,

m – wymiar funkcji,

l_s – liczba szczytów funkcji,

h_i – wysokość i -tego szczytu,

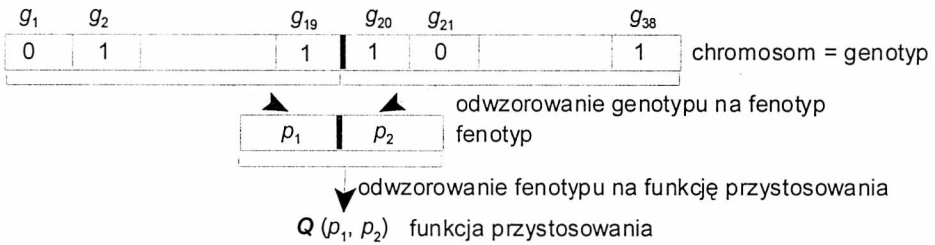
n_i – nachylenie i -tego szczytu.

¹ Kształt i profil tej funkcji pokazano w p. 4.1.

W dalszej części zaprezentowano wyniki uzyskane dla funkcji dwuwymiarowej (możliwość wizualizacji ewolucji) oraz z dwoma szczytami (możliwość obserwacji opuszczenia przez populację lokalnego szczytu przystosowawczego, czyli ucieczki z pułapki ewolucyjnej). W eksperymentach przyjęto, że funkcja ma dwa szczyty, w punktach o współrzędnych (5, 5) i (20, 20). Za wystarczający zakres zmienności x_1 i x_2 można uznać przedział $[-2, 30]$. Nachylenia szczytów wynoszą $n_1 = n_2 = 0,02$, wysokości szczytów: $h_1 = 1, h_2 = 1,5$.

3.2.1. MODEL WERSJA_1: KLASYCZNY ALGORYTM GENETYCZNY

Każdy osobnik w populacji jest reprezentowany jako genotyp składający się z 38 bitów (po 19 na x_1 i x_2). Na poziomie fenotypowym osobnik składa się z dwóch fenów $[p_1, p_2]$, gdzie pierwszy fen (p_1) jest współrzędną x_1 funkcji Q , a drugi fen (p_2) jest współrzędną x_2 (rys. 3.2).



Rys. 3.2. Reprezentacja osobnika w modelu Wersja_1

Wartość fenów jest liczona następująco:

$$p_1 = -2 + (g_1, g_2, \dots, g_{19})_{10} \frac{30 - (-2)}{2^{19} - 1}, \quad (3.2)$$

$$p_2 = -2 + (g_{20}, g_{21}, \dots, g_{38})_{10} \frac{30 - (-2)}{2^{19} - 1},$$

gdzie:

$(g_1, g_2, \dots, g_{19})_{10}$ – wartość dziesiętna liczby dwójkowej $g_1 g_2 \dots g_{19}$,

$(g_{20}, g_{21}, \dots, g_{38})_{10}$ – wartość dziesiętna liczby dwójkowej $g_{20} g_{21} \dots g_{38}$.

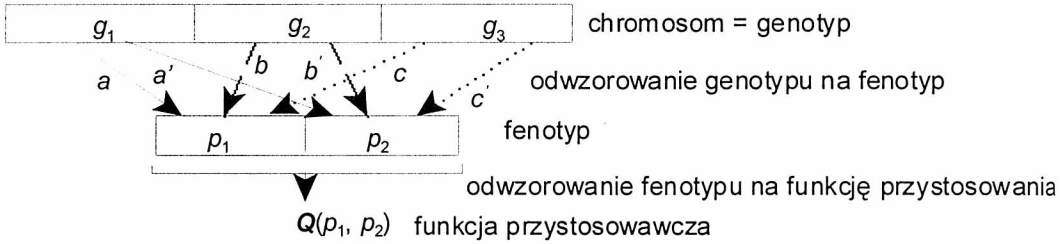
Zaimplementowane operatory genetyczne to:

- jednopunktowe krzyżowanie z zadaniem prawdopodobieństwem P_c oraz
- mutacja z zadaniem prawdopodobieństwem P_m zamiany bitu na wartość odwrotną.

Selekcja osobników może odbywać się jedną z trzech metod: metodą ruletki, metodą stochastyczną według reszt bez powtórzeń lub metodą turniejową (dwuelementową).

3.2.2. MODEL WERSJA_2: GA Z MINIMALNĄ PLEJOTROPOWOŚCIĄ I POLIGENICZNOŚCIĄ

Główna różnica pomiędzy tą wersją a poprzednią polega na sposobie reprezentacji osobnika. Genotyp osobnika składa się z trzech genów (dla dwuwymiarowej funkcji przystosowania) będących liczbami rzeczywistymi z przedziału od 0 do 1 (rys. 3.3): $g_1, g_2, g_3, g_i \in [0,1]$.



Rys. 3.3. Reprezentacja osobnika w modelu Wersja_2

Niewielki efekt pleiotropowy i poligeniczny jest wprowadzony przez przyjęcie liniowej zależności fenów od genów:

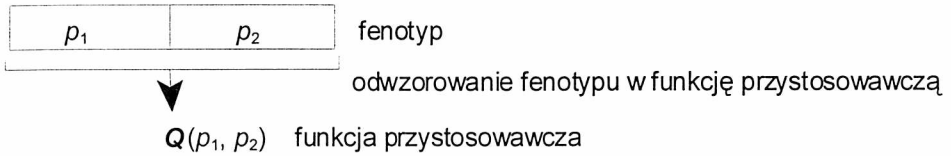
$$\begin{aligned} p_1 &= a g_1 + b g_2 + c g_3 + d, \\ p_2 &= a' g_1 + b' g_2 + c' g_3 + d'. \end{aligned} \quad (3.3)$$

Przyjęto, że na fen p_1 z jednakową siłą wpływają geny g_1 i g_3 , a na fen p_2 – z jednakową siłą geny g_2 i g_3 ($b = a' = 0$).

W czasie rekombinacji, z prawdopodobieństwem P_c następuje wymiana losowego genu pomiędzy dwoma osobnikami, jest to odpowiednik krzyżowania w modelu Wersja_1, przy czym punkt przecięcia chromosomu może wystąpić tylko pomiędzy genami. Mutacja genu polega na losowej zmianie (z prawdopodobieństwem P_m) wartości genu o losową wartość z zadanego przedziału ($g_i' = g_i \pm \Delta_i$, gdzie Δ_i jest losową wartością z przedziału $[0,0001; 0,1]$, g_i' jest wartością zmutowanego i -tego genu).

3.2.3. EWOLUCJA BEZ POZIOMU GENOTYPOWEGO (WERSJA_3)

W tym modelu osobnikiem w populacji jest wektor zmiennych – argumentów optymalizowanej funkcji. Operatory genetyczne są dostosowane do działania na wektorze liczb rzeczywistych i działają bezpośrednio na współrzędnych funkcji (rys. 3.4).



Rys. 3.4. Reprezentacja osobnika w modelu Wersja_3

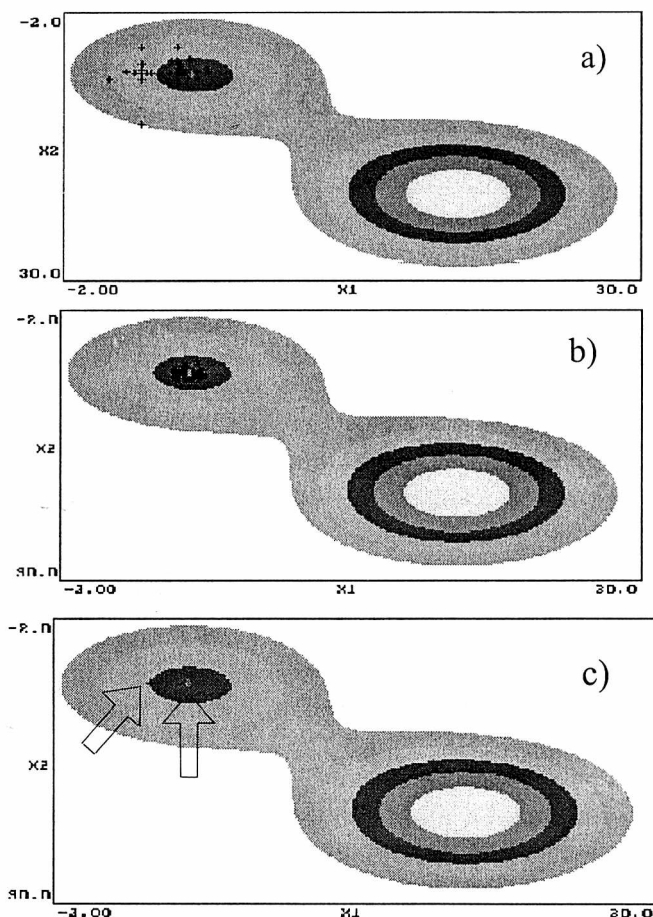
Rekombinacja i mutacja działają identycznie jak w modelu Wersja_2, lecz operują bezpośrednio na wektorze fenów. Zakres zmienności mutacyjnej fenów jest parametrem modelu, jest ustalony tak, by przejście na sąsiedni szczyt przystosowawczy wymagało kilku zmian mutacyjnych tej samej współrzędnej i to w tym samym kierunku.

3.2.4. BADANIA SYMULACYJNE

Zaproponowane wersje modelu ewolucyjnego pozwalają na symulacyjne badanie charakteru ewolucji z uwzględnieniem niewielkiego efektu pleiotropowego i poligenicznego. Jako miarę szybkości ewolucji przyjęto liczbę pokoleń, po której populacja „zasiedla” lepszą niszę ekologiczną, tzn. populacja będzie znajdować się na globalnym optimum. Inaczej mówiąc, interesuje nas zdolność populacji do ucieczki z pułapki ewolucyjnej. Zaprezentowane tu wyniki uzyskano dla podanej wyżej funkcji przystosowania i rozmiaru populacji 100 osobników. Populacja początkowa była umieszczana w lokalnym optimum (punkt o współrzędnych (5, 5)). Czas ewolucji zazwyczaj ustalony był na 100 pokoleń, choć czasami bywał zwiększany, by sprawdzić przewidywania. Podstawowe wartości prawdopodobieństw mutacji i krzyżowania zostały ustalone eksperymentalnie, wynoszą odpowiednio $P_m = 0,3$, $P_c = 0,25$. Symulację powtarzano dla każdego warunków przynajmniej 10 razy. We wszystkich eksperymentach chodziło o porównanie tempa i charakteru ewolucji w trzech wersjach modelu. Dla każdego modelu testowano trzy metody selekcji: stosunkowo miękką selekcję metodą ruletki, bardziej rygorystyczną metodę stochastyczną według reszt bez powtórzeń i stosunkowo twardą, turniejową metodę selekcji.

Symulacja modelu Wersja_1 (geny kodowane binarnie)

Ewolucja w tym modelu nie przebiega zbyt dobrze. Przykładowe wyniki symulacji uzyskane dla typowych parametrów algorytmu i trzech metod selekcji pokazano na rysunku 3.5. Najlepsze rezultaty są uzyskiwane przy stosowaniu selekcji osobników do reprodukcji metodą ruletki. Prawie w każdym przebiegu pojedyncze osobniki znajdowały się w pobliżu wyższego szczytu przystosowawczego, ale cała populacja nie przesuwała się na wyższy szczyt. Po obserwacji 20 prób można powiedzieć, że w około połowie przebiegów populacja była w stanie osiągnąć wyższy szczyt, choć potrzebowała na to więcej niż 100 pokoleń. W pozostałych metodach selekcji populacja po około 100 pokoleniach stawała się niemalże homogeniczna, co uniemożliwiało dalszą ewolucję (zwłaszcza w przypadku metody turniejowej). Dla tych metod selekcji nie zdarzyło się, by choć pojedynczy osobnik pojawił się na wyższym szczycie przystosowawczym.



Rys. 3.5. Ewolucja z binarnymi genami (Wersja_1): (a) metoda ruletki, (b) metoda stochastyczna, (c) metoda turniejowa (strzałki pokazują położenie populacji)

Symulacja modelu Wersja_2 (geny kodowane za pomocą liczb rzeczywistych, niewielki efekt plejotropowy i poligeniczny)

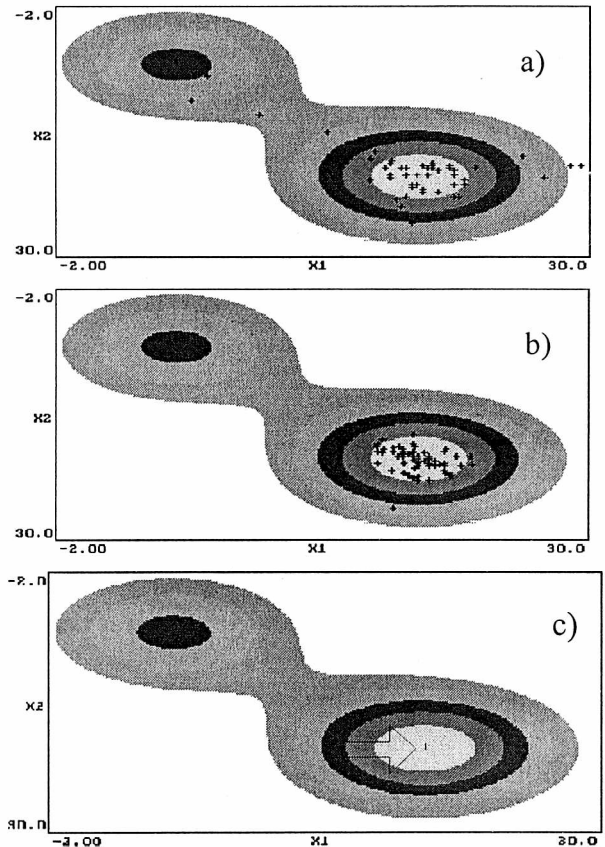
W tej wersji modelu w każdym przebiegu symulacji osobniki przechodziły na wyższy szczyt przystosowawczy w ciągu niespełna 100 pokoleń. W każdym eksperymencie cała populacja stosunkowo szybko zasiedlała wyższy szczyt. Można powiedzieć, że metoda selekcji oraz wartości prawdopodobieństw mutacji i krzyżowania miały niewielki wpływ na efektywność poszukiwań.

Przykładowe rezultaty dla trzech metod selekcji, uzyskane w czasie symulacji, z zastosowaniem tej wersji modelu pokazano na rysunku 3.6. Należy zaznaczyć, że w tej wersji modelu wyraźnie uwidacznia się wpływ metody selekcji na zbieżność populacji.

Im bardziej *miękką metodę selekcji*² (Galar 1994) zastosujemy, tym zróżnicowanie osobników w populacji jest większe, co z kolei powoduje niższą średnią wartość przystosowania osobników w populacji. *Twarda metoda selekcji* (Galar 1994) daje przeciwny efekt: ponieważ słabsze osobniki są dość mocno wypierane z populacji, populacja staje się mało zróżnicowana i średnia wartość przystosowania całej populacji jest zbliżona do przystosowania najlepszego osobnika. Małe zróżnicowanie osobników w populacji, choć daje wysokie średnie przystosowanie populacji, zazwyczaj nie jest korzystne, ponieważ w dłuższej perspektywie prowadzi do stagnacji w populacji.

Metoda ruletki okazuje się stosunkowo miękką metodą selekcji. Po 100 pokoleniach przystosowanie najlepszego osobnika było równe 1,5, a średnie przystosowanie populacji wynosiło około 1,29. Metoda stochastycznego próbkowania daje mniejszą różnorodność populacji. Podczas gdy najlepszy osobnik miał przystosowanie równe 1,5, średnia wartość wynosiła 1,43. Na rysunku 3.6c widać, że populacja osiągnęła wyższy szczyt przystosowawczy, ale po 100 pokoleniach umiejscowiona jest dokładnie w jednym punkcie w przestrzeni fenotypowej. Oznacza to, że średnia i największa wartość przystosowania są sobie równe (wynoszą 1,5), czyli całą populację zdominowały osobniki jednego typu.

Porównując rezultaty uzyskane w czasie symulacji modeli Wersja_1 i Wersja_2 można zauważyć, że ewolucja populacji przebiega szybciej przy reprezentacji osobników za pomocą liczb rzeczywistych z jednoczes-



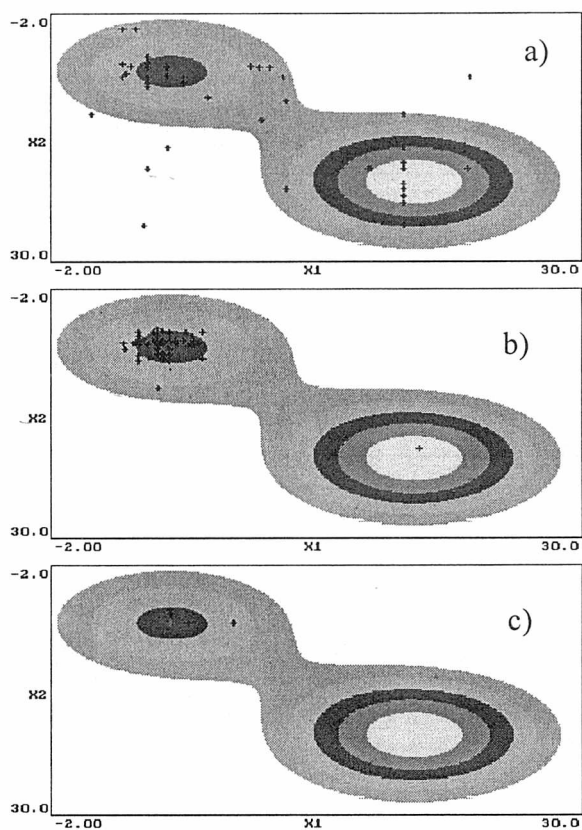
Rys. 3.6. Geny rzeczywiste, mały efekt pleiotropowy i poligeniczny (a) metoda ruletki, (b) metoda stochastyczna, (c) metoda turniejowa (strzałka pokazuje położenie populacji)

² Selekcję nazywamy miękką, jeśli stosunkowo często będzie ona pozwalać na reprodukcję osobnikom gorszym. Im bardziej twarda jest metoda selekcji, tym rzadziej gorszy osobnik będzie miał szanse reprodukcyjne, co prowadzi do zbyt dużej jednorodności osobników w populacji i tzw. przedwczesnej zbieżności GA.

nym stosowaniem efektów plejotropowego i poligenicznego. Populacja osiąga optimum globalne niezależnie od zastosowanej metody selekcji, istnieje jednak niebezpieczeństwo zatrzymania ewolucji przy twardej metodzie selekcji. Gdyby istniały inne, lepsze szczyty przystosowawcze, populacja z selekcją turniejową mogłaby ich nie znaleźć. Sama wspinaczka na szczyt przystosowawczy po jego zboczu jest dość szybka przy silnej preferencji lepszych osobników.

Symulacja modelu Wersja_3 (ewolucja fenotypów: operatory genetyczne działają bezpośrednio na argumentach funkcji celu)

W części eksperymentów, przy selekcji metodą ruletki populacja jest w stanie zasiedlić wyższy szczyt przystosowawczy w ciągu 100 pokoleń (na 10 eksperymentów



Rys. 3.7. Ewolucja fenotypów (bez poziomu genotypowego): (a) metoda ruletki, (b) metoda stochastyczna, (c) metoda turniejowa

zdarzyło się to dwukrotnie). Warto zwrócić uwagę na fakt, że we wszystkich eksperymentach pojedyncze osobniki znajdowały się na wyższym szczycie przystosowawczym, ale nie udało im się zdominować populacji. Dla pozostałych dwóch metod selekcji populacja nie była w stanie zasiedlić globalnego szczytu, nawet jeśli ewolucja trwała 1000 pokoleń.

Po obserwacji w dłuższym okresie przebiegu ewolucji z zastosowaniem dwóch ostatnich metod selekcji można stwierdzić różnicę w charakterze ewolucji: różnicowanie osobników w populacji jest większe przy stosowaniu metody stochastycznej. Zwiększanie częstości mutacji i rekombinacji powoduje, że populacja może znaleźć się na globalnym szczycie. Przykładowe przebiegi ewolucji fenotypów pokazano na rysunku 3.7.

Biorąc pod uwagę ewolucję populacji biologicznych, pojawianie się pojedynczych osobników w okolicy optimum globalnego, które nie ma perspektyw szybkiego przesunięcia całej populacji na ten szczyt, można uznać za niewiele znaczący fakt. Populacja jako całość nie zajmuje nowej, lepszej niszy, pozostaje na lokalnym szczycie, w swego rodzaju

pułapce ewolucyjnej. Tempo ewolucji nie jest zadowalające. Jeśli tworzymy taki algorytm w celu znalezienia najlepszego rozwiązania, a nie badania dynamiki populacji, to taka sytuacja może być korzystna. Przez cały czas ewolucji zapamiętywane jest najlepsze uzyskane rozwiązanie. Nawet jeśli tylko w jednym pokoleniu pojawiło się dobre rozwiązanie, leżące w okolicy optimum globalnego, to jako lepsze od tych na lokalnym szczycie będzie zapamiętane i zwrócone użytkownikowi jako satysfakcjonujące.

W podsumowaniu eksperymentów z trzema wymienionymi wyżej wersjami można stwierdzić:

Stosowanie chromosomów składających się z liczb rzeczywistych i nawet stosunkowo niewielkiej pleiotropowości i poligeniczności może być sposobem na zwiększenie efektywności ewolucji.

Charakter ewolucji zależy od stosowanej metody selekcji oraz przyjętych operatorów, dlatego też warto szukać optymalnych dla danego zadania. Dla stosunkowo miękkiej selekcji, jaką daje metoda ruletki, mogą w populacji pojawić się stosunkowo dobre (leżące na globalnym szczycie) pojedyncze osobniki, niezależnie od przyjętego sposobu kodowania. Twarda selekcja zwiększa szanse na zamknięcie populacji w pułapce ewolucyjnej. Wyniki badań symulacyjnych wskazują, że populacja najczęściej osiąga globalne optimum w modelu z rzeczywistymi chromosomami oraz efektem pleiotropowym i poligenicznym. Dzięki eksperymentom można również zaobserwować, że ewolucja populacji w Wersji_2 zajmuje mniej czasu obliczeniowego niż ewolucja binarnych chromosomów.

3.3. K-MODEL

K-model ma wbudowane w reprezentację osobnika geny nadmiarowe, pleiotropowość i poligeniczność. Geny są liczbami całkowitymi, a nie ciągami binarnymi. Każdy osobnik (potencjalne rozwiązanie) składa się z np genów fenotypowych i pewnej liczby nr genów nadmiarowych. Zarówno geny fenotypowe, jak i nadmiarowe są rozmieszczone na n_{ch} chromosomach. Liczba genów nadmiarowych może się zmieniać w trakcie ewolucji i być różna u poszczególnych osobników. Założona jest tylko maksymalna długość n_{max} pojedynczego chromosomu. Cechy fenotypowe osobnika, czyli argumenty $[p_1, p_2, \dots, p_m]$ m -wymiarowej funkcji celu $Q = Q(p_1, p_2, \dots, p_m)$, są kodowane za pomocą genów będących liczbami całkowitymi, funkcja Q zaś jest nazywana **funkcją przystosowawczą** lub **funkcją jakości** osobników. Biologia nie dostarcza nam przesłanek do sposobu odwzorowania genów na fenoty. Zakładamy, że każdy fen jest liniową kombinacją genów fenotypowych. Gen fenotypowy może oddziaływać na fen pozytywnie lub negatywnie (zwiększać lub zmniejszać wartość fenu). Taka reprezentacja umożliwia zamodelowanie procesu nazywanego *epistazą*³ (Back 1996).

³ W biologii epistazą nazywa się tłumienie przejawiania się cechy, warunkowane przez odpowiedni gen (hipostatyczny) wskutek działania genu (zwanego epistatycznym lub supresorem) należącego do innej pary alleli.

3.3.1. FORMALNY OPIS MODELU

Rozpatrywana jest ewolucja populacji w dyskretnym czasie t , jednostką czasu jest jedno pokolenie. Populacja funkcjonująca w czasie t jest oznaczona przez $\mathcal{P}(t)$ i liczy $N(t)$ osobników. Każdy osobnik $\mathbb{I}_i$ w populacji $\mathcal{P}(t)$ jest reprezentowany na poziomie genotypowym przez określoną liczbę n_{ch} chromosomów. Genom i -tego osobnika G_i składa się więc z n_{ch} chromosomów C_i^j , $j = 1, \dots, n_{ch}$ – będących wektorami liczb całkowitych. Geny fenotypowe (w liczbie np) są rozmieszczone na wszystkich chromosomach u osobnika, mogą zajmować dowolne miejsca, lecz dla uproszczenia implementacyjnego zakładamy, że są umieszczone na pierwszych miejscach chromosomów. Ich liczba może być różna na poszczególnych chromosomach, ale u wszystkich osobników są one jednakowo rozmieszczone. Dalsze pozycje na chromosomach zajmują geny nadmiarowe. Zakładamy liniowe odwzorowanie genów fenotypowych na wektor fenów $Ph = \langle p_1, \dots, p_m \rangle$, m – liczba fenów (wymiar optymalizowanej funkcji), za pomocą macierzy plejotropowości A . Elementami macierzy A są liczby rzeczywiste.

Opis formalny

Populację stanowi $N(t)$ osobników $\mathbb{I}_i$: $\mathcal{P}(t) = \langle \mathbb{I}_1, \mathbb{I}_2, \dots, \mathbb{I}_{N(t)} \rangle$;

Osobnik jest reprezentowany jako jego genom: $\mathbb{I}_i \rightarrow G_i$, $i = 1, \dots, N(t)$;

Genom jest wektorem chromosomów: $G_i = \langle C_i^1, \dots, C_i^{n_{ch}} \rangle$,

Każdy chromosom to zestaw genów fenotypowych i nadmiarowych, identyfikowanych przez numer genu w chromosomie, numer chromosomu j i numer osobnika i

$$C_i^j = \langle g_1^{i,j}, g_2^{i,j}, \dots, g_{np_j}^{i,j}, g_{np_j+1}^{i,j}, \dots, g_{np_j+nr_j^i}^{i,j} \rangle,$$

gdzie:

- C_i^j – j -ty chromosom i -tego osobnika,
- np_j – liczba genów fenotypowych na j -tym chromosomie,
- nr_j^i – liczba genów nadmiarowych na j -tym chromosomie u i -tego osobnika,
- $g_1^{i,j}, \dots, g_{np_j}^{i,j}$ – geny fenotypowe na j -tym chromosomie i -tego osobnika,
- $g_{np_j+1}^{i,j}, \dots, g_{np_j+nr_j^i}^{i,j}$ – geny nadmiarowe na j -tym chromosomie i -tego osobnika,
- $i = 1, \dots, N(t)$, $j = 1, \dots, n_{ch}$, $np_j > 0$, $nr_j^i \geq 0$, $np_j + nr_j^i \leq n_{max}$, $g_k^{i,j} \in \mathbb{C}$,

$$\sum_{k=1}^{n_{ch}} np_k = np,$$

n_{max} – maksymalna liczba genów na jednym chromosomie,

np – liczba genów fenotypowych (u każdego osobnika w populacji),

n_{ch} – liczba chromosomów (jednakowa dla każdego osobnika).

Dla każdego osobnika tworzymy wektor składający się ze wszystkich jego genów fenotypowych

$$AG_i = \langle gp_1^i, \dots, gp_{np}^i \rangle,$$

gdzie:

AG_i – wektor składający się tylko z genów fenotypowych i -tego osobnika,

gp_k^i – wartość k -tego genu fenotypowego i -tego osobnika.

AG_i jest to wektor genów $\langle gp_1^{i,1}, gp_2^{i,1}, \dots, gp_{np_1}^{i,1}, gp_1^{i,2}, \dots, gp_{np_2}^{i,2}, \dots, gp_{np_{nch}}^{i,nch} \rangle$, w którym pominięto indeks oznaczający numer chromosomu i ponumerowano je kolejno (dolny indeks).

Każdy fen i -tego osobnika (p_k^i , $i = 1, 2, \dots, \mathbb{N}(t)$, $k = 1, 2, \dots, m$) jest produktem jego genów fenotypowych oraz macierzy pleiotropowości A

$$Ph^i = \langle p_1^i, \dots, p_m^i \rangle = AG_i \times A,$$

gdzie:

Ph^i – wektor fenów i -tego osobnika,

$[A]_{np \times m}$ – macierz pleiotropowości A , ma rozmiar $np \times m$ oraz $a_{k,l} \neq 0$ jeśli k -ty gen fenotypowy wpływa na l -ty fen.

Wartość przystosowawcza (jakość) i -tego osobnika Q^i zależy bezpośrednio od jego wektora fenów

$$Q^i = Q(\mathbb{I}_i) = Q(Ph^i) = Q(AG_i \times A) = Q(p_1^i, \dots, p_m^i).$$

Opis operatorów genetycznych uwzględnionych w modelu

Rekombinacja (operuje na chromosomach, zachodzi z prawdopodobieństwem P_{rkm}):

$$\begin{aligned} G_i &= \langle C_i^1, \dots, C_i^k, \dots, C_i^{n_{\text{ch}}} \rangle; & G_j &= \langle C_j^1, \dots, C_j^k, \dots, C_j^{n_{\text{ch}}} \rangle; & P_{\text{rkm}} \\ & & & \Rightarrow \\ G_i' &= \langle C_i^1, \dots, C_i^k, \dots, C_i^{n_{\text{ch}}} \rangle; & G_j' &= \langle C_j^1, \dots, C_i^k, \dots, C_j^{n_{\text{ch}}} \rangle; \end{aligned}$$

gdzie:

G_i, G_j – osobniki przed rekombinacją,

G_i', G_j' – osobniki po rekombinacji.

Mutacja (M) operuje na genach, zachodzi z prawdopodobieństwem P_{mut} :

$$\begin{aligned} C_i^j &= \langle g_1^{i,j}, \dots, g_k^{i,j}, \dots, g_{np_j}^{i,j}, g_{np_j+1}^{i,j}, \dots, g_{np_j+nr_j^i}^{i,j} \rangle & P_{\text{mut}} \\ & \Rightarrow \\ C_i^{j'} &= \langle g_1^{i,j}, \dots, g_k^{i,j} + \Delta g, \dots, g_{np_j}^{i,j}, g_{np_j+1}^{i,j}, \dots, g_{np_j+nr_j^i}^{i,j} \rangle, \end{aligned}$$

gdzie:

C_i^j – chromosom przed mutacją genu,

$C_i^{j'}$ – chromosom po mutacji genu $g_k^{i,j}$,

Δg – losowa wartość z zadanego przedziału $[-M_{\text{max}}, M_{\text{max}}]$.

Transpozycja – zamiana genów miejscami, nadmiarowego i fenotypowego (Gould 1991), działa na genach z zadaniem prawdopodobieństwem P_{trs} :

$$C_i^j = \langle g_1^{i,j}, g_2^{i,j}, \dots, g_{np_j}^{i,j}, g_{np_j+1}^{i,j}, \dots, g_{np_j+nr_j^i}^{i,j} \rangle \xrightarrow{P_{\text{trs}}} \\ C_i^{j'} = \langle g_1^{i,j}, g_{np_j+1}^{i,j}, \dots, g_{np_j}^{i,j}, g_2^{i,j}, \dots, g_{np_j+nr_j^i}^{i,j} \rangle,$$

(przykładowa zamiana fenotypowego genu $g_2^{i,j}$ z nadmiarowym $g_{np_j+1}^{i,j}$).

Tranzycja (przeniesienie) – działa na genach, z zadaniem prawdopodobieństwem $P_{\text{tr}}:$

$$G_i = \langle C_i^l, \dots, C_i^k, \dots, C_i^{n_{\text{ch}}} \rangle, \quad G_j = \langle C_j^l, \dots, C_j^k, \dots, C_j^v, \dots, C_j^{n_{\text{ch}}} \rangle \xrightarrow{P_{\text{tr}}} \\ G_i' = \langle C_i^l, \dots, C_i^{k'}, \dots, C_i^{n_{\text{ch}}} \rangle,$$

gdzie: $C_i^{k'} = \langle g_1^{i,k}, g_2^{i,k}, \dots, g_{np_v}^{i,k}, g_{np_v+1}^{i,k}, \dots, g_{np_v+nr_v^i}^{i,k}, g_l^{j,v} \rangle$,

(i -ty osobnik otrzymał kopię l -tego genu z v -tego chromosomu j -tego osobnika (gen $g_l^{j,v}$) jako nadmiarowy (ostatni) gen w k -tym chromosomie, $0 < l \leq np_v + nr_v^j$).

Specjalizowane operatory do modelowania makromutacji polegają na odpowiednim zwiększaniu prawdopodobieństw wymienionych operatorów genetycznych, ich opis znajduje się w części dotyczącej werbalnego opisu.

Selekcja osobników do reprodukcji

Dla każdego osobnika w populacji $\mathbb{I}_i$ liczona jest jego wartość przystosowawcza Q^i . Przed obliczeniem przystosowania osobnika należy zdekodować jego genotyp ($Ph^i = AG_i \times A$). Następnie, zgodnie z rozkładem Poissona, liczona jest liczba potomków każdego osobnika z populacji. Wartość oczekiwana λ_i liczby potomków i -tego osobnika w populacji $\mathcal{P}(t)$ jest równa stosunkowi jakości (wartości przystosowawczej) tego osobnika do średniej jakości populacji, zmodyfikowana o aktualny (w danym czasie) stopień nasycenia środowiska $S(t)$:

$$\lambda_i = \frac{Q^i}{Q_{\text{sr}}(t)} S(t), \quad (3.4)$$

$$Q_{\text{sr}} = \frac{1}{N(t)} \sum_{j=1}^{N(t)} Q^j, \quad (3.5)$$

$$S(t) = 1 + a \frac{N - N(t)}{N}. \quad (3.6)$$

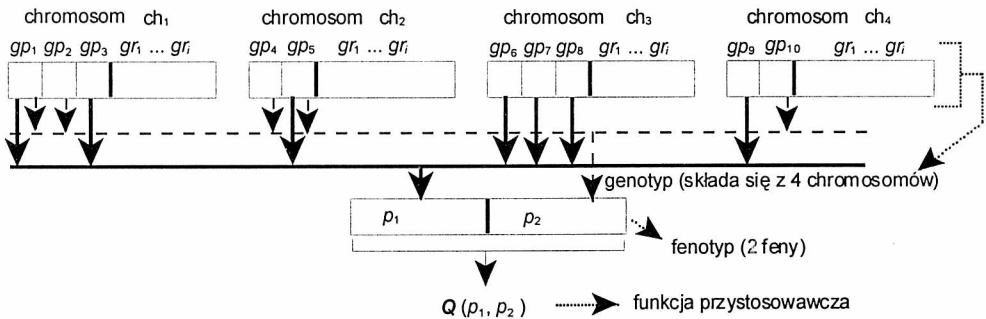
Parametr a w równaniu (3.6) reguluje tempo dochodzenia rozmiaru populacji do poziomu nasycenia środowiska N (zwykle zakładamy $a = 1$).

3.3.2. WERBALNY OPIS MODELU

Na rysunku 3.8 pokazano schemat osobnika w *K*-modelu. Na rysunku genotyp osobnika składa się z czterech chromosomów ($n_{ch} = 4$), dziesięciu genów fenotypowych ($np = 10$) rozmieszczonych na wszystkich chromosomach w liczbie: trzy na pierwszym, dwa na drugim, trzy na trzecim i dwa na czwartym chromosomie. Fenotyp osobnika liczy dwie cechy (dwuwymiarowa funkcja przystosowawcza).

Osobnik

W modelu jest ustalana maksymalna liczba genów w jednym chromosomie oraz liczba genów fenotypowych na każdym chromosomie. Liczba genów nadmiarowych u osobnika jest rezultatem działania zaimplementowanych operatorów genetycznych (tranzycji) i jest zmienna w czasie ewolucji: może się zmieniać od zera (model bez genów nadmiarowych) do wartości maksymalnej $nr = n_{ch} n_{max} - np$. Każdy gen jest liczbą całkowitą (założenie przyjęte ze względu na czas obliczeń).



Rys. 3.8. Schemat osobnika w *K*-modelu

Populacja

Populacja $\mathcal{P}(t)$ składa się z pewnej liczby osobników o jednakowej reprezentacji. Zakładamy, że średnia liczba osobników w populacji wynosi N (wartość ta może być traktowana jako nasycenie środowiska, w którym ewoluuje populacja). Oznacza to, że w czasie ewolucji liczebność populacji $N(t)$ może się wahać wokół średniej wartości. Mechanizm ten jest włączony w proces reprodukcji osobników poprzez aktualny stopień nasycenia S (wzór (3.6)).

Operatory genetyczne

Rekombinacje – jest to zmiana na poziomie chromosomowym. Każdy z chromosomów reprodukowanego osobnika może być zamieniony z odpowiadającym mu chromosomem (o tym samym numerze) innego, losowo wybranego osobnika. Rekombinacje to operator losowy, zachodzi z zadaniem prawdopodobieństwem P_{rkm} .

Mutacje – jest to również losowy operator, działa na pojedynczym genie. Ponieważ geny są liczbami całkowitymi, mutacja polega na zmianie genu o losową wartość

Δg (tzw. mutacja *pełzająca* – *creep mutation* (Basley 1994a), z zadaniem prawdopodobieństwem mutacji P_{mut} . Mutacji podlegają wszystkie geny niezależnie, zarówno fenotypowe, jak i nadmiarowe. Maksymalna zmiana genu M_{max} powodowana mutacją jest zadana jako parametr modelu. Δg jest losową wartością z zakresu $[-M_{max}, M_{max}]$. Mutacje genów nadmiarowych są *mutacjami neutralnymi*, ponieważ nie decydując o fenotypie, nie mają one żadnego wpływu na jakość powstającego osobnika. Nie jest to jednak tylko „sztuka dla sztuki”: zmiana genu nadmiarowego, kumulowana przez pokolenia, może w pewnym momencie ewolucji uwidocznić się, jeśli dany gen zajmie miejsce genu aktywnego (fenotypowego) wskutek działania innego operatora genetycznego, nazywanego transpozycją.

Transpozycja: od czasu do czasu – z zadaniem prawdopodobieństwem P_{trs} , zachodzi reorganizacja genotypu, polegająca na zamianie miejscami genów. Gen będący nadmiarowym zamienia się miejscem z genem fenotypowym, przez co staje się aktywny i wpływa na fenotyp osobnika, a zatem i na jego wartość przystosowawczą. Gen dotychczas fenotypowy staje się genem nadmiarowym. Skumulowane zmiany w genach będących przez pewną liczbę pokoleń nadmiarowymi mogą powodować duże zmiany fenotypowe, na miarę zmian makromutacyjnych. Można więc traktować ten operator jako pewien sposób modelowania makromutacji.

Tranzycja (przeniesienie): z zadaniem prawdopodobieństwem P_{trz} , pojedynczy gen, a właściwie jego kopia, może być przesunięty z jednego osobnika na innego, losowo wybranego. Skopiowaniu połączonemu z przeniesieniem podlega każdy gen, przeniesiona kopia zawsze natomiast zajmuje miejsce genu nadmiarowego na losowo wybranym chromosomie. Operator ten zaczerpnięty jest z biologii, ponieważ – jak pisze Szarski (1986, s. 16) – „Od dość dawna wiadomo, że fragmenty zapisu mogą się przenosić między gatunkami bakterii. Następujące obserwacje sugerują przenoszenie się fragmentów DNA również między gatunkami zwierząt...”.

Specjalizowane operatory dla modelowania makromutacji

Rekrudescencja⁴ – proces nazywany tutaj rekrudescencją zachodzi następująco: w każdym pokoleniu ewoluującej populacji u części osobników (zgodnie z założonym parametrem rcd – procent populacji podlegający rekrudescencji) zwiększane są prawdopodobieństwa mutacji, rekombinacji, tranzycji i transpozycji (P_{mut} , P_{rkm} , P_{trz} i P_{trs}). Powoduje to dość radykalną reorganizację genotypów tych osobników (Basley 1994a, Mayr 1992). Według Mayra jest to utrata spójności genotypów, powodująca najczęściej śmierć osobników, u których wystąpiła, niemniej część takich osobników może przeżyć i zapoczątkować wspinaczkę populacji na lepszy szczyt przystosowawczy. W biologii za takie zmiany w organizmach odpowiedzialne są wewnętrzne czynniki stabilizujące. Rekrudescencja w zaproponowanym ujęciu modeluje makromutacje niezależnie od tego, czy rozpatrujemy populacje osobników z genami nadmiarowymi czy bez. Procent populacji poddawany rekrudescencji zadawany jest przez parametr rcd .

⁴ *Rekrudescencja* – od łacińskiego *recrudesco* – pękać, odnawiać się, ang. *recrudescence* – nowy wybuch po okresie bezczynności, bierności lub drzemki, wznowienie, odnowienie.

Kryzys – ten proces daje radykalne zmiany, podobnie jak rekrudescencja, różni się natomiast zasięgiem w populacji. Dotyczy on wszystkich osobników w populacji, aczkolwiek nie może zdarzać się zbyt często. W biologii zewnętrzne czynniki stabilizujące są odpowiedzialne za ten proces. Może on zdarzać się raz na zadaną liczbę pokoleń. Kryzys jest również modelem makromutacji. Parametr t_{kr} określa, jak często zachodzi kryzys (w pokoleniach).

3.3.3. OPIS PROGRAMU KOMPUTEROWEGO K-MODEL

Do badań zaproponowanego modelu ewolucyjnego napisano program realizujący przedstawiony model. Komputerowy program **K-model**⁵ działa w środowisku Windows'95 oraz Windows NT. Jest przyjaznym programem, umożliwiającym wszechstronne, symulacyjne badanie właściwości zaproponowanego modelu ewolucji. Interfejs użytkownika składa się z systemu okienek. Wygląd ekranu po uruchomieniu programu pokazano na rysunku 3.9. Widoczne są okna zawierające informacje o budowie osobnika: liczba genów, liczba fenów, osobniki z ich wartościami przystosowania itp. Okna te mogą być widoczne lub ukryte. Główne menu programu i listwa narzędziowa pozwalają ustalić wszystkie żądane parametry oraz obserwować na bieżąco wyniki symulacji. Pasek zawierający główne menu programu umożliwia cztery polecenia.

Projekt

W tym poleceniu można tworzyć nowe projekty, odczytywać wcześniej zdefiniowane i zapamiętane projekty lub zapisać aktualny projekt do pliku. Zapisywane są wszystkie informacje, umożliwiające później kontynuację ewolucji. Można również zakończyć pracę z systemem.

Nowy – rozpoczęcie nowego projektu.

Otwórz – otwarcie zapisanego wcześniej pliku (*.km), zawierającego uprzednio zdefiniowany i zapisany projekt.

Zapisz – zapisanie aktualnego projektu do pliku (*.km).

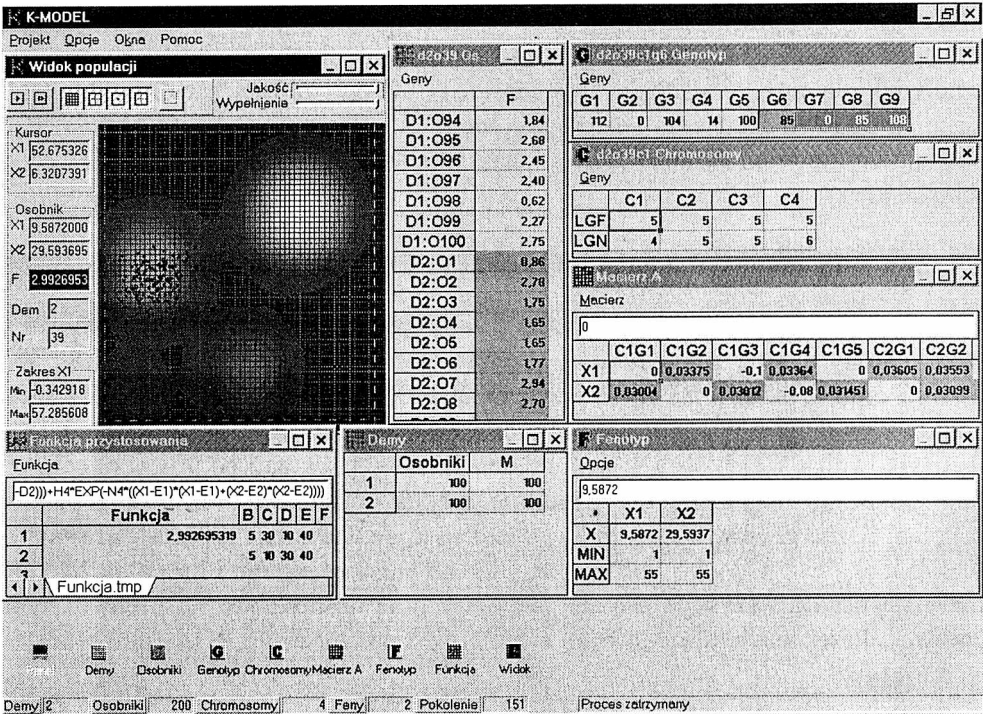
Wyjście – zakończenie pracy z programem.

Opcje

Panel sterowania – umożliwia ustalenie warunków zatrzymania programu (rys. 3.10). Może to być zadana liczba iteracji, zależność od wartości funkcji najlepszego osobnika i(lub) średniej wartości funkcji demu lub zależność od wartości fenu(ów) osobników. Warunki te można łączyć.

Parametry – otwiera okno umożliwiające ustalenie parametrów algorytmu, takich jak liczby fenów, demów, chromosomów u jednego osobnika, rozmiar populacji, poziom nasycenia i szybkość dochodzenia do nasycenia, dane dotyczące sporządzania raportów z wynikami oraz dotyczące poszczególnych operatorów genetycznych (rys. 3.11).

⁵ Implementację programu wykonano w ramach zajęć dydaktycznych ze studentami specjalizacji Inżynierii Oprogramowania, Wydział Informatyki i Zarządzania Politechniki Wrocławskiej.



Rys. 3.9. Główny ekran programu K-model

Metody selekcji – pozwala wybrać jedną z czterech zaimplementowanych metod selekcji.

Wyzeruj licznik pokoleń – umożliwia zerowanie licznika pokoleń.

Generuj raport najlepszy/najgorszy – generowanie raportu wyników (*.txt) zawierającego wartości fenów i funkcji najlepszego i najgorszego osobnika w demach w poszczególnych pokoleniach (co zadaną liczbę pokoleń).

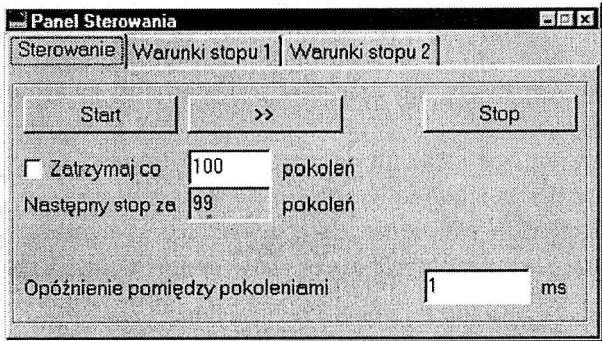
Generuj raport średnich wartości fenu – utworzenie pliku *.txt zawierającego śred-

nie jakości demów i średnie wartości fenów w demach.

Widok populacji. Okno to służy do wizualizacji przebiegu ewolucji (rys. 3.13).

Usuń dane dla raportu najlepszy/najgorszy – usuwa dane gromadzone dla tego raportu.

Usuń dane dla raportu średnich wartości fenów – usuwa dane gromadzone dla tego raportu.



Rys. 3.10. Okno Panel sterowania do ustalania warunków zatrzymania programu

Okna

Zawiera spis wszystkich okien zdefiniowanych w programie. Jest to właściwie system okienek. Są one wzajemnie powiązane, zmiany w jednym oknie wywołują zmiany w innych. W celu właściwego posługiwania się nimi należy skorzystać z instrukcji użytkownika.

Genotyp – zawiera wartości wszystkich genów, wyświetlane są geny wskazanego chromosomu wybranego osobnika. Istnieje możliwość wpisania wartości genów, wylosowania ich z całego lub z wybranego zakresu, wylosowania/ustalenia wybranych, zaznaczonych genów.

Chromosomy – dla każdego chromosomu jest ustalana liczba genów fenotypowych (LGF) i liczba genów nadmiarowych (LGN). Suma genów fenotypowych i nadmiarowych nie może być większa od 25.

Fenotyp – kolumny X1, ..., XN zawierają kolejne fenoty, tzn. współrzędne funkcji przystosowania, w wierszach jest wartość fenów najlepszego osobnika oraz zakres zmienności poszczególnych fenów.

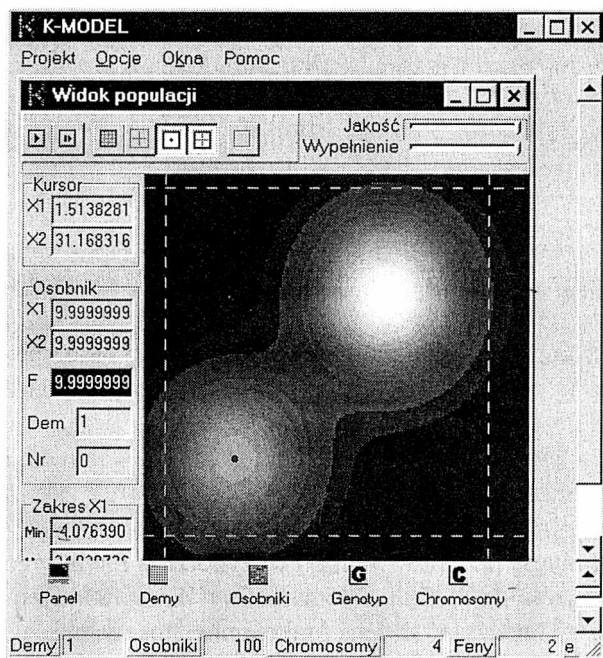
Funkcja – pozwala definiować funkcję przystosowania. Po wyborze z menu tego okna polecenia *Edytuj* następuje przejście do modułu *Formula One Workbook Designer*, gdzie można zdefiniować, zapisać i odczytać żadaną funkcję przystosowania.

Macierz A – macierz współczynników liniowej zależności fenów od genów fenotypowych. Można ręcznie wpisać wartości poszczególnych współczynników lub wylosować je tak, aby punkt startowy w przestrzeni fenotypowej był równy współrzędnym wiersza X w oknie *Fenotyp*. Można część współczynników ustalić ręcznie (np. ustalić wielkość efektów plejotropowego i poligenicznego dzięki ustaleniu części

Rys. 3.11. Okno ustalania parametrów algorytmu i programu

Rys. 3.12. Arkusz definiowania funkcji przystosowania

współczynników równych zero), a resztę dobrać losowo. Zapewniono, by losowane wartości dawały wymagany punkt startowy w przestrzeni fenotypowej i nie różniły się zbyt wielkimi wartościami, aby wyeliminować istnienie „supergenów”, wpływających zbyt mocno na jeden fen. Podobnie jak przy definiowaniu funkcji, tak i tu możemy przejść do modułu *Formula One Workbook Designer*. Dodatkowym wyborem w tym oknie jest polecenie *Generuj* (generowanie wartości współczynników macierzy).



Rys. 3.13. Okno *Widok populacji* – wizualizacja ewoluującej populacji

Osobniki populacji są zaznaczone kropkami, czerwonym kolorem jest wyróżniony najlepszy osobnik oraz żółtym migającym – osobnik, którego informacje są wyświetlane (domyślnie jest to najlepszy osobnik, kursorem myszki można wskazać innego osobnika). W dużym oknie możemy wyświetlić kształt funkcji przystosowania: jaśniejszy odcień odpowiada większej wartości przystosowania. Włączenie opcji „pokaż kratkę” powoduje, że na obraz funkcji nakłada się kratka z podziałem zakresu zmiennych x_1 i x_2 . Podczas rysowania tego wykresu brane są pod uwagę tylko dwa pierwsze argumenty funkcji, dlatego pokazany przez program wykres może nie odpowiadać rzeczywistej funkcji przystosowania – często przy funkcjach więcej niż dwóch zmiennych.

Pomoc

Umożliwia uzyskanie informacji o modelu i programie.

Zawiera ono przyciski sterujące pracą programu (typu *uruchom*, *uruchom jedno pokolenie* itp.), okna z wydrukiem aktualnych informacji – współrzędne kursora myszki (w przestrzeni fenotypowej), wartości współrzędnych (ale tylko dwóch pierwszych) i wartość przystosowawczą najlepszego osobnika w aktualnej populacji itp. Linia przerywaną zaznaczony jest obszar zmienności dwóch pierwszych argumentów funkcji przystosowania. Lewy dolny róg okna to dolne wartości przedziału zmienności x_1 i x_2 , prawy górny róg – to górne zakresy. W ten sposób, znając zakresy zmiennych, możemy się orientować o współrzędnych obszaru zasiedlonego przez popula-

4. WŁASNOŚCI K-MODELU – BADANIA SYMULACYJNE

Empiryczne badania wymagają zaplanowania i wykonania odpowiednich testów. W dalszej części pracy zajmujemy się symulacyjnym badaniem wpływu parametrów K-modelu (takich jak częstość rekombinacji, częstość i zakres mutacji itp.) na tempo i charakter ewolucji. Przez tempo ewolucji rozumiemy czas znalezienia optymalnego rozwiązania oraz czas zasiedlenia przez populację najwyższego szczytu przystosowawczego. Charakter ewolucji to charakterystyka drogi, jaką pokonuje populacja, wędrując z punktu początkowego do najwyższego szczytu, np. czy jest to droga „skokowa” z jednego szczytu na następny, czy też „płynne” przesuwanie się po zboczach szczytów i przez dolinę przystosowawczą.

Z punktu widzenia narzędzia optymalizacyjnego wystarczy, by jeden osobnik znalazł się na globalnym szczycie przystosowawczym. Jest to równoważne znalezieniu rozwiązania bliskiego globalnie najlepszemu. Natomiast z punktu widzenia analizy dynamiki populacji i wyjaśniania charakteru i szybkości ewolucji interesująca jest wędrówka populacji w krajobrazie przystosowawczym: trasa i szybkość wędrówki, „gubienie” dobrych rozwiązań – pojawianie się pojedynczych osobników na globalnym szczycie i ich wymieranie itp.

