

141/2002

**Raport Badawczy**

**RB/45/2002**

**Research Report**

**Algorytm ewolucyjny w problemie  
optymalnego sterowania rozmytego  
systemu deterministycznego  
z niejawnie zadany  
oraz nieskończonym horyzontem  
sterowania**

**J. 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

Kierownik Pracowni zgłaszający pracę:  
Prof. dr hab. inż. Janusz Kacprzyk

Warszawa 2002

# Algorytm ewolucyjny w problemie optymalnego sterowania rozmytego systemu deterministycznego z niejawnie zadaniem oraz nieskończonym horyzontem sterowania.

Jarosław Stańczak

*Institut Badań Systemowych PAN*

*ul. Newelska 6, Warszawa*

*stanczak@ibspan.waw.pl*

## Streszczenie

W pracy niniejszej rozpatrzono dwa problemy optymalnego sterowania rozmytego obiektu deterministycznego. Pierwszy z nich dotyczy obiektu z zadaniem w sposób niejawnie horyzontem czasowym. Czas zakończenia sterowania określa się jako liczbę kroków potrzebną do osiągnięcia po raz pierwszy, jednego z zadanego zbioru stanów terminalnych, kończących proces. Drugie zadanie dotyczy sytuacji z nieskończonym horyzontem planowania. Konieczne jest tu więc znalezienie trajektorii stacjonarnej będącej cyklem. Pierwszy z problemów podany został po raz pierwszy przez Zadeha i Bellmana w pracy [1], drugi sformułowany i rozwiązany został przez Kacprzyka i Staniewskiego np. w pracy [8].

W obu przypadkach sterowanie optymalne wyznaczane jest w tej pracy za pomocą algorytmu ewolucyjnego. Kryterium jakości - zbiór stanów optymalnych w kolejnych iteracjach oraz ograniczenia są zadane w tym zagadnieniu jako funkcje przynależności do odpowiednich zbiorów rozmytych. Funkcje przejść pomiędzy kolejnymi stanami są zadane w postaci macierzy z kolejnymi stanami obiektu.

## 1. Wstęp

Logika rozmyta (ang. fuzzy logic) jest uogólnieniem klasycznej logiki zerojedynkowej 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 [15] przedstawił teoretyczne podstawy dziedziny wiedzy, zwanej odtąd czasem 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 [11], [2] i [4].

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 nieznaney a priori długości dla prostego układu deterministycznego o pewnej skończonej liczbie stanów i sterowań, którego zachowanie w czasie określa pewna funkcja przejścia, w rozpatrywanym przykładzie będzie to macierz przejść. Układ posiada pewne narzucone ograniczenia na sterowania uzależnione od aktualnego stanu w którym się znajduje, określone odpowiednimi funkcjami przynależności oraz również rozmyte kryterium jakości oparte na jego stanie. Celem sterowania jest zapewnienie maksymalizacji w każdym kroku funkcji przynależności określającej kryterium jakości i ograniczenia na sterowania oraz osiągnięcie odpowiedniego stanu końcowego. W przypadku zadania z niejawnie podanym czasem zakończenia jest to jeden z zadanych stanów terminalnych. Dla nieskończonego czasu zakończenia, aby zapewnić wystąpienie cyklu, musi to być stan startowy. Do rozwiązania tego zadania został wykorzystany specjalizowany



algorytm ewolucyjny, którego opis, jak też i podsumowanie otrzymanych wyników znajdują się w kolejnych częściach tej pracy. Zbliżone zadania optymalnego sterowania rozmytego były już rozwiązywane metodami ewolucyjnymi [5], [6], [7], [14] lecz były to przypadki ze znanym a priori horyzontem sterowania, gdzie znana była długość poszukiwanego rozwiązania. W przedstawianej w tej pracy tematyce występuje sytuacja w której długość poszukiwanego rozwiązania (cyklu bądź liczba kroków do osiągnięcia jednego ze stanów terminalnych) nie jest znana.

