

50/2001 4051

Raport Badawczy
Research Report

RB/21/2001

**Algorytm ewolucyjny w problemie
sterowania systemu z zadaną trajektorią
optymalną o postaci zbioru rozmytego**

Jarosław Stańczak

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

Pracę zgłosił: Jarosław Stańczak

Warszawa 2001

Algorytm ewolucyjny w problemie sterowania systemem z zadaną trajektorią optymalną o postaci zbioru rozmytego

Jarosław Stańczak

Instytut Badań Systemowych PAN

ul. Newelska 6, Warszawa

stanczak@ibspan.waw.pl

W pracy niniejszej rozpatrzone problem sterowania obiektów: deterministycznego, stochastycznego i rozmytego z narzuconą optymalną trajektorią w postaci zbioru rozmytego i ustalonym horyzontem czasowym, podany po raz pierwszy przez Bellmana i Zadeha w pracy [1]. Sterowanie optymalne wyznaczane jest za pomocą algorytmu ewolucyjnego, kryterium jakości - zbiór stanów optymalnych w kolejnych iteracjach oraz ograniczenia na sterowania w kolejnych krokach są zadane w tym zagadnieniu jako funkcje przynależności do odpowiednich zbiorów rozmytych. Funkcje przejść pomiędzy kolejnymi stanami są zadane w postaci macierzy z odpowiednio: kolejnymi stanami (obiekt deterministyczny), prawdopodobieństwami przejść do kolejnych stanów (stochastyczny) i wartościami funkcji przynależności do kolejnych stanów (rozmyty).

1. Wstęp

Logika rozmyta (ang. *fuzzy logic*) jest uogólnieniem klasycznej logiki zero-jedynkowej dla zdań i reguł, których nie można z całą pewnością określić mianem prawdziwych lub fałszywych. Podobnie ma też ona zastosowanie w ocenie sytuacji i zjawisk, których nie można lub nie ma potrzeby oceniać w sposób ilościowy, a wystarczy metoda jakościowa, oparta przyporządkowaniu jednej z kilku możliwych ocen (np. mała,

średnia, duża, itp...). Logika rozmyta umożliwia wnioskowanie w oparciu o informację nieprecyzyjną lub przybliżoną. Ułatwia podejmowanie decyzji w sytuacjach związanych z nieokreślonością zarówno przesłanek, jak i reguł decyzyjnych. Nieokreśloność ta nie musi wynikać z niedokładności pomiarów lub też braku odpowiedniej wiedzy o problemie, ale np. ze specyfiki rozpatrywanego zagadnienia. Twórcą teoretycznych podstaw aparatu umożliwiającego przetwarzanie takich danych jest L. A. Zadeh, który w pracy [12] przedstawił teoretyczne podstawy dziedziny wiedzy, zwanej odąd logiką rozmytą. Od tego czasu jest to jeden z najszybciej rozwijających się działów sztucznej inteligencji. Wiele cennych informacji na ten temat można znaleźć np. w pracach [8], [2] i [3].

Jednym z najważniejszych zastosowań tej teorii jest automatyka i sterowanie, gdzie znalazła ona wiele zastosowań praktycznych: od sterowania ruchem pociągów i statków po aparaty fotograficzne i elektroniczny sprzęt powszechnego użytku.

W pracy niniejszej zostanie omówiony przypadek ewolucyjnego poszukiwania optymalnej sekwencji sterowań o pewnej zadanej długości (określony horyzont czasu) dla prostych układów: deterministycznego, stochastycznego i rozmytego o pewnych skończonych liczbach stanów i sterowań, których zachowanie w czasie określają pewne funkcje przejścia, w rozpatrywanym przykładzie będące macierzami przejść. Układ posiada pewne narzucone ograniczenia na sterowania dla każdego kroku działania, określone odpowiednimi funkcjami przynależności oraz również rozmyte kryteria jakości dla każdego kroku działania układu, oparte na jego stanie. Celem sterowania jest zapewnienie maksymalizacji w każdym kroku funkcji przynależności określającej kryterium jakości i ograniczenia na sterowania. Do rozwiązania tego zadania został wykorzystany specjalizowany algorytm ewolucyjny, którego opis, jak też i podsumowanie otrzymanych wyników znajdują się w kolejnych częściach tej pracy.

2. Sformułowanie rozwiązywanego problemu

2.1 Obiekt deterministyczny

Rozwiązanie postawionego problemu polega na znalezieniu skończonej (o ustalonej długości N) sekwencji sterowań $u(t)$ należących do skończonego i znanego zbioru sterowań dopuszczalnych U . Deterministyczny obiekt sterowany opisuje pewna funkcja przejścia:

$$x(t+1) = f(x(t), u(t)) \quad (2.1)$$

dana w rozpatrywanym przykładzie w postaci macierzy przejść pomiędzy stanami $x(t) \in X$ (X jest danym skończonym zbiorem możliwych stanów o liczności $|X|$). W zagadnieniu ustalony jest również początkowy stan układu $x(0)$. Na możliwe sterowania narzucone są w każdym kroku sterowania ograniczenia mające charakter funkcji przynależności $\mu_{C(t)}$ do określonego zbioru rozmytego sterowań dopuszczalnych dla danej iteracji $C(t)$. W każdym kroku sterowania oceniany jest również otrzymany stan układu na podstawie funkcji przynależności $\mu_{G(t)}$, gdzie $G(t)$ stanowi zbiór rozmyty stanów dopuszczalnych w danej iteracji. Ogólnie rozwiązanie problemu polega więc na maksymalizacji wyrażenia:

$$\mu_D(\hat{u}(0), \dots, \hat{u}(N-1) | x(0)) = \max_{u(0), \dots, u(N-1)} [\mu_{G(0)}(u(0)) \wedge \mu_{G(1)}(x(1)) \wedge \dots \wedge \mu_{G(N-1)}(u(N-1)) \wedge \mu_{G(N)}(x(N))] \quad (2.2)$$

gdzie:

$\hat{u}(0), \dots, \hat{u}(N-1)$ - sekwencja sterowań optymalnych;

$u(0) \dots u(N-1)$ - sterowania w kolejnych chwilach czasowych;

$\mu_D(\dots)$ - funkcja oceniająca jakość sterowania;

N - liczba kroków sterowania;

$\mu_{G(t)}(\dots)$ - funkcja przynależności do ograniczenia narzuconego na stan układu ($G(t)$);

$\mu_{C(t)}(\dots)$ - funkcja przynależności do ograniczenia narzuconego na sterowania ($C(t)$);

\wedge - znak oznaczający t-normę¹ (w rozpatrywanym przypadku - funkcja minimum).

