48/2005

Raport Badawczy

Research Report

RB/16/2005

Wykorzystanie metod uczenia maszynowego do oceny jakości działania operatorów ewolucyjnych w zadaniu poszukiwania optimum w przypadku niestacjonarnego zadania optymalizacyjnego

J. Stańczak, K. Trojanowski

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Zbigniew Nahorski

Warszawa 2005

Wykorzystanie metod uczenia maszynowego do oceny jakości działania operatorów ewolucyjnych w zadaniu poszukiwania optimum w przypadku niestacjonarnego zadania optymalizacyjnego

Jarosław Stańczak
IBS PAN
stanczak@ibspan.waw.pl

Krzysztof Trojanowski
IPI PAN
trojanow@ipipan.waw.pl

Algorytm ewolucyjny (AE) jest narzędziem bardzo często wykorzystywanym do rozwiązywania różnego rodzaju zadań optymalizacyjnych. W szczególnym stopniu przydatność ich uwidacznia się w przypadkach zadań trudnych, gdzie inne metody nie dają zadawalających rezultatów. Optymalizacja zadań niestacjonarnych, o parametrach zmiennych w czasie jest jednym z takich przypadków. Jednakże, aby osiągnąć dobre rezultaty przy użyciu AE, należy odpowiednio skonstruować tę metodę. Kluczową częścią AE, decydującą w dużej mierze o jakości otrzymywanych rozwiązań i szybkości działania są operatory ewolucyjne i zastosowana metoda wyboru operatorów do modyfikacji posiadanych rozwiązań. W pracy tej rozpatrywany jest nowy sposób oceny jakości działania operatorów, oparty na regułach zapożyczonych z metod uczenia maszynowego. Ocena jakości działania operatora staje się następnie podstawą do wyboru operatorów modyfikujących populację rozwiązań. Opracowana metoda została z sukcesem przetestowana w środowisku optymalizacji dynamicznej i porównana z metodą wcześniejszą oraz wersją ze stałymi prawdopodobieństwami.

1. Wprowadzenie

Optymalizacja dynamiczna jest dość trudnym zadaniem obliczeniowym. Wymaga ona nie tylko znalezienia optimum (lub suboptimum) funkcji, ale jeszcze nadążania za nim w czasie, gdyż optymalizowana funkcja zmienia swój kształt, a co za tym idzie i położenie optimów w przestrzeni. Zastosowanie AE do rozwiązania tego problemu wymaga położenia nacisku na przyspieszenie jego działania i zwiększenie jego zdolności adaptacyjnych. Metoda z adaptacyjnym doborem operatorów ewolucyjnych wydaje się być dobrym sposobem na polepszenie parametrów AE.

W typowym algorytmie ewolucyjnym stosuje się operatory ewolucyjne ze sztywno ustalonymi prawdopodobieństwami wyboru operatorów ewolucyjnych, modyfikujących populację rozwiązań. Dla stosowanych powszechnie dwóch operatorów: krzyżowania i mutacji stosowano duże, bliskie 1 prawdopodobieństwo wykonania krzyżowania i niewielkie, rzędu 0,1 prawdopodobieństwo mutacji. Wiązało się to ze sposobem działania tych operatorów. Krzyżowanie, operator o charakterze eksploatacyjnym, tworzący potomka będącego rekombinacją cech rodziców powinien działać często, aby prowadził do tworzenia lepszych rozwiązań. Mutacja, operator o charakterze eksploracyjnym, wprowadzający nowe cechy do osobnika potomnego, powinien być stosowany z umiarem, aby nie zniszczyć tego, co znaleziono dotychczas.

Obecnie stosuje się AE o znacznie wyższym poziomie komplikacji, z kodowaniem rozwiązań i operatorami (często heurystycznymi) dostosowanymi do konkretnych potrzeb zadania. Bardzo trudno jest w takim przypadku ocenić, jaki charakter ma dany operator i jakie prawdopodobieństwo jego występowania będzie najlepsze zwłaszcza, że dająca najlepsze efekty częstość jego występowania w trakcie obliczeń ewolucyjnych może się zmieniać. Dlatego też zastosowanie metody w której stosuje się adaptację tych prawdopodobieństw wydaje się bardzo korzystnym rozwiązaniem.

