

JOLANTA SOCAŁA (Katowice)  
WITOLD KOSIŃSKI (Warszawa i Bydgoszcz)  
STEFAN KOTOWSKI (Warszawa)

## O asymptotycznym zachowaniu prostego algorytmu genetycznego

**Streszczenie.** W pracy zdefiniowano prosty algorytm genetyczny w terminach skończonego multizbioru potencjalnych rozwiązań (osobników danej populacji), na którym są określone operacje krzyżowania, mutacji i selekcji, każda z pewnym prawdopodobieństwem. Działając złożeniem tych operacji na dowolną populację, tworzymy nową populację. Istnienie funkcji przystosowania (dopasowania), określonej na osobnikach populacji, pozwala powiązać prawdopodobieństwo selekcji osobników do nowej populacji z wartościami, jakie funkcja przystosowania przyjmuje na osobniku. Przejście z jednej generacji do drugiej jest realizowane przez operator działający na wektory probabilistyczne charakteryzujące rozkład prawdopodobieństwa pojawienia się każdej z możliwych populacji. Jest to operator Markowa. W teorii operatorów Markowa oraz operatorów dodatnich znanych jest wiele twierdzeń dotyczących istnienia punktów stałych oraz zbieżności ciągu iteracji operatora (np. [9, 12, 13]). Korzystając z tych wyników, znaleźliśmy warunki wystarczające i konieczne stabilności operatora Markowa związanego z pewną klasą algorytmów genetycznych.

**Słowa kluczowe:** populacja, krzyżowanie, mutacja, selekcja, operator Markowa, stabilność, zbieżność, rozkład graniczny, prosty algorytm genetyczny.

**1. Wstęp.** Problemy optymalizacyjne pełnią ważną rolę we współczesnych zastosowaniach matematyki do zagadnień modelowych w fizyce i technice. Wiele rozwiązań konkretnych problemów nie byłoby możliwych do uzyskania, gdyby nie zostały odpowiednio rozwinięte metody optymalizacji funkcji rzeczywistych, tj. poszukiwania minimum czy maksimum danego problemu fizycznego. Wszystkie zadania optymalizacji mają w całej swej różnorodności kilka wspólnych cech. Każde z nich jest scharakteryzowane przez pewien zbiór  $\mathcal{X}$ , nazywany przestrzenią rozwiązań lub przestrzenią poszukiwań, o której zakładamy, że jest podzbiorem zwartym i ograniczonym pewnej przestrzeni polskiej (metrycznej i ośrodkowej), której każdy punkt jest punktem skupienia. Jeśli przestrzeń ta ma strukturę liniową, to zakłada się, że przestrzeń jest skończenie wymiarowa. Wśród elementów tego zbioru szukamy rozwiązań zadania. Ponadto na przestrzeni  $\mathcal{X}$  jest określona funkcja

jakości (lub funkcja celu)  $\phi : \mathcal{X} \rightarrow \mathbb{R}$ , przyporządkowująca elementowi  $x \in \mathcal{X}$  jego ocenę ze zbioru liczb rzeczywistych  $\mathbb{R}$ . W funkcji jakości  $\phi$  zawieramy różne kryteria, których spełnienia wymagamy od pożądanego rozwiązania. Funkcja  $\phi$  nie musi być gładka. Dzięki niej możemy porównywać otrzymane rozwiązania.

Znalezienie zadowalającego rozwiązania dla takich problemów wymaga dużej wiedzy o zadaniu i znajomości algorytmów optymalizacyjnych. Należy zdawać sobie sprawę z tego, że rozwiązywanie praktycznych problemów jest w dużej mierze sztuką, w której dużą rolę pełni nie tylko praktyka, ale konieczność korzystania z metod heurystycznych.

Zazwyczaj sam fakt wykorzystywania komputerów w obliczeniach powoduje, że w efekcie obliczeń znajdujemy jedynie pewne rozwiązanie przybliżone. Wynika to stąd, że liczby w komputerze są reprezentowane przez skończone ciągi bitów, co powoduje kumulowanie się błędów zaokrągleń. Ponadto w obliczeniach komputerowych nie realizujemy zazwyczaj procesów ciągłych, a jedynie ich dyskretne przybliżenia, i to przez odpowiednią skończoną liczbę dyskretnych kroków.