Opisany wyżej problem może być rozwiązany wieloma metodami. Należą do nich między innymi: programowanie dynamiczne, metoda podziałów i ograniczeń, sieci neuronowe, *interpolative reasoning* oraz prosty algorytm genetyczny² [3], [4], [6] jednakże przy odpowiednio dużej liczbie kroków i sterowań stają się one zawodne z uwagi na ogromną przestrzeń poszukiwań. Liczbę możliwych rozwiązań w tym zagadnieniu można oszacować, przy ustalonym stanie startowym, na³:

$$n = |U|^N \quad (2.3)$$

gdzie:

$|U|$ - liczność zbioru sterowań;

N - horyzont czasu sterowania;

n - liczba możliwych rozwiązań;

W związku z tym przy dużych wartościach n przydatne stają się sposoby oparte na metodach sztucznej inteligencji. Dlatego też do rozwiązania tego problemu został użyty odpowiednio dostosowany algorytm ewolucyjny.

¹ t-norma i jej konorma, czyli s-norma są uogólnieniem operatorów logiki klasycznej, odpowiednio \wedge - and i \vee - or. W logice rozmytej są one definiowane na wiele sposobów, z których każde mają swoje wady i zalety [8]. W tej pracy przyjęto najprostszą i chyba najczęściej stosowaną wersję: \wedge odpowiada funkcji minimum(...), a \vee maksimum(...).

² Przez pojęcie prosty algorytm genetyczny rozumiana jest metoda z kodowaniem binarnym, prostą mutacją i krzyżowaniem i selekcją ruletkową, natomiast dla bardziej wyrafinowanych metod stosowana jest nazwa algorytm ewolucyjny.

³ Jest to wariacja z powtórzeniami.

2.2 Obiekt stochastyczny

W przypadku obiektu stochastycznego, funkcja przejścia bazuje na prawdopodobieństwach warunkowych osiągnięcia pewnego stanu w zależności od stanu poprzedniego i sterowania:

$$p(x(t+1)|x(t),u(t)) \quad t=0,1..N-1 \quad (2.4)$$

Tak zdefiniowany układ jest przykładem łańcucha Markowa.

Startując od pewnego początkowego rozkładu prawdopodobieństw $p(x(0))$ (stan układu jest wektorem prawdopodobieństw) w każdej iteracji na stan i sterowania obiektu nałożone są ograniczenia mające postać funkcji przynależności do zbiorów rozmytych odpowiednio $\mu_{G(t)}$ i $\mu_{C(t)}$. W tym przypadku kryterium jakości wyraża się w sposób następujący⁴:

$$\mu_D(\hat{u}(0), \dots, \hat{u}(N-1) | x(0)) = \max_{u(0), \dots, u(N-1)} [\mu_{C(0)}(u(0)) \wedge E\mu_{G(1)}(x(1)) \wedge \dots \wedge \mu_{C(N-1)}(u(N-1)) \wedge E\mu_{G(N)}(x(N))] \quad (2.5)$$

gdzie:

$E\mu_{G(t)}(\dots)$ - wartość oczekiwana funkcja przynależności do ograniczenia narzuconego na stan układu ($G(t)$), zdefiniowanego jako:

$$E\mu_{G(t)}(x(t)) = \sum_{i=1}^{i=|X|} p(x(t) | x(t-1), u(t-1)) \cdot \mu_{G(t)}(x(t)) \quad (2.6)$$

pozostałe oznaczenia jak we wzorach (2.2) i (2.4).

Metody rozwiązań tego problemu są podobne do tych, opisywanych w podrozdziale

2.1. Podobnie liczbę wszystkich możliwych rozwiązań można oszacować ze wzoru (2.3),

⁴ Jest to sformułowanie w wersji Bellmana i Zadeha [1].

gdyż zbiór dopuszczalnych i deterministycznych sterowań jest skończony. W niniejszej pracy zostanie zaproponowana metoda rozwiązania tego zadania specjalizowanym algorytmem ewolucyjnym.

2.3 Obiekt rozmyty

Przypadek obiektu opisywanego rozmytym równaniem stanu:

$$X(t+1)=F(X(t),U(t)) \quad (2.7)$$

gdzie:

$X(t), X(t+1)$ - rozmyte stany układu w chwilach t i $t+1$ należące do skończonego zbioru możliwych stanów rozmytych X ;

$U(t)$ - sterowanie (rozmyte lub nie) w chwili t , należące do skończonego zbioru możliwych sterowań U .

można opisać bardziej szczegółowo równaniem opartym na funkcjach przynależności stanów i sterowań (2.8) - sterowania deterministyczne, (2.9) - sterowania rozmyte [3]:

$$\mu_{X(t+1)}(x(t+1)) = \max_{x(t) \in X} \left(\mu_{X(t)}(x(t)) \wedge \mu_{X(t+1)}(x(t+1) | x(t), u(t)) \right) \quad (2.8)$$

$$\mu_{X(t+1)}(x(t+1)) = \max_{u(t) \in U} \left(\max_{x(t) \in X} \left(\mu_{X(t)}(x(t)) \wedge \mu_{X(t+1)}(x(t+1) | x(t), u(t)) \wedge \mu_{U(t)}(u(t)) \right) \right) \quad (2.9)$$

gdzie:

$\mu_{X(t+1)}(x(t+1)), \mu_{X(t)}(x(t))$ - funkcje przynależności stanów x układu do zbioru stanów X w chwilach t i $t+1$;

$\mu_{X(t+1)}(x(t+1) | x(t), u(t))$ - warunkowa funkcja przynależności opisująca przejścia od stanu $x(t)$ do stanu $x(t+1)$ pod wpływem sterowania $u(t)$, w rozpatrywanym przypadku macierz przejść między stanami zawierająca wartości funkcji przynależności.

Na sterowania układu oraz jego stan w każdym kroku działania narzucone są ograniczenia w postaci zbiorów rozmytych $\mu_{C(t)}$ (ograniczenie na sterowania) i $\mu_{G(t)}$ (trajektoria optymalna), podobnie jak miało to miejsce w przypadku układu deterministycznego i stochastycznego. W tym przypadku konieczne jest narzędzie umożliwiające porównywanie zbiorów rozmytych i obliczanie stopnia ich podobieństwa lub też jego braku:

$$\mu_{\bar{C}(t)}(U(t)) = 1 - d(U(t), C(t)) \text{ oraz } \mu_{\bar{G}(t+1)}(X(t+1)) = 1 - d(X(t+1), G(t+1)) \quad (2.10)$$

gdzie:

$d(\cdot)$ - oznacza odległość np.: w sensie Hamminga, euklidesową lub inną;

$C(t)$ - zbiór ograniczeń na sterowania w kroku t ;

$G(t+1)$ - zbiór ograniczeń na stan w kroku $t+1$;

$\mu_{\bar{C}(t)}(U(t))$ i $\mu_{\bar{G}(t+1)}(X(t+1))$ - miary podobieństwa zbiorów rozmytych.