2. Optymalizacja dynamiczna

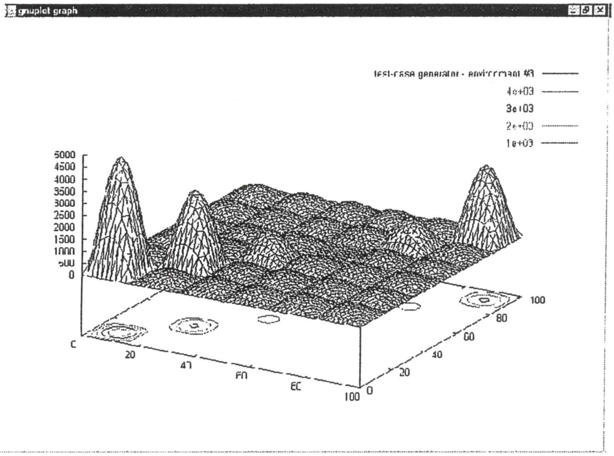
Jak to już zostało zaznaczone we wstępie do niniejszej pracy, optymalizacja obiektów (funkcji) niestacjonarnych, czyli takich, których parametry są zmienne w czasie, jest znacznie trudniejszym przypadkiem niż najczęściej spotykana optymalizacja statyczna. W optymalizacji statycznej należy znaleźć jak najlepsze rozwiązanie i jest na to dowolnie dużo (lub dość dużo) czasu. Ponadto większość wyuczonych lub zauważonych prawidłowości i informacji o optymalizowanym obiekcie nie zmienia się. W przypadku optymalizacji dynamicznej ramy czasowe są przeważnie ograniczone, gdyż optimum zmienia swoje położenie. Dodatkowo często żąda się, aby metoda obliczeniowa nadążała za zmieniającym się położeniem optimum. Niestety w przypadku dynamicznym informacje o optymalizowanym procesie, pozyskane w trakcie poszukiwania optimum mogą się bardzo szybko dezaktualizować i ich wykorzystanie może prowadzić do złych wyników.

W zagadnieniu optymalizacji obiektów niestacjonarnych można wyróżnić przypadki ze zmiennością cykliczną i ze zmiennością przypadkową. W pierwszym przypadku dobre wyniki rozwiązań można uzyskać stosując metody oparte na pamięci zdarzeń przeszłych, czyli nawiązujące w pewnym sensie do metod optymalizacji statycznej. W drugim przypadku metody takie będą raczej zawodne i istotne jest zwiększenie elastyczności w poszukiwaniu nowych rozwiązań.

Te trudności w rozwiązywaniu zadań związanych z optymalizacją dynamiczną prowadzą do częstego stosowania w nich metod opartych na heurystykach, takich jak symulowane wyżarzanie, czy algorytmy ewolucyjne. Te ostatnie wydają się dysponować szczególnie silnym potencjałem do rozwiązywania takich zadań.

3. Zadanie testowe

Do symulacji komputerowych wykorzystane zostało środowisko zaproponowane przez Trojanowskiego i Michalewicza [9].



Rys. 3.1. Środowisko testowe

Definiuje ono wielowymiarową przestrzeń, podzieloną na wiele hipersześcianów z zawartymi w nich unimodalnymi i paraboloidalnymi funkcjami, wśród których poszukuje się maksimum globalnego. Funkcje z hipersześcianów różnią się wartościami zmiennych w czasie maksimów lokalnych. Zmiany zachodzą cyklicznie. W danej chwili istnieje tylko jedno optimum globalne, w jednym z hipersześcianów. Optima lokalne nie przesuwają się w przestrzeni, a jedynie zmieniają swoją wartość, przybierając większe lub mniejsze wartości. Dla przypadku trójwymiarowego otrzymujemy krajobraz przypominający szachownicę z „górką" o maksimum lokalnym wypadającym w środku każdego pola szachownicy. Wysokości wzniesień są różne i cyklicznie się zmieniają.