Do analizy zaproponowanego modelu można podejść w sposób teoretyczny lub empiryczny. Oba podejścia mają swoje ograniczenia (Fogel 1995). Matematyczne wykazanie własności, np. zbieżności algorytmu, jest oczywiście bardzo pożądane, własność wykazana matematycznie nie podlegałaby dyskusji, ale taka analiza, w wielu sytuacjach może być trudna lub wręcz niemożliwa¹. Algorytmy ewolucyjne z reguły zawierają nieliniowe procesy stochastyczne. Aby przeprowadzić jakąkolwiek analizę teoretyczną, zwykle przyjmuje się daleko idące założenia upraszczające. Pytanie, na ile poczynione założenia pozwalają traktować otrzymane wyniki jako dotyczące interesującego nas algorytmu ewolucyjnego, pozostaje bez odpowiedzi. Biorąc pod uwagę tę wątpliwość oraz stopień trudności analizy matematycznej, w niniejszej pracy skupimy się na badaniach empirycznych. W dziedzinie algorytmów ewolucyjnych bada-

¹ Więcej na temat teoretycznej analizy algorytmów ewolucyjnych czytelnik może znaleźć w pracy: David B. Fogel, *Evolutionary Computation Toward a New Philosophy of Machine Intelligence*, IEEE Press, 1995, s. 135–155. Przykłady innych prac zawierających teoretyczne aspekty GA to: Poli, Longdon 1997, Thierens i in. 1998, O'Reilly, Goldberg 1998, Giruere, Goldberg 1998.

nia empiryczne są powszechnie stosowane i akceptowane, szczególnie do celów porównawczych różnych podejść, algorytmów itp. (Fogel 1995, Michalewicz 1996, Cantú-Paz 1995 i in.).

Ze względu na cel badań, populacja początkowa nie będzie dobierana losowo, jak to zwykle czynione jest w literaturze przedmiotu, lecz będzie „ulokowana” na lokalnym szczycie. Dla funkcji mającej więcej niż dwa optima, populacja będzie umieszczana na jednym ze szczytów oddzielonych innymi od najlepszego.

Losowe rozmieszczenie populacji początkowej powoduje, że porównanie poszczególnych eksperymentów jest wątpliwe. W części eksperymentów populacja początkowa mogłaby się tak rozłożyć, że część osobników byłaby już na starcie na zboczach wyższego szczytu przystosowawczego, bądź bliżej tego szczytu niż w innych eksperymentach. Zasiadlenie przez populację globalnego szczytu przystosowawczego mogłoby sprowadzić się tylko do zwykłej wspinaczki po zboczu, a z tym, jak powszechnie wiadomo, algorytmy ewolucyjne radzą sobie dobrze. Wymuszenie startu populacji zawsze z lokalnego optimum umożliwia testowanie możliwości ucieczki z pułapki ewolucyjnej, co oznacza zdolność populacji do opuszczenia lokalnego optimum. Wiadomo, że jest to ważną i pożądaną cechą każdej metody optymalizacji.

Z uwagi na stochastyczny charakter modelu każdy eksperyment wykonywano co najmniej dziesięciokrotnie. Uzyskane wyniki zilustrowano danymi z pojedynczego przebiegu symulacji, charakterystycznego dla danego testu, oraz omówiono uśrednione wyniki dotyczące tempa i charakteru ewolucji.

4.1. FUNKCJE TESTOWE

Do wykonania badań potrzebna jest funkcja, która będzie łatwo modyfikowalna, to znaczy łatwo będzie ustalić liczbę, położenie, wysokość i stromość szczytów oraz możliwe będzie ustalenie jej dowolnego wymiaru (liczba argumentów). Proponowane w literaturze przedmiotu funkcje testowe nie spełniają tych wymagań w wystarczającym stopniu, dlatego też zaproponowano własną postać funkcji testowej. Jest to **rodzina funkcji eksponencjalnych** Q dana ogólnym wzorem (4.1), każda natomiast ze stosowanych funkcji Q_i ma zdefiniowaną inną liczbę zmiennych oraz liczbę, nachylenia, wysokości i położenia szczytów. Nie oznacza to rezygnacji z eksperymentowania z typowymi funkcjami, np. znanymi jako **funkcje testowe de Jonga**.

Ogólna postać rodziny funkcji eksponencjalnych

Ogólny wzór dla rodziny funkcji eksponencjalnych dany jest równaniem

$$Q(x_1, x_2, \dots, x_m) = \sum_{i=1}^{I_s} h_i e^{-n_i \sum_{j=1}^m (x_j - x_j^i)^2}, \quad (4.1)$$

gdzie:

x_i – i -ta współrzędna funkcji,

x_j^i – wartość j -tej współrzędnej i -tego szczytu,

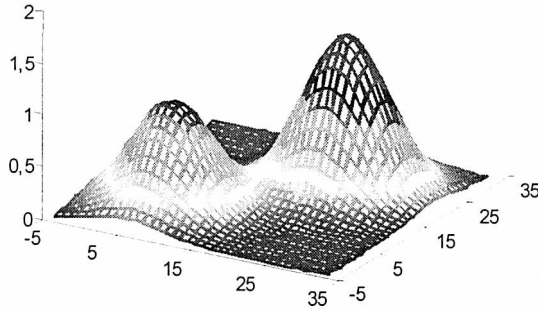
m – wymiar funkcji,
 l_s – liczba szczytów funkcji,
 h_i – wysokość i -tego szczytu,
 n_i – nachylenie i -tego szczytu.

$Q_1(x_1, x_2)$: w wielu podstawowych eksperymentach jako funkcję przystosowawczą stosowano dwuwymiarową funkcję z dwoma szczytami, $Q_1(x_1, x_2)$, której analityczna postać wyraża się wzorem

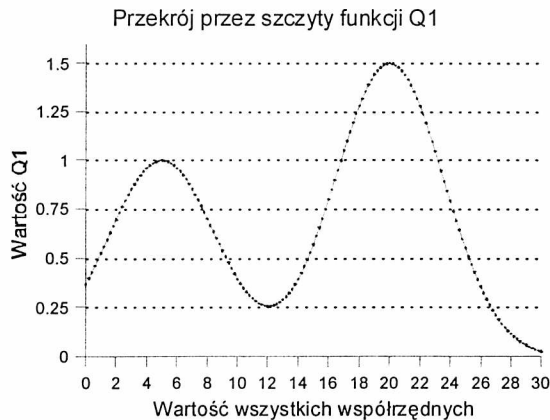
$$Q_1(x_1, x_2) = h_1 e^{-(x_1 - x_1^1)^2 + (x_2 - x_2^1)n_1} + h_2 e^{-(x_1 - x_1^2)^2 + (x_2 - x_2^2)n_2}, \quad (4.2)$$

gdzie: $h_1 = 1$, $h_2 = 1,5$, $(x_1^1, x_2^1) = (5, 5)$, $(x_1^2, x_2^2) = (20, 20)$, $n_1 = n_2 = 0,02$.

Przyjęto, że za wystarczający zakres zmienności x_1 i x_2 można uznać przedział $[0, 30]$. Na rysunku 4.1 pokazano kształt tej funkcji, a na rysunku 4.2 – jej przekrój (profil) wzdłuż obu szczytów.



Rys. 4.1. Wizualizacja funkcji eksponencjalnej z dwoma szczytami (Q_1)



Rys. 4.2. Przekrój Q_1 przez szczyty

$Q_2(x_1, x_2, x_3, x_4, x_5)$: jest to funkcja pięciu zmiennych z dwoma szczytami

$$Q_2(x_1, x_2, x_3, x_4, x_5) = \sum_{i=1}^2 h_i e^{(-n_i \sum_{j=1}^5 (x_i - x_i^j)^2)} \quad (4.3)$$



gdzie: $h_1 = 1, h_2 = 1,5, (x_1^1, x_2^1, x_3^1, x_4^1, x_5^1) = (5, 5, 5, 5, 5), (x_1^2, x_2^2, x_3^2, x_4^2, x_5^2) = (20, 20, 20, 20, 20), n_1 = 0,02, n_2 = 0,00225$ (na rysunku 4.3 pokazano jej przekrój).

Zakres zmienności parametrów: $[0, 30]$.

Rys. 4.3. Przekrój przez szczyty funkcji Q_2

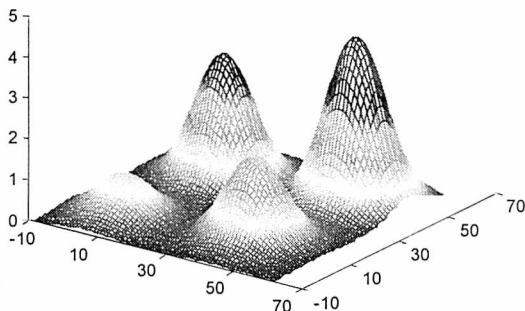
$Q_3(x_1, x_2)$: funkcja dwuwymiarowa z czterema szczytami

$$Q_3(x_1, x_2) = \sum_{i=1}^4 h_i e^{(-n_i ((x_1 - x_1^i)^2 + (x_2 - x_2^i)^2))}, \quad (4.4)$$

gdzie: $h_1 = 1, h_2 = 2, h_3 = 3, h_4 = 4,$
 $(x_1^1, x_2^1) = (5, 5), (x_1^2, x_2^2) =$
 $= (40, 10), (x_1^3, x_2^3) = (10, 40),$
 $(x_1^4, x_2^4) = (45, 45), n_1 = n_2 = n_3 =$
 $= n_4 = 0,01.$

Zakres zmienności parametrów:
 $[0, 60].$

Rys. 4.4. Wizualizacja funkcji eksponencjalnej z czterema szczytami Q_3



$Q_4(x_1, ..., x_{30})$: funkcja z dwoma szczytami, trzydzieści wymiarów

$$Q_4(x_1, x_2, ..., x_{30}) = \sum_{i=1}^2 h_i e^{(-n_i \sum_{j=1}^{30} (x_i - x_i^j)^2)}, \quad (4.5)$$

gdzie: $h_1 = 1, h_2 = 1,5, x_i^1 = 5, i = 1, ..., 30, x_i^2 = 20, n_1 = 0,003, n_2 = 0,0004.$

Zakres zmienności parametrów: $[0, 30].$

$Q_5(x_1)$: funkcja jednej zmiennej z dwoma szczytami

$$Q_5(x_1) = \sum_{i=1}^2 h_i e^{(-n_i(x_i - x_i^j)^2)}, \quad (4.6)$$

gdzie: $h_1 = 3, h_2 = 5, x_1^1 = 5, x_2^2 = 60, n_1 = 0,0018, n_2 = n_1$.

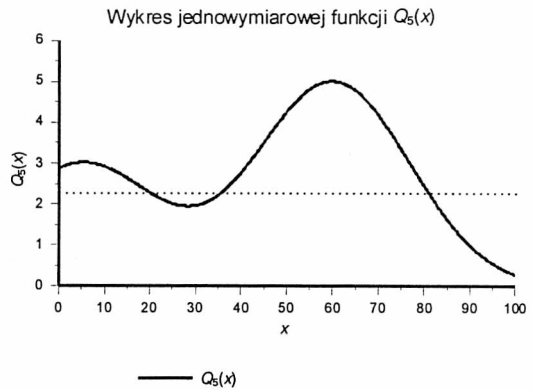
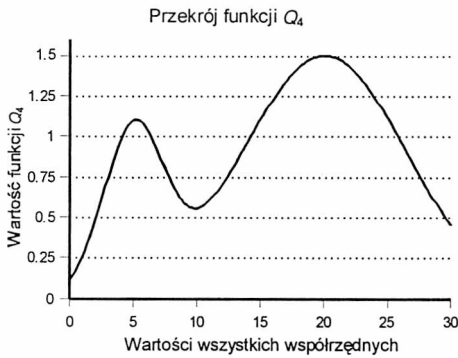
Zakres zmienności parametrów: $[0, 70]$.

$Q_6(x_1, \dots, x_{10})$: funkcja z dwoma szczytami, dziesięć wymiarów

$$Q_6(x_1, x_2, \dots, x_{10}) = \sum_{i=1}^2 h_i e^{(-n_i \sum_{j=1}^{10} (x_i - x_i^j)^2)}, \quad (4.7)$$

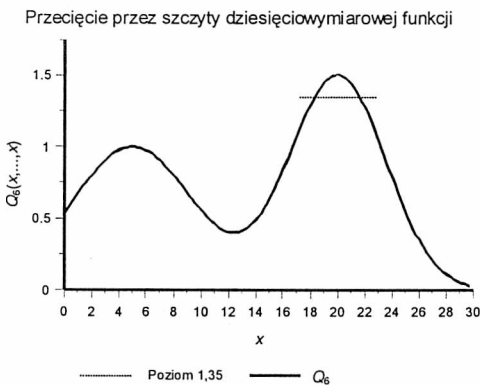
gdzie: $h_1 = 1, h_2 = 1,5, x_i^1 = 5, i = 1, \dots, 10, x_i^2 = 20, n_1 = 0,0025, n_2 = 0,004$.

Zakres zmienności parametrów: $[0, 30]$.

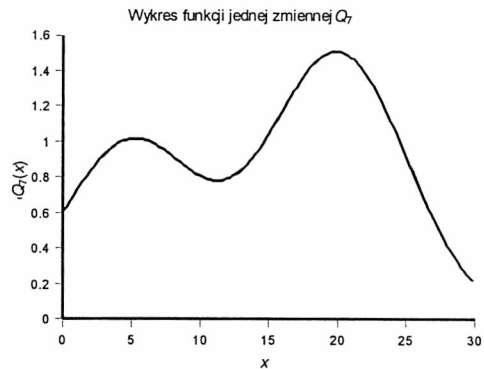


Rys. 4.5. Przekrój przez szczyty 30-wymiarowej funkcji Q_4

Rys. 4.6. Wykres Q_5



Rys. 4.7. Przekrój przez szczyty dla Q_6



Rys. 4.8. Wykres jednowymiarowej funkcji Q_7

$Q_7(x_1)$: jednowymiarowa funkcja z dwoma szczytami położonymi blisko siebie

$$Q_7(x_1) = \sum_{i=1}^2 h_i e^{(-n_i(x_i - x_i^j)^2)}, \quad (4.8)$$

gdzie: $h_1 = 1$, $h_2 = 1,5$, $x_1^1 = 5$, $x_2^2 = 20$, $n_1 = 0,02$, $n_2 = n_1$.

Zakres zmienności parametrów: $[0, 30]$.

Pięć funkcji testowych de Jonga

Funkcje te są uważane za dość dobry test, zwykle dla algorytmów, w których chromosom jest łańcuchem bitów (Goldberg 1989). Zadaniem algorytmu jest znalezienie minimum tej funkcji, funkcja przystosowawcza musi być zatem przekształconą funkcją optymalizowaną (ewolucja działa w kierunku zwiększenia przystosowania).

Funkcja F_1 – model sferyczny – jest najprostszą z możliwych do testowania funkcji

$$F_1(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i^2. \quad (4.9)$$

Szukane: $\min(F_1) = F_1(0, 0, \dots, 0) = 0$.

Rozmiar: 3.

Zakres parametrów: $[-5, 12, 5, 12]$.

Funkcja F_2 – uogólniona funkcja Rosenbrocka

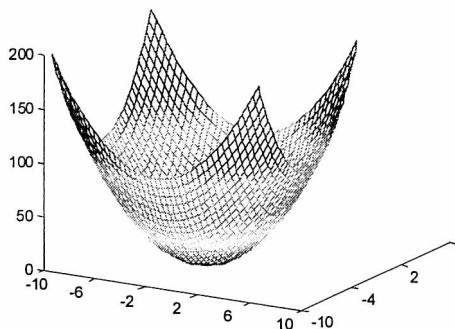
$$F_2(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} 100 (x_i^2 - x_{i+1})^2 + (x_i - 1)^2. \quad (4.10)$$

Szukane: $\min(F_2) = F_2(1, 1, \dots, 1) = 0$.

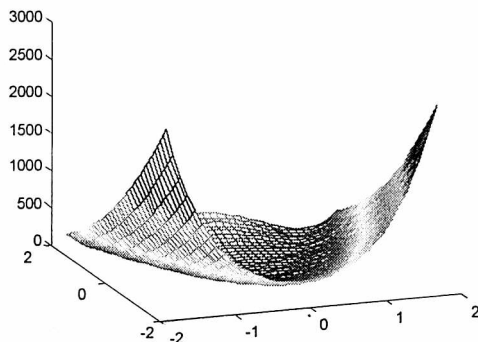
Rozmiar: 2.

Zakres parametrów: $[-2, 048, 2, 048]$.

Jest to trudna funkcja do optymalizacji, swoim kształtem przypomina długi, bardzo stromy wąwóz wygięty w kształcie paraboli. Dno tego stromego wąwozu jest delikatnie nachylone, tworząc minimum w punkcie $(1, 1)$.



Rys. 4.9. Wizualizacja funkcji F_1



Rys. 4.10. Wizualizacja funkcji F_2

Funkcja F_3 – krokowa

$$F_3(x_1, x_2, \dots, x_n) = \sum_{i=1}^n \lfloor x_i \rfloor. \quad (4.11)$$

Szukane: $\min(F_3) = F(x_i) = -30$, dla $x_i \leq -5, 12$, $i = 1, 2, 3, 4, 5$.

Rozmiar: 5.

Zakres parametrów: $[-5, 12, 5, 12]$.

Algorytmy nie mające zmiennej długości kroku nie dają sobie rady z szukaniem optimum takiej funkcji. Ogólnie, funkcja F_3 nie ma globalnego minimum. Wartość minimalna zależy od przyjętych ograniczeń na zmienność argumentów funkcji (leży na brzegu obszaru).

Funkcja F_4 – funkcja potęgowa z nałożonym szumem losowym

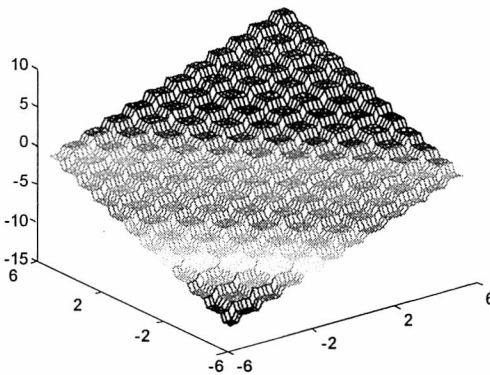
$$F_4(x_1, x_2, \dots, x_n) = \sum_{i=1}^n i x_i^4 + \text{gauss}(0, 1). \quad (4.12)$$

Szukane: $\min(F_4)$, bez nałożonego szumu to $\min(F_4) = F_4(0, 0, \dots, 0) = 0$.

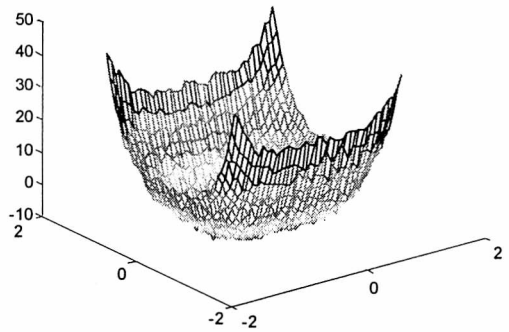
Rozmiar: 30.

Zakres parametrów: $[-1, 28, 1, 28]$.

Funkcja F_4 jest użyteczna do badania, czy algorytm może być skutecznym narzędziem do szukania optimum zasumionych funkcji. Dodawana wartość funkcji Gaussa o średniej 0 i wariancji 1 powoduje, że funkcja nigdy nie przyjmuje w tym samym punkcie takiej samej wartości.



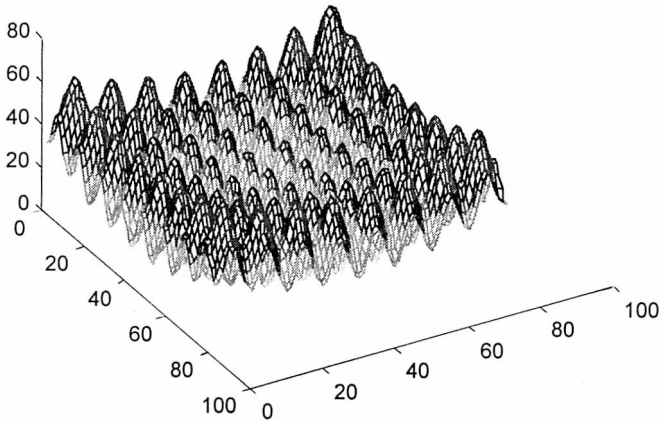
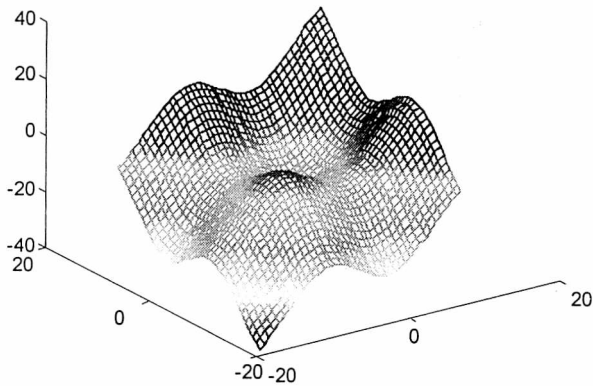
Rys. 4.11. Wizualizacja funkcji F_3



Rys. 4.12 Wizualizacja funkcji F_4

Funkcja F_5

$$F_5(x_1, x_2) = \frac{1}{\frac{1}{K} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}}. \quad (4.13)$$

Rys. 4.14. Wizualizacja funkcji F_6 Rys. 4.15. Wizualizacja funkcji F_7

Założono, że wartości funkcji od F_1 do F_7 nie przekraczają ustalonej wartości $\max_i$ i jako funkcję przystosowania stosowano

$$F'_i = \max_i - ((F_i(\mathbf{x}') + P_i(\mathbf{x}')),$$

gdzie:

$F_i(\mathbf{x}')$ – wartość funkcji F_i w punkcie $\mathbf{x}'$,

$\mathbf{x}'$ – n -wymiarowy wektor ($\mathbf{x}' = (x_1', x_2', \dots, x_n')$),

$P_i(\mathbf{x}')$ – kara za przekroczenie dopuszczalnych, ustalonych dla danej funkcji, zakresów zmiennych x_j w punkcie $\mathbf{x}'$ (zwykle nie zakładano kary, program zabezpiecza przed przekraczaniem wartości argumentów funkcji).

Systematyczne badania wykonano dla rodziny funkcji eksponencjalnych, dla pozostałych natomiast sprawdzano skuteczność zaproponowanego modelu w kilku przebiegach symulacyjnych.

4.2. REKOMBINACJE I MUTACJE W EWOLUCJI

Celem przedstawionych w tej części pracy eksperymentów jest sprawdzenie czy, i w jaki sposób, poszczególne operatory genetyczne, takie jak mutacja, rekombinacja, kryzys itp., wpływają na charakter ewolucji – jej szybkość i zdolność ucieczki z pułapki ewolucyjnej, jaką jest lokalne optimum. Jako podstawową funkcję do eksperymentów wybrano funkcję pięciu zmiennych z dwoma szczytami (Q_2).

Parametry modelu, które należy ustalić przed uruchamianiem programu

Parametry ogólne eksperymentu:

- liczba fenów, jest to wymiar optymalizowanej funkcji,
- liczba demów, równe 1, jeśli populacja nie jest podzielona,
- liczba chromosomów, domyślnie przyjmuje się 4,
- parametr a (tempo dochodzenia rozmiaru populacji do poziomu nasycenia środowiska, ważny, gdy opcja *stała liczebność populacji* jest ustawiona na *nie*), domyślnie ustalany 1,
- stała liczebność populacji (*tak, nie*), wartość *nie* ma sens tylko w metodzie selekcji Poissona, parametr ustalony domyślnie na *tak*.

Parametry ogólne algorytmu, które mogą być zmieniane w trakcie ewolucji:

- zachowaj najlepszego (*tak, nie*), domyślnie *nie*,
- prawdopodobieństwo rekombinacji P_{rkm} ,
- prawdopodobieństwo mutacji P_{mut} ,
- maksymalna zmiana mutacyjna M_{max} ,
- prawdopodobieństwo transpozycji P_{trs} ,
- prawdopodobieństwo tranzycji P_{trz} ,
- prawdopodobieństwo migracji P_{mig} ,

Opcje migracji:

- osobnik migrujący (*najlepszy, losowy*),
- dem docelowy (*najlepszy, losowy*),
- osobnik zastępowany: (*najgorszy, losowy*).
- rekrudescencja:
 - procent populacji podlegający rekrudescencji, rcd
 - dla osobników podlegających rekrudescencji: $P_{\text{rkm}}, P_{\text{mut}}, P_{\text{trs}}, P_{\text{trz}}, P_{\text{mig}}$,
- kryzys:
 - co ile pokoleń, t_{kr} ,
 - dla osobników podlegających kryzysowi: $P_{\text{rkm}}, P_{\text{mut}}, P_{\text{trs}}, P_{\text{trz}}, P_{\text{mig}}$,
- metoda selekcji (wybór jednej):
 - Poissona,
 - ruletka,
 - stochastyczna wg reszt bez powtórzeń,
 - deterministycznego próbkowania.

Po ustaleniu parametrów ogólnych eksperymentu należy zdefiniować:

- liczbę osobników i ewentualnie poziom nasycenia w demach (populacji),
- funkcję przystosowawczą, zakres zmienności argumentów funkcji i ewentualnie położenie początkowej populacji,
- liczbę genów fenotypowych i genów nadmiarowych oraz początkowe wartości genów – zadane wartości, losowe z całego dopuszczalnego przedziału (całkowite wartości typu *Integer*) lub losowe z zadanego przedziału.
- macierz plejotropowości A : można zadać każdy jej element (wtedy punkt początkowy ewolucji w przestrzeni fenotypowej nie musi być zachowany), można zadać część jej elementów (np. elementy zerowe i ujemne), a reszta zostanie wygenerowana automatycznie w taki sposób, aby zachować, w miarę możliwości, położenie początkowej populacji i porównywalne wartości losowanych elementów macierzy.

W omówionych dalej eksperymentach stosowano Q_2 jako funkcję przystosowania oraz przyjęto:

- liczba fenów = 5,
- liczba demów = 1,
- liczba chromosomów = 4,
- parametr $a = 1$,
- stała liczebność populacji: *tak*,
- zachowaj najlepszego: *nie*,
- metoda selekcji: Poissona,
- liczba osobników w populacji = 100,
- położenie początkowej populacji: cała populacja skupiona w niższym szczycie przystosowawczym (punkt o współrzędnych: (5, 5, 5, 5, 5)),

Zmiana któregoś z podanych parametrów będzie wyraźnie zaznaczona w opisie eksperymentu. Wartości pozostałych parametrów podane są przy opisie poszczególnych eksperymentów.

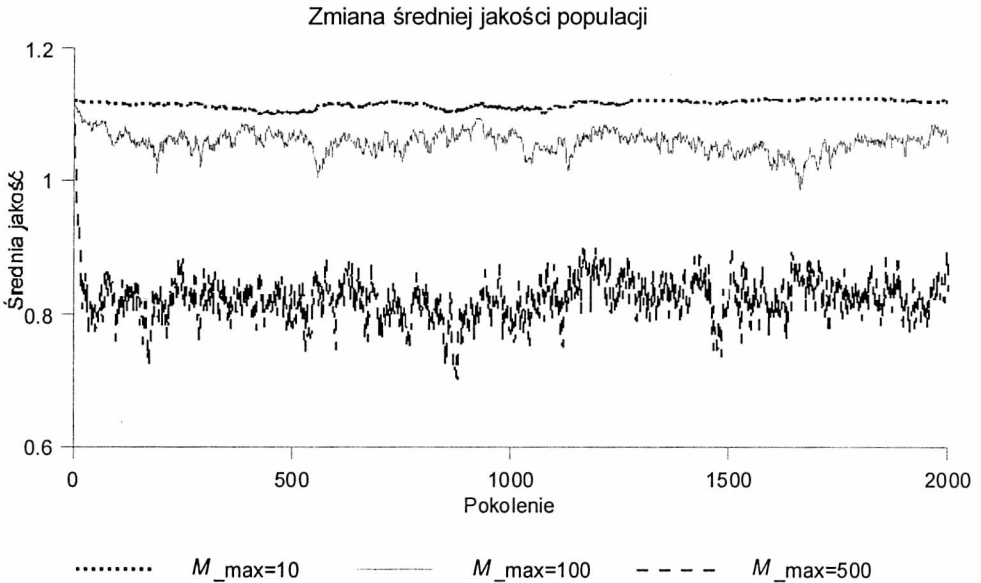
Populacja początkowa zawsze zawiera osobniki bez genów nadmiarowych. Jeśli eksperyment jest prowadzony z genami nadmiarowymi, powstają one stopniowo w populacji dzięki operatorowi tranzykcji. Pozwala to na ustalenie jednakowych warunków początkowych poszczególnych eksperymentów.

4.2.1. ROLA REKOMBINACJI W K-MODELU

Eksperyment 1: *Jaki jest przebieg ewolucji przy niewielkim prawdopodobieństwie mutacji i bardzo małych zmianach mutacyjnych?*

$P_{mut} = 0,20$, $M_{max} = 10$, początkowa wartość genów: 1000 (wszystkie geny równe), $P_{trs} = 0$, $P_{trz} = 0$, $P_{mig} = 0$ (nie dotyczy, ewoluuje cała populacja, nie podzielona na demy). Macierz A składa się z zer, dodatnich i ujemnych elementów, dość jednorodna, elementy macierzy A rzędu $10^{-3} \div 10^{-4}$, $P_{rkm} = 0,25$, po czym zostaje zwiększane (0,5, 0,75, 1), funkcja Q_2 .

We wszystkich przebiegach średnia jakość populacji jest bliska wartości optimum lokalnego (rys. 4.16, linia kropkowana). Populacja jest bardzo skupiona, nie jest w stanie „penetrować” większego obszaru przestrzeni fenotypowej, nie następuje zróżnicowanie osobników (rys. 4.18)². Po zwiększeniu P_{rkm} aż do jedności nie otrzymujemy lepszych rezultatów.



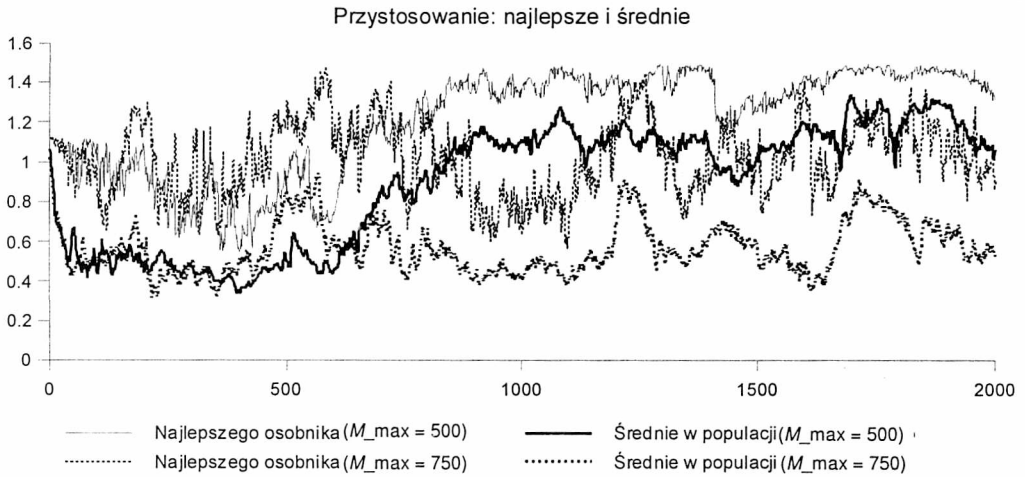
Rys. 4.16. Zmiana średniego przystosowania dla $P_{rkm}=0,25$, $P_{mut}=0,2$ i różnej maksymalnej zmianie mutacyjnej

Eksperyment 2: *Wpływ rekombinacji na ewoluującą populację przy zwiększonej maksymalnej zmianie mutacyjnej.*

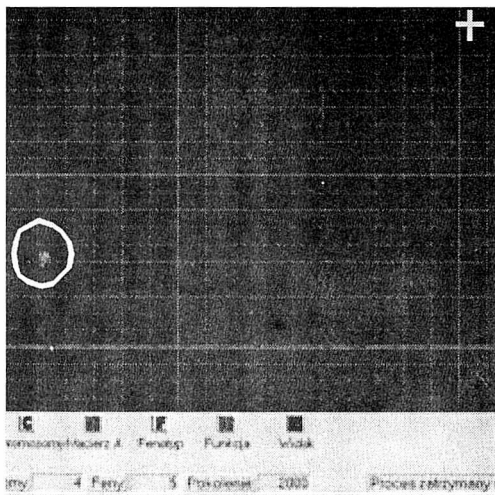
Sprawdźmy, czy zmiana $M_{\max}$ do 100 (10% wartości początkowych genów) zmieni przebieg ewolucji.

Dla $P_{rkm}=0,25$ w żadnym z 10 przebiegów po 10 000 pokoleń populacja nie jest w stanie osiągnąć wyższego szczytu (4.19). Populacja jest bardziej zróżnicowana (jej średnia jakość jest niższa) niż przy $M_{\max}=10$, ale nie ma szans na przejście na wyższy szczyt (rys. 4.16, linia ciągła). Zwiększenie częstości rekombinacji do 0,5 i 1 powoduje zwiększenie różnorodności w obrębie populacji, niemniej optymalny szczyt nie zostaje zasiedlony ani nawet znaleziony przez pojedynczego osobnika. Nic nie wskazuje, by dalsze zwiększanie liczby pokoleń mogło coś zmienić.

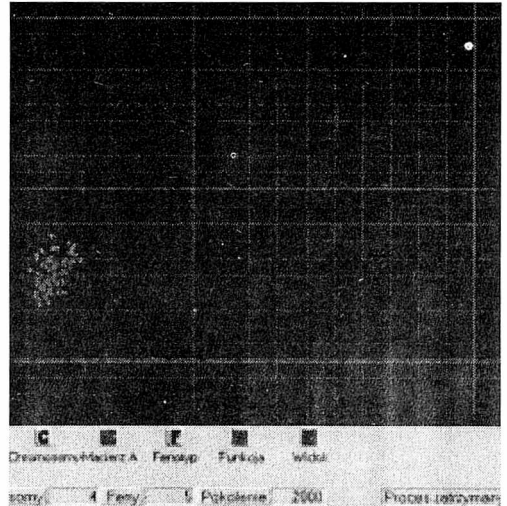
² Należy zwrócić uwagę, że na rysunku widoczne są zmiany tylko dwóch pierwszych argumentów funkcji (x_1 i x_2). W tym jednak przypadku wartości pozostałych zmiennych mieszczą się na niskim szczycie.



Rys. 4.17. Zmiana średniego przystosowania w populacji i przystosowanie najlepszego osobnika przy $P_{\text{rkm}} = 1$, $P_{\text{mut}} = 0,2$, $M_{\max} = 500$ oraz $M_{\max} = 750$



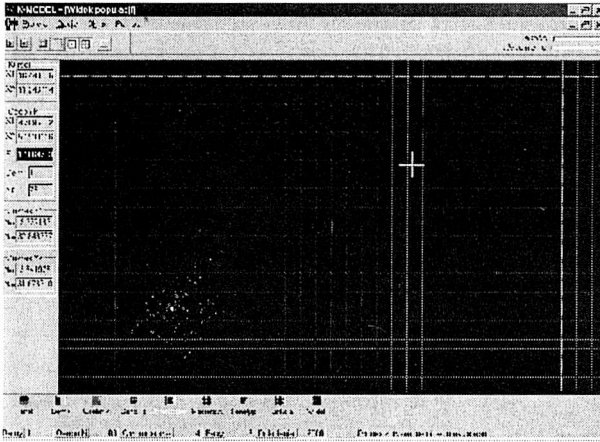
Rys. 4.18. Populacja na niskim szczycie po 2000 pokoleniach, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $M_{\max} = 10$ (kółkiem zaznaczono położenie populacji, która znajduje się tu na lokalnym szczycie, krzyżykiem – drugi szczyt)



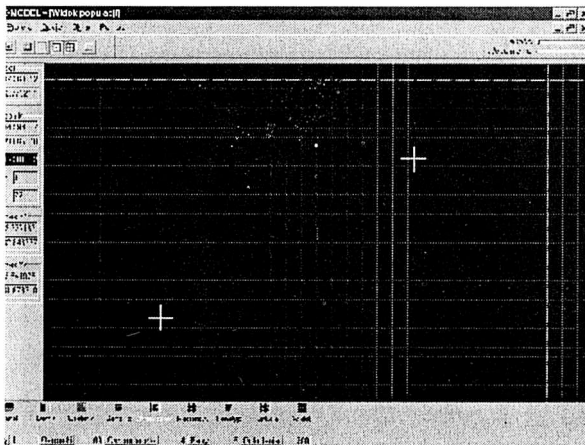
Rys. 4.19. Ewolucja po 2000 pokoleniach, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, większy zakres mutacji: $M_{\max} = 100$ (ten sam wycinek powierzchni funkcji co na rysunku 4.18)

Eksperyment 3: *Rola rekombinacji w ewolucji populacji przy dużej maksymalnej zmianie mutacyjnej.*

Maksymalna zmiana mutacyjna jest zwiększona do $M_{\max} = 500$. Dla $P_{\text{rkm}} = 0,25$ populacja jest zdecydowanie bardziej różnorodna, niż przy małych zmianach mutacyjnych,



Rys. 4.20 Ewolucja po 2000 pokoleniach, $P_{rkm} = 0,25$, $P_{mut} = 0,2$, $M_{max} = 500$



Rys. 4.21. $P_{rkm} = 1$, $P_{mut} = 0,2$, $M_{max} = 500$, pokolenie 700, populacja w drodze na wysoki szczyt

by średnia jakość populacji przekroczyła wartość 1,1 (wysokość niskiego szczytu). Przy $M_{max} = 500$ populacja była w stanie znaleźć najlepsze rozwiązanie na wysokim szczycie (powyżej wartości 1,35) przy $P_{rkm} = 0,75$: zajmowało to jej około 700 pokoleń, przy $P_{rkm} = 0,5$ populacja nie radziła sobie z postawionym jej zadaniem.

jednakże ewoluuje tylko w obrębie niższego szczytu (rys. 4.16). Przy zastosowanie najlepszego osobnika po 2 tysiącach pokoleń waha się w granicach 0,9–1,0, natomiast średnia jakość populacji jest niższa o około 0,1–0,2 (przystosowanie najgorszego osobnika spada do 0,3). Wartości te jednoznacznie wskazują na to, iż cała populacja krąży wokół lokalnego szczytu.

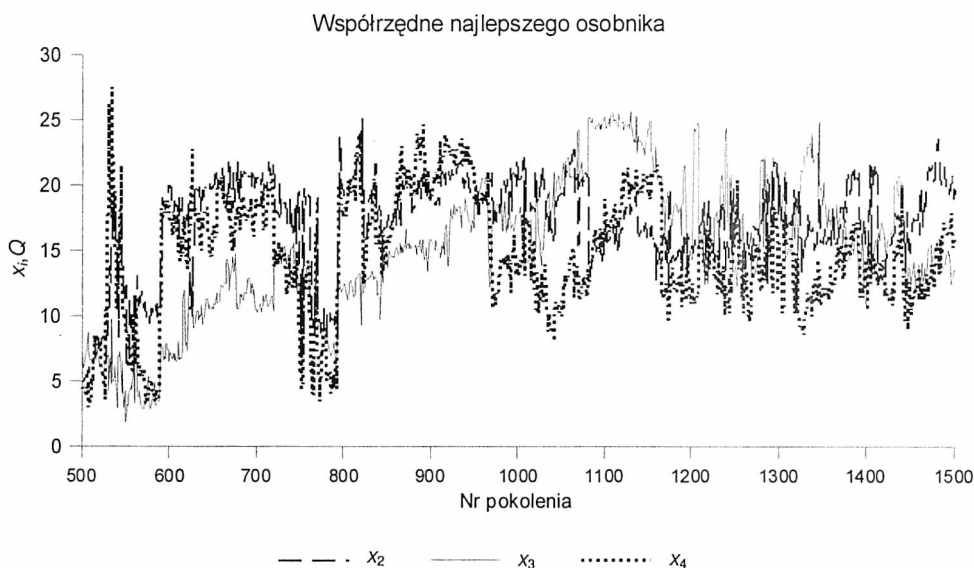
Wzrost prawdopodobieństwa rekombinacji powoduje większe wahania średniej jakości populacji, co oznacza, że populacja penetruje większy obszar krajobrazu przystosowawczego, ale dla $P_{rkm} \leq 0,75$ nie wystarcza to na przejście na wysoki szczyt. Przy rekombinacjach $P_{rkm} = 1$ obserwuje się już bardzo duże wahania średniej jakości populacji (rys. 4.17). Populacja znajduje rozwiązanie na wysokim szczycie zazwyczaj w czasie krótszym niż 1000 pokoleń (rys. 4.21 i 4.22)³. Zróżnicowanie populacji jest bardzo duże i dlatego jej średnia jakość jest stosunkowo niska. W ciągu dziesięciu przebiegów po 2000 pokoleń nie zdarzyło się,

³ Na rysunkach punktami zaznaczone są położenia osobników, jaśniejszy punkt oznacza najlepszego osobnika. Na niektórych rysunkach (zawsze dla funkcji dwuwymiarowych) odcieniami szarości są zaznaczone obszary o różnych wartościach funkcji przystosowania: jaśniejszy kolor odpowiada wyższej wartości funkcji. Na rysunku zaznaczono drugi szczyt jasnym krzyżykiem, jego położenie na ciemnym ekranie jest takie samo dla funkcji Q_2 , Q_4 , Q_6 . Bez zachowania kolorów rysunki nie zawsze są czytelne. Zamieszczono je, gdyż mają charakter poglądowy, ich zadaniem jest pokazanie typowego zachowywania się ewoluującej populacji w różnych warunkach.

samym kształcie, wysokościach i umiejscowieniu szczytów. Przeprowadzamy eksperymenty, podobne do poprzednich, stosując funkcję Q_1 .



Rys. 4.24. Wybrane współrzędne najlepszego osobnika w przestrzeni fenotypowej ($P_{\text{rkm}} = 1$, $P_{\text{mut}} = 0,1$, $M_{\text{max}} = 100$, początkowa wartość wszystkich x_i wynosi 5, x_1 i x_2 ulegały jeszcze mniejszym wahaniom)



Rys. 4.25. Wybrane współrzędne najlepszego osobnika ($P_{\text{rkm}} = 1$, $P_{\text{mut}} = 0,1$, $M_{\text{max}} = 500$, początkowe wartości wszystkich x_i wynosiły 5)

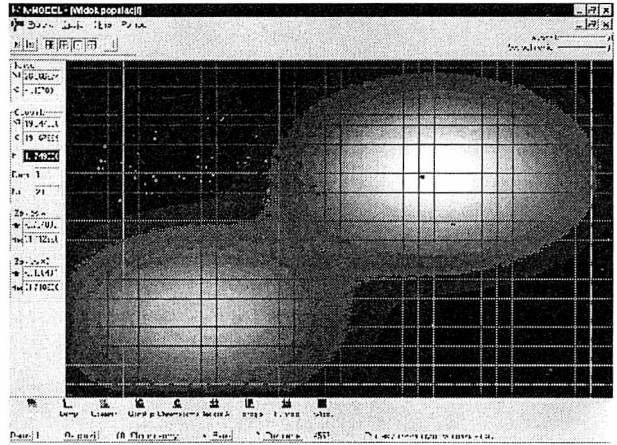
Eksperyment 4: Rola rekombinacji w ewolucji w dwuwymiarowym krajobrazie przystosowawczym.

W eksperymentach korzystamy z funkcji zdefiniowanej jako $Q_1(x_1, x_2)$, zmieniamy częstość rekombinacji (P_{rkm}) i maksymalną zmianę mutacyjną M_{max} .

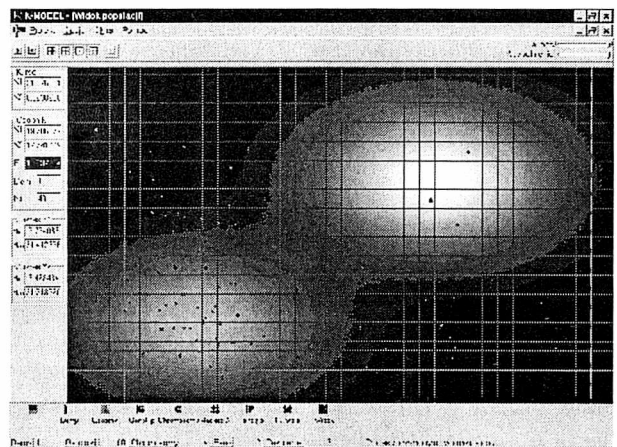
Mimo iż maksymalna zmiana mutacyjna genu wynosi 10 (przy początkowych wartościach genów równych 1000), to przy $P_{\text{rkm}} = 1$ i $P_{\text{mut}} = 0,2$, populacja jest bardzo mocno zróżnicowana (rys. 4.26). Najlepszy osobnik jest raz na niższym szczycie, raz na zboczu wyższego szczytu. W pokoleniu 4556. pojawił się osobnik na wysokim szczycie.

Kolejny raz – w pokoleniu 4601. Na długie pokolenia populacja „zeszła” jednak z wyższego szczytu i rozpoczęła wędrówkę po zakątkach przestrzeni, osiągając w pokoleniu 5000. najlepsze przystosowanie rzędu 0,007. Sytuacja nie zmieniła się do 15 000 pokoleń. W dziesięciu przebiegach ewolucja wyglądała podobnie, zawsze pojawiały się osobniki na wysokim szczycie, choć nie w każdym przebiegu o równie wysokiej jakości, populacja ani razu nie zasiedliła wyższego szczytu.

Przy zmniejszonej częstości rekombinacji, np. do $P_{\text{rkm}} = 0,5$, populacja krąży po niższym szczycie przystosowawczym, w dziesięciu przebiegach ani razu nie znajdując optymalnego rozwiązania, ani też rozwiązania bliskiego optymalnemu. Dla tej samej częstości rekombinacji (0,5), ale przy zwiększonym M_{max} do 100, populacja natychmiast penetruje całą założoną przestrzeń fenotypową, znajdując dość dobre rozwiązanie już w szóstym pokoleniu (rys. 4.27). Populacja cały czas jest już obecna na wyższym szczycie, co widać z wykresu

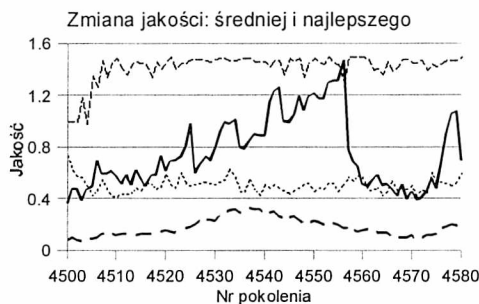


Rys. 4.26. Pokolenie 4556, funkcja Q_1 , $P_{\text{rkm}} = 1$, $P_{\text{mut}} = 0,2$, $M_{\text{max}} = 10$, pierwszy osobnik na wyższym szczycie (jaśniejsze obszary oznaczają szczyty)



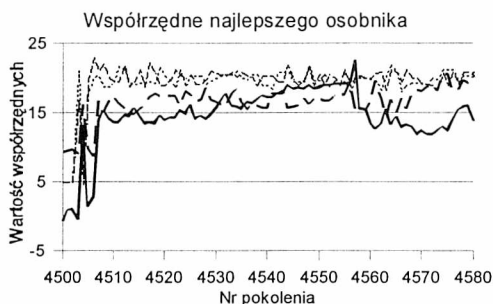
Rys. 4.27. Pokolenie 6., funkcja Q_1 , $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 0,2$, $M_{\text{max}} = 100$

jakości najlepszego osobnika (rys. 4.28)⁴. Cały czas utrzymuje się wokół wartości 1,4, w odróżnieniu od przebiegu przy częstszych rekombinacjach ($P_{rkm} = 1$) i nawet mniejszych jednorazowych zmian mutacyjnych. Patrząc na wartości współrzędnych najlepszego osobnika (rys. 4.29) można stwierdzić, że osobnik reprezentujący najlepsze rozwiązanie przy mniejszym P_{rkm} jest bliżej optimum globalnego (ma wartości bliższe 20).



— Q najlepszego (I) – pokolenia 4500-4580
 - - - Q średnie (I) – pokolenia 4500-4580
 - · - · Q najlepszego (II) – pokolenia od 1 do 80
 ····· Q średnie (II) – pokolenia od 1 do 80

Rys. 4.28. Przystosowanie najlepszego osobnika i średnie w populacji (dwa przebiegi): (I) – $P_{rkm} = 1$, $P_{mut} = 0,2$, $M_{max} = 10$, Q_1 , (II) – $P_{rkm} = 0,5$, $P_{mut} = 0,2$, $M_{max} = 100$, Q_1



— X1 (I) – pokolenia 4500-4580
 - - - X2 (I) – pokolenia 4500-4580
 - · - · X1 (II) – od pokolenia 1 do 80
 ····· X2 (II) – od pokolenia 1 do 80

Rys. 4.29. Współrzędne najlepszego osobnika w dwóch przebiegach: (I) – $P_{rkm} = 1$, $P_{mut} = 0,2$, $M_{max} = 10$, Q_1 , (II) – $P_{rkm} = 0,5$, $P_{mut} = 0,2$, $M_{max} = 100$, Q_1

Konkluzje

W świetle przeprowadzonych eksperymentów uzasadniony wydaje się wniosek, że w proponowanym modelu bardzo częste rekombinacje powodują duże różnicowanie się osobników w populacji i nawet po znalezieniu globalnego optimum populacja nie potrafi skupić się na nim. Rekombinacje mogą być stosowane jako czynnik wprowadzający większe zróżnicowanie w obrębie populacji, ale nie powinny być zbyt częste, bo wprowadzają „chaos” w ewolucji.

Charakter ewolucji obserwowanej w omówionych eksperymentach wskazuje, że rekombinacje nie są tym czynnikiem, który ułatwia populacji wspinaczkę po zboczu szczytu przystosowawczego.

4.2.2. MUTACJE JAKO CZYNNIK EWOLUCJI

Podczas analizy wpływu mutacji na przebieg ewolucji w proponowanym modelu należy brać pod uwagę zarówno częstość mutacji, jak i maksymalny zakres zmiany genu w wyniku pojedynczej mutacji.

⁴ Poszczególne przebiegi są narysowane dla różnych numerów pokoleń.

Eksperyment 5: *Ewolucja przy różnych częstościach mutacji, gdy maksymalna zmiana mutacyjna genu jest relatywnie mała.*

$P_{\text{rkm}} = 0,1$, $M_{\text{max}} = 10$, początkowa wartość genów: 1000 (wszystkie geny równe), $P_{\text{trs}} = 0$, $P_{\text{trz}} = 0$, $P_{\text{mig}} = 0$ (nie dotyczy, ewoluuje cała populacja, nie podzielona na demy). Macierz A składa się z zer, dodatnich i ujemnych elementów, dość jednorodna. Prawdopodobieństwo mutacji zmieniane od 0,2 do 1 ($P_{\text{mut}} = 0,2, 0,5$ oraz 1), funkcja Q_2 .

Dla wszystkich testowanych prawdopodobieństw mutacji, podobnie jak w eksperymencie pierwszym (**Eksperyment 1**), populacja jest mocno skupiona w okolicy niższego szczytu, który jest miejscem początkowym ewolucji. Przy najczęstszych mutacjach ($P_{\text{mut}} = 1$), po 2000 pokoleniach średnie przystosowanie populacji wynosi 0,987408, natomiast najlepszy osobnik w populacji ma współrzędne $x = [5,467528, 4,830633, 4,830633, 8,261687, 5,545798]$, jego przystosowanie jest równe 0,995287, współrzędne zaś najgorszego osobnika wynoszą $x = [5,523100, 4,025471, 4,112328, 7,036841, 6,668301]$, a jego przystosowanie wynosi 0,905968. Dane te (wzięte z jednego przebiegu, ale wszystkie symulacje były tu jakościowo takie same) jednoznacznie wskazują na to, iż populacja zatrzymuje się na lokalnym szczycie i nie jest w stanie, przynajmniej w akceptowalnym czasie, opuścić pułapki ewolucyjnej. W żadnym przebiegu symulacyjnym populacja nie znalazła satysfakcjonującego rozwiązania (bliskiego optymalnemu).

Eksperyment 6: *Ewolucja przy różnych częstościach mutacji i dziesięciokrotnie większej niż poprzednio maksymalnej zmianie mutacyjnej genu.*

Eksperyment jak poprzedni, zmianie ulega zakres mutacji, $M_{\text{max}} = 100$. Prawdopodobieństwo mutacji zmieniamy od 0,2 do 1, z krokiem 0,2. Zadajemy minimalne rekombinacje $P_{\text{rkm}} = 0,1$.

Dla $P_{\text{mut}} = 0,2$ populacja zajmuje większy obszar w krajobrazie przystosowawczym niż w poprzednim eksperymencie, nadal jest skupiona na niższym szczycie, średnia jakość populacji pozostaje na podobnym poziomie, niewiele się zmieniając.

Dla $P_{\text{mut}} = 0,6$ średnia wartość przystosowawcza populacji spada do około 0,9 – populacja jest nadal blisko lokalnego szczytu. Dla większych prawdopodobieństw najniższa średnia jakość w ciągu dwóch tysięcy pokoleń była powyżej 0,8. Można więc uznać, że efektywność ewolucji jest niezadowolająca, populacja nie rozprzestrzeniła się i nie rozpoczęła wędrówki po krajobrazie przystosowawczym. W tabeli 4.1 pokazano fenotypy i wartość przystosowawczą najlepszego i najgorszego osobnika w populacji, po ewolucji przez 2000 pokoleń, dla różnych częstości mutacji. Ostatnia kolumna zawiera średnią jakość w danym przebiegu.

Dziesięciokrotne powtarzanie wszystkich symulacji pozwala stwierdzić, że nawet częste mutacje, o maksymalnym zakresie zmiany genu równym sto, współpracujące z nieczęstymi rekombinacjami, nie są w stanie zapewnić efektywności ewolucji dla funkcji pięciu zmiennych.

Tabela 4.1. Ewolucja przy $M_{\max} = 100$ i różnych częstościach mutacji (pokolenie 2000)

P_{mut}	Najlepszy osobnik						Najgorszy osobnik						Q_{sr}
	x_1	x_2	x_3	x_4	x_5	Q_1	x_1	x_2	x_3	x_4	x_5	Q_1	
0,2	5,112	5,114	2,909	7,602	6,024	0,9997	4,620	2,681	4,786	5,214	5,734	0,8955	0,9555
0,4	5,049	4,709	14,86	11,04	6,948	0,9984	5,314	1,682	14,00	7,714	7,463	0,8009	0,9325
0,6	5,079	4,969	15,45	11,33	7,489	1,0000	5,012	1,314	12,78	5,789	7,312	0,7620	0,9030
0,8	5,017	4,835	10,54	15,86	5,931	0,9996	6,663	0,027	6,532	14,76	14,76	0,5770	0,8765
1,0	5,050	5,097	25,79	21,04	7,832	1,0000	3,244	0,723	24,35	13,01	6,358	0,6522	0,8576

Eksperyment 7: Ewolucja przy różnych częstościach mutacji, gdy maksymalna zmiana mutacyjna genu jest stosunkowo duża.

Zmiana prawdopodobieństwa mutacji od 0,2 do 1, z krokiem 0,2, przy dużej zmianie mutacyjnej ($M_{\max} = 500$). Reszta parametrów jak w eksperymencie poprzednim.