Dla układu ze sterowaniem rozmytym kryterium jakości sterowania ma postać (2.11), dla sterowania deterministycznego (2.12):

$$\mu_D(\hat{U}(0), \dots, \hat{U}(N-1) | X(0)) = \max_{U(0), \dots, U(N-1)} \left[\mu_{\bar{C}(0)}(U(0)) \wedge \mu_{\bar{G}(1)}(X(1)) \wedge \dots \wedge \mu_{\bar{C}(N-1)}(U(N-1)) \wedge \mu_{\bar{G}(N)}(X(N)) \right] \quad (2.11)$$

$$\mu_D(\hat{u}(0), \dots, \hat{u}(N-1) | X(0)) = \max_{u(0), \dots, u(N-1)} \left[\mu_{C(0)}(u(0)) \wedge \mu_{\bar{G}(1)}(X(1)) \wedge \dots \wedge \mu_{C(N-1)}(u(N-1)) \wedge \mu_{\bar{G}(N)}(X(N)) \right] \quad (2.12)$$

gdzie:

$\hat{U}(0), \dots, \hat{U}(N-1)$ - sekwencja rozmytych sterowań optymalnych;

$U(0), \dots, U(N-1)$ - sterowania rozmyte w kolejnych chwilach czasowych;

$\hat{u}(0), \dots, \hat{u}(N-1)$ - sekwencja deterministycznych sterowań optymalnych;

$u(0), \dots, u(N-1)$ - sterowania deterministyczne w kolejnych chwilach czasowych;

$X(0), \dots, X(N)$ - stan układu w kolejnych chwilach czasowych;

$\mu_D(\cdot)$ - funkcja oceniająca jakość sterowania;

N - liczba kroków sterowania;

\wedge - znak oznaczający t-normę;

pozostałe oznaczenia jak we wzorze (2.10).

Podobnie jak w poprzednich przypadkach metody rozwiązań tego problemu obejmują: programowanie dynamiczne, metodę podziałów i ograniczeń, sieci neuronowe, *interpolative reasoning* a także prosty algorytm genetyczny. W przypadku deterministycznych sterowań liczba możliwych rozwiązań wyraża się wzorem (2.3). Natomiast w przypadku ze sterowaniem rozmytym przestrzeń rozwiązań jest ciągła i zawarte są one w hipersześcianie $[0,1]^{N \times |U|}$. Rozwiązania problemu poszukuje się w kostce o boku 1, lecz o bardzo dużej wymiarowości, więc zadanie to jest dość trudne.

3. Algorytm ewolucyjny w problemie optymalnego sterowania rozmytego

Mimo swej ogromnej uniwersalności do rozwiązania konkretnego zagadnienia algorytm ewolucyjny trzeba odpowiednio dostosować. Bez zmian pozostaje jedynie ogólna koncepcja działania metody: dzięki niewielkim zmianom genotypu, zachodzącym w kolejnych pokoleniach (iteracjach) i selekcji promującej najlepsze osobniki, następuje rozwój populacji w kierunku coraz lepszego spełniania funkcji celu (lub opartej na niej funkcji dopasowania) postawionego zagadnienia. Dostosowanie to polega na odpowiednim zakodowaniu problemu⁵ oraz doborze operatorów ewolucyjnych, umożliwiających

⁵ Stosowanie „na siłę”, przyjętego dawniej jako podstawa w algorytmach genetycznych, kodowania binarnego jest w tym przypadku niecelowe. Obecnie stosuje się tę metodę tylko tam, gdzie ułatwia ona przedstawienie problemu.

modyfikację populacji zakodowanych osobników. Kryterium stopu w AE jest zazwyczaj liczba iteracji, osiągnięcie pewnych zadowalających rezultatów lub też brak pozytywnych zmian w ciągu ustalonej liczby pokoleń.

1. Losowa inicjacja populacji rozwiązań.
2. Reprodukacja i modyfikacja rozwiązań przy użyciu operatorów genetycznych.
3. Ocena otrzymanych rozwiązań.
4. Selekcja do populacji potomnej.
5. Jeśli nie spełnione kryterium stopu, powrót do punktu 2.

Rys. 3.1. Schemat działania algorytmu ewolucyjnego.

3.1 Zakodowanie osobnika w AE

3.1.1 W przypadku sterowań deterministycznych

Z tym rodzajem kodowania wiążą się zadania sterowania obiektem deterministycznym, stochastyczny i rozmyty ze sterowaniem nierozmytym (deterministycznym). W takim przypadku rozwiązywanego zadania całą informację o konkretnym rozwiązaniu i jego jakości niesie sekwencja sterowań obiektu [3], przy założeniu stałego stanu początkowego. Tak więc genotypem pojedynczego osobnika może być sekwencja numerów, oznaczających sterowania ze zbioru sterowań (nie są to konkretne wartości sterowań, a jedynie ich indeksy ze zbioru sterowań).

3.1.2 W przypadku sterowania rozmytego

W przypadku sterowania rozmytego obiektem opisywany rozmytym równanie stanu (2.7) i (2.8), sterowaniem w każdym kroku działania układu jest funkcja przynależności oparta na zbiorze sterowań dopuszczalnych \mathcal{U} . Tak więc jeden sygnał sterujący jest wektorem wartości rzeczywistych o długości równej mocy zbioru $|\mathcal{U}|$ określających

wartości funkcji przynależności do kolejnych sterowań ze zbioru \mathcal{U} . Zatem cały przedstawiciel populacji rozwiązań algorytmu ewolucyjnego składa się z N takich wektorów (N - liczba kroków sterowania). Cały osobnik jest wobec tego macierzą liczb rzeczywistych z przedziału $\langle 0; 1 \rangle$, o rozmiarach $N \times |\mathcal{U}|$.

