WARSZTATY ALGORYTMÓW GENETYCZNYCH

Wdzydze, 4 – 5 Września 2003-08-21

Materiały konferencyjne

pod redakcją Romana Śmierzchalskiego

GDYNIA 2003

Organizatorzy warsztatów:

Roman Śmierzchalski, Andrzej Łebkowski, Krzysztof Dziedzicki

Katedra Automatyki Okrętowej Akademii Morskiej w Gdyni

Opracowanie i przygotowanie do druku:

Krzysztof Dziedzicki, Roman Śmierzchalski

ISBN

Gdynia 2003

Wydawnictwo Akademii Morskiej w Gdyni

81-225 Gdynia

ul. Morska 83

SPIS TREŚCI

Słowo wstępne 5

Jarosław Arabas, Robert Jakubowski, Rafał Józwiak
Odporność algorytmu ewolucyjnego a liczność populacji bazowej 7

Aleksander Byrski
Immunologiczne systemy ewolucyjne 11

Artur Chorążyczewski
Losowy wybór a zwartość populacji. 13

Ireneusz Czarnowski, Piotr Jędrzejowicz
Zastosowanie algorytmu uczenia populacji do selekcji cech w problemie uczenia maszynowego 15

Rafał Dreżewski
Classical and Agent-Based Co-Evolutionary Techniques 17

Krzysztof Dziedzicki, Roman Śmierzchalski
Koncepcja algorytmu ewolucyjnego do harmonogramowania produkcji elektrociepłowni 19

Przemysław Dziemieszkiewicz
Porównanie selekcji konkurencyjnej i turniejowej zastosowanych w modelowaniu ewolucji populacji w środowisku mozaikowym. 21

Roman Galar
Proceduralizacja z perspektywy algorytmów ewolucyjnych 23

Iwona Karcz-Dulęba
Analiza zjawiska skupiania się populacji w ewolucji fenotypowej 25

Joanna Kołodziej
Modeling the structure of genetic strategies with stochastic L_Systems 27

Zdzisław Kowalczuk, Tomasz Białaszewski
Zastosowanie rodzajnika genetycznego jako kryterialnego niszowania 29

Bartłomiej Jacek Kubica
Metody optymalizacji globalnej w zastosowaniu do systemów kolejkowych 31

Halina Kwaśnicka
Systemy hybrydowe – łączenie algorytmów ewolucyjnych i sieci neuronowych 33

Andrzej Łebkowski, Roman Śmierzchalski
Wykorzystanie technik komputerowych do kierowania statkiem 35

Przemysław Miazga
Algorytm ewolucyjny do optymalizacji zadań z kosztowną funkcją celu 37

Tomasz Milewski, Maciej Tryka, Piotr Kaszyński, Paweł D. Domański
Predykcyjne sterowanie ewolucyjne 39

Piotr Nycz, Jarosław Stańczak, Krzysztof Sęp
Ewolucyjne uczenie logicznej sieci rozmytej z redukcją argumentów danych trenujących 47

Tomasz Samotyjak
Synteza parametrów modeli laboratoriów zwarciowych z wykorzystaniem algorytmów genetycznych 51

T. Wójcik, K. Wala
Algorytm ewolucyjny harmonogramowania procesów obliczeniowych
w wieloprocesorowych systemach czasu rzeczywistego 53

SŁOWO WSTĘPNE

Mamy zaszczyt przywitać Państwa na trzecich już Warsztatach Algorytmów Genetycznych zorganizowanych tym razem we Wdzydzach. Warsztaty te są kontynuacją idei nieformalnych spotkań osób zainteresowanych szeroko pojęta tematyką związaną metodami sztucznej inteligencji zapoczątkowanej przez Profesora Romana Galara.

Mamy nadzieję, iż klimat spotkań będzie sprzyjać wymianie poglądów i podjęciu twórczej dyskusji dotyczącej zagadnień związanych z tą tematyką w podobnym stopniu jak na Pierwszych i Drugich Warsztatach Algorytmów Genetycznych zorganizowanych przez Profesora Romana Galara oraz Profesora Roberta Schaefera i ich zespoły.

Jako przyczynek do dyskusji będącej podstawową formą aktywności naukowej podczas Warsztatów, mają posłużyć prezentowane w niniejszych „Materiałach" komunikaty. Mają one szczególny charakter, ich celem, jak i samych Warsztatów jest nie tylko zwięzłe przedstawienie osiągniętych już rezultatów ale i prezentacja zagadnień nierozwiązanych, penetrujących słabo poznane obszary wiedzy. Sądzimy, że podczas Warsztatów mogą wyłonić się inspirujące pomysły dotyczące ewolucji adaptacyjnej i optymalizacji oraz ich możliwości aplikacyjnych. Liczymy na to, iż atmosfera Warsztatów wpłynie na zacieśnienie współpracy oraz skłoni do twórczego wykorzystania możliwości jakie niosą ze sobą algorytmy genetyczne.

Z ogromną przyjemnością oddaję w Państwa ręce zeszyt będący zbiorem komunikatów prezentowanych na trzecich Warsztatach Algorytmów Genetycznych.

Roman Śmierzchalski

Wdzydze 2003

ODPORNOŚĆ ALGORYTMU EWOLUCYJNEGO A LICZNOŚĆ POPULACJI BAZOWEJ

JAROSŁAW ARABAS, ROBERT JAKUBOWSKI, RAFAŁ JÓZWIAK

Wydział Elektroniki i Technik Informacyjnych Politechnika Warszawska
{j.arabas, rdjakubowski, rjozwiak}@elka.pw.edu.pl

STRESZCZENIE

Niniejszy komunikat przedstawia wyniki eksperymentów z algorytmem ewolucyjnym zastosowanym do prostego zadania optymalizacji funkcji dwóch zmiennych. Badana jest sprawność znajdowania otoczenia maksimum globalnego w funkcji liczności populacji bazowej przy dwóch różnych sposobach inicjacji populacji bazowej – z równomiernie losowym rozkładem w przestrzeni przeszukiwań oraz z rozkładem skoncentrowanym w podobszarze znacznie oddalonym od maksimum globalnego. Celem eksperymentu jest ilustracja związku między licznością populacji bazowej a zdolnością algorytmu do opuszczania pułapek ewolucyjnych oraz zdolnością do eksploracji przestrzeni przeszukiwań.

1. WPROWADZENIE

W środowisku badaczy algorytmów ewolucyjnych nie ma zgody co do zalecanej liczby osobników przetwarzanych przez algorytm w populacji bazowej. Dość powszechne wydaje się być mniemanie, że im więcej tych osobników, tym wyższa sprawność algorytmu w dochodzeniu do maksimum globalnego przystosowania [1,2]. Taki sposób rozumowania przyjmuje jako aksjomat, że podstawowym źródłem odporności algorytmu na ekstrema lokalne jest liczebność populacji, co pozwala już od pierwszej populacji bazowej uchwycić otoczenie maksimum lokalnego. Zwróćmy jednak uwagę na tkwiącą w tej argumentacji sprzeczność: jeśli uchwyci się już otoczenia wszystkich minimów lokalnych (w tym globalnego), to wystarczy przeprowadzić lokalną optymalizację startując z każdego z tych punktów, a następnie wybrać najlepszy wynik jako rozwiązanie. Na domiar złego, doświadczenie pokazuje, że algorytm ewolucyjny nie jest sprawną metodą optymalizacji lokalnej.

Jeśli założyć, że wymiarowość przestrzeni oraz rozmiary obszarów przyciągania minimów lokalnych prowadzą do niewielkich prawdopodobieństw „uchwycenia" punktu w otoczeniu minimum lokalnego od losowego strzału, wówczas konieczne staje się przyjęcie założenia, że głównym zadaniem algorytmu ewolucyjnego jest umiejętne wychodzenie z obszarów przyciągania minimów lokalnych – opuszczanie pułapki ewolucyjnej. Dla tego zadania z kolei panuje przekonanie, że mniejsze populacje bazowe są bardziej skuteczne [3,4].

Zadaniem tego komunikatu jest ilustracja sytuacji praktycznej, z którą ma do czynienia użytkownik algorytmu ewolucyjnego traktujący go jako narzędzie optymalizacji, a nie przedmiot badań naukowych. Użytkownik taki umie najczęściej generowań punkty z rozkładem jednostajnym w przestrzeni rozwiązań, ale nie zna położenia minimum globalnego. Pytanie, które sobie stawia, dotyczy wielkości populacji bazowej, którą ma przetwarzać. Zakłada się także, że obliczenie wartości funkcji przystosowania jest nieporównanie kosztowniejsze niż jakiekolwiek operacje właściwe algorytmowi ewolucyjnemu, dlatego też kryterium zatrzymania opiera się na liczbie obliczeń wartości przystosowania, a nie na liczbie przetworzonych pokoleń algorytmu ewolucyjnego.

2. EKSPERYMENT

Eksperyment przeprowadzono przy populacjach bazowych o różnych licznościach (1, 2, 5, 10, 20, 50, 100, 200 osobników). Były one inicjowane losowo z rozkładem jednostajnym w całej przestrzeni rozwiązań oraz poza obszarem przyciągania minimum globalnego (znanego w przypadku przyjętych funkcji testowych). Wśród wykorzystanych metod reprodukcji znalazły się: reprodukcja proporcjonalna zmodyfikowana, progowa (ρ=0.2) oraz reprodukcja proporcjonalna

standardowa z dodaniem wartości stałą do funkcji przystosowania (wartość stałej była możliwie mała, lecz wystarczająca, by zapewnić nieujemne wartości funkcji przystosowania). Eksperymenty przeprowadzono dla dwóch sukcesji prostej i elitarnej (liczność elity: 1 osobnik). Jedynym stosowanym operatorem genetycznym była mutacja gaussowska ($\sigma = 0.1$).

Eksperymenty przeprowadzono z oprogramowaniem gabi [5]. Działanie algorytmu było przerywane po obliczeniu 10 tys. wywołań funkcji przystosowania. Do oceny jakości algorytmu brano pod uwagę wartość przystosowania najlepszego osobnika uśrednioną ze 100 niezależnych uruchomień algorytmu.

Jako funkcję testową wykorzystano funkcję Shuberta

$$f\left(x^*\right) = \sum_{i=1}^{m} i\cos[(i+1)x_1 + i]\sum_{i=1}^{m} i\cos[(i+1)x_2 + i]$$

Przyjęto m=5, $-10 \le x_1$, $x_2 \le 10$. Znane jest 18 minimów globalnych i 760 lokalnych tej funkcji, a wartość w minimum wynosi -186,73 (taką też wartość wykorzystano do korekty przystosowania w metodzie reprodukcji proporcjonalnej standardowej).

2.1. Populacja inicjowana rozkładem jednostajnym

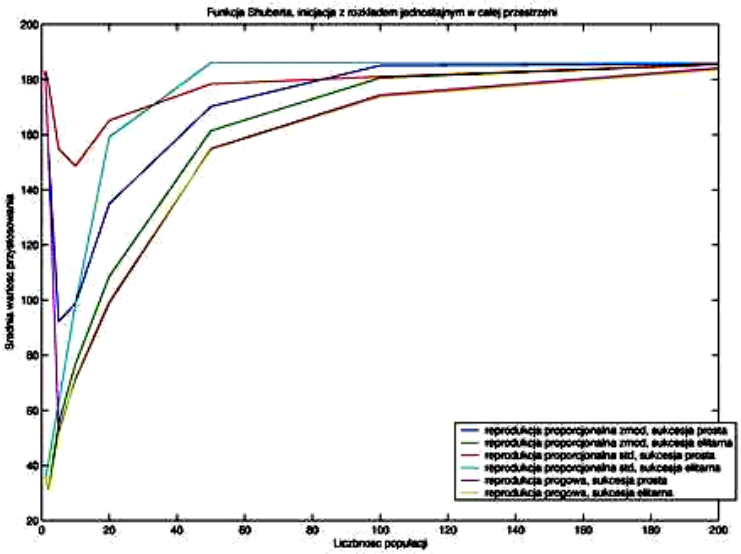
Przykładowe wyniki działania algorytmu dla populacji bazowej inicjowanej rozkładem jednostajnych w przestrzeni rozwiązań przedstawia Rys.1. Na uwagę zasługuje fakt, że zależność jakości uzyskiwanych rozwiązań dla nieelitarnej sukcesji jest niemonotoniczną funkcją liczby osobników w populacji bazowej – oprócz ogólnie narastającej tendencji wzrostowej (mającej charakter nasycający się), obserwuje się także wyraźne maksimum jakości algorytmu dla zaledwie kilku osobników w populacji bazowej.

Kształt otrzymanych zależności można tłumaczyć jako złożenie dwóch zjawisk – „chwytania" osobników w otoczeniach minimum lokalnego (tym bardziej prawdopodobne, im jest ich więcej) oraz umiejętności opuszczania pułapki ewolucyjnej (paradoksalnie – im więcej osobników, tym mniejsza skłonność populacji do wyjścia z pułapki).

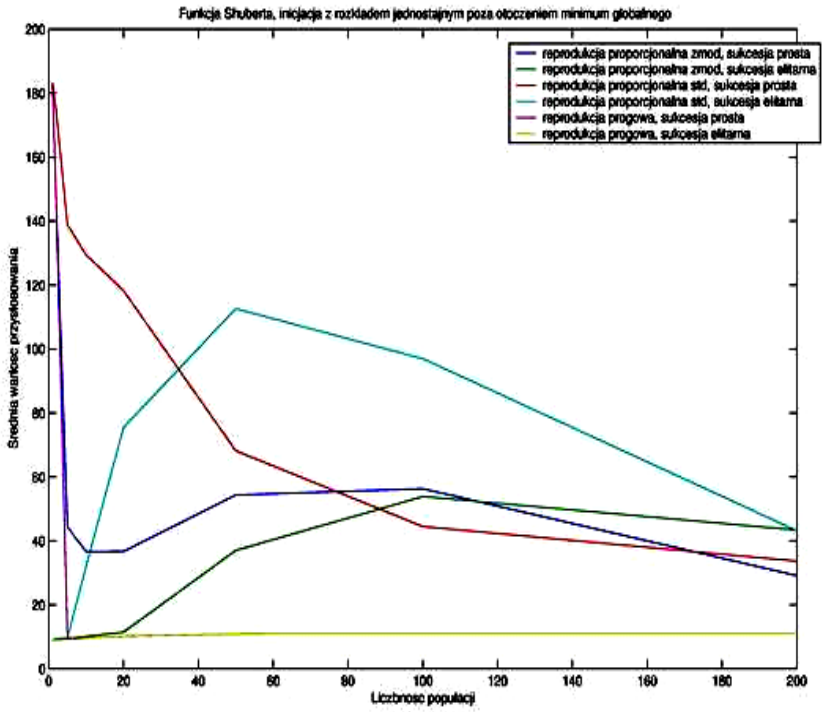
2.2. Populacja inicjowana daleko od minimum globalnego

Weryfikacji powyższych hipotez można dokonać inicjując populację bazową tak, aby prawdopodobieństwo wygenerowania osobnika w otoczeniu minimum globalnego było zerowe. Wówczas będzie można uchwycić zdolność opuszczania pułapek ewolucyjnych oraz eksploracji w zależności od liczności populacji. Wynik eksperymentu przedstawia Rys.2.

Jak łatwo zauważyć, widoczna jest wyraźna tendencja graniczna do pogarszania jakości algorytmu wraz ze wzrostem liczności populacji. Optymalne liczności populacji silnie zależą od przyjętego schematu sukcesji – przy sukcesji prostej, jest to kilka (2-5) osobników. Sukcesja elitarna z kolei albo nie działa (w połączeniu z reprodukcją progową), albo maksimum sprawności przypada dla kilkudziesięciu osobników.



Rys. 1 Sprawność algorytmu dla populacji bazowej rozłożonej równomiernie



Rys. 2 Sprawność algorytmu dla populacji bazowej inicjowanej poza otoczeniem minimum globalnego

3. WNIOSKI

Eksperyment ilustruje zależność jakości algorytmu od sposobu sukcesji liczby osobników w populacji bazowej. Okazuje się, że w przypadku selekcji nieelitarnej, zarówno duża (kilkadziesiąt lub więcej) jak i mała liczność populacji bazowej pozwala na równie sprawne działanie pod warunkiem jednak, że populacja bazowa jest rozsiana równomiernie w przestrzeni przeszukiwań. Dobra sprawność algorytmu z małą populacją wynika ze zdolności opuszczania pułapek ewolucyjnych. Selekcja elitarna ma tę zdolność mocno upośledzoną.

Kwestią otwartą pozostaje, jak zależności powyższe będą się utrzymywać przy zwiększaniu liczby wymiarów zadania, Można przewidywać, że kilkadziesiąt osobników może być niewystarczające przy modelu „dużej populacji", gdyż możliwość trafienia w otoczenie maksimum globalnego gwałtownie maleje wraz ze wzrostem liczby wymiarów. Podobnie przy zmniejszaniu zasięgu mutacji. Trudno natomiast przewidzieć bez dodatkowych badań, jak będą się zachowywać algorytmy z selekcją nieelitarną z małą populacją przy wzroście liczby wymiarów – czy kilka osobników będzie nadal optymalne, czy raczej należałoby zwiększyć tę liczbę, czy też wreszcie efekt dobrej jakości algorytmu z małą liczbą osobników zaniknie. Wymienione powyżej problemy będą przedmiotem dalszych badań.

LITERATURA

1. D. Goldberg, *Genetic Algorithms in Search*, Optimization and Machine Learning, Addison Wesley, 1989
2. J. Koza, *Genetic Programming*, 1992
3. R.Galar, *Miękka selekcja w losowej adaptacji gobalnej w R^n*, Politechnika Wrocławska, 1990
4. I. Karcz-Dulęba, „*Dynamika ewolucji dwuelementowych populacji w przestrzeni stanów populacji – przypadek symetrycznych funkcji jakości*", KAEiOG, Lądek, 2000
5. Oprogramowanie gabi: http://elektron.elka.pw.edu.pl/~jarabas/

IMMUNOLOGICZNE SYSTEMY EWOLUCYJNE

ALEKSANDER BYRSKI

Akademia Górniczo-Hutnicza,
Kraków olekb@agh.edu.pl

Ludzkie ciało jest ciągle atakowane przez antygeny (obce proteiny), system odpornościowy tworzy antyciała (limfocyty), które mają za zadanie odróżnienie komórek własnych organizmu od komórek obcych, z którymi powinny się związać (wykorzystując zjawisko fagocytozy) i doprowadzić do ich usunięcia z organizmu. Antyciała wykształcane są w grasicy oraz w szpiku kostnym. Grasica wytwarzając limfocyty kieruje się tzw. algorytmem selekcji negatywnej. Niedojrzałe limfocyty są umieszczane wśród komórek własnego organizmu, jeśli limfocyt zwiąże się z własną komórką, zostaje on usunięty jako niewłaściwy. Tylko takie limfocyty które nie reagują na komórki własne organizmu zostają przekazane dalej do organizmu, w celu wykrywania antygenów próbujących go zainfekować [2].

Sztuczne systemy odpornościowe oparte są na zasadzie działania biologicznego systemu immunologicznego. Występują w nich odpowiedniki komórek odpornościowych - osobniki o zróżnicowanych funkcjach, których głównym zadaniem jest eliminowanie zagrożeń systemu (nieodpowiednich, nieprawidłowo działających osobników populacji), kierując się algorytmem selekcji negatywnej, oraz różnicując osobników populacji należących do przestrzeni rozwiązań problemu na „swoich" i „obcych". Swoi są ignorowani, natomiast obcy są usuwani z systemu [3].

Działanie całego systemu opiera się na wprowadzeniu do populacji dwóch typów komórek - B oraz T. Są one odpowiednikami rzeczywistych limfocytów [2].

- Komórki typu B działają w przestrzeni fenotypów, obserwują one efekty działania sąsiadujących osobników populacji. Jeśli wyniki działania danego osobnika nie są wystarczająco dobre - komórka B wiąże się z danym osobnikiem i usuwa go z systemu, ewentualnie doprowadza do wykształcenia na podstawie jego genotypu komórek typu T, gdyż możliwe jest, że nieprawidłowy fenotyp rozwinął się z nieprawidłowego genotypu.
- Komórki typu T działają w przestrzeni genotypów. Niedojrzałe komórki typu T (powstałe z antygenu związanego przez komórkę typu B) są poddawane algorytmowi selekcji negatywnej. Jeśli zwiążą się z komórkami działającymi prawidłowo („swoimi") zostaną usunięte z systemu, w przeciwnym wypadku osiągają dojrzałość. Komórki T stanowią swego rodzaju ograniczenie ewolucji, gdyż mają za zadanie reagować na pojawienie się w systemie osobników o podobnym do nich genotypie. Bez oczekiwania na wykształcenie się fenotypu takie osobniki zostaj ą usunie te z systemu.

Techniki oparte na immunologii mogą wprowadzić nową jakość do badań nad systemami ewolucyjnymi, doprowadzając do automatycznego wykształcenia się ograniczeń algorytmu ewolucyjnego. Dzięki temu teoretycznie można zwiększyć dynamikę populacji, przez wprowadzanie większej liczby osobników w trakcie reprodukcji, w nadziei że nieprawidłowe osobniki zostaną usunięte jeszcze przed wykształceniem fenotypu. Wydaje się, że takie rozwiązanie dobrze pasuje do założeń działania i struktury ewolucyjnych systemów wieloagentowych [1].