Pojawiają się również inne problemy: niedoskonałość metod numerycznych czy duża złożoność obliczeniowa. W takich przypadkach zastosowanie znanego algorytmu dokładnego nie jest możliwe i jesteśmy zmuszeni poszukiwać innych podejść do rozwiązywanego problemu optymalizacyjnego. Klasą algorytmów, które dowiodły swojej przydatności w najtrudniejszych zadaniach praktycznych, są *algorytmy probabilistyczne*. Algorytmy te znajdują rozwiązanie problemu (zwykle suboptymalne) z prawdopodobieństwem mniejszym od 1. Natomiast ich przebieg działania zależy nie tylko od właściwości zadania, ale także od czynników losowych.

Jednym z najczęściej używanych sposobów poszukiwania niegładkich rozwiązań problemów optymalizacyjnych jest *algorytm genetyczny*. Jest usytuowany w nurcie *obliczeń ewolucyjnych* i pojawił się jako pewna implementacja procesu ewolucji (por. Holland [2] i Hollstien [3]) organizmów żywych, gdy osobniki danego gatunku, lepiej przystosowane do warunków, w których żyją, mają większe prawdopodobieństwo przeżycia niż osobniki gorzej przystosowane.

Obliczenia ewolucyjne należą do rozwijającej się dziedziny wiedzy zwanej inteligencją obliczeniową (ang. *computational intelligence*), wyrosłej z dziedziny zwanej sztuczną inteligencją. Rozwijane w tej dziedzinie metody obliczeniowe czasami noszą nazwę *metod miękkich* (ang. *soft computing*), gdyż stosowane tutaj algorytmy nie są dobrze umotywowane, a inspirację dla nich zaczerpnięto z przyrody; brak jest jednak pełnych dowodów ich poprawności.

Algorytmy probabilistyczne, a wśród nich algorytmy genetyczne, na ogół nie przeszukują przestrzeni poszukiwań (odpowiednik przestrzeni potencjalnych rozwiązań problemu optymalizacyjnego) w sposób wyłącznie losowy.

Zwykle bazują one na pewnych heurystykach, np. na założeniach dotyczących kształtu i regularności funkcji jakości zadania. Heurystyczne są tutaj nie tylko założenia o funkcji jakości, ale też same metody tworzenia algorytmu obliczeniowego.

Skoro kod (chromosom) opisuje budowę wszystkich żywych organizmów i służy do przechowywania i przekazywania materiału genetycznego, można się pokusić, by w sposób choćby uproszczony zareprezentować taki kod w obliczeniach komputerowych. Mamy tutaj do dyspozycji najprostszy sposób, jest nim ciąg bitów. W ten sposób pojawiają się algorytmy wykorzystujące mechanizmy ewolucji, zwane algorytmami ewolucyjnymi; szczególnym ich rodzajem są **binarne algorytmy genetyczne** (BAG), będące tematem artykułu Kiesia i Michalewicza opublikowanym w naszym piśmie przed pięciu laty [4]. Natomiast artykuł profesora Lasoty [1] był inspiracją obecnych badań.

Jak już wspomnieliśmy, organizm ma tym większe szanse przeżycia w swoim środowisku, im bardziej jest do tego środowiska przystosowany. Aby powstawały coraz lepsze organizmy, musi być tworzona wielka liczba nowych odmian gatunku, które są następnie „testowane” w środowisku. Naturalna selekcja powoduje, że organizm słabo przystosowany umiera i jego kod DNA zanika, natomiast dobrze przystosowany ma większe szanse przetrwania, czyli rozpowszechnienia swojego kodu DNA w większej liczbie potomków. Procesy tworzenia i umierania są też do pewnego stopnia losowe. Jednak losowość ma swoje „preferencje” wynikłe za każdym razem z oceny stopnia dostosowania każdego osobnika do otoczenia. Ten aspekt losowości jest niezmiernie ważny — odróżnia on algorytmy ewolucyjne (tutaj genetyczne) od całkowicie przypadkowych metod, np. metody Monte Carlo. Proces tworzenia nowych osobników jest realizowany za pomocą modyfikacji kodu DNA w wyniku krzyżowania i mutacji. Główna idea w procesie tworzenia nowych populacji (generacji) polega na całkowitym oddzieleniu procesu tworzenia nowych osobników i procesu oceny ich przystosowania. Istotnie, te dwa mechanizmy, pozornie nie mając ze sobą nic wspólnego, są odpowiedzialne za cały świat przyrody.