3.2 Funkcja dopasowania

We wszystkich rozpatrywanych przypadkach funkcja celu odpowiedniego zadania (2.2, 2.5, 2.11 lub 2.12) ma postać właściwą do tego, aby stać się bezpośrednio funkcją dopasowania dla populacji osobników algorytmu ewolucyjnego. Związane jest to z tym, że funkcja celu dla każdego z zadań przyjmuje wartości z zakresu $\langle 0; 1 \rangle$. W przypadku funkcji dopasowania istotne jest, aby wartości te były nieujemne (w przeciwnym wypadku zawodzi wiele metod selekcji osobników, np. stosowane w rozwiązaniu tego zagadnienia). Istotnym faktem jest też wzrost wartości funkcji dopasowania wraz ze wzrostem jakości osobnika, co również w tym przypadku jest zachowane.

3.3 Operatory genetyczne

Dla tak skonstruowanych osobników należało teraz zaproponować kilka operatorów ewolucyjnych, umożliwiających ich modyfikację w kolejnych pokoleniach. Zarówno dla przypadku kodowania całkowitoliczbowego, jak i rzeczywistoliczbowego, opracowanie prostych losowych operatorów nie nastrocza większych problemów i można tu zastosować standardowe operacje typu: mutacji, krzyżowania i inwersji. Symulacje komputerowe wykazały jednak, że taki zestaw operatorów dawał dość mierne wyniki. W badanych problemach, dla których N wynosiło odpowiednio 10 i 100 (i w niektórych przypadkach 1000 - dla tak dużej liczby kroków w przypadkach niedeterministycznych czas obliczeń był bardzo długi), jedynie w pierwszym przypadku i to nie w każdej symulacji (przy

losowym generowaniu populacji startowej) otrzymywano satysfakcjonujące rezultaty, w pozostałych nie zostało znalezione żadne rozwiązanie o wartości funkcji celu większej od 0. Sytuacja ta spowodowała konieczność znalezienia bardziej efektywnego zestawu operatorów, a taki mógł powstać jedynie przy zastosowaniu heurystyk. Testowano przy tym dwie metody: generowania populacji startowej w sposób nielosowy - przy użyciu prostej metody zachłannej i zastosowania ślepych operatorów oraz przy losowym generowaniu populacji startowych i zastosowaniu operatorów heurystycznych. To drugie rozwiązanie dawało zdecydowanie lepsze rezultaty. W tym celu powstały nowe operatory.

Oczywiście żaden z opisanych dalej operatorów heurystycznych nie daje gwarancji znalezienia rozwiązania o wartości globalnej funkcji dopasowania większej od poprzedniej, lecz daje znacznie większą szansę znalezienia niż operator losowy. W symulacjach stosowano po kilka wersji każdego z operatorów heurystycznych (dotyczy to w szczególności problemów z kodowaniem całkowitoliczbowym), różniących się sposobem wybierania kolejnych sterowań z listy możliwych w danym kroku (od wyboru najlepszego na danym etapie, przez wybór losowy z prawdopodobieństwem zależnym od jakości do zupełnie losowego). Zastosowanie tak dużej liczby operatorów spowodowało konieczność użycia mechanizmu odpowiedniego losowania operatorów w każdej iteracji algorytmu, o czym jest mowa w punkcie 3.4.

3.3.1 Przypadek całkowitoliczbowego kodowania rozwiązań

Zastosowano trzy operatory: krzyżowanie (wymianę losowo wybranych końcowych fragmentów osobników), mutację (zamianę jednego losowo wybranego sterowania na wylosowaną inną wartość) i inwersję (przestawienie kolejności sterowań w losowo wybranym podciągu osobnika). Oprócz tego wystąpiły te same operatory w wersji wielokrotnej (3 powtórzenia tego samego operatora). Przedstawione wyżej operatory

należą do klasy tzw. operatorów ślepych, a więc działających w pełni losowo, bez jakiegokolwiek wiedzy o problemie. Jak już to zostało wspomniane, wyniki doświadczalne [11] oraz zamieszczone w tej pracy, wykazały konieczność zastosowania operatorów heurystycznych, aby możliwe było efektywne przeprowadzanie obliczeń dla sekwencji sterowań dłuższych niż 10 kroków.

1. $i := I$;
2. Obliczenie wartości

$$\mu(i) := \mu_{C(i-1)}(u(i-1)) \wedge \mu_{G(i)}(x(i));$$
3. Jeśli $\mu(i) \leq \mu_D(u(0), \dots, u(N-1) | x(0))$
 - a) utworzenie listy dozwolonych par $(u(i-1), x(i))$, dla których

$$\mu(i) > \mu_D(u(0), \dots, u(N-1) | x(0));$$
 - b) wybór jednej z par (możliwe są różne kryteria wyboru) i modyfikacja zestawu sterowań osobnika na pozycji i (w razie braku możliwości poprawy nie następuje modyfikacja rozwiązania);
4. $i := i+1$;
5. Jeśli $i \leq N$ powrót do punktu 2;
6. Koniec

Rys. 3.2. Algorytm działania operatora „w przód”.

Pierwszy z nich przechodzący sekwencję sterowań od początkowego do końcowego (operator „w przód”) i próbujący korygować miejsca, gdzie wartość funkcji dopasowania była mniejsza lub równa wartości dopasowania całego osobnika. Jest to wykonywane przez sprawdzenie innych możliwych sterowań w miejsce poprzednio używanego. Tworzona jest w ten sposób lista sterowań z nowymi cząstkowymi wartościami funkcji celu, lepszymi od tego, które było dotychczas. Jeśli lista nie jest pusta, to spośród zapisanych tam możliwości wybierane jest nowe sterowanie. Używane są różne metody wyboru: najlepszego rozwiązania, losowa i losowa skalowana (im lepsze rozwiązanie, tym większe

prawdopodobieństwo wyboru). W obliczeniach zastosowano wszystkie trzy wersje operatora. Algorytm działania tego operatora przedstawiony jest na Rys. 3.2.

Drugi operator heurystyczny wykonywał zbliżone działania, lecz od końca do początku sekwencji sterowań. Korygowane sterowania wybierane były z wytwarzanej w każdym kroku listy sterowań i stanów dających większą od poprzedniej wartość funkcji dopasowania w danym kroku tak, aby tworzyły konsekwentną całość z następnymi krokami (czyli wybranymi w poprzednich krokach działania operatora). Dodatkową trudnością jest tu również konieczność „trafienia” w założony stan początkowy $x(0)$. Powoduje to sytuację, w której użycie tego operatora możliwe jest tylko w przypadku obiektu deterministycznego, w którym można dokładnie określić początkowy stan obiektu, a nie jedynie podać rozkład prawdopodobieństw czy funkcję przynależności do zbioru stanów. Algorytm działania operatora „do tyłu” przedstawiony jest na Rys. 3.3.