BIBLIOGRAFIA

1. Aleksander Byrski, Marek Kisiel-Dorohinicki. *Agent-Based Evolution of Neural Network Architecture.* IASTED Press 2002.
2. Willis H. Johnson, Louis E. DeLanney, Thomas A. Cole. *Podstawy biologii.* Państwowe Wydawnictwo Rolnicze i Leśne, 1975.
3. Jamie Twycross. *An Immune System Approach to Document Classification,* HPL-2002-288. Information Infrastructure Laboratory HP Laboratories Bristol, 2002.

LOSOWY WYBÓR A ZWARTOŚĆ POPULACJI.

ARTUR CHORĄŻYCZEWSKI

Politechnika Wrocławska Instytut Cybernetyki Technicznej
Wyb. Wyspiańskiego 27, 50-370 Wrocław, achorazy@cyber.ict.pwr.wroc.pl

Rezultaty eksperymentów symulacyjnych ([1]) wskazują, że przez większość czasu populacja utrzymuje zwartą formę[1]. Stany, w których tak nie jest należą do kategorii przejściowych (np. początek procesu ewolucyjnego, stan po przejściu siodła dużej populacji) i są krótkotrwałe. Utrzymanie populacji w kształcie, który nie stanowi formy zwartej wymaga specjalnych mechanizmów dynamicznie stabilizujących populację ([2]).

Większość przytoczonych obserwacji dotyczyła algorytmu z miękką selekcją ([3]). Można postawić pytanie, czy postulowana własność zwartości związana jest wyłącznie ze schematem selekcji proporcjonalnej? Autor niniejszej pracy skłania się do nie tylko do tezy, e tak nie jest, ale co więcej, że własność ta jest wspólna dla wszystkich schematów losowego wyboru.

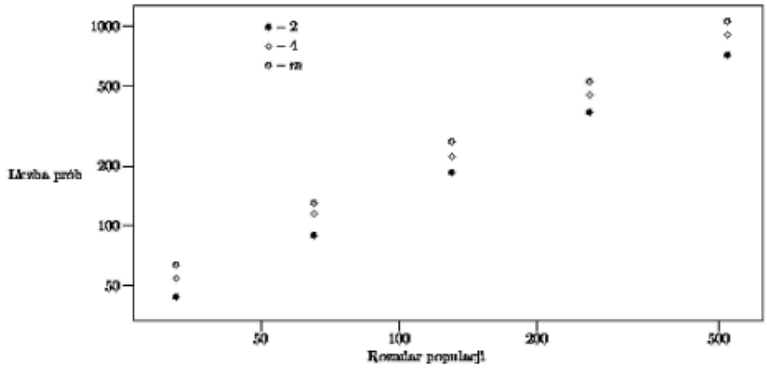
Rozpatrzmy prosty przypadek. Mamy urnę z *m* jednakowymi kulami (odpowiednikami osobników), z jednakowym prawdopodobieństwem wyboru. Odpowiada to sytuacji, gdy mamy rozproszoną populację, w której nie występuje presja selekcyjna (losowy wybór rodziców, ale wszystkie osobniki mają jednakowe szansę na wydanie potomka) i wyłączona jest mutacja. Dzielimy kule w urnie na równoliczne grupy, każdą grupę znacząc innym kolorem, a następnie rozpoczynamy proces reprodukcji. Proces reprodukcji w tym przypadku polega na wrzuceniu do nowej urny kuli w kolorze kuli wylosowanej. Cykl losowania powtarza się, aż do zapełnienia nowej urny *m* kulami, następnie nowa urna staje się urną z kulami "rodzicielskimi". To co nas interesuje czas po jakim osiągniemy stan jednorodności, czyli stan, w którym pozostanie tylko jedna grupa osobników[2] . Na rys. l przedstawiono średnią liczbę cykli potrzebną do uzyskania jednorodności. Jak można zauważyć, wpływ rozmiaru populacji na czas zaniku różnorodności jest liniowy, co zgadza się z rezultatami obliczeń analitycznych [5]. Podział populacji na większą liczbę podgrup, aż do granicznego przypadku, gdy każdy osobnik w populacji jest traktowany jako oddzielna podpopulacja, prowadzi do zwiększenia średniego czasu do ujednolicenia populacji. Jednak nawet w skrajnym przypadku różnica pomiędzy dwoma, a *m* grupami jest raczej ilościowa niż jakościowa.

Sytuacja zmienia się istotnie dopiero, gdy wprowadzone zostanie rozróżnienie w szansach wyboru dla poszczególnych grup. Rys. 2 przedstawia uśrednione wyniki z symulacji dla tego przypadku. Można zauważyć, że nawet niewielka zmiana preferencji selekcyjnych powoduje *znaczne* przyspieszenie ujednolicania się populacji. Ciekawe jest również to, że począwszy od kilkuprocentowej różnicy prawdopodobieństw wyboru, rozmiar populacji nie wpływa na czas ujednolicania, który spada do kilku, kilkunastu cykli.

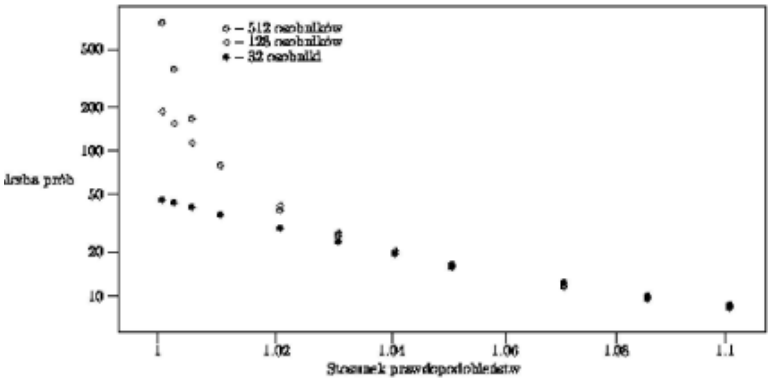
---

[1] Przez zwartość należy rozumieć fakt, że osobniki w populacji znajdują się w przestrzeni fenotypów średnio nie dalej niż kilka odchyleń standardowych mutacji

[2] Tak zdefiniowany proces można przybliżyć zaczerpniętym z genetyki populacyjnej ([4]) modelem zaniku allelu w populacji - grupy odpowiadają w tym przypadku dwóm różnym formom allelu



Rys. 1Wpływ liczby grup w populacji osobników o jednakowych prawdopodobieństwach przeżycia na średni czas dochodzenia do jednorodności



Rys. 2 Wpływ zwiększonego prawdopodobieństwa wyboru jednej grupy na średni czas osiągania jednorodności

Podsumowując, ujednolicanie populacji wydaje się być procesem nieuniknionym, właściwym losowemu wyborowi i postępującym tym szybciej, im populacja jest mniejsza i im wyraźniejsze różnice występują w preferencjach selekcyjnych. Jeśli teraz zauważymy, że mutacja bez selekcji pozwala z biegiem czasu na dowolne zróżnicowanie populacji, to należy oczekiwać, że połączenie obu mechanizmów wytworzy nowy stan równowagi dynamicznej. Równowagi pomiędzy nieuchronnym procesem zanikania różnorodności w efekcie działania losowej selekcji, a procesem rozpraszania populacji w wyniku mutacji. Rezultaty eksperymentalne wskazują, że równowaga ta następuje, gdy rozproszenie osobników wynosi kilka odchyleń standardowych mutacji.

BIBLIOGRAFIA

1. A. Chorążyczewski, R. Galar, *Evolutionary Dynamics in Space of States*, Proc. of CEC 2001, ss. 1366-1373, Seul, Korea, 2001
2. R. Kopciuch, R. Galar, *Zniecierpliwienie i polaryzacja w procesach ewolucyjnych*, materiały III KAEiOG, ss. 115-122, Potok Złoty, 1999
3. R. Galar, *Miękka selekcja w losowej adaptacji globalnej w* R^n, Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej, seria Monografie nr 84, Wrocław, 1990
4. M. Kimura, *Diffusion Models in Population Genetics*, J. Appl. Prob., vol.l , ss. 177-232, 1964
5. D. Thierens, D. Goldberg, *Convergence Models of Genetic Algorithm Selection Schemes*, Proc. Of 3rd Conf. Parallel Problem Sohingfrom

ZASTOSOWANIE ALGORYTMU UCZENIA POPULACJI DO SELEKCJI CECH W PROBLEMIE UCZENIA MASZYNOWEGO

IRENEUSZ CZARNOWSKI, PIOTR JĘDRZEJOWICZ

Katedra Systemów Informacyjnych, Akademia Morska w Gdyni
ul. Morska 83, 81-225 Gdynia, irek, pj@am.gdynia.pl

Algorytmy selekcji cech znajdują zastosowanie w zagadnieniach uczenia maszynowego. W przypadku uczenia pod nadzorem zbiór wektorów wejściowych składa się z cech, które mogą nie wnosić istotnych informacji do procesu uczenia. Pozostawienie wszystkich cech szczególnie nieistotnych oraz generujących błędy nie gwarantuje zadawalającej dokładność predykcji oraz zwiększa wymagania, co do zasobów obliczeniowych.

Wybór jednych, a odrzucenie innych cech jest problemem trudnym, chociaż kryteria są dość oczywiste. W zbiorze uczącym powinny pozostać cechy silnie skorelowane ze zmienną wyjściową, ale jednocześnie słabo skorelowane pomiędzy sobą. Własności te są szczególnie istotne przy budowie modeli prognostycznych. [4]

Znane metody selekcji cech można podzielić na metody zupełne, heurystyczne i losowe, które do oceny zredukowanego zbioru cech wykorzystują funkcje oceny bazujące na informacji odległościowej, współczynniku korelacji oraz błędzie klasyfikacji. [1,3]

Proponowany algorytm uczenia populacji (PLA) [2] bazuje na ewolucji populacji. Podstawowe cechy algorytmu PLA to: populacja startowa jest dużym zbiorem dopuszczalnych rozwiązań (tzw. osobników) wygenerowanych przy wykorzystaniu wybranego mechanizmu losowego, proces uczenia populacji osobników przebiega etapami, w kolejnych etapach używa się coraz bardziej złożonych metod uczenia (poprawy), do kolejnych etapów uczenia przechodzą osobniki spełniające kryteria selekcji. W ten sposób liczebność populacji stopniowo zmniejsza się, najlepsze rozwiązanie na etapie finalnym traktowane jest jako rozwiązanie problemu.

W przypadku selekcji cech wektora uczącego algorytm PLA dokonuje wyboru cech istotnych, dla których współczynnik korelacji wielorakiej R jest maksymalny. Czyli użycie proponowanej metody ma na celu odnalezienie jak najsilniejszego związku pomiędzy zmienną zależną a rozpatrywanymi zmiennymi objaśniającymi.

Zatem problem selekcji cech jest problemem maksymalizacji funkcji celu:

$$R(Y,X_1...X_n)\rightarrow\max$$

gdzie Y jest wartością zmiennej wyjściową (klasą wektora lub wartością zmiennej zależnej) i $X_1 .X_n$ są wartościami wybranych cech.

W zaprojektowanym algorytmie rozwiązanie zostało zakodowane jako ciąg binarny. Wystąpienie jedynki na dowolnej pozycji oznaczało, iż odpowiednia cecha powinna być reprezentowana w zbiorze uczącym. W przypadku wystąpienia zera, cechę należało wyeliminować ze zbioru.

Do pozostałych założeń zaprojektowanego algorytmu PLA należą:
- populacja startowa generowana losowo,
- trzy etapy uczenia,
- trzy metody uczenia (mutacja, krzyżowanie oraz przeszukiwanie z ruchami zabronionymi - ang. *tabu search*),
- wspólne kryterium selekcji (do kolejnego etapu przechodzą rozwiązania, których wartość funkcji celu jest mniejsza lub równa od średniej jej wartości w populacji).

Proponowana metoda została wykorzystana do selekcji cech wektora uczącego w problemie prognozowania jakości produkcji szkła. Problem ten polegał na znalezieniu zależności między 29

wartościami wejściowymi i 5 wyjściowymi w horyzoncie czasowym dwóch tygodni z krokiem piętnastominutowym (1344 wektorów). Zbiór treningowy składał się z wektorów uczących opisujących czternastotygodniowy proces produkcji (8064 wektorów).

Do opisu powyższej zależności wykorzystano model oparty na strukturze dwuwarstwowej sieci neuronowej. W warstwie środkowej umieszczone zostały neurony o nieliniowych właściwościach przetwarzania a w przypadku warstwy wyjściowej neurony z liniową funkcją transferu.

Na wejście sieci zostały podane cechy wektora treningowego oraz dwie dodatkowe informacje tj. średnia krocząca oraz średnia arytmetyczna liczona po wartościach cech oryginalnego wektora treningowego. Średnią krocząca zdefiniowano jako:

$$S(t,k)=(Y(t-l)+Y(t-2)+...+Y(t-K))/k,$$

gdzie t jest krokiem prognozy, k jest liczbą kroków, dla których jest liczona średnia a Y reprezentuje wartość zmiennej zależnej rzeczywistej lub wynikającej z prognozy.

Zaprojektowany algorytm opisujący zależność dla pełnego zestawu cech wektora uczącego generował błąd prognozy na poziomie 40%. W wyniku selekcji cech istotnych silnie skorelowanych ze zmienną wyjściową i niezależnych od siebie poziom błędu został zredukowany do 10%.

Selekcji cech przeprowadzono z wykorzystaniem takich technik jak: forward sequential selection, backward sequential selection oraz algorytmu PLA wykorzystującego błąd prognozy jako funkcję oceny. Uzyskane tymi metodami zbiory o zredukowanym wymiarze nie przyczyniły się do tak znaczącej redukcji błędu prognozy.

Po wprowadzeniu do procesu selekcji dodatkowej informacji o cechach silnie skorelowanych ze zmienna objaśnianą, którą otrzymano w wyniku zastosowania proponowanej metody maksymalizacji współczynnika korelacji wielorakiej, poziom błędu prognozy uległ znacznemu zmniejszeniu.

LITERATURA

1. Dash, M., Liu, H., *Feature Selection for Classification*. Intelligent Data Analysis l, 131-156, 1997
2. Jędrzejowicz, P., *Social Learning Algorithm as a Tool for Solving Some Difficult Scheduling Problems*. Foundation of Computing and Decision Sciences, 24, 51-66, 1999
3. Roman, B., loerger, T. R., *Enhancing Learning Using Features and Example Selection*. Journal of Machine Learning Research (2003) (przyjęty do publikacji w 2003)
4. Hyvarinen, A., *Survey on Independent Component Analysis*, Neural Copmuting Surveys 2,94-128,1999

CLASSICAL AND AGENT-BASED CO-EVOLUTIONARY TECHNIQUES

RAFAŁ DREŻEWSKI

Department of Computer Science AGH University of Science and Technology,
Krakow, Poland drezew@ agh. edu. pi

Evolutionary algorithms (EAs) have demonstrated in practice efficiency and robustness as global optimization techniques. However, they often suffer from premature loss of population diversity which results in premature convergence and may lead to locating local optima instead of a global one. What is more, both the experiments and analysis show that for multi-modal problem landscapes a simple EA will inevitably locate a single solution. The loss of diversity also limits the adaptive capacities of EAs in dynamic environments. Co-evolutionary techniques are aimed at improving adaptive capacities and introducing openended evolution into EAs [4].

In classical EAs each individual in the population is considered to be a potential solution of the problem being solved. The fitness of each individual depends only on how well it solves the problem. Selection pressure causes that better fit individuals have the greater chance to survive and/or reproduce and less fit ones have the smaller chance.

In co-evolutionary systems the fitness of each individual depends not only on the quality of solution to the given problem but also on other individuals' fitness. As the result of ongoing research many co-evolutionary techniques have been proposed. Generally, each of these techniques belongs to one of two classes: "Competitive Fitness Functions" (CFF) or multipopulation [4]. Also some of the niching techniques may be considered as co-evolutionary

In CFF based systems two (or more) individuals compete in a game and their "Competitive Fitness Functions" are calculated based on their relative performance in that game [2], Each time step given individual competes with different opponents, so its fitness value varies.

The second group consists of systems that use multiple populations. In such systems a problem is decomposed into sub-problems and each of them is then solved by different EA [5]. Each individual is evaluated within a group of randomly chosen individuals coming from different sub-populations. Its fitness value depends on how well the group solved the problem and on how well the individual assisted in the solution.

Some of the niching techniques may also be considered as being co-evolutionary since fitness of each individual depends on other individuals in a population. In *co-evolutionary shared niching (CSN)* technique [3] (inspired by the economic model of *monopolistic competition)* two co-evolving populations are used. The customer population is the usual population of candidate solutions. The businessman population evolve to obtain the largest possible payoff (cover the peaks in multi-modal domain).

The main idea of *evolutionary multi-agent system (EMAS)* is the modeling of evolution process in multi-agent system (MAS) [1]. *Co-evolutionary multi-agent system (CoEMAS)* allows co-evolution of several species of agents. CoEMAS can be applied, for example, to multi-objective optimization and multi-modal function optimization *(niching co-evolutionary* multi-agent system - NCoEMAS).

In CoEMAS several (usually two) different species co-evolve. One of them represents solutions. The goal of the second species is to cooperate (or compete) with the first one in order to force the population of solutions to locate Pareto frontier or proportionally populate and stably maintain niches in multi-modal domain.

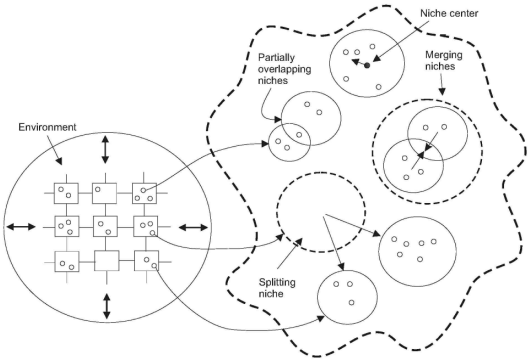


Figure 1: Sample niching co-evolutionary multi-agent system

In figure 1 a sample system for multi-modal optimization with two co-evolving species: niches and solutions is presented. In such NCoEMAS we can model niches as individuals that are characterized by parameters like location, radius, etc. and evolve to best cover real niches in multi-modal domain. Two additional operators can be introduced for niches: splitting and merging. Each niche can make decision on splitting into two niches based on the current distribution of its subpopulation Also the decision of merging can be made by two niches that are close enough and that are located on the same peak in the multi-modal domain.

It seems that CoEMAS is especially suited for modeling different co-evolutionary interactions (resource competition, predator-prey and host-parasite co-evolution, sexual preferences, etc.)

REFERENCES

1. K. Cetnarowicz, M. Kisiel-Dorohinicki, and E. Nawarecki: *The application of evolution process in multi-agent world to the prediction system*. In Proc. of the 2nd Int. Conf. on Multi-Agent Systems — ICMAS'96, Osaka, Japan, 1996. AAAI Press.
2. P. J. Darwen and X. Yao: *On evolving robust strategies for iterated prisoner's dilemma*. Lecture Notes in Computer Science, 956, 1995.
3. D. E. Goldberg and L. Wang: *Adaptive niching via coevolutionary sharing.* Technical Report IlliGAL 97007, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA, 1997.
4. J. Morrison and F. Oppacher: *A general model of co-evolution for genetic algorithms*. In Int. Conf. on Artificial Neural Networks and Genetic Algorithms ICANNGA 99, 1999.
5. M. A. Potter and K. A. De Jong. *Cooperative coevolution: An architecture for evolving coadapted subcomponents*. Evolutionary Computation, 8(1), 2000.

KONCEPCJA ALGORYTMU EWOLUCYJNEGO DO HARMONOGRAMOWANIA PRODUKCJI ELEKTROCIEPŁOWNI

KRZYSZTOF DZIEDZICKI† , ROMAN ŚMIERZCHALSKI‡

†Politechnika Gdańska, Katedra Elektroenergetyki, dziedzic@ely.pg.gda.pl
‡Akademia Morska w Gdyni, Katedra Automatyki Okrętowej, roms@am.gdynia.pl

WPROWADZENIE

Opracowanie harmonogramu produkcji rozumiane jest jako wyznaczenie ilości wytwarzanej w elektrociepłowni mocy elektrycznej $P_{wy}(t)$ i ciepła $Q_{wy}(t)$ zapewniającej maksimum zysku. Otrzymany w wyniku prowadzonych obliczeń harmonogram powinien zawierać plan produkcji dla poszczególnych jednostek wytwórczych. Należy uwzględnić dynamikę samej elektrociepłowni jak i miejskiej sieci ciepłowniczej, z którą współpracuje modelowana elektrociepłownia. Rozruch poszczególnych bloków wiąże się z dodatkowymi kosztami. Koszt rozruchu jest tym większy im „chłodniejsza" była uruchamiana jednostka produkcyjna. Ciepło wytwarzane w elektrociepłowni oddawane jest do miejskiej sieci ciepłowniczej. Temperatura wody zasilającej limitowana jest umową ze spółką dystrybucyjną. Tym samym dopuszczalna temperatura wody na wejściu do sieci ciepłowniczej ogranicza obszar pracy elektrociepłowni. Temperatura wody powracającej z systemu grzewczego zmienia poziom możliwej do wyprodukowania energii elektrycznej. Tworzy się tym samym sprzężenie zwrotne. Możliwe jest pewne funkcjonalne uniezależnienie od stanu miejskiej sieci ciepłowniczej przez zastosowanie dociążenia pseudokondensacyjnego lub zbiornika ciepła. Miejska sieć ciepłownicza sama w sobie również jest obiektem dynamicznym. Sygnał temperaturowy z wejścia pojawia się na wyjściu w postaci temperatury wody powrotnej i przepływu z pewnym opóźnieniem. Wykorzystany model szczegółowo opisano w [3]. Nieliniowa, i całkowitoliczbowa charakterystyka modelu skłania do zastosowania algorytmów ewolucyjnych.