Wyniki dotyczące najlepszego i najgorszego osobnika w ciągu 2000 pokoleń są zebrane w tabeli 4.2. Przy tak dużej maksymalnej zmianie mutacyjnej $M_{\max}$ populacja jest w stanie znaleźć wysoki szczyt już dla częstości mutacji 60%. W przebiegu ujętym w tabeli najlepsze rozwiązanie znaleziono w 388. pokoleniu, jego jakość wynosiła 1,36, a w 406. pokoleniu średnia jakość populacji przekroczyła wysokość niskiego szczytu (wartość 1,1). Do 2000. pokolenia populacja utrzymała średnie przystosowanie na tym poziomie, ale nie zasiedliła wysokiego szczytu (jej jakość nie przekroczyła wartości 1,35). Podobnie przebiega ewolucja dla $P_{\text{mut}} = 0,8$ i 1,0. W tym ostatnim przypadku najlepsze rozwiązanie o jakości większej niż 1,35 pojawia się około setnego pokolenia.

Tabela 4.2. Ewolucja przy $M_{\max} = 500$ i różnych częstościach mutacji (pokolenie 2000.)

P_{mut}	Najlepszy osobnik						Najgorszy osobnik						Q_{sr}
	x_1	x_2	x_3	x_4	x_5	Q	x_1	x_2	x_3	x_4	x_5	Q	
0,2	5,37	5,54	5,88	6,07	5,74	1,0986	6,59	6,67	0,93	12,5	6,5	0,3697	0,8674
0,4	5,71	4,94	4,24	6,30	5,74	1,0706	5,11	11,32	4,6	14,57	4,78	0,3229	0,8156
0,6	18,69	21,71	18,22	22,34	18,34	1,451	17,4	5,46	19,18	10,83	16,39	0,737	1,1043
0,8	18,49	19,74	18,18	20,16	19,66	1,4807	20,57	8,32	21,52	4,28	23,29	0,6069	1,1210
1,0	19,63	18,71	21,12	21,20	19,14	1,4822	17,52	4,6	24,3	9,4	17,1	0,6372	1,2325

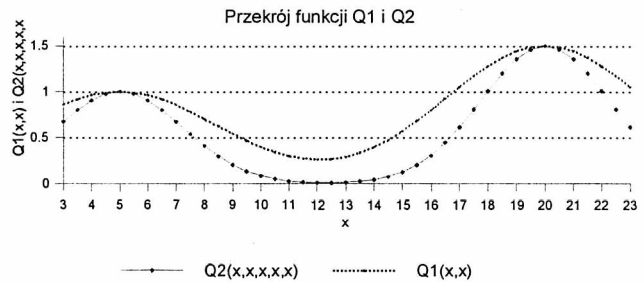
Eksperymenty te pokazują, że skupienie populacji jest większe przy bardzo częstych mutacjach niż przy bardzo częstych rekombinacjach. Przy dużym prawdopodobieństwie mutacji i stosunkowo dużych zmianach mutacyjnych można mówić o tym, że populacja zasiedla wysoki szczyt. Skupienie populacji nie jest zbyt duże, ale maksymalna wartość przystosowania jest dość wysoka. Z porównania omówionych eksperymentów widać, że zwiększanie częstości mutacji, podobnie jak jej maksymalnego zakresu, powoduje większe zróżnicowanie osobników w populacji, może umożliwić populacji osiągnięcie wysokiego szczytu kosztem niewysokiego średniego przystosowania.

Duże i częste mutacje wydają się być lepszym środkiem na znalezienie dobrego rozwiązania niż duże mutacje i częste rekombinacje. Bez odpowiednio dużego zakresu jednorazowej zmiany mutacyjnej ani częste rekombinacje, ani częste mutacje nie umożliwiają populacji opuszczenia pułapki ewolucyjnej.

Eksperyment 8: Sprawdzenie, czy same mutacje potrafią „wprowadzić” populację na szczyt, eksperymenty prowadzimy dla pięciowymiarowej i dwuwymiarowej funkcji.

Interesujących wyników można się spodziewać, jeśli populację początkową umieścimy w siodle, pomiędzy dwoma szczytami.

Zidentyfikujemy położenie siodła, pomoże w tym rysunek przekrojów przez szczyty wykorzystywanych funkcji (rys. 4.30). Przyjmijmy, że będzie to punkt o współrzędnych (12, 12, 12, 12, 12). Po zmodyfikowaniu P_{mut} i M_{max} przeprowadzamy dziesięciokrotnie testy dla tych samych warunków.



Rys. 4.30. Przekrój przez szczyty dwóch funkcji: dwu- i pięciowymiarowej

Funkcja $Q_2(x_1, x_2, x_3, x_4, x_5)$

Uzyskane wyniki zebrane są w tabeli 4.3. W badaniach interesowano się głównie czy, przy danych wartościach P_{mut} i M_{max} , populacja zdoła wejść na szczyt przystosowawczy i w jakim czasie. Oddzielnie notowano też, na który szczyt populacja zdołała zawędrować.

Tabela 4.3. Wspinaczka po zboczu dla $Q_2(x_1, x_2, x_3, x_4, x_5)$

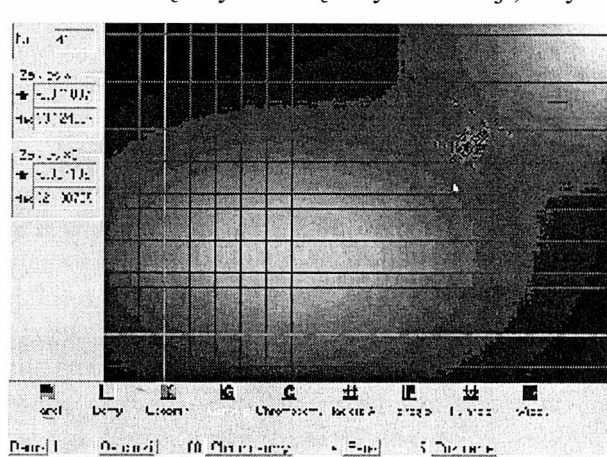
M_{max}	P_{mut}	Na dziesięć przebiegów populacja zasiedliła			
		niski szczyt		wysoki szczyt	
		liczba przebiegów	średni czas wejścia [pokolenia]	liczba przebiegów	średni czas wejścia [pokolenia]
10	0,2	4	5104	6	3900*
10	0,8	9	1238	1	861*
100	0,2	7	115	3	149
100	0,4	4	90	6	75
100	0,8	2	44	8	70
500	0,2	10	12	0	—
500	0,8	9	8	1	21

* oznacza, że w tym przebiegu populacja zasiedliła wyższy szczyt, jakoś najlepszego osobnika przekroczyła 0,9

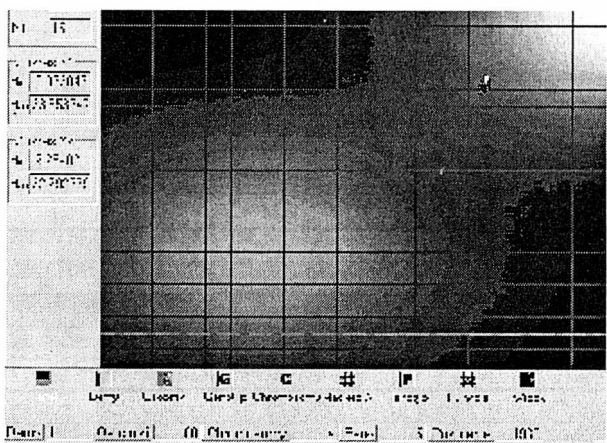
Podsumowując można stwierdzić, że w proponowanym modelu, podobnie jak w klasycznym algorytmie genetycznym, mutacje są wystarczającym czynnikiem ewolucji, umożliwiającym populacji wspinaczkę po zboczu szczytu przystosowawczego.

Od rozmiaru mutacji zależy tempo wchodzenia na szczyt. Im większy zakres maksymalnej zmiany mutacyjnej, tym szybciej populacja wspina się na szczyt. Skutkiem ubocznym tego może być obniżenie średniej jakości populacji – populacja bardziej zróżnicowana zajmuje większy obszar w krajobrazie przystosowawczym.

Natomiast to, który ze szczytów zasiedli populacja jest w dużej mierze przypadkiem. Dla większych i częstszych mutacji, zwykle wyższy szczyt jest osiągany. Dla



Rys. 4.31. $P_{mut} = 0,2$, $M_{max} = 500$, start: [12,12,12,12,12], populacja po pierwszym pokoleniu (najlepszy osobnik jest położony na zboczu niższego szczytu)

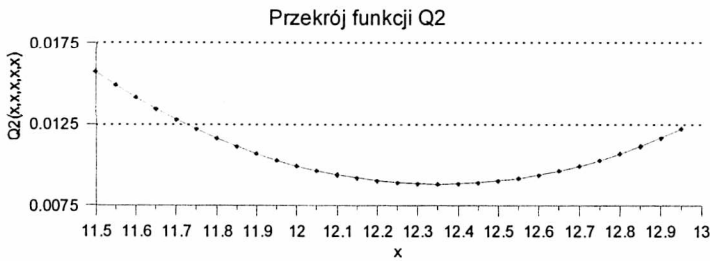


Rys. 4.32. Ewolucja populacji przy $P_{mut} = 0,2$ i $M_{max} = 10$, pokolenie 1930: populacja wspina się na wyższy szczyt

małych częstości mutacji i małego jej zakresu, mutacje w pierwszych pokoleniach są decydujące. Jeśli populacja rozpocznie już wędrówkę w kierunku niższego szczytu, trudno jej będzie odwrócić swój bieg. W początkowej fazie ewolucji, najlepszy osobnik w populacji może znajdować się na zmianę raz na zboczu niskiego, raz wysokiego szczytu (rys. 4.31 i 4.34). Po pierwszym pokoleniu, najlepszy osobnik wyraźnie zajmuje miejsce na zboczu niższego szczytu, po czym, po kilkukrotnych zmianach, już w 11. pokoleniu populacja jest na najwyższym szczycie (jakość powyżej 1,35). Od tego czasu, najlepsze osobniki „przeciągają” populację na ten szczyt. Rozproszona populacja szybko zasiedla nową niszę, w odróżnieniu od ewolucji z małymi zmianami mutacyjnymi, gdzie populacja wędruje zwartą grupą (rys. 4.32).

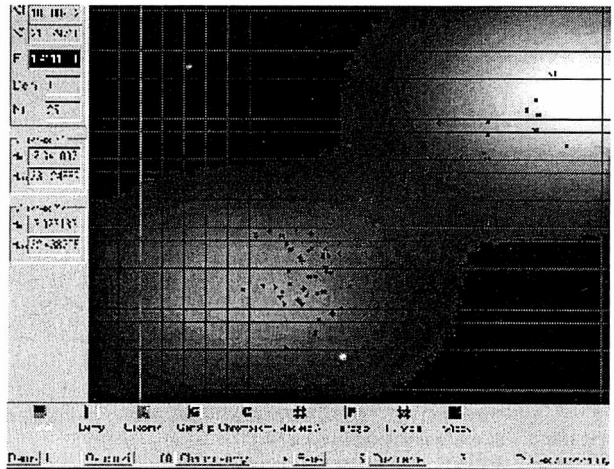
Sprawdzenie, czy rzeczywiste wybory kierunku ewolucji jest czysto losowy wymagałoby wykonania dużo więcej prób. Należałoby również dokładnie ustalić punkt początkowy, ponieważ ten

stosowany w prezentowanym eksperymencie jest odrobinę na zboczu niższego szczytu (rys. 4.33).



Rys. 4.33. Fragment przekroju funkcji Q_2 w okolicy siódła

Rys. 4.34. $P_{\text{mut}} = 0,2$, $M_{\text{max}} = 500$, start (12,12,12,12,12), rozwiązanie o jakości większej niż 1,35 w pokoleniu 11., w pokoleniu 19. widać, jak populacja „przeprowadza się” na wyższy szczyt



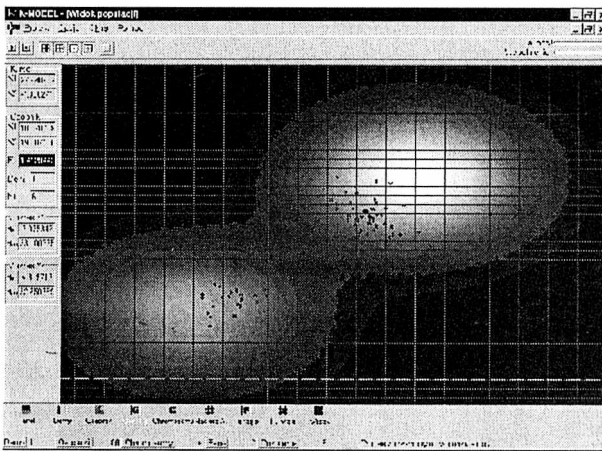
Funkcja dwuwymiarowa $Q_1(x_1, x_2)$

Populacja początkowa jest umieszczona w dolinie pomiędzy dwoma szczytami: punkt o współrzędnych (12,12) (najniższa wartość Q_1 wzdłuż przekroju przez szczyty jest w p. (12.1, 12.1), z dokładnością do piątego miejsca po przecinku). Przeprowadzamy symulację takie, jak dla funkcji Q_2 .

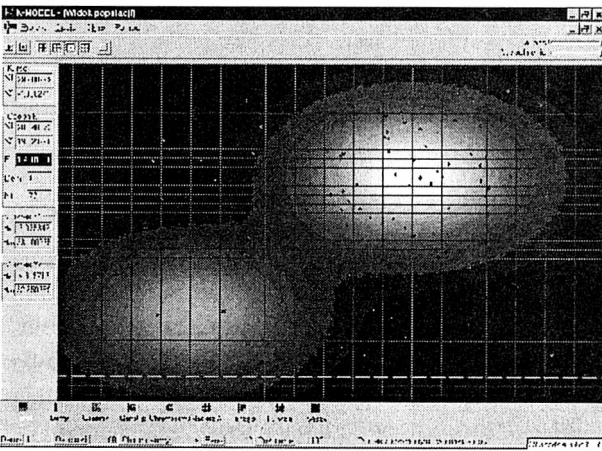
Dla małych maksymalnych zmian mutacyjnych genu ($M_{\text{max}} = 10$) populacja bardzo dobrze radziła sobie z zasiedleniem wysokiego szczytu przystosowawczego (tab. 4.4). Zarówno dla $P_{\text{mut}} = 0,2$ jak i $P_{\text{mut}} = 0,8$ zdarzało się, że w początkowych pokoleniach, kiedy pojawiał się osobnik na wyższym szczycie, duża część populacji była jeszcze na niskim szczycie (rys. 4.35). Po kilkunastu pokoleniach jednak populacja w całości była na lepszym szczycie. Jeśli natomiast populacja rozpoczęła wędrówkę głównie w kierunku niższego szczytu, zasiedlała niski szczyt i po dalszych 200 pokoleniach nic nie wskazywało na to, by mogła ten szczyt opuścić.

Tabela 4.4. Wspinaczka po zboczach szczytu przystosowawczego dla $Q_1(x_1, x_2)$

$M_{\max}$	P_{mut}	Na dziesięć przebiegów populacja zasiedliła			
		niski szczyt		wysoki szczyt	
		liczba przebiegów	średni czas wejścia [pokolenia]	liczba przebiegów	średni czas wejścia [pokolenia]
10	0,2	1	19	9	18
10	0,8	1	21	9	8
100	0,2	Już w drugim pokoleniu pojawiały się osobniki na wysokim szczycie (jakość > 1,35), ale populacja nie była w stanie skupić się na tym szczycie. Zajmowała całą przestrzeń fenotypową (rys. 4.36)			
100	0,4				
100	0,8				
500	0,2				
500	0,8				



Rys. 4.35. Najlepszy osobnik na wyższym szczycie, ale część populacji jest na niskim szczycie ($M_{\max} = 10, P_{\text{mut}} = 0,2$)



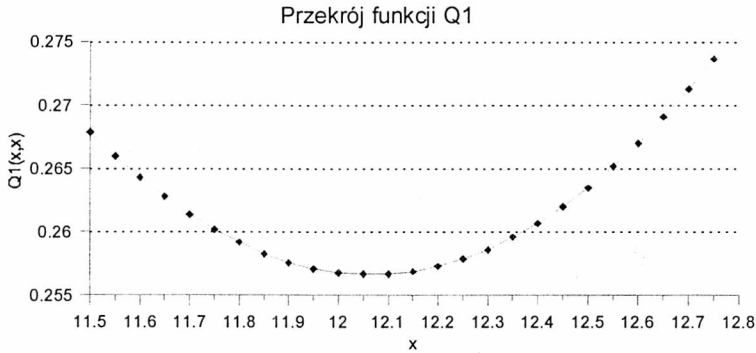
Rys. 4.36. Populacja po 100 pokoleniach, $M_{\max} = 100, P_{\text{mut}} = 0,2$ (pierwszy osobnik na wysokim szczycie był w 7. pokoleniu)

Dla $M_{\max} = 100$ lub więcej, nawet przy niskim prawdopodobieństwie mutacji, populacja nie była w stanie zasiedlić jednego szczytu. Można się domyślać, że duże mutacje niszczyły dobre rozwiązania, powodując pojawianie się osobników w odległych miejscach.

Porównując wyniki zestawione w tabelach 4.3 i 4.4, widzimy bardzo duże różnice zarówno w tempie, charakterze jak i kierunku ewolucji.

Dużo szybsze tempo ewolucji (przy tych samych operatorach genetycznych i ich parametrach) dla funkcji Q_1 wynika z mniejszego wymiaru tej funkcji. Problemy wielowymiarowe są znacznie trudniejsze od jedno- czy dwuwymiarowych. Powstanie lepszego osobnika wymaga jednoczesnej pozytywnej zmiany kilku fenów, co jest mniej prawdopodobne niż pozytywna zmiana pojedynczego fenu.

Charakter ewolucji – skupienie lub nie, penetracja całej przestrzeni czy powolne przesuwanie

Rys. 4.37. Fragment przekroju Q_1 w okolicy siodła

nie się zwartą grupą, wynika z tego samego faktu: przy niewielkiej liczbie fenów łatwiej o zmiany mutacyjne dające osobnika istotnie różnego od rodzica. Natomiast fakt, że przy funkcji dwuwymiarowej znacznie rzadziej populacja wędruje na niski szczyt można wytłumaczyć różnym nachyleniem funkcji Q_1 i Q_2 w otoczeniu punktu położenia początkowej populacji.

Przy funkcji Q_1 zmiana fenu o wartość Δx daje większy przyrost przystosowania, jeśli jest to zmiana w kierunku wyższego szczytu, natomiast przy funkcji Q_2 jest odwrotnie (początkowa populacja położona jest lekko na zboczu niskiego szczytu), co wyraźnie widać na rysunkach 4.33 i 4.37.

Eksperyment 9: Sprawdzenie, czy rekombinacje i mutacje mają porównywalną moc w prowadzeniu populacji po zboczu wzgórza przystosowawczego.

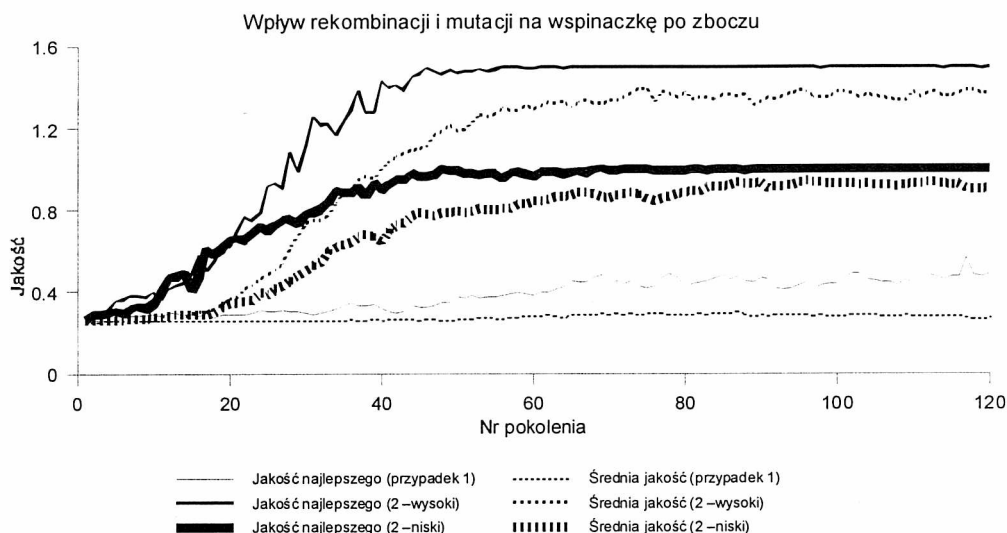
Populacja zaczyna ewolucję w dolinie przystosowawczej – punkt o współrzędnych (12, 12, 12, 12, 12). Wykonano wiele symulacji, ze względu na pokazanie istotnej różnicy wystarczy przytoczyć tu tylko wyniki uzyskane w typowym przebiegu dla przypadku:

(1) $P_{rek} = 1$ i $P_{mut} = 0,2$,

(2) $P_{rek} = 0,2$ i $P_{mut} = 1$.

Zmiany średniej jakości populacji, a zwłaszcza jakości najlepszego osobnika w populacji, informują o różnym charakterze ewolucji w przypadku (1) i (2).

Na wykresie na rysunku 4.38 widoczne są duże wahania jakości najlepszego osobnika w populacji i relatywnie duże wahania średniej jakości populacji przy częstych rekombinacjach (przypadek 1). Populacja krążyła trochę „na oślep” po krajobrazie przystosowawczym. Gdy rekombinacje były niewielkie, a mutacje nawet 100%, populacja poruszała się mniej chaotycznie, przesuując się w kierunku jednego ze szczytów, po jego zasiedleniu nie zdarzało się, by z niego zeszła, co było obserwowane w przypadku rekombinacji.



Rys. 4.38. Zmiana średniej jakości populacji oraz jakości najlepszego osobnika w populacji, w 120 pierwszych pokoleniach ewolucji: 2 – wysoki, oznacza przypadek (2), przy czym populacja weszła na wysoki szczyt, 2 – niski – populacja weszła na niższy szczyt

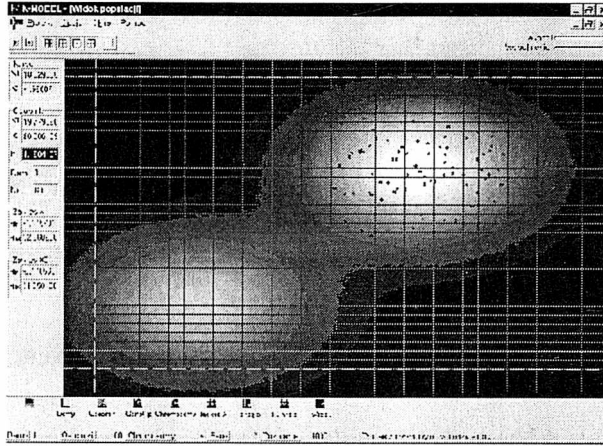
Eksperyment 10: Weryfikacja obserwacji z eksperymentów 5–9 dla dwuwymiarowej funkcji $Q_1(x_1, x_2)$.

Przy wartościach parametrów mutacji jak w eksperymencie 5 ($M_{\max} = 10$, $P_{\text{mut}} = 0,2$) populacja ewoluuje na niskim szczycie. Nie jest tak mocno skupiona, penetruje cały obszar niskiego szczytu, ale nie udało się jej przejść na wysoki szczyt. Zwiększanie prawdopodobieństwa mutacji powoduje większą różnorodność populacji, ale nadal nie radzi sobie ona z zasiedleniem wysokiego szczytu. Dopiero przyjęcie $P_{\text{mut}} = 1$ daje w efekcie szansę na przejście populacji na wysoki szczyt. Średnio populacja potrzebuje około 500–700 pokoleń na znalezienie pierwszego rozwiązania na tym szczycie, następnie w ciągu około 200 pokoleń szczyt ten zostaje zasiedlony. Populacja jest jednak dość rozproszona po okolicy tego szczytu (rys. 4.39). Przy mniejszych P_{mut} , np. $P_{\text{mut}} = 0,8$, w niektórych przebiegach pojawiają się pojedyncze osobniki (1000–2000 pokoleń) o przystosowaniu większym od 1,35, ale są one gubione i populacja nie przechodzi na optymalny obszar przestrzeni fenotypowej.

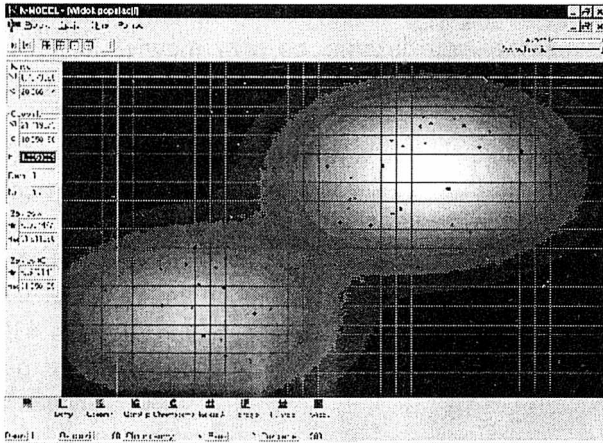
Po zwiększeniu $M_{\max}$ do 100, sytuacja się zmienia. Przy $P_{\text{mut}} = 0,2$ zazwyczaj pierwszy osobnik pojawia się na wysokim szczycie w pierwszym tysiącu pokoleń (średnio w 720 pokoleń). Populacja jest jednak bardzo różnorodna, rozwiązania te są gubione. Musi upłynąć kilkadziesiąt pokoleń, by populacji udało się zasiedlić globalne optimum, ale nie dzieje się to zawsze. Czasami jest to pokolenie 3300, a czasami 5000 to za krótki okres.

Jeśli populacja znajduje się na wysokim szczycie, jej średnia jakość jest wyższa od jedności (poziom niższego szczytu). Zwiększenie P_{mut} do 1 powoduje, że najlepsze

rozwiązanie jest znalezione już w pierwszym pokoleniu, ale populacja nie potrafi skupić się na tym szczycie. Nawet po upływie 2000 pokoleń jest rozproszona po całej przestrzeni (rys. 4.40).



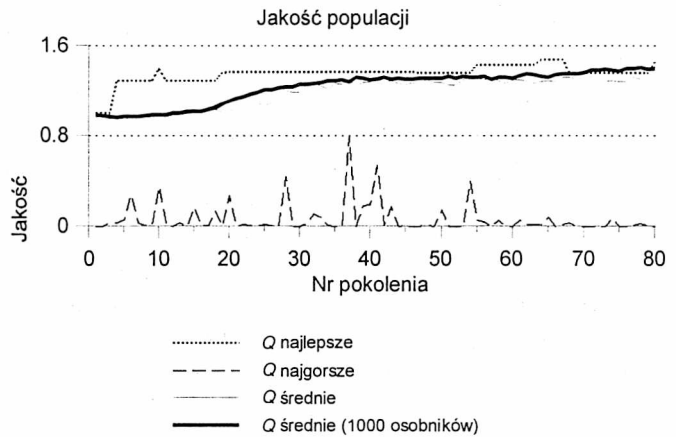
Rys. 4.39. Populacja po 1000 pokoleniach ($M_{\max} = 10$, $P_{\text{mut}} = 1$)



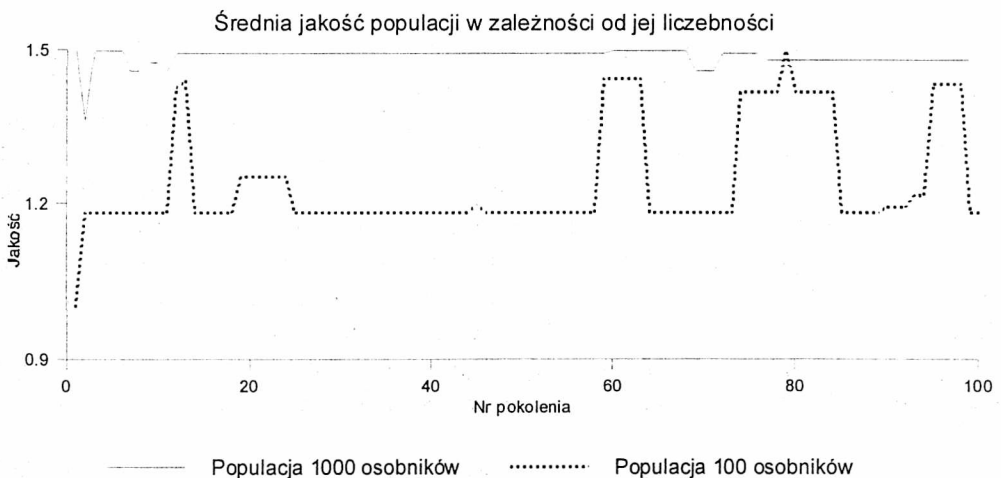
Rys. 4.40. Populacja po 2000 pokoleniach, $M_{\max} = 100$, $P_{\text{mut}} = 1$

Przy $M_{\max} = 500$ i $P_{\text{mut}} = 0,2$ populacja w ciągu kilku pokoleń (od 2 do 6) znajduje optymalne rozwiązanie (o jakości powyżej 1,35), ale nie potrafi skupić się na tym szczycie. Jest dość mocno rozproszona po całej przestrzeni, co powoduje, że jej średnia jakość jest stosunkowo niska. Okazuje się jednak, że populacja ma tu wyższe średnie przystosowanie niż przy wartościach parametrów $M_{\max} = 100$, $P_{\text{mut}} = 1$, co wyraźnie widać, jeśli spojrzymy na rysunek 4.42. Zwiększanie prawdopodobieństwa mutacji powoduje większe obniżenie średniej jakości populacji niż zwiększanie maksymalnej

od optimum, ale skupienie się populacji zaledwie w kilku punktach (chwilami w trzech) powoduje taki efekt, że średnia jakość jest stosunkowo wysoka. Czy taki wynik używa się również przy bardziej licznych populacjach? Zwiększamy więc liczebność populacji do 1000 i obserwujemy wyniki, powtarzając je kilkakrotnie: średnia jakość populacji jest porównywalnie wysoka, jak dla populacji stuelementowej. Przy tak licznej populacji najlepsze rozwiązanie jest bardzo wysokie, zazwyczaj ponad 1,48, często powyżej 1,49. Porównanie najlepszych rozwiązań znalezionych w ciągu pierwszych 100 pokoleń przez ewoluujące populacje liczące 100 i 1000 elementów pokazano na rysunku 4.44. Widać więc, że zwiększenie liczebności populacji przy dużych i częstych mutacjach pozwala na znalezienie dokładniejszego rozwiązania.



Rys. 4.43. Średnia jakość populacji na tle najlepszego i najgorszego osobnika w populacji ($M_{\max} = 500$, $P_{\text{mut}} = 1$)



Rys. 4.44. Zmiana jakości najlepszego osobnika w populacji, w ciągu 100 pokoleń ewolucji dla $M_{\max} = 500$, $P_{\text{mut}} = 1$ oraz różnej liczebności populacji

Należy nadmienić, że populacja licząca 1000 elementów nie radzi sobie ze znalezieniem wyższego szczytu, jeśli ewoluuje w pięciowymiarowej przestrzeni – dla funkcji $Q_2(x_1, x_2, x_3, x_4, x_5)$, nawet przy $M_{\max} = 500$ i $P_{\text{mut}} = 1$.

4.3. MAKROMUTACJE W POSZUKIWANIU OPTIMALNEGO ROZWIĄZANIA

K-model umożliwia modelowanie makromutacji na trzy sposoby:

1. Kumulowanie się zmian w genach nadmiarowych i zamiana miejscami genu nadmiarowego z genem fenotypowym (*nadmiarowość*).

2. Radykalne zwiększenie prawdopodobieństwa mutacji i (lub) rekombinacji, transycji, transpozycji oraz migracji dla zadanej parametrem rcd części populacji, zachodzi w każdym pokoleniu u losowo wybranych osobników (*rekrudescencja*),

3. Radykalne zwiększenie prawdopodobieństwa mutacji i (lub) rekombinacji, transycji, transpozycji oraz migracji dla wszystkich osobników w populacji, z częstością t_{kr} , liczoną w pokoleniach – zjawisko to nazwane jest *kryzysem*, ponieważ może spowodować bardzo niekorzystne zmiany w całej populacji.

W kolejnych podrozdziałach pokrótce przedstawiono eksperymenty pokazujące rolę poszczególnych rodzajów makromutacji oraz ich kombinacji. Systematyczne badania wykonano dla pięciowymiarowego krajobrazu przystosowawczego – funkcja Q_2 .

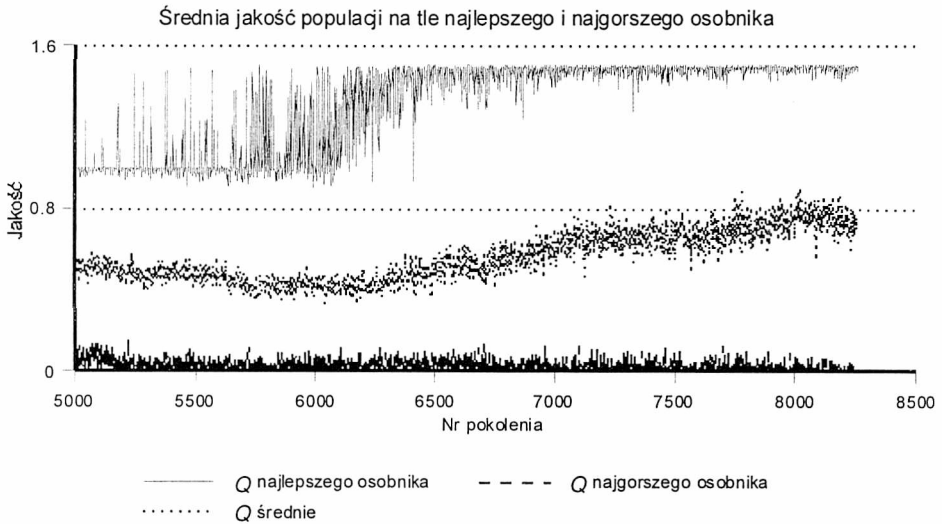
4.3.1. GENY NADMIAROWE

Dla maksymalnej zmiany mutacyjnej genu równej 10 żadne wartości parametrów P_{rkm} , P_{mut} , P_{trz} , P_{trs} nie umożliwiają populacji znalezienia optimum. Po pięciu tysiącach pokoleń nic nie wskazywało na taką możliwość, gdyby ewolucja była kontynuowana.

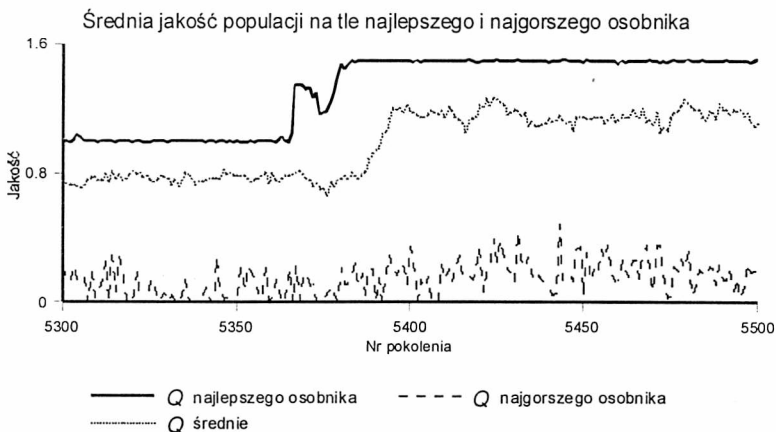
Zwiększono $M_{\max}$ do 100. Dla $P_{\text{rkm}} = 0,25$ i $P_{\text{mut}} = 0,2$ zmieniano P_{trz} i P_{trs} od 0,1 do 1, populacja nie potrafiła znaleźć wyższego szczytu. Dopiero dla wartości $P_{\text{rkm}} = 0,5$ i $P_{\text{mut}} = 1$, P_{trz} i P_{trs} równych 0,2, pierwsze rozwiązanie o jakości powyżej 1,35 (czyli na wysokim szczycie) jest znajdowane w ciągu około 3300 pokoleń. Ponieważ zarówno częstość mutacji, jak i rekombinacji jest stosunkowo duża, populacja łatwo gubi dobre rozwiązania. Niemniej jest jednak w stanie, po odpowiednio długim czasie (z reguły między piątym a ósmym tysiącem pokoleń), zasiedlić wyższy szczyt. Populacja jest bardzo rozproszona, co powoduje niskie jej średnie przystosowanie (rys. 4.45).

Bez istnienia genów nadmiarowych, przy $M_{\max} = 100$ i dowolnych wartościach prawdopodobieństw mutacji i rekombinacji, populacja nie była w stanie znaleźć optymalnego rozwiązania. Z genami nadmiarowymi i przy częstych mutacjach i re-

kombinacjach, populacja jest w stanie znaleźć satysfakcjonujące rozwiązanie, zajmuje jej to jednak zbyt dużo czasu, zbieżność populacji też wydaje się mało zadowalająca. Jak wiadomo, algorytmy ewolucyjne muszą znaleźć równowagę pomiędzy poszukiwaniem nowych (lepszyc) rozwiązań z jednej strony a eksploatacją odkrytych tzw. obiecujących rozwiązań z drugiej. Nowe rozwiązania mogą leżeć na zboczach szczytów przystosowawczych i dobrze byłoby, gdyby algorytm dość szybko sprawdzał, czy dany szczyt jest optymalny.



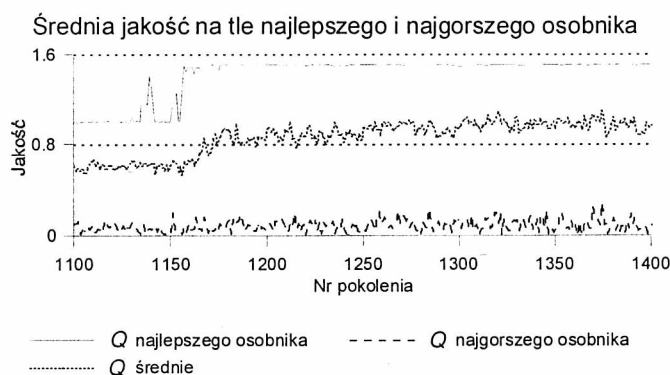
Rys. 4.45. Średnia jakość populacji oraz jakość najlepszego i najgorszego osobnika, $M_{\max} = 100$, $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 1$, $P_{\text{trz}} = P_{\text{trs}} = 0,2$



Rys. 4.46. Średnia jakość populacji oraz jakość najlepszego i najgorszego osobnika, $M_{\max} = 500$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trz}} = P_{\text{trs}} = 0,01$

Przy $M_{\max} = 500$ już dla $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trs}} = P_{\text{trz}} = 0,01$ populacja znajduje rozwiązanie na wysokim szczycie w czasie 4500–5500 pokoleń. W tym przypadku obserwujemy większą zbieżność populacji, średnia jakość populacji jest wyższa niż poprzednio (por. rys. 4.45 i 4.46). Zwiększanie prawdopodobieństw tranzycji i transpozycji powoduje, że najlepsze rozwiązanie znajdowane jest szybciej, ale średnia jakość populacji jest niższa. Dla $P_{\text{trs}} = P_{\text{trz}} = 0,05$, optymalny osobnik pojawia się często przed upływem 1500 pokoleń, zwykle nie później niż w 2000., cała populacja przechodzi na wysoki szczyt po czasie dłuższym niż 5000 pokoleń (rzadko zdarza się wcześniej). Przy $P_{\text{trs}} = P_{\text{trz}} = 0,2$ czas pojawienia się osobnika na globalnym optimum skraca się do około 200–600 pokoleń, a przejście całej populacji na ten szczyt zajmuje od 1500 do 3000 pokoleń.

Można powiedzieć z całą pewnością, że nadmiarowość w reprezentacji osobników pomaga znaleźć najlepsze rozwiązanie. Zbyt częsta zamiana genu nadmiarowego z fenotypowym (duże P_{trs}) powoduje niepożądane efekty: zbyt duży brak zbieżności populacji, co powoduje, że populacja nie może do końca wykorzystać wykrytych nisz ekologicznych – będąc nawet w okolicy najlepszego szczytu populacja nie potrafi wspiąć się na niego, by dać rozwiązanie możliwie bliskie samemu optimum.



Rys. 4.47. Średnia jakość populacji oraz jakość najlepszego i najgorszego osobnika, $M_{\max} = 500$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$

Podobne testy prowadzono dla funkcji dwuwymiarowej Q_1 . Funkcja dwuwymiarowa jest znacznie łatwiejsza dla algorytmów ewolucyjnych niż pięciowymiarowa. Tylko z rekombinacjami i mutacjami, dla $M_{\max} = 10$, w ciągu 1000 pokoleń nie jest znajdowany wysoki szczyt ($P_{\text{rkm}} = 0,25$; $P_{\text{mut}} = 0,2$). Dodanie nadmiarowości ($P_{\text{trs}} = P_{\text{trz}} = 0,05$) powoduje, że w każdym z 10 przebiegów populacja znalazła rozwiązanie na wysokim szczycie, średnio w 674 pokoleniach (rozzrzt od 439 do 824).

4.3.2. REKRUEDESCENCJA

– RADYKALNE ZMIANY U CZĘŚCI POPULACJI

W eksperymentach prowadzonych dla Q_2 i ustalonych trzech wartości $M_{\max} = 10$, 100 oraz 500, zmieniano procent populacji poddanej reorganizacji od 10 do 50, praw-

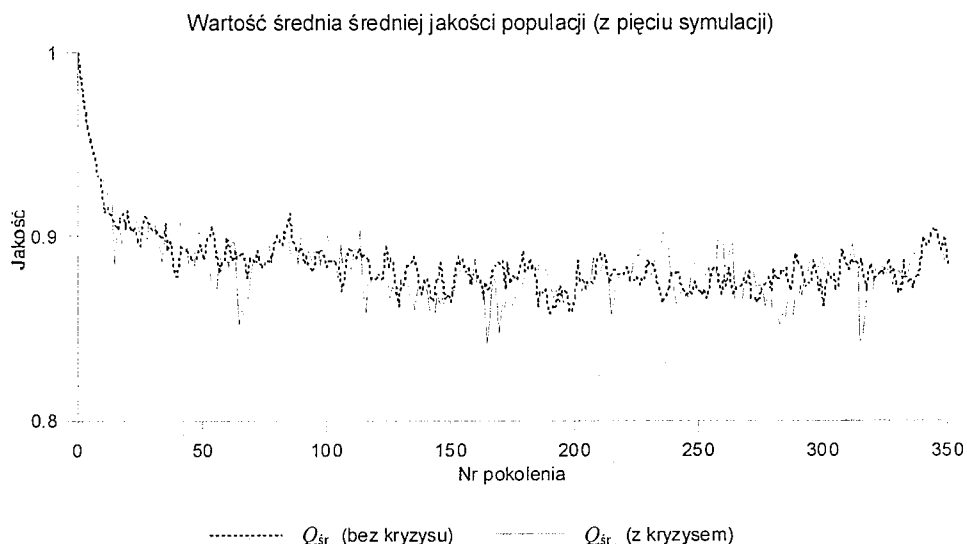
dopodobięństwa rekombinacji i mutacji dla części populacji podlegającej rekrudescencji zwiększano na 0,5, 0,8 oraz 1. Wraz ze wzrostem wartości tych operatorów populacja stawała się bardziej zróżnicowana, ale w żadnym przypadku nie uzyskano rozwiązania na wysokim szczycie (symulacje trwały 5000 pokoleń).

Sama rekrudescencja nie jest czynnikiem, który istotnie przyspieszałby ewolucję. Ta obserwacja potwierdza się również dla funkcji dwywymiarowej. Rolą rekrudescencji może być zapobieganie przedwczesnej zbieżności populacji.

4.3.3. KRYZYS

– REORGANIZACJA WSZYSTKICH OSOBNIKÓW

Podobne eksperymenty przeprowadzono, dodając do podstawowych operatorów mutacji ($M_{\max} = 100$) i rekombinacji czynnik nazwany kryzysem, powodujący reorganizację genotypów wszystkich osobników w populacji. Reorganizacja wszystkich osobników w populacji następowała co 20 i co 50 pokoleń (P_{rkm} i P_{mut} zmieniano na 0,5, 0,8 i 1). W żadnym wypadku populacja nie znalazła optymalnego rozwiązania. Taka gwałtowna zmiana wszystkich osobników powoduje zazwyczaj obniżenie się średniej jakości populacji, niemniej wśród osobników może pojawić się taki, który, co prawda będzie słabo przystosowany, ale będzie położony w okolicy wysokiego szczytu i jego potomstwo może stać się najlepsze. Na wykresie (rys. 4.48) pokazano zmiany średniej jakości – liczone jako wartości średnie z pięciu przebiegów – populacji ewoluującej z kryzysem co 50 pokoleń oraz bez kryzysu (pozostałe parametry są te same).

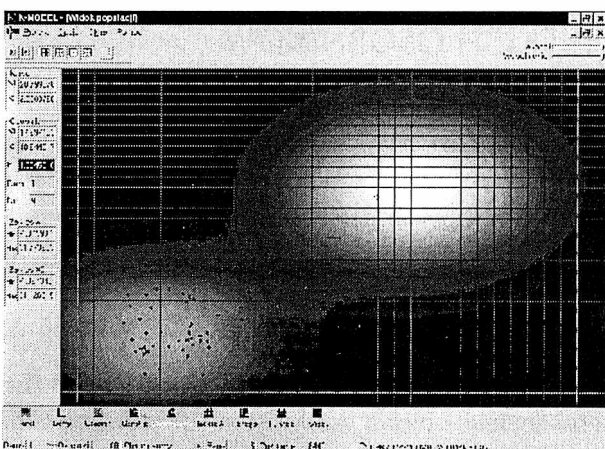


Rys. 4.48. Wartość średnia średniej jakości populacji z pięciu przebiegów z istnieniem kryzysu i bez; $M_{\max} = 100$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trz}} = P_{\text{trs}} = 0$, $t_{\text{kr}} = 20$ (w kryzysie: $P_{\text{rkm}} = 1$, $P_{\text{mut}} = 1$)

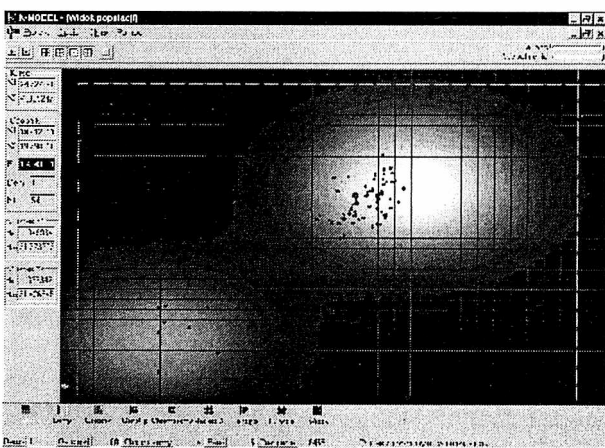
Dla pięciowymiarowej funkcji, przy stosowaniu ustalonych w powyższych eksperymentach parametrów operatorów genetycznych, żaden model makromutacji w pojedynkę nie daje zadowalających wyników.

4.3.4. ŁĄCZNY EFEKT NADMIAROWOŚCI, REKRUDESCENCJI I KRYZYSU

Eksperymenty prowadzono dla pięciowymiarowej funkcji Q_2 . Dla maksymalnej zmiany genu $M_{\max} = 10$ nie udało się w ciągu 10 000 pokoleń uzyskać rozwiązania



Rys. 4.49. Pokolenie 8440, $M_{\max} = 200$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trz}} = P_{\text{trs}} = 0,1$, $\text{rcd} = 5\%$, $t_{\text{kr}} = 20$ (dla rekrudescencji i kryzysu: $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 1$, $P_{\text{trz}} = P_{\text{trs}} = 0,2$), pierwszy osobnik o jakości $>1,35$



Rys. 4.50. Pokolenie 8455, $M_{\max} = 200$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trz}} = P_{\text{trs}} = 0,1$, $\text{rcd} = 5\%$, $t_{\text{kr}} = 20$ (dla rekrudescencji i kryzysu: $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 1$, $P_{\text{trz}} = P_{\text{trs}} = 0,2$), średnia jakość $>1,1$

leżącego na wysokim szczycie (o jakości większej niż 1,35). Dla $M_{\max} = 100$, mimo dość dużej różnorodności populacji, również w ciągu nawet 20 000 pokoleń nie udało się znaleźć optymalnego rozwiązania. Wydaje się, że po dostatecznie długim czasie byłoby to możliwe, ale taka skuteczność nie jest zadowalająca.

Uruchomiono program ustalając $M_{\max} = 200$ z typowymi rekombinacjami i mutacjami, rekrudescencją i kryzysem: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trz}} = P_{\text{trs}} = 0,1$, $\text{rcd} = 5\%$, $t_{\text{kr}} = 20$ (dla rekrudescencji i rekrudescencji: $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 1$, $P_{\text{trz}} = P_{\text{trs}} = 0,2$). Przy tych parametrach, pierwszy osobnik o jakości większej niż 1,35, a więc na wysokim szczycie, często pojawiał się przed upływem 10 000 pokoleń (rys. 4.49). Co jest charakterystyczne, populacja potrzebuje niewiele czasu, by zasiedlić wysoki szczyt i uzyskać średnią jakość większą niż 1,1 – czyli większą niż wysokość niskiego szczytu (rys. 4.50). Jak pokazuje wykres średniej jakości populacji (rys. 4.51), nie jest ona stosunkowo wysoka, i co jest typowe dla przebiegu z włączy-

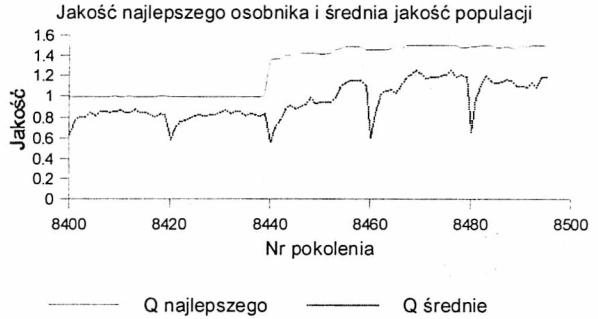
nym operatorem kryzysu, obserwuje się jej gwałtowne spadki. Jest to związane z okresami wystąpienia kryzysu. Podobne, na ogół krótkotrwałe załamania średniej jakości populacji są też widoczne na następnym wykresie (rys. 4.52) (dla $M_{\max} = 500$).

W miarę wydłużania okresu ewolucji następuje stopniowe zmniejszanie się spadków średniej jakości. Wynika to raczej z charakteru funkcji Q_2 – ma ona dość łagodny, szeroki szczyt.

Obserwacja ewolucji populacji dla dwuwymiarowej funkcji Q_1 , gdzie stosowano połączone modele makromutacji: nadmiarowość z rekrudescencją i nadmiarowość z kryzysem, prowadzi do wniosku, że bardziej efektywna jest modyfikacja całej populacji co pewien czas niż zmiana w każdym pokoleniu niewielkiej części populacji. Średni czas oczekiwania na rozwiązanie o jakości powyżej 1,35 (na wysokim szczycie) skrócił się z 674 (istnienie nadmiarowości, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$) do 501 po włączeniu kryzysu co 20 pokoleń ($t_{\text{kr}} = 20$, $P_{\text{rkm}} = P_{\text{mut}} = 0,5$ w kryzysie, reszta parametrów bez zmian).

Wyraźnego przyspieszenia ewolucji nie obserwowano, gdy zamiast kryzysu włączono rekrudescencję dla 5% osobników w każdym pokoleniu. Średnio, dla $rcd = 5\%$ i 100-elementowej populacji, w ciągu 20 pokoleń rekrudescencji podlegało 100 osobników – czyli tyle samo, co przy kryzysie – cała stu-elementowa populacja. Średnia jakość populacji jest tu bardzo niska – w żadnym przebiegu nie przekroczyła wartości 1,1, populacja jest rozrzucona po przestrzeni fenotypowej.

Zmiana t_{rkm} na 10, następnie na 50 nie przyniosła widocznych wyników. Ewolucja z częstym kryzysem powoduje większy rozrzut populacji, ale nie ma to zauważalnego, przyspieszającego wpływu na czas znajdowania rozwiązania. Po znalezieniu optymalnego rozwiązania, w żadnym z tych przypadków populacja nie skupiła się na wysokim szczycie, nie osiągnęła średniego przystosowania większego niż 1,1.



Rys. 4.51. Średnia jakość populacji na tle najlepszego osobnika: $M_{\max} = 200$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trz}} = P_{\text{trs}} = 0,1$, $rcd = 5\%$, $t_{\text{kr}} = 20$ (dla rekrudescencji i kryzysu: $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 1$, $P_{\text{trz}} = P_{\text{trs}} = 0,2$)



Rys. 4.52. Średnia jakość populacji na tle najlepszego osobnika: $M_{\max} = 500$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trz}} = P_{\text{trs}} = 0,05$, $rcd = 5\%$, $t_{\text{kr}} = 20$ (dla rekrudescencji i kryzysu: $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 0,5$, $P_{\text{trz}} = P_{\text{trs}} = 0,1$)

4.4. LICZEBNOŚĆ POPULACJI I EWOLUCJA W K-MODELU

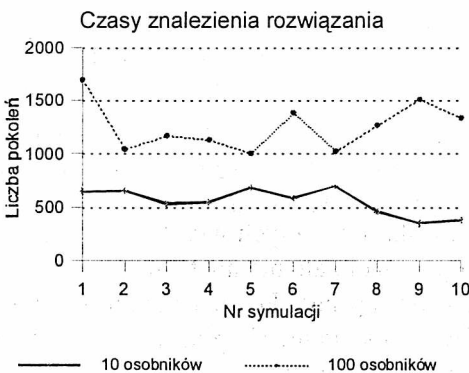
Na drodze wstępnych eksperymentów ustalono, że dla funkcji pięciowymiarowej Q_2 możliwe jest znalezienie wysokiego szczytu przy następujących parametrach: $M_{\max} = 600$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$. Te parametry przyjęto do badań wpływu rozmiaru populacji na szybkość i charakter ewolucji. Czas symulacji ustalono na 2000 pokoleń, badania prowadzono dla trzech różnych rozmiarów populacji: 1000, 100 i 10 elementów. Dla każdego rozmiaru populacji wykonano dziesięć przebiegów.

W tabeli 4.5 podano zestawienie wyników uzyskanych w dziesięciu symulacjach. Interesował nas czas (mierzony w pokoleniach) potrzebny na znalezienie optimum globalnego. Podobnie jak w poprzednich eksperymentach, za optimum globalne przyjęto takie rozwiązanie, którego wartość funkcji przystosowania przekracza 1,35. Zmiana średniego przystosowania w populacji powie nam o charakterze ewolucji – czy populacja zasiedla wysoki szczyt i czy robi to stopniowo, czy raczej skokowo.