W niniejszej pracy zdefiniowano prosty algorytm genetyczny w terminach skończonego multizbioru potencjalnych rozwiązań (osobników danej populacji), na którym są określone operacje krzyżowania, mutacji i selekcji, każda z pewnym prawdopodobieństwem. Złożenie tych operacji, działając na daną populację, tworzy nową populację. Istnienie funkcji przystosowania (dopasowania), określonej na osobnikach populacji, pozwala powiązać prawdopodobieństwo selekcji osobników do nowej populacji z wartościami, jakie funkcja dopasowania przyjmuje dla danego osobnika. Złożenie tych operacji spełnia własność Markowa, dzięki czemu istnieje analogia do operatora (macierzy) Markowa. Zbieżność ciągu operacji może być w pewnych sytuacjach badana metodami znanymi z teorii operatorów Markowa.

W pracy wprowadzamy tylko podstawowe i niezbędne — dla zrozumienia niniejszych wyników — pojęcia z algorytmów genetycznych, nie wdając się w dokładne wyjaśnienia. Dociekliwego czytelnika odsyłamy do [4] oraz przeglądowego artykułu Rowe'a [10] i łatwo dostępnych książek w języku polskim, które na początku wprowadzają od podstaw ideę algorytmów genetycznych, a następnie omawiają bardziej szczegółowo zagadnienia ważne dla tego tematu. Najważniejsze z nich to książki Michalewicza [5], Goldberga [6] i Cytowskiego [7]. Bardziej zaawansowane podejście z punktu widzenia pewnych układów dynamicznych przynoszą pozycje Vose [8] oraz polska monografia Schaefera [11]. Większość oznaczeń pochodzi właśnie z pozycji [8, 10].

Główny cel naszego artykułu to badanie zachowania asymptotycznego SGA i niezależności rozkładu granicznego od populacji początkowej. W pracy ograniczamy się do algorytmu o sekwencji selekcja–mutacja (2.11), ale sądzimy, że zmiana kolejności na mutacja–selekcja nie wpływa na prawdziwość naszych wyników, jeśli istnieje dla niej zależność analogiczna do (3.3), opisująca prawdopodobieństwo uzyskania wskazanej populacji, gdy startuje się z danej populacji początkowej.

**2. Prosty algorytm genetyczny jako układ dynamiczny.** Dzięki dobrze rozwiniętym metodom układów dynamicznych możemy się pokusić o przedstawienie podstawowych koncepcji algorytmów genetycznych w tym języku. Dla dobrego przedstawienia wszystkich wyników należy rozszerzyć listę potrzebnych dla tego celu pojęć, dodając pojęcia charakterystyczne dla algorytmów genetycznych.

Wśród badaczy zajmujących się algorytmami ewolucyjnymi (a w szczególności genetycznymi) przyjęto używać terminologii biologicznej. I tak przetwarzany w danej iteracji (w danym kroku) multizbiór jest nazywany populacją (ang. *population*), rozwiązania należące do przestrzeni potencjalnych rozwiązań nazywane są osobnikami (ang. *individual*) lub fenotypami, a zamiast o kroku obliczeń mówi się o pokoleniu (ang. *generation*). Nowe osobniki, zwane potomkami (ang. *offspring*), są otrzymywane z osobników z poprzedniego pokolenia, zwanych rodzicami (ang. *parents*) w wyniku działania operacji selekcji (ang. *selection*). Elementy przestrzeni kodowej nazywane są chromosomami (ang. *chromosomes*), ze względu na ich podobieństwo do chromosomów składających się z cząsteczek kwasu DNA, lub genotypami. Wyróżnione funkcjonalnie fragmenty chromosomu nazywane są genami (ang. *gene*). Operatory podstawowe to *krzyżowanie* (ang. *crossover*), jeżeli dokonuje się wymiana informacji między osobnikami, oraz *mutacja* (ang. *mutation*), która modyfikuje pojedynczego osobnika. Używana jest często zamiennie funkcja jakości, czy funkcja celu, zaś jej odpowiednik, przetransformowany do przestrzeni kodowej to *funkcja przystosowania* (ang. *fitness function*); wartość funkcji przystosowania dla konkretnego osobnika

nazywana jest jego przystosowaniem. Funkcja przystosowania wraz z topologią przestrzeni kodowej definiowaną przez wybrane operatory jest nazywana krajobrazem przystosowania (ang. *fitness landscape*).