KONCEPCJA SPECJALOZOWANEGO ALGORYTMU EWOLUCYJNEGO

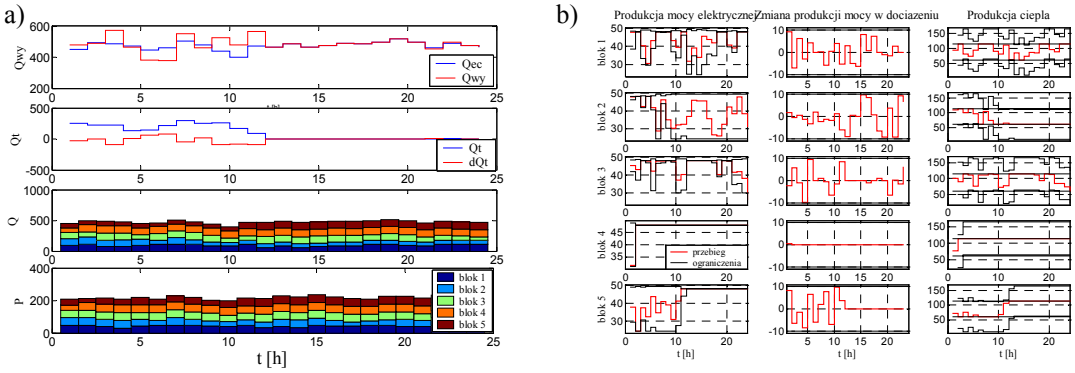
Wykorzystywany model elektrociepłowni zapisano w środowisku MATLAB®. Przyjęta struktura chromosomu zawiera wektory binarne kodujące stan pracy jednostek wytwórczych (***X***) i rzeczywiste reprezentujące poziomy produkcji (***P***, ***Q***) oraz przepływy w opcjonalnym zbiorniku ciepła. Logiczną strukturę chromosomu przedstawiono na poniższym rysunku.

X_1 P_1 Q_1	X_2 P_2 Q_2	...	X_n P_n Q_n	dQ_t

Rys. 1 Logiczna struktura chromosomu

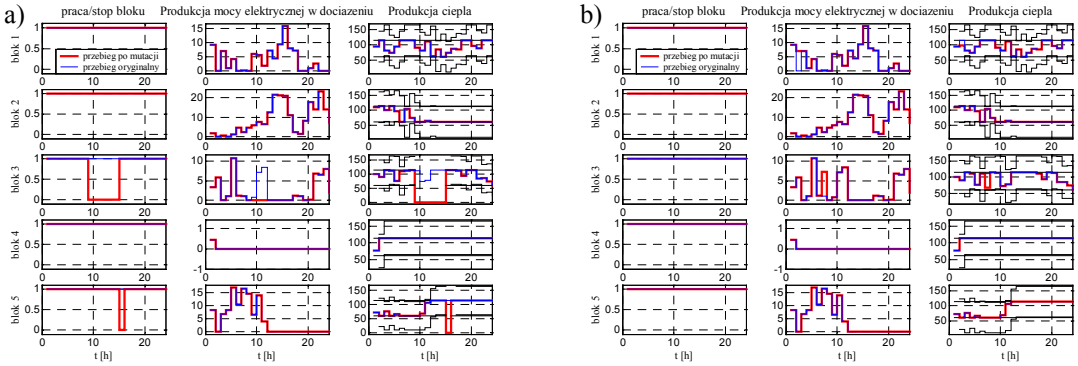
Sformułowany model zawiera dużą liczbę ograniczeń. Dodatkowo ograniczenia na dopuszczalny poziom produkcji energii elektrycznej zależą od poziomu produkcji ciepła. W związku z powyższym w utworzonej za pomocą "czystego" losowania populacji znajduje się niewiele osobników spełniających wszystkie ograniczenia. Konieczne jest przygotowanie generatora populacji początkowej. Przykładowy, wylosowany za jego pomocą harmonogram umieszczono na rysunku 2a. Rysunek 2b przedstawia ograniczenia nałożone na poszczególne zmienne opisujące osobnika z rysunku 2a. Podczas losowania populacji nie są sprawdzane ograniczenia wynikające ze współpracy z miejską siecią ciepłowniczą.

Zastosowanie standardowych operatorów genetycznych wydaje się niecelowe z tych samych powodów, dla których konieczny jest specjalizowany generator populacji początkowej. Rozważa się zastosowanie szeregu operatorów specjalizowanych dostosowanych do specyfiki postawionego problemu Krzyżowanie można zrealizować jako wymianę całych (lub fragmentów) harmonogramów pracy poszczególnych jednostek wytwórczych i zbiornika ciepła. Przewidywane są cztery operatory mutacji zmieniające poszczególne parametry opracowywanego harmonogramu.



Rys. 2 Przykładowy harmonogram produkcji a) przepływy ciepła w zbiorniku oraz produkcja ciepła i mocy elektrycznej z podziałem na poszczególne bloki; b) ograniczenia technologiczne i wynikające z dynamiki elektrociepłowni

Pierwszy, zmienia stan bloku praca/zatrzymanie. Przełączenie odbywa się przez pewną ilość kolejnych kroków z pewnym prawdopodobieństwem zakończenia (rysunek 3a). W przypadku włączenia jednostki wytwórczej poziom produkcji jest losowany. Drugi operator realizuje losową zmianę poziomu produkcji ciepła a co za tym idzie i mocy elektrycznej (rysunek 3b). Następny zmieniający poziom produkcji mocy w dociążeniu pseudokondensacyjnym. Ostatni z proponowanych operatorów powodowałby zmianę przepływów w zbiorniku ciepła. Procedury wyznaczające punkt cięcia wymienianych podciągów harmonogramów produkcji podczas krzyżowania należy skonstruować tak, aby działanie operatora nie powodowało łamania ograniczeń pracy elektrociepłowni. Takie same wymagania stawiane są procedurom losującym w proponowanych operatorach mutacji.



Rys. 3 Wyniki działania proponowanych operatorów mutacji: a) zmiana stanu praca/stop bloku; b) zmiana poziomu produkcji.

Zapewnienie spełnienia ograniczeń nałożonych na stan miejskiej sieci ciepłowniczej na poziomie operatorów genetycznych wydaje się niecelowe ze względu na duży związany z tym koszt. W związku z tym do formuły funkcji celu wprowadzono wykładniczą funkcję kary zależną od “odległości” od ograniczenia.

LITERATURA

1. Arabas J.: *Wykłady z algorytmów ewolucyjnych*. Warszawa, WNT 2001.
2. Michalewicz Z.: *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. Warszawa WNT 1996.
3. Dziedzicki K., Śmierzchalski R., Augusiak A.: *A model of combined heat and power generating plant with urban heat distribution network for production scheduling;* 9th International Conference MMAR; Międzyzdroje 2003.

PORÓWNANIE SELEKCJI KONKURENCYJNEJ I TURNIEJOWEJ ZASTOSOWANYCH W MODELOWANIU EWOLUCJI POPULACJI W ŚRODOWISKU MOZAIKOWYM.

PRZEMYSŁAW DZIEMIESZKIEWICZ

Instytut Cybernetyki Technicznej Politechniki Wrocławskiej
Wyb. Wyspiańskiego 27, 50-370 Wrocław; pdziemieszkiewicz@lukas.com.pl

WPROWADZENIE

Przedmiotem naszych zainteresowań są symulacyjne badania populacji ewoluujących w środowisku mozaikowym. Przez środowisko mozaikowe rozumiane jest środowisko składające się z kilku rodzajów miejsc (nisz ewolucyjnych). Jakość osobnika w takim środowisku zależy nie tylko od jego fenotypu, ale również od rodzaju zajmowanego miejsca.

W poprzednich pracach [np. 2] zaproponowano prosty model ewolucji w środowisku mozaikowym. Interesowano się dynamiką symulowanego procesu. Szczególnie analizowano warunki, w których populacja może osiągnąć i stabilnie utrzymać bimodalny rozkład fenotypów. Badano kilka schematów selekcyjnych i stwierdzono, że metoda nazwana selekcją konkurencyjną w pewnych warunkach doprowadza do specjacji[1], czyli spontanicznego rozdzielenia populacji na dwie fenotypowo rozłączne podpopulacje. Porównanie kilku metod pozwoliło lepiej zrozumieć mechanizmy, które determinują tę zdolność. Okazuje się, że wiele zależy od tego, czy osobniki pretendujące do miejsca w środowisku są oceniane funkcją przystosowania związaną z danym miejscem. Chodzi o to, że w środowisku mozaikowym mamy co najmniej dwie nisze ewolucyjne i co najmniej dwie, związane z nimi funkcje przystosowania. Może się zdarzyć, że potomek trafi do innej niszy, niż ta z której pochodzi jego rodzic. W efekcie potomek dziedziczy cechy rodzica, ale nie jego jakość. Jeśli selekcja jest oparta na ocenie rodziców wtedy do następnego pokolenia trafiają osobniki niezbyt dobrze przystosowane do zastanych warunków. Inaczej jest, gdy pretendentów do danego miejsca oceniamy funkcją jakości związaną z danym miejscem. Dzieje się tak w selekcji konkurencyjnej, gdzie o każde miejsce konkuruje kilka niedojrzałych osobników ocenianych wg kryteriów związanych z danym miejscem. Przy takim schemacie selekcji wybór osobnika do następnego pokolenia bezpośrednio zależy od jego jakości w danym miejscu.

Metoda konkurencyjna przypomina selekcję turniejową, z tym, że w selekcji konkurencyjnej zwycięzcę wybiera się za pomocą metody ruletkowej, a w selekcji turniejowej zwycięża osobnik o najwyższym wskaźniku przystosowania – obie metody zostaną dokładniej opisane w kolejnym rozdziale.

We wcześniejszych badaniach nie badano selekcji turniejowej. Ponieważ oba sposoby selekcji bazują na wyborze zwycięzcy z grupy pretendentów postanowiono sprawdzić, czy symulowane z ich użyciem procesy dają podobne wyniki. W szczególności, czy zastosowanie selekcji turniejowej również prowadzi do bimodalnych rozkładów populacji.

ALGORYTM SYMULACJI

Rozważana jest populacja m osobników zajmujących m miejsc w środowisku. Środowisko składa się z dwóch (l=1,2) równolicznych (m/2) rodzajów miejsc. Obie nisze ewolucyjne związane są z różnymi funkcjami przystosowania q_l. Osobniki charakteryzowane są przez wektor cech rzeczywistych $\mathbf{x_e}=(x_{e1}, ..., x_{en})$ oraz przez jakość — fitness $q_l(\mathbf{x_e})$. Reprodukcja jest aseksualna. Symulacje przebiegały w następujących krokach:

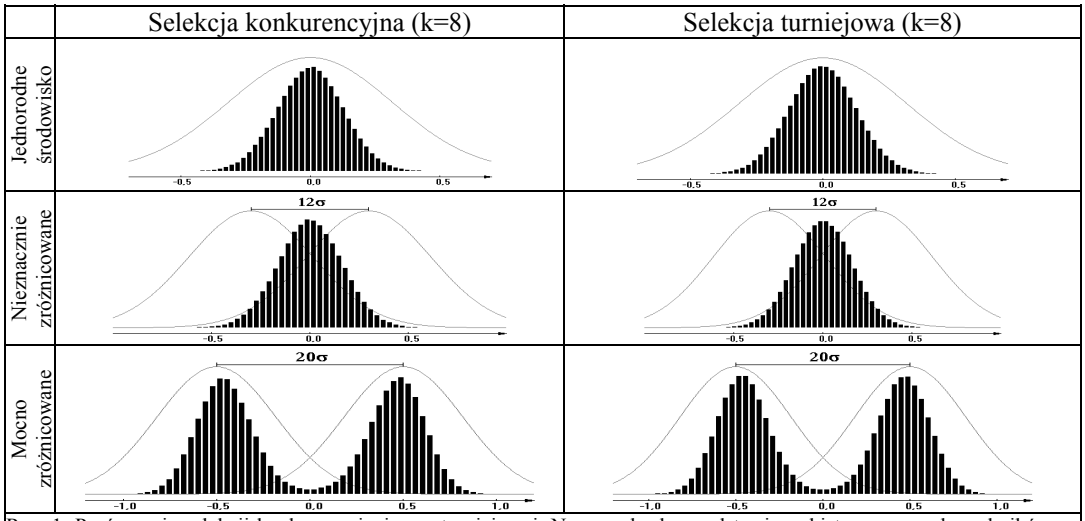
1. *Utworzenie początkowej populacji* — ustalona zostaje początkowa generacja $G(0)$ składająca się z m jednakowych osobników (o takim samym wektorze cech).
2. *Losowanie konkurentów* — wśród osobników generacji $G(i)$ losuje się elementy bazowe

[1] W biologii *specjacja* oznacza proces powstawania nowego gatunku.

(rodzicielskie) elementów generacji $G(i+1)$ w liczbie k na każde z m miejsc środowiska. Losowania są niezależne, a szanse rodziców na wylosowanie są równe i wynoszą $1/m$.

3. *Modyfikacja* — cechy elementów potomnych wyznacza się, poprzez dodanie do wartości każdej z cech rodzica przyrostu losowego o rozkładzie normalnym $N(0,\sigma)$.
4. *Ocena potomków* — wylicza się fitness wszystkich elementów potomnych, przy czym jakość osobnika e zależy od jego typu i od rodzaju miejsca l w środowisku, o które osobnik konkuruje $q(e)=q_l(\mathbf{x_e})$. Jeśli stosowana jest selekcja konkurencyjna to przejście do punktu 5, a jeśli turniejowa to przejście do punktu 6.
5. *Selekcja konkurencyjna* — Spośród k konkurujących osobników losowany jest jeden, który zajmie dane miejsce w środowisku. Losowania są niezależne, a prawdopodobieństwo wylosowania elementu e jest równe $q(e)/\Sigma q(e_j)$ $(j=1...k)$ — tzw. selekcja proporcjonalna. Przejście do punktu 7.
6. *Selekcja turniejowa* — Spośród k konkurujących osobników wybierany jest jeden osobnik o największym $q(e)$, który zajmie dane miejsce w środowisku — tzw. twarda selekcja.
7. Jeśli nie jest spełniony warunek końca symulacji to przejście do punktu 2 i rozpoczęcie procesu od nowa.

WYNIKI I PODSUMOWANIE



Rys. 1. Porównanie selekcji konkurencyjnej oraz turniejowej. Na rysunkach przedstawiono histogramy cech osobników na tle przekrojów krajobrazów adaptacyjnych. Krajobrazy adaptacyjne konstruowano używając n-wymiarowych funkcji Gaussa. W lewej kolumnie zaprezentowano wyniki uzyskane, przy zastosowaniu selekcji konkurencyjnej, a w prawej kolumnie selekcji turniejowej. Widać, że w przytoczonych pomiarach obie metody dały praktycznie identyczne wyniki. Szczególnie interesujący jest ostatni wiersz. Wynik ten świadczy o tym, że podobnie jak, przy selekcji konkurencyjnej zastosowanie selekcji turniejowej pozwala stabilnie utrzymać bimodalny rozkład fenotypów populacji. Parametry symulacji: m=64; n=4; σ=0,05; liczba generacji= 10 000.

Na rys. 1. pokazano, że zastosowanie selekcji turniejowej daje bardzo zbliżone wyniki do tych uzyskanych przy użyciu selekcji konkurencyjnej. W obu metodach o każde miejsce w środowisku konkuruje kilka osobników ocenianych wg funkcji przystosowania związanej z danym miejscem. Pozwala to potwierdzić, że cecha ta ma istotny wpływ na proces ewolucyjny w środowisku mozaikowym i na rozkłady fenotypów populacji.

LITERATURA

1. P. Dziemieszkiewicz, R. Galar: *Symulacja procesów dywersyfikacji populacji ewoluującej w niejednorodnym środowisku*. X Konf. — Sym. Procesów Dynamicz, 1998 Zakopane.
2. R. Galar: *Miękka selekcja w losowej adaptacji globalnej w R^n*. 1990, Wydawnictwo Politechniki Wrocławskiej.

PROCEDURALIZACJA Z PERSPEKTYWY ALGORYTMÓW EWOLUCYJNYCH

ROMAN GALAR
Instytut Cybernetyki Technicznej, Politechnika Wrocławska
Wyb. Wyspiańskiego 27, 50-370 Wrocław, rbg@ict.pwr.wro5c.pl

Analiza działania algorytmów ewolucyjnych pomaga zrozumieć, w jaki sposób miękko kontrolowana spontaniczność indywidualna prowadzi do dynamicznego porządku grupowego, który nie tylko wykorzystuje istniejące możliwości dostosowawcze, ale też sprawnie poszukuje nowych. Tego typu wiedza wydaje się oferować interesującą perspektywę poznawczą na możliwości i ograniczenia proceduralizacji, która w znacznym stopniu zdominowała organizację współczesnych społeczeństw.

Atrakcyjność proceduralizacji jest oczywista. Radzenie sobie z problemami w miarę jak się pojawiają i najlepiej jak się potrafi ma szereg wad. Szczególnie irytujące (osobliwie w sferze publicznej) jest to, że różni różnie sobie dają radę. Stąd, zwłaszcza dla mniej krytycznych umysłów, rozwiązanie proceduralne wydaje się tak pociągające. Istnieje pokusa, aby wyeliminować niejednoznaczność rozstrzygnięć, z góry określając, jakie decyzje, w jakiej sytuacji muszą być podjęte. Próbuje się zredukować nieskończoną rozmaitość wszystkich realnie możliwych sytuacji problemowych przez ich odwzorowanie w pewną liczbę z góry ustalonych sytuacji standardowych. Do zaprojektowania odpowiedniej procedury klasyfikacyjno-decyzyjnej należy oczywiście zatrudnić najlepszych fachowców. Zakłada się, że, gdy raz zrobią swoje, reszta będzie już prosta.

Zgodnie z proceduralnym ideałem każdy pojawiający się problem jest klasyfikowany i załatwiany na zasadzie odruchu Pawłowa. Wykonanie procedury aplikuje „zapuszkowaną" wiedzę jej twórcy, sam zaś decydent znajduje się przy tym w stanie, który można by określić jako *bypass* mózgu (trudno, taki przepis!). Jeśli coś idzie źle zapewne należy wprowadzić doskonalsze procedury. Takie są powszechne przekonania, w praktyce występują trudności:

- Jakość procedur instytucjonalnych jest uporczywie mizerna (wystarczy wczytać się w formularz podatkowy by odczuć skalę intelektualnej konfuzji jego twórców). Uwikłane w procesy decyzyjne operatory są rozpaczliwie proste Znane wszystkim ze szkoły pierwiastki i logarytmy nie przebiły się dotąd do sfery procedur. Nawet wiodące instytucje naukowe rozporządzają swymi finansami za pomocą przedziałów i kategorii, a nie eleganckich funkcji celu.
- Myślenie proceduralne ma skłonność do ignorowania czynnika czasu i kosztów własnych. Szacuje się, że zakupienie zwykłego komputera zgodnie z procedurami przetargowymi rządu USA trwa rok i kosztuje o 80% więcej niż w sklepie [1].
- Rzeczywistość rzadko daje się dzielić na rozłączne części. Coraz to pojawia się potrzeba bardziej szczegółowych wyróżnień i odpowiednich modyfikacji proceduralnych. Pomaga to załatwiać bieżące kłopoty, ale w konsekwencji mnoży trudności*.
- Proceduralizacja pozwala w założeniu podejmować poprawne decyzje nawet osobom o umiarkowanych kwalifikacjach. W takich warunkach kwalifikacje te ulegają jednak dalszemu obniżeniu.

Istnieją też przyczyny zasadnicze, dla których najdoskonalej nawet zaprojektowane i egzekwowane procedury nie rozwiążą pewnych ważnych problemów, a to z tej racji, że nie da się ich dobrze

---

* *Precise rules, most people believe, "close off loopholes". It happens to be the other way around. Loopholes exists because of precise rules. The constitution, a short document of general principles, has no loopholes. The tax law, all 36 000 pages of it, is practically nothing but loopholes.* (P. H. Howard, op.cit. p.43)

rozwiązać w obrębie proceduralnego paradygmatu działania. W pewnych kontekstach procedury urzędowe — z samej swojej — istoty wymuszać będą praktyki sprzeczne z rozeznaniem współczesnej wiedzy. Oto trzy przykłady ze sfery gospodarczej:

1. Jak wiadomo sprawna gospodarka opiera się na wolnej grze sił rynkowych. Aprioryczna przewidywalność decyzji proceduralnych okazuje się w tych warunkach upośledzeniem. W grze interesów strona, która z góry deklaruje swe ruchy zwiększa szanse przegranej.
2. Rynek jest systemem adaptacyjnym, który powinien ewoluować w stronę lepszego zaspokojenia potrzeb konsumenckich. Determinizm procedur instytucjonalnych ogranicza jednak potencjał optymalizacyjny rynku — silne metody optymalizacji opierają się na poszukiwaniach losowych i zmiękczonej selekcji.
3. Wszelkie organizacje są pod ustawiczną presją agentów, starających się wykorzystać zasoby organizacji do własnych celów. Proces ten przypomina interakcję pasożytów i złożonych organizmów. Ponieważ pasożyty znacznie szybciej się modyfikują, ostateczną linią obrony złożonych organizmów jest ich zróżnicowanie — zmutowane pasożyty mogą przełamać obronę immunologiczną niektórych, ale nie wszystkich. Uniwersalność proceduralna niszczy różnorodność i naraża cały system na te same niebezpieczeństwa.