Tabela. 4.5. Wyniki ewolucji dla Q_2 przy różnej liczbie osobników w populacji (maksymalny czas ewolucji ustalony na 3000 pokoleń)

Rozmiar populacji	Czas znalezienia rozwiązania lepszego niż 1,35 (tzn. na wysokim szczycie), w pokoleniach, w poszczególnych symulacjach									
	1	2	3	4	5	6	7	8	9	10
10	649	652	532	548	681	586	697	462	348	385
100	1693	1041	1170	1132	1005	1380	1021	1269	1514	1330
1000	–	2598	2266	–	2049	–	2438	–	–	3863

Należy zauważyć, że wyniki te są dość zaskakujące. Łatwo znaleźć w literaturze przedmiotu uwagi, że większa liczba ewoluujących osobników, to co prawda wydłużenie czasu obliczeń (zwiększa się liczba obliczeń funkcji przystosowania w każdym pokoleniu), ale też większe szanse na znalezienie dobrego rozwiązania w mniejszej



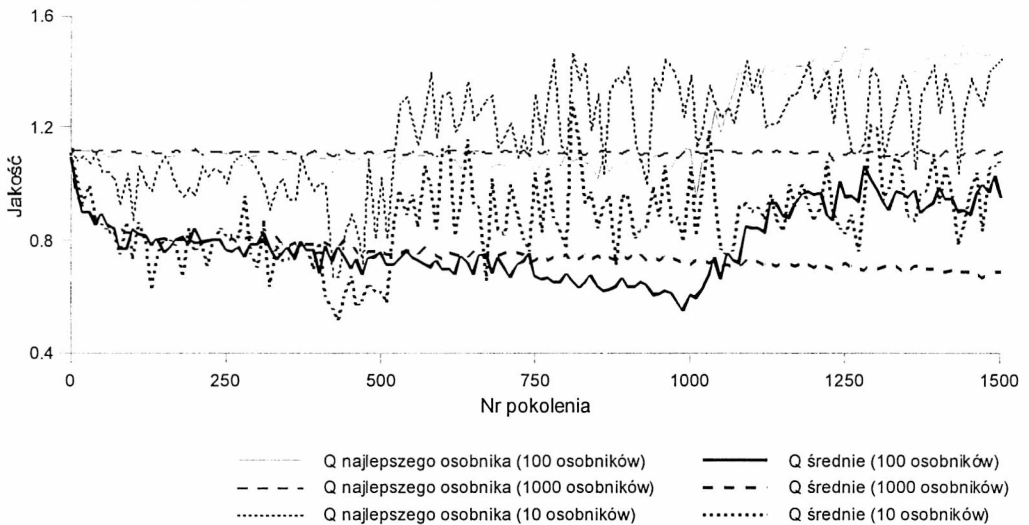
liczbie pokoleń. Uzyskane tutaj wyniki są jednak sprzeczne z tą opinią. Dane z tabeli 4.5 przedstawiono w postaci wykresu (rys. 4.53). Brakuje na nim danych dla populacji liczącej 1000 osobników, ponieważ w żadnym z dziesięciu przebiegów ewolucja nie znalazła optymalnego rozwiązania w czasie krótszym niż 2000 pokoleń. Wi-

Rys. 4.53. Średni czas znalezienia optymalnego rozwiązania w dziesięciu symulacjach dla 10- i 100-elementowej populacji, funkcja Q_2 – dane z tabeli 4.5

dać wyraźnie, że im więcej osobników w ewoluującej populacji, tym ewolucja przebiega gorzej (biorąc pod uwagę czas znalezienia optymalnego rozwiązania). Przy małym rozmiarze populacji ewolucja wydaje się być dość skuteczna, a wyniki są dość powtarzalne.

Popatrzmy na zmiany średnich jakości populacji i jakości najlepszego osobnika w populacji (rys. 4.54). Populacja licząca 1000 osobników ewoluuje „łagodnie”, stopniowo⁵. Po znalezieniu lepszego rozwiązania populacja stopniowo przechodzi na wysoki szczyt, pozostając na nim. Średnia jakość populacji zwiększa się, jednak nie jest zbyt wysoka, ponieważ ewoluująca z makromutacjami populacja nie staje się jednorodna. Wartość średniej jakości stabilizuje się na pewnym poziomie (rys. 4.56).

Zmiany najlepszej i średniej jakości w zależności od rozmiaru populacji



Rys. 4.54. Zmiany jakości w populacji (najlepszego osobnika i średniej) dla populacji o różnej liczbie osobników – przykładowe przebiegi dla funkcji Q_2 (dane co 10 pokoleń)

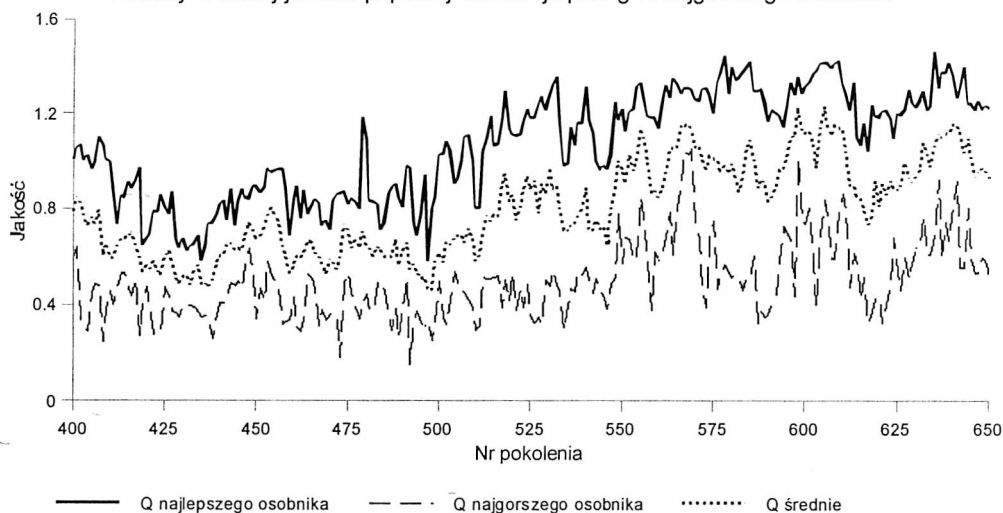
Trochę większe wahania średniego przystosowania populacji występują przy 100 osobnikach, ewolucja populacji dziesięcioelementowej pełna jest natomiast gwałtownych zmian. Populacja wciąż gubi optymalne rozwiązanie, po czym ponownie je znajduje, wędrując dość szybko po przestrzeni fenotypowej. Widać to dobrze na rysunku 4.55, gdzie pokazano zmiany średniej, najlepszej i najgorszej jakości w populacji w wybranym przedziale czasu – w okolicy pierwszego znalezienia optimum.

Otrzymany wynik – skuteczniejsza w znajdowaniu optimum ewolucja populacji o niewielkim rozmiarze – daje się wytłumaczyć. Zadanie znalezienia optimum przy

⁵ Wykresy na rysunku 4.54 wykonano na podstawie danych zapisywanych co 10 pokoleń, nie muszą więc dokładnie oddawać danych zebranych w tabeli 4.5, w której odnotowano pierwsze znalezienie optimum (zazwyczaj ginęło ono w następnym pokoleniu). To właśnie pokolenie może nie być uwidocznione na wykresie.

przyjętych parametrach algorytmu jest na tyle trudne, że łatwiej jest je znaleźć poprzez zejście z niskiego szczytu i przejście przez dolinę przystosowawczą niż przez „przeskoczenie” doliny. Populacja licząca mało osobników wędruje po przestrzeni fenotypowej, przypominając trochę błądzenie przypadkowe – o czym świadczą dość duże wahania średniej jakości populacji i obserwacja populacji w trakcie ewolucji. Taka populacja może zatem zejść w dolinę, po czym rozpocząć wędrówkę na nowy szczyt. Na żadnym szczycie taka populacja nie zatrzyma się na długo ani nie skupi na jego małym obszarze.

Zmiany średniej jakości populacji oraz najlepszego i najgorszego osobnika

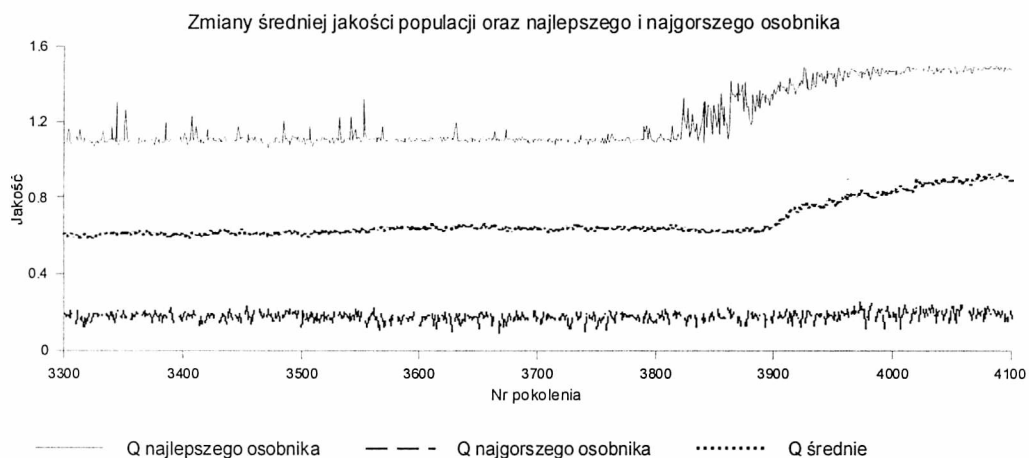


Rys. 4.55. Zmiany średniej jakości w populacji na tle najlepszego i najgorszego osobnika przy 10-elementowej populacji – przykładowy przebieg dla funkcji Q_2 (dane co jedno pokolenie)

Małe populacje podlegają większemu dryfowi również w klasycznych algorytmach genetycznych, a nie obserwuje się tam ich większej skuteczności. Wygląda na to, że odgrywa tu rolę nadmiarowość genów. Zauważmy, że w tych eksperymentach osobniki podlegają makromutacjom – kumulowane zmiany w genach nadmiarowych powodują duże zmiany fenotypowe podczas ich przesunięcia (transpozycji) do grupy genów fenotypowych. Małej populacji pozwala to zawędrować w gorsze rejony przystosowawcze, gorszy osobnik – położony dalej od lokalnego szczytu – ma szansę mieć potomka. Jego relatywne przystosowanie (w stosunku do lepszych i średniej w populacji) jest lepsze niż przy dużych populacjach, gdzie najlepszy osobnik jest bliżej lokalnego szczytu i średnie przystosowanie też jest większe. Wskutek makromutacji osobnik z „peryferii” może mieć potomka, którego fenotyp będzie w okolicy wysokiego szczytu.

W celu zweryfikowania tego wyjaśnienia nietypowego wyniku, sprawdzimy charakter ewolucji przy braku nadmiarowości. Dla funkcji pięciowymiarowej trudno jest dobrać parametry algorytmu tak, aby możliwe było osiągnięcie wysokiego szczytu bez

nadmiarowości i przy stosunkowo niedużej maksymalnej zmianie mutacyjnej (która też jest pewnego rodzaju makromutacją). Kilkanaście tysięcy pokoleń nie daje pozytywnych wyników, trudno jest zatem sprawdzić, jak rozmiar populacji wpływa na tempo i charakter ewolucji. Dlatego eksperymenty takie wykonano dla funkcji dwuwymiarowej Q_1 .



Rys. 4.56. Zmiany średniej jakości w populacji liczącej 1000 osobników na tle najlepszego i najgorszego osobnika – przykładowy przebieg dla funkcji Q_2 (dane co jedno pokolenie)

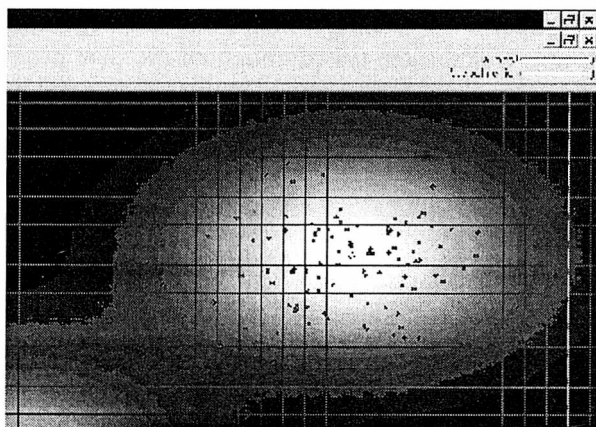
Eksperymenty z dwuwymiarową funkcją Q_1 bez genów nadmiarowych

Wstępne eksperymenty pokazały, że możliwe jest osiągnięcie optimum dla następujących parametrów (bez makromutacji i genów nadmiarowych): $M_{\max} = 20$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trs}} = P_{\text{trz}} = 0$ (początkowe wartości genów równe są 1000). Podobnie jak dla Q_2 , tak i teraz wyniki zebrano w tabeli 4.6.

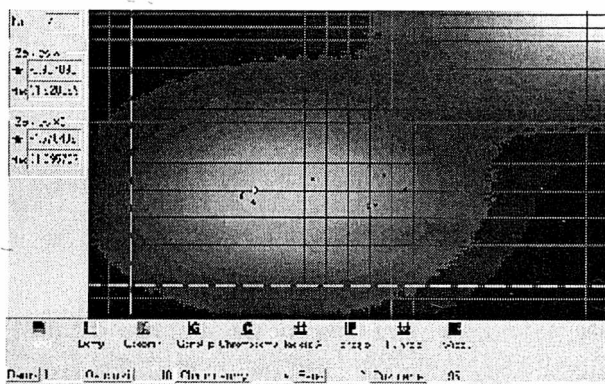
Tabela 4.6. Wyniki ewolucji dla Q_1 , przy różnej liczbie osobników w populacji

Rozmiar populacji	Czas znalezienia rozwiązania lepszego niż 1,35 (tzn. na wysokim szczycie) w pokoleniach, w poszczególnych symulacjach									
	1	2	3	4	5	6	7	8	9	10
10	przy czasie maksymalnym 10 000 pokoleń nie udało się znaleźć optimum									
100	2695	22935	6410	1532	–	9022	8045	7989	–	3954
1000	314	357	877	433	1879	555	920	399	486	631

Zauważmy, że teraz wyniki są zgodne z tym, czego oczekiwaliśmy: większa liczba osobników w ewoluującej populacji ułatwia znalezienie optimum, choć wydłuża się czas obliczeń pojedynczego pokolenia. Przy pewnych parametrach mało liczne populacje, mimo iż nie są mocno skupione – widoczny jest efekt dryfu genetycznego (rys.



Rys. 4.57. Populacja po 10000 pokoleniach $M_{\max} = 20$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trs}} = P_{\text{trz}} = 0$, 100 osobników, funkcja Q_1



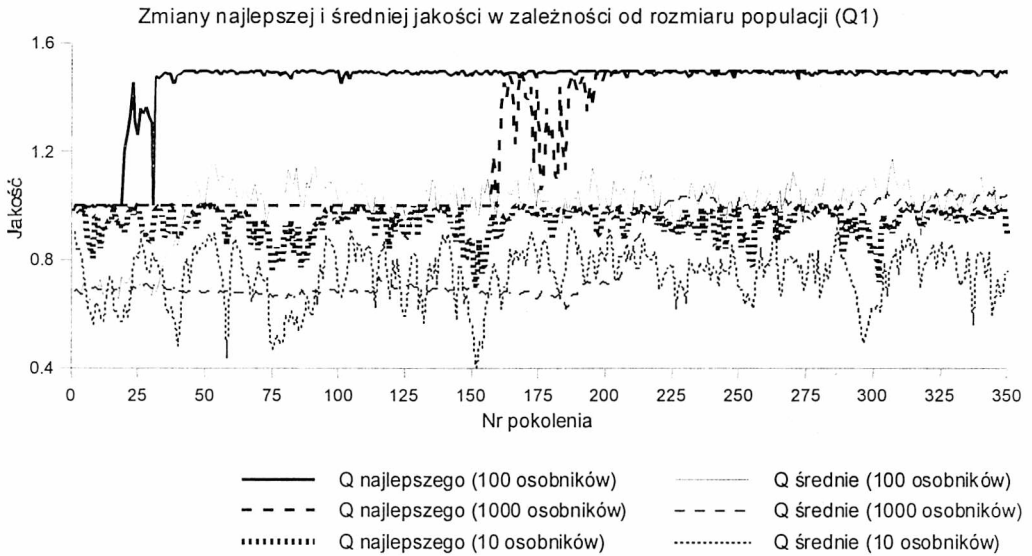
Rys. 4.58. Populacja po 1000 pokoleniach $M_{\max} = 20$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trs}} = P_{\text{trz}} = 0$, 10 osobników, funkcja Q_1

4.58), nie są w stanie znaleźć lepszego szczytu, podczas gdy populacja licząca 1000 osobników (rys. 4.57) radzi sobie z tym średnio w ciągu kilkuset pokoleń. Tu – w odróżnieniu od poprzedniej sytuacji – widać sensowność stosowania większych populacji.

Na rysunku 4.59 pokazano zmiany średniego przystosowania populacji oraz zmiany przystosowania najlepszego osobnika w populacji. Dane pochodzą z przykładowej, typowej symulacji. Należy zwrócić uwagę, że na osi odciętych są podane numery pokoleń, które nie są identyczne dla wszystkich przebiegów. Ponieważ ewolucja przebiega w różnym tempie, pokazano zmiany przystosowania w ciągu tych 350 pokoleń, w których znajdowane jest optimum, z wyjątkiem populacji dziesięcioosobowej, gdzie optimum nie było znalezione (na wykresie populację dziesięcioelementową pokazano w tych samych pokoleniach co stu-elementową).

W każdej z dziesięciu symulacji (dla 100- i 1000-elementowej populacji) średnia jakość populacji tylko na krótko przekraczała wartość 1,1 – na ogół wahała się wokół jedności. Podobnie jak poprzednio, największe wahania występują przy populacji dziesięcioelementowej, choć tym razem tej populacji ani razu nie udało się znaleźć optimum, mimo iż część symulacji prowadzono do 20 000 pokoleń.

Zobaczmy jeszcze, jak wpłynie na podane wyniki dodanie nadmiarowych genów i związanych z tym makromutacji: zamiany genów nadmiarowych z fenotypowymi. Mamy zatem eksperymenty dla liczebności populacji 100, 1000 i 10 osobników, przy wartościach parametrów: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$, funkcja Q_1 , dla trzech wartości $M_{\max}$ (równe 10, 20 i 50). Zmiana tego parametru miała na celu sprawdzenie, czy taki sposób zwiększania mutacji wpływa na ewolucję małych, średnich i dużych populacji. Wyniki zebrano w tabeli 4.7.



Rys. 4.59. Zmiany jakości w populacji (najlepszego osobnika i średniej) dla populacji o różnej liczbie osobników – przykładowe przebiegi dla funkcji Q_1 (dane dla: 100 i 10 osobników – pokolenia 9000–9350, 1000 osobników – pokolenia 1000–1350)

Tabela 4.7. Wyniki ewolucji dla Q_1 , przy różnej liczbie osobników w populacji, z nadmiarowością i różną maksymalną zmianą mutacyjną

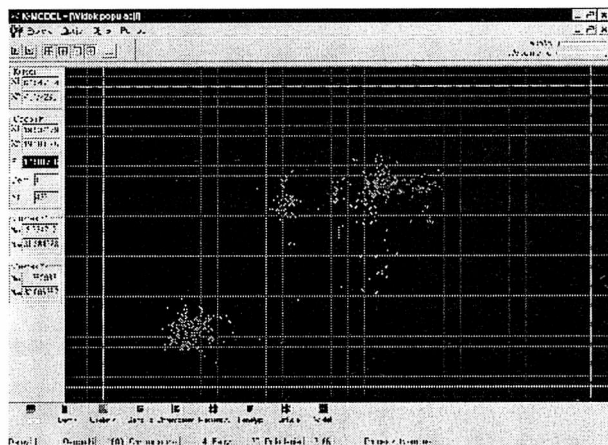
Lp.	Czas znalezienia optymalnego rozwiązania (liczba pokoleń)								
	100 osobników			1000 osobników			10 osobników		
	$M_{\max}=10$	$M_{\max}=20$	$M_{\max}=50$	$M_{\max}=10$	$M_{\max}=20$	$M_{\max}=50$	$M_{\max}=10$	$M_{\max}=20$	$M_{\max}=50$
1	869	296	21	266	66	15	452	232	98
2	711	249	26	677	102	11	705	307	81
3	796	296	20	435	72	12	732	603	114
4	675	151	22	264	241	9	603	413	101
5	578	168	23	469	175	10	704	392	89
6	408	253	26	401	42	16	683	665	121
7	394	282	20	722	205	12	712	383	96
8	446	439	24	524	55	13	582	401	94
9	768	328	19	389	189	11	960	396	108
10	919	301	21	424	135	15	541	406	111

Dla $M_{\max} = 50$, populacja tak szybko znajdowała rozwiązanie, że właściwie makromutacje związane z zamianą genów nie zdążają odegrać żadnej roli. Dla najmniejszej maksymalnej zmiany mutacyjnej, nadal populacja bardzo liczna (1000 osobników) ewoluje szybciej niż mniejsze, ale zysk nie jest już tak duży. Dla $M_{\max} = 10$, po zwiększeniu dziesięciokrotnie liczby osobników z 10 na 100 i na 1000, otrzymujemy średnie czasy znalezienia optimum odpowiednio: 667, 656 i 452 pokoleń.

Eksperymenty z trzydziestowymiarową funkcją Q_5

Podobne eksperymenty wykonano dla funkcji trzydziestu zmiennych, z dwoma szczytami (funkcja Q_5). Ze względu na długi czas obliczeń, symulacje powtarzano tylko pięciokrotnie. Wartości parametrów to: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$, $M_{\text{max}} = 600$.

Populacja licząca 1000 osobników znajdowała optymalne rozwiązanie w czasie od 3181 do 3543 pokoleń, 100 osobników potrzebowало na to od 3422 do 3775, natomiast 10 osobników – od 4505 do 6854. W tym ostatnim przypadku populacja krążyła



Rys. 4.60. Ewolucja populacji tysiącelementowej w pokoleniu 3166 (w 3181 pokoleniu znalezione optimum)

wokół najlepszego szczytu już od 1500.–3000. pokolenia, ale bardzo trudno było jej znaleźć optimum. Stuelementowa i tysiącelementowa populacja po znalezieniu optimum przechodzi na większy szczyt, pozostając na nim.

Charakterystyczne jest, że przy funkcji Q_5 , to jest przy 30 zmiennych i dwóch szczytach, populacja zawsze wcześniej osiągała wysoką średnią jakość (w granicach 1,1–1,2), a dopiero później znajdowała optimum. Dla funkcji Q_1 i Q_2 sytuacja była odwrotna, jedynie przy bardzo

małych populacjach (10 osobników) zdarzają się takie przebiegi. Tu zadanie znalezienia optimum jest trudne, populacje tysiąc- i stuelementowe dają porównywalne wyniki, najmniejsza ewoluuje natomiast najgorzej (biorąc pod uwagę liczbę pokoleń potrzebną na znalezienie optimum). Wydawałoby się, że te wyniki nie są zgodne z poprzednimi – mała populacja, przy trudnym zadaniu, nie daje lepszych wyników niż duża, choć porównanie czasu znalezienia optimum przez populacje stu- i tysiącelementowe nie daje podstaw do stwierdzenia, że ta ostatnia jest lepsza (zwłaszcza, jeśli uwzględnimy czas obliczeń). W gruncie rzeczy należy tu rozważyć wpływ wymiaru zadania na szybkość znajdowania optimum.

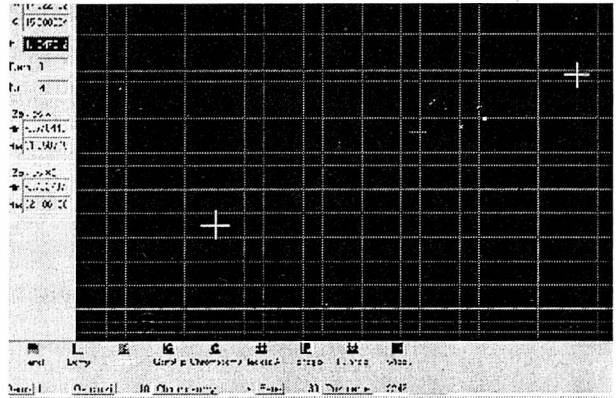
W naszych eksperymentach za optimum uważamy każdy punkt w przestrzeni fenotypowej, dla którego wartość funkcji Q jest wyższa od 1,35. Przeprowadźmy eksperyment myślowy, polegający na losowym wybieraniu punktu na tej przestrzeni. Prawdopodobieństwo trafienia w obszar uznawany za optimum jest równe stosunkowi pola tego obszaru do pola powierzchni całej rozpatrywanej przestrzeni. Popatrzmy, jak to się zmienia ze zmianą wymiaru. Weźmy przypadek jednowymiarowy: Q niech będzie funkcją jednej zmiennej x_1 . Załóżmy, że interesujący nas obszar to 10% całego obszaru, czyli 10% odcinka $[x_{1\min}, x_{1\max}]$. Prawdopodobieństwo trafienia w ten obszar (rzu-

cając kulką na cały odcinek) wynosi 0,1. Teraz założymy dwa wymiary i, dla uproszczenia, niech cały obszar stanowi kwadrat, a nas interesuje wycinek tego kwadratu – po 10% każdego z dwóch wymiarów. Inaczej mówiąc, 10% każdej współrzędnej to obszar optimum. Ale teraz prawdopodobieństwo trafienia w ten obszar to nie 0,1, ale $(0,1)^2 = 0,01$. Dla pięciu wymiarów będzie to $(0,1)^5$, a dla trzydziestu – niezmiernie małe, równe $(0,1)^{30}$. Nic więc dziwnego, że ewoluującej populacji jest bardzo

trudno trafić w interesujący nas fragment powierzchni, populacja „trafia” na wysoki szczyt, ale niedokładnie w ten obszar. Ewoluuąc po zboczu wysokiego szczytu zmierzając w kierunku optimum, osiągając go zazwyczaj w niedługim czasie po chwili, gdy jej średnia jakość zrównuje się z wartością niskiego szczytu.

W prowadzonych eksperymentach zajmowało to populacji nie więcej niż 20 pokoleń. Nie dotyczy to populacji dziesięcioelementowych, które nie potrafią skutecznie wspinać się po zboczu. Na ogół populacje małe wędrowały na zbocze wysokiego szczytu w krótszym czasie niż większe (od 1280 do 3151), ale nie potrafiły tego wykorzystać.

Po porównaniu populacji liczących 100 i 1000 osobników możemy śmiało stwierdzić, że nie opłaca się skracanie czasu szukania optimum kosztem zwiększania liczebności populacji (z około 3600 do około 3200, przy wzroście liczebności populacji ze stu do tysiąca). Okupione jest zbyt mocno wydłużonym czasem obliczeń. Z drugiej strony, nie można też stosować zbyt małych populacji, zwłaszcza przy wielowymiarowych funkcjach, ponieważ szanse znalezienia optimum (w akceptowalnym czasie) są wtedy zbyt małe.



Rys. 4.61. Ewolucja populacji dziesięcioelementowej, pokolenie 2242 (optimum zostało znalezione w 4505 pokoleniu, krzyzykami zaznaczone optima)

4.5. INNE PARAMETRY MODELU

Dostępne w K-modelu metody selekcji a charakter ewolucji

W K-modelu zaimplementowano cztery metody selekcji, wszystkie z nich dają efekty raczej *miękkiej selekcji*: metoda Poissona, ruletki, stochastyczna według reszt bez powtórzeń i metoda deterministycznego próbkowania. Przeprowadzono eksperymenty dla pięciowymiarowej i dwuwymiarowej funkcji. W badaniach tak ustalono

wartości parametrów, by możliwe było znalezienie lepszego rozwiązania. Umożliwia to porównanie przebiegów przy różnych metodach selekcji.

Dla funkcji pięciowymiarowej Q_2 wykonano symulacje z następującymi parametrami: $M_{\max} = 600$, $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$. Czas symulacji ustalono na 2000 pokoleń. Wyniki wstępnych eksperymentów pokazują, że przy tych parametrach populacja znajduje rozwiązanie na wysokim szczycie.

Dla funkcji dwuwymiarowej Q_1 tak dobrano parametry, by niekonieczne było stosowanie nadmiarowości, a możliwe natomiast znalezienie optymalnego rozwiązania. Na podstawie wstępnych symulacji stwierdzono, że jest to możliwe przy $M_{\max} = 30$, $P_{\text{rkm}} = 0,25$ oraz $P_{\text{mut}} = 0,2$.

W obu przypadkach najbardziej miękką selekcją okazały się metoda ruletki i deterministycznego próbkowania. Przy deterministycznym próbkowaniu najlepsze rozwiązanie było najszybciej znajdowane (średnio 558 pokoleń dla Q_2 i 100 dla Q_1), ale też obserwowany jest tu duży rozrzut wyników (np. od 414 do 859, czy od 70 do 270). Zbliżone wyniki daje metoda ruletki: 654 i 103, rozrzut od 504 do 752 i od 57 do 181. Metoda Poissona trochę gorzej sobie radzi z zadaniem znalezienia optimum tych funkcji, dla Q_2 jest to średnio 1124 pokolenia (od 770 do 1782) i 121 (od 47 do 269) dla Q_1 . Metoda stochastyczna według reszt bez powtórzeń na ogół nie daje sobie rady przy tych parametrach. W żadnym z tych przebiegów populacja nie osiągnęła średniego przystosowania większego niż 1,1, mimo iż jej osobniki w większości były na wysokim szczycie i w jego okolicy. Rozrzut osobników był tak duży, że populacja jako całość nie jest dobrze przystosowana.

Biorąc pod uwagę powyższe wyniki, można powiedzieć, że brak jest jednoznacznych przesłanek do uznania którejś z nich za istotnie lepszą od pozostałych.

Liczba chromosomów w genotypie osobnika

W K -modelu liczba chromosomów u jednego osobnika jest ściśle związana z operatorem rekombinacji. Przypomnijmy, że rekombinacja pomiędzy osobnikami polega na wymianie odpowiednich chromosomów. Decydując o liczbie chromosomów, należy ten fakt mieć na uwadze. Przy stałym prawdopodobieństwie mutacji średnia liczba osobników, u których będzie następowała zmiana spowodowana rekombinacjami, zależy od liczby chromosomów: dla $P_{\text{rkm}} = 0,25$, przy genotypach czterochromosomowych, średnio wymieniany będzie jeden chromosom u każdego osobnika, przy dwuchromosomowych osobnikach – jeden chromosom u co drugiego osobnika. Przy dwóch chromosomach u jednego osobnika zmiana fenotypowa osobnika spowodowana rekombinacją jest stosunkowo większa (połowa jego materiału genetycznego ulega zmianie). Eksperymentując np. z funkcją pięciu zmiennych Q_2 (parametry: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$ i $M_{\max} = 600$) najlepsze wyniki w sensie czasu otrzymania najlepszego rozwiązania otrzymujemy dla genotypu dwuchromosomowego (średnio 600–1000 pokoleń), czterochromosomowego – 900–1350 pokoleń, ośmiochromosomowego – 1650–1850 pokoleń. Różnice nie są zbyt wyraźne. Jeśli dla dwuchromosomowych genotypów zwiększymy prawdopodobieństwo rekombinacji do $P_{\text{rkm}} = 0,5$ (średnio jeden

chromosom u jednego osobnika jest wymieniany), to wtedy następuje przyspieszenie znalezienia optymalnego rozwiązania do 200–650 pokoleń. Populacja jest zróżnicowana, możliwe są duże zmiany fenotypowe powodujące penetrację odległych obszarów w przestrzeni fenotypowej. Średnia jakość populacji jest stosunkowo niska.

Zachowanie najlepszego osobnika w populacji

Implementacja K-modelu pozwala na włączenie opcji „zachowaj najlepszego”. Wtedy najlepszy osobnik bezwarunkowo i bez żadnych zmian jego genotypu przechodzi do nowej populacji (następnego pokolenia). Można postawić pytanie: *czy to się opłaca?* Wszak natura nie stworzyła takiego precedensu – nieśmiertelności, dopóki jest się najlepszym.

Wyniki eksperymentów wskazują na celowość rozróżnienia: ewolucja polegająca na wspinaniu się na jeden szczyt i ewolucja w przestrzeniach wielomodalnych, w których należy znaleźć optimum globalne. Przy wspinaniu się na szczyt zachowywanie najlepszego osobnika może przyspieszyć wspinaczkę. Inaczej wygląda sytuacja, gdy mamy do czynienia z funkcjami wielomodalnymi, w których pojedyncza zmiana mutacyjna nie jest w stanie przesunąć osobnika w rejony wyższe niż lokalny szczyt, by móc konkurować z najlepszym dotychczas osobnikiem. Zachowywanie najlepszego osobnika może nawet zablokować populację na lokalnym szczycie. Średnia jakość populacji jest podwyższana przez istnienie osobnika wysoko na lokalnym szczycie, co zmniejsza szanse reprodukcyjne osobników położonych na krańcach terenu zajętego przez populację, natomiast to takie właśnie osobniki mają większe szanse wydać potomka położonego bliżej sąsiedniego szczytu.

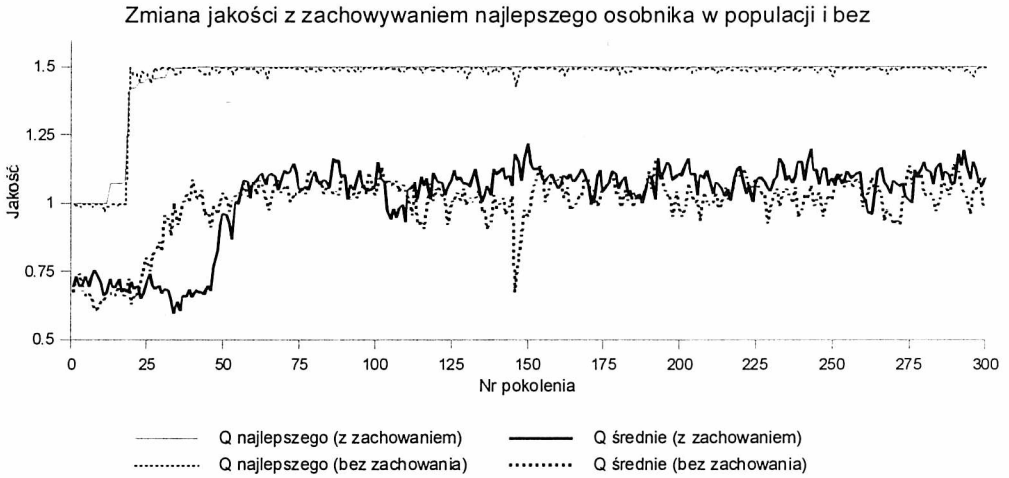
Jeśli w eksperymentach nie stosowano żadnej postaci makromutacji (tylko mutacje i rekombinacje, jak na rysunku 4.62)⁶, to czasy znajdowania optimum są w miarę porównywalne – średnio kilka tysięcy pokoleń, z dużymi wahaniami.

Znalezione optimum gubione jest co najwyżej na kilka pokoleń, dlatego też nie ma zbyt dużych różnic pomiędzy ewolucją populacji zachowującej najlepszego osobnika i nie zachowującej. Dla Q_1 , bez makromutacji, optimum wydaje się być znajdowane trochę szybciej, jeśli nie zachowuje się najlepszego osobnika, natomiast raz znalezione nie jest już gubione i trochę szybciej populacja zasiedla wyższy szczyt, jeśli najlepszy osobnik będzie „nieśmiertelny”.

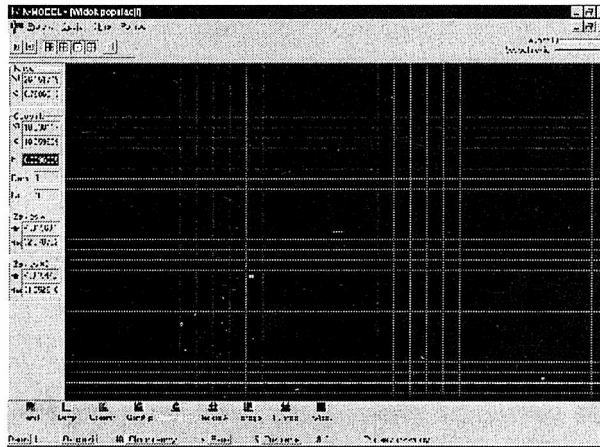
Inaczej sytuacja będzie się miała, kiedy ewoluująca populacja musi zejść ze szczytu, by poprzez dolinę dostać się na sąsiedni szczyt. Pamiętamy, jak szybko znajdowane było optymalne rozwiązanie przy dziesięcioelementowej populacji i funkcji Q_2 . Sprawdźmy, jak będzie ewoluować ta sama populacja, przy tych samych parametrach, z wyjątkiem opcji „zachowaj najlepszego”.

Czas znalezienia optymalnego rozwiązania, mierzony liczbą pokoleń, mamy zebrany w tabeli 4.5, wynosi on od 348 do 681 pokoleń. Pamiętamy, że populacja wędruje po przestrzeni, natrafiając po drodze na optymalny szczyt, nie potrafi się na nim skupić i zatrzymać.

⁶ Numery pokoleń różne dla poszczególnych przebiegów.

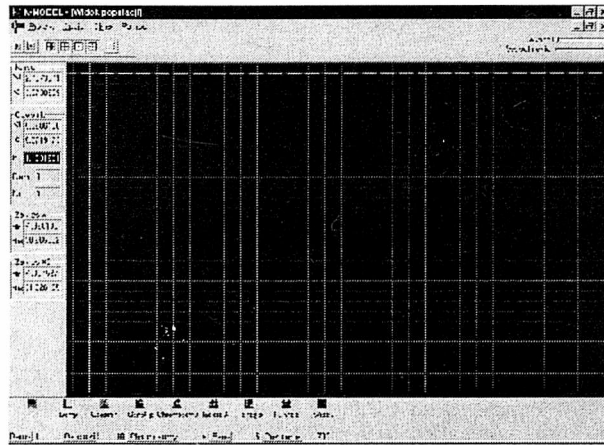


Rys. 4.62. Wykres zmian jakości najlepszego osobnika i średniej jakości w populacji przy zachowywaniu najlepszego osobnika w populacji i bez zachowywania dla funkcji Q_1 , parametry: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0$, $M_{\text{max}} = 20$ (numery pokoleń odliczone od kilkunastu pokoleń przed znalezieniem optimum)



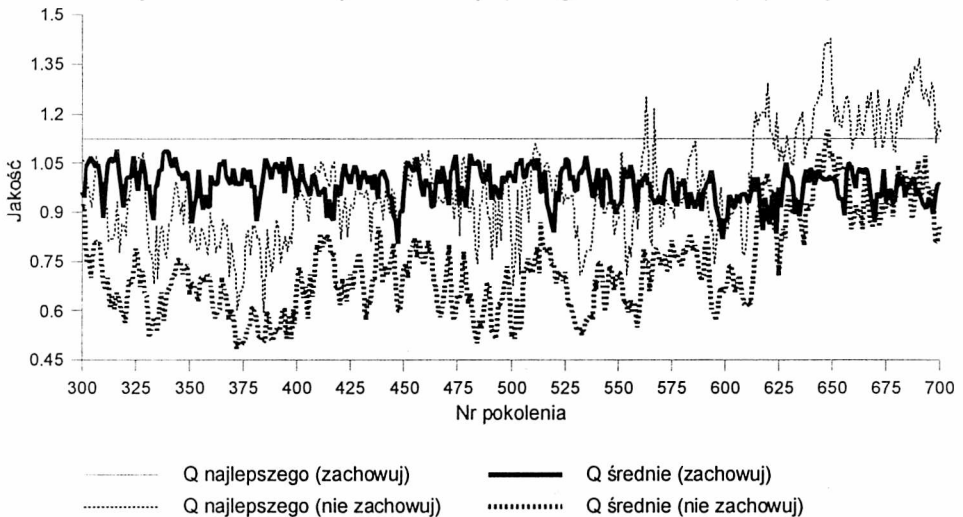
Rys. 4.63. Funkcja Q_2 , 10 osobników, fenotyp najdalej wysuniętego osobnika [10,3 10,3 13,4 8,4 9,7]

Przykład takiej „wędrującej” populacji pokazany jest na rysunku 4.63. Populacja ewoluująca przez 700 pokoleń, ale z włączoną opcją „zachowuj najlepszego”, nie zdołała ani na chwilę opuścić lokalnego optimum (rys. 4.64). Wykresy na rysunku 4.65 pokazują jak mocno spada zarówno średnia jakość populacji, jak i przystosowanie najlepszego osobnika, gdy nie zachowujemy najlepszego. Ale to umożliwia populacji znalezienie optimum.



Rys. 4.64. Funkcja Q_2 , 10 osobników „zachowaj najlepszego” fenotyp najdalej wysuniętego osobnika [6,0 6,99 5,68 7,5 6,0]

Zmiana jakości z zachowywaniem najlepszego osobnika w populacji i bez

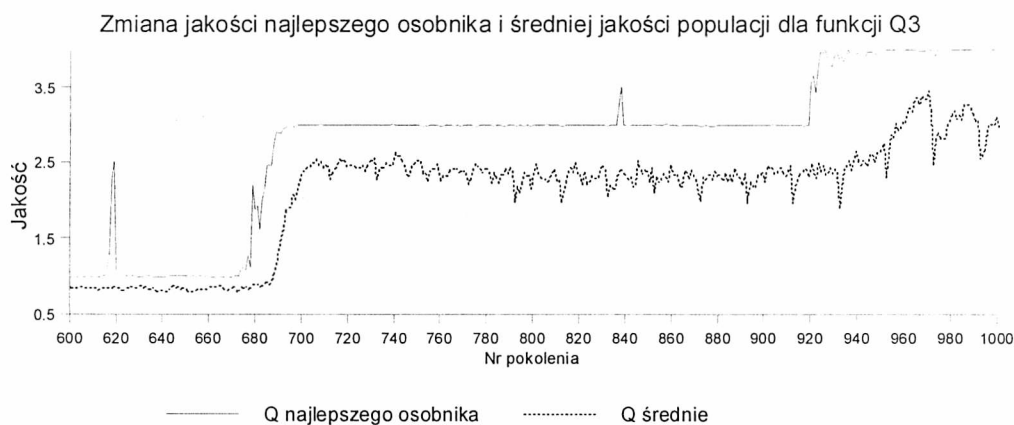


Rys. 4.65. Wykres zmian jakości najlepszego osobnika i średniej jakości w populacji przy zachowaniu najlepszego osobnika w populacji i bez zachowywania dla funkcji Q_2 , parametry: $P_{rkm} = 0,25$, $P_{mut} = 0,2$, $P_{trs} = P_{trz} = 0,05$, $M_{max} = 600$

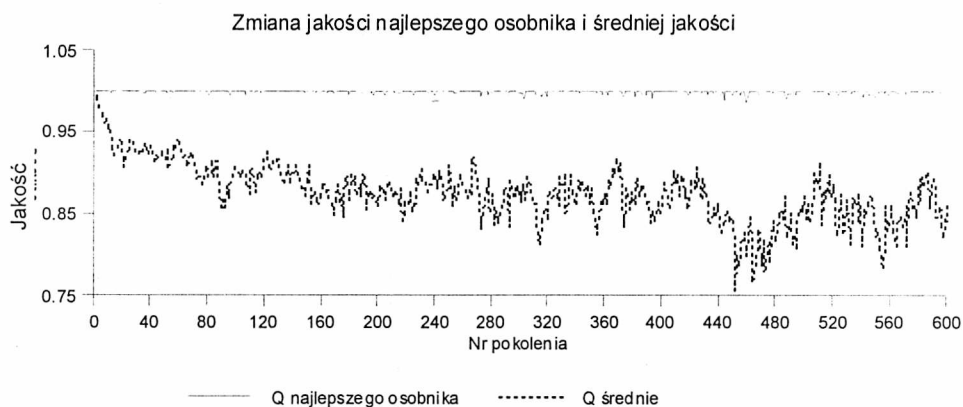
Opcja „zachowaj najlepszego” dla ewolucji przy czterech szczytach (funkcja Q_3)

Wyniki wstępnych eksperymentów pokazały, że populacja radzi sobie z szukaniem najwyższego szczytu przy następujących wartościach parametrów: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,02$, $M_{max} = 100$, $t_{kr} = 20$, w kryzysie: $P_{rkm} = P_{mut} = 0,5$, $P_{trs} = P_{trz} = 0,1$, symulacje bez zachowywania najlepszego osobnika (rys. 4.66 i 4.67). Ponumerujmy szczyty: I – wysokość = 1, współrzędne: 5, 5; II – wysokość 2,

współrzędne: 40, 10; III – wysokość = 3, współrzędne: 10, 40 i IV – wysokość = 4, współrzędne: 45, 45.



Rys. 4.66. Wykres zmian jakości najlepszego osobnika i średniej jakości dla funkcji Q_3 mającej cztery szczyty, parametry: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,02$, $M_{max} = 100$, $t_{kr} = 20$, ostatnie 400 pokoleń, w których populacja zmieniała szczyty (bez zachowywania najlepszego)

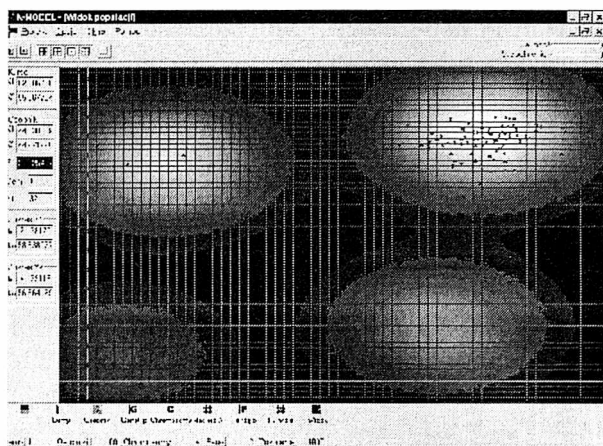


Rys. 4.67. Wykres zmian jakości najlepszego osobnika i średniej jakości dla funkcji mającej cztery szczyty Q_3 , pierwsze 600 pokoleń, w których populacja ciągle ewoluuje na najniższym szczycie (parametry jak na poprzednim rysunku)

Zazwyczaj w okolicach 600.–650. pokolenia populacja przechodziła na trzeci szczyt (jego współrzędne to 10, 40), czasami wcześniej pojedyncze osobniki pojawiały się na drugim szczycie (40, 10), ale nie w każdym przebiegu, po czym w około 1100. pokoleniu populacja zasiedlała najwyższy szczyt (jego współrzędne to 45, 45).

Teraz powtarzamy te same symulacje, ale z włączoną opcją „zachowaj najlepsze”. Ewolucja przebiega znacznie wolniej. Tylko raz na dziesięć symulacji popu-

lacja znalazła trzeci co do wysokości szczyt (populacja osiągnęła go bezpośrednio z pierwszego szczytu) w ciągu 4000 pokoleń (w 3908 pokoleniu), ale nawet wtedy w następnych trzech tysiącach nie była w stanie znaleźć najwyższego szczytu. Okazuje się, że mimo dość dużych makromutacji (włączona jest nadmiarowość genów i kryzys powodujący duże zmiany całej populacji co 20 pokoleń) zachowywanie najlepszego osobnika zatrzymuje populację na najniższym szczycie, uniemożliwiając jej dalszą skuteczną ewolucję (rys. 4.69).

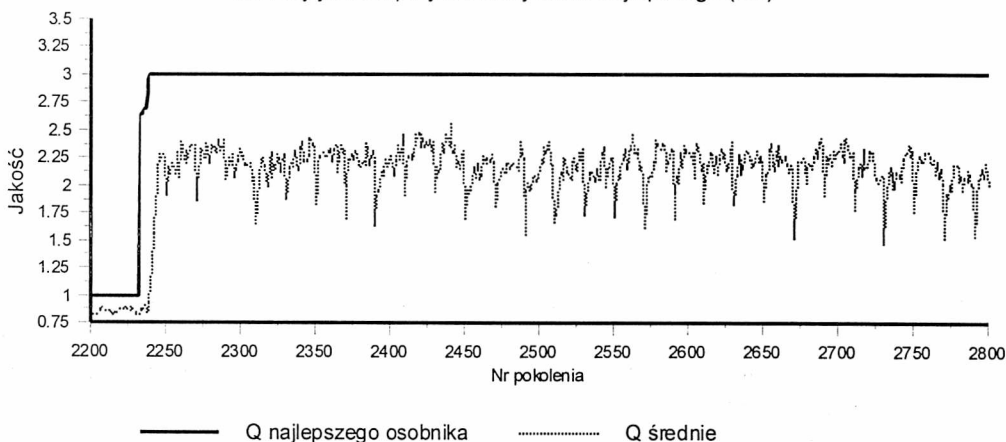


Rys. 4.68. Populacja po 1000 pokoleniach dla Q_3 , parametry: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,02$, $M_{max} = 100$, $t_{kr} = 20$, (bez zachowywania najlepszego)

Populacja dziesięcioelementowa radzi sobie z zadaniem znalezienia optimum globalnego, jeśli najlepszemu osobnikowi nie pozwalamy przechodzić do następnego pokolenia. W ciągu 2000 pokoleń populacja zazwyczaj znajdowała globalne optimum, przechodząc bądź przez drugi, bądź przez trzeci szczyt. Natomiast z opcją „zachowaj najlepszego”, po 15 tysiącach pokoleń mała populacja zawsze tkwiła na lokalnym szczycie, na którym rozpoczynała ewolucję.

Jak wynika z omówionych krótko eksperymentów, tylko dla stosunkowo prostej funkcji, dwóch zmiennych o niezbyt oddalonych dwóch szczytach, włączenie opcji

Zmiany jakości przy zachowywaniu najlepszego (Q_3)

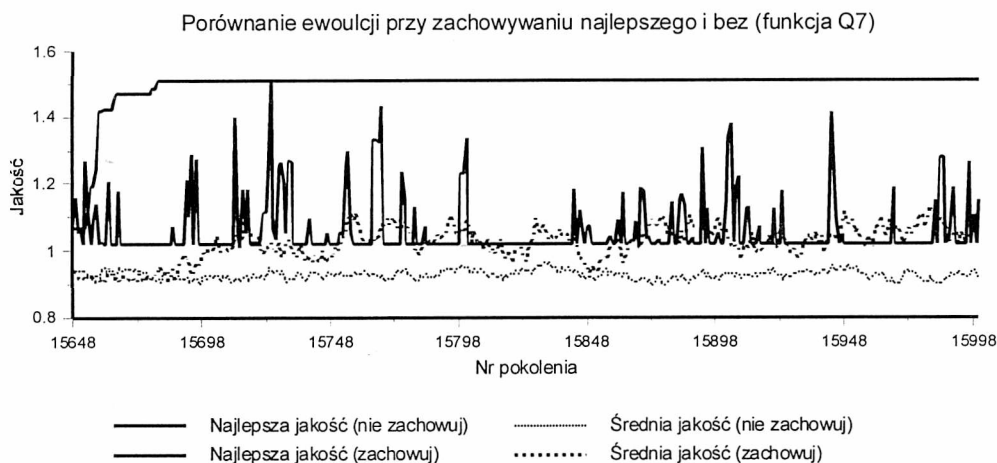


Rys. 4.69. Wykres zmian jakości najlepszego osobnika i średniej jakości populacji dla funkcji z czterema szczytami, parametry: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,02$, $M_{max} = 100$, $t_{kr} = 20$, z zachowaniem najlepszego (pokolenia, w których populacja zmieniła szczyt, do 7000 pokoleń brak optimum)

„zachowaj najlepszego” nie przeszkadza znacząco w ewolucji. W pozostałych przykładach bezwarunkowe przeżywanie najlepszego blokuje ewolucję.

Porównanie tej samej ewolucji z opcją „zachowaj najlepszego” i bez

Dla jednowymiarowej funkcji z dwoma szczytami Q_7 wykonano eksperymenty, w których, po znalezieniu najlepszego rozwiązania, leżącego na zboczu wysokiego szczytu (ale dość nisko), pozwalano populacji ewoluować z zachowaniem najlepszego osobnika i bez. Szybkość wspinania się populacji na wysoki szczyt jest zdecydowanie większa przy niepozwalaniu na gubienie rozwiązania, leżącego w okolicy tego szczytu. Pokazano to na rysunku 4.70 – począwszy od pokolenia, w którym najlepszy osobnik był bliżej wysokiego szczytu niż niskiego, powtórzono dwukrotnie ewolucję: z włączoną opcją „zachowaj najlepszego” i bez. Wartości parametrów podczas symulacji: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0,02$, $M_{\text{max}} = 100$. Przy zachowywaniu najlepszego osobnika średnia jakość populacji przekroczyła wartość 1,1 w 15715. pokoleniu, podczas gdy bez tej opcji – w 18566. pokoleniu. Przy samej wspinaczce na szczyt zachowywanie najlepszego osobnika jest pomocne.

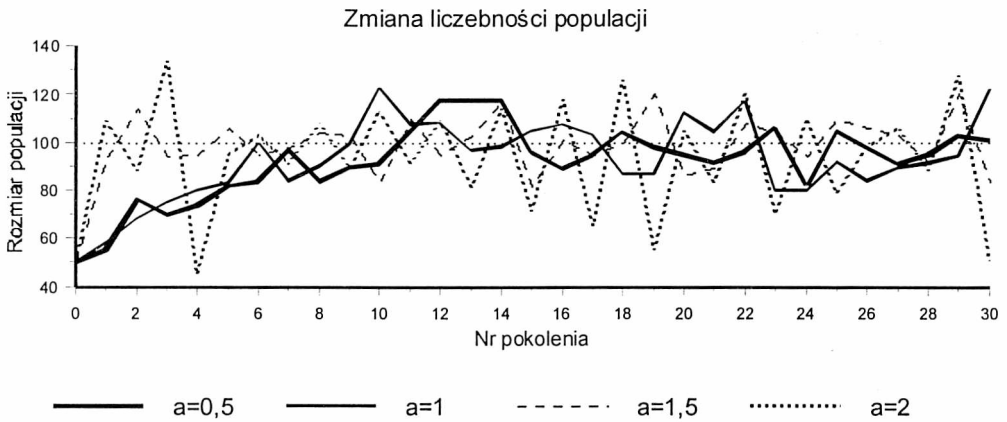


Rys. 4.70. Porównanie przebiegów ewolucji przy kontynuacji jej bez zachowywania najlepszego osobnika i z zachowywaniem

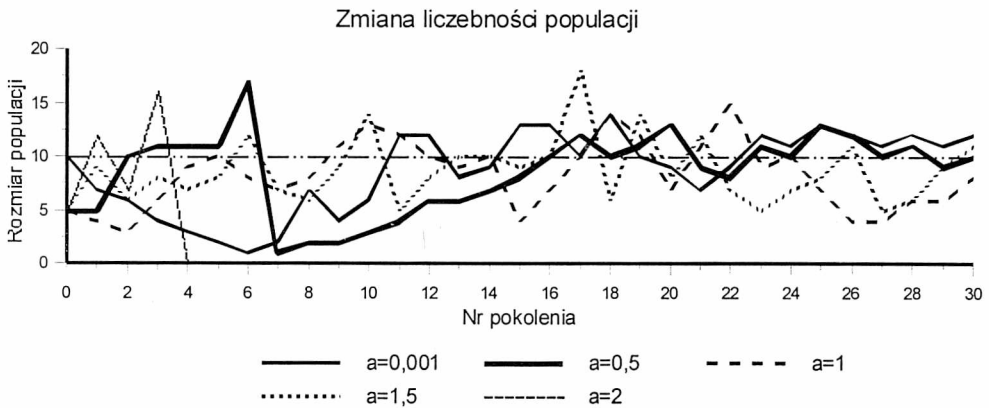
Szybkość osiągnięcia poziomu nasycenia przy zmiennym rozmiarze populacji

Stosując metodę selekcji Poissona mamy możliwość ewolucji populacji o zmiennym rozmiarze. Liczba potomków osobnika ustalana jest zgodnie z rozkładem Poissona o wartości średniej równej stosunkowi przystosowania tego osobnika do średniego przystosowania populacji. Liczba osobników populacji w następnym pokoleniu jest sumą wszystkich potomków. Przy takiej reprodukcji dochodzi do wymierania populacji, dlatego też ustalany jest poziom nasycenia⁷.