## 2. Sformułowanie rozwiązywanego problemu

### 2.1 Prosty obiekt deterministyczny w optymalnym sterowaniu rozmytym

Deterministyczny obiekt sterowany opisuje pewna funkcja przejścia [7]:

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

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 ograniczenia mające charakter funkcji przynależności  $\mu_{C(x(t))}$  do określonego zbioru rozmytego sterowań dopuszczalnych dla danej iteracji  $C(x(t))$ , zależne od aktualnego stanu układu. W odpowiednich<sup>1</sup> krokach sterowania oceniany jest również otrzymany stan układu na podstawie funkcji przynależności  $\mu_{G(x(t))}$ , gdzie  $G(x(t))$  stanowi zbiór rozmyty stanów dopuszczalnych.

### 2.2 Zagadnienie z niejawnym czasem zakończenia

W zagadnieniu tym horyzont sterowania nie jest jawnie podany, stąd nazwa zagadnienia. Jest on jednak podany pośrednio w postaci pewnego zbioru stanów terminalnych  $W$ , będącego podzbiorem zbioru wszystkich możliwych stanów układu  $X$ . Gdy układ osiągnie któryś ze stanów należących do

---

<sup>1</sup> Dla układu z niejawnym czasem zakończenia ocenie podlega najczęściej tylko stan kończący działanie, dla układu z nieskończonym czasem zakończenia ocena stanu następuje w każdym kroku działania.

W kończy swoje działanie. Tak więc czas działania układu jest znany po osiągnięciu po raz pierwszy jednego ze stanów należących do  $W$ . Oczywiście stany te muszą być osiągalne z stanów należących do  $X \setminus W$ , gdyż w przeciwnym wypadku układ nigdy się nie zatrzyma. W przypadku układu, który działał przez czas oznaczony jako  $N^*$  optymalizowana jest następująca funkcja (2.2):

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

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 (nieznana a priori);

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

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

$\wedge$  - znak oznaczający t-normę<sup>2</sup> (w rozpatrywanym przypadku - funkcja minimum).

W przypadku zadania z niejawnym czasem zakończenia maksymalny rozmiar przestrzeni poszukiwań rozwiązania optymalnego można oszacować na (zakładając osiągalność stanów terminalnych i brak cykli w rozwiązaniu):

$$n = (|X \setminus W|)^{|U|} \quad (2.3)$$

gdzie:

$n$  - liczba możliwych rozwiązań;

$|X \setminus W|$  - maksymalna długość możliwego rozwiązania nie zawierającego powtórzeń stanów;

$|U|$  - liczba możliwych sterowań.

---

<sup>2</sup> 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 [11]. W tej pracy przyjęto najprostszą i chyba najczęściej stosowaną wersję:  $\wedge$  odpowiada funkcji minimum(...), a  $\vee$  maksimum(...).

Wzór (2.3) wynika z faktu, że aby osiągnąć któryś ze stanów terminalnych należy co najwyżej raz znaleźć się w każdym ze stanów przejściowych, a następnie w jednym z terminalnych. Na każdej pozycji możliwe jest dowolne sterowanie. Dłuższe rozwiązania muszą zawierać cykle, co niepotrzebnie komplikuje rozwiązanie i spowalnia obliczenia.

Po raz pierwszy zadanie to podali i rozwiązali metodą iteracyjną Bellman i Zadeh w pracy [1]. Następnie rozwiązanie oparte na teorii grafów zostało podane w pracy [9] Komołowa, metoda podziałów i ograniczeń została wykorzystana w pracy [3] Kacprzyka. W pracy niniejszej prezentowane będzie podejście ewolucyjne do rozwiązania tego zagadnienia.

### 2.3 Zagadnienie z nieskończonym czasem zakończenia

W tym przypadku mamy do czynienia z zadaniem, w którym czas zakończenia jest znany i wynosi  $\infty$ . Rozwiązanie tego przypadku wymaga znalezienia strategii stacjonarnej będącej cyklem o długości  $T$ . W szczególności chodzi o znalezienie cyklu optymalizującego wyrażenie (2.4):

$$\mu_D(\hat{u}(0), \hat{u}(1), \dots | x(0)) = \max_{u(0), u(1), \dots} \lim_{N \rightarrow \infty} \min_{t=0..N} b^t \bullet [\mu_{C(x(t))}(u(t) | x(t)) \wedge \mu_{G(x(t+1))}(x(t+1))] \quad (2.4)$$

gdzie:

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

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

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

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

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

$b$  - współczynnik dyskonta,  $b \geq 1$ , zmniejszający znaczenie późniejszych, słabiej określonych stanów układu;

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

W przypadku zadania z nieskończonym czasem zakończenia maksymalny rozmiar przestrzeni poszukiwań rozwiązania optymalnego można oszacować na:



$$n=(|X|)^{|U|} \quad (2.5)$$

gdzie:

$n$  - liczba możliwych rozwiązań;

$|X|$  - maksymalna długość możliwego rozwiązania nie zawierającego powtórzeń stanów;

$|U|$  - liczba możliwych sterowań.