Innym ważnym obszarem, w którym proceduralizacja okazuje się zagrożeniem dla sensu podejmowanych działań jest dziedzina badań naukowych i innowacyjności. Świadomość tego zdaje się przenikać ustalenia rozwijającej się współcześnie teorii zarządzania wiedzą [2]. Proceduralizacja w tych dziedzinach prowadzi bowiem do praktycznego zablokowania możliwości pojawiania się hipotez i wynalazków przełomowych [3].

Podejście proceduralne musiało się pojawić, gdy społeczności ludzkie przekroczyły antropologiczną granicę kilku tysięcy osób, poza którą władza sprawowana przez osobiście rozpoznawane osobiste autorytety staje się niemożliwa. W tym czasie techniki informacyjne stały na bardzo jeszcze niskim poziomie, procedury były więc z konieczności proste i dotyczyły spraw najważniejszych. Rozwój technik informacyjnych w naszych czasach zaowocował gwałtownym uszczegółowieniem i ekspansją procedur — w rezultacie wady podejścia proceduralnego ujawniają się dobitniej niż kiedykolwiek.

Jednocześnie jednak pojawiły się możliwości, aby inteligencję wmontowaną w procedury instytucjonalne oceniać z perspektywy sztuki programowania komputerów. Choć sztuczna inteligencja wypada wciąż żenująco w zestawieniu z inteligencją zwykłą, to wyraźnie już przekracza poziom sproceduralizowanej inteligencji instytucjonalnej. Istnieje więc możliwość, że pewne niedoskonałości proceduralne zostaną skorygowane w miarę, jak główne obciążenie w tej dziedzinie przerzucane będzie na komputery. Rozeznanie wypływające ze zrozumienia dynamiki ewolucyjnych procesów adaptacyjnych (rola miękkiej selekcji, małych populacji, niszowania itd.) może się przy tym okazać bardzo użyteczne.

LITERATURA

1. P. H. Howard, The Death of Common sense – How Law is Suffocating America, Random House, New York, 1994
2. *Knowledge Management in the Learning Society. Educations and Skills*, Centre for Educational Research and Innovation, OECD 2000
3. Galar R. *Gospodarka oparta na wiedzy i innowacje przełomowe*, W: A. Kukliński (ed.) Gospodarka oparta na wiedzy; wyzwanie dla Polski XXI wieku, KBN, Oficyna Wydawnicza Rewasz, Warszawa 2001

ANALIZA ZJAWISKA SKUPIANIA SIĘ POPULACJI W EWOLUCJI FENOTYPOWEJ

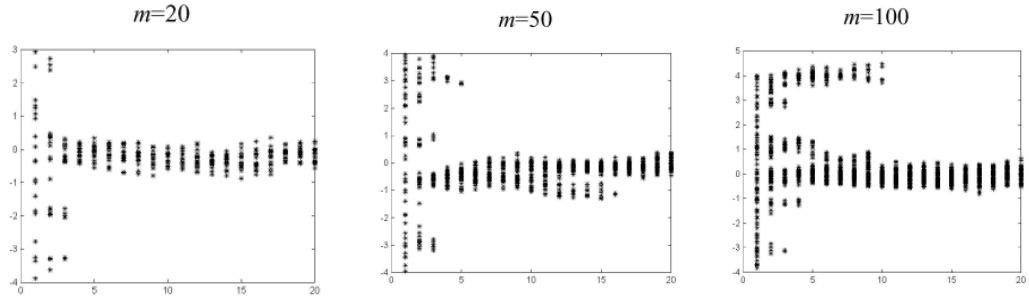
IWONA KARCZ-DULĘBA

Instytut Cybernetyki Technicznej Politechnika Wrocławska
Wyb. Wyspiańskiego 27, 50-370 Wrocław, kdiwona@ict.pwr.wroc.pl

Zjawisko przedwczesnej zbieżności niepokoi środowisko badaczy zajmujących się algorytmami ewolucyjnymi [1]. Utrata różnorodności w populacji powoduje bowiem dużą podatność na zatrzymanie poszukiwań w pułapce ewolucyjnej oraz zmniejszenie się zdolności penetracyjnych algorytmów. Zjawisku zapobiega się różnymi metodami np. stosując algorytmy hybrydowe, steruj c czasem życia osobników, ograniczając zasięg selekcji. Arabas [1] rozważał także wpływ inicjacji populacji bazowej na utrzymanie różnorodności populacji. Jednym z wniosków analizy cytowanego autora było spostrzeżenie, że w celu zapewnienia algorytmowi jak najlepszych warunków początkowych populacja początkowa powinna być jak najbardziej zróżnicowana. Czy jednak to wystarczy? Nasze doświadczenia [2-4] sugerują, że powodem przedwczesnej zbieżności mogą być zjawiska należące do istoty procesów ewolucji, czyli tendencja do skupiania się początkowo rozproszonej populacji. W badaniach symulacyjnych [2,3] obserwowano szybkie skupianie się populacji i jej dalszą wędrówkę w krajobrazie funkcji jakości, już jako zwartej „chmurki" osobników. Dokonane obserwacje potwierdza *analiza* teoretyczna ewolucji małych populacji w przestrzeni stanów populacji [4].

W niniejszej pracy przedstawiono wyniki analizy szybkości skupiania się początkowo rozproszonych populacji. Badano ewolucję fenotypów w-elementowych populacji w krajobrazie jednowymiarowej, unimodalnej gaussowskiej funkcji jakości z optimum w zerze. Populacje ewoluowały zgodnie z zasadami selekcji proporcjonalnej oraz mutacji z rozkładem normalnym o odchyleniu standardowym *o*.

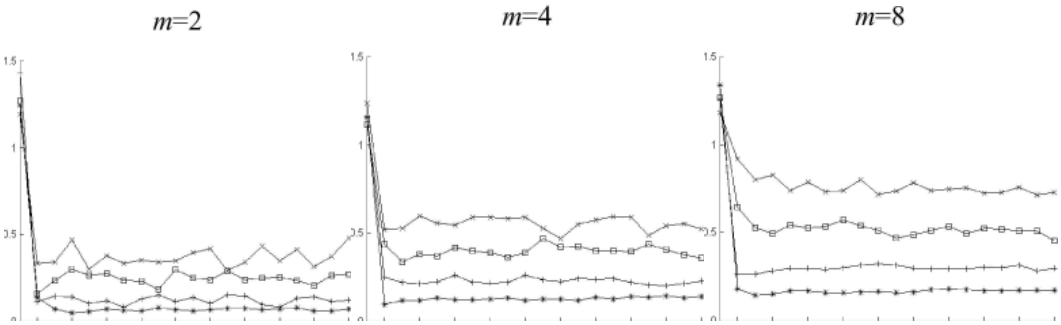
Przeprowadzono serię eksperymentów, badając rozproszenie różnolicznych populacji (rys.l-3). Na rys.l. zobrazowano położenie osobników dużych populacji (w=20,50,100) w kolejnych 20 generacjach. Populacje, których osobniki są początkowo rozłożone losowo w przedziale [-4,4] tracą różnorodność już po kilku generacjach, tym szybciej im populacja jest mniejsza. Osobniki skupiają się w grupę elementów o podobnych cechach, lokalizujących okolice optimum funkcji jakości.



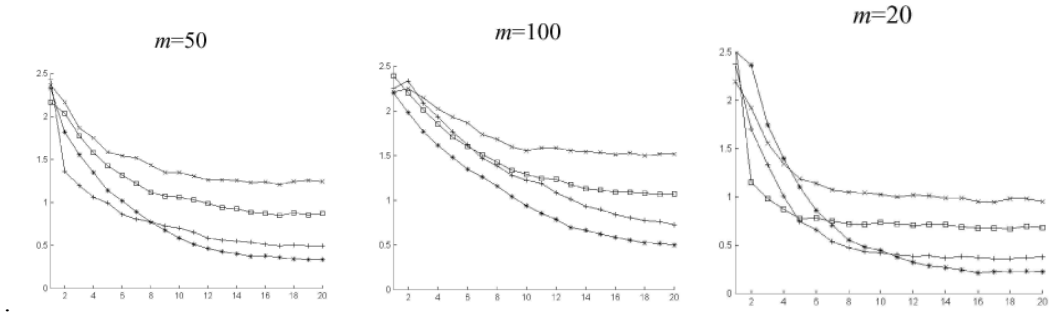
Rys.l. Położenie osobników w kolejnych 20 generacjach ewolucji (q=Q.l).

Na rys.2 przedstawiono średnie rozproszenie małych populacji (w=2,4,8) obserwowane w ciągu 20 generacji, dla różnych wartości odchylenia standardowego mutacji a. Średnie uzyskano ze 100 powtórzeń ewolucji, a populację początkową wybrano losowo z przedziału [-2,2] tak, aby jej rozproszenie było stosunkowo duże. Początkowo rozproszone populacje szybko skupiają się, a ich rozproszenie w dalszym przebiegu ewolucji wynosi około 0. Średnie rozproszenie dużych populacji dla powiększonego przedziału [-4,4], z którego wybrano losowo populację początkową

zilustrowano na rys.3. Wraz ze wzrostem liczebności populacji można zaobserwować zwiększenie się rozproszenia populacji do dwóch, a nawet trzech s.



Rys.2. Średnie rozproszenie małych populacji ze 100 powtórzeń ewolucji dla 20 generacji oraz różnych wartości standardowego rozproszenia mutacji: *-s=0.l, + -s=0.2, -s=0.4, x - s=0.6



.

Rys.3. Średnie rozproszenie dużych populacji ze 100 powtórzeń ewolucji dla 20 generacji oraz różnych wartości standardowego rozproszenia mutacji: *-s=0.l, + -s=0.2, -s=0.4, x - s=0.6.

Przedstawione powyżej wyniki potwierdzają więc wcześniejsze obserwacje, że zjawisko skupiania się populacji zdaje się być immanentną cechą procesów ewolucyjnych. Skupianie się populacji może być wolniejsze, gdy wzrośnie wymiarowość przestrzeni poszukiwań. Na analizowane zjawisko ma także wpływ liczba optimów funkcji jakości. Już wstępne analizy pokazują jednak, że populacje także skupiają się w okolicach optimów.

BIBLIOGRAFIA

1. J. Arabas, *Wykłady z algorytmów ewolucyjnych*, WNT, Warszawa, 2001
2. A. Chorążyczewski, R. Galar, I. Karcz-Dulęba: *Consideringphenotypic Evolution in the Space ofPopulation States*.
3. W: L.Rutkowski, R.Tadeusiewicz (ed) *Proceedings of the Fifth Conf. on Neural Networks and Soft Computing, Zakopane*, Poland 2000, s. 615-620.
4. R. Galar, *Miękka selekcja w losowej adaptacji globalnej w R^n*, Prace Naukowe ICT Politechniki Wrocławskiej, Monografie Nr 17, 1990
5. R.Galar, I.Karcz-Dulęba: *The evolution oftwo: An Example of Space of States Approach*. W: A.V. Sebald, L. J. Fogel (eds), Proceedings of the Third Annual Conf. on Evolutionary Programming, San Diego CA, World Scientific 1994, s. 261-268.

MODELING THE STRUCTURE OF GENETIC STRATEGIES WITH STOCHASTIC L-SYSTEMS

JOANNA KOŁODZIEJ

Institute of Mathematics,University of Bielsko-Biała,
ul. Willowa 2, 43-300 Bielsko-Biała, jkolodziej@ath.bielsko.pl

The best way of determining the efficiency of genetic algorithm is to construct its theoretical model and to apply it for the analysis the properties of the algorithm. Vose et al. (see [5]) have investigated an exact mathematical model for SGA, which forms a Markov chain in case of finite sizes of evaluating populations. But Vose's theory can't be easy extended to other strategies.

Another idea is using the stochastical grammars to construct theoretical models of genetic algorithms. A big advantage of the grammar approach is that the rules of the grammar can be used to generate strings which then automatically belong to the language of the grammar.

Lindenmayer systems (L-systems) are usually used in theoretical biology for describing and simulating natural growth processes (see [4]). All letters in a given word are replaced in parallel and simultaneously. This feature makes L-systems especially suitable for describing fractal structures, cell divisions in multicellular organism or flowering stages of herbaceous plants.

Koza (see [2]) applied Lindenmayer systems in genetic programming. Jacob and Rehder (see [1]) constructed the simple theoretical model of neural networks based on hierarchical grammar system. Using their methods we apply a context-sensitive stochastic L-system to describe the structure of Hierarchical Genetic Strategy (HGS) (see [3]).

Let us denote by $X_{s_j}^{n_j}$ the set of all possible populations of the size n_j consisting of genotypes of the length s_j, $X_{s_j}^{n_j} = \left(p_{i,j}^{e}\right);\ i = \left(i_1,\ldots,i_m\right);\ i_j \in \mathbf{N}$; $j = 1,\ldots,m$; $e = 1,2,\ldots$.

Our grammar $G = \left(\Sigma, \mathrm{A}, P\right)$ is characterized by:

- Alphabet

$$\Sigma = \bigcup_{j=1}^{m} X_{s_j}^{n_j} ,$$

- set of start symbols

$$\mathrm{A} = X_{s_1}^{n_1} ,$$

- terminals

$$T = \{;\} \cup \bigcup_{j=1}^{m} X_{s_j}^{n_j} ,$$

- non-terminals

$$N = \{poplist,\ pop\} ,$$

- production rules P specified below:
- $poplist := ";"\ pop$
- $pop := \left(p_{i,j}^{e} \middle| p_{i_1,j_1}^{e_1}\right)^{*}, \quad p_{i,j}^{e} \in X_{s_j}^{n_j}, \quad p_{i_1,j_1}^{e_1} \in X_{s_{j1}}^{n_{j1}}$
- elongation the core

$$p_{(1,0,\ldots 0),1}^{e} \xrightarrow{(\gamma_1)} p_{(1,0,\ldots 0),1}^{e+1};\quad e = 1,2,\ldots ;$$

– $p^0_{i,j} < p^1_{i,j} \xrightarrow{(\gamma_2)} p^0_{\tilde{\imath},j+1}$; $i = (i_1,...,i_{j-1},1,0,...,0)$; $\tilde{\imath} = (\tilde{\imath}_1,...,\tilde{\imath}_{j-1},1,1,0,...,0)$

– $p^e_{i,j} \xrightarrow{(\gamma_3)} p^0_{\hat{\imath},j+1}$; $\hat{\imath} = (\hat{\imath}_1,...,\hat{\imath}_{j-1},\hat{\imath}_j,0,...,0)$; $\hat{\hat{\imath}} = (\hat{\hat{\imath}}_1,...,\hat{\hat{\imath}}_{j-1},\hat{\hat{\imath}}_j,1,0,...,0)$

The probability γ_3 if the prefix comparison operator $PC(\omega_1,\omega_2,s_j) = 0$ for all $\omega_1,\omega_2 \in p^{e_1}_{\check{\imath},j+1}$ such that $\check{\imath} = (\check{\imath}_1,...,\check{\imath}_j,0,...0), 1 \le e_1 \le e-1, \quad 1 \le j \le m-1$.

– $p^e_{\bar{\imath},j} \xrightarrow{(\gamma_4)} p^{e+1}_{\bar{\bar{\imath}},j+1}$; $\bar{\imath} = (\bar{\imath}_1,...,\bar{\imath}_{j-1},\bar{\imath}_j,0,...,0)$; $j = 1,...m$

The probability $\gamma_4 = 1-\gamma_3$,

– $p^e_{i,m} \xrightarrow{(\gamma_5)} p^{e+1}_{i,m}$

The following populations can be created by this grammar after 3 metaepochs depending of the outcome of prefix comparison operator applieed after the second metaepoch:

$$p^3_{(1,0,...,0),1}; p^2_{(1,1,0,...,0),2}; p^1_{(1,2,0,...,0),2}; p^1_{(1,1,1,0,...,0),3}$$

or

$$p^3_{(1,0,...,0),1}; p^2_{(1,1,0,...,0),2}; p^1_{(1,1,1,0,...,0),3} .$$

In our research we want to definine the probabilites of production in this grammar applyig Vose's theory of genetic algorithms (see [5]).

BIBLIOGRAPHY

1. Jacob C., Rehder J.: *Evolution of Neutral Net Architectures by a Hierarchical Grammar-Based Genetic system*, *ICANNGA'93*, Insbruck , Springer Vlg., Wien, 1993, pp. 602-606,
2. Koza J.R. : *Discovery of Rewrite Rules in Lindenmayer Systems and State Transition Rules in Cellular automata Via Genetic Programming*, Symposium on Pattern Formation, 1993, Claremont, CA,
3. Kołodziej J. [2002] *Modelling Hierarchical Genetic Strategy as a Family of Markov Chains*, in R. Wyrzykowski, J. Dongarra, M. Paprzycki & J. Waśniewski (eds.),*Parallel Processing and Applied Mathematics,* LNCS, vol. 2328, Springer Vlg., s. 595-599,
4. Prusinkiewicz P., Lindenmayer A. : *The Algorithmic Beauty of Plants*, Springer Vlg., New York, 1990
5. Vose M.D.: *The Simple Genetic Algorithm*, MIT Press (1999).

ZASTOSOWANIE RODZAJNIKA GENETYCZNEGO JAKO KRYTERIALNEGO NISZOWANIA

ZDZISŁAW KOWALCZUK, TOMASZ BIAŁASZEWSKI

Politechnika Gdańska, Wydział Elektroniki Telekomunikacji i Informatyki
ul. Narutowicza 11/12, 80-952 Gdańsk, e-mail: {kova, bialas}@eti.pg.gda.pl

Streszczenie: Przedstawia się nowe podejście do rozwiązywania zadań wielokryterialnej optymalizacji oparte na algorytmach ewolucyjnych [1], w których każdemu rozwiązaniu przyporządkowana jest informacja o stopniu jego przynależności do danego podzbioru kryteriów, sygnowanego określonym genetycznym rodzajnikiem. Koncepcja wariantu genetycznego zasadza się na podziale zbioru funkcji celów na podzbiory. Podział ten ma pewne własności niszowania [1], jednak jest on związany z charakterem rozważanych celów (np. cele jakościowe i odpornościowe). Realizowana w ten sposób suboptymalizacja wykorzystywana jest w procesie ewolucji, w którym tylko rozwiązania o rożnym wariancie genetycznym mogą tworzyć pary rodzicielskie generujące nowe rozwiązania. Ilustrację rozważanego podejścia stanowić może przykład wielokryterialnej syntezy detekcyjnego obserwatora stanu [2], oraz wielokryterialna synteza regulatorów PID.

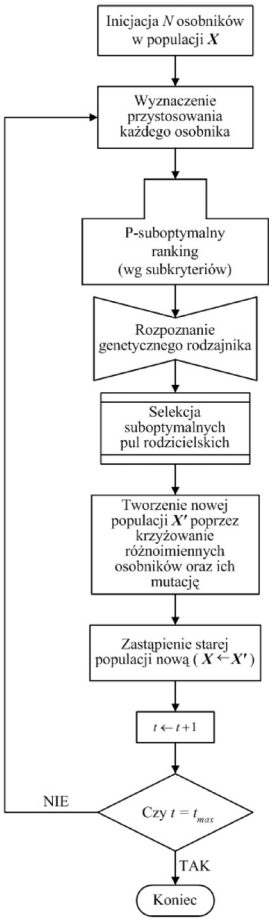
W naturze podział osobników ze względu na płeć wiąże się nie tylko z ich funkcjami rozrodczymi, ale również kojarzy się z ich funkcjonalną przydatnością do życia w danej społeczności. Zgodnie z tą obserwacją, proponowana idea genetycznego wariantu (GVAR) zasadza się na podziale zbioru funkcji celów na kilka rozłącznych podzbiorów, którym przypisywany może być określony atrybut (rodzajnik; X*j*) będący elementem zbioru GEND [3-4]. Do ustalonego podzbioru/wariantu (X*j*) mogą należeć kryteria o 'jednoimiennych' cechach, które są jedynie w stanie 'wewnętrznej' rywalizacji (posiadają w przybliżeniu jednakowe znaczenie). Taki zbiorczy sposób kwalifikacji zwalnia projektanta z trudnego zadania wyodrębnienia pojedynczego rozwiązania spośród wielu P-optymalnych rozwiązań. Natomiast zbiór rodzajników (wariantów) genetycznych (X*j*) wyraża różne grupy 'interesów' (kryteriów), które są trudne do apriorycznego wyważenia przez projektanta. Podział na owe podzbiory może być stosowany w celu umożliwienia 'zewnętrznej' rywalizacji 'różnoimiennych' cech, która nie jest prosta do rozwiązania. Do ostatecznej oceny uzyskanych rozwiązań można stosować na przykład koncepcję P-optymalności.