⁷ W biologii poziom nasycenia środowiska dla organizmów danego gatunku wynika np. z ograniczonej ilości pożywienia, terytorium itp.



Rys. 4.71. Zmiana liczebności populacji przy różnym parametrze a (funkcja Q_1 , tylko mutacje i rekombinacje: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $M_{\text{max}} = 20$, poziom nasycenia = 100 osobników)



Rys. 4.72. Zmiana liczebności populacji przy różnym parametrze a (funkcja Q_1 , tylko mutacje i rekombinacje: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,2$, $M_{\text{max}} = 20$, poziom nasycenia równy 10 osobników)

Parametr a w K -modelu odpowiada za szybkość osiągnięcia poziomu nasycenia przez ewoluującą populację, jego domyślna wartość to 1. Zbyt duża wartość a powoduje, że oscylacje rozmiaru populacji wokół poziomu nasycenia są dość duże, co przy niewielkiej populacji może doprowadzić do jej wymarcia.

Jak pokazano na rysunkach 4.71 i 4.72, duża wartość parametru a (większe od 1,5) powoduje, że ewoluująca populacja może łatwo wyginać. Dla mniejszych a stosunkowo liczna populacja nie wyginie, ale mała populacja (np. 10 osobników) wyginie nawet przy małych a , ponieważ uwidacznia się tu duży efekt losowości.

4.6. PODZIAŁ POPULACJI NA PRAWIE IZOLOWANE DEMY

Algorytmy genetyczne mogą dostarczyć satysfakcjonującego rozwiązania bez konieczności czasochłonnego przeszukiwania całej przestrzeni rozwiązań, jednakże i tu, do przyspieszenia obliczeń, coraz częściej stosuje się przetwarzanie równoległe. W różnych architekturach równoległych ewoluująca populacja jest dzielona na pewną liczbę prawie izolowanych demów, które ewoluują niezależnie, wymieniając od czasu do czasu pojedyncze osobniki między sobą⁸. Pierwsze z nasuwających się pytań to: na ile demów powinna być podzielona populacja, by ewolucja przebiegała możliwie najefektywniej? Kolejne pytanie dotyczy wpływu różnych zaawansowanych operatorów genetycznych na tempo i charakter ewolucji podzielonej populacji. Interesuje nas zysk czasowy, wynikający nie tyle z wykorzystania np. wieloprocesorowych komputerów, ile w sensie mniejszej liczby pokoleń, wynikającej z samego faktu optymalnego podziału populacji i zastosowania specyficznych operatorów. Badania takie powinny inspirować do poszukiwań skutecznych sposobów zrównoleglania algorytmów genetycznych.

Dalej przedstawiono eksperymenty mające na celu sprawdzenie, czy sam podział populacji na pewną liczbę prawie izolowanych podpopulacji, czyli demów, pozwala skrócić czas ewolucji, mierzony liczbą pokoleń (Kwaśnicka i Markowska-Kaczmar 1996). Te same eksperymenty przeprowadzono dla klasycznego **GA** z kodowaniem binarnym (*H-model*) i dla modelu z nadmiarowością genów, plejotropowością i poligenicznością. Badania symulacyjne wykonano dla funkcji Q_1 , populacja początkowa znajdowała się w lokalnym optimum, liczyła 100 osobników.

4.6.1. SYMULACJA Z *H*-MODELEM

H-model jest to klasyczny algorytm genetyczny. Przyjęto, że każda cecha fenotypowa będzie kodowana na 32 bitach. Cała populacja składa się ze stałej liczby osobników N . Populacja może być podzielona na pewną liczbę demów o rozmiarach $N_1, N_2, \dots, N_m$, gdzie $N = \sum N_i$. Zaimplementowano dwie metody selekcji (działające na osobnikach w obrębie demów): ruletka i stochastyczna według reszt bez powtórzeń oraz standardowe operatory mutacji i krzyżowania. Krzyżowanie zachodzi pomiędzy dwoma osobnikami z tego samego demu. Jedynym dodatkowym operatorem jest migracja. **Migracja** oznacza przejście osobnika z danego demu do innego. Migrować może najlepszy osobnik z danego demu lub też losowy, demem docelowym dla migrującego osobnika może być dem o najwyższej średniej jakości lub losowy. Aby zachować stałą liczbę osobników w demach, migrujący osobnik zastępuje w docelo-

⁸ Patrz *Równoległe przetwarzanie algorytmów genetycznych*, p. 1.4.

wym demie osobnika najgorszego lub losowego. Na rysunku 4.73 pokazano okno ustalania parametrów dla *H*-modelu.

W badaniach założono, że w każdym pokoleniu, średnio u 80% wszystkich osobników w populacji występuje mutacja pojedynczego bitu w części kodującej poszczególne fenoty, oraz 80% osobników ma szansę podlegać krzyżowaniu z losowo wybranym osobnikiem ($P_m = P_c = 0,8$). Wartości te zostały dobrane eksperymentalnie, dały najlepsze rezultaty dla róż-

Liczba osobników subpopulacji	
1	20
2	20
3	20
4	20
5	20

Rys. 4.73. Ustalanie parametrów *H*-modelu

nych wartości pozostałych parametrów. Były one stosowane we wszystkich eksperymentach z tym modelem. Prawdopodobieństwo migracji było zmieniane od 0,2 do 0,8, z krokiem 0,1. Częstość migracji ma niewielki wpływ na szybkość ewolucji, dla wartości powyżej 0,2 nie widać istotnej różnicy. Zbyt rzadkie migracje powodują brak wymiany między demami, co jest równoważne ewolucji niezależnych populacji. Warunek zatrzymania ewolucji to znalezienie globalnego szczytu przynajmniej przez jeden dem. Symulację powtarzano minimum dziesięciokrotnie dla każdego warunków. Rezultaty wskazują, że tempo ewolucji zależy od takich parametrów, jak: który osobnik migruje (najlepszy czy losowy); do którego demu migruje (najlepszego czy losowego); oraz który osobnik jest zastępowany (losowy czy najgorszy).

Jak pokazują wyniki eksperymentów, podział populacji na niewielką liczbę demów (od dwóch do czterech) powoduje przyspieszenie ewolucji (mierzone liczbą pokoleń). Dalszy podział populacji na demy o małej liczbie osobników wydłuża czas ewolucji. Na przykład, przy prawdopodobieństwie migracji równym 0,8, przy migracji najlepszego osobnika, do najlepszego demu, zastępowania w nim najgorszego osobnika, średnia liczba pokoleń potrzebna na znalezienie najwyższego szczytu wynosiła:

- dla całej populacji stuelementowej: 361,
- dla populacji podzielonej na dwa demy: 226,
- dla populacji podzielonej na trzy demy: 242,
- dla populacji podzielonej na cztery demy: 490.

Dla innych wartości prawdopodobieństw mutacji, krzyżowania i migracji otrzymywano inne wyniki, ale tendencje pozostawały te same.

4.6.2. SYMULACJA Z *K*-MODELEM

Prowadzono również eksperymenty z *K*-modelem w jego wersji stosowanej we wcześniejszych badaniach. Zaimplementowane są dwa mechanizmy wzajemnych oddziaływań demów na siebie: jeden to migracja, drugi – przy podziale populacji na

demy operator tranzycji może spowodować przeniesienie pojedynczego genu na osobnika należącego do innego demu, aczkolwiek większe szanse ma przenieść go na osobnika z tego samego demu. Jeśli gen jest przenoszony do innego demu, to prawdopodobieństwo przyłączenia się genu do osobnika z danego demu jest jednakowe dla wszystkich demów i wynosi $\text{jeden}/(\text{liczba_demów} - \text{jeden})$. Migracja jest losowana dla każdego osobnika w demie, zachodzi zgodnie z zadaniem prawdopodobieństwem.

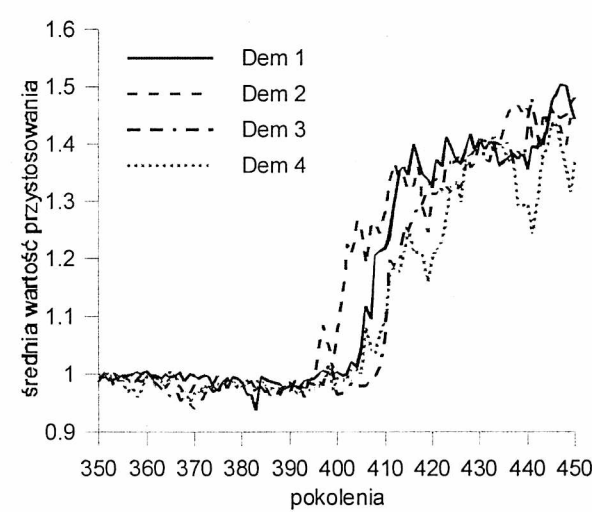
Podział populacji na różną liczbę równolicznych demów

Parametry symulacji: $P_{\text{rkm}} = 0,02$, $P_{\text{mut}} = 0,002$, $P_{\text{trs}} = 0,01$, $P_{\text{trz}} = 0,005$, $P_{\text{mig}} = 0,1$, $M_{\text{max}} = 600$

W tabeli 4.8 podano rezultaty uzyskane w tym eksperymencie. Jak widać, wyraźnych różnic tu nie obserwujemy, charakterystyczny jest wzrost czasu szukania optimum przy podziale populacji na dużą liczbę równolicznych, małych demów. Wydaje się, że w tych eksperymentach optymalny podział populacji to dziesięć demów.

Tabela 4.8. Wyniki ewolucji populacji podzielonej na demy, przy istnieniu genów nadmiarowych, efektu pleiotropowego i poligenicznego

Liczba ewoluujących demów	1	2	3	8	10	15
Średni czas (w pokoleniach) potrzebny do zasiedlenia nowego szczytu przystosowawczego	340	310	385	345	278	363
Odchylenie standardowe (w pokoleniach)	94	72	56	104	163	62
Liczba przebiegów symulacji	15	12	10	10	10	10
Liczba niepowodzeń	6	4	2	3	2	3



Rys. 4.74. Zmiany średniej jakości czterech ewoluujących demów

Zmiany średnich jakości ewoluujących demów przy podziale populacji na cztery demy pokazano na rysunku 4.74. Ponieważ istnieje wymiana informacji genetycznej pomiędzy demami (migracja osobników i tranzycja genów), znalezienie lepszego rozwiązania w jednym demie jest propagowane do innych demów, które bardzo szybko zasiedlają lepszy szczyt.

Zmiana częstości migracji na $P_{\text{mig}} = 0,3$

Pozostałe parametry są takie same jak poprzednio, ewolucja

również przebiega podobnie jak poprzednio. Populacja podzielona na dwa demy osiąga optymalny szczyt średnio po 348 pokoleniach, odchylenie standardowe wynosi 88 pokoleń, 3 przebiegi nie kończą się sukcesem. Przy podziale populacji na 10 demów, potrzeba jest średnio 270 pokoleń na wejście na globalny szczyt, odchylenie standardowe wynosi 101 pokoleń, 2 przebiegi nie kończą się sukcesem. Jakościowo wyniki te są podobne do tych, które były otrzymywane dla *H*-modelu.

4.6.3. SYMULACJA ZE ZMODYFIKOWANYM K-MODELEM

K-model został dostosowany do potrzeb niniejszych badań tak, aby umożliwiał testowanie różnych podziałów populacji, różne sposoby migracji, różne operatory dla poszczególnych demów oraz aby czas symulacji był akceptowalny.

Parametry podstawowego modelu tak dobrano, by w ciągu 2000 pokoleń możliwe było zasiedlenie przez populację wysokiego szczytu, dlatego też inna jest struktura genotypu, wartości genów i macierz *A*. Wartość przystosowawcza jest skalowana w każdym pokoleniu, co ułatwia przechodzenie przez doliny przystosowawcze i skupianie się na szczycie⁹.

Struktura genotypu to 4 chromosomy z rozkładem genów fenotypowych jak w tabeli 4.9. Wartości genów populacji początkowej to:

[1000, 8000, 5000, 3500, 2000, 7500, 1000, 2500, 4000, 4500],

natomiast macierz *A* to:

$a_{1,j}$: 0,555, 0,000, 0,270, 0,000, 1,020, 0,000, -0,500, -0,182, 0,500, 0,000,

$a_{2,j}$: 0,300, 0,500, 0,000, -0,200, 1,000, -0,200, 0,000, 0,000, 0,000, 0,200

Tabela 4.9. Rozkład genów fenotypowych

Numer chromosomu	Liczba genów fenotypowych
1	3
2	2
3	3
4	2

Dobór tych parametrów zapewnia umieszczenie całej populacji początkowej w lokalnym szczycie. Maksymalna zmiana mutacyjna jest stosunkowo duża i wynosi $M_{\max} = 6000$. Selekcja osobników do reprodukcji jest dokonywana metodą Poissona, rozmiar demów oscylował wokół zadanego poziomu nasycenia. Początkowa liczba osobników w demach zawsze była równa ich poziomowi nasycenia, parametr $\alpha = 0,001$. Wartości pozostałych parametrów to $P_{\text{mut}} = 0,02$, $P_{\text{rkm}} = 0,01$, $P_{\text{trz}} = 0,0055$, $P_{\text{trs}} = 0,01$. Jako warunek zatrzymania ewolucji ustalono:

⁹ Zastosowano proste skalowanie – w każdym pokoleniu do funkcji Q_1 dodawana jest wartość C , która jest maksymalną wartością przystosowania uzyskaną w poprzednich pokoleniach, co powoduje, że doliny stają się płytsze.

- (1) $Q_{\text{sr populacji}} \geq 1,35$ lub 2000 pokoleń, warunek dotyczy całej populacji,
 (2) $Q_{\text{sr demu}} \geq 1,35$ lub 2000 pokoleń, warunek ma być prawdziwy przynajmniej dla jednego demu.

Podział populacji na równoliczne demy

Symulacje powtarzano przynajmniej dziesięciokrotnie, ewolucję uznawano za kończącą się niepowodzeniem, jeśli w tym czasie średnie przystosowanie populacji nie osiągnęło wartości 1,35.

Tabela 4.10. Ewolucja populacji podzielonej na równoliczne demy

Zatrzymanie przy średniej wartości jakości całej populacji				Zatrzymanie przy średniej wartości jakości jednego demu			
liczba demów	liczba osobników w demie	średni czas zatrzymania (pokolenia)	liczba niepowodzeń	liczba demów	liczba osobników w demie	średni czas zatrzymania (pokolenia)	liczba niepowodzeń
1	300	1110	4	1	300	1110	4
3	100	707	6	3	100	1354	6
5	60	887	4	5	60	1154	3
10	30	1120	7	10	30	1099	5
30	10	1525	9	30	10	402	0
60	5	–	10	60	5	115	0

Prosty podział całej populacji na kilka równolicznych demów może spowodować skrócenie czasu koniecznej ewolucji, ale dalszy jej podział – na zbyt dużo drobnych demów – może spowolnić proces ewolucji w stosunku do całej populacji. Próbuując wyjaśnić to zjawisko, przeprowadzono symulację, gdzie zadaniem było uzyskanie wysokiego przystosowania przez jeden dem. Tu rezultaty dla małych i dużych demów znacznie się różnią. Zgodnie z oczekiwaniami, mniejsze populacje zajmują mały obszar w przestrzeni fenotypowej, dużą rolę odgrywa dryf genetyczny – taka populacja może zatem wędrować po szczytach i dolinach, jak zawędruje w okolice szczytu, to jej średnia wartość przystosowawcza może być w miarę wysoka, ale – jak wiadomo¹⁰ – dem długo się nie utrzyma na szczycie.

Dlaczego zatem cała populacja nie ewoluuje wtedy szybciej (w sensie średniego przystosowania), gdy jest podzielona na dużą liczbę małych demów? Wydaje się, że można to wyjaśnić nałożeniem się efektu migracji i dryfu małych populacji. Efekt dryfu zaczyna dominować nad pozytywnym wynikiem migracji – rozprzestrzenianiem się dobrego osobnika z najlepszego demu do innych demów, w których mógłby rozpocząć zasiedlanie szczytu.

Wyróżnienie w populacji jednego demu większego i kilku małych

Wydaje się, że taka struktura populacji powinna łączyć stabilizacyjne działanie selekcji w większym demie z mniejszą presją selekcyjną i dużym dryfem w małych

¹⁰ Patrz rozdział *Liczebność populacji i ewolucja w K-modelu*.

demach, co powinno dać przyspieszenie. Małe demy powinny stosunkowo szybko znajdować dobre rejony, rozwiązania z tych demów powinny trafiać poprzez migrację do większego demu, gdzie będą „eksploatowane”, stopniowo wypierając gorsze osobniki.

Zastosowano trochę inny model migracji – migrujący osobnik przyłącza się do danego demu z prawdopodobieństwem proporcjonalnym do jego rozmiaru. Prawdopodobieństwo przyłączenia się osobnika do demu i wynosi: $liczba_osobników_w_demie_i / rozmiar_populacji$.

Eksperymenty przeprowadzono dla czterech struktur populacji:

1. Jeden dem liczący 270 osobników i 6 demów po 5 osobników,
2. Jeden dem liczący 240 osobników i 12 demów po 5 osobników,
3. Jeden dem liczący 270 osobników i 3 demy po 10 osobników,
4. Jeden dem liczący 240 osobników i 6 demów po 10 osobników.

Prawdopodobieństwo migracji ustalano na:

1. $P_{mig} = 0,01$ dla wszystkich demów,
2. $P_{mig} = 0,05$ dla wszystkich demów,

Otrzymane wyniki zestawiono w tabeli 4.11. Powtarza się obserwacja, że tworzenie zbyt dużo małych demów nie jest opłacalne. Jeden dem duży z sześcioma małymi po 5 osobników lepiej sobie radził niż z dwunastoma takimi samymi. Podobnie 3 demy dziesięcioelementowe z jednym dużym dają lepsze efekty niż sześć takich demów z tym samym dużym. Uruchomiono również dziesięciokrotnie symulację z takimi samymi strukturami populacji, ale z różnym prawdopodobieństwem migracji dla elementów z dużego demu ($P_{mig} = 0,005$) i dla osobników z małych demów ($P_{mig} = 0,05$).

Tabela 4.11. Ewolucja populacji podzielonej na jeden duży dem i kilka małych

P_{mig}	0,01		0,05	
	Średni czas zatrzymania	Liczba niepowodzeń	Średni czas zatrzymania	Liczba niepowodzeń
270 + 6×5	696	0	294	3
240 + 12×5	942	0	937	1
270 + 3×10	481	4	478	4
240 + 6×10	649	2	906	5

Wyniki były różne od tych, jakie uzyskiwano przy jednakowym prawdopodobieństwie migracji dla wszystkich demów. W tym przypadku najlepsze efekty (średni czas: 391, 0 niepowodzeń) uzyskano dla największej liczby małych demów (240 + 12×5), podczas gdy przy sześciu demach 5-elementowych czas ten wynosił 768, jedno niepowodzenie. Przy liczności tych małych demów równej 10 nadal populacja z jednym dużym demem i trzema 10-elementowymi ewoluuje szybciej (577 pokoleń, 2 niepowodzenia) niż z sześcioma małymi demami (601, 2 niepowodzenia).

Różne parametry operatorów genetycznych dla poszczególnych demów

Przeprowadzono dziesięć symulacji, w których populacja składała się z dwóch demów o różnych operatorach mutacji (przy $P_{mig} = 0,01$):

Dem1: liczba osobników: 100, $P_{mut} = 0,05$, $M_{max} = 12000$.

Dem 2: liczba osobników: 10, $P_{mut} = 0,002$, $M_{max} = 500$.

Dem pierwszy jest tutaj demem poszukującym, ma duże możliwe zmiany mutacyjne. Dem mniejszy jest demem wspinającym się na zbocze szczytu, na którym znajduje, lub dostanie z demu poszukującego, dobre rozwiązanie. Ma on małe prawdopodobieństwo mutacji, jego osobniki będą zatem bardziej skupione. Są to własności, które pozwalają wspiąć się wysoko na szczyt, ale nie dają możliwości szerokich poszukiwań.

W tym przypadku rozwiązanie jest uzyskiwane bardzo szybko, średnio w ciągu 45 pokoleń, wyniki są dość zbliżone we wszystkich próbach (odchylenie standardowe wynosi 5).

Wykonane symulacje, zarówno z klasycznym modelem, stosującym binarnie kodowane chromosomy (H -modelem), jak i z rozszerzonym modelem ewolucji (K -modelem) nie dają jednoznacznej odpowiedzi na postawione w niniejszym rozdziale pytanie, tzn. na ile demów podzielić populację, by uzyskać najefektywniejszą ewolucję, w sensie liczby pokoleń potrzebnych do osiągnięcia globalnego szczytu przystosowawczego. Uzasadnione wydaje się stwierdzenie, że podzielenie populacji na stosunkowo niewielką liczbę demów może zwiększyć tempo ewolucji. Dla pierwszego modelu, z binarnym kodowaniem, dla funkcji Q_1 , poprawę daje podzielenie populacji na 2, 3 lub 4 demy. Dla K -modelu poprawę uzyskuje się przy podziale populacji na większą liczbę demów (około dziesięć).

Wydaje się jednak, że podzielenie populacji na różne demy, tzn. o różnej liczebności i różnych operatorach genetycznych, zmienianych w trakcie ewolucji, być może nawet o różnych sposobach kodowania, może dać znaczne zwiększenie skuteczności ewolucji. Jest to przedmiotem rozpoczętego, nowego tematu, dla którego prezentowane wyżej badania były inspiracją.

5. EWOLUCJA Z WIELOWYMIAROWYMI I WIELOMODALNYMI FUNKCJAMI

Znalezienie rozwiązania przez ewoluującą populację jest tym trudniejsze, im więcej jest lokalnych optimum oraz im więcej zmiennych należy dostosić. Widać to w eksperymentach z funkcją dwu-, pięcio-, dziesięcio- i trzydziestowymiarową. W K -modelu istnieje możliwość ukierunkowywania ewolucji przez odpowiedni dobór macierzy pleiotropowości A (przejścia z genotypów na fenotyp). Przed przystąpieniem do eksperymentów z różnorodnymi funkcjami przystosowania spróbujemy pokazać, jakie zmiany fenotypowe daje macierz A przy losowych zmianach genów.

5.1. PLEJOTROPOWOŚĆ I POLIGENICZNOŚĆ W UKIERUNKOWYWANIU EWOLUCJI

W K -modelu wektor fenów jest kombinacją liniową wektora składającego się z genów fenotypowych. Wartość kodowanych przez genom fenów zależy zarówno od genów, jak i od elementów macierzy A . Nasuwa się pytanie, jaki wpływ może mieć postać macierzy A na tempo i charakter ewolucji, przy czym w rozróżnianiu „postaci macierzy A ” bierzemy pod uwagę takie jej cechy, jak:

- liczbę i rozmieszczenie elementów zerowych w macierzy,
- znak elementów macierzy A , tzn. czy wszystkie elementy są tego samego znaku, czy nie,
- jeśli występują elementy ujemne i dodatnie, to jaka jest ich liczba, jak są rozmieszczone i na ile różne są ich wartości (wartość bezwzględna elementu macierzy decyduje o mocy, z jaką wpływa odpowiadający mu gen na dany fen).

Aby możliwe było analizowanie czy, i ewentualnie w jaki sposób, postać macierzy A wpływa na przebieg ewolucji, dobrze będzie wyróżnić kilka podstawowych typów macierzy A i wykonać badania symulacyjne dla każdego typu A .

Wyróżnimy 4 podstawowe typy macierzy pleiotropowości A :

1. Macierz składająca się tylko z dodatnich elementów – **typ 1** (o zbliżonych wartościach, tzw. jednorodna macierz dodatnia oraz tzw. macierz niejednorodna, zawierająca istotnie różniące się co do wartości elementy).

2. Macierz składająca się tylko z ujemnych elementów – **typ 2** (tzw. jednorodna i niejednorodna macierz ujemna).

3. Macierz zawierająca równomiernie rozłożone elementy zerowe, dodatnie i ujemne – **typ 3** (może być jednorodna lub nie, w jednorodnej sumy elementów macierzy wpływających na poszczególne fenoty mają zbliżone wartości).

4. Macierz prowadząca do zależności jeden gen – jeden fen, **typ 4** (tzw. macierz kodowania bezpośredniego, eliminuje plejotropowość). Podobnie jak poprzednio, będziemy mówić o macierzy jednorodnej, jeśli poszczególne jej elementy mają zbliżone wartości.

Eksperymentując z K -modelem zauważono, że dla tych samych warunków początkowych ewolucja przebiega inaczej w zależności od macierzy A . Nasuwa to sugestię, że dobierając odpowiednio elementy tej macierzy możemy odpowiednio „kanalizować” ewolucję. Aby zweryfikować tę hipotezę przeprowadzono poniższe eksperymenty.

Założono stosunkowo prosty genotyp, składający się tylko z czterech genów fenotypowych $[g_1, g_2, g_3, g_4]$, i dwuwymiarową przestrzeń fenotypową $[p_1, p_2]$, gdzie każdy fen zależy od genów zgodnie z poniższym wzorem

$$[p_1, p_2] = [g_1, g_2, g_3, g_4] \times \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \\ a_{4,1} & a_{4,2} \end{bmatrix}. \quad (5.1)$$

W eksperymencie tym nie było żadnej selekcji, nie jest istotna też funkcja przystosowania. Interesuje nas jak zmienia się fenotyp osobnika w czasie ewolucji, jeśli jego geny będą losowo zmieniane (jak podczas mutacji) przy różnych postaciach A . Innymi słowy, jeśli nie ma konkurencji, to jakie przestrzenie może penetrować osobnik przy różnych A . Należy zapewnić tę samą sekwencję genów dla różnych macierzy współczynników A . Sprawdzano ruch osobnika w przestrzeni fenotypowej przy różnych warunkach:

Różne wartości początkowe genów

- (a) wszystkie geny mają tę samą wartość początkową [1000, 1000, 1000, 1000],
- (b) różne wartości początkowych genów [4000, 1000, 1500, 8000].

Różna liczba zmienianych genów w pojedynczym kroku

- (i) w jednym kroku zmieniany jest jeden, losowo wybrany gen,
- (ii) każdy gen może niezależnie ulec losowej zmianie w jednym kroku.

Różny zakres jednorazowej zmiany genu

I – zmiana genu o losową wartość, przy czym wartość bezwzględna maksymalnej zmiany mutacyjnej w jednym kroku jest mała w stosunku do początkowych wartości genu (wynosi 10 dla przypadku (a) i 100 dla (b)),

II – zmiana genu o losową wartość, przy czym wartość bezwzględna maksymalnej zmiany mutacyjnej w jednym kroku jest stosunkowo duża w odniesieniu do początkowych wartości genów (odpowiednio 100 i 500 dla (a) i (b)),

III – zmiana zawsze wynosi plus lub minus 10% aktualnej wartości genu.

Różne postacie macierzy A

Przy doborze macierzy starano się, by osobnik w pierwszym kroku był mniej więcej w tym samym miejscu w przestrzeni fenotypowej, w punkcie (100, 100). Wymagało to zdefiniowania różnych macierzy dla przypadków (a) i (b). Do uwzględnienia w A jednego elementu istotnie większego od innych trudno było znaleźć elementy A dające ten sam punkt początkowy w przestrzeni fenotypowej. Przyjęto, że dla tych przypadków będą to okolice punktu (500, 500). Dla wszystkich kombinacji wymienionych wyżej parametrów, tj. (a-i-I), (a-i-II), ..., (b-ii-III) analizowano położenie fenotypu przy 10 różnych macierzach. Otrzymywane wyniki były podobne jakościowo dla wymienionych postaci A , zaprezentowane zatem zostaną tylko wyniki charakterystyczne w poszczególnych grupach.

Typowe macierze stosowane do testowania dla genów początkowych o różnych wartościach, punkt (i), to:

$$A_1 = \begin{bmatrix} 0,00626 & 0,0628 \\ 0,02523 & 0,0248 \\ 0,01669 & 0,01676 \\ 0,00309 & 0,003119 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0,03744 & -0,02 \\ 0 & 0,08916 \\ 0,100167 & 0 \\ -0,025 & 0,011355 \end{bmatrix} \quad A_3 = \begin{bmatrix} 0,02 & -0,02 \\ 0 & 0,08916 \\ 0,2 & 0 \\ -0,035 & 0,011263 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0,01248 & 0 \\ 0 & 0,06499 \\ 0,03339 & 0 \\ 0 & 0,00625 \end{bmatrix} \quad A_5 = \begin{bmatrix} 0,03095 & 0,01659 \\ 0,12632 & 0,3 \\ 0,08319 & 0,04467 \\ 0,01564 & 0,00083 \end{bmatrix}$$

A_1 – macierz jednorodna dodatnia (typ 1),

A_2 – macierz zawierająca zera, ujemne i dodatnie wartości, jednorodna (typ 3),

A_3 – macierz zawierająca zera, ujemne i dodatnie wartości, mniej jednorodna (typ 3),

A_4 – macierz bez poligeniczności, dość jednorodna (typ 4),

A_5 – macierz zawierająca zera, ujemne i dodatnie wartości zawiera element istotnie większy od pozostałych (niejednorodna, typ 3).

Typowe macierze stosowane do testowania dla genów początkowych o różnych wartościach, punkt (ii), to:

$$A_6 = \begin{bmatrix} 0,0249 & 0,025 \\ 0,0252 & 0,0249 \\ 0,025 & 0,025 \\ 0,0248 & 0,0251 \end{bmatrix} \quad A_7 = \begin{bmatrix} 0,05976 & -0,025 \\ 0 & 0,06284 \\ 0,06024 & 0 \\ -0,02 & 0,06216 \end{bmatrix} \quad A_8 = \begin{bmatrix} 0,06493 & -0,03 \\ 0 & 0,2 \\ 0,06507 & -0,0346 \\ -0,035 & -0,0354 \end{bmatrix}$$

$$A_9 = \begin{bmatrix} 0,04995 & 0 \\ 0 & 0,04996 \\ 0,050005 & 0 \\ 0 & 0,05004 \end{bmatrix} \quad A_{10} = \begin{bmatrix} 0,25 & 0,126117 \\ 0,05012 & 0,125183 \\ 0,04988 & 0,12355 \\ 0,2 & 0,12515 \end{bmatrix}$$

$A_6, A_7, A_8, A_9, A_{10}$ – macierze o typach jak A_1 – A_5 , odpowiednio.

Korzystając z generatora liczb pseudolosowych wyznaczono sekwencje genów zmienianych zgodnie z regułami I, II i III. Długość sekwencji to 200 kroków. Poniżej pokazano kilka pierwszych wektorów genów z poszczególnych sekwencji dla poszczególnych przypadków (początkowe wartości genów równe/nierówne – a/b; możliwa zmiana genów jednego/wszystkich – i/ii; wielkość zmiany genu mała/duża/procentowa – I/II/III):

(a-i-I): [1000,1000,1000,1008], [1007,1000,1000,1008], [1007,1000,1000,1008], [1007,1000,1000,1014]
 (a-i-II): [1000,1096,1000,1000], [1000,1096,974,1000], [1000,1096,1016,1000], [1000,1139, 1016, 1000]
 (a-i-III): [1100,1000,1000,1000], [1100,1000,1100,1000], [1100,1000,990,1000], [1100,1000,990,900]
 (b-i-I): [4000,912,1500,8000], [4000,1002,1500,8000], [4086,1002,1500,8000], [4041,1002,1500,8000]
 (b-i-II): [4000,1000,1050,8000], [4000,1254,1050,8000], [4000,1254,1327,8000], [4000,1096,1327,8000]
 (b-i-III): [4400,1000,1500,8000], [4400,1100,1500,8000], [4400,1100,1650,8000], [4400,1210,1650,8000]
 (a-ii-I): [1000,1000,1000,1000], [1000,1000,991,1000], [1006,992,985,1005], [1013,991,983,1005], [1007,991,981,1005]
 (a-ii-II): [1000,1000,997,1069], [937,925,1050,1069], [937,869,1050,1069], [857,817,995,1042], [863,795,995,1042]
 (a-ii-III): [1100,1100,1000,1000], [1210,1100,1000,1000], [1331,1210,1100,1100], [1331,1331,1100,1210]
 (b-ii-I): [4000,1000,1500,8000], [3994,1000,1472,8000], [4065,1065,1468,8000], [4047,981,1468,8024]
 (b-ii-II): [4000,1000,1500,8000], [4365,1000,1500,8000], [4836,1000,1523,7934], [5295,625,1306,8093]
 (b-ii-III): [4000,1000,1500,8000], [4000,1000,1650,8000], [4000,1100,1650,8000], [4400,1100,1650,8000]

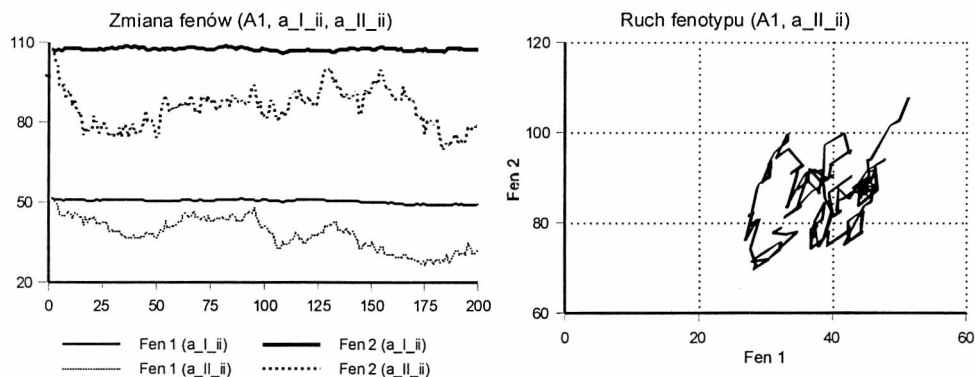
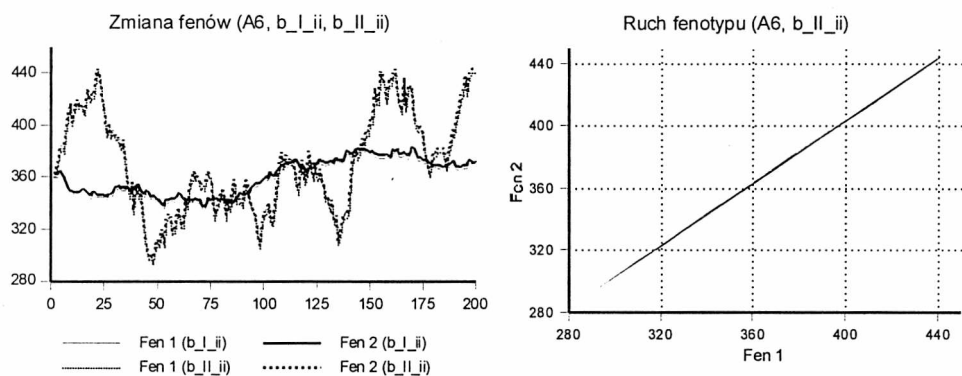
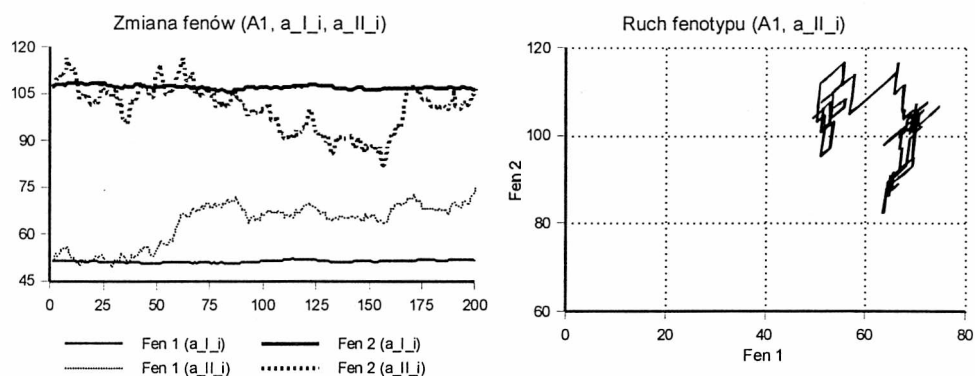
Otrzymane wyniki dla różnych macierzy A

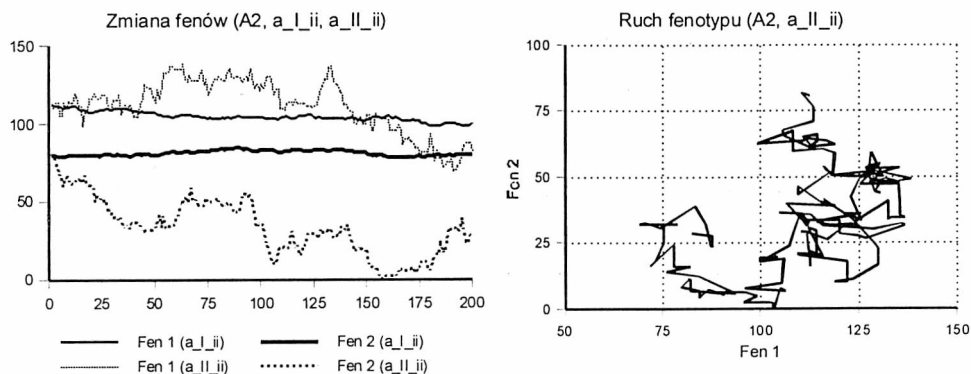
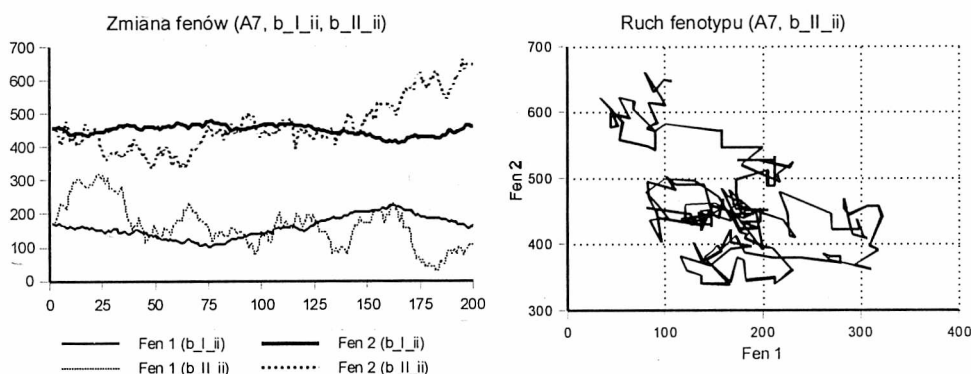
Eksperyment 1 (E1): *Dodatnie elementy A o porównywalnych wartościach.*

Gdy macierz A zawiera tylko dodatnie elementy o porównywalnych wartościach (jednorodna, typ I), osobnik przesuwają się średnio po przekątnej na płaszczyźnie fenotypowej. Kierunek zmian zależy w dużej mierze od tego, czy przewagę mają mutacje zwiększające czy zmniejszające wartości genów. Dla macierzy składającej się z samych ujemnych elementów sytuacja jest podobna, z tym, że mutacje zmniejszające wartości genów będą powodować wzrost wartości fenów. W dwustu krokach fenotyp testuje stosunkowo niewielki obszar przestrzeni (rys. 5.1–5.3). Gdy w jednym kroku zmieniany jest tylko jeden gen, wyniki nie różnią się jakościowo, zmniejsza się zakres zmian fenotypowych. Taka macierz A nie wydaje się efektywna dla funkcji wielomodalnych, w których lokalne szczyty leżałyby wzdłuż jednego z argumentów. Należy zauważyć, że rekombinacje mogą zmienić ten obraz.

Eksperyment 2: *Elementy ujemne, dodatnie oraz zerowe, macierz stosunkowo jednorodna.*

Macierze A_2 i A_7 zawierają wartości ujemne, dodatnie i zerowe, są dość jednorodne. Na rysunkach 5.4 i 5.5 pokazano wyniki uzyskane dla tych macierzy. Tutaj zmiany fenotypu mają inny charakter. Trajektoria fenotypu zajmuje dużo większy obszar, zakresy zmian poszczególnych fenów są porównywalne. Osobnik jest w stanie testować większą część przestrzeni fenotypowej niż przy macierzy jednorodnej, dodatniej.

Rys. 5.1. Zmiany fenotypu przy macierzy A_1 Rys. 5.2. Zmiany fenotypu przy macierzy A_6 Rys. 5.3. Zmiany fenotypu przy macierzy A_1 (w jednym kroku zmiana pojedynczego genu)

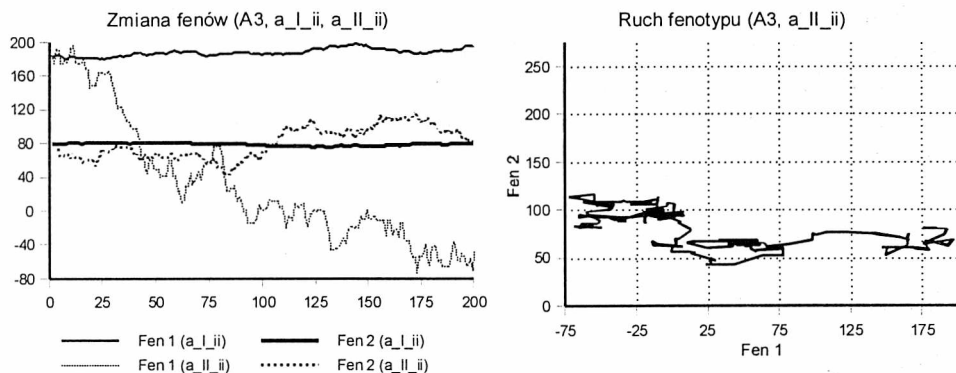
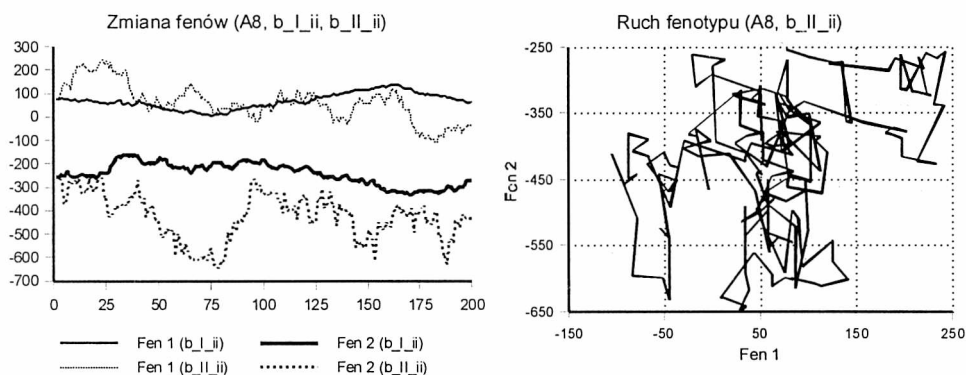
Rys. 5.4. Zmiany fenotypu przy macierzy A_2 Rys. 5.5. Zmiany fenotypu przy macierzy A_7

Eksperyment 3: *Elementy ujemne, dodatnie oraz zerowe, macierz o stosunkowo zróżnicowanych elementach.*

Macierz A_3 jest dość podobna do A_2 , ale elementy wpływające na poszczególne fenotypy w większym stopniu różnią się wartościami. A_3 promuje pierwszy fen, ponieważ wpływa na niego element $a_{2,1}$, który jest o rząd większy od pozostałych (rys. 5.6). Ruch fenotypu przebiega więc wzdłuż osi Fen 1. Macierz A_8 ma największy element w drugiej kolumnie, drugi fen powinien zatem ulegać większym zmianom. Ale jeśli wziąć pod uwagę sumy elementów w kolumnach (tzn. wpływających na poszczególne fenotypy), to są to bardzo zbliżone wartości (0,095 i 0,1). Dlatego też widoczne w trajektorii zmiany obu fenów są porównywalne (rys. 5.7).

Eksperyment 4: *Brak efektu pleiotropowego (odpowiednie elementy a_{ij} są zerowe: A_4 i A_9).*

W tym przypadku każdy gen może wpływać tylko na jeden fen (wartość fenu jest wynikiem oddziaływania kilku genów). Jeśli poszczególne geny będą wpływać na

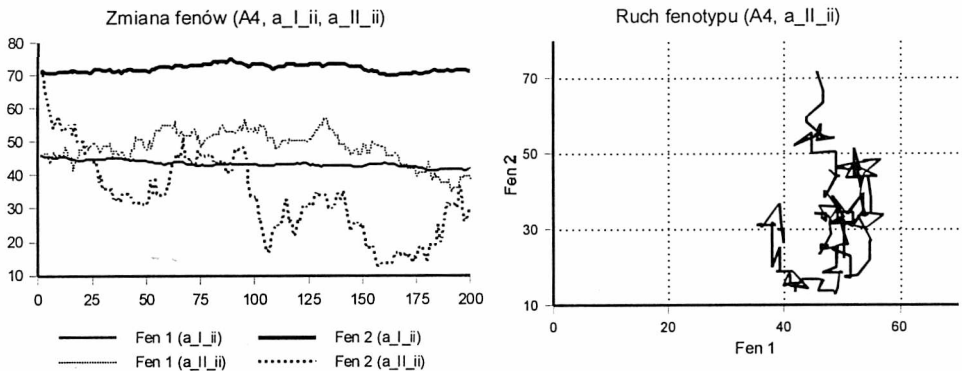
Rys. 5.6. Zmiany fenotypu dla macierzy A_3 Rys. 5.7. Zmiany fenotypu przy macierzy A_8

feny z podobną siłą, a wartości genów będą porównywalne, to ruch osobnika w przestrzeni fenotypowej będzie podobny do takiego, jak w **E1**. Jeśli natomiast w którejś kolumnie suma elementów będzie znacznie większa niż w pozostałych, to odpowiadający tej kolumnie fen będzie zmieniany szybciej niż pozostałe fen. W macierzy A_4 suma elementów w pierwszej kolumnie jest mniejsza niż w drugiej (0,046 i 0,071), początkowe wartości wszystkich genów wynoszą 1000, nic więc dziwnego, że zmiany drugiego fenu są większe (rys. 5.8). Macierz A_9 ma niemalże identyczne wartości elementów wpływających na poszczególne fen, a mimo to ruch fenotypu przebiega niemalże równoległe do osi Fen 2 (rys. 5.9). W tym przypadku wynika to z wartości genów – wartości genów wpływających na ten fen są znacznie większe niż genów decydujących o wartości pierwszego fenu.

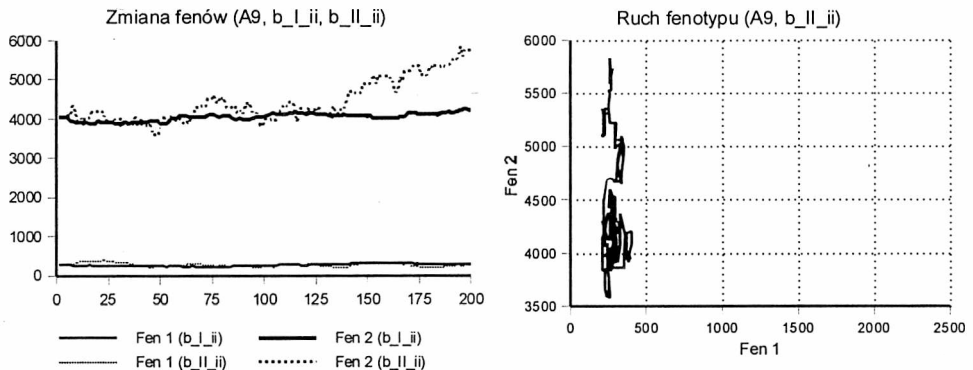
Eksperyment 5: Pełna plejotropowość i poligeniczność, różne wartości a_{ij} .

Macierze A_5 oraz A_{10} składają się tylko z dodatnich elementów, ale różniących się wartościami (o rząd), przy czym różne są sumy elementów w kolumnach: w A_5

suma elementów w drugiej kolumnie jest większa o około 30% (0,256 i 0,362), a w A_{10} suma w pierwszej kolumnie jest większa o 10% (0,55 i 0,5). Na poniższych wykresach widać, że faktycznie, dla A_5 , drugi fen ulega większemu zróżnicowaniu niż pierwszy. Natomiast dla A_{10} mocniej zmieniany jest pierwszy fen. Efekt ten jest tu wzmocniony, ponieważ większe elementy A_{10} odpowiadają większym wartościom genów.



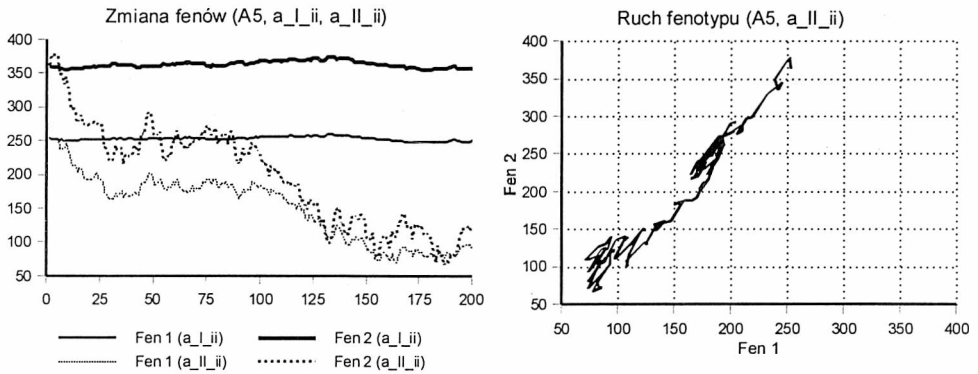
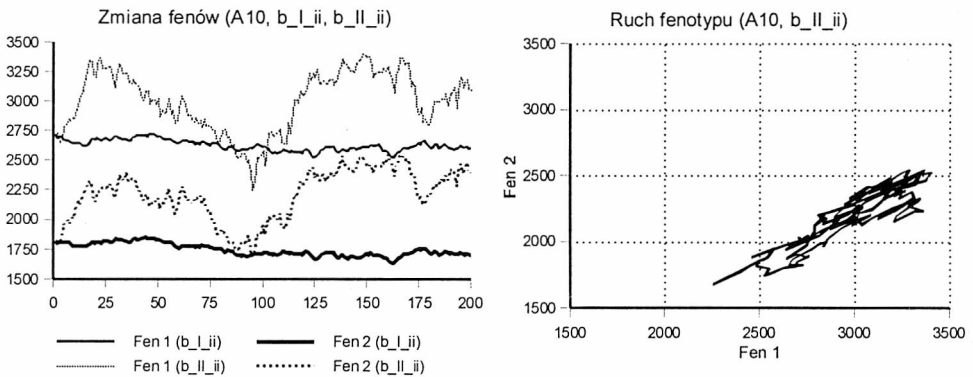
Rys. 5.8. Zmiany fenotypu przy macierzy A_4



Rys. 5.9. Zmiany fenotypu przy macierzy A_9

Podsumowanie wyników

W omawianych eksperymentach wykorzystywano 15 różnych macierzy A . Ponieważ zmiany zestawu genów są losowe, dla każdego przypadku wylosowano trzy zestawy zmieniających się genów. Dla wyróżnionych czterech typów macierzy uzyskano jakościowo podobne wyniki. Na rysunkach przedstawiono charakterystyczne w poszczególnych grupach przebiegi. Nie pokazano eksperymentów dla macierzy typu 2 (same ujemne elementy), ponieważ charakter zmian fenotypowych jest tu taki sam, jak dla macierzy typu 1 (same dodatnie elementy).

Rys. 5.10. Zmiany fenotypu dla macierzy A_5 Rys. 5.11. Zmiany fenotypu dla macierzy A_{10}

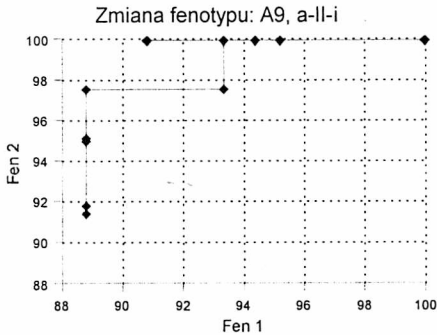
Wniosek 1 (A_1 , A_5 , A_6 i A_{10}). Zwiększenie jednego genu powoduje zwiększenie wszystkich fenów. Ruch osobnika odbywa się wzdłuż krzywej, nachylonej pod pewnym kątem do każdej współrzędnej przestrzeni fenotypowej. Kąt z j -tą współrzędną przestrzeni fenotypowej zależy od sumy (po j) współczynników a_{ij} . Przy jednorodnej macierzy trajektoria ruchu osobnika tworzy mniej więcej jednakowe kąty ze wszystkimi współrzędnymi przestrzeni fenotypów. Zarówno dla jednakowych wartości początkowych genów, jak i dla różnych, uzyskiwane wyniki nie różnią się jakościowo.

Wniosek 2 (zanegowane A_1 , A_5 , A_6 i A_{10}). Jest to ta sama sytuacja co poprzednio, tylko jest odwrócony kierunek zmian.