4. Algorytm ewolucyjny wykorzystany do rozwiązania zagadnienia

4.1 Zakodowanie rozwiązania i funkcja dopasowania osobnika

Rozwiązywane zadanie jest typowym przykładem optymalizacji funkcji ciągłej w R^n. Osobnik populacji jest więc wektorem liczb rzeczywistych o długości n. Dodatkowo zawiera pewne elementy, wymagane przez zmodyfikowany AE użyty do rozwiązania problemu. Tymi dodatkowymi elementami są:

- wektor liczb rzeczywistych o długości równej liczbie zastosowanych operatorów genetycznych, zawiera on współczynniki jakości operatorów, potrzebne do wyboru operatora (więcej szczegółów o tej metodzie znajduje się w punkcie 4.3);
- liczba całkowita – numer aktualnie wybranego operatora genetycznego do modyfikacji rozwiązania;
- flaga (true/false) – identyfikująca, czy dany osobnik był już modyfikowany w danej iteracji algorytmu.

Funkcja dopasowania oparta jest na opisanym wcześniej zadaniu optymalizacji dynamicznej i wprowadza przeskalowanie problemu do zakresu (0,1).

4.2 Operatory ewolucyjne zastosowane do rozwiązania problemu

Efektywność metod opartych na AE polega w dużej mierze na możliwości doboru operatorów ewolucyjnych, dopasowanych do specyfiki rozwiązywanego zadania. W opisywanym przypadku zastosowano następujące operatory:

- mutacja I – wstawienie na wybranej pozycji wektora rozwiązania wylosowanej liczby z zakresu możliwych do przyjęcia wartości w danym wymiarze;
- mutacja II – niewielka perturbacja wartości wektora rozwiązań na wylosowanej pozycji o wartość z przedziału (-5% - +5%);

- mutacja III – działa podobnie jak mutacja I, lecz obowiązkowo dla wszystkich współrzędnych wektora;
- mutacja IV – działa podobnie jak mutacja II, lecz obowiązkowo dla wszystkich współrzędnych wektora;
- krzyżowanie I – wymienia przecięte w wylosowanym miejscu fragmenty wektorów z rozwiązaniami dwóch osobników populacji;
- krzyżowanie II – na wylosowanych pozycjach wektora rozwiązań przeprowadza uśrednienie tych wartości, pochodzących od obu osobników.

4.3 Algorytm ewolucyjny z oceną operatorów

Obecność wielu operatorów genetycznych spowodowała konieczność ustalenia prawdopodobieństw ich występowania. Zastosowano tu metodę doboru, bazującą na właściwościach operatora, opisaną w pracach [6], [7] i [8] z dwiema opisanymi niżej metodami oceny operatora.

Ogólna zasada działania metody jest następująca: im bardziej dany operator polepsza funkcję celu danego osobnika, tym większa staje się przypisana mu wartość współczynnika jakości, a co za tym idzie, większe jest prawdopodobieństwo jego występowania, co ilustruje wzór (4.1).

$$p_{ij}(t) = \frac{q_{ij}(t)}{\sum_{i=1}^{i=L(t)} q_{ij}(t)} \tag{4.1}$$

gdzie:

$q_{ij}(t)$ - współczynnik jakości i-tej operacji w chwili t dla j-tego osobnika populacji;

$p_{ij}(t)$ - prawdopodobieństwo wystąpienia i-tej operacji w chwili t dla j-tego osobnika;

$L(t)$ - liczba operacji genetycznych (może ona zależeć od czasu).

Każdy osobnik populacji posiada, oprócz zakodowanego rozwiązania, swój własny ranking operatorów, na podstawie którego losuje operator, który będzie go modyfikował w bieżącej iteracji. Metoda ta umożliwia adaptacyjne strojenie prawdopodobieństw występowania operatorów ewolucyjnych w zależności od potrzeb członków populacji rozwiązań i zgodnie z ich „upodobaniami". Jest bardzo prawdopodobne, że w zależności od fazy działania optymalizacji ewolucyjnej, jak i położenia danego osobnika w przestrzeni rozwiązań lepiej działają różne operatory.