W podejściu GVAR stosuje się mechanizm przydziału rodzajnika podczas całego ewolucyjnego cyklu w celu tworzenia nowych rozwiązań, ale tylko z osobników rodzicielskich o różnych wariantach/rodzajnikach (X*j*). Wektor funkcji zysku $\boldsymbol{f}(x)=[f_1(x_1)\ f_2(x_1)\ \ldots\ f_m(x_1)]^T$ może być zatem podzielony na s podwektorów: $\boldsymbol{f}(x)=[f_1(x_1)\ f_2(x_1)\ \ldots\ f_s(x_1)]^T$ $\left(f_j(x)^T \in \Re^{m_i},\ m=\sum_{i=1}^{s} m_i,\ j=1,2,\ldots,s\right)$ definiujących podzbiór cech (wariant X*j*). W zakresie elementów tego podzbioru ustala się rangę opartą na pojęciu P-optymalności. Zatem każdemu z osobników przyporządkowuje się wektor rang $\boldsymbol{r}(x_i)=[r_1(x_1)\ r_2(x_1)\ \ldots\ r_s(x_1)]^T$, w którym $r_j(x_j)$ ($j = 1,2,\ldots,s$) reprezentuje rangę osobnika x_i w ramach j-tego subkryterium (wariantu X*j*). Przypisanie konkretnego rodzajnika osobnikom realizowane jest poprzez ich preselekcję (suboptymalizację) tzn.: , gdzie względna ranga $\varphi_i = \max_{i=1,2,\ldots,s} \varphi_i^j$, $l_i = \arg\max_{i=1,2,\ldots,s} \varphi_i^j$, gdzie względna ranga $\varphi_i^j = \dfrac{r_j(x_i)}{r_{j\max}}$, zaś $r_{j\max} = \max_{i=1,2,\ldots,N}\{r_j(x_i)\}$. Dzięki zastosowanej normalizacji, φ_i . stanowi najwyższy (rozmyty) stopień suboptymalności, tj. przynależności i-tego osobnika do l_i-tego (optymalnego dla niego) wariantu genetycznego, zaś r_{jmax} oznacza maksymalną rangę jaką osiągają wszystkie rozwiązania względem j-tego subkryterium (X*j*) [3-4].

Należy podkreślić, iż tylko osobniki różnoimienne (o odmiennym wariancie) tworzą pary biorące udział w generowaniu potomków poprzez proces krzyżowania. Procedura selekcji osobników do suboptymalnych (wariantowych, rodzajnikowych) pul rodzicielskich jest przeprowadzana metodą

stochastycznego doboru resztowego [2] na podstawie (efektywnego) stopnia przynależności (suboptymalności) φ_i.

Algorytm ewolucyjny z zastosowaniem wariantu/rodzajnika genetycznego przedstawia poniższy rysunek.



Rys.1. Schemat algorytmu ewolucyjnego z podejściem GVAR.

Z przeprowadzonych eksperymentów wyłania się kilka praktycznych aspektów proponowanej metody, które są w opozycji do tradycyjnego podejścia do wielokryterialnej optymalizacji: uzyskane fronty Pareto są bardziej regularne i jest ich więcej; zachowuje się rozmaitość rozwiązań; istnieje możliwość łatwego formułowania subkryteriów oraz są przesłanki co do końcowego rozstrzygnięcia.

LITERATURA

1. Goldberg D.E.: *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, Reading, 1989.
2. Kowalczuk Z., Białaszewski T.: *Pareto-optimal observers for ship propulsion systems by evolutionary computation*. Proc. IFAC Safeprocess, Budapest, 2000, ss.°914-919.
3. Kowalczuk Z., Białaszewski T.: *Evolutionary multi-objective optimisation with genetic sex recognition*. Symp. MMAR, Międzyzdroje, Poland, 2001, ss.°143-149.
4. Kowalczuk Z., Białaszewski T.: *Performance and robustness design of control systems via genetic-sex multi-objective optimization*. IFAC World Congress, Spain, 2002, T-Th-M16.

METODY OPTYMALIZACJI GLOBALNEJ W ZASTOSOWANIU DO SYSTEMÓW KOLEJKOWYCH

BARTŁOMIEJ JACEK KUBICA

4 sierpnia 2003

Systemy kolejkowe - zwłaszcza działające w warunkach niepewności - opisywane są skomplikowanymi nieliniowymi modelami. Często - nawet w przypadku stosunkowo prostych układów - trudno opisać je precyzyjnie. Stosuje się wówczas metody przybliżone lub symulacyjne.

Tym trudniejszym zadaniem jest zatem *optymalizacja* takich układów. Zastosowanie wyszukanych metod poszukiwania ekstremum wydaje się tu nieodzowne.

Jednym z ciekawych zagadnień jest zastosowanie mechanizmów cenowych do sterowania obciążeniem systemów kolejkowych. Problemem tym zajmowano się m. in. w pracach [l, 4, 5, 6].

Można rozważać wiele wariantów tego zadania - klienci systemu obsługi znają lub nie jego stan przed podjęciem decyzji o dołączeniu do kolejki, po wejściu do kolejki mogą lub nie ją opuścić. Różna też może być informacja, jaką dysponują klienci oraz operator, ustalający wartości opłat.

W najprostszym przypadku (opisanum przez Naora w pracy [6]) mamy tylko jedną kolejkę $M/M/1$ i jedną klasę klientów, pragnących uzyskać usługę, która ma dla nich wartość R. Koszt ich oczekiwania na realizację usługi jest liniowy i wynosi c za każdą jednostkę czasu.

Jeśli w systemie znajduje się n klientów, to oczekiwany czas przepustowości dla ostatniego z nich wynosi:

$$\mathbf{E}[\tau;n] = n \cdot \frac{1}{\mu}$$

gdzie μ jest intensywnością obsługi. Jeśli klient podejmuje decyzję na podstawie tego czasu oczekiwanego, to dołączy on dej kolejki, jeśli

$$R \geq p + c \cdot \mathbf{E}[\tau;n]$$

gdzie p to opłata. Rozmiar kolejki jest zatem de facto ograniczony do pewnego n_{max} wynikającego z powyższej nierówności.

$$\max \lambda\,(1 - \pi_{nmax})\,p$$

lub zadanie optymalizacji „dobra społecznego":

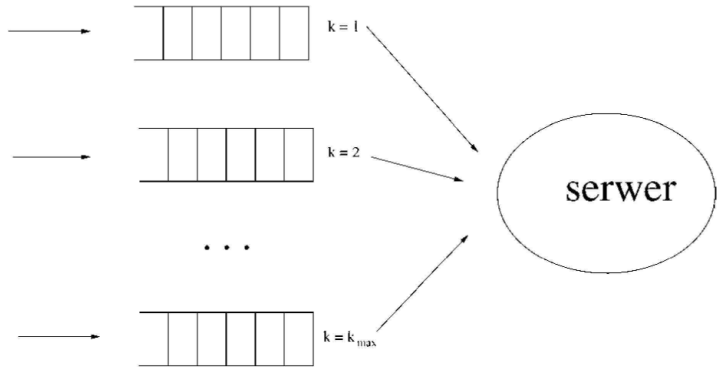
$$\max \lambda\,(1 - \pi_{nmax})\,(R - \underline{c}\mathbf{E}\tau)$$

Oczywiście wielkości π_n i $\mathbf{E}_\tau$ określone są znanymi wyrażeniami (patrz np. [2]):

Znając tę wielkość możemy precyzyjnie opisywać zachowanie systemu (oczywiście w kategoriach stochastycznych) i sformułować zadanie optymalizacji zysków dostawcy usług:

$$\pi_n = \frac{1-\rho}{1-\rho^{n+1}} \cdot \rho^n \tag{1}$$

Powyższe zagadnienie ma stosunkowo proste, jednoznaczne rozwiązanie. Jeśli jednak mamy do czynienia z kolejką priorytetową ([3], patrz rysunek), z wieloma klasami klientów ([l]) lub innymi bardziej złożonymi modelami, nie da się już rozwiązać problemu prosto.



Rys. 1 Schemat serwera, przyjmującego zgłoszenia od klientów o różnych priorytetach

Konieczne wydaje się wówczas zastosowanie metod optymalizacji niewypukłej, np. algorytmów ewolucyjnych albo metod podziałowych.

Przykładowe modele systemów i propozycje algorytmów ich optymalizacji autor ma zamiar zaprezentować na Warsztatach.

LITERATURA

1. N. M. Edelson, D. K. Hildebrand: *Congestion tolls for Poisson Queuing Processes*, Econometrica, Vol. 43, No. 1, 1975.
2. B. Filipowicz: *Modele stochastyczne w badaniach operacyjnych. Analiza i synteza systemów obsługi i sieci kolejkowych*, WNT, 1996
3. J. M. Harrison: *A Priority Queue with Discounted Linear Costs,* Operations Research, Vol. 23, No. 2, 1975
4. N. Chr. Knudsen: *Individual and Social Optimization in a Multiserver Queue with a General Cost-benefit Structure*, Econometrica, VoL 40., No. 3, 1972
5. H. Mendelson, S. Whang: *Optimal Incentive - Compatible Priority Pricing for the M/M/l Queue,* Operations Research, Vol. 38, No. 5, 1990
6. P. Naor: *The Regulation of Queue Size by Levying Tolls*, Econometrica, Vol. 37, No. l,

SYSTEMY HYBRYDOWE – ŁĄCZENIE ALGORYTMÓW EWOLUCYJNYCH I SIECI NEURONOWYCH

HALINA KWAŚNICKA

Politechnika Wrocławska, Zakład Informatyki, Wydział Informatyki i Zarządzania,
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Halina.Kwasnicka@pwr.wroc.pl

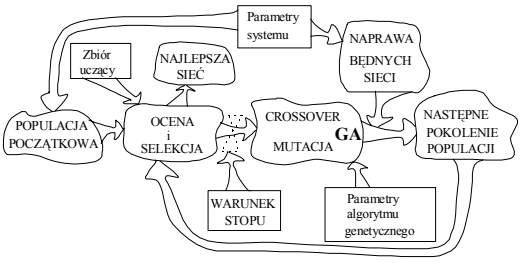
Łączenie różnych technik sztucznej inteligencji pozwala na budowanie systemów o znacznie większych możliwościach. Systemy hybrydowe są tu rozumiane jako systemy łączące w sobie kilka technik sztucznej inteligencji, np. systemy ekspertowe, sieci neuronowe, algorytmy genetyczne. Łączą one w sobie zalety poszczególnych technik, eliminując ich wady.

Integracja sieci neuronowych i algorytmów genetycznych to obszar bardzo intensywnych i szerokich badań. Użyteczną w systemach hybrydowych cechą algorytmów genetycznych jest ich zdolność do przeszukiwania dużych, złożonych przestrzeni potencjalnych rozwiązań. Mogą one współpracować z sieciami neuronowymi na różne sposoby. W pracy zajmiemy się tzw. *hybrydowymi systemami luźno powiązanymi* (ang. *loosely coupled*), w których jeden moduł systemu komunikuje się z drugim poprzez odpowiedni interfejs. Wśród możliwych połączeń algorytmów ewolucyjnych i sztucznych sieci neuronowych wyróżnić można:

1. Projektowanie sieci neuronowych przy wykorzystaniu algorytmów genetycznych,
2. Przygotowanie danych uczących dla sieci neuronowych (selekcja i konstrukcja cech),
3. Pozyskiwanie wiedzy z sieci neuronowych w postaci zrozumiałej dla człowieka.

PROJEKTOWANIE SIECI NEURONOWYCH ZA POMOCĄ AE

Zastosowanie algorytmów genetycznych do sieci neuronowych wiąże się z pewnymi problemami, należą do nich: (i) reprezentacja sieci neuronowej w postaci chromosomu, (ii) ocena przystosowania sieci neuronowej, (iii) stosowanie operatorów genetycznych zapewniających poprawność produkowanych sieci [4]. Schemat systemu wykorzystującego **EA** do projektowania **NN** pokazany jest na rysunku 1. Ciekawe efekty uzyskano podczas badań nad użytecznością algorytmów genetycznych z efektami plejotropowości i poligeniczności do projektowania sieci neuronowych [3]. Badania przeprowadzono dla pięciu problemów: XOR, Autoenkoder, Termometr, Dodawanie liczb dwubitowych, Rozpoznawanie cyfr. Dla każdego z nich algorytm wykorzystujący efekt plejotropowości i poligeniczności miał lepszą skuteczność w znajdowaniu rozwiązań w stosunku do klasycznego **GA**, liczba pokoleń potrzebnych do znalezienia rozwiązania też zazwyczaj była mniejsza. Wyniki eksperymentów wskazują, że zarówno skuteczność jak i liczba pokoleń potrzebnych na znalezienie rozwiązania nie zależą zbytnio od liczebności populacji. Niestety proces przejścia z genotypu na fenotyp jest bardziej czasochłonny, ale biorąc pod uwagę potrzebną liczność populacji i liczbę pokoleń, wada ta staje się zdecydowanie mniej istotna

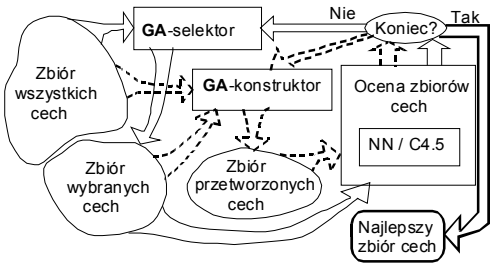


Rys. 1. Projektowanie sieci neuronowych za pomocą GA

GA W ZADANIU DOBORU ODPOWIEDNIEGO ZESTAWU CECH

Niezależnie od wybranej metody automatycznego uczenia się (NN, ID3, itp.), przygotowanie odpowiedniego zestawu cech jest zadaniem istotnym. System powinien wykorzystywać możliwie najmniejszą liczbę cech, pozwalającą na poprawne uczenie się. Uwzględnianie cech, które nie mają własności rozróżniających, wydłuża proces obliczeniowy nie zwiększając poprawności uczenia. Z drugiej strony, dążąc do minimalizacji liczby cech, nie można zrezygnować z cech istotnych dla właściwego rozpoznania podawanych na wejście wzorców.

Do wyboru odpowiedniego zestawu cech można z powodzeniem stosować algorytm genetyczny. **GA** może zarówno przeprowadzić selekcję cech, jak i skonstruować nowe cechy, będące kombinacją tych wybranych, co w efekcie jeszcze zmniejsza liczbę koniecznych cech do analizowania przez system uczący się (Rysunek 2). W literaturze można znaleźć przykłady zastosowania **GA** do przygotowania danych dla takich algorytmów, jak **ID3** czy **C4.5**. Dane wejściowe dla całego systemu stanowi zbiór możliwych cech. Można zaimplementować dwa niezależne moduły stosujące **GA**: jeden, tzw. **GA**–*selektor* ma za zadanie wybrać te cechy, które są istotne z punktu widzenia zadania rozpoznawania, drugi to **GA**–*konstruktor*, jego zadanie to stworzenie nowych cech będących kombinacją cech wejściowych. Moduły te stosują inną reprezentację chromosomów, mogą działać niezależnie. **GA**–*selektor* operuje na osobnikach będących podzbiorem możliwych cech, efektywne jest kodowanie osobnika na tylu bitach, ile jest cech. Jedynka na danej pozycji oznacza, że dana cecha jest włączona do zestawu wybranych cech, zero – eliminację cechy. **GA**–*konstruktor* musi wykorzystywać inną reprezentację – nowe cechy są tworzone jako kombinacja istniejących. Naturalną reprezentacją dla takich zadań jest struktura drzewiasta.



Rys. 2. Schemat systemu uczącego się z wstępnym doborem cech przez GA

W [2] zastosowano **GA** do ograniczenia liczby cech dla sieci neuronowej i algorytmu generowania drzew decyzyjnych **C4.5**. Eksperymenty przeprowadzono na 4 benmarchkowych bazach testowych: Arrhythmia (297 cech, 452 wektory uczące), Waveform (40 atrybutów, z czego 19 to szumy, 5000 wektorów), Sonar (60 atrybutów, 208 wektorów) i Satimage (36 atrybutów, 6435 wektorów). Wyniki są zaskakująco dobre: w większości przypadków udało się nie tylko zmniejszyć zestaw cech w danych uczących (nawet o 75%), ale też istotnie poprawić dokładność klasyfikacji przez wyuczoną na pomniejszonym zbiorze sieć (od kilu do blisko 20 %).

EKSTARKCJA REGUŁ Z SIECI NEURONOWEJ ZA POMOCĄ ALGORYTMÓW EWOLUCJNYCH

Pozyskiwanie z wyuczonej sieci neuronowej zrozumiałych dla człowieka reguł jest ostatnio bardzo popularnym tematem. Wśród wielu podejść można znaleźć takie, które wykorzystują algorytmy ewolucyjne. Tematowi temu poświęcona jest praca [1], w której przeanalizowano użyteczność różnych podejść. Okazuje się, że obiecujące wyniki daje metoda, w której **GA** przeszukuje przestrzeń atrybutów (tzn., które atrybuty uwzględniać w tworzonej regule), a strategie ewolucyjne przeszukują zakresy atrybutów.

W niniejszym dokumencie, ze względu na szczupłość miejsca, jedynie naszkicowano możliwe (i celowe) łączenia algorytmów ewolucyjnych i sieci neuronowych – dwóch metod wzorowanych na naturze.

LITERATURA

1. Kwaśncika H., Markowska-Kaczmar U., Osojca T.: *Usefuleness of Evolutionary Methods in the Rule Extraction*. Mendel Conference, Brno, Czechy, 2003.
2. Orski P.: *Użyteczność algorytmów genetycznych w przygotowywaniu danych dla algorytmów uczących się*. Praca magisterska, Politechnika Wrocławska, Wrocław 2003.
3. Rynkiewicz S.: *Zaawansowane operatory genetyczne w projektowaniu sieci neuronowych*. Praca magisterska, Politechnika Wrocławska, Wrocław 2000.
4. Stanley K. O., Miikkulainen R.: *Evolving Neural Networks through Augmenting Topologies*. Evolutionary Computation 10(2):99-127, 2002. http://www.cs.utexas.edu/users/nn/pages/publications/ne-methods.html

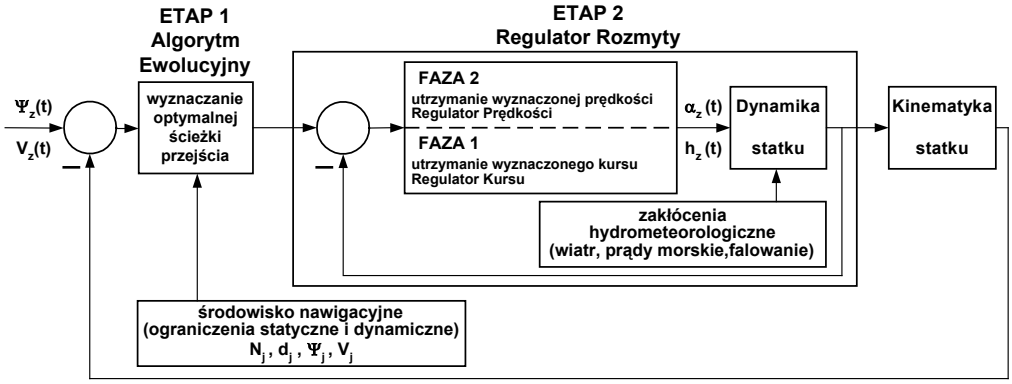
WYKORZYSTANIE ALGORYTMU EWOLUCYJNEGO I LOGIKI ROZMYTEJ DO KIEROWANIA STATKIEM

ANDRZEJ ŁEBKOWSKI†, ROMAN ŚMIERZCHALSKI‡

‡ Akademia Morska w Gdyni, Katedra Automatyki Okrętowej, roms@am.gdynia.pl
† Politechnika Gdańska, Katedra Automatyki, andrzejl@ely.pg.gda.pl

Rozwój technik komputerowych w ostatnich latach, wpłynął na zmiany w sposobie prowadzenia nawigacji na współczesnym statku floty handlowej. Dysponując doświadczeniami związanymi z zatonięciem promu „Estonia" pod pokładem którego życie utraciły 852 osoby (95 ciał odnaleziono, 757 uznano za zaginione) oraz innych jednostek morskich, opracowano Konwencję STCW-95 nakładającą na oficerów nawigacyjnych i personel pokładowy, obowiązek przyswajania i odświeżania prawideł morskich na kursach ponawianych w regularnych odstępach czasu. Jednak wprowadzone w 1995 roku przepisy nie wyeliminowały katastrof i wypadków do jakich dochodzi na morzu. Bezpieczeństwo życia na morzu nie zależy tylko od jakości zainstalowanej aparatury, ale jak wskazuje praktyka od wiedzy i umiejętności nawigatorów [1]. Statystyki wypadków do jakich dochodzi na morzu potwierdzają, że czynnik ludzki jest jedną z ich głównych przyczyn. Udział błędów ludzkich będących przyczyną powstania katastrof wynosi około 80 %.

Dlatego mając na uwadze obecny stan wyposażenia mostków nawigacyjnych oraz statystyki wypadków morskich, opracowana została koncepcja hybrydowego ewolucyjno-rozmytego systemu sterowania statkiem o nazwie PFC (Path Finder and Controller for Ship) [3].



Rys.1. Struktura sterowania statkiem w sytuacji kolizyjnej z zastosowaniem systemu PFC, Ψ_z – zadany kurs własnego statku, V_z – zadana prędkość własnego statku, α_z – zadane wychylenie płetwy sterowej, h_z – zadana wartość prędkości obrotowej silnika głównego, N_j – namiar na j-ty obiekt, d_j – dystans do j-tego obiektu, Ψ_j – kurs j-tego obiektu, V_j – prędkość j-tego obiektu.