1. $i := N$;

2. Obliczenie wartości

$$\mu(i) := \mu_{C(i-1)}(u(i-1)) \wedge \mu_{G(i)}(x(i))$$

dla wszystkich możliwych par $(u(i-1), x(i))$;

3. Wybór jednej z par tak, aby:

$$\mu(i) > \mu_D(u(0), \dots, u(N-1) | x(0))$$

oraz wstawienie na i -tej pozycji osobnika wybranego numeru sterowania (w razie braku możliwości poprawy nie następuje modyfikacja rozwiązania);

4. Znalezienie w tablicy funkcji przejścia stanu $x(i-1)$, dla którego:

$$x(i) := f(x(i-1), u(i-1));$$

5. $i := i - 1$;

6. Jeśli $i > 0$ powrót do punktu 2;

7. Koniec.

Rys. 3.3. Algorytm działania operatora „do tyłu”.

3.3.2 Przypadek rzeczywistoliczbowego kodowania rozwiązań

Zakodowanie osobnika w postaci tablicy liczb rzeczywistych powoduje konieczność wprowadzenia zupełnie nowych operatorów innych niż w przypadku kodowania całkowitoliczbowego.

1. $i:=1$;

2. Obliczenie wartości:

$$\mu(i) = \mu_{\bar{c}(i-1)}(U(i-1)) \wedge \mu_{\bar{g}(i)}(X(i))$$

3. Jeśli $\mu(i) \leq \mu_D(U(0), \dots, U(N-1) | X(0))$ to:

a) jeśli $\mu_{\bar{c}(i-1)}(U(i-1)) \leq \mu_D(U(0), \dots, U(N-1) | X(0))$ to:

- $U_p(i-1) = C(i-1)$;
- $\mu_p(i) = \mu_{\bar{c}(i-1)}(U_p(i-1)) \wedge \mu_{\bar{g}(i)}(X_p(i))$;
- jeśli $\mu_p(i) > \mu(i)$ to $U(i-1) = U_p(i-1)$, $X(i) = X_p(i)$;

b) jeśli $\mu_{\bar{g}(i)}(X(i)) \leq \mu_D(U(0), \dots, U(N-1) | X(0))$ to:

- $U_p(i-1) = (|\mathcal{X}| * C(i-1) + |\mathcal{V}| * G(i)) / (|\mathcal{X}| + |\mathcal{V}|)$;
- $\mu_p(i) = \mu_{\bar{c}(i-1)}(U_p(i-1)) \wedge \mu_{\bar{g}(i)}(X_p(i))$;
- jeśli $\mu_p(i) > \mu(i)$ to $U(i-1) = U_p(i-1)$, $X(i) = X_p(i)$;

4. $i := i+1$;

5. Jeśli $i \leq N$ powrót do punktu 2;

6. Koniec.

Rys. 3.4. Algorytm działania operatora heurystycznego dla problemu ze sterowaniem rozmytym.

Część z nich to operatory losowe stosowane podobne do stosowanych w strategiach ewolucyjnych i algorytmach genetycznych do rozwiązywania zadań ciągłych:

- mutacja - losowa zmiana liczby z zakresu $\langle 0; 1 \rangle$ na inną z tego samego przedziału;
- przestawienie - wymiana miejscami wektorów sterowań w różnych krokach w obrębie tego samego osobnika;

- krzyżowanie - wymiana wylosowanych części osobników (wektorów sterowań) pomiędzy różnymi osobnikami;
- inwersja - odwrócenie kolejności wektorów sterowań w wylosowanym fragmencie osobnika.

Oprócz operatorów losowych zastosowano również operator heurystyczny o następującym sposobie działania: (Rys. 3.4).

3.4 Algorytm ewolucyjny z oceną operatorów

Obecność wielu operatorów genetycznych spowodowała konieczność ustalenia prawdopodobieństw ich występowania. W symulacjach zastosowano nową metodę doboru, bazującą na właściwościach operatora, opisaną w pracach [7], [9] i [10]. 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 (3.1).

$$p_{ij}(t) = \frac{q_{ij}(t)}{\sum_{i=1}^{L(t)} q_{ij}(t)} \quad (3.1)$$

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. Oczywiście operatory słabsze również mają pewne prawdopodobieństwo wystąpienia, gdyż zawsze utrzymywana jest dla nich pewna minimalna wartość współczynnika jakości, dzięki występowaniu we wzorze (3.2) składowej $q_{0ij}(t)$ (pierwszy człon wzoru). Każdy operator, nawet nie przynoszący bezpośrednich korzyści, może jednak wprowadzać do populacji odpowiednie modyfikacje (tak, jak to ma miejsce z operatorem mutacji w tradycyjnym algorytmie genetycznym). Należy zaznaczyć, że na danego osobnika działa w jednej iteracji tylko jeden operator, można więc jego działaniu jednoznacznie przypisać zasługi modyfikacji osobnika:

$$q_{ij}(t+1) = \begin{cases} q_{0ij}(t) + x_{ij}(t+1) + \alpha_{ij}(t) \cdot q_{ij}(t) & \text{dla } i = l \\ q_{ij}(t) & \text{dla pozostałych } i \end{cases} \quad (3.2)$$

gdzie:

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

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

$x_{ij}(t+1)$ ⁶ - 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;

⁶ W pracy [9] wartość poprawy rozwiązania jest odpowiednio normalizowana tak, aby sprowadzić ją do zakresu (0, 1), jednakże w tym przypadku nie ma takiej potrzeby, gdyż wartości funkcji celu do takiego właśnie przedziału należą.

$a_{ij}(t)$ - współczynnik zapominania, uwydatniający rolę najświeższych danych, przy osłabianiu dawniejszych (może również zależeć od czasu⁷);

l - indeks operatora wybranego w danej iteracji.

Drugi człon wzoru (3.2) - $x_{ij}(t)$ - stanowi poprawę funkcji celu rozwiązywanego zadania. W przypadku braku poprawy jest on równy zero.

Trzeci człon wzoru (3.2) 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.

3. 5 Selekcja sterowana

Wykorzystana do rozwiązania tego problemu metoda selekcji sterowanej składa się z dwóch metod składowych o znacząco różnych właściwościach: selekcji histogramowej (mającej właściwość znaczącego zwiększania zróżnicowania populacji) i deterministycznej ruletki (z silnym naciskiem na promowanie najlepszych osobników) [9]. Są one losowo wybierane i wykonywane w trakcie działania obliczeń ewolucyjnych. Prawdopodobieństwo wyboru i wykonania każdej z metod jest określane metodą pokazaną na Rys. 3. 5.