4.3.1 Metoda oceny operatorów z zapominaniem osi ągnięć

Metoda ta została opisana w pracy [6] i [7]. Sposób działania tej metody przedstawia wzór (4.2). Zakłada ona, że w danej iteracji tylko jeden, wybrany operator modyfikuje rozwiązanie, dzięki czemu można mu przypisać nagrodę bądź karę za uzyskany efekt. Współczynniki jakości niewybranych osobników pozostają bez zmian (dolna część wzoru (4.2) dla *i≠l*), natomiast efekty zadziałania wybranego operatora są oceniane na podstawie górnej części wzoru (4.2) (dla *i=l*).

$$q_{ij}(t+1) = \begin{cases} q_{0ij}(t) + \dfrac{x_{ij}(t+1)}{\overline{x}(t)} + \alpha_{ij}(t) \cdot q_{ij}(t) \text{ dla } i = l \\ q_{ij}(t) \text{ dla } i \neq l \end{cases} \tag{4.2}$$

gdzie:

$q_{ij}(t)$ - wartości współczynnika jakości operacji *i* osobnika *j* w iteracjach *t*;

$q_{0ij}(t)$ - niewielka wartość stała (kredyt), uniemożliwiająca całkowite wyeliminowanie jakiegoś operatora (może zależeć od czasu);

$x_{ij}(t+1)$ - poprawa funkcji celu, $x_{ij}(t+1) = Q(t) - Q(t+1)_{ij}$, ($Q(t)$ - najlepsze rozwiązanie znalezione dotychczas, $Q(t+1)_{ij}$ - wartość funkcji celu bieżącego osobnika) osiągnięte przez operację *i* dla osobnika *j* w iteracji *t*, w przypadku braku polepszenia równa zero;

$\overline{x}(t)$ - średnia wartość popraw funkcji celu uzyskana dotychczas;

$a_{ij}(t)$ - współczynnik zapominania, uwydatniający rolę najświeższych danych, przy osłabianiu dawniejszych (może również zależeć od czasu);

l - indeks operatora wybranego w danej iteracji do modyfikacji rozwiązania.

Pierwszy człon wzoru (4.2) zapewnia utrzymywanie pewnej minimalnej wartości współczynnika jakości dla operatorów nieprzynoszących bezpośrednich korzyści. Mogą one jednak wprowadzać do populacji odpowiednie modyfikacje, które w kolejnych iteracjach zaowocują korzystnymi zmianami, dlatego też nie powinny być eliminowane przez wyzerowanie prawdopodobieństwa ich wystąpienia.

Drugi człon wzoru (4.2) stanowi poprawę funkcji celu rozwiązywanego zadania. W przypadku braku poprawy jest on równy zero.

Trzeci człon wzoru (4.2) przechowuje informację o dawniejszych osiągnięciach operatora, przemnożonych przez współczynnik zapominania $\alpha(t)$. Współczynnik zapominania jest odpowiedzialny za właściwe wyważenie wpływu nowszych i starszych osiągnięć na wartość współczynnika jakości operatora.

4.3.1 Nowa metoda oceny operatorów oparta na uczeniu maszynowym i algorytmie TD

W podejściu wzorowanym na uczeniu maszynowym i idei uczenia ze wzmocnieniem (ang. reinforcement learning) osobnik populacji traktowany jest jako pewien agent, wykonujący akcję polegającą na wywołaniu operatora ewolucyjnego. Operator wykonuje pewne akcje a_i, które powodują osiągnięcie przez agenta nowego stanu s_i (w tym przypadku jest to nowe rozwiązanie). Agent otrzymuje za akcje pewną nagrodę lub karę w zależności od efektu. Celem działania agenta jest wybór optymalnych akcji, dających optymalny zdyskontowany przychód (V^*) w długim (teoretycznie nieskończonym – od chwili t do nieskończoności) czasie:

$$V^{\pi} = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1}\right\} \tag{4.3}$$