**2.1. Zbiór możliwych populacji.** Prosty algorytm genetyczny (ang. *simple genetic algorithm*) (SGA) jest modelem binarnych algorytmów genetycznych BGA [4], w których ma miejsce binarne kodowanie (odzworowanie elementów przestrzeni poszukiwań) oraz stosuje się (losową) selekcję proporcjonalną poprzedzającą operację mutacji i krzyżowania. Do losowej selekcji wykorzystuje się rozkład prawdopodobieństwa wyznaczony przez funkcję przystosowania, niezależnie od przyjętych prawdopodobieństw dla punktowej mutacji i standardowego algorytmu krzyżowania, określanego dla dwóch osobników.

W algorytmie genetycznym składnikiem zależnym od rozwiązywanego zadania — oprócz funkcji przystosowania — jest sposób kodowania elementów przestrzeni rozwiązań w chromosomy. Wtedy działanie algorytmów mutacji i krzyżowania jest prawie niezależne od zadania, gdyż wprowadzane operatory krzyżowania i mutacji (zwane w [4] operatorami przemieszczenia) działają nie na rozwiązaniach, ale na elementach przestrzeni kodowej.

W binarnym algorytmie genetycznym działamy na elementach przestrzeni kodowej, która jest obrazem przestrzeni rozwiązań podległej pewnej operacji kodowania (binarnego). Elementy przestrzeni kodowej są tutaj reprezentowane przez binarne chromosomy, które mają tę samą długość  $l$ , a więc zbiorem wszystkich chromosomów, czyli przestrzenią kodową  $Z$ , jest zbiór  $Z = \{0, 1\}^l$ . Możemy, po uporządkowaniu wszystkich możliwych chromosomów, zapisać  $Z$  jako zbiór  $\{z_0, \dots, z_{s-1}\}$ , gdzie  $s = 2^l$ . W dalszej części naszego artykułu dla prostoty zamiast mówić o elemencie przestrzeni kodowej będziemy używać terminu *komórka*.

Populację, czyli skończony multizbiór o rozmiarze  $r$ , zwanym rozmiarem populacji (ang. *PopSize*), możemy utożsamiać z próbą losową powstałą zgodnie z zasadą losowania ze zwracaniem  $r$  elementów ze zbioru  $Z = \{0, 1\}^l$ . Składa się ona z pewnej liczby tych samych kopii elementów przestrzeni kodowej. Możemy populację spróbować utożsamiać z uporządkowaną  $s$ -tką liczb wymiernych, przy czym każdy element tej  $s$ -tki reprezentuje stosunek liczby kopii elementu w populacji do liczby wszystkich (w tym i tych powtarzających się) elementów multizbioru. Innymi słowy, jeśli  $a_k$  jest liczbą kopii elementu  $z_k$  (ze zbioru  $Z$ ) w populacji o rozmiarze  $r$ , to na  $k$ -tym miejscu wystąpi ułamek

$$(2.1) \quad p_k = \frac{a_k}{r},$$

przy czym

$$(2.2) \quad \sum_{k=0}^{s-1} p_k = 1.$$

Oznaczamy  $p = (p_0, \dots, p_{s-1})$ . Ta  $s$ -tka ułamków odpowiadająca populacji tworzy wektor częstości  $p$ . Czasami dla skrótów samo  $p$  będziemy nazywać populacją.

Własność (2.2) pozwala traktować poszczególne współrzędne wektora częstości  $p$  jako prawdopodobieństwa występowania danego elementu z przestrzeni kodowej  $Z$  w populacji. Tym samym  $p$  staje się wektorem probabilistycznym (por. [1]). Dalsze szczegóły można znaleźć w pozycji [8] oraz artykule przeglądowym [10].