Wzór (2.2) wynika z faktu, że aby powrócić do stanu początkowego, należy co najwyżej raz znaleźć się w każdym z pozostałych stanów. Na każdej pozycji możliwe jest dowolne sterowanie. Dłuższe rozwiązania muszą zawierać dodatkowe cykle, co niepotrzebnie komplikuje rozwiązanie i spowalnia obliczenia.

Rozwiązanie tego zagadnienia bazuje na metodzie strategii iteracyjnej i zostało przedstawione w pracach [8], [4], [7].

### 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.

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.*

Dostosowanie to polega na odpowiednim zakodowaniu problemu oraz doborze operatorów ewolucyjnych, umożliwiających 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.

### **3.1 Zakodowanie osobnika w AE**

#### **3.1.1 Zagadnienie z niejawnym czasem zakończenia**

W tym przypadku całą informację o konkretnym rozwiązaniu i jego jakości niesie sekwencja sterowań obiektu [4], przy założeniu stałego stanu początkowego. Długość sekwencji jest ustalana dynamicznie, tak więc poszczególne osobniki populacji różnią się długościami kodowanych rozwiązań. Genotypem pojedynczego osobnika jest sekwencja numerów, oznaczających sterowania ze zbioru sterowań (nie są to konkretne wartości sterowań, a jedynie ich indeksy ze zbioru sterowań). Osobnik pamięta również aktualna długość sekwencji. Długość ta może być modyfikowana przez operatory genetyczne. Populacja startowa zawiera również osobniki o zróżnicowanych długościach sekwencji. Osobnik taki może kodować rozwiązania nie kończące się jednym ze stanów terminalnych, dlatego też każdy operator genetyczny wywołuje funkcję korygującą, która stara się zmodyfikować "koniec" sekwencji tak, aby otrzymać rozwiązanie dopuszczalne. W przypadku, gdy to zawiedzie takie rozwiązanie jest karane znacznym obniżeniem wartości funkcji dopasowania.

#### **3.1.2 Zagadnienie z nieskończonym czasem zakończenia**

Podobnie jak w poprzednim zagadnieniu całą informację o konkretnym rozwiązaniu i jego jakości niesie sekwencja sterowań obiektu [4], przy założeniu znanego stanu początkowego z tym, że w tym przypadku staje się on także stanem końcowym dla poszukiwanego cyklu. Długość sekwencji jest również dynamicznie i zawiera sekwencję numerów, oznaczających sterowania ze zbioru sterowań. Aktualna długość sekwencji jest również pamiętana w każdym osobniku. Długość ta może być modyfikowana przez operatory genetyczne. W populacji startowej generowane są osobniki o losowo wybranych długościach (z pewnego określonego przedziału). Podobnie i w tym przypadku

osobnik może kodować rozwiązania nie tworzące cyklu, dlatego też każdy operator genetyczny wywołuje funkcję korygującą, która sprawdza poprawność i wobec jej braku stara się zmodyfikować "koniec" sekwencji tak, aby otrzymać rozwiązanie dopuszczalne. Jeśli w dalszym ciągu rozwiązanie nie spełnia warunków zadania, funkcja kary znacznie obniża wartość jego funkcji dopasowania.

### **3.2 Funkcja dopasowania**

We wszystkich rozpatrywanych przypadkach funkcja celu odpowiedniego zadania (2.2 i 2.4) 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 te same z zakresu  $(0;1)$ . W przypadku funkcji dopasowania istotne jest aby wartości te były nieujemne (w przeciwnym wypadku zawodzi wiele metod selekcji osobników, np. te 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. Problemem jest jedynie traktowanie zdarzających się od czasu do czasu rozwiązań sprzecznych z założeniami zadania (czyli nie kończących się jednym ze stanów terminalnych lub też nie kodujących cyklu). W takim przypadku otrzymywana wartość funkcji dopasowania jest dzielona przez 100, czyli stosowana jest metoda funkcji kary za naruszenie ograniczeń.