- | |
|---|
| <ol style="list-style-type: none">1. If $3 * \alpha(F) < \max(F_{mean} - F_{min}, F_{max} - F_{mean})$ then $p_{his} = p_{his} - a * p_{det}$;2. If $0,5 * \alpha(F) > \max(F_{mean} - F_{min}, F_{max} - F_{mean})$ then $p_{his} = p_{his} * (1 + a)$;3. If $0,5 * \alpha(F) \leq \max(F_{mean} - F_{min}, F_{max} - F_{mean}) \leq 3 * s(F)$ then $p_{his} = p_{his} * (0,5 - p_{det}) * a$; |
|---|

Rys. 3.5 Algorytm automatycznego strojenia prawdopodobieństw wystąpienia elementów selekcji mieszanej.

⁷ Przypadek zmiennego α i q_0 jest opisany w pracy [10].

Znaczenie symboli na Rys. 3.5: p_{his} - prawdopodobieństwo wystąpienia selekcji histogramowej, p_{det} - prawdopodobieństwo deterministycznej ruletki, F_{mean} , F_{min} , F_{max} - średnia, minimalna i maksymalna wartość funkcji dopasowania w populacji.

Jeżeli populacja osobników posiada zbyt małe odchylenie standardowe ($\sigma(F)$) w stosunku do rozpiętości wartości funkcji dopasowania ($\max(F_{mean}, F_{min}, F_{max}) - F_{mean}$), wtedy następuje zwiększenie prawdopodobieństwa wystąpienia selekcji histogramowej (pozycja 2 w Rys. 3.5). W przeciwnym wypadku zwiększane jest prawdopodobieństwo wystąpienia selekcji metodą deterministycznej ruletki (pozycja 1 w Rys. 3.5). Jeśli natomiast parametry populacji zawarte są w przedziale uznanym za korzystny, to prawdopodobieństwa wystąpienia obu metod selekcji są prawie równe (pozycja 3 w Rys. 3.5). Należy zaznaczyć także fakt, że zawsze $p_{his} + p_{det} = 1$, czyli któraś z metod zawsze musi wystąpić.

4. Rezultaty przeprowadzonych symulacji

Rezultaty uzyskane dla prostego algorytmu genetycznego były obiecujące, lecz mało dokładne, szczególnie dla większej liczby kroków sterowania. Aby poprawić tę sytuację, konieczne było zastosowanie operatorów specjalizowanych. Wyniki ich zastosowania przedstawione są w tym rozdziale. Poniżej wymienione są parametry symulacji wspólne dla wszystkich wykonanych symulacji:

- przedstawione wyniki są uśrednieniem najlepszych wyników uzyskanych w ciągu 10 symulacji dla każdego zadania;
- populacje startowe były generowane losowo;
- w każdym rozwiązywanym przykładzie zastosowano tę samą metodę selekcji sterowanej oraz metodę oceny operatorów, różne były jedynie zestawy operatorów

genetycznych;

- strategia $(\mu+\lambda)^8$, gdzie $\mu=60$, $\lambda=360$.

4.1 Obiekt deterministyczny

W przeprowadzonych symulacjach rozpatrywano trzy zadania o długościach horyzontu sterowania N równych odpowiednio 10, 100 i 1000, przy 20 możliwych stanach obiektu sterowanego i 32 sterowaniach (tablica przejść miała rozmiar 20×32). Liczba narzuconych ograniczeń na sterowania i wskaźników jakości na stan układu była w każdym przypadku równa liczbie kroków sterowania.

Wartości stanów, jak i sygnałów sterujących zostały przyjęte jako liczby z zakresu $(0; 1)$ - dla uproszczenia obliczeń i postaci funkcji przynależności. Nie stanowi to przeszkody dla ogólniejszych rozważań, jeśli uwzględni się możliwość przeskalowania tych wartości do innych zakresów. Funkcje przynależności do zbiorów rozmytych są trapezoidalne.

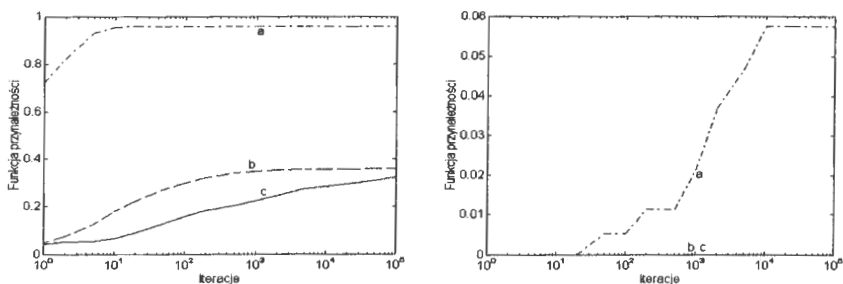
1. Algorytm z operatorami losowymi i heurystycznymi												
Iteracje	0	1	5	10	50	100	500	1000	5000	10000	50000	100000
N=10	0.0000	0.7233	0.9307	0.9563	0.9586	0.9586	0.9586	0.9586	0.9586	0.9586	0.9586	0.9586
N=100	0.0000	0.0503	0.1279	0.1788	0.2712	0.2968	0.3399	0.3470	0.3556	0.3561	0.3577	0.3600
N=1000	0.0000	0.0437	0.0572	0.0683	0.1297	0.1582	0.2023	0.2221	0.2727	0.2837	0.3081	0.3232
2. Algorytm wyłącznie z losowymi operatorami												
Iteracje	0	1	5	10	50	100	500	1000	5000	10000	50000	100000
N=10	0.0000	0.0000	0.0000	0.0000	0.0054	0.0054	0.0113	0.0214	0.0470	0.0575	0.0575	0.0575
N=100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
N=1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Tab. 4.1 Wyniki przeprowadzonych symulacji dla obiektu deterministycznego.

Rys. 4.1 przedstawia porównanie działania algorytmu ewolucyjnego z obecnością operatorów heurystycznych i bez dla różnych wartości N (a - $N=10$, b - $N=100$, c - $N=1000$). W tabeli 4.1 przedstawiono zestawienie wyników, na podstawie których

⁸ Symbol $(\mu+\lambda)$ oznacza, że do populacji potomnej wybierane są osobniki spośród rodziców i uzyskanych od nich potomków, gdzie liczebności tych dwóch podpopulacji są oznaczone odpowiednio przez μ i λ .

sporządzono wykresy na Rys. 4.1.