Klasyczny mechanizm stosowania algorytmów genetycznych powoduje, że przejście z populacji  $p$  do  $p'$  jest efektem wielokrotnego zastosowania operatorów genetycznych (selekcji, mutacji, krzyżowania), a nie jest wynikiem jednokrotnego losowania o znanym rozkładzie. Wyznaczenie takiego rozkładu jest na ogół trudne, a niekiedy nieefektywne. Jednak działanie algorytmu genetycznego w kroku  $t$  można utożsamiać (rozumieć) jako kolejne jednokrotne losowanie nowej populacji  $q$  ze zbioru wszystkich możliwych populacji z odpowiednim rozkładem prawdopodobieństwa. Te kolejne losowania mogą być powtarzane, przy czym rozkład prawdopodobieństwa w kroku następnym jest uwarunkowany przez populację aktualną. Taki obraz działania algorytmu genetycznego można modelować jednorodnym łańcuchem Markowa [11], w którym przestrzenia stanów są populacje.

Warto rozróżnić populację i wektor, który charakteryzuje rozkład częstościowy komórek, składających się na nią, czyli prawdopodobieństwo ich występowania w multizbiorze. Realizacja algorytmu genetycznego operuje na populacjach i ich generowaniu w kolejnych krokach, choć przejście między populacjami jest losowe. Operator probabilistyczny, działając na konkretnej (wyjściowej) populacji (wektorze częstości), daje w wyniku nowy wektor probabilistyczny. Wektor ten opisuje prawdopodobieństwa pojawienia się nowej populacji w tym kroku. Nasza analiza dotyczy procesu losowego, a nie jego konkretnej realizacji, a tematem naszego artykułu są graniczne własności wektorów prawdopodobieństw.

Zbiór wszystkich możliwych populacji (wektorów częstości) oznaczamy przez  $\Lambda$ , tj.

$$(2.3) \quad \Lambda = \left\{ p \in \mathbb{R}^s : p_k \geq 0 \text{ jest wymierne } \forall k \text{ oraz } \sum_{i=0}^{s-1} p_i = 1 \right\}.$$

W tej definicji nie jest istotny rozmiar populacji  $r$ , gdyż nie dopuszczamy jedynie, aby współrzędne wektora  $p$  przyjmowały wartości niewymierne. W sytuacji, gdy analizujemy proces losowy, działający na wektorach probabilistycznych, ich wartości są przyjmowane w zbiorze

$$(2.4) \quad \bar{\Lambda} = \left\{ x \in \mathbb{R}^s : x_k \geq 0 \forall k \text{ oraz } \sum_{i=0}^{s-1} x_i = 1 \right\}.$$

Zauważmy, że gdy rozmiar populacji dąży do nieskończoności, zbiór możliwych wektorów częstości staje się gęsty w sympleksie  $\bar{\Lambda}$ .

**2.2. Operator selekcji.** Istotnym zadaniem przy badaniu dynamiki algorytmu genetycznego jest wyznaczenie prawdopodobnego rozkładu (zawartości) następnej populacji, gdy mamy daną aktualną populację, operatory genetyczne i ich charakterystyczne parametry. Zaczniemy od operatora selekcji.

Dana jest populacja  $p$  scharakteryzowana przez jej wektor częstości  $(p_0, \dots, p_{s-1})$  oraz funkcja przystosowania  $f : Z \rightarrow \mathbb{R}^+$ , która jest złożeniem funkcji jakości  $\phi$  z funkcją kodującą oraz — jeśli to było konieczne — z innymi funkcjami przekształcającymi funkcję jakości do wymaganych przez rozpatrywany problem własności (np. nieujemności czy ograniczenia od dołu itp). Zakładając stosowanie selekcji proporcjonalnej (por. [4, 5]), jak to jest wymagane w SGA, możemy wyznaczyć prawdopodobieństwo, że element  $z_k$  wystąpi w następnej populacji:

$$(2.5) \quad \frac{f(z_k)p_k}{\bar{f}(p)},$$

gdzie  $\bar{f}(p)$  jest średnim przystosowaniem populacji  $p$  wyznaczonym przez

$$(2.6) \quad \bar{f}(p) = \sum_{k=0}^{s-1} f(z_k)p_k.$$

Teraz możemy postarać się wyznaczyć nową populację i związany z nim wektor  $q$ , składający się z odpowiednich prawdopodobieństw. W tym celu okreśmy macierz diagonalną  $S$  o wymiarze  $s$ , w której na głównej przekątnej występują wartości funkcji kolejnych elementów przestrzeni kodowej, tzn.