Wniosek 3 (A_2 , A_3 , A_7 , i A_8). Zwiększenie pojedynczego genu daje różne efekty dla poszczególnych fenów. Dla jednorodnych macierzy i genów o podobnych wartościach osobnik będzie poruszał się po dość foremnej podprzestrzeni: dla dwóch wymiarów będzie to płaszczyzna o kształcie zbliżonym do kwadratu. Rozmiar penetrowanego obszaru zależy od wartości zmian genów i wartości elementów macierzy A . Dla niejednorodnych macierzy A osobnik penetruje większy obszar wzdłuż niektórych współrzędnych prze-

strzeni fenotypowej (współrzędna ta odpowiada kolumnie zawierającej element o zdecydowanie wyższej wartości od pozostałych elementów macierzy). Należy podkreślić w tym miejscu, że przy zmianie mutacyjnej pojedynczego genu możliwe jest tu uzyskanie zwiększenia wartości jednej współrzędnej przestrzeni fenotypowej przy jednoczesnym zmniejszeniu innej współrzędnej. Dla różnych wartości początkowych genów wyniki różnią się zakresem zmian, ale jakościowo pozostają takie same.

Wniosek 4 (A_4 , A_9). Zmiana wartości pojedynczego genu powoduje ruch tylko wzdłuż pojedynczej współrzędnej osobnika w przestrzeni fenotypowej. Długość pojedynczego „kroku” zależy od wartości genu i odpowiedniego elementu macierzy A .



Rys. 5.12. Ruch osobnika w przestrzeni fenotypowej przy kodowaniu jeden gen – jeden fen

Na rysunku 5.12 pokazano punkt początkowy i pierwszych dziesięć ruchów osobnika w przestrzeni fenotypowej w eksperymencie, gdzie w jednym kroku zmieniany był losowo tylko jeden gen, a początkowe geny i maksymalna zmiana wynoszą odpowiednio 1000 i 100. Widać wyraźnie, że osobnik porusza się ruchami równoległymi do poszczególnych współrzędnych.

Wyniki analizy charakteru zmian fenotypowych w zależności od postaci macierzy poligeniczności A , to jest od wartości jej elementów i ich umiejscowienia, wydają się wskazywać, że możliwe jest „kanalizowanie” zmian ewolucyjnych. Jeśli przy rozwiązywaniu zadania optymalizacji dostępne byłyby informacje na temat położenia optimum, można byłoby zwiększać efektywność algorytmu ewolucyjnego przez dobór odpowiedniej macierzy A , definiującej plejotropowość i poligeniczność sztucznych organizmów.

5.2. WZROST WYMIARU FUNKCJI A EWOLUCJA

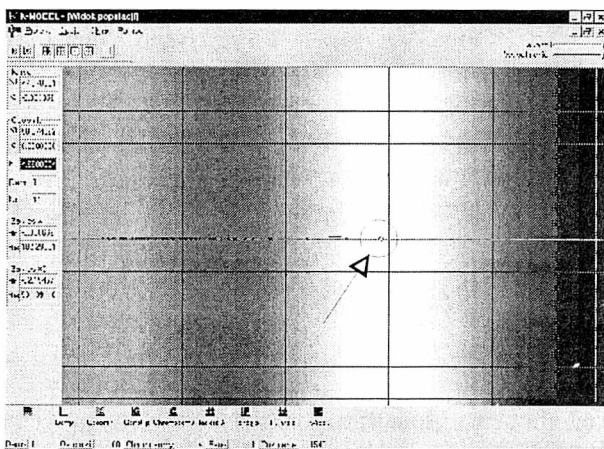
Sprawdźmy, jak K -model radzi sobie z szukaniem optimum globalnego dla rodziny funkcji eksponencjalnych o różnej liczbie zmiennych: funkcja jednej zmiennej, dwóch zmiennych, pięciu zmiennych, dziesięciu zmiennych i trzydziestu zmiennych. Za każdym razem populacja początkowa jest umieszczana w optimum lokalnym. Przedstawione wyniki nie są uśredniane z dziesięciu przebiegów. Dla każdej kombinacji parametrów uruchamiano symulację kilkakrotnie, by przekonać się o powtarzalności wyników. Jeśli w ciągu pięciu pierwszych symulacji występowały niepowodzenia (przy założonej liczbie pokoleń), symulację powtarzano więcej razy, by przedstawić tu typowe, uśrednione wyniki. Jeśli obserwowano powtarzalność wyników, rezy-

gnowano z kolejnych symulacji, chodzi bowiem tu tylko o sprawdzenie, czy K-model potrafi dać satysfakcjonujące rozwiązanie w akceptowalnym czasie.

Funkcja jednej zmiennej

Na początek sprawdzimy, czy K-model jest w stanie znaleźć optimum dla jednowymiarowej funkcji Q_5 , z bardzo oddalonymi od siebie szczytami o wysokościach 3 i 5, przy czym na początku populacja zajmuje miejsce na lokalnym szczycie. Oczywiście, „oddalenie szczytów” oznacza tu, że żadna dopuszczalna mutacja nie jest w stanie przenieść osobnika w rejon wysokiego szczytu. Wartości elementów w równomiernej dodatniej macierzy A wynosiły około 0,00024–0,00026 (typ 1) oraz od $-0,0001$ do 0,00056 w macierzy zawierającej elementy zerowe, dodatnie i ujemne (typ 3). Maksymalna zmiana genu w czasie pojedynczej mutacji to $M_{\max} = 600$, jeśli wystąpi ona u jednego genu to może spowodować zmianę fenu o $600 \cdot 5,6 \cdot 10^{-4} = 3360 \cdot 10^{-4} = 0,336$. Odległość pomiędzy szczytami wynosi 55. Nawet jeśli założymy, że wszystkie geny wpływają na fen z maksymalną siłą i w tym samym kierunku (gen może zwiększyć lub zmniejszyć swoją wartość), to i tak otrzymujemy: 4 chromosomy $\cdot$ 5 genów $\cdot$ 0,336 = 6,72, co jest dalekie od 55. Bez specjalnych makromutacji stuelementowa populacja nie opuści lokalnego szczytu. W symulacji uwzględniamy zatem geny nadmiarowe i kryzys.

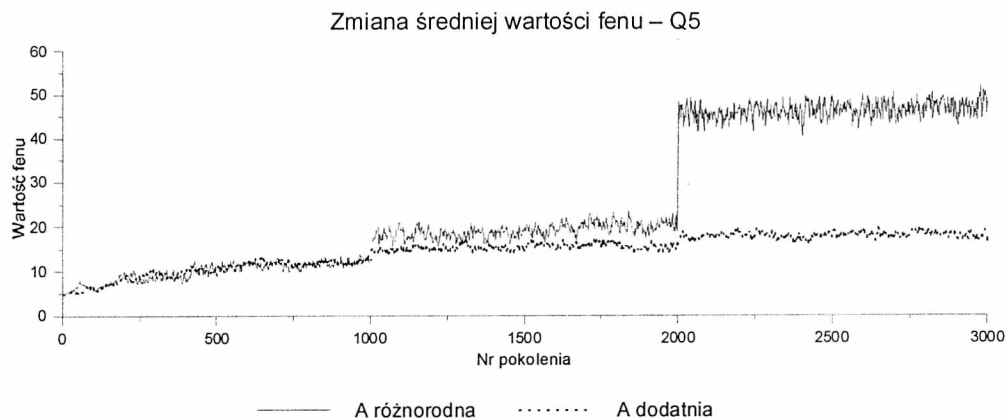
Przy parametrach symulacji: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$, $M_{\max} = 600$, $t_{\text{kr}} = 20$, $rcd = 0$, w czasie kryzysu: $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 0,5$, $P_{\text{trs}} = P_{\text{trz}} = 0,1$ oraz dodatniej jednorodnej macierzy A , populacja licząca 100 osobników potrzebowała 25 333 pokolenia na znalezienie pierwszego rozwiązania lepszego od 4,7 (znalezione rozwiązanie to: $x = 55,35$, $Q_5(x) = 4,84$). Dobór macierzy A tak, by zawierała również elementy zerowe, dodatnie i ujemne (typ 3) spowodował skrócenie czasu szukania optimum. Przy tych samych parametrach populacja potrzebowała 4813 pokoleń na pierwsze znalezienie osobnika z przystosowaniem powyżej 4,7 ($Q_5(58,144) = 4,988$ – rys. 5.13). W tych eksperymentach populacja dosłownie schodzi z niskiego szczytu, dość długo „miota się” w dolinie przystosowawczej, po czym przechodzi na wysoki szczyt. Cały proces jest stosunkowo powolny. Obrazują to zmiany średniej jakości populacji pokazane na rysunku 5.14, na którym, dla większej czytel-



Rys. 5.13. Populacja odkrywa wysoki szczyt – najbardziej na prawo wysunięty punkt to najlepszy osobnik: $Q_5 = 4,98$, $x = 58,144$ (jasny obszar oznacza wartości x na wyższym szczycie)

ności, umieszczono dane z wybranych pokoleń. Średnia jakość populacji, aczkolwiek z wahaniem, pnie się jednak w górę. Znalezione natomiast najlepsze osobniki jest możliwe dzięki makromutacjom – najlepszy osobnik powstaje zatem czasami jako dość odległy od reszty populacji (np. jak na rysunku 5.13).

Zmiany średniej wartości fenu w populacji mają podobny charakter jak zmiany średniej jakości. Populacja przesuwając swój środek masy raz w lewo, raz w prawo, ale zachowuje trend stopniowego przesuwania populacji w kierunku wysokiego szczytu.



Rys. 5.14. Zmiana średniej wartości fenu – wybrane pokolenia: pierwsze 1000 pokoleń, następnie numery 1001 do 2001 odpowiadają pokoleniom 4000–5000, numery 2002 do 3002 – pokoleniom 8400–9400

W zbiorze funkcji testowych zaproponowano drugą funkcję jednowymiarową, z przybliżonymi szczytami – funkcja Q_7 . Dla tej funkcji K-model daje następujące wyniki (populacja stuelementowa):

- przy parametrach: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,02$, $M_{max} = 100$, $t_{kr} = 0$, $rec = 0$ oraz macierzy A zawierającej elementy dodatnie, ujemne i zerowe (typ 3), pokolenie 15006: $x = 17,72$, $Q_7(x) = 1,39$, wartość średnia populacji przekracza 1,1 w pokoleniu 18 566.,

- przy parametrach: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,02$, $M_{max} = 100$, $rcd = 0$, $t_{kr} = 20$, w kryzysie: $P_{rkm} = 0,5$, $P_{mut} = 0,5$, $P_{trs} = P_{trz} = 0,1$, oraz macierzy A zawierającej elementy dodatnie, ujemne i zerowe, pokolenie 4788: $x = 18,78$, $Q_7(x) = 1,47$, wartość średnia populacji przekracza 1,1 w pokoleniu 7960.

Dla funkcji jednowymiarowej opuszczenie pułapki ewolucyjnej, jaką stanowi lokalny szczyt, nie jest łatwe. Lepsze rezultaty otrzymuje się po zastosowaniu macierzy pleiotropowości A z elementami o różnych znakach niż jednorodnej dodatniej. Należy zauważyć, że zadanie byłoby łatwiejsze, gdyby początkowa populacja nie była skupiona na lokalnym szczycie, a losowo rozłożona wzdłuż zmiennej x . Ze względu jednak na zdolności optymalizacyjne ważne jest opuszczenie zasiedlonego już, lokalnego szczytu.

Zmieniono macierz A po ustaleniu wartości pewnych, losowo wybranych jej elementów jako istotnie większe od pozostałych (różnią się około dziesięciokrotnie, jest to nadal typ 3, ale macierz jest niejednorodna). Dla tak dobranej macierzy ewolucja przy jednowymiarowej funkcji Q_5 (z odległymi szczytami) przebiega znacznie szybciej:

- przy parametrach: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,02$, $M_{max} = 100$, $rcd = 0$, $t_{kr} = 0$, pokolenie 172. – $Q_5 = 4,389$, wartość średnia populacji przekracza 4,5 w pokoleniu 195.,

- przy parametrach: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0$, $M_{max} = 100$, $rcd = 0$, $t_{kr} = 0$, pokolenie 576. – $Q_5 = 3,59$, wartość średnia populacji przekracza 4,5 w pokoleniu 613.,

- przy parametrach: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,2$, $M_{max} = 10$, $rcd = 0$, $t_{kr} = 0$, pokolenie 2382. – $Q_5 = 3,85$, rozwiązanie to jest gubione na kilkanaście do kilkudziesięciu pokoleń, średnie przystosowanie populacji nie przekracza 4,5 w ciągu 3000 pokoleń, populacja jest mocno różnorodna, nic nie wskazuje, by w najbliższych pokoleniach skupiła się.

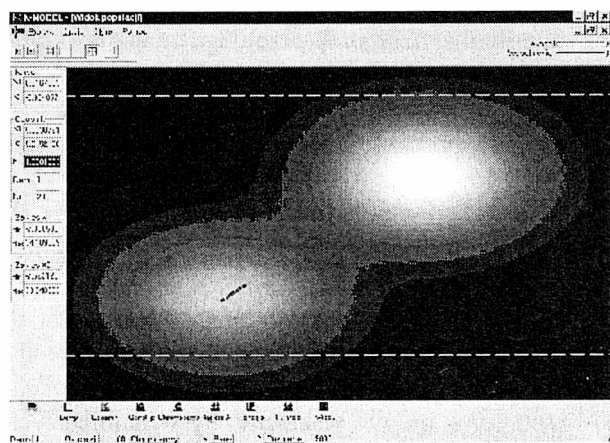
Jak widać, funkcja jednowymiarowa sprawia trochę kłopotów algorytmowi. Jest tu, co prawda, bardzo duża odległość pomiędzy szczytami (w stosunku do możliwej maksymalnej zmiany fenu), zwłaszcza w Q_5 , ale też i nie ma efektu pleiotropowości – nie może jeden gen wpływać na wiele fenów, ponieważ nasz osobnik ma tylko jeden fen.

Inną sprawą jest poprawa efektywności ewolucji obserwowana dla niejednorodnej macierzy A typu 3 (zawiera elementy ujemne, dodatnie i zerowe, część dodatnich jest znacznie większa od pozostałych). Efektu tego nie obserwuje się przy wszystkich funkcjach. W tym przypadku poruszanie się populacji w kierunku wysokiego szczytu wymaga stosunkowo dużej zmiany jednego fenu. Istotnie większy element macierzy a_i spowoduje, że zmiana i -tego genu będzie mieć znacznie większy wpływ na fenotyp niż zmiana innego genu. Może to być postrzegane jako swojego rodzaju makromutacja, przesuwająca osobnika wzdłuż osi fenu. Daje to więc pozytywny efekt. Ponieważ niewiele jest takich a_i (zacytowane wyniki uzyskano, gdy trzeci gen na pierwszym chromosomie miał około dziesięciokrotnie większy wpływ od pozostałych), nie dochodzi do bezładnego poruszania się po przestrzeni.

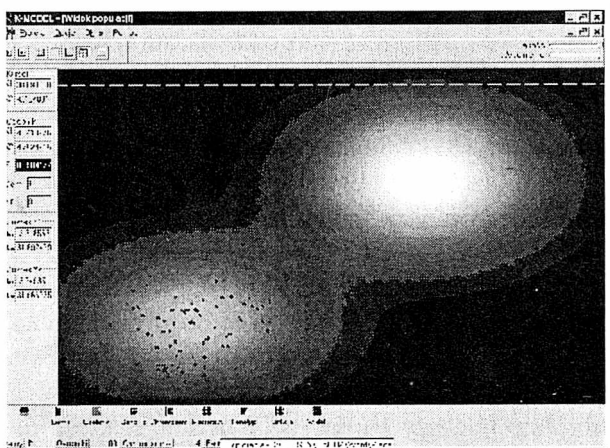
Dwuwymiarowa funkcja Q_1

Funkcja dwuwymiarowa nie sprawia większych kłopotów K -modelowi. Populacja stuelementowa znajduje wysoki szczyt nawet dla $M_{max} = 30$ przy $P_{rkm} = 0,25$, $P_{mut} = 0,1$, bez dodatkowych operatorów, przy jednorodnej macierzy A zawierającej elementy różnych znaków i zera (typ 3). W pięciu przebiegach średnio czas znalezienia rozwiązania o jakości większej niż 1,35 wynosił 3601 (wahał się od 1571 do 6329, nie ma prób kończących się niepowodzeniem). Przy mniejszej zmianie mutacyjnej i tych samych pozostałych parametrach, w ciągu 10 000 pokoleń nie obserwowano sukcesu.

Macierz A zawierająca tylko dodatnie elementy (typ 1) daje dużo gorsze efekty. Nawet przy $M_{\max} = 300$, nie ma szans na pojawienie się osobnika na zboczu wysokiego szczytu. Jest to wynik powtarzający się w każdej z kilku wykonanych prób. W 5000. pokoleniu populacja nadal zajmuje lokalny szczyt i nic nie zapowiada, by sytuacja mogła się zmienić (rys. 5.15). Włączenie kryzysu z $t_{kr} = 20$ niewiele pomaga. Macierz A wydaje się być jednym z elementów mających bardzo duży wpływ na tempo i charakter ewolucji. Dla porównania pokazano jak wygląda rozmieszczenie osobników w populacji po 50 pokoleniach, z różnorodną macierzą A i przy maksymalnej zmianie mutacyjnej wynoszącej tylko $M_{\max} = 30$ (rys. 5.16). Najlepsze znalezione rozwiązanie to $Q_1 = 1,496$. Jak widać, algorytm dość łatwo radzi sobie z tą dwuwymiarową funkcją. Duża zasługa jest w tym efektów pleiotropowego i poligenicznego oraz macierzy A .



Rys. 5.15. Populacja po 5000 pokoleń dla funkcji Q_1 , $M_{\max} = 300$, $P_{km} = 0,25$, $P_{mut} = 0,1$, z jednorodną, dodatnią macierzą A



Rys. 5.16. Populacja po 50 pokoleniach dla funkcji Q_1 , $M_{\max} = 30$, $P_{km} = 0,25$, $P_{mut} = 0,1$, z macierzą A o elementach dodatnich, ujemnych i zerowych

Jest niższy szczyt. Jeśli ma ona zawierać zera, ujemne i dodatnie wartości, to użytkownik programu ustala arbitralnie wybrane jej elementy na zera, innym nadaje wybrane przez siebie wartości ujemne, resztę program wygeneruje automatycznie. Jeśli okaże się to niemożliwe (uzyskanie zadanego punktu początkowego fenotypów), to program zgłosi ten fakt użytkownikowi. We wszystkich omawianych eksperymentach elementy zerowe i ujemne były rozmieszczane przypadkowo, reszta elementów była generowana przez program. Tak dobrana macierz A okazuje się skuteczna.

Funkcja pięciu zmiennych Q_2

Wzrost wymiaru funkcji powoduje, że zadanie dla algorytmów ewolucyjnych staje się trudniejsze. Dla funkcji pięciu zmiennych Q_2 , zmiana mutacyjna $M_{\max} = 100$ nie jest w stanie spowodować przejścia populacji na wysoki szczyt, nawet jeśli ustalimy częsty kryzys, np. $t_{kr} = 10$, po 15 tysiącach pokoleń populacja nie znajduje rozwiązania.

Przy zmianach mutacyjnych $M_{\max} = 400$, $P_{\text{mut}} = 0,1$ oraz $P_{\text{rkm}} = 0,25$ ($P_{\text{trs}} = P_{\text{trz}} = 0$, $rcd = 0$, $t_{kr} = 0$) populacja znalazła rozwiązanie na wysokim szczycie po 2218 pokoleniach (jest to najlepszy wynik z trzech symulacji, średnia to 2816). Podobne rezultaty osiągane są przy $M_{\max} = 500$ i $M_{\max} = 600$. Istotne różnice można zauważyć, gdy wprowadzimy makromutacje przez włączenie nadmiarowości genów: $P_{\text{trs}} = P_{\text{trz}} = 0,01$. Przy $M_{\max} = 500$ optimum znajdowane jest średnio w 371 pokoleniach (np. 290 czy 490 pokoleń). Najlepsze uzyskane rozwiązanie to $Q_2 = 1,47$, dla $x = [18,71, 19,66, 21,96, 19,5, 18,1]$.

W podanych eksperymentach z Q_2 nie było zbyt dużych i gwałtownych zmian fenotypów, dlatego też populacja potrafiła osiągnąć wysokie średnie przystosowanie. Przykładowo, przy $M_{\max} = 400$, warunek $Q_2 \text{ śr} > 1,35$ osiągnięty został w 2393. pokoleniu, dla $M_{\max} = 500$ i $P_{\text{trs}} = P_{\text{trz}} = 0,01$: $Q_2 \text{ śr}$ przekracza 1,35 w pokoleniach 330.–722.

Niejednorodna macierz A typu 3, zgodnie ze spostrzeżeniem przy analizie jednowymiarowej funkcji, zwiększa efekt mutacji niektórych genów. Starano się tak dobrać te elementy a_{ij} ustalone jako „mocne”, aby były rozłożone równomiernie (w każdej kolumnie macierzy był taki element). Przy $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, kryzysie $t_{kr} = 50$ (prawdopodobieństwo mutacji i rekombinacji zwiększone w kryzysie do 0,5) oraz maksymalnej zmianie mutacyjnej $M_{\max} = 100$ można znaleźć wysoki szczyt nawet po 98 pokoleniach, ale populacja jest bardzo zróżnicowana i w ciągu 2000 pokoleń nie potrafi się skupić. Dla $M_{\max} = 600$ zdarza się w ciągu kilku pokoleń znaleźć dobre rozwiązanie, ale nawet w 2000 pokoleń populacja nie potrafi się skupić na szczycie. Średnia jakość populacji nie może osiągnąć wartości 1,35, w trakcie 100–200 pokoleń osiąga wartość 1,1 – jest to wartość porównywalna z wysokością niskiego szczytu.

Jednorodna dodatnia macierz A nie daje tu pozytywnych wyników.

Funkcja dziesięciu zmiennych Q_6

Po zastosowaniu rekombinacji, mutacji i nadmiarowości genetycznej możemy uzyskać rozwiązanie dla dziesięciowymiarowej funkcji Q_6 przy maksymalnej zmianie mutacyjnej równej $M_{\max} = 600$ oraz $P_{\text{mut}} = 0,1$, $P_{\text{rkm}} = 0,25$, $P_{\text{trs}} = P_{\text{trz}} = 0,05$ ($rcd = 0$, $t_{kr} = 0$). Przy tych parametrach i stuelementowej populacji oraz macierzy A typu 3 uzyskiwano najlepsze rozwiązanie średnio po około 10 000 pokoleń (najszybciej w 8605. pokoleniu). Populacja po tak dużej liczbie pokoleń jest już mocno zróżnicowana i do 20 000 pokoleń nie jest w stanie zwiększyć średniego przystosowania powyżej wartości 1,1. W czterech symulacjach spełniających warunek zatrzymania $Q_6 \geq 1,35$, uzyskano najlepsze rozwiązanie $Q_6 = 1,386$ dla $x = [19,14, 20,45, 18,36, 21,35, 21,0, 20,77, 19,75, 20,9, 18,06, 17,03]$.

Po włączeniu kryzysu, $t_{kr} = 20$, w kryzysie $P_{rkm} = 0,5$, $P_{mut} = 0,5$, $P_{trs} = P_{trz} = 0,1$, populacja znajduje wysoki szczyt przy $M_{max} = 400$. Potrzebuje na to około 9600 pokoleń. Pozostałe typy macierzy A nie dają lepszych rezultatów. Przy niejednorodnej macierzy występuje duże zróżnicowanie osobników w populacji, ale – podobnie jak przy losowym próbkowaniu – trudno jest trafić w ekstremum.

Funkcja trzydziestu zmiennych Q_4

Optimum globalne funkcji trzydziestu zmiennych osiągnięto przy rekombinacjach zachodzących z prawdopodobieństwem $P_{rkm} = 0,25$, mutacjach z $P_{mut} = 0,1$, z wynikającymi z istnienia genów nadmiarowych makromutacjami: $P_{trs} = P_{trz} = 0,05$, dla maksymalnych zmian mutacyjnych $M_{max} = 500$ ($rcd = 0$, $t_{kr} = 0$). Przy tych wartościach parametrów i macierzy A typu 3 najlepsze rezultaty uzyskane przez stulementową populację to:

- pokolenie 7907. – $Q_{4\text{ sr}}$ przekroczyło wartość 1,1,
- pokolenie 7981. – $Q_{4\text{ max}} = 1,362$ (najlepszy osobnik ma wartość funkcji większą od 1,35),
- pokolenie 10 643. – $Q_{4\text{ sr}}$ jest większe niż 1,35 – populacja zasiedliła wysoki szczyt.

Zwiększając M_{max} do 600, powodujemy, że przy $P_{trs} = P_{trz} = 0,01$ i $t_{kr} = 0$, ewolucja przebiega szybciej i mniejsze jest prawdopodobieństwo, że w ciągu 10 000 pokoleń zakończy się niepowodzeniem:

- pokolenie 2689. – $Q_{4\text{ sr}}$ przekroczyło wartość 1,1,
- pokolenie 2800. – $Q_{4\text{ max}} = 1,3519$,
- pokolenie 3915. – $Q_{4\text{ sr}}$ jest większe niż 1,35 – populacja zasiedliła wysoki szczyt.

Przy częstszym kryzysie populacja coraz trudniej skupia się, by uzyskać wysokie średnie przystosowanie. Jest to zrozumiałe, ponieważ zbyt często cała populacja jest poddawana dużym zmianom i ciśnienie selekcyjne nie zdąża skupić osobników wokół dobrego rozwiązania.

Przykładowe rozwiązanie dla $Q_{4\text{ max}} = 1,3519$ to: $x = [22,04, 19,58, 20,94, 19,08, 20,53, 21,06, 19,51, 23,11, 18,66, 21,82, 21,00, 20,90, 17,36, 22,06, 19,52, 24,99, 20,85, 22,07, 20,33, 20,63, 13,54, 19,10, 18,73, 19,98, 20,95, 18,37, 19,81, 18,35, 15,19, 20,52]$.

Podobnie jak poprzednio, tu też trudno jest uzyskać lepsze lub nawet podobne rezultaty, stosując inne typy macierzy A niż jednorodna typu 3.

5.3. EWOLUCJA Z FUNKCJAMI TESTOWYMI DE JONGA

Zadaniem algorytmu ewolucyjnego jest znalezienie minimum funkcji znanych pod nazwą funkcji testowych de Jonga. Funkcje te będą przekształcane do postaci funkcji przystosowania, które muszą być nieujemne i maksymalizowane.

Pierwsza funkcja de Jonga to F_1 – funkcja sferyczna trzech zmiennych, z jednym optimum. Poprzednie doświadczenia sugerują, że nie powinna ona sprawiać zbyt dużych kłopotów algorytmowi ewolucyjnemu. Ponieważ należy szukać minimum F_1 , przekształcimy ją do funkcji przystosowania: $Q = 80 - F_1$. Przekształcona funkcja przyjmuje maksimum $Q_{\max} = 80$ dla współrzędnych $x_1 = x_2 = x_3 = 0$ ($\min F_1 = F_1(0,0,0) = 0$).

Ustalono jednorodną macierz A typu 3 i następujące wartości parametrów algorytmu: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0$, $t_{\text{kr}} = 0$, $rcd = 0$, $M_{\max} = 50$. Początkową populację umieszczono w punkcie o współrzędnych $[0,1, 0,1, 0,1]$. Chodzi więc o zdolność „dostrojenia” rozwiązania. Populacja, choć powoli, poprawiała istniejące rozwiązanie. Po tysiącu pokoleń włączono opcję „zachowaj najlepszego”. Postęp w dostrajaniu rozwiązania był następujący:

- pokolenie 1962. – $Q = 79,999012$, rozwiązanie $x = [0,01824, 0,018561, 0,01767]$,
- pokolenie 2305. – $Q = 79,999905$, rozwiązanie $x = [0,00356, 0,00822, 0,00372]$,
- pokolenie 3133. – $Q = 79,999999$, rozwiązanie $x = [0,000072, 0,00041, 0,00074]$.

Zwiększono wartość maksymalnej zmiany mutacyjnej genu, $M_{\max} = 100$, jako warunek zakończenia ewolucji ustalono $Q \geq 79,999999$. Populacja znajdowała punkt o współrzędnych, które dopiero na czwartej pozycji po przecinku miały cyfrę różną od zera, w czasie 500–600 pokoleń, natomiast w granicach 1300–1500 pokoleń populacja znajdowała rozwiązanie spełniające warunek zatrzymania ($x = [0,0001338, 0,0009635, 0,00004556]$).

Niejednorodna macierz A daje tu dużo gorsze wyniki, populacja jest rozrzucona niemal po całym obszarze przestrzeni fenotypowej, ale nie ma zdolności dostrojenia do akceptowalnego rozwiązania, niezależnie od położenia początkowej populacji.

Przedstawione wyniki można uznać za zadowalające pamiętając, że zadanie „dostrajania” optimum jest bardzo trudne dla algorytmów genetycznych. GA zwykle stosuje się do zadań, gdzie wystarcza przybliżone rozwiązanie.

Druga funkcja, F_2 , mimo iż jest funkcją tylko dwóch zmiennych, jest trudna dla GA, ponieważ jest długim, wygiętym wąwozem. Przekształcimy ją w funkcję przystosowania $Q = 3900 - F_2$. Maksimum Q jest w punkcie $x = [1, 1]$ i wynosi 3900.

Rozpoczynamy ewolucję z punktu $[2, 2]$, z macierzą A jednorodną typu 3 (ustalone elementy zerowe i ujemne, reszta automatycznie generowana) i parametrami: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0$, $t_{\text{kr}} = 0$, $rcd = 0$, $M_{\max} = 100$. Nie zachowujemy najlepszego osobnika, jak wynika z kształtu funkcji – nie chodzi tu o zwykłe wspinanie się po zboczu. W ciągu około 400 pokoleń populacja znajduje rozwiązanie, w którym $Q(1,00294, 1,00673) = 3899,9999$. Jest to dość dobra dokładność. Przy zmniejszonym zakresie zmian mutacji – $M_{\max} = 50$ otrzymujemy podobne rozwiązanie w czasie 1309 pokoleń.

Ponieważ tutaj nie możemy startować z lokalnego optimum, a kształt funkcji nie jest symetryczny, próbujemy rozpocząć ewolucję z różnych miejsc. Początkową populację umieszczono w punkcie o współrzędnych $[-2, -2]$. Przy $M_{\max} = 100$ i pozostałych parametrach jak wyżej, populacja na długie pokolenia utknęła w siodle, na

obszarze wokół punktu $x = [0,15, 0,3]$ i do 15 tysięcy pokoleń nie opuściła tego obszaru. Powtórzono eksperyment, uwzględniając geny nadmiarowe: $P_{trs} = P_{trz} = 0,01$. Populacja i tym razem zatrzymała się w tym samym obszarze. Spróbowano niejednorodnej macierzy. Jeśli punkt początkowy ewolucji ma ujemne współrzędne, to automatycznie generowana macierz A , oprócz ustalonych wcześniej zer, zawiera tylko elementy ujemne (automatycznie generowane są elementy jednakowego znaku, w tym przypadku dodatnie wartości a_{ij} nie zapewniłyby początkowego położenia populacji). Zatem A staje się macierzą jednorodną ujemną – nie zawiera elementów dodatnich. Ustalono w każdej kolumnie wartość jednego, losowego elementu na dużą wartość dodatnią (około dziesięciokrotnie większą od wartości bezwzględnych pozostałych elementów). Uruchomiono ewolucję tylko przy $P_{rkm} = 0,25$, $P_{mut} = 0,1$. W pokoleniu 136, najlepszy osobnik to $Q(1,0407, 1,0788) = 3899,9902$. Tym razem ewolucja jest efektywna nawet przy $M_{max} = 50$: w pokoleniu 188., $Q(0,99705, 0,99306) = 3899,9998$. Włączenie opcji „zachowaj najlepszego” nie wpływa zauważalnie na ewolucję, podobne rozwiązanie uzyskuje się mniej więcej w tym samym czasie.

Za punkt początkowy ewolucji przyjęto $[-2, 0]$, macierz A tworzona jak poprzednio (niejednorodna, typ 3). Włączono opcję „zachowaj najlepszego”, $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $M_{max} = 5$. W pokoleniu 278. najlepszy osobnik to $Q(1,0112, 1,0252) = 3899,9992$. Populacja jest bardzo rozproszona, trudno znaleźć rozwiązanie bardziej dokładne.

Przy niejednorodnej macierzy A typu 3, populacja dość dobrze radzi sobie z szukaniem optimum.

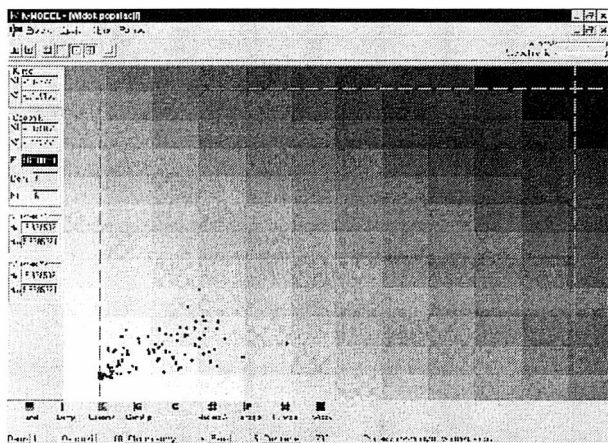
Optimum pięciowymiarowej funkcji F_3 zależy od zakresu zmienności jej argumentów. Jest to funkcja trudna dla algorytmów ewolucyjnych, ponieważ na pewnych podprzestrzeniach przyjmuje tę samą wartość. Jeśli populacja jest na tyle skupiona, iż mieści się w takim obszarze, to nie ma żadnego ciśnienia selekcyjnego. Przedstawione dalej eksperymenty mają pokazać, jak radzi sobie K -model z tym zadaniem. Funkcja przystosowania jest zdefiniowana jako $Q = 26 - F_3$. Dla rozpatrywanego przedziału zmiennych $x_i \in [-5,12, 5,12]$, $i = 1, 2, \dots, 5$, minimum funkcji jest w obszarze $x_i \leq -5,0$ i wynosi $F_3 = -30$. Ten sam obszar to maksimum funkcji przystosowania Q , równe 56.

Raz znalezionego lepszego „schodka” najlepiej byłoby nie gubić, wobec tego zdecydowano się włączyć opcję „zachowaj najlepszego”. Przyjęto jednorodną macierz A typu 3, założono: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,05$, $t_{kr} = 50$, $rcd = 0$, $M_{max} = 600$ (w kryzysie: $P_{rkm} = 0,5$, $P_{mut} = 0,5$, $P_{trs} = P_{trz} = 0,1$). Populacja początkowa jest umieszczona w punkcie $x = [0,01, 0,01, 0,01, 0,01, 0,01]$. Po 10 000 pokoleń najlepszy osobnik wciąż nie był na właściwym „schodku”: $Q(-5,11, -5,05, -3,0575, -5,02) = 54$. Dalsza ewolucja wyraźnie nie radziła sobie z poprawą tego rozwiązania.

Przyjrano się macierzy A . Ustalono w każdej kolumnie ujemny element o relatywnie dużej wartości bezwzględnej (w stosunku do pozostałych). Ponowiono symulację, z następującymi parametrami: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0$, $t_{kr} = 50$, $M_{max} = 600$ (w kryzysie: $P_{rkm} = 0,5$, $P_{mut} = 0,5$, $P_{trs} = P_{trz} = 0,1$). Uzyskano dużą poprawę

wę efektywności ewolucji, optimum było znajdowane w pokoleniach od 401 do 790 (rys. 5.17). Wartość Q wynosiła 56, a przykładowe wartości argumentów funkcji to: $x = [-5,08, -5,11, -5,03, -5,05, -5,1]$.

Zmniejszenie makromutacji przez wyłączenie kryzysu spowodowało wydłużenie się czasu ewolucji, ale nadal optimum jest znajdowane, tyle, że potrzeba na to około 1500 pokoleń, natomiast gdy zmniejszono jeszcze $M_{\max}$ do 500, czas ten wydłużył się do ponad 3500 pokoleń. Nadal jednak optimum globalne było znajdowane.



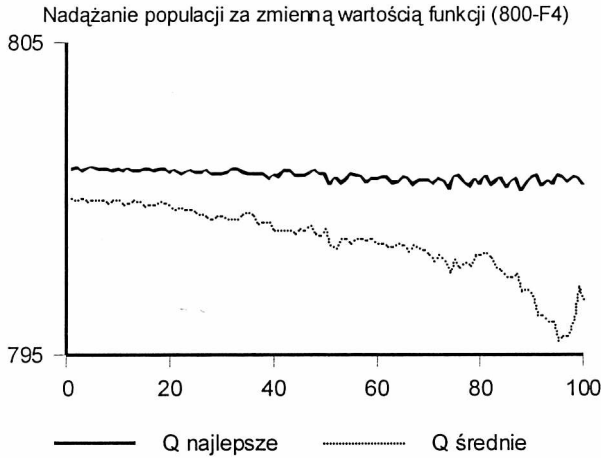
Rys. 5.17. Populacja znalazła rozwiązanie dla funkcji testowej F_3 de Jonga (pokolenie 790.)

Problem z dalszym poprawianiem ewolucji polegał na tym, że w modelu dopuszczane są tylko dodatnie wartości genów. Jeśli ustalimy niskie początkowe wartości genów, a macierz A będzie zawierać głównie dodatnie elementy, to może okazać się, że optimum będzie nieosiągalne. Należałoby rozpoczynać ewolucję z genami ustalonymi w połowie ich przedziału zmienności (w K -modelu gen może przyjmować całkowitą wartość od zera do 32 000). Jeśli to nie pomoże, to pozostają duże możliwości z doбором elementów macierzy A . Problem w tym, że zazwyczaj nie wiemy nic o możliwym kształcie funkcji przystosowania. Pewne przesłanki co do przyczyn zatrzymania ewolucji można uzyskać z analizy dotychczasowych przebiegów. Zidentyfikowanym przyczynom można próbować przeciwdziałać przez zmianę parametrów lub nawet zmianę macierzy A . W skrajnie trudnym przypadku można uruchomić ewolucję demów, każdy z inną macierzą.

Funkcja F_4 to, co prawda, w niewielkim stopniu, ale zmienne środowisko. Na stałą funkcję trzydziestu zmiennych nałożony jest losowy szum. Optimum też zmienia swoje położenie oraz wysokość. W tej sytuacji populacja powinna nadążać za zmianami funkcji. Funkcję przystosowania zdefiniowano jako: $Q = 1000 - F_4$. Funkcja F_4 przyjmuje wartość minimalną w punkcie $[0, \dots, 0]$ i tu jest maksimum przystosowania Q (bliskie 1000).

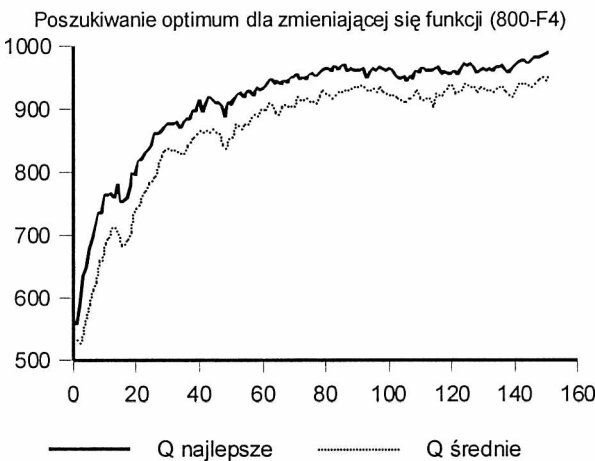
Eksperymenty przeprowadzano przy włączonej opcji „zachowaj najlepszego”, choć najlepszy osobnik w danym pokoleniu niekoniecznie był najlepszym w następnym pokoleniu. Nawet jeśli pozostawał nim, to i tak jego wartość przystosowawcza ulegała zmianie. Macierz A jest jednorodną macierzą typu 3. Populację początkową umieszczono bardzo blisko optimum (dokładne położenie optimum nie jest znane, ze względu na element losowy funkcji przystosowania), w punkcie $[0,1, \dots, 0,1]$. Parametry symulacji to: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0$, $t_{\text{kr}} = 50$, $rcd = 0$, $M_{\max} = 100$

(w kryzysie: $P_{rkm} = 0,5$, $P_{mut} = 0,5$, $P_{trs} = P_{trz} = 0,1$). Najlepsze przystosowanie w populacji, mimo iż zachowywano najlepszego osobnika, wahało się nieznacznie (rys. 5.18). Przyczyny tego należy upatrywać w losowych szumach nałożonych na funkcję F_4 . Gdy populację umieszczano w punkcie trochę oddalonym od optimum, $[1, \dots, 1]$



Rys. 5.18. Populacja zmienia najlepsze rozwiązanie wraz ze zmianą funkcji celu

my na ile dokładne mamy aktualne rozwiązanie, rezygnujemy z dalszych eksperymentów. Uzyskane rezultaty są zadowalające, populacja dość szybko dochodzi do optymalnego rejonu, zmienia najlepsze rozwiązanie, podążając za zmianami funkcji.



Rys. 5.19. Populacja zmienia najlepsze rozwiązanie wraz ze zmianą funkcji celu

(zakres parametrów to $[-1,28, 1,28]$), populacja zmierzała w kierunku optimum dość systematycznie (rys. 5.19). Uzyskiwane wyniki przy podanych wyżej parametrach to wartość $Q = 1000,0485$, a wartości argumentów wahały się od 0,02 do 0,5. Takie rozwiązanie jest znajdowane w ciągu kilkuset pokoleń, ale przez następnych kilka tysięcy pokoleń nie jest dostrajane. Można zmienić parametry na takie, które bardziej pozwolą skupić się populacji, czyli mniejsze P_{rkm} i P_{mut} między kryzysami, ewentualnie rzadziej stosować kryzys. Ponieważ i tak nie wie-

Kolejna funkcja to F_5 – nazywana czasami „lisimi jamami”. Jest ona uważana za trudną dla GA funkcję dwóch zmiennych. Przekształcamy ją na funkcję przystosowania: $Q = 501 - F_5$. Wartość minimalną, bliską jedności, przyjmuje dla argumentów równych $[-32, -32]$.

K-model radzi sobie z tą funkcją całkiem nieźle. Przy parametrach $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0,02$, $t_{kr} = 20$, $rcd = 0$, $M_{max} = 600$ (początkowe wartości genów równe są 15 000), w kryzysie $P_{rkm} = 0,5$, $P_{mut} = 0,5$, $P_{trs} = P_{trz} = 0,1$, populacja w cią-

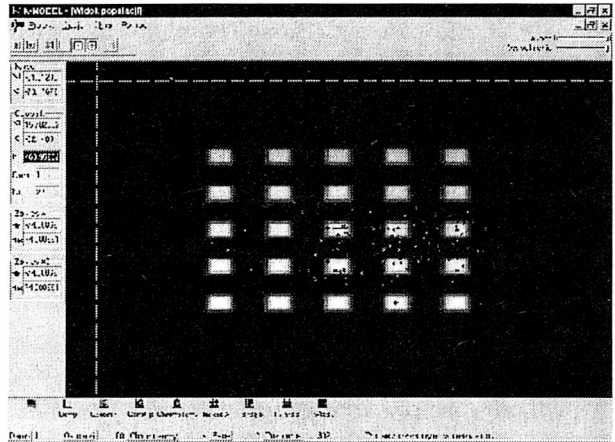
gu 1873 pokoleń znalazła rozwiązanie bliskie optymalnemu: $Q = 491,1954$, $x = [-32,294, -31,8034]$ (macierz A dość jednorodna, typu 3). Populacja nie potrafiła zatrzymać się na najlepszym szczycie, wędrowała po sąsiadujących szczytach (rys. 5.20). Na początku ewolucji populacja umieszczona była w punkcie $[40, 40]$.

Wyłączono kryzys, zostawiając pozostałe parametry bez zmian. Optimum równe 491,19, w punkcie $x = [-32,1929, -32,348]$ znalezione zostało w pokoleniu 673. Populacja również i tym razem nie pozostaje na tym szczycie.

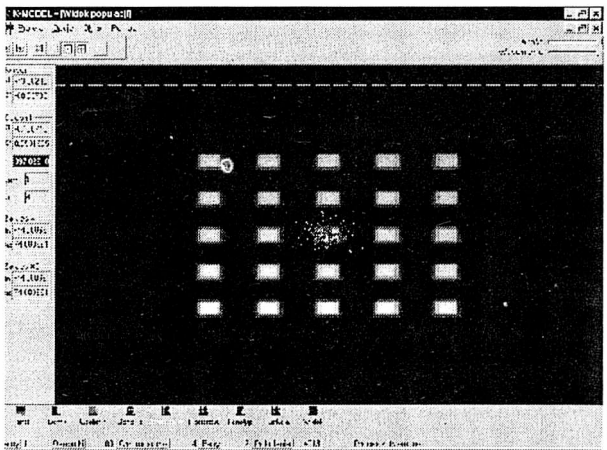
Uruchomiono symulację dla $M_{\max} = 500$, bez genów nadmiarowych i kryzysu. I tym razem populacja znalazła całkiem dobre rozwiązanie w przyzwoitym czasie 9646 pokoleń.

Ponieważ niemożliwe jest inne przejście populacji z jednego szczytu na drugi, jak tylko „przeskoczenie” doliny przystosowawczej, raz znaleziony lepszy szczyt mógłby nie być już gubiony. Wydaje się więc, że włączenie opcji „zachowaj najlepszego” powinno przyspieszyć ewolucję. Przy tych samych parametrach co poprzednio, populacja ewoluje nieco szybciej, na znalezienie wysokiego szczytu potrzebuje 6376 pokoleń: $Q = 490,87$, $x = [-31,46, -31,32]$. Populacja skupia się wokół kolejnych szczytów, przez które przechodzi (rys. 5.21), ale też nie pozostaje na długo w globalnym optimum, raczej wędruje, spędzając trochę czasu w sąsiadujących z optimum szczytach.

Jak widać z zaprezentowanych eksperymentów, K-model radzi sobie z trudnymi funkcjami testowymi de Jonga. Jak zawsze, problemem może być dobór odpowiednich wartości parametrów algorytmu. Bardzo dokładne dostrojenie otrzymanego rozwiązania jest trudniejszym zadaniem, ale ten problem pojawia



Rys. 5.20. Populacja wędruje po przestrzeni pełnej wysokich, lokalnych szczytów ($501-F_5$), mutacje, rekombinacje, nadmiarowość i kryzys



Rys. 5.21. Populacja wędruje po przestrzeni pełnej wysokich, lokalnych szczytów ($501-F_5$), tylko mutacje i rekombinacje, przy $M_{\max} = 500$

się zawsze, gdy chcemy stosować algorytmy ewolucyjne. Po porównaniu z klasycznym algorytmem genetycznym można stwierdzić, że w wielu przykładach, np. przy funkcji F_3 , zaproponowany w tej pracy model radzi sobie lepiej.

5.4. SZUKANIE OPTIMÓW POZOSTAŁYCH FUNKCJI TESTOWYCH

Funkcja F_6 jest wielomodalną funkcją, która przyjmuje minimum równe zeru w punkcie o współrzędnych $[0, 0, 0, 0, 0]$. Jest to trudna funkcja, wystarczy spojrzeć na jej wartości w kilku punktach (przykładowe wartości F_6 w punktach o jednakowych współrzędnych, $i = 1, 2, \dots, 5$, x_i są tego samego znaku – tab. 5.1). Funkcja przystosowania to $Q = 202 - F_6$.

Tabela 5.1. Przykładowe wartości funkcji testowej F_6

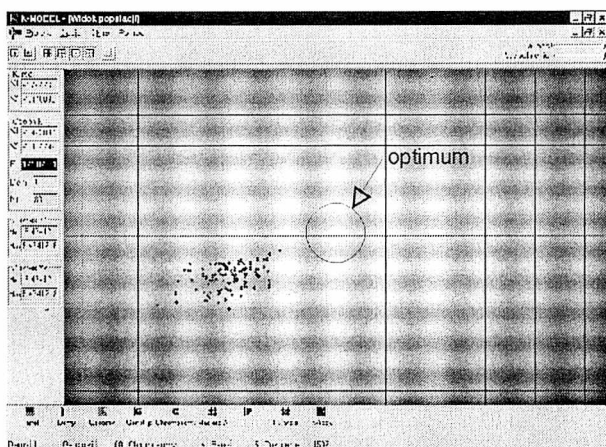
$x_i = 0$	$x_i = \pm 0,1$	$x_i = \pm 0,5$	$x_i = \pm 1$	$x_i = \pm 1,5$	$x_i = \pm 2$	$x_i = \pm 2,5$	$x_i = \pm 3$	$x_i = \pm 3,5$	$x_i = \pm 4$	$x_i = \pm 4,5$	$x_i = \pm 5$
0	9,59	101,25	5	111,25	20	131,25	45	161,25	80	201,25	125

Rozpoczynając ewolucję z odległego punktu $[-5, -5, -5, -5, -5]$, populacja dość szybko zbliża się do optimum. Trudniej jest znaleźć samo optimum. Jest to bardzo mały obszar, otoczony dolinami przystosowawczymi. Prawdopodobieństwo trafienia w ten obszar jest małe.

Założono jednorodną macierz A typu 3. Wartości parametrów ustalono następująco: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0$, $t_{kr} = 20$, $rcd = 0$, $M_{max} = 500$ (początkowe

wartości genów równe są 1500, w kryzysie $P_{rkm} = 0,5$, $P_{mut} = 0,5$, $P_{trs} = P_{trz} = 0$). Populację po 1500 pokoleniach pokazano na rysunku 5.22. Widać, że populacja zmierza w dobrym kierunku, jednakże długo nie może znaleźć samego optimum.

Zmieniono macierz A : zawiera ona teraz elementy ujemne i w każdej kolumnie jeden element dodatni, przynajmniej o rząd większy co do modułu od pozostałych. Uruchomiono symulację z parametrami: $P_{rkm} = 0,25$, $P_{mut} = 0,1$, $P_{trs} = P_{trz} = 0$, $t_{kr} = 20$



Rys. 5.22. Funkcja testowa F_6 , populacja w okolicy optimum (pokolenie 1502.)

(w kryzysie $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 0,5$, $P_{\text{trs}} = P_{\text{trz}} = 0,1$), $M_{\text{max}} = 50$, punkt początkowy populacji – wszystkie współrzędne są równe -5 . Pozwolono na ewolucję przez 300 pokoleń, po tym czasie populacja przybliżyła się stosunkowo szybko do optimum (najlepszy osobnik był w okolicach punktu o współrzędnych równych -1). Teraz zmieniono M_{max} na wartość równą 25. W 2453. pokoleniu znalezione zostało najlepsze rozwiązanie spośród wszystkich dotychczasowych: $Q = 200,708$, dla argumentów $x = [-0,0397, -0,008465, -0,0292, 0,0611, 0,0179]$.

Zatem i to trudne zadanie dało się rozwiązać przy użyciu K-modelu i to z całkiem dobrą dokładnością.

Funkcja F_7 to wielomodalna, dwuwymiarowa funkcja. Informację o jej kształcie daje tabela 5.2 i np. rysunek 5.23. Zakres zmienności argumentów to $[-20, 20]$. Populację początkową umieszczano w punkcie $[-15, -15]$, tak aby populacja musiała napotkać przeszkodę w postaci lokalnego szczytu. Funkcja jest symetryczna, można szukać jej maksimum, ale i tak należy ją przekształcić, bo przyjmuje wartości ujemne, które nie mogą być przystosowaniem. Jako funkcję przystosowania przyjęto $Q = 40 + F_7$.

Tabela 5.2. Przykładowe wartości funkcji testowej F_7

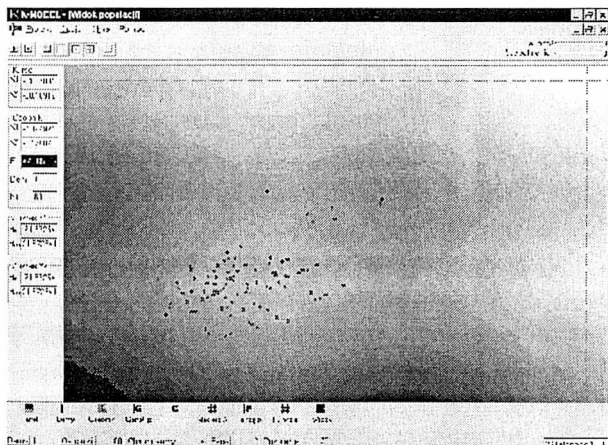
x_1	21,6	20	15	10	5	2,5	0	-2,5	-10	-20	0	20	5
x_2	21,6	20	15	10	5	2,5	0	-2,5	-10	-20	-20	0	0
F_7	43	38,8	20	0,4	-7,8	-4,9	0	-4,9	-0,4	-38,8	-19	19	-3,9

Ponieważ wartości fenotypów populacji początkowej są ujemne, a wartości genów dodatnie, program wygeneruje macierz A o ujemnych elementach. Wstawiono do tej macierzy elementy dodatnie, równe 0,1 po dwa takie elementy w każdej kolumnie, pozostałe elementy wynosiły około $-0,019...$ w kolumnie pierwszej i $-0,011...$ w kolumnie drugiej.

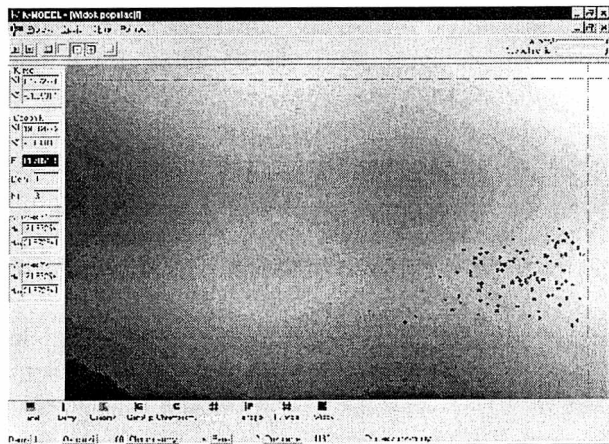
Parametry symulacji to: $P_{\text{rkm}} = 0,25$, $P_{\text{mut}} = 0,1$, $P_{\text{trs}} = P_{\text{trz}} = 0$, $t_{\text{kr}} = 20$ (w kryzysie $P_{\text{rkm}} = 0,5$, $P_{\text{mut}} = 0,5$, $P_{\text{trs}} = P_{\text{trz}} = 0,1$), $M_{\text{max}} = 20$. Typowy przebieg ewolucji ilustrują rysunki. Uzyskiwane rezultaty wydają się być w pełni zadowalające:

- pokolenie 1325., $Q = 78,3$, $x = [19,53, 19,98]$,

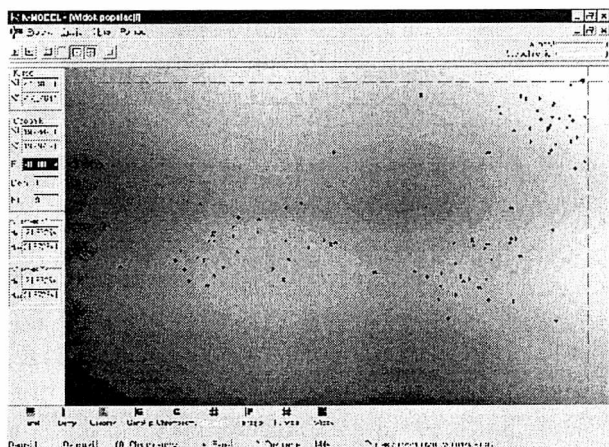
- pokolenie 1444., $Q = 78,3$, $x = [19,64, 19,998]$, po dalszej ewolucji, w pokoleniu 1464., $Q = 78,558$, $x = [19,89, 19,9]$.