### **3.3 Operatory genetyczne**

Dla tak skonstruowanych osobników należało teraz zaproponować kilka operatorów ewolucyjnych, umożliwiających ich modyfikację w kolejnych pokoleniach. Opracowanie prostych losowych operatorów nie nastęrcza większych problemów i można tu zastosować standardowe operacje typu: mutacji, krzyżowania i inwersji uzupełnionych o wynikające ze sposobu kodowania operatory usunięcia lub dodania kroku sterowania (o losowo wybranej wartości) do rozwiązania. Symulacje komputerowe wykazały jednak, że taki zestaw operatorów działa dość wolno, szczególnie przy większych rozmiarach macierzy przejść. Sytuacja ta spowodowała konieczność znalezienia bardziej



efektywnego zestawu operatorów, a taki mógł powstać jedynie przy zastosowaniu heurystyk. 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 takiego niż operator losowy. W symulacjach stosowano po kilka wersji każdego z operatorów heurystycznych 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 dość 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 Zadanie z niejawnym czasem zakończenia

Zastosowano trzy operatory: krzyżowanie (wymianę losowo wybranych, końcowych fragmentów osobników), mutację (zamianę jednego losowo wybranego sterownia na wylosowane inne) i inwersję (przestawienie kolejności sterowań w losowo wybranym podciągu osobnika). Oprócz tego wystąpiły te same operatory w wersji wielokrotnej (3 powtórzenia tego samego operatora). Zestaw operatorów losowych został uzupełniony o operatory losowego usunięcia lub też dodania nowego kroku o również losowo wybranej wartości. Przedstawione tu operatory należą do klasy tzw. operatorów ślepych, a więc działających w pełni losowo bez jakiegokolwiek wiedzy o problemie. Jak już to zostało wspomniane, do przyspieszenia obliczeń ewolucyjnych wprowadzono kilka operatorów heurystycznych.

Pierwszy z nich przechodzący sekwencję sterowań od początkowego do końcowego (operator "w przód") i próbujący korygować miejsca, gdzie wartość funkcji dopasowania była równa lub mniejsza<sup>3</sup> od wartości dopasowania całego osobnika. Jest to wykonywane przez sprawdzenie innych

---

<sup>3</sup> Ponieważ najniższa z wartości funkcji dopasowania staje się globalną wartością funkcji dopasowania, to mogłoby się wydawać, że napotkanie niższych wartości niż obliczona wartość funkcji dopasowania jest niemożliwe. Jednakże może się tak zdarzyć w przypadku, gdy rozwiązanie było modyfikowane przez ten operator i w trakcie jego działania rozwiązanie się zmieniło, jak i występujące w nim wartości funkcji przynależności.

możliwych sterowań (wybieranych z utworzonej wcześniej listy sterowań dopuszczalnych w danej sytuacji) 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 jest niepusta, to spośród zapisanych tam możliwości wybierane jest nowe sterowanie. Używane są różne metody wyboru: najlepszego rozwiązania, losowa i losowa skalowana (czym lepsze rozwiązanie, tym większe prawdopodobieństwo wyboru). W obliczeniach zastosowano wszystkie trzy wersje operatora. Algorytm działania tego operatora przedstawiony jest na Rys. 3.2.

1.  $i := I$ ;

2. Obliczenie wartości:

a) jeśli bieżący stan nie jest jednym z terminalnych:

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

b) w przeciwnym wypadku:

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

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

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);

c) jeśli  $i=k$  (koniec rozwiązania), wybierane jest rozwiązanie takie, aby  $x(k)$  było jednym ze stanów terminalnych, jeśli jest to niemożliwe, to otrzymamy rozwiązanie niedopuszczalne, jeśli natrafiono na stan terminalny dla  $i < k$ , to pozostała część rozwiązania jest odrzucana i  $k = N^*$ ;

4.  $i := i + 1$ ;

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

6. Koniec

**Rys. 3.2. Algorytm działania operatora "w przód".**

Drugi operator heurystyczny wykonuje korekcję od końca do początku sekwencji sterowań, startując z jednego ze stanów terminalnych. Sterowania wybierane są 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. Pewną trudnością jest tu również konieczność "trafienia" w założony stan początkowy  $x(0)$ . Algorytm działania operatora "do tyłu" przedstawiony jest na Rys. 3.3.