$$(2.7) \quad S_{kk} = f(z_k).$$

Pozwala to na konsekwentny zapis w postaci

$$(2.8) \quad q = \mathcal{F}p = \frac{1}{\bar{f}(p)}Sp,$$

który daje nową populację, a jednocześnie określa rozkład prawdopodobieństwa występowania  $s$  komórek (chromosomów) w następnej populacji po zastosowaniu operatora selekcji. Zauważmy, że zapis ten ma zastosowanie w dwu przypadkach: w pierwszym, gdy wychodzimy od konkretnej populacji o znanej względnej liczbie kopii elementu w populacji, i wtedy  $p$  reprezentuje wektor częstości, oraz w drugim, gdy obiektem, na który działa operator selekcji, jest tylko rozkład prawdopodobieństwa występowania elementów przestrzeni kodowej w populacji, tzn.  $p$  jest wektorem probabilistycznym. Często w tym miejscu dla skrótów mówi się o rozkładzie prawdopodobieństwa populacji. Już w tym miejscu widać, że nawet jeśli startujemy z  $p \in \Lambda$ , wynik w (2.8) nie musi być w tym samym zbiorze, gdyż wartości elementów

macierzy  $S$  mogą być niewymierne. Tak więc bezpiecznie jest napisać, iż  $q \in \bar{\Lambda}$ .

Warto podkreślić, że dzięki takiej definicji selekcji, zawierającej wartości funkcji przystosowania, będziemy w stanie w następnym rozdziale określić operator przejścia  $T$  (por. (3.4)) działający na rozkładach prawdopodobieństwa (wektorach probabilistycznych  $u$ ) wszystkich populacji. Dzięki temu operator  $T$  staje się operatorem liniowym, zależnym od funkcji przystosowania.

**2.3. Operator mutacji.** Przejdźmy do operatora mutacji. Dla prostego operatora genetycznego naturalne jest rozpatrzenie najpierw mutacji binarnej, równomiernej, o parametrze  $\mu$ . Oznacza to, że dowolny gen w chromosomie może być zmutowany z prawdopodobieństwem  $\mu$ .

Wyjdźmy od dowolnego elementu  $z_j$ . Wiemy, że prawdopodobieństwo występowania tego elementu jest równe  $q_j$ . Prawdopodobieństwo przejścia w element  $z_i$  na drodze mutacji z populacji  $q$  wynosi

$$(2.9) \quad \sum_{j=0}^{s-1} U_{ij} q_j,$$

gdzie  $U_{ij}$  jest elementem macierzy  $U$  opisującej prawdopodobieństwa mutacji elementu  $z_j$  w element  $z_i$  w przypadku  $i \neq j$ ; gdy  $i = j$ , jest to prawdopodobieństwo przetrwania elementu  $z_i$  w trakcie mutacji.

Sposób wyznaczania elementów tej macierzy pokazuje następujący przykład. Gdy  $z_i$  różni się od  $z_j$  na  $c$  pozycjach, to

$$(2.10) \quad U_{ij} = \mu^c (1 - \mu)^{l-c}.$$

Składając operacje mutacji i selekcji, otrzymujemy zależność, wiążącą populację w chwili  $t + 1$ , oznaczaną tutaj przez  $p(t + 1)$ , z populacją  $p(t)$ :

$$(2.11) \quad p(t + 1) = U \circ \mathcal{F}p(t) = \frac{1}{\bar{f}(p(t))} U S p(t),$$

gdzie zmienna  $t$  nosi też nazwę *numeru populacji*.