Rys 4.1 Graficzne przedstawienie wyników symulacji dla obiektu deterministycznego, z lewej dla obu typów operatorów, z prawej tylko dla losowych (a - $N=10$, b - $N=100$, c - $N=1000$).

Przewaga algorytmu ewolucyjnego z operatorami heurystycznymi nad algorytmem bez nich jest w tym przykładzie ogromna.

4.2 Obiekt stochastyczny

Dla obiektu stochastycznego przeprowadzono symulacje komputerowe dla $N=10$ i $N=100$ (dla $N=1000$ obliczenia trwają bardzo długo) z odpowiednią liczbą ograniczeń rozmytych na sterowania s stan układu.

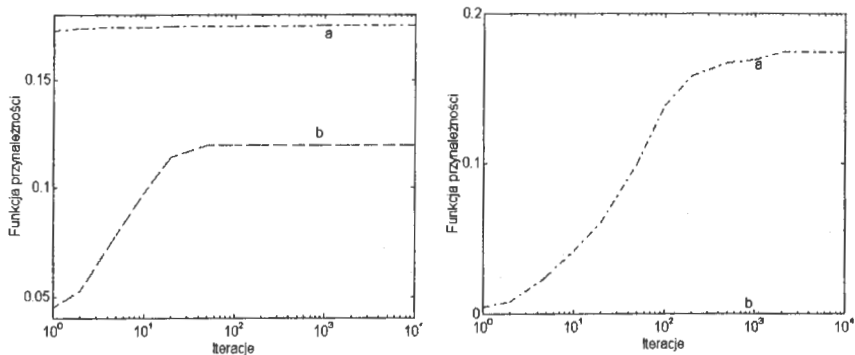
Podobnie jak w poprzednim przypadku zastosowano układ o 20 stanach i 32 możliwych sterowaniach, wylosowanych z przedziału $\langle 0;1 \rangle$. Dla obu wersji AE rozwiązywano to samo zadanie. Macierz prawdopodobieństw przejść pomiędzy stanami była również losowana, z tym że wartości prawdopodobieństw spełniają warunek sumowania się do jedności tak, aby zawsze można było wyjść z każdego stanu. Rozmiary zbiorów ograniczeń na sterowania i stan układu były równe N .

Graficzne wyniki symulacji przedstawia Rys 4.2, dokładne dane liczbowe pokazuje

tabela Tab. 4.2. Bez specjalnych trudności można zauważyć, że algorytm tylko z losowymi operatorami daje znacznie gorsze wyniki, szczególnie dla większych wymiarów zadań.

1. Algorytm z operatorami losowymi i heurystycznymi										
Iteracje	0	1	5	10	50	100	500	1000	5000	10000
N=10	0.0000	0.1725	0.1741	0.1744	0.1748	0.1748	0.1749	0.1749	0.1749	0.1749
N=100	0.0000	0.0448	0.0783	0.0976	0.1195	0.1195	0.1196	0.1196	0.1196	0.1196
2. Algorytm wyłącznie z operatorami losowymi										
Iteracje	0	1	5	10	50	100	500	1000	5000	10000
N=10	0.0000	0.0046	0.0244	0.0415	0.0996	0.1389	0.1678	0.1696	0.1749	0.1749
N=100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Tab. 4.2 Wyniki przeprowadzonych symulacji dla obiektu stochastycznego.



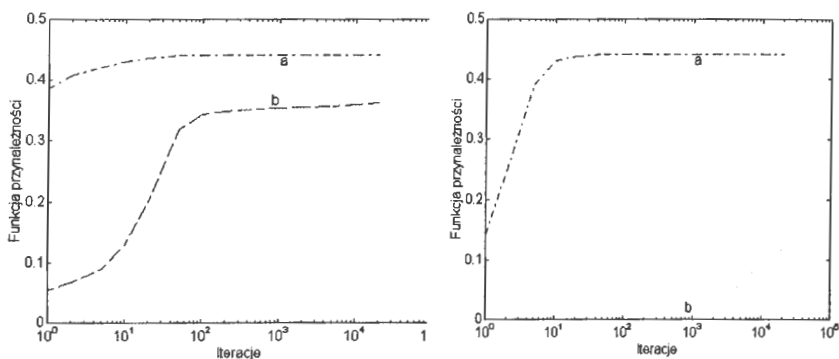
Rys 4.2 Graficzne przedstawienie wyników symulacji dla obiektu stochastycznego, z lewej oba typy operatorów, z prawej tylko losowe (a - $N=10$, b - $N=100$).

4.3 Obiekt rozmyty ze sterowaniem deterministycznym

W przypadku obiektu rozmytego ze sterowaniem deterministycznym badano również zachowanie się AE na przykładzie obiektu o 32 możliwych wartościach sterowań i 20 rozmytych stanach.

Macierz zawierająca wartości funkcji przynależności przejść między stanami o wymiarach $32 \times 20 \times 20$ była generowana losowo. Zawiera ona wartości rzeczywiste z zakresu $\langle 0; 1 \rangle$. Obliczenia przeprowadzono dla $N=10$ i $N=100$ z odpowiednią liczbą

ograniczeń rozmytych na sterowania i stan układu. Dla obu wersji AE rozwiązywano to samo zadanie. Wyniki symulacji przedstawiają rysunek 4.3 i tabela 4.3.



Rys 4.3 Graficzne przedstawienie wyników symulacji dla obiektu rozmytego ze sterowaniem deterministycznym, z lewej obie wersje operatorów, z prawej tylko losowe (a - $N=10$, b - $N=100$).