1.  $i = k$ ,  $k = N^*$  - start od stanu będącego terminalnym, zakładamy, że długość rozwiązania jest czasem zakończenia działania układu;
2. Obliczenie wartości:
  - a) dla stanu terminalnego ( $i = k$ ):
 
$$\mu(k) := \mu_{C(x(k))}(u(k-1)) \wedge \mu_{G(x(k))}(x(k))$$
  - b) jeśli bieżący stan nie jest jednym z terminalnych:
 
$$\mu(i) := \mu_{C(x(i))}(u(i-1))$$

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:
  - a) jeśli  $i > 0$  to  $x(i) := f(x(i-1), u(i-1))$ ;
  - b) jeśli  $i = 0$  to  $x(0) := f(x(i-1), u(i-1))$ ;
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 Zadanie z nieskończonym czasem zakończenia

Podobnie jak w poprzednim przypadku zastosowano trzy operatory losowe: krzyżowanie (wymianę losowo wybranych, końcowych fragmentów osobników), mutację (zamianę jednego losowo wybranego sterownia na wylosowane inne) i inwersję (przestawienie kolejności sterowań w losowo wybranym podciągu osobnika). Zastosowano również te operatory w wersji wielokrotnej (3 powtórzenia tego samego operatora). Ponieważ zastosowany jest tu również dynamiczny sposób



kodowania rozwiązań, więc do typowych operatorów losowych dodano operatory losowego usunięcia lub też dodania nowego kroku o również losowo wybranej wartości.

Oprócz operatorów losowych zastosowano również operatory heurystyczne. Pierwszy z nich (operator "w przód") stara się korygować miejsca, gdzie wartość funkcji dopasowania jest równa lub mniejsza od wartości dopasowania całego osobnika. W takim miejscu sprawdzane są inne możliwe sterowania, wybierane z utworzonej wcześniej listy sterowań dopuszczalnych w danej iteracji w miejsce poprzednio używanego. Jeśli możliwości takich jest więcej, to tworzona jest lista możliwych zamienników aktualnego sterowania. Z listy tej wybierane jest jedno ze sterowań. Stosowane są następujące metody wyboru: rozwiązanie najlepsze, losowa lub losowa z prawdopodobieństwem proporcjonalnym do jakości nowego sterowania. W obliczeniach zastosowano wszystkie trzy wersje wyboru. Algorytm działania tego operatora przedstawiony jest na Rys. 3.2.

<ol style="list-style-type: none"><li>1. <math>i := I</math>;</li><li>2. Obliczenie wartości<math display="block">\mu(i) := \mu_{C(x(i-1))}(u(i-1)) \wedge \mu_{G(x(i))}(x(i));</math></li><li>3. Jeśli <math>\mu(i) \leq \mu_D(u(0), \dots, u(T-1) x(0))</math><ol style="list-style-type: none"><li>a) utworzenie listy dozwolonych par <math>(u(i-1), x(i))</math>, dla których<math display="block">\mu(i) &gt; \mu_D(u(0), \dots, u(N-1) x(0));</math></li><li>b) wybór jednej z par (możliwe są różne kryteria wyboru) i modyfikacja zestawu sterowań osobnika na pozycji <math>i</math> (w razie braku możliwości poprawy nie następuje modyfikacja rozwiązania);</li><li>c) jeśli <math>i=T</math>, wybierane jest rozwiązanie takie, aby <math>x(0)</math> było równe <math>x(T)</math>, jeśli jest to niemożliwe, to otrzymamy rozwiązanie niedopuszczalne.</li></ol></li><li>4. <math>i := i+1</math>;</li><li>5. Jeśli <math>i \leq T</math> powrót do punktu 2;</li><li>6. Koniec</li></ol>
--

Rys. 3.2. Algorytm działania operatora "w przód".

Drugi operator heurystyczny wykonywał zbliżone działanie 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  $x(0)$ . Algorytm działania operatora "do tyłu" przedstawiony jest na Rys. 3.3.