Wzór (4.4) jest podstawą algorytmu TD(0), służącego do wyznaczania optymalnej strategii Π, umożliwiającej uzyskanie maksymalnego zdyskontowanego zysku:

$$V(s_{t+1}) = V(s_t) + \alpha\left[r_{t+1} + \gamma V^{*}(s_{t+1}) - V(s_t)\right] \tag{4.4}$$

gdzie:

$V(s_t)$- wartość współczynnika jakości (zdyskontowany zysk);

$V^*(s_{t+1})$ – estymata najlepszego współczynnika jakości, w symulacjach stosowana jest tu wartość współczynnika jakości osiągnięta przez najlepszy operator;

α - współczynnik nauki;

γ - współczynnik dyskontowania;

r_{t+1} - nagroda uzyskiwana za akcję;

t – bieżąca chwila;

We wzorze (4.4) przyjąłem notację stosowaną w opisywanej dziedzinie, jeśli porównać ją z zastosowaną we wzorze (4.2), to V jest odpowiednikiem q, a r to odpowiednik x.

4. 4 Metoda selekcji zastosowane do rozwiązania zadania - selekcja niszowa

Selekcja niszowa opracowana została specjalnie do rozwiązywania zadań optymalizacji dynamicznej. Opiera się ona na podziale populacji na kilka podpopulacji. Kryterium podziału są wartości funkcji dopasowania osobników. Na wstępie podobnie jak w populacji histogramowej tworzona jest lista osobników o różnych wartościach funkcji dopasowania w populacji[1]. Cały zakres występujących wartości tej funkcji jest dzielony na pewną liczbę przedziałów. Liczba ta jest jednym z parametrów metody, w symulacjach stosowano trzy popdpopulacje o następujących parametrach:

1. 90%-100% najlepszych wartości funkcji dopasowania;
2. 50%-90% najlepszych wartości funkcji dopasowania;
3. 10%-50% najlepszych wartości funkcji dopasowania.

---

[1] W wielu przypadkach może się zdarzyć, że równym wartościom funkcji dopasowania odpowiadają różne osobniki. W zastosowanych wersjach selekcji histogramowej i mieszanej taki przypadek jest uwzględniany.

Każdy przedział ma zagwarantowany pewien procent osobników w populacji potomnej. Oczywiście im gorsze wartości funkcji dopasowania, tym mniejszy powinien być zagwarantowany procent potomków w populacji potomnej. Rozkład procentowy liczby potomków dla każdego przedziału również można dostosować do wymogów problemu, a w symulacjach zastosowano rozkład. Pierwszy przedział otrzymuje 60% miejsc w populacji potomnej, drugi przedział 30% miejsc i trzeci przedział 10% miejsc. Suma wszystkich pul z przedziałów daje całą populację potomną. Wybór osobników które przechodzą do populacji potomnej w ramach jednego przedziału może być dokonywany na różne sposoby. W symulacjach komputerowych został zastosowany sposób wyboru najlepszych osobników w ramach przedziału.

5. Wyniki symulacji komputerowych

Przeprowadzono symulacje komputerowe dla zadania optymalizacyjnego z podziałem przeszukiwanego fragmentu przestrzeni trójwymiarowej na 36 (6x6) sześcianów z funkcjami typu paraboloidalnego o nieznanej i zmiennej w czasie wartości optimum lokalnego. Do poszukiwania maksimum globalnego funkcji użyto AE z trzema wersjami mechanizmu wyboru operatorów ewolucyjnych – dwiema opisanymi w punkcie 4.3 oraz dla porównania wersji bez oceny operatorów ze stałymi prawdopodobieństwami wyboru operatora i operatorami opisanymi w punkcie 4.2. Przetestowano zachowanie opisanych metod oceny operatorów w przypadku optymalizacji dynamicznej. Do porównania wyników zastosowano współczynnik jakości [10], zdefiniowany wzorem (5.1), będący miarą dostrojenia algorytmu do poszukiwanego optimum. Współczynnik ten ułatwia ocenę jakości metody rozwiązującej problem. W idealnym przypadku powinien wynosić 0, ale szczególnie dla chwil tuż po przełączeniu musi mieć on znacznie większą wartość, gdyż algorytm musi mieć trochę czasu na znalezienie nowego optimum.