1. Algorytm z operatorami losowymi i heurystycznymi												
Iteracje	0	1	5	10	50	100	500	1000	2000	5000	10000	20000
N=10	0.0617	0.3854	0.4206	0.4311	0.4406	0.4406	0.4406	0.4406	0.4406	0.4406	0.4406	0.4406
N=100	0.0000	0.0538	0.0899	0.1299	0.3182	0.3427	0.3520	0.3534	0.3540	0.3555	0.3584	0.3620
2. Algorytm wyłącznie z operatorami losowymi												
Iteracje	0	1	5	10	50	100	500	1000	2000	5000	10000	20000
N=10	0.1062	0.1421	0.3912	0.4312	0.4406	0.4406	0.4406	0.4406	0.4406	0.4406	0.4406	0.4406
N=100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

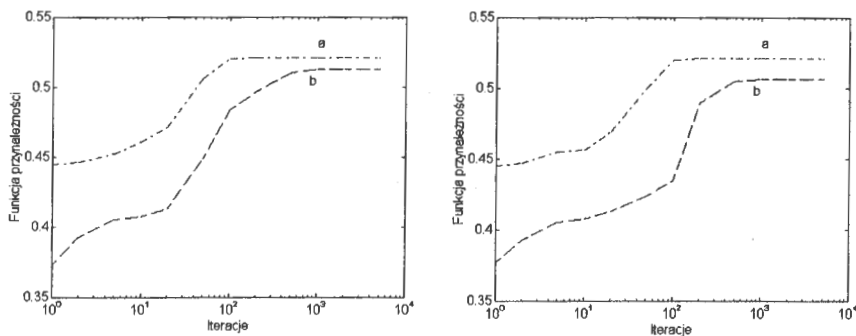
Tab. 4.3 Wyniki przeprowadzonych symulacji dla obiektu rozmytego ze sterowaniem deterministycznym.

Również i w tym przypadku widoczna jest znaczna przewaga AE wzbogacanego wiedzą o rozwiązywanym zadaniu nad klasycznym „ślepy” algorytmem ewolucyjnym.

4.4 Obiekt rozmyty ze sterowaniem rozmytym

Dla problemu sterowania rozmytego i obiektu rozmytego, testowano użycie AE na przykładzie obiektu o 32 możliwych sterowaniach rozmytych i 20 rozmytych stanach. Macierz zawierająca wartości funkcji przynależności przejść między stanami, o

wartościach rzeczywistych z zakresu $\langle 0;1 \rangle$ i wymiarach $32 \times 20 \times 20$, była generowana losowo. Obliczenia przeprowadzono dla $N=10$ i $N=100$ z odpowiednią liczbą ograniczeń rozmytych na sterowania i stan układu. Dla obu wersji AE rozwiązywano to samo zadanie. Wyniki symulacji przedstawiają rysunek 4.4 i tabela 4.4.



Rys 4.4 Graficzne przedstawienie wyników symulacji dla obiektu rozmytego ze sterowaniem rozmytym, z lewej z operatorami heurystycznymi, z prawej tylko losowe (a - $N=10$, b - $N=100$).

1. Algorytm z operatorami losowymi i heurystycznymi											
Iteracje	0	1	2	5	10	50	100	500	1000	2000	5000
N=10	0.4384	0.4448	0.4465	0.4522	0.4610	0.5063	0.5205	0.5212	0.5213	0.5213	0.5213
N=100	0.3705	0.3728	0.3928	0.4054	0.4077	0.4489	0.4841	0.5107	0.5130	0.5130	0.5130
2. Algorytm wyłącznie z operatorami losowymi											
Iteracje	0	1	2	5	10	50	100	500	1000	2000	5000
N=10	0.4432	0.4451	0.4470	0.4552	0.4561	0.5025	0.5200	0.5212	0.5213	0.5213	0.5213
N=100	0.3763	0.3768	0.3928	0.4054	0.4077	0.4234	0.4347	0.5053	0.5064	0.5064	0.5064

Tab. 4.4 Wyniki przeprowadzonych symulacji dla obiektu rozmytego ze sterowaniem rozmytym.

W tym przypadku różnica między działaniem algorytmu z operatorami heurystycznymi i bez nie jest już tak wyraźna. Wynika to z faktu, że dla przypadku optymalizacji ciągłej dość trudno jest wymyślić efektywne operatory heurystyczne. Można tu opierać się jedynie na dość nikłych przesłankach odnośnie ograniczeń występujących w

zadaniu, co tylko w niewielkim stopniu polepsza znalezione rozwiązania.

6. Wnioski

Dane zaprezentowane w poprzedniej części pracy pokazują, że istnieją problemy, których nie da się rozwiązać używając wyłącznie losowych, nie wzbogacanych wiedzą o problemie operatorów. Istotną sprawą byłoby odkrycie odpowiedniej proporcji występowania tych operatorów, co mogłoby korzystnie wpłynąć na czas obliczeń. Wielką zaletą algorytmów ewolucyjnych jest możliwość ich stopniowego rozwoju od rozwiązań „ślepych” do przystosowanych do rozwiązywania konkretnego problemu i wzbogaconych wiedzą o nim bez zasadniczej zmiany sposobu ich działania.

Literatura

- [1] R. E. Bellman, L. A. Zadeh: „Decision making in a fuzzy environment”, *Management Science*, 17, 1970, pp. 141-164;
- [2] E. Czogała, W. Pedrycz: „Elementy i metody teorii zbiorów rozmytych”, Warszawa PWN 1985;
- [3] J. Kacprzyk: „Multistage Fuzzy Control”, John Wiley & Sons, 1997.
- [4] J. Kacprzyk: „Multistage Control of a Stochastic System in a Fuzzy Environment Using a Genetic Algorithm”, *International Journal of Intelligent Systems*, Vol. 13, 1998, pp. 1011-1023;
- [5] J. Kacprzyk: „Multistage control under fuzziness using genetic algorithms”, *Control and Cybernetics*, Vol. 25, 1996, pp. 1181-1215;
- [6] J. Kacprzyk, R. A. Romero, F. A. C. Gomide: „Involving Objective and Subjective Aspects in Multistage Decision Making and Control under Fuzziness: Dynamic Programming and Neural Networks”, *International Journal of Intelligent Systems*, Vol. 14,

1999, pp. 79-104;

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

[8] A. Piegat: „Modelowanie i sterowanie rozmyte”, Akademska Oficyna Wydawnicza EXIT, Warszawa, 1999;

[9] J. Stańczak: „Rozwój koncepcji i algorytmów dla samodoskonalących się systemów ewolucyjnych”, praca doktorska, Politechnika Warszawska, 1999.

[10] J. Stańczak: „Algorytm ewolucyjny z populacją "inteligentnych" osobników”, Materiały IV Krajowej Konferencji Algorytmy Ewolucyjne i Optymalizacja Globalna, Łądek Zdrój, 2000;

[11] J. Stańczak: „Evolutionary Algorithm with Heuristic Operators in the Problem of Optimal Fuzzy Control”, Materiały V Krajowej Konferencji Algorytmy Ewolucyjne i Optymalizacja Globalna, Jastrzębia Góra, 2001;

[12] L. A. Zadeh: „Fuzzy sets”, Information and Control, vol. 8, 1965, pp. 338-353.