1.  $i := T, x(T) = x(0)$  - start od stanu będącego początkowym i jednocześnie końcowym cyklu;

2. Obliczenie wartości

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

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

3. Wybór jednej z par tak, aby:

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

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:

a) jeśli  $i > 0$  to  $x(i) := f(x(i-1), u(i-1))$ ;

b) jeśli  $i = 0$  to  $x(T) := f(x(i-1), u(i-1))$ ;

5.  $i := i - 1$ ;

6. Jeśli  $i > 0$  powrót do punktu 2;

7. Koniec.

**Rys. 3.3. Algorytm działania operatora "do tyłu".**

### 3.4 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 [10], [12] i [13]. 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).

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

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 jakiegось operatora (może zależeć od czasu);

$x_{ij}(t+1)^4$  - poprawa funkcji celu,  $x_{ij}(t+1) = Q(t) - Q(t+1)_{ij}$ ,  $Q(t)$  - najlepsze rozwiązanie

---

<sup>4</sup> W pracy [12] 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.



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;

$\alpha_{ij}(t)$  - współczynnik zapominania, uwydatniający rolę najświeższych danych, przy osłabianiu dawniejszych (może również zależeć od czasu<sup>5</sup>);

$l$  - 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) [12]. Są one losowo wybierane i wykonywane w trakcie działania obliczeń ewolucyjnych. Prawdopodobieństwo wyboru i wykonania każdej z metod jest określane metodą, którą pokazuje wzór (3. 3):

$$p_{his}(t+1) = \begin{cases} p_{his}(t) \cdot (1-a) & \text{for } \max(F_{mean} - F_{min}, F_{max} - F_{mean}) > 3 \cdot \sigma(F) \\ p_{his}(t) \cdot (1-a) + a & \text{for } \max(F_{mean} - F_{min}, F_{max} - F_{mean}) < 0.5 \cdot \sigma(F) \\ p_{his} - (0.5 - p_{det}(t)) \cdot a & \text{for } \left( \begin{array}{l} \max(F_{mean} - F_{min}, F_{max} - F_{mean}) \geq 0.5 \cdot \sigma(F) \text{ and} \\ \max(F_{mean} - F_{min}, F_{max} - F_{mean}) \leq 3 \cdot \sigma(F) \end{array} \right) \end{cases} \quad (3.3)$$

gdzie:

$p_{his}$  - 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 ( $\max(F_{max} - F_{min}, F_{max} - F_{mean})$ ), 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 dzięki zastosowaniu operatorów heurystycznych w algorytmie ewolucyjnym są bardzo obiecujące. Wyniki ich zastosowania przedstawione są w tym rozdziale.

##### 4.1 Problem z niejawnym czasem zakończenia

W przeprowadzonych symulacjach rozpatrywano zadanie o 20 możliwych stanach obiektu sterowanego i 32 sterowaniach (tablica przejść miała rozmiar  $20 \times 32$ ). Liczba narzuconych ograniczeń na sterowania była zależna od liczby możliwych stanów obiektu. W przypadku rozmytych ograniczeń na stan obiektu istotne były tylko wartości uzyskiwane dla stanów terminalnych (dla stanów przejściowych traktowane były jako nieistotne).

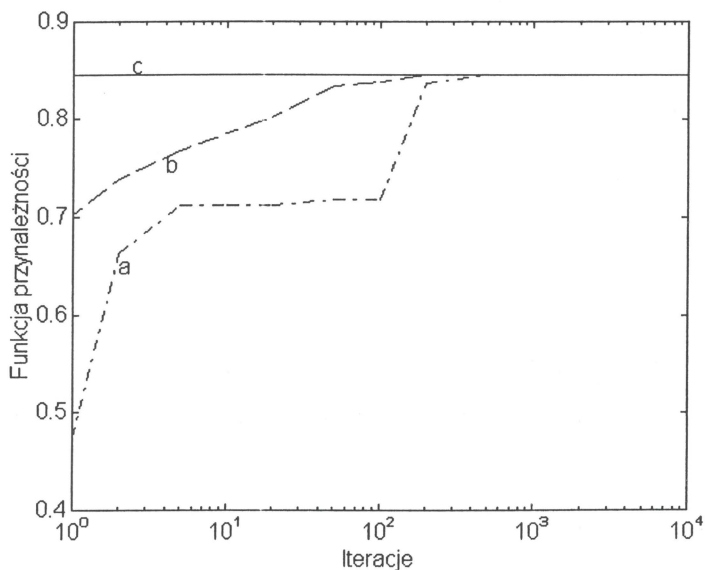
Rys. 4.1 przedstawia wykresy sporządzone dla minimalnej, średniej i maksymalnej wartości funkcji celu uzyskanej w danym kroku dla 100 przeprowadzonych symulacji. Pod uwagę brano najlepszy wynik w populacji otrzymany w danej iteracji. W tabeli 4.1 przedstawiono zestawienie wyników, na