Przedstawiony system ma umożliwić sterowanie statkiem, w sytuacjach kolizyjnych, gdzie problem wyznaczenia bezpiecznej ścieżki przejścia statku, sprowadzony został do zadania dwuetapowego (rys. 1.):

- w pierwszym etapie AE na podstawie zadanego punktu docelowego, określa optymalną parametryzowaną czasem ścieżkę przejścia statku, zawierającą sekwencję odcinków o ustalonym kursie i prędkości w zależności od sytuacji nawigacyjnej rejestrowanej przez system ARPA [2].
- w drugim etapie RR uwzględniając dynamikę statku oraz panujące warunki hydrometeorologiczne środowiska, wyznacza sterowanie dla rzeczywistej trajektorii przejścia statku. Etap ten składa się z dwóch równolegle prowadzonych faz obliczeniowych:
  - pierwszej, w której kontrolowane i korygowane jest na bieżąco odchylenie od wyznaczonego kursu, w zależności od sytuacji nawigacyjnej, wpływu prądów morskich, falowania i wiatru (regulator kursu),

- drugiej, w której kontrolowana jest prędkość przemieszczania się po zadanym torze ruchu (regulator prędkości).

Rozwiązania obydwu etapów pozwalają na przeprowadzenie z wyprzedzeniem czasowym animacji scenariusza rozwoju sytuacji kolizyjnych z uwzględnieniem trajektorii przejścia statku własnego.

Istotną cechą prezentowanego systemu będzie możliwość zobrazowania aktualnej sytuacji nawigacyjnej dla statku będącego w podróży, z określeniem parametrów ruchu spotkanych obiektów. W zależności od zaistniałej sytuacji nawigacyjnej (nasilenia ruchu, regionu żeglugi) i panujących warunków meteorologicznych, nawigator będzie miał możliwość ustawienia parametrów mijania ze spotkanymi obiektami dynamicznymi oraz ograniczeniami natury statycznej.

System PFC kierując statkiem, wyposażony będzie w procedury alarmujące załogę w przypadku nadmiernego zbliżenia do innej jednostki będącej w ruchu lub ograniczenia o charakterze statycznym, jak również w procedury alarmowe dla sytuacji w której nie może znaleźć zadowalającego rozwiązania - powodującej chwilowe zatrzymanie statku. Problem ten jest ważny w momencie wejścia statku na akwen, na którym nie działa system nadzoru ruchu a poruszające się jednostki nie podlegają Konwencji SOLAS, są to: jachty, statki wycieczkowe żeglugi przybrzeżnej, statki rybackie, holowniki, łodzie itp. Wymienione jednostki najczęściej nie posiadają odpowiednich środków łączności, a manewrowanie w ich otoczeniu stwarza sytuację zagrożenia kolizyjnego. Jak wynika z światowych statystyk – 90% akcji ratowniczych podejmowanych na morzu przez Morskie Centra Koordynacji Ratownictwa, dotyczy właśnie w/w jednostek [4].

Opisany hybrydowy system PFC bezpiecznego sterowania statkiem w sytuacji kolizyjnej, wykorzystujący do wyznaczenia optymalnej trajektorii przejścia statku techniki komputerowe: algorytmy ewolucyjne i sterowanie rozmyte, stanowi nowe podejście do problemu unikania kolizji na morzu w środowisku z ograniczeniami nawigacyjnymi o charakterze statycznym i dynamicznym. Na podstawie przeprowadzonych rozważań, przyjęto iż:

- możliwe jest zastosowanie systemu PFCS do rozwiązywania problemu unikania kolizji na morzu, gdzie środowisko modelowane jest w postaci wielokątów reprezentujących ograniczenia nawigacyjne, jak i poruszające się ze zmienną prędkością obiekty obce,
- użycie hybrydowego algorytmu ewolucyjnego składającego się z dwóch różnych części o modyfikowanych operatorach genetycznych wpłynie na poprawę skuteczności i czasu obliczeń mających na celu wyznaczenie optymalnej ścieżki przejścia dla statku będącego w podróży,
- zastosowanie regulatora rozmytego charakteryzującego się mniejszą wrażliwością na zakłócenia niż dotychczas powszechnie stosowane konwencjonalne regulatory autopilotów okrętowych oraz możliwość wbudowania w bazę wiedzy regulatora elementów wiedzy i bazy wiedzy eksperta, spowoduje w niedługiej przyszłości powszechne zastosowanie tego typu koncepcji do sterowania statkiem na morzu.
- zastosowanie systemu PFC stwarza realne szanse dla zmniejszenia ilości katastrof do jakich dochodzi na morzu oraz wzrostu poziomu bezpieczeństwa, gdyż system ten nie jest uzależniony od systemów kontroli ruchu. System PFC może pracować autonomicznie w każdej sytuacji nawigacyjnej w jakiej znajdzie się statek w niego wyposażony.

LITERATURA

1. Śmierzchalski R. - *Synteza metod i algorytmów wspomagania decyzji nawigatora w sytuacji kolizyjnej na morzu*, Monografia, Zeszyty Naukowe WSM, Gdynia 1998, pp. 1-171.
2. Śmierzchalski R., Michalewicz Z. - *Modeling of a Ship Trajectory in Collision Situations at Sea by Evolutionary Algorithm*. J. of IEEE Transaction on Evolutionary Computation No. 3 Vol. 4, pp. 227-241, 2000.
3. Śmierzchalski R., Łebkowski A. *Genetic operators of evolutionary algorithm in problem of collision avoidance at sea* - Konferencja Algorytmy Ewolucyjne i Optymalizacja Globalna, Kraków 2002 r.
4. Czajkowski J. *Podręcznik operatorów systemu GMDSS*, Gdańsk 1998 r.

ALGORYTM EWOLUCYJNY DO OPTYMALIZACJI ZADAŃ Z KOSZTOWNĄ FUNKCJĄ CELU

PRZEMYSŁAW MIAZGA

Instytut Radioelektroniki Politechniki Warszawskiej
ul.Nowowiejska 15/19, 00-665 Warszawa, p.miazga@ire.pw.edu.pl

STRESZCZENIE

W referacie przedstawiono koncepcję hybrydowego algorytmu do optymalizacji zadań z niegładką i kosztowną funkcją celu. Procedura wykorzystuje metodę „krigging" do aproksymacji rzeczywistej funkcji celu (response surface) oraz algorytm ewolucyjny do poszukiwania optimum aproksymowanej funkcji.

WSTĘP

Do projektowania urządzeń elektronicznych pracujących w wysokich pasmach częstotliwości, stosowane są symulatory elektromagnetyczne SE [1]. SE do obliczania odpowiedzi obwodu wykorzystują metody iteracyjnego rozwiązywania równań Maxwella w dziedzinie czasu. Obwód jest dzielony na elementarne sześciany, tzw. oczka, czyli węzły, dla których, zdefiniowane są wektory pól elektrycznego i magnetycznego. Na wejściu obwodu definiujemy składowe pól, zmienne w czasie, o z góry założonym widmie. Następnie, co określony odcinek czasu Δt obliczane są zmiany pól w węzłach obwodu. Charakterystyki obwodu obliczane są na bieżąco przy użyciu szybkiej transformaty Fouriera. Dokładność symulacji jest zależna od wymiaru oczka a i kwantu czasowego Δt – im mniejsze są te parametry, tym większa dokładność, ale nakład obliczeń rośnie wykładniczo w stosunku do a, a wymagana pamięć jest proporcjonalna do sześcianu Δt. Przykładowe czasy dokładnych symulacji sięgają od kilku minut do kilku tygodni. Można zmniejszyć dokładność symulacji, ale odbywa się to kosztem zniekształcenia charakterystyce obwodu (najczęściej następuje przesunięcie optimum), oraz zwiększenia błędu systematycznego symulacji. Jest to o tyle istotne, że dwukrotna zmiana wielkości oczka może spowodować kilkunastokrotne zwiększenie zaburzeń funkcji celu.

Jednym z rozwiązań problemu jest optymalizacja zadania w aproksymowanej przestrzeni funkcji celu. W pracy [2] zaprezentowano bezgradientową metodę poszukiwań prostych w przestrzeni funkcji aproksymowanej wielomianowo. Rozwiązanie to pozwala na znalezienie sensownego lokalnego optimum dla 1-3 zmiennych w czasie kilku godzin/dni. Niestety, algorytm jest lokalny, a przy tym, charakteryzuje się bardzo słabą zbieżnością

ALGORYTM EWOLUCYJNY Z APROKSYMACJĄ FUNKCJI CELU

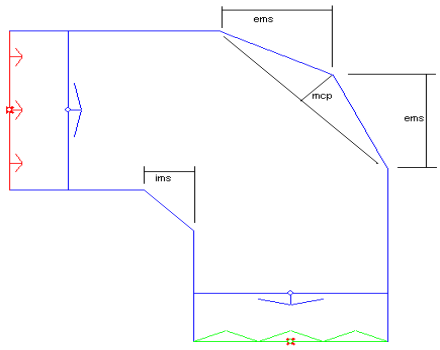
Zastosowanie efektywniejszej metody aproksymacji („krigging") symulowanej funkcji celu [3] oraz globalnego (przynajmniej z natury) algorytmu ewolucyjnego wydaje się być lepszym rozwiązaniem. W ramach pracy opracowano hybrydowy algorytm, którego zadaniem jest optymalizacja funkcji celu w przestrzeni aproksymowanej, modyfikowanej w trakcie obliczeń przez dodanie punktów uzyskanych w wyniku symulacji najlepszych rozwiązań. Sieć działań proponowanego algorytmu przedstawiono poniżej:

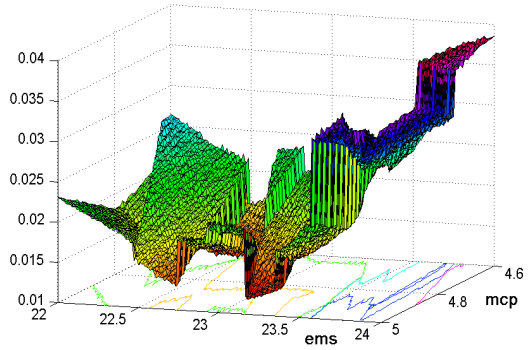
1. generujemy początkowy zbiór punków w przestrzeni rzeczywistej (symulacja)
2. generujemy populację początkową AE
3. aproksymujemy funkcję celu na podstawie punktów symulowanych
4. optymalizujemy funkcję w przestrzeni aproksymowanej algorytmem ewolucyjnym z ograniczoną liczbą generacji
5. dla jednego lub kilku najlepszych osobników z populacji dokonujemy symulacji,
6. sprawdzamy kryterium stopu, jeśli nie jest spełnione przechodzimy do p.3

Zastosowano algorytm μ,λ z selekcją proporcjonalną i kodowaniem całkowitoliczbowym. Zmienne o rzeczywistych wartościach odwzorowuje się na liczbą całkowitą z dokładnością do trzeciego miejsca po przecinku. Do zaburzania zmiennych służą operatory mutacji o rozkładzie gaussowskim oraz operator rekombinacji uśredniającej.

NUMERYCZNA WERYFIKACJA ALGORYTMU

Algorytm został zastosowany do optymalizacji zagięcia falowodu prostokątnego [1], przedstawionego na rysunku, z dwoma lub trzema wymiarami optymalizowanymi. Na prawej części rysunku przedstawiono wykres funkcji celu, uzyskanej w wyniku symulacji.





Funkcja celu jest zaburzona przez występowanie błędu systematycznego symulacji, objawiającego się w postaci ostrych stoków i wąwozów. W wyniku optymalizacji jest uzyskiwane rozwiązanie różniące się o kilka procent od optymalnego (w praktyce pomijalny błąd) w czasie rzędu 1-2 godzin to jest 30-40 symulacji. Dla porównania uzyskanie rozwiązania metodą bez aproksymacji funkcji celu wymaga 180 do 300 symulacji.

Rzeczywista funkcja celu, jaką można uzyskać w bardzo dokładnej symulacji jest gładka i wypukła. Można ją stosunkowo łatwo zoptymalizować (np. metodą Gaussa-Seidla lub Powella), ale czas optymalizacji, pomimo wymaganej niewielkiej liczby symulacji, jest znacznie większy - rzędu kilkunastu godzin.

PODSUMOWANIE

Należy podkreślić, że zastosowanie Algorytmu Ewolucyjnego w przestrzeni aproksymowanej funkcji celu pozwala na znalezienie rozwiązania globalnego dla większej niż dotychczas liczby zmiennych (testowano problem z 2, 3 i 6-cioma zmiennymi) przy mniejszej wymaganej dokładności symulacji (metoda jest mniej wrażliwa na błąd systematyczny), co pozwala na znaczne zmniejszenie czasu syntezy układu. Metodę można w prosty sposób rozproszyć. Równolegle może być wykonywany proces AE oraz dowolna (zależna od liczby procesorów/stacji roboczych) liczba symulacji.

LITERATURA

1. *QuickWave 3D electromagnetic simulator users manual*, „Qwed” sp. z o. o. Warszawa
2. J.W.Bandler et al., *Electromagnetic optimization of 3D structures*, IEEE Trans on MTT, vol 45, May 1997, pp. 770—779
3. D.W.Jones, A *Taxonomy of Global Optimization Methods Based on Responce Surfaces*, Journal of Global Optimization 21, 2001, pp. 345-383

PREDYKCYJNE STEROWANIE EWOLUCYJNE

TOMASZ MILEWSKI, MACIEJ TRYKA, PIOTR KASZYŃSKI, PAWEŁ D. DOMAŃSKI

Institute of Control and Computation Engineering, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland, e-mail: domanski@ia.pw.edu.pl.

STRESZCZENIE

Celem było stworzenie ewolucyjnego nieliniowego sterownika predykcyjnego z modelem typu perceptron jednowarstwowy. Proponowana struktura jest napędzana przy pomocy niestacjonarnego algorytmu ewolucyjnego. Sterowania są bezpośrednio rozwiązaniami optymalizacji ewolucyjnej, przy czym iteracje (epoki) algorytmu ewolucyjnego przebiegają w czasie rzeczywistym. Jako przykład testowy regulator za zadanie kontrolować ruchy płetwą sterową statku tak, by ten jak najbliżej trzymał się linii kursu wyznaczonej przez oś wzdłużną statku. Ponadto uwzględniono również praktyczne aspekty sterowania, takie jak na przykład minimalizacja ruchów płetwą sterową. Do testów obrano uproszczony model tankowca umożliwiający sprawdzenie poprawności działania sterownika. Opisy poszczególnych elementów sterownika przedstawiamy poniżej.

OPTYMALIZACJA

Do optymalizacji postanowiono użyć algorytm genetyczny zmodyfikowany tak, by spełniał założenia czasowe. Według tychże założeń, sterownik powinien wystawiać sterowania w interwałach czasowych nie dłuższych niż jedna sekunda. Oczywiście szybkość algorytmu (ilość epok na sekundę) mocno związana jest z mocą maszyny, na której jest on testowany. Testy przeprowadzano, więc na komputerze PC średniej klasy starając się udoskonalić tak działanie sterownika, by już na takim komputerze był on dostatecznie sprawny i by zwiększanie mocy obliczeniowej poprawiało działanie, lecz nie było konieczne. Sam algorytm opiera się na dość standartowych schematach ewolucyjnych [1].

KODOWANIE OSOBNIKÓW

Każdy osobnik składa się z dwóch chromosomów. Jeden z nich zwiera geny będące zmianą sterowania w kolejnych sekundach w przeciągu całego horyzontu predykcji. Drugi z nich ma zakodowane współczynniki potrzebne do mutacji kolejnych genów pierwszego chromosomu. Pierwszy chromosom zwiera, więc liczby rzeczywiste oznaczające zmianę wychylenia płetwy sterowej w stopniach w kolejnych sekundach. Na każdy gen nałożone zostały ograniczenia kostkowe, które z jednej strony dostosowane są do realiów sterowania statkiem, z drugiej uniemożliwiają przypadkowe, duże skoki sterowania. Jest to szczególnie ważne przy niedeterministycznej optymalizacji z tak restrykcyjnymi ograniczeniami czasowymi. Drugi chromosom umożliwia samoczynną adaptację zasięgu mutacji i pozwala dokładniej przeszukiwać maksima lokalne funkcji przystosowania. Jest to klasyczny schemat (μ,λ) z dodatkową modyfikacją, która nie selekcjonuje osobników potomnych. Do następnej generacji brane są wszystkie osobniki wygenerowane podczas danej epoki. Przyspiesza to działanie algorytmu i pozwala uniknąć operacji sortowania.

REPRODUKCJA

Jako metodę reprodukcji osobników wybrano reprodukcję turniejową, jako jedną z najprostszych i najlepszych, umożliwiającą jednocześnie łatwe zmiany nacisku selektywnego na populację. Metoda ta nie wymaga szczególnego nakładu obliczeniowego, nie wymaga sortowania (jak na przykład reprodukcja rangowa), toteż w naszym przypadku jest szczególnie przydatna. Do populacji potomnej dobieranych jest dokładnie tyle osobników, ile jest potrzebne do wygenerowania populacji potomnej o tej samej liczności, co populacja rodzicielska (po uwzględnieniu krzyżowania).

KRZYŻOWANIE

Wybór schematu krzyżowania również był podyktowany względami czasowymi, wobec tego zastosowano krzyżowanie najprostsze z możliwych, spełniające założenia o nieobciążoności. Krzyżowanie generuje z dwóch rodziców jednego potomka i odbywa się według wzoru:

$$X' = aX^1 + (1-a)X^2 \tag{1}$$

gdzie a jest realizacją zmiennej losowej o rozkładzie jednostajnym na odcinku [0;1]. Schemat ten stosowany jest zarówno dla chromosomu sterowań, jak i chromosomu wariancji.

MUTACJA

Podobnie jak w poprzednich operatorach, tak i tu przy wyborze schematu kierowaliśmy się przede wszystkim względami czasowymi. Jest to standardowy schemat mutacji dla strategii (μ,λ) wykorzystujący chromosom wariancji. Mutacja odbywa się według wzoru:

$$X_i^{'} = X_i + \sigma_i \xi_{N(0,1),i} \tag{2}$$

gdzie i to numer genu w chromosomie, a ξ oznacza zmienną losową o rozkładzie standartowym normalnym, różną dla każdego genu. Modyfikacja chromosomu wariancji odbywa się według odmiennej reguły, umożliwiająca samoczynną adaptację zasięgu mutacji pierwszego chromosomu:

$$\begin{gathered} \sigma_i^{'} = \sigma_i \exp(\tau' \xi_{N(0,1)} + \tau \xi_{N(0,1),i}) \\ \tau = \frac{K}{\sqrt{2n}} \qquad \tau' = \frac{K}{\sqrt{2\sqrt{n}}} \qquad K = 0.1 \end{gathered} \tag{3}$$

Umożliwia ona dość szybką zbieżność algorytmu uzyskując sporą liczbę znajdywanych coraz to lepszych osobników, szczególnie przy wymaganiu dużych zmian sterowania.

MODEL TESTOWY

Do testów sterownika obrano uproszczony model statku. Jest to model nieliniowy z wprowadzonymi zakłóceniami w postaci zmieniającego się w czasie wiatru oraz siły wywoływanej przez fale. O statku założono, że jest statkiem klasy tankowiec, o długości 100 i szerokości 10 metrów, przy orientacyjnej masie dla tankowców wynoszącej 3500 ton. Przyjęliśmy uproszczony model kadłuba jako prostopadłościan, co umożliwiło łatwe obliczenie momentu bezwładności wokół pionowej osi obrotu w środku ciężkości statku. Statek, na podstawie [2] opisany jest układem trzech równań, które po przekształceniu dają układ trzech równań różnicowych opisujących prędkości, z jakimi porusza się statek wzdłuż odpowiednich osi:

$$\begin{gathered} u_n = dt\left[\frac{F_{x,n-1}}{m} + r_{n-1}\cdot v_{n-1} + \frac{\frac{x}{2}(N_{n-1} - xF_{y,n-1})}{I - mx^2}\right] + 0.8185^{dt} u_{n-1} \\ v_n = dt\left[\frac{F_{y,n-1}}{m} - r_{n-1}\cdot u_{n-1} - \frac{\frac{x}{2}(N_{n-1} - xF_{y,n-1})}{I - mx^2}\right] + 0.8185^{dt} v_{n-1} \\ r_n = dt\left[\frac{N_{n-1} - xF_{y,n-1}}{I - mx^2}\right] + 0.8^{dt} r_{n-1} \end{gathered} \tag{4}$$

gdzie u [m/s] jest prędkością w kierunku dzioba, v [m/s] prędkością w kierunku prawej burty, natomiast r [rad/s] jest prędkością kątową statku (prędkość skręcania). Parametr dt [s] opisuje, z jaką częstotliwością próbkowany jest cały układ.

Do testów przyjęto dt=0.01s. Ponadto I oznacza wymieniony wcześniej moment bezwładności a x jest odległością płetwy sterowniczej od środka ciężkości statku; m jest masą.

Osobnego opisu wymagają siły zewnętrzne działające na statek. Wśród nich we wzorze występują siła F_x symbolizująca siłę pchającą statek do przodu, czyli siłę napędową śruby oraz siły związane z zakłóceniami. W stronę prostopadłą do siły F_x działa siła F_y zaczepiona na rufie statku. N symbolizuje moment siły działający na statek. Do równań różnicowych dodano również współczynniki tarcia uniezależnione od czasu próbkowania. Celem była minimalizacja pozycji statku względem kursu, który dla uproszczenia jest linią prostą, przebiegającą przez oś wzdłużną statku w momencie wystartowania automatycznego sterowania. Wobec powyższego, aby nie skupiać się na utrzymywaniu stałej prędkości u statku, dodano uproszczony sterownik typu P, który w zależności od uchybu wartości u od zadanej zwiększa bądź zmniejsza siłę F_x. Oczywiście jest to bardzo uproszczona wersja zakładająca idealną prędkość zmiany obrotów śruby statku. Założenie takie pozwalało skupić się na jednowymiarowym sterowaniu i jednocześnie łatwiej zobrazować działanie sterownika. Dodanie dodatkowego wymiaru sterowania zapewne nie stanowi większego problemu.