$$A_d = \frac{1}{K}\sum_{i=1}^{K}(err_{i,\tau-1}) \qquad (5.1)$$

gdzie:

τ - liczba iteracji, pomiędzy dwiema kolejnymi zmianami funkcji celu;

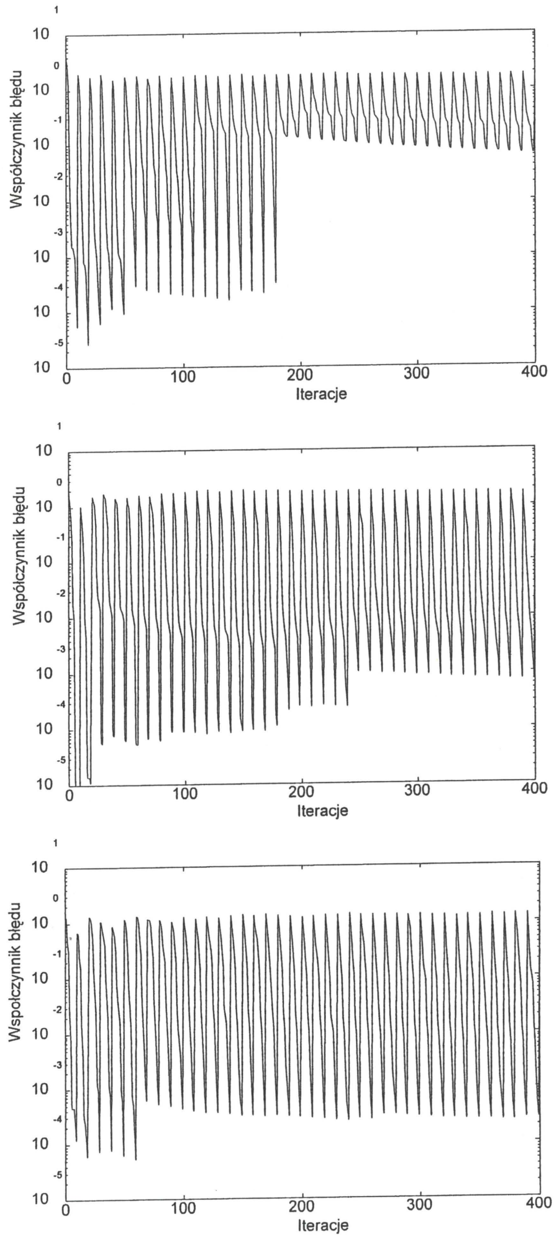
K – liczba zmian funkcji celu podczas symulacji;

$err_{i,\tau-1}$ – różnica między wartością najlepszego znalezionego rozwiązania, a wartością optymalną w ostatniej iteracji przed zmianą optymalizowanej funkcji.

We wszystkich symulacjach komputerowych wartość τ wynosiła 10, liczność populacji rodzicielskiej 60 osobników, potomnej 360 osobników, z których wybierano kolejną populację rodzicielską

Rys. 5.1 przedstawia zależność współczynnika A_d (5.1) dla obu badanych metod oceny operatorów oraz w dodanej dla porównania wersji ze stałymi prawdopodobieństwami wyboru operatorów w trakcie przebiegu 100 iteracji obliczeń ewolucyjnych. Widoczne jest, że metoda bez selekcji daje najgorsze rezultaty, gdy tymczasem obie metody z selekcją operatorów mają znacznie niższe wartości współczynnika błędu A_d.

Rys. 5.2 przedstawia wartość współczynnika błędu A_d tuż przed zmianą „wyglądu" funkcji celu. Pokazuje on stopień dostrojenia algorytmu do optimum po 10 iteracjach obliczeń. Dla obu metod z selekcją operatorów widoczne jest, że wartość współczynnika błędu A_d jest o 2 rzędy wielkości mniejsza niż dla metody bez selekcji opartej na osiągnięciach operatorów. Widać również, że metoda selekcji operatorów oparta na metodzie TD jest nieco lepsza, od metody z „zapominaniem" dawniejszych osiągnięć operatora. Wynika z tego, że obie metody wyboru operatora na podstawie generowanych przez niego zysków znacząco polepsza efektywność działania algorytmu ewolucyjnego, a ma to szczególne znaczenie w sytuacjach, gdzie liczy się szybkość obliczeń, czyli np. w przypadku optymalizacji dynamicznej