---

<sup>5</sup> Przypadek zmiennego  $\alpha$  i  $q_0$  jest opisany w pracy [13].

podstawie których sporządzono wykres na Rys. 4.1.

Wyniki pokazane w tabeli 4.1 wskazują na sytuację w której rozwiązywane zadanie było dość proste i bardzo szybko udało się znaleźć prawdopodobnie optymalne rozwiązanie. Oczywiście wszystko tu zależy od rozmiaru zadania i postaci narzuconych ograniczeń. Jednakże algorytm ewolucyjny jest bardzo efektywnym narzędziem i bardzo szybko znajduje bardzo dobre rozwiązania. Niestety nie można z całą pewnością stwierdzić, że są to rozwiązania optymalne, gdyż wynika to ze specyfiki algorytmów ewolucyjnych.



*Rys 4.1 Graficzne przedstawienie wyników symulacji: a - rozwiązanie najgorsze, b - średnia z wyników, c - wynik najlepszy w ciągu 100 symulacji.*

Iteracje	0	1	5	10	50	100	200	500	1000	2000	5000	10000
min	0.0770	0.4738	0.7119	0.7119	0.7173	0.7173	0.8361	0.8453	0.8453	0.8453	0.8453	0.8453
śr.	0.4809	0.7034	0.7677	0.7859	0.8343	0.8381	0.8451	0.8453	0.8453	0.8453	0.8453	0.8453
maks.	0.7173	0.8453	0.8453	0.8453	0.8453	0.8453	0.8453	0.8453	0.8453	0.8453	0.8453	0.8453

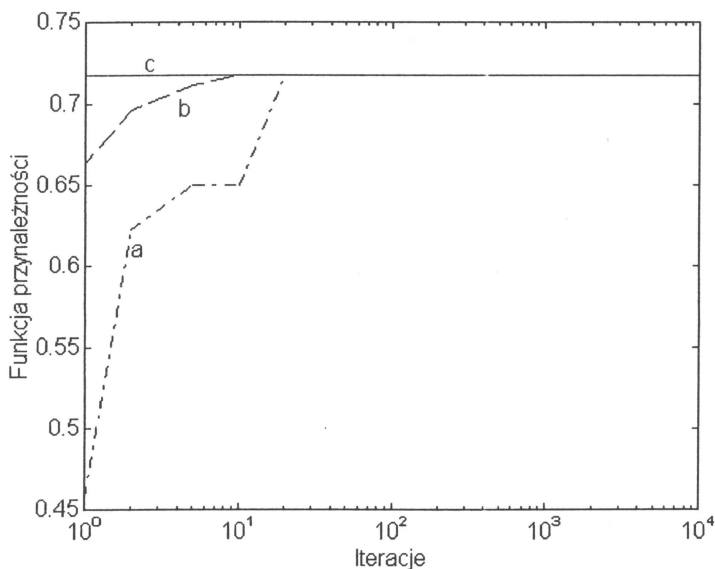
*Tab. 4.1 Wyniki przeprowadzonych symulacji.*

## 4.2 Problem z nieskończonym czasem zakończenia

Dla problemu z nieskończonym czasem symulacji również rozpatrywano zadanie o 20 możliwych

stanach obiektu sterowanego i 32 sterowaniach (tablica przejść miała rozmiar 20x32) z liczbą narzuconych ograniczeń na sterowania zależną od liczby możliwych stanów obiektu. Każdy stan posiadał własną wartość funkcji przynależności, stałą w czasie symulacji.

Na rysunku 4.2 przedstawiono wykresy uzyskane dla minimalnej, średniej i maksymalnej wartości funkcji celu. Wyniki zebrano ze 100 przeprowadzonych symulacji. Pod uwagę brano najlepszy wynik w populacji otrzymany w danej iteracji. W tabeli 4.2 przedstawiono wyniki, na podstawie których powstał rysunek 4.2.