Osobny temat stanowią zakłócenia, które wpływają na statek. Są to, jak już wcześniej wymieniono wiatr oraz fale. Również idąc za [1] zarówno turbulencja wiatru (odchylenie od jego wartości średniej), jak i kształt fal realizowane są jako proces stochastyczny o z góry zadanej gęstości widmowej. W modelu procesy te generowane są poprzez odpowiednią filtrację szumu białego równaniami różnicowymi:

$$\begin{cases} x_{1,n} = \Theta_1 x_{1,n-1} + \Theta_2 x_{2,n-1} + k_t e_n + x_{1,n-1} \\ x_{2,n} = x_{1,n-1} + x_{2,n-1} \\ y_n = x_{1,n} \end{cases} \tag{5}$$

Gdzie odpowiednie współczynniki wynoszą

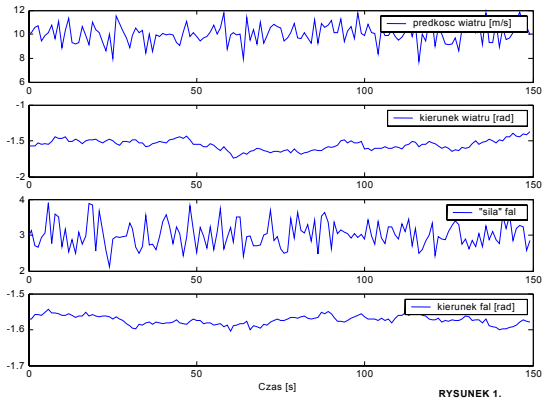
Dla wiatru:

$$k_t = 0.012032 \cdot V_t^{\frac{3}{2}}; \Theta_1 = -0.01208 \cdot V_t; \Theta_2 = -1.644 \cdot 10^{-5} \cdot V_t^2 \tag{6}$$

Dla fal:

$$k_t = 0.1057 \cdot h^{\frac{3}{4}}; \Theta_1 = -0.5586 \cdot h^{-\frac{1}{2}}; \Theta_2 = -1.5876 h^{-1} \qquad h = 0.015 \cdot V_t^2 + 1{,}5 \tag{7}$$

gdzie V_t oznacza średnią prędkość wiatru, a h wysokość fal, zależną od prędkości wiatru. e_n jest natomiast realizacją znormalizowanej zmiennej losowej. Oddziaływanie fal oraz wiatru na statek jest bardzo uproszczone w stosunku do prawdziwych sił zachodzących w rzeczywistości i uwzględnia między innymi wysokość statku (w naszym przypadku *10* m), co ma szczególny wpływ na siły spowodowane wiatrem. Zależą one głównie od azymutu, jaki akurat ma statek względem kierunku wiatru (powierzchnia, na którą działa wiatr), a ponadto nieliniowo zależą od siły wiatru. Co więcej, nie tylko siła wiatru i kształt fal zmieniają się w czasie. Również kierunek wiatru oraz kierunek oddziaływania fal zmienia się w czasie. Modelowane jest to po prostu jako dwa niezależne szumy białe o różnych wariancjach. Przykładowa realizacja w czasie wymienionych powyżej zakłóceń znajduję się na poniżej.



Rys. 1 Przykładowa realizacja zakłóceń

Na koniec warto wspomnieć również o tym, że płetwa sterowa jest sterowana pompą mającą inercyjność ze stałą czasową równą 1,5 sekundy, co dodatkowo wprowadza utrudnienie do sterowania statkiem.

MODELOWANIE OBIEKTU

Statek jest obiektem nieliniowym z zakłóceniami. Sieć neuronowa powinna być efektywnym zastępstwem, który dobrze reprezentuje model statku. Ze względu na wymóg możliwie szybkiego działania tego modułu wybrano prostą sieć neuronową z jedną warstwą ukrytą [3]. Neurony pierwszej warstwy opisane są funkcją nieliniową – *arcus tangens*, drugiej – wyjściowej – są opisane prostą funkcją liniową. Neurony obu warstw mają dodatkowe wejścia o stałej wartości *1* (tzw. *bias*). Jest to przykład standardowej i efektywnej sieci neuronowej.

Pierwszym zadaniem jest określenie wejść i wyjść z opisywanej sieci, oraz ilości neuronów w jej warstwie ukrytej. Do tego celu potrzebna była analiza przykładowych danych opisujących w wyczerpujący sposób zachowanie statku. Wstępna analiza wejść i wyjść pozwoliła określić początkowy ich zbiór.

Wartości mierzalne i dające nam opis stanu modelu to: sterowanie, prędkość do przodu i w bok, prędkość kątowa, azymut, siła i kierunek zakłóceń pochodzących od wiatru i fal. Jako wyjścia można uznać azymut, prędkość kątową, prędkości w przód i w tył.

Oczywiście zmienne te można również zastąpić zmianami ich wartości, co pozwoli nam zejść z uchybem do zera. Tak wybrane wejścia i wyjścia poddano analizie. Przykładowe pary wejścia-wyjścia sprawdzano w sposób, który pozwalał wyeliminować redundancję informacji wejściowych i odrzucenie danych, które nie miały wpływu na okręt. Otrzymany minimalny zbiór wejść to:

– Prędkość statku do przodu - u
– Zmiana prędkości statku w bok - dv_we
– Zmiana azymutu statku - dkat_we
– Kierunek wiatru - kw
– Siła wiatru - sw
– Aktualne sterowanie - ster_akt

Oraz wyjść:

– Zmiana azymutu statku - dkat_wy
– Zmiana prędkości statku w bok - dv_wy

Ilość neuronów w warstwie ukrytej ustalono na 6. Testy wypadły pomyślnie, to znaczy różnica wyjść między modelem fizycznym a siecią dla tych samych wejść był bardzo mały (10^{-6}).

Model powinien być w stanie liczyć stan statku na kilka chwil w przód – stan dla całego chromosomu. W tym celu trzeba było przyjąć pewne założenie. Uznano wiatr w kolejnych chwilach za stały, co może wprowadzać błąd do naszych ograniczeń. Jednak natura wiatru pozwala nam przewidzieć jedynie, że z pewnym prawdopodobieństwem nie zmieni się znacząco. Zamknięto sieć neuronową w pętlę. Za każdym razem na wejście sieci neuronowej podawano stan statku i sterowanie z chromosomu. Wyjście sieci i poprzedni stan służą do stworzenia kolejnego stanu, który znów podajemy na wejście sieci. Taka pętla daje na wyjściu odpowiedni wektor wyjść sieci, które łatwo da się przekształcić na wektor odchyleń statku od zadanej trajektorii. Dzięki temu algorytmowi otrzymano moduł, który pełni rolę modelu okrętu i przyjmuje na wejście cały chromosom sterowań.

Parametry algorytmu dobrano empirycznie, sprawdzając kolejne możliwości i obserwując wyniki symulacji. Ostatecznie najlepsze wyniki uzyskano obierając liczność populacji równą *20* osobników, podobnie jak długość chromosomów. Jednocześnie ustalając długość chromosomu na *20* genów, otrzymano horyzont predykcji sterownika. Pierwszy gen z najlepszego osobnika jest brany w każdym okresie próbkowania jako sterowanie dla statku.

Na koniec należało określić funkcję przystosowania dla populacji. Celem sterowania była minimalizacja odległości od założonego w momencie włączenia sterownika kursu. Trzymając się tradycji związanej z algorytmami genetycznymi funkcja przystosowania jest funkcją maksymalizowaną, toteż należało przemnożyć błąd wychylenia przez –1. Funkcja przystosowania wyglądała tak:

$$F(X) = -\sum_{i=1}^{20} g_i(w)^2 \tag{8}$$

gdzie funkcja g_i zwraca położenie statku w osi x układu współrzędnych kartezjańskich w chwili *i-tej* od początku predykcji. Funkcja g jest oczywiście funkcją nieliniową zależną od wektora wejść dla sieci neuronowej i jest realizacją tejże sieci. Sam wektor zmienia się w kolejnych chwilach w zależności od tego, co jest na wyjściu sieci neuronowej w danej chwili, jak było to opisane w poprzednim punkcie. Oczywiście, im bliżej statek znajduje się wyznaczonego kursu, tym większa wartość funkcji przystosowania.

Opisany schemat optymalizacji wypróbowany na stworzonym przez modelu neuronowym okazał się działać dość dobrze uwzględniając jedynie minimalizację odległości od wyznaczonego kursu. Uchyb ustalony był rzędu centymetrów, co wydaje się być wynikiem bardzo dobrym, jak na tak duży statek.

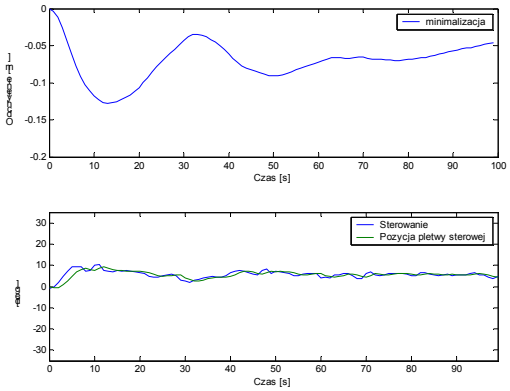
Zauważono że w momencie kiedy statek znajduje się bardzo blisko wyznaczonego kursu, sterowania wystawiane przez sterownik stają się bardzo chaotyczne i bardzo często mają skrajne, przeciwne wartości. Jest to niedobre ze względów praktycznych, gdzie niedopuszczalnymi byłyby praktycznie bezcelowe zmiany sterowania. Spowodowane jest to natomiast tym, że tak skonstruowana funkcja przystosowania zmusza sterownik do jak najszybszej zmiany kursu, by zminimalizować jak najszybciej odległość od kursu. Konieczna stała się, więc zmiana funkcji przystosowania tak, by uwzględniała ograniczenie związane z częstością zmian sterowania przy dostatecznie małym odchyleniu od kursu. Postanowiono przeformułować funkcję przystosowania na:

$$F(X) = -\sum_{i=1}^{20} g_i(w)^2 \left(\frac{i^2}{10} + 1\right) + \frac{0.00001}{g_i(w)^2} \cdot x_i^2 \tag{9}$$

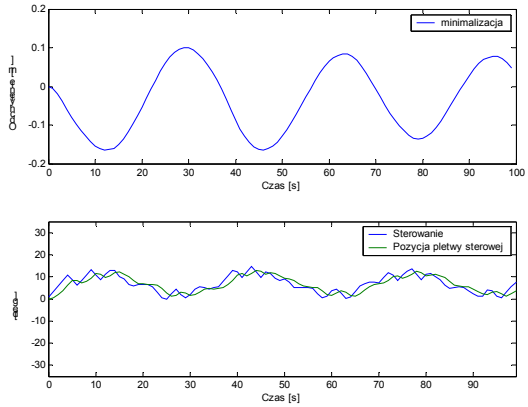
Gdzie x_i to wartość kolejnego genu z chromosomu pierwszego danego osobnika. Jak widać ze wzoru, zmiana dotyczyła zarówno wag, które przykładane są dla kolejnych chwil predykcji (najważniejsza jest chwila ostatnia) oraz wagi zależnej od odległości od linii kursu przyłożonej dla sterowań zawartych w chromosomie. Współczynniki dobraliśmy tak, by zmiany sterowania były jak najmniejsze, szczególnie przy małej odległości od linii kursu, przy zachowaniu jak najlepszej

minimalizacji tejże odległości. Znając ponadto naturę sterowania, uznaliśmy, że najlepszym inicjowaniem populacji bazowej jest inicjalizacja nie rozkładem jednostajnym na całym obszarze dopuszczalnym, a inicjalizacja rozkładem normalnym o odchyleniu standardowym równym 0.5. Sprawia to, że na starcie osobniki znajdują się w miejscu bardziej prawdopodobnym, szczególnie w przypadku gdy statek znajduje się blisko wyznaczonej linii kursu.

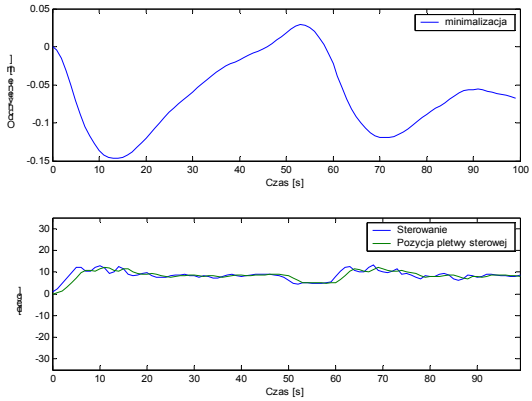
Poniżej znajdują się graficzne wyniki symulacji dla różnych kierunków wiatru i różnych prędkości zadanych statku.



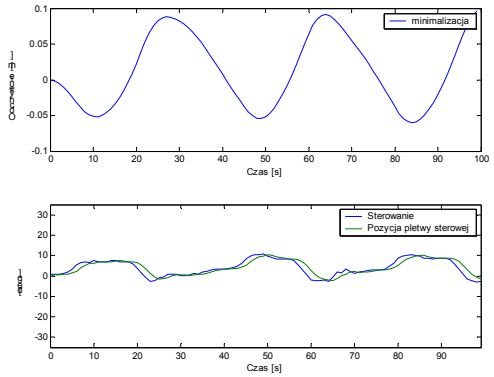
Rys. 2 Prędkość wiatru Vt=15 [m/s], prędkość u=7 [m/s], kierunek wiatru: -90°



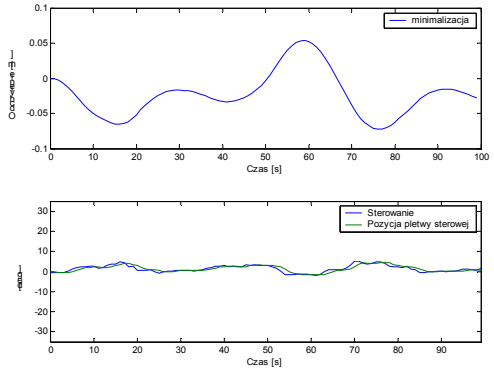
Rys. 3 Prędkość wiatru Vt=23 [m/s], prędkość u=10 [m/s], kierunek wiatru: -90°



Rys. 4 Prędkość wiatru Vt=25 [m/s], prędkość u=7 [m/s], kierunek wiatru: -45°



Rys. 5 Prędkość wiatru Vt=10 [m/s], prędkość u=5 [m/s], kierunek wiatru: -45°



Rys. 6 Prędkość wiatru Vt=5 [m/s], prędkość u=7 [m/s], kierunek wiatru: -135°

Powyższe *100* sekundowe symulacje pokazują, że sterownik zachowuje się dobrze w różnych sytuacjach, choć nie zawsze tak samo. Można zauważyć, że przy zbyt małej lub przy zbyt dużej prędkości statku, kurs statku jest oscylacyjny. Nie jest to jednak wielka wada, gdyż oscylacje te są gasnące, a co najważniejsze nie są dużych rozmiarów. Odległość rzędu kilku, nawet kilkunastu centymetrów nie robi przecież większej różnicy dla statku o szerokości *10* metrów. Najlepsze wyniki uzyskiwaliśmy dla prędkości *7 m/s*, gdzie zarówno minimalizacja odległości od linii kursu była bardzo dobra, jak również ruchy płetwy nie były ani gwałtowne, ani znaczne w przeciągu całego czasu testowego.

Podsumowując, zaproponowano, zrealizowano oraz przetestowano sterownik predykcyjny z powtarzaną optymalizacją i ograniczonym horyzontem predykcji. Sterownik ten został przetestowany na komputerowym modelu statku-tankowca. Nowością, która została wprowadzona do klasycznego modelu sterownika predykcyjnego, była optymalizacja sterowań na horyzoncie przy pomocy algorytmu genetycznego. Uzyskane wyniki zdają się być obiecujące i dają podstawy do poszukiwania dalszych zastosowań algorytmów genetycznych w dziedzinie sterowników predykcyjnych.

LITERATURA

1. J. Arabas, *Wykłady z algorytmów ewolucyjnych*, WNT, Warszawa, 2001.
2. M. Rostgaard, N. K. Poulsen, B. G. Thygesen, O. Ravn: *Stochastic modelling and adaptive LQG control of a ship*. Proc. 12th IFAC World Conference, Sydney, Australia, July 1993.
3. J. Hertz, R. Krogh, A. Palmer: *Introduction to the theory of neural computing*, Addison Wesley, 1992

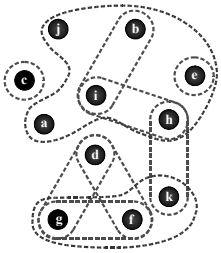
EWOLUCYJNE UCZENIE LOGICZNEJ SIECI ROZMYTEJ Z REDUKCJĄ ARGUMENTÓW DANYCH TRENUJĄCYCH

PIOTR NYCZ, JAROSŁAW STAŃCZAK, KRZYSZTOF SĘP

Instytut Badań Systemowych Polskiej Akademii Nauk
ul. Newelska 6, 01-447 Warszawa, Polska e-mail: {nycz, stanczak; sep}@ibspan.waw.pl

REDUKCJA ZBIORU ARGUMENTÓW

Wiele funkcji, bądź zależności występujących w rzeczywistym świecie opisywana jest nadmiarową liczbą argumentów czyli atrybutów. Niestety znalezienie minimalnego zbioru argumentów jest problemem z klasy **NP**, oznacza to wykładniczą złożoność algorytmów dokładnych rozwiązujących ten problem. W klasie **NP**-complete każdy problem można, w wielomianowym czasie sprowadzić do innego problemu **NP**-zupełnego, dla którego istnieją dobrze opracowane algorytmy aproksymacyjne. Łatwo można pokazać algorytm o złożoności wielomianowej, sprowadzający problem redukcji argumentów do problemu minimalnej transwersali w hipergrafie.



Rys. 1 Hipergraf

Definicja 1. Hipergrafem nazywamy parę $H=(X, F)$, gdzie $X=\{x_1, \ldots , x_n\}$ jest skończonym, niepustym zbiorem wierzchołków, a $F=\{f_1, \ldots , f_m\}$ jest rodziną niepustych podzbiorów zbioru X, takich dla których spełniony jest następujący warunek:

$$\underset{k \le n}{\forall k \in \mathbf{N}} \; \underset{i \le m}{\exists i \in \mathbf{N}} \; \left(x_k \in X \;\wedge\; f_i \in F\right) \;\Rightarrow\; \left(x_k \in f_i\right)$$

Elementy zbioru F nazywamy hiperkrawędziami.

Definicja 2. Dowolny zbiór wierzchołków Tr takich, że:

$$Tr \subseteq X \;\wedge\; \underset{f_i \in F}{\forall} \; \underset{x_k \in Tr}{\exists} \; x_k \in f_i \,,$$

nazywamy Transwersalą lub bazą wierzchołkową hipergrafu $H = (X, F)$.

Definicja 3. Najmniejszą transwersalą T_M w hipergrafie nazywamy transwersalę spełniającą warunek:

$$T_M \subseteq Tr \;\wedge\; \underset{T \subseteq Tr}{\forall} Card\left(T_M\right) \;\le\; Card\left(T\right).$$

Do znajdowania minimalnej transwersali w hipergrafie zastosowaliśmy algorytm aproksymacyjny Lovásza–Johnsona-Chwatala. Algorytm aproksymacyjny Lovásza–Johnsona-Chwatala [2] działa w następujący sposób:

1. Na początku zbiór wierzchołków Tr mających tworzyć transwersalę jest zbiorem pustym.

2. Wybieramy wierzchołek pokrywający najwięcej hiperkrawędzi, usuwamy wierzchołek i pokrytą przez niego hiperkrawędź z hipergrafu.
3. Usunięty wierzchołek dodajemy do zbioru *Tr*.
4. Jeżeli zbiór pozostałych hiperkrawędzi w hipergrafie jest zbiorem pustym, zbiór *Tr* jest szukaną transwersalą, jeżeli nie wracamy do punktu 2.

Znaleziona transwersala jest jednocześnie szukanym reduktem zbioru argumentów. Złożoność algorytmu Lovásza–Johnsona-Chwatala jest wielomianowa, a jakość aproksymacji jest opisana poniższą zależnością:

$$t_{opt} \leq t_{app} \leq t_{opt} \bullet (1+lg_2 n)$$

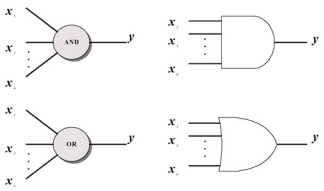
Gdzie n jest mocą zbioru wierzchołków, t_{opt} mocą najmniejszego zbioru wierzchołków pokrywających hipergraf, a t_a mocą zbioru wierzchołków uzyskanym w wyniku przedstawionej aproksymacji.

Zmniejszenie zbioru argumentów prowadzi niejednokrotnie do uproszczenia problemu [1], a czasami umożliwia rozwiązanie go w rozsądnym czasie. Ta idea przyświecała nam jako cel zastosowania redukcji argumentów w zbiorze danych trenujących. Redukcja argumentów danych trenujących oparta jest na aproksymacyjnym algorytmie znajdowania minimalnej transwersali w hipergrafie.