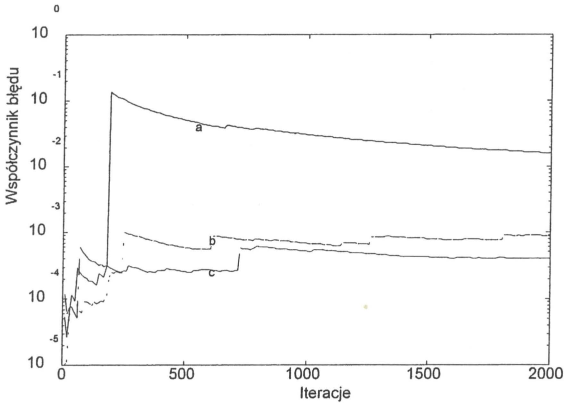


a)

b)

c)

Rys. 5.1. Wartość współczynnika błędu A_d w dla 100 początkowych iteracji optymalizacji: a – bez mechanizmu selekcji operatorów, b – selekcja metodą tradycyjną z „zapominaniem”, c - selekcja metodą TD.



Rys. 5.2. Wartość współczynnika błędu A_d w ostatniej iteracji przed zmianą funkcji celu: a – bez selekcji operatorów, b – selekcja metodą tradycyjną z „zapominaniem", c - selekcja metodą TD.

6. Wnioski

Celem pracy było sprawdzenie właściwości dwóch metod oceny operatorów w zagadnieniu optymalizacji dynamicznej i porównanie ich z podejściem tradycyjnym w którym stosowane są stałe prawdopodobieństwa wyboru operatorów. Przeprowadzone testy wykazały, że obie proponowane metody znacznie przewyższają wersję tradycyjną. Obie opisywane metody oceny operatorów działają bardzo dobrze z tym, że metoda oparta na algorytmie TD jest nieco prostsza w implementacji komputerowej i w związku z tym nieco szybsza.

Literatura

[1] Branke J.: *Memory Enhanced Evolutionary Algorithm for Changing Optimisation Problems*, CEC'99, IEEE Press, 1999.

[2] Cedeno W., Vemuri V. R.: *On the Use of Niching for Dynamic Landscapes*, ICEC'97, IEEE Publishing, Inc., 1997.

[3] Cobb H., Grefenstette J. J.: *Genetic Algorithms for Tracking Changing Environments*, V ICGA'93, Morgan Kauffman, 1993.

[4] Goldberg D. E., Smith R. E.: *Nonstationary Function Optimisation Using Genetic Algorithms with Dominance and Diploidy*, II ICGA, Lawrence Erlbaum Associates, 1987.

[5] Hadad B. S,. Eick C. F.: *Supporting Polyploidy in Genetic Algorithms Using Dominance Vectors*, EP'97, vol 1213 LNCS, Springer, 1997.

[6] Mulawka J. Stańczak J.: *Genetic Algorithms with Adaptive Probabilities of Operators Selection*, Proceedings of ICCIMA'99, New Delhi, India, 1999.

[7] Stańczak J.: *Rozwój koncepcji i algorytmów dla samodoskonalących się systemów ewolucyjnych*, PhD, Politechnika Warszawska, 1999.

[8] Stańczak J.: *Algorytm ewolucyjny z populacją "inteligentnych" osobników*, Materiały IV Krajowej Konferencji Algorytmy Ewolucyjne i Optymalizacja Globalna, Lądek Zdrój, 2000.

[9] Trojanowski K., Michalewicz Z.: *Searching for Optima in Non-Stationary Environments*, Proc. Of the 1999 Congress on Evolutionary Computation – CEC'99, IEEE Publ., 1999.

[10] Trojanowski K. Wierzchoń S. T.: *Studying Properties of Multipopulation Heuristic Approach to Non-Stationary Optimisation Tasks*, Proc. of IIPWM'03, 2003.