**Rys 4.2. Graficzne przedstawienie wyników symulacji: a - rozwiązanie najgorsze, b - średnia z wyników, c - wynik najlepszy w ciągu 100 symulacji.**

Iteracje	0	1	2	5	10	50	100	200	500	1000	5000	10000
min.	0.0035	0.4580	0.6228	0.6501	0.6501	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176
śr.	0.1583	0.6643	0.6963	0.7115	0.7169	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176
maks.	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176	0.7176

**Tab. 4.2 Wyniki przeprowadzonych symulacji.**

Podobnie jak w poprzednim zadaniu wyniki pokazane w tabeli 4.2 wskazują na to, że rozwiązywane zadanie było dość proste (macierz przejść i ograniczenia na sterowania były takie same jak w

zadaniu z niejawnym czasem zakończenia) i nawet udało się znaleźć losowo prawdopodobnie optymalne rozwiązanie. Oczywiście wszystko tu zależy od rozmiaru zadania i postaci narzuconych ograniczeń.

## **6. Wnioski**

Dane zaprezentowane w tej pracy pokazują działanie algorytmu ewolucyjnego bazującego na operatorach wzbogacanych wiedzą o rozwiązywanym problemie i przystosowanych do używanych danych z jednoczesną obecnością operatorów tradycyjnych. Ich obecność zapobiega możliwości “wyczerpania się” operatorów heurystycznych i utrzymuje zwiększony element eksploracyjny w algorytmie ewolucyjnym. Wielką zaletą algorytmów ewolucyjnych jest możliwość ich stopniowego rozwoju od rozwiązań “ślepych”, bez wiedzy o rozwiązywanym problemie, do “inteligentnych” - przystosowanych do rozwiązywania konkretnego problemu, bez zasadniczych zmian w sposobie ich działania.

## **Literatura**

- [1] R. E. Bellman, L. A. Zadeh: “Decision making in a fuzzy environment”, *Management Science*, 17, pp. 141-164.
- [2] E. Czogała, W. Pedrycz: “Elementy i metody teorii zbiorów rozmytych”, Warszawa PWN 1985.
- [3] J. Kacprzyk: “Multistage Decision Making under Fuziness”, Verlag TÜV Rheinland, Cologne, 1983.
- [4] J. Kacprzyk: “Multistage Fuzzy Control”, John Wiley & Sons, 1997.
- [5] 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.
- [6] J. Kacprzyk: “Multistage control under fuzziness using genetic algorithms”, *Control and Cybernetics*, Vol. 25, 1996, pp. 1181-1215.
- [7] J. Kacprzyk: “Wieloetapowe sterowanie rozmyte”, WNT, Warszawa, 2001.
- [8] J. Kacprzyk, P. Staniewski: “Control of a deterministic system in a fuzzy environment over an



- infinite planning horizon”, Fuzzy Sets and Systems, Vol. 10, 1983, pp. 291-298.
- [9] S. V. Komolov, S. P. Makeev, G. P. Serov, L. F. Shakhnov: “On the problem of optimal control of a finite automaton with fuzzy constraints and fuzzy goal”, Kybernetika, Vol 6, Kijów, 1979, pp 30-34.
- [10] J. Mulawka, J. Stańczak: “Genetic Algorithms with Adaptive Probabilities of Operators Selection”, Proceedings of ICCIMA'99, New Delhi, India, 1999.
- [11] A. Piegat: “Modelowanie i sterowanie rozmyte”, Akademicka Oficyna Wydawnicza EXIT, Warszawa, 1999.
- [12] J. Stańczak: “Rozwój koncepcji i algorytmów dla samodoskonających się systemów ewolucyjnych”, praca doktorska, Politechnika Warszawska, 1999.
- [13] J. Stańczak: “Algorytm ewolucyjny z populacją "inteligentnych" osobników”, Materiały IV Krajowej Konferencji Algorytmy Ewolucyjne i Optymalizacja Globalna, Łądek Zdrój, 2000.
- [14] 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.
- [15] L. A. Zadeh: “Fuzzy sets”, Information and Control, Vol. 8, 1965, pp. 338-353.