ROZMYTA SIEĆ LOGICZNA

Sieci neuronowe są obiektem dobrze znanym i niejednokrotnie stosowanym w praktyce. W klasycznym modelu neuronu wartość wyjścia **y** neuronu wylicza się ze wzoru $y = \sum_{i=1}^{n} z_i w_i + w_0$, gdzie $\boldsymbol{x_i}$ jest wartością wejścia o indeksie $\boldsymbol{i}$, $\boldsymbol{w_i}$ wartość wagi przyporządkowanej do tego wejścia $\boldsymbol{w_0}$ jest wartością stałą zwaną potocznie obciążeniem (ang. **bias**). Istnieją propozycje budowy sieci opartych o alternatywny model neuronu nazywany **neuronem logicznym**. Wartość wyjścia neuronu logicznego jest wyliczana za pomocą operacji **T-normy** lub **S-normy** znanych z teorii zbiorów rozmytych. Najczęściej stosowaną **T-normą** jest operacja **minimum**, a **S-normą** operacja **maksimum**. Rozpatrując neuron typu **AND** wartość wyjścia będziemy obliczać ze wzoru $y=min\{x_i, w_i\}$, natomiast dla neuronu **OR** $y=max\{x_i, w_i\}$ gdzie $x_i, w_i \in \{0,1\}$ gdzie $\boldsymbol{y}$ jest wyjściem sieci $\boldsymbol{x_i}$ wejście sieci o indeksie i oraz w_i - waga odpowiedniego wejścia neuronu. Szczegółowy opis zagadnienia przedstawiono w [3]. Rozmiar warstwy wejściowej oraz wyjściowej jest zależny od ilości zmiennych wejściowych oraz wyjściowych w zbiorze danych. Projektowanie rozmytej sieci logicznej (jak i neuronowej) sprowadza się do ustalenia ilości warstw ukrytych, rodzaju oraz liczby neuronów w poszczególnych warstwach oraz zdefiniowaniu połączeń pomiędzy poszczególnymi neuronami w warstwach. Najczęściej zakłada się że sygnał jest propagowany od warstwy wejsciowej do warstwy wyjściowej i nie dopuszcza się przekazywania sygnału z warstw późniejszych do wczesniejszych (siec typu **feedforward**). Oddzielną klasą są sieci **rekurencyjne** gdzie nie zakłada się wcześniej wspomnianego ograniczenia. Odpowiednie zaprojektowanie sieci zapewni poprawne jej działanie. Zbyt duża liczba neuronów w warstwach ukrytych i połączeń pomiędzy nimi będzie prowadzić do zjawiska przeuczenia się sieci (ang. **overloading**) czego konsekwencją będzie utrata zdolności uogólniania przetwarzanych danych oraz znacznie wydłużony proces uczenia. Zbyt mała ilość neuronów i połączeń nie pozwoli na dostatecznie dobre dopasowanie sieci do przetwarzanych danych i sieć nie "nauczy" się oczekiwanego zachowania. Projektując sieci logiczne (neuronowe) można korzystać z wiedzy zaczerpniętej z doświadczenia - praktyki inżynierskiej lub korzystać z pomocy algorytmu genetycznego. Podejście "inżynierskie" sprawdzi się w przypadku gdy projektant sieci zna chociaż częściowo charakter danych wejściowych oraz ukryte w nim zależności i na podstawie tej wiedzy może odpowiednio zaprojektować sieć. Jeżeli te wiadomości są niedostępne proces projektowania będzie przebiegał metodą prób i błędów aż do osiągnięcia pożądanych rezultatów. Jeżeli do poszukiwania rozwiązania

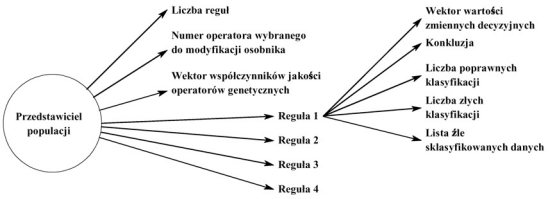
użyjemy algorytmu genetycznego proces projektowania sprowadzimy do losowego przeszukiwania przestrzeni rozwiązań co jest równoznaczne z metodą prób i błędów tyle że wykonywaną przez odpowiednio zaprogramowany komputer.



Rys. 2 Neurony i odpowiadające im bramki logiczne

EWOLUCYJNA METODA BUDOWY I UCZENIA ROZMYTEJ SIECI LOGICZNEJ.

W związku z tym, że nie istnieje efektywny algorytm tworzenia i uczenia rozmytych sieci logicznych, proponujemy użycie algorytmu ewolucyjnego do generowania optymalnej lub sub-optymalnej struktury oraz nauki rozmytej sieci logicznej. Osobnikiem populacji w tym podejściu jest cała sieć w której można modyfikować liczbę warstw jak i ilość węzłów (neuronów) w każdej warstwie. Dodatkowo modyfikacjom podlegają wagi połączeń pomiędzy węzłami (nauka sieci). W naszej metodzie zastosowaliśmy następujące operatory genetyczne: mutacja wag (losowe zaburzenie wybranych wag), krzyżowanie przystających osobników (wymiana warstw o tej samej liczbie neuronów pomiędzy osobnikami), ekspansja i redukcja liczby warstw i neuronów w warstwie (są to właściwie dwa osobne zestawy operatorów, z których jeden działa na warstwach, a drugi na neuronach), usunięcie połączeń o małych wagach (usuwane są połączenia o wagach mniejszych od pewnej wartości progowej – ma to prowadzić do uproszczenia struktury sieci).



Rys. 3 Struktura danych kodująca osobnika populacji

Funkcja celu dla każdego osobnika jest obliczana po przedstawieniu sieci (osobnikowi) zbioru danych trenujących jako suma kwadratów różnic między odpowiedzią otrzymaną, a oczekiwaną. Jest ona pomniejszana jest o pewną niewielką wartość zależną od liczby wag i neuronów w sieci w celu wymuszenia optymalizacji struktury sieci:

$$F = \sum_{i=1}^{i=N} \left(y_i - y_i^w \right)^2 - a * l^n - b * l^w$$

gdzie:
F – funkcja dopasowania;
N – liczba przykładów trenujących;
y_i, y_i^w – wyjście uzyskane od sieci i wyjście wzorcowe;
a, b niewielkie współczynniki funkcji kary;
l^n, l^w – odpowiednio liczb neuronów i wag w sieci.

LITERATURA

1. J. Stańczak, K. Sęp: *Hybrydowy algorytm pozyskiwania reguł klasyfikacyjnych*, BOS Warszawa 2002
2. Lovász: *On the ratio of optimal integral and fractional covers*, Budapest 1975.
3. A. Di Nola; W. Pedrycz; S. Sessa: *Fuzzy relational structures: State of Art*. Fuzzy Sets and Systems 75 1995

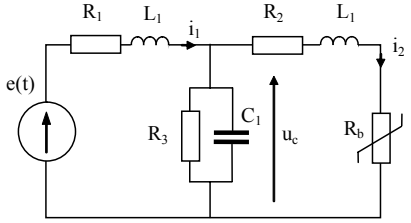
SYNTEZA PARAMETRÓW MODELI LABORATORIÓW ZWARCIOWYCH Z WYKORZYSTANIEM ALGORYTMÓW GENETYCZNYCH

TOMASZ SAMOTYJAK

Państwowa Wyższa Szkoła Zawodowa w Elblągu
ul. Grunwaldzka 137, t.samotyjak@pwsz.elblag.pl

Do symulacji stanów przejściowych użyto pakietu PSpice. Założono, że informacje dotyczące obwodu dostarczane są do momentu zapalenia się łuku elektrycznego we wkładce bezpiecznika (czas topienia wkładki). Oznacza to, że upraczając przyjęto R_b=10mΩ.

Na podstawie znanych właściwości wchodzących w skład układu elementów założono określoną strukturę schematu zastępczego (rys.1) i poszukiwano wartości elementów tej struktury [1].



Rys. 1 Przyjęty schemat zastępczy układu zwarciowego.

W prezentowanej pracy rozwiązanie tak sformuowanego zagadnienia syntezy potraktowano jako zagadnienie optymalizacji i zaproponowano jako sprowadzenie do poszukiwania minimum funkcji celu. Rozważa się zastosowanie dwóch typów funkcji celu, tzn. minimalizacja błedu poprzez porównanie przebiegów czasowych lub charakterystyk amplitudowo-fazowych [2, 3].

$$\mathrm{F} = \sum_{k=1}^{n} \left(W_z(t_k) - W_s(t_k)\right)^2 w_k \quad [1] \tag{1}$$

gdzie: F – funkcja celu, $Wz(t_k)$ –wielkość znana, $W_s(t_k)$ – wielkość otrzymana z syntezy, t_k – czas w k-tym pomiarze, w_k – waga dla k-tego pomiaru, n – liczba pomiarów,

$$\mathrm{F}(filtr) = \left(1 - w_{fm}\right)\sum_{k=1}^{n} w_k \left(F_m(\omega_k) - F_{mo}(\omega_k)\right)^2 + w_{fm} \sum_{k=1}^{n} w_k \left(F_f(\omega_k) - F_{fo}(\omega_k)\right)^2 \quad [3] \tag{2}$$

gdzie: $F_{mo}(\omega_k)$, $F_{fo}(\omega_k)$ – zadane charakterystyki częstotliwościowe, odpowiednio amplitudowa i fazowa, $F_m(\omega_k)$, $F_f(\omega_k)$ – charakterystyki częstotliwościowe ocenianego filtru, odpowiednio amplitudowa i fazowa, w_{fm} – współczynnik moduł-faza, określający udział charakterystyk amplitudowej i fazowej w wartości kryterium, w_k – wagi poszczególnych pulsacji.

Zbiór wartości maksymalnych elementów schematu zastępczego wyznaczono na podstawie znanego przebiegu napięcia i prądu zwarciowego stosując metodę zmiennych zespolonych.

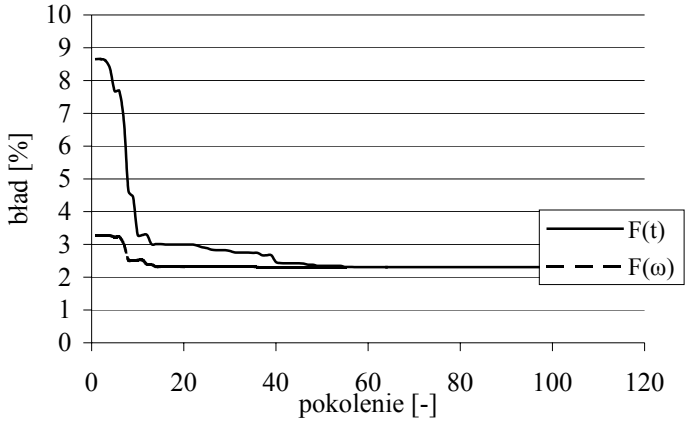
Propozycja parametrów algorytmu genetycznego:

1. Inicjacja populacji startowej metodą posiewu równomiernego w granicach określonych przez syntezę układu metodą zmiennych zespolonych,
2. Trzy podpopulacje – 10, 15 i 20 osobników w każdej – większa liczba osobników w populacjach eksploatacyjnych,
3. Zasięg mutacji odpowiednio dla każdej populacji 0,2, 0,05, 0,001. Podział populacji na eksploracyjne (pierwsza i druga) i eksploatacyjną (trzecia), rozważa się zastosowanie strojenia parametrów mutacji w celu odnajdywania bardziej zadawalających rozwiązań,

4. Nacisk selektywny 1,6 (możliwość strojenia nacisku metodą deterministyczną),
5. Metoda mutacji z adaptacją (derandomized self adaptation),
6. Selekcja w lokalnym sąsiedztwie,
7. Wymiana między podpopulacjami, co 10 generacji,
8. Zatrzymanie algorytmu po przekroczeniu maksymalnego czasu (najważniejsze kryterium) lub po przekroczeniu określonego błędu,
9. Archiwizacja wyników symulacji co jedna generacja (najlepsze jednostki w populacji, błąd minimalny, czas obliczeń, rozkład błędu w podpopulacji i wiele innych).

Do obliczeń zastosowano bibliotekę programu Matlab - GEATbx (Genetic and Evolutionary Algorithm Toolbox). Adres internetowy twórcy biblioteki to http://www.geatbx.com/.

Przykładowy przebieg poszukiwania rozwiązania dla wyżej wymienionych funkcji celu przedstawia rys. 2.



Rys. 2. Poszukiwanie rozwiązania dla obu funkcji celu.

Otrzymane rezultaty obliczeń genetycznych są bardzo zachęcające i skłaniają do podjęcia dalszych prac mających na celu minimalizacje czasu obliczeń oraz uzyskanie bardziej zadawalających rozwiązań (mniejszy błąd). Dla założonego schematu badanego układu zgodność prądów otrzymanych z syntezy i eksperymentów jest zadawalająca. Lepsza dla badanego układu okazała się druga funkcja celu F(ω). Jednak obie metody poszukiwań utknęły przy tym samym błędzie. W związku z tym rozważa się zastosowanie metod hybrydowych poszukiwań rozwiązania.

LITERATURA:

1. Jakubiuk K., Lipski T., Partyka R.: *H.F. Current Oscilations by Short-Circuit Test of Current-Limiting Devices*. Proc. 43 Int. Wissenschaftliches Kolloquim, Band 4, pp. 107-112. 21-24.09.1998. Ilmenau (Germany).
2. Czosnowski J.:*Zastosowanie programu genetycznego do automatycznego projektowania filtrów LC*.
3. Czosnowski J.: *Automatyczne projektowanie filtrów LC przy pomocy programu ewolucyjnego*.

ALGORYTM EWOLUCYJNY HARMONOGRAMOWANIA PROCESÓW OBLICZENIOWYCH W WIELOPROCESOROWYCH SYSTEMACH CZASU RZECZYWISTEGO

T. WÓJCIK, K. WALA
Katedra Automatyki, , Akademia Górniczo-Hutnicza
Al. Mickiewicza 30, 30-059 Kraków, e-mail: kwa@ia.agh.edu.pl

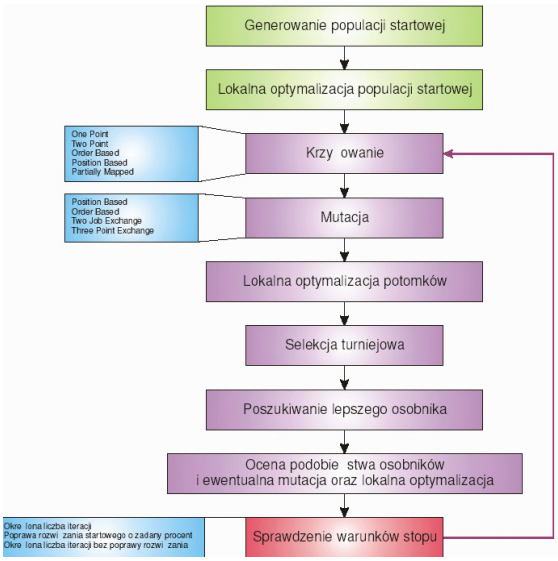
PROBLEM HARMONOGRAMOWANIA

Systemy czasu rzeczywistego są to systemy, których poprawność działania zależy nie tylko od wyników oblicze_, ale także od czasu w jakim te wyniki uzyskano, stąd zastosowania systemów czasu rzeczywistego wymuszają dostarczenie mocy obliczeniowej potrzebnej do wykonania zadanego zbioru zadań obliczeniowych. Jednym ze środków zwiększenia mocy obliczeniowej systemu komputerowego jest zastosowanie jednostek wieloprocesorowych, co umożliwia wykonywanie poszczególnych zadań przez większą liczbą procesorów, tj. wieloprocesorową realizacją zadań obliczeniowych. Efektywne zastosowanie techniki oblicze_ równoległych wymusza rozwiązywanie problemów harmonogramowania (szeregowania) zadań.

W pracy badane s_ zagadnienia statyczne harmonogramowania zbioru $J = \{1, 2, ..., j, ..., n\}$ zadań niezależnych i niewywłaszczalnych, gdzie J jest zbiorem indeksów zadań. Zakłada się, że system komputerowy składa się ze zbioru $M = \{ 1, 2, ..., i, ..., m \}$ niejednorodnych (niezależnych) procesorów i do wykonania zadania j O J konieczny jest zbiór Mj, Mj A M, równolegle pracujących dedykowanych procesorów przez czas pj wykonywania zadania, który nie podlega przerwaniu. Celem jest uszeregowanie p = (p(1), p(2), ..., p(n)) (p jest permutacją_ numerów zadań) zadań tak, ażeby żadne dwa zadania nie korzystały jednocześnie ze wspólnej maszyny oraz maksymalne opóźnienie zadań $Lmax = max\{ Lj: j$ O $J\}$ było minimalne, gdzie Cj – termin zakończenia zadania j, $Lj = Cj – dj$ – opóźnienie zadania j, dj – planowany termin zakończenia (linia krytyczna) zadania j .

ZASTOSOWANY ALGORYTM EWOLUCYJNY

Jako algorytm optymalizacji harmonogramów zastosowano wariant algorytmu ewolucyjnego nazywany algorytmem memetycznym, [1], będący heurystyka przeszukiwania dla kombinatorycznych problemów optymalizacji, bazujący na populacji i optymalizacji lokalnej (por. Rys. 1).



Rys. 1 Zastosowany algorytm ewolucyjny.

PRZYKŁAD NAPISANEJ APLIKACJI – JEJ MOŻLIWOŚCI ORAZ OSIĘGNIĘTE REZULTATY:

W celu przetestowania zastosowanego algorytmu genetycznego została napisana specjalna aplikacja. Umożliwia ona między innymi:

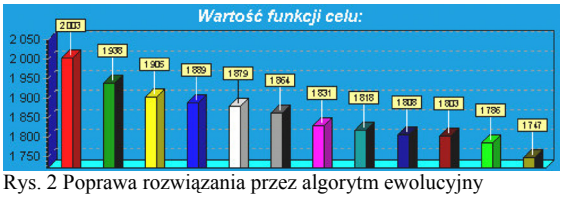
- wprowadzanie danych wejściowych ręcznie, z pliku oraz losowe ich generowanie,
- wybór jednego z trzech wariantów algorytmu konstrukcyjnego,
- ustawianie wielu parametrów w algorytmie ewolucyjnym (rodzaj ziarna, liczność populacji bazowej, liczność potomków, rodzaje i procentowy udział w procesie operatorów krzyżowania oraz operatorów mutacji, próg podobieństwa osobników w populacji, a także kryteria stopu – liczbą wszystkich iteracji, liczbą iteracji bez poprawy rozwiązania, poprawa rozwiązania o zadany procent),
- wielokrotne startowanie algorytmu ewolucyjnego z tym samym zestawem danych, lecz o różnych parametrach, co umożliwia porównywanie wyników,
- podgląd najważniejszych danych dotyczących stanu oblicze_ w trakcie działania programu,
- znalezione przez algorytm ewolucyjny rozwiązanie jest prezentowane w czytelnej postaci ciągu oraz wykresu Gantt'a.

Algorytm ewolucyjny został poddany serii prób. Każdorazowo uzyskiwał poprawą rozwiązania startowego (wyznaczonego algorytmem konstrukcyjnym) powyżej 10%. _średnio poprawiał rozwiązanie konstrukcyjne o ponad 30%, a czasami nawet o ponad 50%.

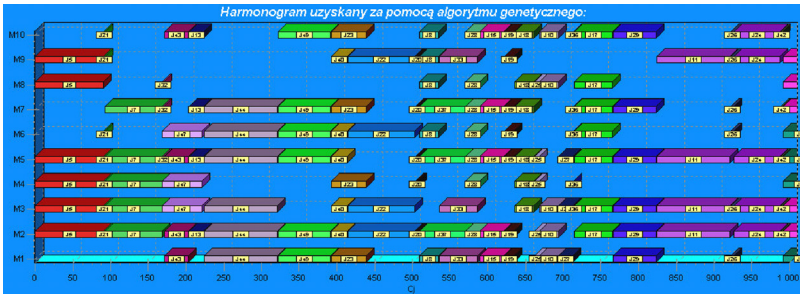
Szybkość działania algorytmu popraw jest zależna przede wszystkim od rozmiaru problemu (*n,m*) oraz od wielkości populacji bazowej i liczby potomków. W aplikacji zostały narzucone na dane sztuczne ograniczenia, które służą temu, aby możliwa była jakakolwiek wizualizacja i prezentacja wyników, a jednocześnie prędkość działania algorytmu była akceptowalna.

Same procedury i funkcje są uniwersalne, mogą działać z dowolnym zestawem danych, ograniczonym jedynie przez możliwości komputera.

Aplikacja jest dostępna pod adresem: http://student.uci.agh.edu.pl/~twojcik/projekty/badop/



Rys. 2 Poprawa rozwiązania przez algorytm ewolucyjny



Rys. 3 Znalezione przez algorytm ewolucyjny rozwiązanie

LITERATURA:

1. P. Merz, B. Freisleben, Fitness landscapes, memetic algorithms and greedy operators for graph *bi-partitioning,* Technical Report No. TR-98-01, University of Siegen, Germany.
2. M. Pawlak – Algorytmy ewolucyjne jako narządzie harmonogramowania produkcji, Wydawnictwo Naukowe PWN, Warszawa 1999.