Rys. 5.23. Funkcja F_7 , pokolenie 70. – populacja próbuje wydostać się z lokalnego optimum



Rys. 5.24. Funkcja F_7 , pokolenie 1180. – populacja przeszła do następnego optimum

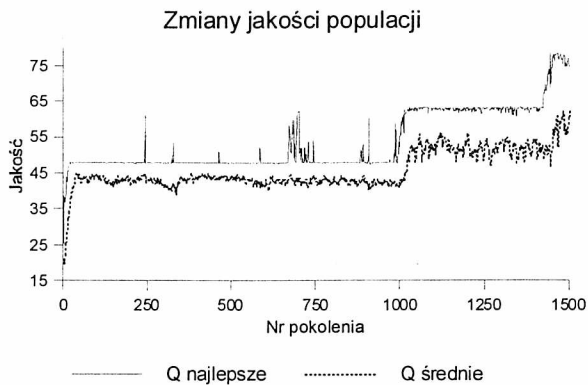


Rys. 5.25. Funkcja F_7 , pokolenie 1444. – populacja znalazła optimum

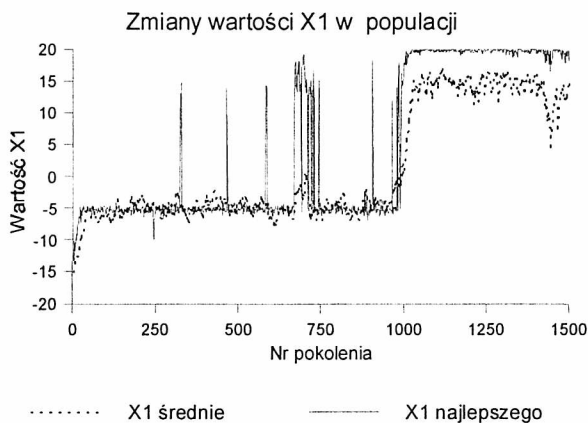
porównywalny do poprzednich wyników (np. 1104. czy 1233. pokolenia). Trudno wnioskować o pozytywnym lub negatywnym wpływie włączenia tej opcji na szybkość ewolucji przy tej funkcji. Zysk z zajmowania nowego optimum po pierwszym jego „odkryciu” jest niwelowany trudniejszym opuszczeniem dotychczas zajmowanego szczytu.

Z wyników wszystkich eksperymentów wyłania się jedna konkluzja: jeśli znamy kształt funkcji, możemy – mając pewne doświadczenie w tej dziedzinie – stosunkowo łatwo tak dobrać parametry modelu, zwłaszcza macierz plejotropowości A i parametry odpowiedzialne za makromutacje, aby w rozsądnym czasie uzyskać akceptowalne rozwiązanie. Problem w tym, że w rzeczywistych zagadnieniach raczej nic lub niewiele możemy powiedzieć o kształcie funkcji. Ale z obserwacji przebiegu ewolucji

Zarówno dokładność otrzymanego wyniku, jak i czas ewolucji można uznać za wystarczające. Populacja omija przeszkody głównie dzięki kryzysowi, okres pomiędzy kryzysem wykorzystuje natomiast na umocnienie swojej pozycji w lepszym miejscu przestrzeni fenotypowej. Jak widać na rysunkach 5.26 i 5.27, populacja kilka razy „odkrywała” kolejne szczyty przystosowawcze i gubiła je – widoczne to było podczas obserwacji ewolucji. Zarówno zmiany przystosowania najlepszego osobnika, jak i wartości jego współrzędnych pokazują wyraźne skoki, które świadczą o tym, że w tych pokoleniach osobnik był na wyższym szczycie. Sprawdzano, czy włączenie opcji „zachowaj najlepszego” daje pozytywny efekt. Jak pamiętamy, „nieśmiertelność” najlepszego osobnika z jednej strony pozwala nie tracić raz znalezionych lepszych rozwiązań, z drugiej zaś powoduje zatrzymywanie populacji na lokalnym szczycie. W przeprowadzonych symulacjach czas ewolucji był



Rys. 5.26. Zmiany jakości najlepszego osobnika i średniej jakości populacji, funkcja F_7



Rys. 5.27. Zmiany współrzędnej x_1 najlepszego osobnika i jej średniej wartości w populacji, funkcja F_7 (podobnie wyglądają zmiany x_2)

przy typowych parametrach można wnioskować o możliwych przyczynach braku jej efektywności i przeciwdziałać temu przez zmianę operatorów. Od tego już mały krok do następnej konkluzji: spróbujmy nauczyć tego nasz program – przekażmy mu wiedzę zdobytą w naszych doświadczeniach, nauczmy analizować efektywność ewolucji i identyfikować możliwe przyczyny braku postępu. Dostarczmy mu wiedzę, jak ma się zachować, co i jak modyfikować, aby wyraźnie poprawić swoje działanie. Pewne przemyślenia na temat, jak to zrobić, znajdują się w końcowym rozdziale tej pracy.

Czy to oznacza, że jeszcze nic nie można powiedzieć o możliwościach aplikacyjnych K-modelu? Wydaje się, że nie, przykład takiej aplikacji znajduje się w następnym rozdziale.

6. PRZYKŁAD APLIKACJI K-MODELU – MODELOWANIE EWOLUCYJNE W EKONOMII

Wszystkie procesy ewolucyjne, poczynając od ewolucji biologicznej, przez rozwój kulturowy, społeczny i technologiczny oraz rozwój naszej wiedzy indywidualnej, mają pewne wspólne właściwości. Proces ewolucyjny jest ze swej natury procesem dynamicznym i historycznym, w którym globalne charakterystyki rozwoju są rezultatem aktywności pojedynczych elementów (osobników biologicznych, firm, badaczy itp.). Podstawowymi cechami procesów ewolucyjnych jest różnorodność i heterogeniczność zachowań. Dobór (selekcja) oraz poszukiwanie nowych rozwiązań są dwoma podstawowymi mechanizmami rozwoju. Powinniśmy jednak pamiętać, że każdy pojedynczy proces ewolucyjny ma swoją specyfikę. W przypadku procesów gospodarczych takimi wyróżniającymi cechami, które wymuszają specyficzne widzenie rozwoju gospodarczego jako procesu ewolucyjnego, są inwestycje, tworzenie kapitału oraz ceny. Mechanizmy poszukiwania nowych rozwiązań (innowacji) wydają się być wspólne wszystkim procesom ewolucyjnym. Dlatego też w przedstawionym dalej ewolucyjnym modelu przemysłu (**EMP**) ten fragment został wyraźnie zapożyczony z modelu ewolucji populacji biologicznych (*K*-model). Widać to wyraźnie w używanej nomenklaturze (mutacje, rekombinacje – pojęcia tak dobrze znane w modelach ewolucji biologicznej). **EMP** może być z powodzeniem stosowany do opracowania gry menedżerskiej o dużych walorach dydaktycznych i badawczych.

6.1. EWOLUCYJNY MODEL PRZEMYSŁU (**EMP**)

Ewolucyjny model przemysłu (**EMP**) opisuje zachowanie się pewnej liczby firm produkujących funkcjonalnie podobne produkty (tzn. produkty zaspokajające określoną potrzebę), funkcjonujących na określonym rynku. Decyzje każdej firmy, odnoszące się do wielkości inwestycji, ceny produktów, wielkości pożądanego zysku itp., są dokonywane na podstawie oceny przyszłego zachowania się konkurentów i oczekiwanej reakcji rynku (konsumentów) na wprowadzenie na rynek określonej liczby produktów danej firmy. Wiedza firm dotycząca zachowania się konsumentów oraz wiedza o przyszłych zachowaniach konkurentów jest ograniczona i cechuje się dużym pozio-

mem niepewności. Decyzje są podejmowane przez każdą firmę niezależnie w tym samym czasie – na początku umownej jednostki symulacji, np. roku lub kwartału.

Po podjęciu decyzji firmy produkują taką ilość produktów, jaką spodziewają się sprzedać na rynku. Rzeczywista liczba produktów danej firmy kupionych przez klientów zależy od względnej ceny ich produktów, względnej oceny ich jakości (tzn. stopnia zaspokojenia danej potrzeby przez konkretny produkt) oraz nasycenia rynku. Na rynku mogą wystąpić stany niezaspokojenia lub nadprodukcji zarówno w skali globalnej całego rynku, jak i w skali „lokalnej” na poziomie specyficznych produktów. W długim okresie produkty oceniane jako lepsze (tzn. produkty o niższej cenie i lepszej jakości) są kupowane chętniej przez konsumentów i ich udział na rynku rośnie. Dokładny opis modelu znaleźć można w publikacjach Kwaśnickiego (1994/1996) oraz Kwaśnickiego i Kwaśnickiej (1992).

Symulacja rozwoju przemysłu w proponowanym **EMP** odbywa się w dyskretnym czasie, w czterech etapach:

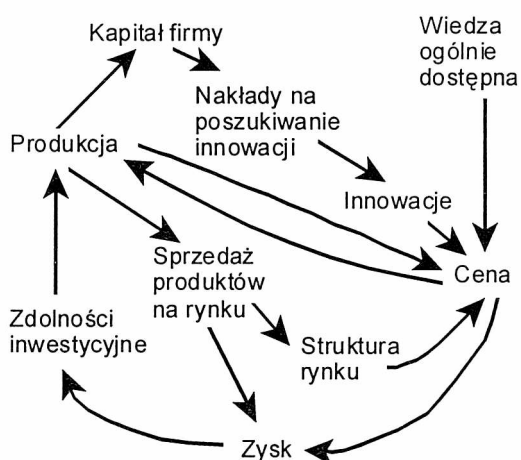
1. Poszukiwanie *wynalazków*, których wprowadzenie do produkcji (*innowacje*) może zastąpić „stare”, aktualnie stosowane produkty (metody produkcji).

2. Oszacowanie i porównanie wielkości inwestycji, produkcji, dochodu netto, zysku oraz innych charakterystyk rozwoju firmy, wymaganych do kontynuowania produkcji „starymi” metodami oraz po zastosowaniu nowo odkrytych. W rezultacie podjęcie decyzji o: (1) kontynuowaniu produkcji starymi metodami lub dokonanie modernizacji produkcji oraz (2) ewentualnym otwarciu nowego zakładu, gdzie istotnie nowe sposoby produkcji będą zastosowane.

3. Wejście nowych firm na rynek.

4. Rynekowa ocena całej puli produktów oferowanych przez wszystkie firmy do sprzedaży. Obliczenie charakterystyk rozwoju firm (produkcji sprzedanej, udziału firm na rynku) oraz globalnych charakterystyk rozwoju gałęzi przemysłu.

Ogólną strukturę ewolucyjnego modelu rozwoju przemysłu przedstawiono na rysunku 6.1. Wyznaczana przez producenta cena produktów zależy od dostępnych innowacji wprowadzanych przez daną firmę, aktualnej struktury rynku i od wielkości produkcji, jaką spodziewa się ona sprzedać na rynku. Dwie strzałki pomiędzy *ceną* i *produkcją* wskazują, że cena jest ustalana w sposób interaktywny, tak by zaspokoić kryteria funkcjonowania firmy, tzn. utrzymać zysk firmy na relatywnie wysokim poziomie i jednocześnie zapewnić jej długookresowy rozwój. W modelu rozróżniamy *innowacje*



Rys. 6.1. Ogólna struktura modelu rozwoju przemysłu (za: Kwaśnicki 1994/1996)

i *wynalazki* (*wynalazki* są to nowości, które potencjalnie mogą być wprowadzone do praktyki i stać się *innowacjami*). Istnieją dwa podstawowe sposoby poszukiwania nowych rozwiązań (*wynalazków*), mianowicie autonomiczne badania prowadzone przez daną firmę oraz naśladownictwo konkurentów. Modernizacja produkcji przez wprowadzenie innowacji lub też zainicjowanie nowej produkcji przez zastosowanie jakiegoś radykalnie nowego wynalazku wymaga określonego kapitału inwestycyjnego. Dlatego firmy, myśląc o wprowadzeniu innowacji, uwzględniają w swoich decyzjach także uwarunkowania ekonomiczne, związane np. ograniczonymi zasobami inwestycyjnymi.

Każda z firm dąży do tego, by polepszyć swoją sytuację w porównaniu z jej konkurentami. Podstawowym sposobem na to jest wprowadzanie innowacji i odpowiednie ustalanie ceny swoich produktów. Aktywność badawcza firm „obejmuje modyfikowanie i rekombinacje bieżących rozwiązań technologicznych i pomysłów organizacyjnych w połączeniu z dostępnymi zdolnościami pracowników w określonym kontekście gospodarczym” (Winter 1984). Decyzje rynkowe zależą natomiast od charakterystyk technicznych produktów oferowanych na sprzedaż i ich ceny. Możemy mówić o istnieniu dwóch przestrzeni: przestrzeni rutyn i przestrzeni charakterystyk produktów¹.

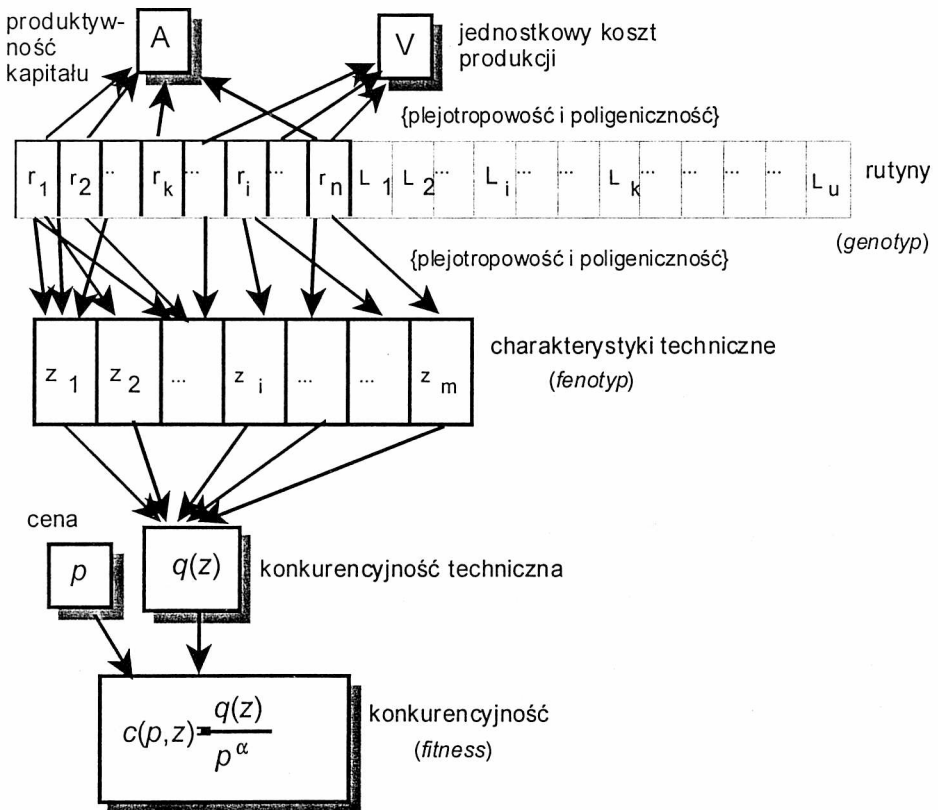
W modelu **EMP** firmy mogą jednocześnie wytwarzać produkty o różnych charakterystykach technicznych oraz różnych cenach; mówimy, że firma może być przedsiębiorstwem wielozakładowym. Różne zakłady wytwarzają produkty, stosując inne zbiory rutyn² powstałe w trakcie działalności badawczej. Nowe technologiczne lub organizacyjne rozwiązania (tzn. nowe zbiory rutyn) mogą okazać się znacznie lepsze od aktualnie stosowanych, koszty pełnej modernizacji bieżącej produkcji mogą jednak znacznie przekraczać zdolności inwestycyjne firmy. W takiej sytuacji firma kontynuuje produkcję, stosując stary zbiór rutyn (starych metod produkcji), i jednocześnie rozpoczyna produkcję w nowym zakładzie na podstawie nowego zestawu rutyn, ale na znacznie mniejszą skalę. W trakcie dalszego rozwoju firmy dotychczasowa produkcja jest ograniczana, a „nowa” rozwijana tak, że po pewnym czasie produkowane są jedynie nowocześniejsze produkty.

¹ Przestrzeń rutyn i przestrzeń charakterystyk produktów mają w modelu **EMP** rolę podobną do tej, jaką w ewolucji biologicznej odgrywają przestrzenie genotypów i fenotypów. Istnienie tych dwóch rodzajów przestrzeni jest cechą ogólną wszystkich procesów ewolucyjnych. Wymiar przestrzeni rutyn (genotypów) jest znacznie większy niż wymiar przestrzeni charakterystyk (fenotypów).

² Nelson i Winter (1982, s. 14) definiują rutynę jako „regularne i przewidywalne zachowanie się firm, ... techniczne rutyny odnoszące się do sposobów produkcji ... procedury naboru pracowników i ich zwalniania, składania zamówień i magazynowania, zwiększania produkcji na którą istnieje popyt, polityki dotyczące inwestowania, prowadzenie badań i rozwoju, prowadzenia kampanii reklamowych, strategii prowadzenia interesów dotyczących dywersyfikacji produkcji i inwestycji zagranicznych”. W dużym stopniu działalność badawcza jest też zrutyinizowana. „Rutyny kierują wyborem jak również opisują metody, są odzwierciedleniem praktyki zarządzania i socjologii organizacji oraz odnoszą się do technologii produkcji.” (Winter, 1984).

6.1.1. REPREZENTACJA FIRM W EMP

Zakładamy, że każda firma jest charakteryzowana przez zbiór (zbiory) rutyn. Wyóżniamy dwa rodzaje rutyn: *aktywne*, tzn. rutyny rzeczywiście stosowane przez firmę w jej codziennej praktyce – odpowiadają genom fenotypowym w K-modelu, oraz *nadmiarowe* (albo inaczej *ukryte*), które są dostępne dla firmy, do użycia ich w praktyce, ale z różnych powodów nie są stosowane – odpowiadają genom nadmiarowym. Rutyny nadmiarowe mogą być włączone do zbioru rutyn aktywnych w dowolnej chwili. Cały zbiór rutyn jest również podzielony na rozłączne podzbiory, zwane segmentami, składające się z rutyn stosowanych przez firmę w różnych sferach jej działalności. Przykładami mogą być segmenty rutyn odnoszące się do działalności produkcyjnej, zarządzania i organizacji firmy, marketingu, reklamy itp. W każdym segmencie mogą istnieć zarówno rutyny aktywne, jak i nadmiarowe. Każdy segment odpowiada jednemu chromosomowi w K-modelu. Zbiór rutyn może ewoluować. Istnieją cztery podstawowe ewolucyjne mechanizmy generowania nowych zbiorów rutyn, mianowicie: *mutacje*, *rekombinacje*, *tranzycje* i *transpozycje*.



Rys. 6.2. Reprezentacja firmy: od rutyn do jednostkowego kosztu produkcji, produktywności kapitału i konkurencyjności (za: Kwaśnicki 1994/1996)

Prawdopodobieństwo odkrycia całkowicie nowej rutyny (*mutacja*) zależy od wielkości funduszy przeznaczonych przez firmę na finansowanie autonomicznych badań prowadzonych przez zatrudniony w niej personel badawczy. Firma może przeznaczyć też część swojego funduszu badawczego na zdobycie wiedzy od konkurentów i próbować naśladować (*rekombinacje*) rozwiązania stosowane przez konkurujące firmy.

Zakładamy, że rekombinacje mogą zachodzić pomiędzy całymi segmentami. Dlatego dzięki rekombinacji firma może uzyskać odpowiednią wiedzę dotyczącą całej sfery działalności innej firmy, np. przez zakup licencji.

Pojedyncza rutyna może być przekazywana z pewnym prawdopodobieństwem przez tzw. *tranzycję*. Wskutek tranzycji rutyna jest dołączana do zbioru rutyn ukrytych. W każdej chwili, z pewnym prawdopodobieństwem, rutyna ukryta może być przeniesiona do zbioru rutyn aktywnych (*transpozycja*).

Prawdopodobieństwa tranzycji i transpozycji nie zależą od wielkości nakładów na badania i mają tę samą wartość dla wszystkich rutyn. Wartości tych prawdopodobieństw są raczej małe w swym „naturalnym” stanie, jednakże od czasu do czasu gwałtownie rosną. Obserwujemy wówczas względnie krótkotrwały okres intensywnego poszukiwania nowych kombinacji rutyn. Zjawisko to nazywamy *kryzysem*.

Kryzys zależy od zdolności badaczy pracujących w danej firmie do poszukiwania oryginalnych, radykalnych innowacji przez zastosowanie odważnych, czasami pozornie szalonych pomysłów. Zdolności te zależą przede wszystkim od osobowości personelu badawczego, często czynniki całkowicie przypadkowe odgrywają zasadniczą rolę w tym procesie.

Można powiedzieć, że regułą jest znajdowanie względnie małych ulepszeń istniejących rozwiązań dzięki mutacjom, rekombinacjom i transpozycji, których wartości prawdopodobieństwa są stosunkowo małe w długich okresach, innowacje radykalne powstają natomiast dzięki kryzysowi, w stosunkowo krótkich okresach, kiedy możliwości znajdowania ulepszeń starych rozwiązań (opartych na starym paradygmacie) wyczerpują się.

6.1.2. DECYZJE FIRM

Problem zrozumienia i odpowiedniego zamodelowania procesu podejmowania decyzji wydaje się być jednym z kluczowych w ekonomii. Zaproponowane przez Witolda Kwaśnickiego (1994/1996) podejście może być uznane za pierwsze przybliżenie formalnego ujęcia procesu podejmowania decyzji na poziomie firmy. Zakładamy, że firmy podejmują swoje decyzje na podstawie spodziewanych decyzji podejmowanych przez konkurentów i swoich oczekiwań o przyszłym rozwoju całej gałęzi przemysłu (w formie np. oczekiwanej średniej ceny i średniej konkurencyjności produktów). Znając swoje możliwości inwestycyjne, aktualną sytuację na rynku oraz możliwe do wprowadzenia innowacje, każda firma określa wielkość produkcji, jaką spodziewa się

sprzedać w bieżącym roku oraz bieżącą cenę swoich produktów, tak by maksymalizować kryterium działalności firmy.

Jak wynika z eksperymentów symulacyjnych, najskuteczniejszym kryterium pozwalającym firmom na długie funkcjonowanie i uzyskiwanie stosunkowo dużych zysków była kombinacja kryterium krótkookresowego (w postaci bieżącego dochodu firmy) i długookresowego (dążenia do uzyskania możliwie dużego udziału danej firmy na rynku).

W sposób formalny kryterium to (O_i) można zapisać następująco (za: Kwaśnicki 1994/1996):

$$O_i(t+1) = (1 - F_i) \frac{\Gamma_i(t+1)}{\Gamma(t)} + F_i \frac{Q_i^s(t+1)}{QS(t)}, \quad (6.1)$$

$$F_i = a_4 \exp \left(-a_5 \frac{Q_i^s(t+1)}{QS(t)} \right), \quad (6.2)$$

gdzie:

$\Gamma_i(t)$ – dochód firmy i w roku t ,

Q_i^s – produkcja sprzedana firmy i ,

Γ – dochód globalny,

QS – produkcja globalna,

F_i – waga kryterium długookresowego i krótkookresowego,

a_4 i a_5 – parametry kontrolujące zmiany tej wagi.

W przeprowadzanych eksperymentach zazwyczaj przyjmowano $a_4 = 1$ i $a_5 = 5$ (co oznacza, że kryterium długookresowe jest ważniejsze niż kryterium krótkookresowe oraz że firmy stosują strategię elastyczną, zmniejszając wagę kryterium długookresowego w miarę wzrostu firmy – dla małych firm waga kryterium długookresowego jest znacznie większa niż dla firm dużych).

Dochód Γ_i i zysk Π_i firmy i określają wzory:

$$\Gamma_i = Q_i^s(t)(p_i(t) - V_i v(Q_i^s(t)) - \eta), \quad (6.3)$$

$$\Pi_i = \Gamma_i - K_i(t)(\rho - \delta), \quad (6.4)$$

gdzie:

p_i – cena produktów firmy i ,

V_i – koszty jednostkowe,

$v(Q_i^s)$ – współczynnik wpływu ekonomii skali na koszty jednostkowe,

ρ – normalna stopa zwrotu kapitału,

δ – amortyzacja.

6.1.3. ZRÓŻNICOWANIE PRODUKTÓW I KONKURENCJA

Produktywność kapitału, zmienne koszty produkcji i charakterystyki produktów są funkcjami rutyn stosowanych przez każdą firmę. Każda rutyna może wpływać w różnorodny sposób na wiele charakterystyk produktów oraz na produktywność kapitału i jednostkowe koszty produkcji, podobnie każda z charakterystyk produktu oraz produktywność i koszty produkcji mogą zależeć od wielu różnych rutyn (plejotropowość i poligeniczność w biologii).

Zakładamy zatem, że istnieje pewna transformacja zbioru rutyn na zbiór charakterystyk produktów określanych przez m funkcji F_k

$$z_k = F_k(r), \quad k = 1, 2, 3, \dots, m, \quad (6.5)$$

gdzie:

z_k – wartość k -tej charakterystyki,

m – liczba charakterystyk produktów,

r – zbiór rutyn aktywnych charakteryzujących daną firmę.

Atrakcyjność produktów na rynku zależy od wartości charakterystyk produktów (*konkurencyjności technicznej* produktów) i ich ceny.

Zakładamy istnienie pewnej funkcji q umożliwiającej wyliczenie konkurencyjności technicznej w zależności od wartości charakterystyk technicznych danego produktu.

Ostateczna *konkurencyjność produktów* c (tzn. pewien hipotetyczny wskaźnik, którym kierują się konsumenci przy zakupie produktów) zależy od konkurencyjności technicznej i ceny produktów

$$c(p, z) = \frac{q(z)}{p^\alpha}, \quad z = (z_1, z_2, \dots, z_m), \quad (6.6)$$

gdzie:

z – wektor charakterystyk technicznych produktu,

p – cena produktu,

$q(z)$ – konkurencyjność techniczna,

α – elastyczność cenowa konkurencyjności.

Konkurencyjność c jest zatem analogiem wartości przystosowawczej (*fitness*) osobników biologicznych.

Przyjmujemy, że funkcja popytu zależy od wartości średniej ceny produktów oferowanych do sprzedaży na rynku. Wielkość produkcji $Q^d(t)$, którą klienci są gotowi kupić

$$Q^d(t) = \frac{M(t)}{p^e(t)}, \quad (6.7)$$

gdzie:

$p^e(t)$ – średnia cena produktów oferowanych do sprzedaży,

$M(t)$ – wartość pieniędzy, którą konsumenci gotowi są wydać na zakup danych produktów.

Zakładamy, że wartość pieniędzy wydanych na zakup produktów

$$M(t) = N \exp(\gamma t) (p^e(t))^\beta, \quad (6.8)$$

gdzie:

N – wielkość charakteryzująca początkowy rozmiar rynku,

γ – współczynnik wzrostu rozmiaru rynku w czasie,

β – elastyczność cenowa.

Elastyczność β charakteryzuje rodzaj produktów, $0 < \beta < 1$ dla produktów podstawowych (koniecznych do życia, jak np. żywność, ubrania, mieszkania), $\beta < 0$ natomiast dla produktów luksusowych (wyższego rzędu).

Faktyczna wielkość produkcji sprzedanej na rynku $QS(t)$ jest równa mniejszej z dwóch wartości:

(1) produkcji $Q^d(t)$ wynikającej z funkcji popytu, lub

(2) globalnej produkcji $Q^s(t)$ oferowanej przez wszystkie firmy do sprzedaży na rynku $QS(t) = \min\{Q^d(t), Q^s(t)\}$.

Oznacza to, że kupowane są produkty lepsze, tzn. te o większej konkurencyjności c_i . W miarę upływu czasu udział produktów lepszych kupowanych przez konsumentów rośnie.

Udział f_i produktów firmy i jest wyznaczany na podstawie ogólnego równania selekcji

$$f_i(t) = f_i(t-1) \frac{c_i(t)}{c^e(t)}, \quad (6.9)$$

gdzie:

c_i – konkurencyjność produktów firmy i ,

c^e – średnia cena produktów oferowanych na sprzedaż, $c^e(t) = \sum_i f_i(t-1)c_i(t)$.

Faktyczna wielkość produkcji firmy i sprzedanej na rynku

$$Q_i^d(t) = QS(t)f_i(t). \quad (6.10)$$

Równanie selekcji (6.9) określa niejako idealną sytuację konkurencji produktów oferowanych na sprzedaż na danym rynku, to znaczy, kiedy globalna podaż jest równa globalnemu popytowi oraz popyt na specyficzne produkty każdej firmy i jest równy wielkości produkcji oferowanej na sprzedaż przez tę firmę. Te ogólne równania selekcji powinny być dopasowane do sytuacji rzeczywistej, kiedy mogą występować niezrównoważenia zarówno popytu i podaży w skali całej gałęzi przemysłu, jak i popytu i podaży na produkty pojedynczych firm. Dostosowanie tych ogólnych równań selekcji do sytuacji niezrównoważonego popytu i podaży jest przedstawione m.in. w publikacji Kwaśnickiego (1994/1996).

Uzyskiwane wyniki symulacji sugerują duże podobieństwo zachowania się modelu do rzeczywistych procesów przemysłowych obserwowanych w gospodarce ryn-

kowej. Pozwala to sądzić, że model ten dosyć dobrze odzwierciedla rzeczywiste mechanizmy rozwoju gospodarki rynkowej.

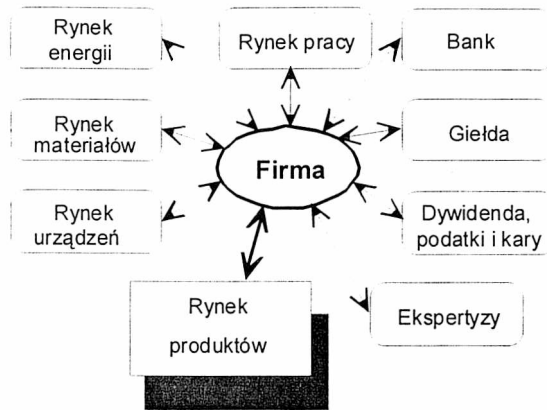
6.2. KONKURENCYJNA GRA EKONOMICZNA OPARTA NA EMP

Opracowany przez Kwaśnickiego (1994/1996) ewolucyjny model przemysłu był inspiracją do opracowania gry kierowniczej. W tej grze algorytm genetyczny jest aktywny przez cały czas, podejmując decyzje w każdej umownej jednostce czasu. Jedynym graczem jest człowiek, pozostałymi firmami steruje komputer.

Model opisuje funkcjonowanie pewnej liczby konkurencyjnych firm na danym rynku (np. firmy produkujące dobra zaspokajające tę samą potrzebę). Firmy uczestniczące w grze mają formę spółek akcyjnych, muszą zatem płacić dywidendy akcjonariuszom oraz upubliczniać niektóre informacje o nich, np. przychody, obroty, zysk, dywidenda, zatrudnienie. Do produkcji niezbędne są: zatrudnienie, energia, linie produkcyjne, zakup materiałów (rys. 6.3). Firma może zaciągać kredyt bankowy, emitować akcje lub obligacje. Obligacje nie dają możliwości zmiany właściciela firmy (przejęcie przez konkurenta), ale muszą być zwrócone w określonym terminie wraz z oprocentowaniem. Gracz może zakupić akcje innej firmy i ewentualnie mieć wpływ na jej decyzje. Wolny kapitał można także lokować na oprocentowanym koncie w banku lub w obligacjach innych firm. Firmy płacą podatek dochodowy (można zaobserwować wpływ wysokości podatku na działanie firmy, zmieniając jego wysokość). Produkcja powoduje zanieczyszczanie środowiska, od poziomu zanieczyszczania zależy wysokość kar nakładanych na firmę przez lokalne władze. Zarówno podatki, jak i kary pobierane są automatycznie z kont firm na koniec każdego umownego roku. Firma, podejmując decyzję może korzystać z raportów finansowych. W grze istnieje możliwość zakupu analiz, prognoz i ekspertyz zewnętrznych. Firma może zbankrutować, jeśli zajdzie jeden z warunków:

- niewypłacenie dywidendy w ustalonym okresie,
- wielkość kredytów przekroczy wartość majątku firmy,
- niezapłacenie podatku dochodowego lub kar za zanieczyszczenie środowiska,
- niewypłacenie pensji pracownikom przez okres dwóch umownych miesięcy,
- zerowy stan konta firmy,
- udział firmy w rynku spadnie poniżej 1%.

Każda firma jest opisana przez zbiór rutyn, które można traktować jako odpowiedniki genotypów w biologii. Rutyny wpływają na charakterystyki firmy, takie jak techniczna jakość jej produktów, poziom organizacyjny, efektywność produkcji itp. Wszystkie firmy dążą do polepszenia swojej sytuacji na rynku przez wprowadzenie nowych kombinacji rutyn. Wprowadzanie innowacji jest istotnym elementem poprawiania sytuacji firmy. Wyróżnia się innowacje oraz wynalazki (nowości, które poten-



Rys. 6.3. Otoczenie firmy na symulowanym rynku

cialnie mogą być wprowadzone). Firma może poszukiwać innowacji przez prowadzenie autonomicznych badań oraz naśladowanie konkurentów. W modelu wyróżnione są dwa poziomy, na których charakteryzowana jest firma: poziom rutyn i poziom charakterystyk firmy. Poziom charakterystyk może być postrzegany jako odpowiednik poziomu fenotypowego w biologii.

Etap symulacji rozwoju przemysłu obejmuje te same co **EMP** cztery kroki: szukanie nowych kombinacji rutyn, obliczanie i porównywanie charakterystyk firm po zastąpieniu starych rutyn nowymi oraz ewentualna zamiana rutyn, wejście nowych firm na rynek ocenia produkty firm.

Każdy gracz (firma) podejmuje decyzje dotyczące:

- ceny produktu,
- wielkości produkcji,
 - zakupu(sprzedaży) linii produkcyjnych,
 - zatrudniania(zwalniania) pracowników,
 - zakupu(sprzedaży) surowców,
 - zakupu energii,
- wydatków na marketing,
- wydatków na badania i rozwój (R&D),
 - rozwoju technologicznego produktów,
 - inwestycji związanych z organizacją i zarządzaniem,
- wydatków na remonty linii produkcyjnych,
- wysokości dywidendy,
- zaciągania(splaty) (kredytów) globalna stopa procentowa jest zadana,
- lokowania kapitału w banku (oprocentowanie nieco niższe od stopy procentowej),
- emisji i sprzedaży akcji(obligacji),
- zakupu ekspertyz,
- zakupu patentów, licencji, wywiadu gospodarczego.

Każdy gracz może na żądanie otrzymać powstające raporty z gry. Są to:

- raport na temat rynku (podatki, kary, wskaźnik globalnej koniunktury itp.),
- raport finansowy (stan konta, wpływy, wydatki, procentowy udział firmy na rynku, wielkość zaciągniętych kredytów, wysokość dywidendy, kursy akcji poszczególnych firm itp.),
- raport produkcyjny (wielkość produkcji, koszty jednostkowe i całkowite, zużycie linii, zatrudnienie, wysokość pensji),
- raport o konkurencji (dane o poziomie sprzedaży, zatrudnieniu, konkurencyjności technicznej produktów, konkurencyjności marketingowej itp.).

Na początku gry są ustalane parametry dotyczące:

- stanu nowej firmy na rynku (początkowe zasoby finansowe i produkcyjne),
- rynku jako całości (elastyczność cenowa, współczynnik zmiany koniunktury, współczynnik straty rynku przy braku zaspokojenia popytu, prawdopodobieństwo wejścia nowej firmy na rynek, początkowa liczba firm, początkowa wielkość rynku, minimalna płaca, koszt ekspertyzy, sezonowość sprzedaży),
- stopy procentowej (zakres zmiany stopy procentowej, zakres zmiany oprocentowania obligacji),
- podatków i kar (wysokość podatku dochodowego, wysokość kar za zanieczyszczenie, możliwości zmiany podatku i kar),
- produkcji (miesięczne zużycie linii produkcyjnych, ich koszt, maksymalna wydajność linii, zasoby potrzebne do wyprodukowania jednostki produktu),
- gry: czas trwania rozgrywki i liczba „miesięcy” wykorzystana do tworzenia raportów, oraz parametry algorytmu genetycznego.

Algorytm ewolucyjny zastosowany w opisywanej grze

Pewne mechanizmy dla szerokiej klasy procesów ewolucyjnych są wspólne (selekcja połączona z konkurencją, poszukiwanie nowych rozwiązań). Podczas modelowania danego procesu należy uwzględnić jego specyfikę. Zacząć należy od pytania, co jest podmiotem ewolucji, np. w biologii organizm, w ekonomii technologia (modele substytucji technologicznej) lub firma (modele dynamiki przemysłu). Firma będzie zatem odpowiednikiem osobnika w populacji biologicznej. Firma jest opisana przez zbiór rutyn (zestaw genów). Kot (1997) zastosował algorytm ewolucyjny z diploidalną reprezentacją osobnika, tzn. osobnik składa się z par odpowiadających sobie chromosomów (nazywanych chromosomami homologicznymi). Każdy z odpowiadających sobie chromosomów koduje te same charakterystyki firmy, ale zawiera inne wartości rutyn. Która rutyna (z pierwszego czy drugiego chromosomu homologicznego) będzie brana pod uwagę przy obliczaniu charakterystyk firmy, zależy od tego, która z nich zostanie uznana za dominacyjną. Zastosowano model dominacji Hollstiena-Hollanda³ (Goldberg 1989).

³ Diploidalny trialleliczny GA Hollstiena-Holladna: alfabet składa się z trzech symboli $\{-1, 0, 1\}$, jedynka jest dominująca w stosunku do zera, -1 oznacza recesywną jedynkę $(0, 0 \rightarrow 0; 0, -1 \rightarrow 0; 0, 1 \rightarrow 1; -1, 0 \rightarrow 0; -1, -1 \rightarrow -1; -1, 1 \rightarrow 1; 1, 0 \rightarrow 1; 1, -1 \rightarrow 1; 1, 1 \rightarrow 1)$.

W swoim zbiorze rutyn firma ma również rutyny, które aktualnie nie są wykorzystywane, tzw. rutyny *ukryte* lub nieaktywne. Ukryte rutyny mogą w przyszłości stać się aktywne. Rutyny są pogrupowane w segmenty – odpowiedniki chromosomów w biologii. W każdym segmencie są rutyny aktywne i ukryte. Poszczególne segmenty odpowiadają rutynom odpowiedzialnym za określoną działalność firmy. Podobnie jak genotypy osobników, rutyny podlegają ewolucji, tzn. działają na nich operatory genetyczne, takie jak krzyżowanie, mutacja, tranzycja i transpozycja. Mutacje, rekombinacje, tranzycje i transpozycje mają takie samo znaczenie jak w **EMP**. Parametry określające stan firmy, takie jak produktywność kapitału, zmienny koszt produkcji oraz charakterystyki produktu, są wyznaczane na podstawie rutyn. Inwestycje, modernizacja, produkcja i cena są wyznaczane według kryteriów funkcjonowania firmy na rynku, np. maksymalizacja zysku. Odpowiednie wzory można znaleźć w pracach Kwaśnickiego (1994/1996) i Kota (1997).

Zbudowano pierwszą wersję programu DYPLOM, działającego w środowisku Microsoft Windows 95 (Kot 1997). Za jednostkę gry przyjęto umowny miesiąc. W każdym kroku wyróżnia się trzy fazy gry:

- fazę decyzyjną, firmy podejmują decyzje,
- fazę symulacji, wykonywane są obliczenia w modelu – rezultaty podjętych decyzji,
- fazę analityczną – gracze mogą „analizować” wyniki swoich decyzji, korzystając przy tym z dostępnych raportów.

Wykonano symulacje, które wykazały przydatność programu do celów edukacyjnych. Uwzględnienie w konkurencji firm, oprócz konkurencyjności technologicznej takich elementów, jak wizerunek firmy, kampania reklamowa czy sposób zarządzania firmą, pozwala na tworzenie bardziej skomplikowanych strategii konkurencji na rynku. Zdarzenia losowe – pojawiające się nowe firmy na rynku czy też załamanie na giełdzie (Nowicki 1992) – pozwalają graczowi sprawdzić swoje zdolności menedżerskie w zmiennym środowisku. Gra pozwala testować różne strategie podejmowania decyzji, np. maksymalizacja wydatków na badania własne firmy czy duże nakłady inwestycyjne. Dopracowania wymagają strategie podejmowania niektórych decyzji przez komputer. Zaimplementowana w programie DYPLOM strategia jest wynikiem wielu różnych eksperymentów. Uwzględnia ona takie reguły, jak: jeśli firma nie pokrywa swojego segmentu rynku, to zakup linii; jeśli ponad połowa firm ma lepszy wizerunek (image), to zwiększ o 10% wydatki na marketing; jeśli stan techniczny linii jest zużyty w ponad 75%, to zwiększ wydatki na remont itp. Poprawienie tego elementu gry umożliwiłoby opracowanie w pełni profesjonalnej, dydaktycznej gry menedżerskiej.

7. UWAGI KOŃCOWE

Zabawa w naśladowanie natury trwa. Czy tylko zabawa? Czy tak właśnie można nazwać projekt pod kierunkiem Hugo de Garis'ego, realizowany przez międzynarodowy zespół ludzi (w połowie 1998 roku zaangażowanych było 20 naukowców z 9 krajów) w *Advanced Telecommunications Research*, w japońskim mieście Kioto (Buller 1998). Projekt „CAM-Brain”, którego celem jest opracowanie sztucznego mózgu, istnieje już od kilku lat. Po ośmiu latach realizacji powinien powstać sztuczny mózg, składający się z miliarda neuronów, o inteligencji małego kotka (*Robokoneko* to japońska nazwa małego kociaka-robota – ma być on gotowy i publicznie pokazywany pod koniec 1999 roku). Jest to pierwszy etap projektu mającego za główny cel budowę sztucznego mózgu o inteligencji dorównującej człowiekowi.

CAM to nazwa pochodząca od ang. *Cellular Automata Machine* – maszynowa implementacja automatu komórkowego. Projektowany mózg to elektroniczny automat komórkowy. Mocno upraszczając, automat komórkowy to specyficzna symulacja komputerowa: definiuje się reguły dotyczące zachowania się komórki, w zależności od jej aktualnego stanu i stanów sąsiednich komórek. Układ komórek zmienia się dynamicznie, ponieważ kolejno pobierają one odpowiednie dla nich reguły i zmieniają się zgodnie z nimi. Wydaje się to proste, ale tylko zasada działania jest prosta, jak natomiast zbudować dobrze funkcjonujący CAM-Brain? A jak powstał mózg człowieka? Wzorując się na naturze, można wyewoluować sztuczny mózg. Trudno jest jednak zdefiniować funkcję oceniającą ewoluujące osobniki, tak ważną dla sztucznej ewolucji. Jak oceniać wyewoluowany mózg, ile czasu trwałoby tworzenie i ocenianie pokoleń takich mózgów!

Konstruktorzy sztucznego mózgu przyjęli inną metodę. Na drodze programowania genetycznego będą ewoluować moduły, które następnie będą łączone. Główny projektant musi określić zadania dla pojedynczych modułów, które będą dalej dekomponowane na mniejsze bloczki i dla nich będą określane zadania, te znowu będą dzielone itd., aż zejdziesz się do małych modułów-bloczków, których czas ewolucji będzie rozsądny. Znalezione na niższym poziomie bloczki będą następnie łączone w większe bloki i testowane. Cały proces może wymagać powrotów do niższych poziomów – trzeba będzie modyfikować małe bloczki, by większy blok wypełniał przeznaczone mu zadanie. Czy nadal zadanie budowy sztucznego mózgu wydaje się być mrzonką czy też trudnym, ale już realnym zadaniem? Więcej informacji – publikacje, raporty itp. – można znaleźć na stronie internetowej ATR: <http://www.hip.atr.co.jp/~degaris>.

Jak widać, popularność obliczeń ewolucyjnych trwa. Czy wśród tych wszystkich prac jest miejsce dla zaproponowanego w niniejszej pracy *K*-modelu? Wydaje się, że odpowiedź twierdząca jest uzasadniona. Główne cechy wyróżniające prezentowany model to nadmiarowość genetyczna, plejotropowość i poligeniczność. Czynniki te umożliwiają efektywne sterowanie możliwościami ewoluującej populacji. Po dodaniu do tego rekrudescencji i kryzysu dysponujemy całkiem mocnym narzędziem do szukania optimum globalnego. Jak w każdym modelu, tak i tutaj dobór parametrów podczas symulacji jest krytyczny dla działania *K*-modelu. Najprostszy klasyczny **GA** również podlega tej regule. W *K*-modelu mamy znacznie więcej parametrów do ustalenia niż w prostym **GA**, wydaje się więc, że próby zastosowania go sprawią większą trudność. Można powiedzieć – nic za darmo, proponowany tu model ma większe zdolności optymalizacyjne. Rozumiejąc go, można – przez dobór macierzy plejotropowości **A** i parametrów oraz odpowiednią zmianę jego parametrów już w czasie symulacji – otrzymać optimum globalne trudnych funkcji w niezbyt długim czasie i następnie „dostroić” rozwiązanie. Dwa niezbędne do tego wymagania są trudne do spełnienia:

1. Dobre rozumienie działania *K*-modelu i jego wrażliwości na parametry (to można spełnić, aczkolwiek wymaganie takie w stosunku do każdego potencjalnego użytkownika powoduje znaczne ograniczenie użyteczności modelu).

2. Znajomość „z góry” kształtu optymalizowanej funkcji – to wymaganie jest całkiem nierealne, gdy mówimy o rzeczywistych problemach.

Znowu można powiedzieć, że wymagania te jednakowo odnoszą się do wszystkich algorytmów ewolucyjnych. Z reguły w doborze parametrów stosuje się metodę prób i błędów – chociażby przy korzystaniu z profesjonalnego narzędzia optymalizacyjnego na bazie klasycznego algorytmu genetycznego C DARWIN II, firmy NovaCast. Możliwe, że uda się ominąć tę trudność. Ostatnio rozpoczęliśmy prace nad zaprojektowaniem uniwersalnego narzędzia optymalizującego, którego parametrami będzie sterował *inteligentny moduł sterujący*.

Omówione w niniejszej pracy badania własności modelu służą zebraniu jak najpełniejszej wiedzy na temat zależności tempa i charakteru ewolucji (duże skoki w przestrzeni ewolucyjnej, łagodne „zsuwanie się” populacji po zboczach szczytów itp.) oraz wrażliwości modelu na poszczególne jego parametry.

Intrygującym wynikiem w opisanych badaniach była zależność ewolucji od liczebności populacji. Okazuje się, że stereotyp – liczniejsza populacja wymaga mniej-
szej liczby pokoleń na znalezienie rozwiązania – nie jest zawsze prawdziwy. W *K*-modelu dla dwóch wymiarów populacja dziesięcioelementowa szybciej znajduje akceptowalne rozwiązanie niż populacja licząca 100 lub 1000 osobników. Dla genotypów bez genów nadmiarowych nie jest to już prawdą, jest wręcz odwrotnie. Jeśli nawet uwzględnimy geny nadmiarowe, ale dopuścimy bardzo małą maksymalną zmianę mutacyjną pojedynczego genu (np. 10 przy początkowych wartościach genów 1000), to nadal populacja duża będzie ewoluować krócej (mierząc ją liczbą pokoleń), ale zysk jest niewielki w stosunku do małych populacji, np. przy liczebnościach populacji

10:100:1000 średni czas znalezienia akceptowalnego rozwiązania jest 667:656:452 pokoleń. Biorąc pod uwagę czas przetwarzania dużych i małych populacji – nic nie zyskujemy, realnie czekamy na rozwiązanie dłużej przy dużych populacjach. Charakterystyczny jest też efekt wymiaru optymalizowanej funkcji. Dla funkcji trzydziestowymiarowej większe populacje lepiej sobie radziły, ponieważ małej populacji, której ruch w przestrzeni fenotypowej bardziej przypomina błądzenie losowe, trudniej jest znaleźć się na samym szczycie. Duża populacja po znalezieniu rozwiązania w okolicy wysokiego szczytu znacznie łatwiej go osiąga. Biorąc jednak czas obliczeń – liczenie przystosowania 100 osobników przez 3500 pokoleń zajmuje więcej czasu (350 000 obliczeń funkcji celu) niż 10 osobników nawet przez 7000 pokoleń (70 000 obliczeń).

Wiadomo, że rekombinacje i mutacje są źródłem zmienności w populacji. Duża częstość mutacji nie daje jednak takiego efektu jak bardzo częste rekombinacje – te ostatnie mogą dać efekt chaotycznego ruchu. Jako mechanizm do przeszukiwania rozległych obszarów przestrzeni rozwiązań¹ mogą się nadawać, ale nie pozwalają populacji wykorzystać do końca znalezionych obszarów. Częste mutacje powodują, że populacja zajmuje (bada) stosunkowo rozległy obszar przestrzeni, nie tracąc do końca możliwości wykorzystania znalezionych rejonów.

Makromutacje to czasami jedyna możliwość ucieczki z pułapki ewolucyjnej. Bardzo ciekawy efekt daje operator nazwany *kryzysem* – duża reorganizacja genotypów u wszystkich osobników w populacji co zadaną liczbę pokoleń. Badacze algorytmów genetycznych podkreślają konieczność zachowania równowagi pomiędzy *exploring* a *exploiting*, czyli z jednej strony algorytm musi mieć zdolności do penetracji przestrzeni rozwiązań, z drugiej zaś rozwiązania w dobrych rejonach (np. na zboczu globalnego szczytu) powinny być zastosowane do sprawdzenia, jak wysoki jest ten szczyt. Operatory EA, powodujące dużą zmienność populacji (częste mutacje i krzyżowanie), zazwyczaj przeszkadzają efektywnie wspinać się na szczyt. W K-modelu operator kryzysu pozwala na zachowanie takiej równowagi: wystąpienie kryzysu to faza znacznie zwiększonych poszukiwań odległych rozwiązań, natomiast okres pomiędzy kryzysem to raczej spokojna ewolucja, pozwalająca wykorzystać znalezione rozwiązania i zbadać okoliczne rejony. Oczywiście, powstałe w obiecującym rejonie osobniki są gubione, selekcja nie zapewnia, że przetrwają one do następnego pokolenia, nawet jeśli są lepsze od pozostałych, a często są to osobniki z drugiej strony doliny przystosowawczej, na sąsiednim, wyższym szczycie, ale ich przystosowanie jest nadal gorsze niż najlepszych osobników w populacji. To tylko kwestia czasu, jeśli skutek kryzysu pojawiają się elementy populacji na innym szczycie, to w którymś z okresów spokojnej ewolucji rozmnożą się i zapoczątkują zasiedlanie tego szczytu. Operator kryzysu wydaje się być mocnym atutem proponowanego modelu.

¹ D.E. Goldberg w *The Design of Innovation: Lessons from Genetic Algorithms, Lessons for the Real World*, raport IlliGAL No. 98004, 1998, s. 3 cytuje J. Hadamarda: „...jest oczywiste, że wynalazki lub odkrycia, mające miejsce w matematyce lub gdziekolwiek indziej, dokonywane są przez łączenie idei”.

Na podstawie wyników symulacji można stwierdzić, że opłacalne jest podzielenie populacji na pewną liczbę prawie izolowanych demów, między którymi zachodzi wymiana osobników. Interesujące jest, że opłaca się wyróżnić jeden większy dem i niewielką liczbę małych demów. Małe demy oznaczają tu od kilku do dziesięciu osobników. Tak podzielona populacja wydaje się lepiej radzić sobie ze znajdowaniem optimum.

Zachowywanie najlepszego osobnika w populacji – jego bezwarunkowe przechodzenie bez zmian do następnego pokolenia – jest z jednej strony opłacalne, z drugiej nie. Jeśli populacja wspina się po zboczu, opcja taka przyspiesza ewolucję, natomiast przy wielomodalnych funkcjach jego rola nie jest jednoznaczna. Jeśli populacja znajdzie rozwiązanie po drugiej stronie doliny przystosowawczej, lepsze od najlepszego dotychczas, to nie powinna go gubić. Osobnik taki powinien zapoczątkować zasiedlanie nowego szczytu. Jeśli jednak znalezione rozwiązanie będzie gorsze od aktualnie najlepszego, to zachowywanie najlepszego nie jest korzystne, powoduje bowiem, że ten nowy osobnik ma mniejsze szanse przetrwania. Wynika to z tego, że populacja zachowująca najlepszego osobnika zasiedla rejon bliższy zajmowanemu szczytowi, a zatem ma większe średnie przystosowanie niż bez takiej opcji. Wyższa średnia jakość populacji powoduje, że nowy, odległy od szczytu osobnik jest mniej konkurencyjny. W przypadkach, kiedy niemożliwe jest przeskoczenie populacji na sąsiedni szczyt i jedynym wyjściem jest zejście populacji niżej w dolinę, by móc osiągnąć następny szczyt, opcja „zachowaj najlepszego” wręcz uniemożliwia dalszą ewolucję.

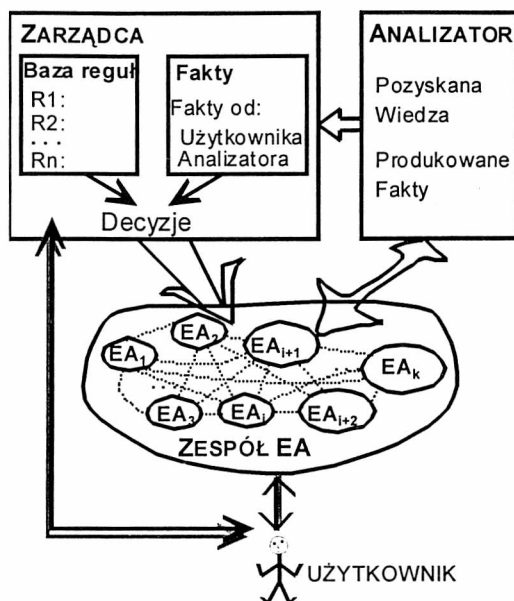
Macierz pleiotropowości A może być stosowana do ukierunkowywania ewoluującej populacji. Eksperymenty wskazują jednoznacznie, że jednorodna macierz składająca się z elementów tylko dodatnich (lub tylko ujemnych) jest najmniej efektywna. Nawet jeśli wymagane są zmiany fenu tylko w jednym kierunku (np. zwiększanie go doprowadziłoby do optymalnej wartości), to przecież geny zmieniają się losowo w obu kierunkach, zarówno zmniejszają, jak i zwiększają swoją wartość, dlatego też macierz o elementach podobnych co do wartości nie daje dobrych rezultatów. Oczywiście jest, że macierz A działa w połączeniu z wartościami genów, dlatego te wielkości powinny być dobierane łącznie. Program K -model pozwala ustalać wartości genów, w tym losowo z zadanego przedziału, pozwala na ustalenie np. części elementów A i losowanie pozostałych elementów tak, aby cała populacja początkowa (lub jej część) była umieszczona w zadanym punkcie. Liczbą i miejscem zer w macierzy decydujemy o rozmiarze pleiotropowości i poligeniczności, wartości zaś pozostałych elementów decydują o sile i kierunku oddziaływania genów na feny. Jeśli obserwujemy zbyt wolną zmianę jednego z argumentów funkcji, to możemy zwiększyć odpowiedni element macierzy A , jeśli zmiany zachodzą w niepożądanym kierunku – możemy zmienić znak odpowiedniego elementu (lub odpowiednich elementów) A . Daje to duże możliwości zwiększenia efektywności ewolucji, co pokazano w niektórych eksperymentach z wielomodalnymi i wielowymiarowymi funkcjami.

Ze względu na to, że na genach wykonywanych jest więcej operacji, opłaca się przyjmować je jako wartości całkowite, obliczenia na nich są szybsze. Elementy macierzy A muszą należeć do zbioru liczb rzeczywistych, aby możliwe były rzeczywiste wartości argumentów funkcji celu. W modelu należy mnożyć wartości genów przez macierz A , ale w zamian nie wykonuje się dekodowania binarnych chromosomów. W eksperymentach prowadzonych dla tych samych funkcji i rozmiarów populacji z H -modelem (GA z kodowanymi binarnie chromosomami) i K -modelem (np. podczas badania wpływu podziału populacji na demy czy wpływu plejotropowości i poligeniczności) nie obserwowano, by K -model dłużej wykonywał przetwarzanie kolejnych pokoleń, dlatego też nie było to specjalnym przedmiotem badań.

Chcielibyśmy wiedzę dotyczącą efektywnego ustalania wartości parametrów wpływających na ewolucję przekazać systemowi komputerowemu i niech on się „martwi” o to, jakie wybrać operatory, jak i kiedy modyfikować parametry itp. Również niech zadaniem systemu będzie monitorowanie i analizowanie osiąganych rezultatów, co umożliwi mu identyfikację problemów i zastosowanie posiadanej wiedzy do przeciwdziałania im. Rozpoczynamy prace mające na celu zaprojektowanie i budowę takiego systemu. Przyszły system, nazwijmy go roboczo **INT-EVOL** (*inteligentna ewolucja*), będzie składał się z trzech zasadniczych modułów (rys. 7.1): inteligentnego modułu sterującego (**ZARZĄDCA**), inteligentnego modułu analizującego (**ANALIZATOR**) i zespołu ewoluujących populacji (**Zespół EA**)².

Każda ewoluująca populacja ma swój zestaw parametrów, wynikający z jej roli w rozwiązywaniu zadania. Będą na przykład małe demy (tzw. szperacze), ukierunkowane na szerokie przeszukiwanie przestrzeni fenotypowej, ich liczba nie powinna być mała, ale też i nie za duża. Inne algorytmy (wspinacze) miałyby za zadanie szybko wspiąć się na szczyty odkryte przez grupę szperaczy. **ANALIZATOR** kontroluje działania poszczególnych **EA**, analizuje ich efektywność i na tej podstawie identyfikuje ewentualne problemy. Wynik analizy przekazywany do **ZARZĄDCY** posłuży temu ostatniemu do interwencji. **ZARZĄDCA**, wykorzystując swoją wiedzę (najprawdopodobniej będzie to system ekspertowy z wiedzą zebraną w postaci reguł), będzie modyfikował odpowiednie **EA** tak, aby lepiej wypełniały swoją rolę. Poszczególne **EA** mogą stosować różny sposób kodowania – jedne mogą być binarne (demy binarne), inne typu K -model (demy plejotropowe), jeszcze inne mogą operować bezpośrednio na fenotypach (demy fenotypowe). Pomiedzy osobnikami z różnych demów nie może zachodzić krzyżowanie, ale możliwa jest między nimi migracja. Pomiedzy demami binarnymi i demami fenotypowymi migracja jest możliwa w obie strony, łatwe jest przekodowanie genotypów. Migracja z demów plejotropowych do pozostałych też jest

² Wyniki uzyskane w realizowanym obecnie projekcie *Komplementarne algorytmy genetyczne* wskazują na sensowność proponowanego systemu **INT-EVOL**. W tej pracy rozważane są dwa algorytmy z kodowaniem binarnym o różnych parametrach krzyżowania i mutacji, wymieniające między sobą najlepszego osobnika. Ich efektywność w pokonywaniu lokalnych szczytów i osiąganiu globalnego optimum dla wielomodalnej funkcji dwuwymiarowej jest znacznie lepsza niż pojedynczego **GA**.



Rys. 7.1. Schemat systemu INT-EVOL

prosta, ale tylko w tym jednym kierunku. W drugą stronę automatyczne przeliczanie genów jest trudne nie tyle od strony realizacji, ile mogą się pojawić problemy, jeśli dany zestaw fenów nie da się reprezentować przy stosowanej przez dem macierzy plejotropowości i dopuszczalnych wartościach genów. Baza wiedzy dla ZARZĄDCA, jak i sposób analizy przebiegu ewolucji i identyfikacji problemów dla ANALIZATORA muszą być opracowane i przetestowane. **INT-EVOL** powinien umożliwiać użytkownikowi definiowanie swojego problemu w terminach akceptowanych przez system (podobnie jak umożliwia to system **C DARWIN II**). Zestaw domyślnych wartości parametrów i inteligentny system sterujący powinny zdjąć z użytkownika obowiązek długiego eksperymentowania z różnymi parametrami, jak również zapewnić, by ewolucja była efektywna.

Baza wiedzy dla planowanego systemu nie jest gotowa, można jedynie podać pewne przesłanki do jej tworzenia:

- liczba osobników w szperaczach powinna być mała (ok. 10); jeden z nich może być większy od pozostałych (może kilka),
- szperaczami powinny być demy plejotropowe z dość częstymi makromutacjami – w miarę często powinien występować operator zwany kryzysem,
- niektóre demy szperacze mogą zachowywać najlepszego osobnika w populacji, ale nie wszystkie,
- wspinacze powinny mieć większą liczbę osobników, mniejsze wartości parametrów odpowiedzialnych za różnorodność osobników (częstość mutacji i rekombinacji, mniejsza maksymalna zmiana mutacyjna, niewielka nadmiarowość genetyczna, kryzys – jeśli obecny, to rzadko),

- powinno być znacznie więcej szperaczy niż wspinaczy, większość tych ostatnich powinna zachowywać najlepszego osobnika w populacji,
- demy binarne w roli szperacza powinny mieć niezbyt małe prawdopodobieństwo krzyżowania i stosunkowo duże prawdopodobieństwo mutacji bitu,
- demy fenotypowe w roli szperacza powinny mieć stosunkowo duże prawdopodobieństwo mutacji i maksymalną zmianę mutacyjną,
- każdy szperacz powinien kopię najlepszego osobnika wysyłać do wspinaczy,
- dla demu plejotropowego w roli szperacza, gdy zróżnicowanie populacji jest zbyt małe (ze względu na wartości fenów), należy zwiększyć maksymalną zmianę mutacyjną, jeśli ta jest w miarę duża – zwiększyć częstość kryzysu, jeśli ta jest dość duża – częstość transpozycji i tranzycji (makromutacji wynikających z nadmiarowości genetycznej),
- jeśli w demie plejotropowym jest obserwowany mały rozrzut jednego z fenów, należy zmienić odpowiedni element macierzy plejotropowości A ,
- modyfikacja A może pomóc, jeśli populacja na długo „utknęła” w jednym obszarze przestrzeni fenotypowej,
- jeśli przy sporym zróżnicowaniu fenów w demie zróżnicowanie przystosowania osobników jest zbyt małe, należy wprowadzić skalowanie funkcji przystosowania,
- jeśli wartości fenów osobników w demie często przekraczają dopuszczalne zakresy ich zmienności, należy wprowadzić do funkcji przystosowania (lub zwiększyć) wartości kar.

Jest to wstępna propozycja ogólnej wiedzy, którą należy przekazać ZARZĄDCY, w trakcie realizacji projektu opracowana zostanie dokładna baza reguł. Po doświadczeniach z prototypem systemu możliwe będzie jego rozszerzenie o reguły rozmyte wraz z funkcjami przynależności. Zaprojektowanie ANALIZATORA wydaje się łatwiejsze, choć z całą pewnością to również będzie wymagać wielu prób.

O ile wiadomo, dotychczas nie wykonywano prób wykorzystania efektów plejotropowości i poligeniczności przy tworzeniu NN^3 . Jednym z planów przewidzianych do realizacji jest włączenie modelu z plejotropowością i poligenicznością do projektowania sieci neuronowych. Wydaje się możliwe i sensowne zastosowanie takiego podejścia do nauczania sieci neuronowych. Jak wspomniano wcześniej (p. 2.3), **GA** potrafi dość szybko znacznie obniżyć błąd odpowiedzi sieci, ale nie potrafi dostroić rozwiązania. Wstępne testy wykazują⁴, że warto sprawdzić, czy proponowane tu po-

³ Altenberg (1994) zastosował plejotropowość do tworzenia początkowego genotypu dla zadania optymalizacji funkcji będącej sumą funkcji składowych. Liczba genów była równa liczbie składowych funkcji. Binarny genotyp tworzony był cyklicznie przez: próbę dodania nowego genu; wylosowania dla niego funkcji, na które on będzie oddziaływał; akceptacji dodanego genu, jeśli poprawia on przystosowanie tworzonego osobnika, odrzucenie, jeśli pogarsza; przeprowadzenie mutacji genotypu (jeśli jest korzystna). Czynności powtarzano, aż genotyp osiągnął zadaną długość. Istotne spostrzeżenie Altenberga to, że rozmiar plejotropowości dla początkowych genów był znacznie większy, zmniejszał się dla kolejno dodawanych genów.

⁴ Prace ze studentami specjalizacji Inżynieria oprogramowania w ramach kursu *Algorytmy genetyczne*.

dejsście da lepsze efekty w genetycznym nauczaniu sieci neuronowych. Jeśli wektor wag potraktujemy jako fenotyp, to zastosowanie *K*-modelu nie wydaje się sprawiać trudności. Pozostaje problem z czasem obliczeń, ale to należy rozpatrywać w połączeniu z niezbędną liczbą pokoleń do uzyskania satysfakcjonującego rozwiązania. Ze wstępnych szacowań wynika, że podejście to byłoby opłacalne w stosunku do prezentowanego w (Montana, Davis 1989) przy dwukrotnym zmniejszeniu liczby pokoleń.

Pomysł Altenberga z optymalizacją początkowych warunków do nauczania sieci (zestawu wag sieci) według „rozrostowego” algorytmu też wydaje się sensowny. Należy sprawdzić, czy taki sposób tworzenia początkowego wektora wag pozwala na lepsze nauczanie sieci metodą wstecznej propagacji błędu lub uczenia genetycznego.

Kolejny temat, który należałoby kontynuować, wynika z dotychczasowych doświadczeń z realizacją prostej gry menedżerskiej. Jest on bardzo interesujący i obiecujący. Prace powinny być prowadzone w zespole interdyscyplinarnym (specjalistów od organizacji i zarządzania, ekonomistów, praktykujących menedżerów), by doprowadzić do powstania edukacyjnej gry menedżerskiej, systemu komputerowego, który – oprócz typowych zadań edukacyjnych dla uczestników kursów z zakresu organizacji i kierowania – będzie umożliwiał badania dotyczące zachowania się różnych strategii firm funkcjonujących w różnych środowiskach. Można na przykład założyć, że firma kieruje się kryterium maksymalizacji krótkoterminowego zysku, a firmy na konkurencyjnym rynku mogą się kierować:

- każdą inną strategią,
- różnymi strategiami z przewagą jednej, np. maksymalizacji zysku długoterminowego,
- tą samą, wybraną strategią (np. proinnowacyjną), w zmieniającym się środowisku itp.

Można pokusić się również o automatyczne wygenerowanie strategii radzącej sobie dobrze na poszczególnych rynkach. Proponowany tu projekt wymaga zdefiniowania analizowanych strategii oraz pełniejszego modelu funkcjonowania firm stosujących poszczególne strategie. Do tego potrzebny jest udział specjalistów od zarządzania i (lub) ekonomii.

Wymienione trzy główne kierunki dalszej pracy są ciekawe i obiecujące, trudno jest dokonać między nimi wyboru. Wydaje się realne prowadzenie prac równolegle, ponieważ wszystkie opierają się na zastosowaniu odpowiednio zaprojektowanego algorytmu ewolucyjnego.

LITERATURA

Aşveren T., Molitor P. (1996), *New Crossover Methods for Sequencing Problems*, Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, September 1996, Berlin.

Altenberg L. (1994), *Evolving Better representations through Selective Genome Growth*, Proceedings of the IEEE 1994 World Congress on Computational Intelligence, s. 182–187, (dostępne pod adresem: <http://puco.mhpcc.edu/~altenber/PAPERS/>).

Back T. (1996), *Evolutionary Algorithms in Theory and Practice*, New York, Oxford, Oxford University Press.

Bala J., Huang J., Vafaie H., DeJong K., Wechsler H. (1995), *Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification*, IJCAI Conference, Montreal.

Beasley D., Bull D.R., Martin R.R. (1994), *An Introduction to Genetic Algorithms*, Vivek, A Quarterly in Artificial Intelligence, January 1994, National Centre for Software Technology, Bombay.

Beasley D., Bull D.R., Martin R.R. (1994a), *Research Topics in Genetic Algorithms*, Vivek, A Quarterly in Artificial Intelligence, April 1994, National Centre for Software Technology, Bombay.

Bolc L., Cytowski J. (1989), *Metody przeszukiwania heurystycznego* (t. I i II), Warszawa, PWN.

Bornholdt S., Graudenz D. (1992), *General Asymmetric Neural Networks and Structure Design by Genetic Algorithms*, Neural Networks, Vol. 5, s. 327–334.

Buckles B.P., Petry F.E. (ed.) (1992), *Genetic Algorithms*, IEEE Computer Society Press Technology Series, Los Alamitos, California.

Buller A. (1998), *Sztuczny mózg, to już nie fantazje*, Warszawa, Prószyński i S-ka.

Cantú-Paz E. (1995), *A summary of Research on Parallel Genetic Algorithms*, IlliGAL Report 95007, Urbana, Illinois, <http://gal4.ge.uiuc.edu>.

Cantú-Paz E. (1997), *Designing Efficient Master-Slave Parallel Genetic Algorithms*, IlliGAL Report 97004, <http://gal4.ge.uiuc.edu>.

Cantú-Paz E. (1997a), *A Survey of Parallel Genetic Algorithms*, IlliGAL Report No. 97003, <http://gal4.ge.uiuc.edu>.

Cantú-Paz E., Goldberg D.E. (1999), *Parallel genetic algorithms with distributed panmictic populations*, IlliGAL Report 99006, <http://gal4.ge.uiuc.edu>.

Cantú-Paz E. (1999a), *Topologies, migration rates and multi-population parallel genetic algorithms*, IlliGAL Report 99007, <http://gal4.ge.uiuc.edu>.

Cińcio S., Kwaśnicka H., Markowska-Kaczmarska U. (1999), *Parallel genetic algorithm as a method of neural network architecture design*, IV Conference Neural Networks and Their Applications, Politechnika Częstochowska i Polskie Towarzystwo Sieci Neuronowych, Zakopane (w druku).

Coleman K. (1993) *The AI Marketplace in the Year 2000*, AI EXPERT, January, s. 35–38.

Conor R. (1997), *Diploidy without Dominance*, Proceedings of the 3NWGA, Helsinki.

- Cordon O., Herrera F., Lozano M.** (1996), *On the Bidirectional Integration of Genetic Algorithms and Fuzzy Logic*, Technical report DECSAI 95129, December, <http://decsai.ugr.es/~lozano/public.html>.
- Cox E.** (1992), *The Great Myths of Fuzzy Logic*, AI EXPERT, January, s. 40–45.
- Dasgupta D., McGregor D.R.** (1992), *Designing Application-Specific Neural Networks Using the Structured Genetic Algorithms*, International Workshop on Combinations of Genetic Algorithms and Neural Networks, COGAN-92, IEEE Computer Society Press.
- Davis L.** (1987), *Genetic Algorithms and Simulated Annealing*, San Mateo, California, Morgan Kaufmann Publishers, Inc.
- Davis L.** (ed) (1991), *Handbook of Genetic Algorithms*, New York, Van Nostrand Reinhold.
- Dawkins R.** (1995), *Rzeka genów*, Warszawa, Wydawnictwo CIS, Oficyna Wydawnicza MOST.
- Dawkins R.** (1996), *Samolubny gen*, Warszawa, Prószyński i S-ka.
- Dąbrowski T.** (1998) *Konstrukcja funkcji oceniającej w grach logicznych za pomocą algorytmów genetycznych*, praca magisterska, Wydziałowy Zakład Informatyki, Politechnika Wrocławska, Wrocław.
- De Jong K.** (1998), *Evolutionary Computation: Where We Are and Where We're Headed*, Fundamenta Informaticae, Vol. 35, No. 1–4, s. 247–259.
- Eiben A.E., Schippers C.A.** (1998), *On Evolutionary Exploration and Exploitation*, Fundamenta Informaticae, Vol. 35, No. 1–4, s. 35–50.
- Fogel D.B.** (1992), *Evolving Artificial Intelligence*, A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Engineering Sciences, San Diego, University of California.
- Fogel D.B.** (1995), *Evolutionary Computation Toward a New Philosophy of Machine Intelligence*, New York, IEEE Press, The Institute of Electrical and Electronics Engineers, Inc.
- Fogel L.J.** (1996) *Artificial Intelligence Through Simulated Evolution*, Chichester, Wiley.
- Fogel D.B.** (1998), *Unearthing a Fossil from the History of Evolutionary Computation*, Fundamenta Informaticae, Vol. 35, No. 1–4, s. 1–16.
- Freeman J.A., Skapura D.M.** (1992), *Neural Networks, Algorithms, Application and Programming Techniques*, Addison-Wesley Publishing Company.
- Galar R.** (1984), *The rate of development of evolutionary processes*, Proceedings of VI International Congress of Cybernetics and Systems, Paris.
- Galar R.** (1985), *Handicapped individua in evolutionary processes*, Biological Cybernetics, Vol. 51, s. 1–9.
- Galar R.** (1990), *Evolutionary Search with Soft Selection*, Biological Cybernetics, Vol. 60, s. 357–364.
- Galar R.** (1990a), *Miękka selekcja w losowej adaptacji globalnej w R^n . Próba biocybernetycznego ujęcia rozwoju*, Prace naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej nr 84, Wrocław.
- Galar R.** (1991), *Simulation of Local Evolutionary Dynamics of Small Populations*, Biological Cybernetics, Vol. 65, s. 37–45.
- Galar R.** (1994), *Soft Selection in Global Optimization*, XVII krajowa konferencja: Teoria obwodów i sieci elektronicznych, Wrocław–Polanica Zdrój.
- Galar R., Kwaśnicka H., Kwaśnicki W.** (1983), *Technological substitution forecasting with a model based on biological analogy*, Technological Forecasting and Social Change, Vol. 23, No. 1, s. 41–58.

- Giguère P., Goldberg D.B.** (1998), *Population Sizing for Optimum Sampling with Genetic Algorithms: A Case Study of the Onemax Problem*, Proceedings of the Third Annual Genetic Programming Conference, Madison, 1998, dostępny pod adresem: <http://gal4.ge.uiuc.edu>.
- Goldberg D.E.** (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc.
- Goldberg D.E.** (1993), *A Wright brothers theory of genetic algorithm flight*, ISCIe, Vol. 37, No. 8, s. 450–458.
- Goldberg D.E.** (1993a), *The Wright brothers, genetic algorithms, and the design of complex systems*, Proceedings of SYNAPSE'93.
- Goldberg D.E.** (1994), *Genetic and Evolutionary Algorithms Come of Age*, Communications of the ACM, March, Vol. 37, No. 3.
- Goldberg D.E.** (1995), *Algorytmy genetyczne i ich zastosowania*, Warszawa, WNT.
- Goldberg D.E., Kargupta H., Horn J., Cantu-Paz E.** (1995), *Critical Deme Size For Serial and Practical Genetic Algorithms*, IlliGAL Report No. 95002, (<http://gal4.ge.uiuc.edu>).
- Goldberg D.E.** (1991), *Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking*, Complex Systems, 5(2), s. 139–168.
- Goldschmidt R.** (1940), *The Material Basis of Evolution*, New Haven: Yale University Press.
- Goonatilake S., Traleaven P.** (ed.) (1995), *Intelligent Systems for Finance and Business*, Wiley.
- Gould S.J.** (1991), *Niewczesny pogrzeb Darwina*, Warszawa, PIW.
- Gwiazda T.** (1995), *Algorytmy genetyczne. Wstęp do teorii*, Biblioteka Sztucznej Inteligencji, Warszawa.
- Harp S.A., Samad T., Guha A.** (1989), *Towards the genetic synthesis of Neural Networks*, Proceeding of the third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., San Mateo, California.
- Harik G., Lobo F.** (1999), *A parameter-less genetic algorithm*. Technical Report 99009, <http://gal4.ge.uiuc.edu>
- Heider H., Drabe T.** (1997), *Fuzzy System Design with a Cascaded Genetic Algorithm*, IEEE International Conference on Evolutionary Computation (IEC-97).
- Heider H., Tryba V., Mühlenfeld E.** (1994), *Automatic Design of Fuzzy Systems by Genetic Proc.* of the Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU. Paris, s. 665–670.
- Herrera F., Lozano M.** (1995), *Generating Fuzzy Rules From Examples Using Genetic Algorithms*, <http://decsai.ugr.es/~lozano/public.html>
- Hoffman A.** (1983), *Wokół ewolucji*, Biblioteka Myśli Współczesnej, Warszawa PIW.
- Holland J.H.** (1975), *Adaptation in Natural and Artificial Systems*, The University of Michigan.
- Horn J., Nafpliotis N.** (1993), *Multiobjective optimization using the niched Pareto genetic algorithm*, Technical Report No. 93006, IlliGAL, Illinois, 1993. (<http://GAL4.GE.UIUC.EDU>)
- Hutchinson A.** (1994), *Algorithmic Learning*, Oxford, Clarendon Press.
- Kacprzak T., Ślot K.** (1995), *Sieci neuronowe komórkowe: Teoria, projektowanie, zastosowania*, Warszawa, Wydawnictwo Naukowe PWN.
- Kloch J.** (1996) *Świadomość komputerów? Argument „Chińskiego pokoju” w krytyce mocnej sztucznej inteligencji według Johna Searle’a*, Ośrodek Badań Interdyscyplinarnych przy Papieskiej Akademii Teologicznej, Kraków.

Knight L., Sen S. (1995), *PLEASE: A Prototype Learning System Using Genetic Algorithms*, Proceedings of the Sixth International Conference on Genetic Algorithms (s. 429–435), Pittsburgh, Pennsylvania USA, (<http://euler.mcs.utulsa.edu/~sandip/PLEASE.ps>).

Korbicz J., Obuchowicz A. (1994), *Sztuczne sieci neuronowe. Podstawy i zastosowania*, Warszawa, Akademicka Oficyna Wydawnicza PLJ.

Kot W. (1997), *Konkurencja firm na rynku – gra symulacyjna*, praca magisterska, Wydziałowy Zakład Informatyki, Wydział Informatyki i Zarządzania, Politechnika Wrocławska, Wrocław.

Koza J.R. (1990), *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*. Report No. STAN-CS-90-1314, Stanford University.

Koza J.R. (1996), *Future Work and Practical Applications of Genetic Programming*, Version 3, June 25, for "Handbook of Evolutionary Computation", dostępny w sieci <http://www-cs-faculty.stanford.edu/~koza>.

Kozieja R. (1997), *Zastosowanie algorytmów genetycznych w modelowaniu ekonomicznym: poszukiwanie optymalnej strategii gry kierowniczej*, praca magisterska, Wydziałowy Zakład Informatyki, Wydział Informatyki i Zarządzania, Politechnika Wrocławska, Wrocław.

Kwaśnicka H. (1994), *Algorytmy genetyczne w rozwiązywaniu realnych problemów – Aktualny trend w informatyce*, Informatyka na wyższych uczelniach i dla gospodarki narodowej, materiały konferencyjne, Gdańsk, Politechnika Gdańska, Wyd. Elektroniki, t. 2., s. 136–140.

Kwaśnicka H. (1994a), *Próba naśladowania natury – algorytmy genetyczne w informatyce*, Informatyka, nr 12, s. 12–16.

Kwaśnicka H. (1995), *Neural Network Design with Genetic Algorithm*, Computer Science, Proceedings of the Conference, Department of Computer Science VSB – Technical University, Ostrava, s. 386–393.

Kwaśnicka H. (1996), *Włączanie sztucznej inteligencji do komputerowych systemów czasu rzeczywistego*, Systemy czasu rzeczywistego'96, Szklarska Poręba 10–13 września 1996, materiały konferencyjne III konferencji, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, s. 106–115.

Kwaśnicka H. (1997), *Evolutionary Approach to Artificial Neural Network Design – Good Way or Blind Alley*, MENDEL'97 III International Mendel Conference on Genetic, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets, Brno, Technical University of Brno, s. 342–347.

Kwaśnicka H. (1997a), *Nadmiarowość genetyczna, poligeniczność i pleiotropowość w algorytmach ewolucyjnych*, KAEiOG'97, Ryty, materiały II krajowej konferencji: Algorytmy ewolucyjne i optymalizacja globalna, Politechnika Warszawska, Wyd. Elektroniki i Technik Informacyjnych, s. 137–144.

Kwaśnicka H. (1997b), *Pleiotropy and Polygene in Evolutionary Algorithms*, VI międzynarodowe sympozjum: Inteligentne systemy informacyjne, Zakopane, Warszawa, IPI PAN.

Kwaśnicka H. (1997c), *Redundancy of Genotypes as the Way for Some Advanced Operators in Evolutionary Algorithms – Simulation Study*, VIVEK A Quarterly in Artificial Intelligence, National Centre for Software Technology, Mumbai, Vol. 10, No. 3.

Kwaśnicka H. (1997d), *The Role of Redundant Genes in Evolutionary Algorithms – Simulation Study*, 15 IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, Berlin, 20–25 August 1997, Proceedings, Vol. 2: Numerical Mathematics, s. 645–650, Berlin, W&T.

Kwaśnicka H. (1998), *Evolutionary Computation in Applications*, The invited lecture at the Seventh International Colloquium on Numerical Analysis and Computer Science with Applications, (wykład wygłoszony, dostępny w <http://www.ci.pwr.wroc.pl/~kwasnick>), August 13–17, 1998, Plovdiv.

Kwaśnicka H. (1998a), *Efficiency of Genetic Algorithms with Pleiotropy and Polygene Effect – Simulation Study*, VI międzynarodowe sympozjum Inteligentne systemy informacyjne, Malbork.

Kwaśnicka H. (1998b), *Algorytmy genetyczne w uczeniu się maszyn*, konferencja: Pozyskiwanie wiedzy z baz danych, Rzeczka, 8–10 maja, materiały konferencyjne Akademii Ekonomicznej we Wrocławiu, Wrocław, s. 148–169.

Kwaśnicka H., Kilińska M., (1995), *Application of genetic algorithm to neural networks design*, Proceedings of the 12th International Conference on Systems Science, ed. Z. Bubnicki, Wrocław, 12–15 September, Vol. 1. System Theory. Control Theory, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, s. 164–168.

Kwaśnicka H., Kwaśnicki W. (1986), *Diversity and Development: Tempo and Mode of Evolutionary Processes*, Technological Forecasting and Social Change, Vol. 30, No. 3, s. 223–243.

Kwaśnicka H., Kwaśnicki W. (1986a), *Diversity and Development: Tempo and Mode of Evolutionary Processes*, Technological Forecasting and Social Change, Vol. 30, No. 3, s. 223–243.

Kwaśnicka H., Kwaśnicki W. (1987), *Struktura informacji dziedzicznej a optymalność procesów ewolucyjnych*, Kosmos, t. 36, nr 3, s. 315–336.

Kwaśnicka H., Kwaśnicki W. (1989), *Multi-unit firms in an evolutionary model of a single industry development*, w R. Day (ed.) *Evolutionary Dynamics and Nonlinear Economics*, Austin.

Kwaśnicka H., Kwaśnicki W. (1991), *An Evolutionary Outlook of Novelty Formation in Science and Technology*, w Werner Ebeling, Manfred Peschel and Wolfgang Weidlich (eds.), *Models of Selforganization in Complex Systems MOSES*, Berlin, Akademie Verlag, s. 227–235.

Kwaśnicka H., Kwaśnicki W. (1996), *Long-Term Diffusion Factors of Technological Development: An Evolutionary Model and Case Study*, Technological Forecasting and Social Change, Vol. 52, No. 1, s. 31–57.

Kwaśnicka H., Kwaśnicki W. (1996a), *Simulation of Technological Substitution-Diffusion Processes: An Evolutionary Approach*, Modelling and System Simulation. Proceedings of the 30th Spring International Conference MOSIS '96, Krnov, s. 256–261.

Kwaśnicka H., Markowska-Kaczmar U. (1994), *Możliwości zastosowania systemów ekspertowych w ubezpieczeniach*, Wiadomości Ubezpieczeniowe, nr 7–9, s. 20–24.

Kwaśnicka H., Markowska-Kaczmar U. (1994a), *Sztuczna inteligencja w sterowaniu*, Informatyka, nr 6, s. 3–9.

Kwaśnicka H., Markowska-Kaczmar U. (1995), *Zastosowanie algorytmów genetycznych w problemach optymalizacyjnych*, Informatyka, nr 3, s. 1–7.

Kwaśnicka H., Markowska-Kaczmar U. (1996), *Do Semi-Isolated Subpopulations Evolve Quicker? Genetic and Evolutionary Algorithms*, MENDEL 96 2nd International Mendel Conference on Genetic Algorithms, Brno, Technical University of Brno, s. 60–65.

Kwaśnicka H., Markowska-Kaczmar U. (1998), *Chromosome Coding and Evolutionary Algorithms Efficiency*, CAI'96, praca opublikowana w materiałach konferencyjnych *Colloquia in Artificial Intelligence Theory and Applications*, Łódź, s. 78–85, przyjęta do publikacji w: *Folia Informatica*, Uniwersytet Łódzki, nr 1.

Kwaśnicka H., Markowska-Kaczmar U., Nuno M. (1998), *Hybrid Intelligent Systems – Combination of Genetic Algorithms and Fuzzy Logic*, CAI'98 Colloquia in Artificial Intelligence, Theory and Applications, Second Polish Conference on Theory and Applications of Artificial Intelligence, wrzesień 28–30, Łódź, s. 131–153.

Kwaśnicka H., Nowostawski M. (1997), *The Search Engine for Information Systems Based on Parallel Genetic Algorithm*, Proceedings of the 2nd International Conference on Parallel Processing & Applied Mathematics, Zakopane, Częstochowa, PCZ IMI, s. 442–451.

Kwaśnicka H., Szerszon P. (1997), *NetGen – Evolutionary Algorithms in Designing Artificial Neural Networks*, Proceedings of the Third Conference Neural Networks and Their Applications and Summer

School on Neural Networks Applications to Signal Processing, Kule k.Częstochowy, s. 671–676, Częstochowa, Polskie Towarzystwo Sieci Neuronowych.

Kwaśnicki W. (1996), *Knowledge, Innovation, and Economy. An Evolutionary Exploration*, Cheltenham, Brookfield, Edward Elgar Publishing Limited (pierwsze wydanie: Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 1994).

Kwaśnicki W., Kwaśnicka H. (1992), *Market, Innovation, Competition. An Evolutionary Model of Industrial Dynamics*, Journal of Economic Behavior and Organization, Vol. 19, s. 343–68.

Kwaśnicki W., Kwaśnicka H. (1997), *Genetic Operators in an Evolutionary Model of Industrial Dynamics*, MENDEL'97 III International Mendel Conference on Genetic, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets. Technical University of Brno, s. 77–82.

Lieske B. (1995), *Automatyczna optymalizacja sztucznych sieci neuronowych za pomocą algorytmów genetycznych*, praca magisterska, Wydział Informatyki i Zarządzania, Politechnika Wrocławska.

Lastowski K. (1987), *Rozwój teorii ewolucji. Studium metodologiczne*, Poznań, Uniwersytet im. A. Mickiewicza.

Manfred E. (1973), *The Origin of Biological Information*, in *The Physicist's Conception of Nature*, J. Mehra ed., Dordrecht.

Markowska-Kaczmar U., Kwaśnicka H. (1994), *Zastosowanie sieci neuronowych w systemach czasu rzeczywistego*, Informatyka nr 11, s. 3–6, 27.

Mayr E. (1963), *Animals species and evolution*, Cambridge, Harvard University Press.

Mayr E. (1982), *Speciation and macroevolution*, Evolution 36 (6).

Mayr E. (1982a), *Speciation and Macroevolution*, Evolution 36 (6).

Mayr E. (1991), *One Long Argument. Charles Darwin and the Genesis of Modern Evolutionary Thought*, The Penguin Press.

Medsker L.R. (1995), *Hybrid Intelligent Systems*, Boston, Kluwer Academic Publishers.

Michalewicz Z. (1996) *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, Warszawa, WNT.

Miller G.F., Todd P.M., Hedge S.U. (1989), *Designing Neural Networks using Genetic Algorithms*, Proceeding of the third International Conference on Genetic Algorithms, San Mateo, California, Morgan Kaufmann Publishers, Inc.,

Mitchell T.M. (1997), *Machine Learning*, The McGraw-Hill Companies, Inc.

Montana J.D., Davis L. (1989), *Training Feedforward Neural Networks Using Algorithms*, Cambridge, BBN Systems and Technologies Corp.

Mulawka J.J. (1996), *Systemy ekspertowe*, Warszawa, WNT.

Murray D. (1994), *Tuning Neural Networks with Genetic Algorithms*, AI Expert, June.

Nadel L., Stein D. (ed.) (1993), *Lectures in Complex Systems*, Lecture Vol. V, Santa Fe Institute, Addison-Wesley Publishing Company.

Nelson R.R., Winter S.G. (1982), *An Evolutionary Theory of Economic Change*, Cambridge, Harvard University Press.

Neri F., Zini F. (1994), *Learning Disjunctive Concepts with Distributed Genetic Algorithms*, http://www.dai.ed.ac.uk/groups/evalg/eag_local_copies_of_papers.body.html Dipartimento di Informatica, Università di Torino.

NeuroForecaster – State of the Art Business Forecasting System, Information about the system, Futura (1994), Singapore.

Nowicki A. (1992), *Analiza możliwości wykorzystania ewolucyjnego modelu rozwoju przemysłu do modelowania postępu technicznego w symulacyjnej grze menedżerskiej*, praca magisterska, Wydziałowy Zakład Informatyki, Politechnika Wrocławska.

Nuno M. (1997), *Automatic Design of a Fuzzy Controller Using Genetic Algorithms*, A Diploma Work, Departamento de Informatica, Universidade Nova de Lisboa, and Department of Computer Science, Technical University of Wrocław.

O'Reilly U., Goldberg D.B. (1998), *How Fitness Structure Affects Subpopulation Acquisition in Genetic Programming*, dostępny pod adresem: <http://gal4.ge.uiuc.edu>.

Osmera P., Kvasnicka V., Pospichal J. (1997), *Genetic Algorithms with Diploid Chromosomes*, MENDEL '97, Brno.

Ossowski S. (1994), *Sieci neuronowe (Neural Networks)*, Warszawa, Oficyna Wydawnicza Politechniki Warszawskiej.

Petry F.E., Buckles B.P., Kraft D.H. (1993), *Generating fuzzy information retrieval queries via genetic programming*, Proceedings Fifth International Fuzzy Systems Association World Congress (IFSA'93), Seoul, s. 481–484.

Pierwsze kroki z programem BrainMaker for DOS, Arkus Electronics, Wrocław, 1994 (written on the base of "Getting Started with BrainMaker", California Scientific Software).

Porto V.W., Fogel D.B., Fogel L.J. (1995), *Alternative Neural Network Training Methods*, IEEE EXPERT, June, s. 16–21.

Potter M.A., De Jong K.A., Grefenstette J.J. (1995), *A Coevolutionary Approach to Learning Sequential Decision Rules*, Proceedings of the Sixth International Conference (ICGA), Pittsburgh, Pennsylvania, Morgan Kaufmann Pub.

Radcliffe N.J., Surry P.D. (1995), *Co-operation through Hierarchical Competition in Genetic Data Mining*, http://www.dai.ed.ac.uk/groups/evalg/eag_local_copies_of_papers.body.html Edinburg Parallel Computing Centre, Scotland.

Rusón A. (1995), *Analiza skuteczności projektowania – z wykorzystaniem AG – NN dla określonego zadania, (Efficiency of NN Design using GA)*, praca magisterska, Wydział Informatyki i Zarządzania, Politechnika Wrocławska.

Rutkowska D., Piliński M., Rutkowski L. (1996), *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*, Warszawa, Wydawnictwo Naukowe PWN.

Schafer J.D., Whitley D., Eshelman L.J. (1992), *Combinations of Genetic Algorithms and Neural Networks: A Survey of the State* International Workshop on Combinations of Genetic Algorithms and neural Networks, Baltimore, Maryland.

Schalkoff R.J. (1990), *Artificial Intelligence: An Engineering Approach*, McGraw-Hill Publishing Company.

Schwefel H.P. (1989), *Collective Phenomena in Evolutionary Systems*, Abteilung INFORMATIK, Dortmund, Universität Dortmund.

Sen S., Knight L. (1995), *A Genetic Prototype Learner*, in the Proceedings of the International Joint Conference on Artificial Intelligence (s. 725–731), Montreal, (<http://euler.mcs.utulsa.edu/~sandip/prototype.ps>).

Seredynski F. (1998), *New Trends in Parallel and Distributed Evolutionary Computing*, Fundamenta Informaticae, Vol. 35, No. 1–4, August, s. 211–230.

Simpson G.G. (1984), *Tempo and Mode in Evolution*, New York.

- Smith R.E., Goldberg D.E.** (1992), *Diploidy and Dominance in Artificial Genetic Search*, Complex Systems, Vol. 6, s. 251–285.
- Szarski H.** (1986), *Mechanizmy ewolucji*, Warszawa, PWN.
- Szerszon P.** (1997), *Projektowanie sieci neuronowych z wykorzystaniem algorytmów genetycznych*, praca magisterska, Wydział Informatyki i Zarządzania, Politechnika Wrocławska, Wrocław.
- Tadeusiewicz R.** (1992), *Systemy wizyjne robotów przemysłowych*, Warszawa, WNT.
- Tadeusiewicz R.** (1993), *Sieci neuronowe*, wyd. II, Warszawa, Akademicka Oficyna Wydawnicza.
- Tadeusiewicz R.** (1994), *Problemy cybernetyki*, Warszawa, Wydawnictwo Naukowe PWN.
- Tadeusiewicz R.** (1995), *Tryumf czy kapitulacja rozumu?*, Znak, nr 484, wrzesień, s. 59–69.
- Thierens D., Goldberg D.B., Pereira A.G.** (1998), *Domino Convergence, Drift, and the Temporal-Salience Structure of Problems*, dostępny pod adresem: <http://gal4.ge.uiuc.edu>.
- Traczyk W.** (red.) (1995), *Problemy sztucznej inteligencji*, Warszawa, Wiedza i Życie.
- Voigt H-M., Mühlenbein H., Schlierkamp-Voosen D.** (1996), *The Response to Selection Equation for Skew Fitness Distributions*, Proceedings of International Conference on Evolutionary Computation (ICEC '96), Nagoya, s. 820–825.
- Waddington C.H.** (1977), *Stabilization in Systems. Chreods and Epigenetic Landscapes*, Futures, April, s. 139–146.
- Walker R.F., Haasdijk E.W., Gerrets M.C.** (1995), *Credit Evaluation Using a Genetic Algorithm*, w: *Intelligent Systems for Finance and Business*, s. 39–69, Goonatilake S. And Treleaven P. (ed.), Wiley.
- Whitley L.D., Schaffer J.D.** (ed.) (1992), *International Workshop on Combination of Genetic Algorithms and Neural Networks*, IEEE Computer Society Press Technology Series, Los Alamitos, California.
- Winter S.** (1984), *Schumpeterian competition in alternative technological regimes*, J. of Economic Behavior and Organization 5, s. 287–320.
- Yager R.R., Filev D.P.** (1995), *Podstawy modelowania i sterowania rozmytego*, Warszawa, WNT.
- Yager R.R., Zadeh L.A.** (red.) (1994), *Fuzzy Sets, neural Networks, and Soft Computing*, Nowy Jork, van Nostrand Reinhold VNR.
- Yuret D.** (1994), *From Genetic Algorithms To Efficient Optimization* MIT, Artificial Intelligence Laboratory, <http://alpha-bits.ai.mit.edu/people/deniz/html/aitr1569/msthesis.html>.

ZAŁĄCZNIK 1.

SŁOWNIK WAŻNIEJSZYCH TERMINÓW¹

Terminy biologiczne

Adaptacja – (przystosowanie), procesy wytwarzania się dziedzicznych właściwości przystosowawczych anatomicznych i fizjologicznych w obrębie populacji, w zmienionych warunkach środowiska; polega na zmianach puli genetycznej populacji, wywołanych przez działanie selekcji naturalnej.

Allele – para lub więcej genów, z których każdy wywołuje odmienne wykształcenie tej samej cechy; zajmują jednakowe położenie w chromosomach homologicznych; w organizmach diploidalnych wykluczają się w gametach, a występują parami w komórkach somatycznych.

Chromosomy – samoodtwarzające się, stałe składniki jądra komórkowego, będące nośnikami genów. Zbudowane głównie z DNA i białek. Ich liczba w komórce, kształty, wielkość i organizacja są charakterystyczne dla danego gatunku.

Chromosomy homologiczne – chromosomy o jednakowej budowie, w których geny (allele) ułożone są w takiej samej kolejności; w komórkach diploidalnych jeden z pary chromosomów homologicznych pochodzi z gamety męskiej, drugi z gamety żeńskiej.

Crossingover – wymiana odpowiadających sobie odcinków między chromatydami (chromatydą – każda z połówek podzielonego podłużnie chromosomu).

Diploid – organizm, który w jądrach komórek somatycznych (komórki ciała z wyłączeniem komórek płciowych) ma 2 zespoły homologicznych chromosomów (2 genomy); większość organizmów (roślinnych i zwierzęcych) to organizmy diploidalne; zjawisko występowania diploidów – diploidalność.

Dominowanie – panowanie cech, w genetyce, jest przewaga u mieszańca cechy jednego z rodziców nad odpowiadającą jej cechą drugiego; jest skutkiem przewagi oddziaływania jednego z pary genów – alleli – warunkujących tę cechę (gen dominujący). Jest to przeciwieństwo recesywności.

¹ Hasła opracowano głównie na podstawie: *Encyklopedia popularna*, Warszawa, PWN, 1992 oraz *Mały słownik biologiczny*, Warszawa, Wiedza Powszechna, 1961.

Dobór naturalny – używany zamiennie z selekcją naturalną, główny czynnik ewolucji organizmów (Ch. Darwin); pozostawia przy życiu osobniki o cechach korzystnych, eliminuje gorzej przystosowane.

Dryf genetyczny – nieregularne, losowe zmiany częstości występowania genu w małych, izolowanych populacjach. Czynnik ewolucji – eliminując dany gen lub zwiększając jego częstość hamuje lub przyspiesza procesy selekcji.

Epistaza – tłumienie przejawiania się cechy, warunkowane przez odpowiedni gen (hipostatyczny) wskutek działania genu (zwanego epistatycznym lub supresorem) należącego do innej pary alleli.

Ewolucja (organizmów) – zachodzące stopniowo przeobrażenia budowy anatomicznej i cech fizjologicznych gatunków roślin i zwierząt, prowadzące do powstawania nowych gatunków, coraz lepiej przystosowanych do środowiska; zachodzi w skali populacyjnej i w ciągu wielu pokoleń, jest wynikiem zmian częstości genów w populacjach, a kierunek przystosowawczy nadaje tym zmianom dobór naturalny.

Ewolucjonizm (biologicznie) – nauka o ewolucji organizmów; bada jej mechanizmy, przebieg, kierunki, tempo – na podstawie danych paleontologii, genetyki populacyjnej, anatomii porównawczej, biogeografii i in.

Ewolucjonizm syntetyczny – współczesna postać darwinowskiej teorii ewolucji organizmów, tłumacząca jej istotę zmianą częstości genów (cech) w populacjach, wywołaną współdziałaniem mutacji i rekombinacji, selekcji naturalnej i dryftu oraz izolacji.

Fenotyp – zespół cech organizmu (anatomicznych, fizjologicznych, morfologicznych i in.) ukształtowanych pod wpływem czynników dziedzicznych (genów) i warunków środowiska.

Gen – jednostka dziedziczenia związana z przekazywaniem poszczególnych cech organizmu. Geny ułożone są liniowo i mają określone i stałe miejsce w chromosomach jądra komórkowego roślin, zwierząt oraz w genoforach (np. u bakterii).

Genom – podstawowy, haploidalny (pojedynczy) zespół chromosomów jądra komórkowego, zawierający pojedynczy zestaw genów. U organizmów diploidalnych (większość) występują w komórkach somatycznych (tzn. wszystkich komórkach organizmu z wyjątkiem komórek rozrodczych i ich prekursorów) dwa homologiczne genomy, w gametach – jeden.

Genotyp – zespół wszystkich genów organizmu (skład genetyczny), warunkujący jego właściwości dziedziczne.

Hipostaza – genetycznie: nieprzejawianie się cechy uwarunkowanej przez określony gen (hipostatyczny), mimo jego obecności wskutek tłumiącego działania genu (epistatycznego) z innej pary alleli.

Mutacja – dziedziczna zmiana zachodząca w żywym organizmie. Samorzutne mutacje, zwykle genowe są źródłem zmienności dziedzicznej oraz podstawą procesów selekcji i ewolucji.

Nisza ekologiczna (biol.) – pozycja populacji danego gatunku w biocenozie, określona przez właściwości biologiczne i ekologiczne tej populacji oraz jej związki ze środowiskiem biotycznym i abiotycznym ekosystemu.

Populacja – grupa krzyżujących się ze sobą osobników jednego gatunku, zajmująca określony teren i izolowana rozrodczo od innych populacji tego gatunku.

Recesywność – ustępowanie cech, w genetyce, zupełne lub częściowe nie przejawianie się danej cechy w organizmie mimo istnienia genu warunkującego tę cechę (gen recesywny) wskutek przewagi oddziaływania drugiego, odpowiadającego mu allelicznego genu (genu dominującego).

Rekombinacja – występowanie u potomstwa odmiennych kombinacji genów – a w następstwie i innych zespołów, lub nawet nowych cech – niż u form rodzicielskich (przypadkowa, niezależna segregacja chromosomów i procesu crossingover, zachodzących podczas mejozy przy powstawaniu gamet, losowe łączenie się gamet).

Selekcja naturalna – używana zamiennie z doбором naturalnym, termin określający znaczenie odsiewu genotypów gorzej przystosowanych i mniej wydajnych rozrodczo, nadaje kierunek ewolucji.

Inne terminy

Adaptacyjne układy – (automatyka) układy sterowania, których struktura i parametry zależą od zmieniających się warunków pracy; stosowane, gdy warunki pracy układu zmieniają się w sposób przypadkowy (np. pilot automatyczny).

Ewolucja (filoz.) – forma rozwoju, proces ciągłych, stopniowych, drobnych przemian (ilościowych), sukcesywnego (nie skokowego) przechodzenia w wyższe stadia rozwojowe; także synonim rozwoju.

Ewolucjonizm (filoz.) – kierunek w filozofii i naukach społecznych, zwłaszcza w drugiej połowie XIX w., (H. Spencer, Ch. Darwin) wyjaśniający formy istnienia i strukturę rzeczywistości przyrodniczych i społecznych jako rezultat działania praw ewolucji.

Optymalizacja – (optymizacja), wyznaczanie najlepszego, ze względu na wybrane kryterium (np. koszt, zysk, niezawodność) i spełniającego zadane ograniczenia, rozwiązania danego problemu, np. ukształtowania elementów konstrukcji, sterowania procesem technologicznym, rozdziału zasobów, organizacji systemu.

SKOROWIDZ

- alfabet genotypu, 43
- algorytm
 - asynchroniczny rozproszony, 20
 - ewolucyjny, 8
 - genetyczny, 6, 13
 - półsynchroniczny scentralizowany, 20
 - sieciowy, 20
 - synchroniczny scentralizowany, 20
- automatyczne uczenie, 7
- CAM-BRAIN, 159
- Cantú-Paz E., 20
- charakter ewolucji, 61
- chromosom, 6, 28, 35, 52
- Cuvier G., 11
- Darwin Ch., 11
- Dawkins R., 41
- de Garis H., 159
- dem, 20
- drobnoziarniste PGA, 21
- efekt
 - plejotropowy, 9, 42, 44
 - poligeniczny, 42, 44
- ewolucyjny model przemysłu (EMP), 144
- fenotyp, 12, 14
- Fogel D., 8
- Fogel L., 18
- funkcje testowe, 62
 - de Jonga, 66
 - rodzina funkcji eksponencjalnych, 62
- Ga i FLS, 27
- Galar R., 49
- genom, 12
- genotyp, 12, 14
- geny
 - aktywne, 42
 - fenotypowe, 42, 51
 - nadmiarowe, 42, 51
 - nadmiarowe w *K*-modelu, 90
 - redundancyjne, 42
 - ukryte, 42
- gra kierownicza, 152
- gruboziarniste PGA, 20
- Gwiazda T., 6
- Haffman A., 8
- Holland J., 6, 13, 20
- hybrydowe systemy inteligentne, 22
- Kacprzak T., 6
- Kloch J., 5
- K*-model, 41, 51
 - funkcja de Jonga F_1 , 135
 - funkcja de Jonga F_2 , 135
 - funkcja de Jonga F_3 , 136
 - funkcja de Jonga F_4 , 137
 - funkcja de Jonga F_5 , 138
 - funkcja dwóch zmiennych, 131
 - funkcja dziesięciu zmiennych, 133
 - funkcja jednej zmiennej, 129
 - funkcja pięciu zmiennych, 133
 - funkcja testowa F_6 , 140
 - funkcja testowa F_7 , 141
 - funkcja trzydziestu zmiennych, 134
 - opis formalny, 52
 - osobnik, 55
 - populacja, 52
 - program, 57
 - selekcja, 54
- kodowanie, 14
- Korbicz J., 6
- Koza J., 18
- krajobraz przystosowawczy, 12
- kryzys, 57
 - w *K*-modelu, 90
- krzyżowanie, 16
- Kwaśnicki W., 148
- liczebność populacji w *K*-modelu, 96
- logika rozmyta, 25
- Łukasiewicz J., 25
- macierz plejotropowości, 53
 - dobór, 120
 - jednorodna dodatnia, 119

- typ 1, 119
- typ 2, 120
- typ 3, 120
- typ 4, 120
- typy, 119
- makromutacja, 41
- makromutacje, 54, 56
 - w *K*-modelu, 90
- martwe neurony, 39
- Mayr E., 8
- metoda selekcji
 - Poissona, 54
 - próbkowania deterministycznego, 15
 - ruletka, 15
 - stochastyczna wg reszt bez powtórzeń, 16
 - stochastyczna wg reszt z powtórzeniami, 15
 - turniejowa, 16
- metody selekcji, 15
- Michalewicz Z., 6, 17
- miękką selekcja, 49
- migracja, 21
- Mulawka J., 6
- mutacja rutyny, 148
- mutacje, 13, 16, 53, 55
 - neutralne, 56
 - w *K*-modelu, 80
- neuron, 31
- obliczenia ewolucyjne, 8
- Obuchowicz A., 6
- operatory genetyczne, 55
- optymalizacja, 11
- Ossowski S., 6, 32, 34
- plejotropowość, 42
- poligeniczność, 42
- populacja potomna, 14,
- poziom fenotypowy, 14
- poziom genotypowy, 14
- programowanie ewolucyjne, 18
- programowanie genetyczne, 18
- projektowanie systemu rozmytego, 28
- przystosowanie, 12
- pułapka ewolucyjna, 75
- rekombinacja rutyny, 148
- rekombinacje, 53, 55
 - w *K*-modelu, 71
- rekrudescencja, 56
 - w *K*-modelu, 92
- reprezentacja firm w EMP, 147
- reprezentacja osobnika, 14
- reprodukcja, 13, 15, 16
- równoległe algorytmy genetyczne (PGA), 20
- Russel B., 25
- rutyny aktywne, 147
- rutyny nadmiarowe, 147
- Schwefel H., 17
- selekcja, 49
- sieci neuronowe, 6, 31
 - projektowanie, 32
 - system NETGEN, 39
- strategie ewolucyjne, 17
- symulacja rozwoju przemysłu, 145
- system
 - FUSSION*, 23
 - informacyjny, 22
 - NeuroForecaster*, 34
- systemy ekspertowe, 5
- Szarski H., 8
- Sztuczna inteligencja, 5, 8
- Ślot K., 6
- Tadeusiewicz R., 5, 6
- tempo ewolucji, 61
- teoria ewolucji, 11
- test Turinga, 5
- topografia adaptacyjna, 12
- Traczyk W., 5, 6
- transpozycja, 54, 56
- tranzycja, 54, 56
- twarda selekcja, 49
- twierdzenie schematów, 43
- Wallace A., 11
- wartość przystosowawcza, 53
- Wright S., 12
- wyszukiwanie dokumentów, 23
- zachowaj najlepszego, 16
- Zadech L., 25
- zbiory rozmyte, 25





Uczenie się, zdolność adaptacji do nowych warunków to bardzo pożądane właściwości projektowanych przez człowieka systemów. Stworzenie takiego systemu wciąż jest trudnym zadaniem. Podglądanie i próba naśladowania natury może dać pozytywne wyniki. Jednym z naturalnych procesów, powszechnie naśladowanym, jest ewolucja biologiczna.

Poznanie wrażliwości algorytmów ewolucyjnych na stosowane operatory pozwoli na tworzenie skutecznych aplikacji.

Wydawnictwa Politechniki Wrocławskiej
są do nabycia w następujących księgarniach:
„Politechnika”
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław
budynek A-1 PWr., tel. (071) 320-25-34,
„Tech”
plac Grunwaldzki 13, 50-377 Wrocław
budynek D-1 PWr., tel. (071) 320-32-52
Prowadzimy sprzedaż wysyłkową

ISBN 83-7085-421-4