Dla poprawności naszych wyników zawartych w następnych rozdziałach nie ma potrzeby ograniczać się do macierzy  $U$ , której elementy dane są przez (2.10). Wyniki będą poprawne dla przypadku ogólniejszego, w szczególności dla niebinarnych operatorów mutacji. Będziemy jedynie wymagać, by elementy macierzy  $U$  były nieujemne oraz by ich suma w każdej kolumnie była równa jeden. To oznacza, że macierz  $U$  przeprowadza wektory probabilistyczne w siebie, a więc jest macierzą Markowa [1]. Dla dowodu własności asymptotycznych algorytmów genetycznych będziemy dalej zakładać, że  $U_{jj} > 0$  dla  $j = 0, 1, \dots, s - 1$ . A to oznacza, w przypadku postaci (2.10), że prawdopodobieństwo  $\mu$  mutacji jest mniejsze od 1.



Większość wyników naszego artykułu dotyczy algorytmu genetycznego, w którym został pominięty następny element prostego algorytmu genetycznego, a mianowicie krzyżowanie.

**2.4. Operator krzyżowania.** Aby określić operator krzyżowania  $\mathbf{C}$ , musimy wprowadzić kilka dodatkowych pojęć. Niech macierze  $\mathbf{C}_0, \dots, \mathbf{C}_{s-1}$  będą określone w ten sposób, że element  $(i, j)$  macierzy  $\mathbf{C}_k$  oznacza prawdopodobieństwo, że element  $z_i$  przestrzeni kodowej skrzyżowany z elementem  $z_j$  wygeneruje element  $z_k$ .

Dla przybliżenia wprowadzanych pojęć rozpatrzmy pokrótce przypadek chromosomu o długości  $l = 2$ . Wówczas elementy przestrzeni kodowej są postaci

$$(2.12) \quad z_0 = 00, \quad z_1 = 01, \quad z_2 = 10, \quad z_3 = 11.$$

Gdy krzyżowanie jest jednolite, tzn. wszystkie elementy mogą podlegać krzyżowaniu ze wszystkimi, macierz  $\mathbf{C}_0$  ma postać

$$(2.13) \quad \mathbf{C}_0 = \begin{pmatrix} 1,0 & 0,5 & 0,5 & 0,25 \\ 0,5 & 0,0 & 0,25 & 0,0 \\ 0,5 & 0,25 & 0,0 & 0,0 \\ 0,25 & 0,0 & 0,0 & 0,0 \end{pmatrix}.$$

Macierze  $\mathbf{C}_k$  są symetryczne. Następnie tworzymy operator  $\mathbf{C}$ , który działa na dowolną populację  $p$  według wzoru

$$(2.14) \quad \mathbf{C}(p) = (p \cdot \mathbf{C}_0 p, \dots, p \cdot \mathbf{C}_{s-1} p),$$

gdzie kropka  $\cdot$  oznacza iloczyn skalarny dwóch wektorów z  $s$ -wymiarowej przestrzeni.

Działanie prostego algorytmu genetycznego [8, 10, 11] przy przejściu od danej populacji do następnej jest opisywane operatorem  $\mathcal{G}$  będącym złożeniem trzech operatorów: selekcji, mutacji i krzyżowania:

$$(2.15) \quad \mathcal{G} = \mathbf{C} \circ \mathbf{U} \circ \mathcal{F}.$$

Czytelnika zainteresowanego szczegółowym opisem tych mechanizmów odsyłamy do pozycji bibliografii [8, 10].

**3. Model z selekcją i mutacją dla populacji skończonych.** Niech  $p = (p_0, \dots, p_{s-1})$  będzie wektorem probabilistycznym. Gdybyśmy rozpatrywali  $p \in \bar{\Lambda}$ , wówczas operatory opisane w rozdziale 2 przeprowadzałyby zbiór  $\bar{\Lambda}$  w siebie. Jednak kiedy mamy do czynienia z konkretną populacją, to znaczy  $p$  jest wektorem częstości i  $p \in \Lambda$ , wówczas operatory te mogą wyprowadzać poza zbiór  $\Lambda$ . Działanie algorytmu genetycznego w pierwszym (a także w każdym następnym) kroku jest następujące. Jeśli mamy daną populację  $p$ , to losujemy ze zwracaniem  $r$  elementów ze zbioru  $Z$ , przy czym

prawdopodobieństwo wylosowania elementów  $z_0, \dots, z_{s-1}$  opisane jest wektorem  $\mathcal{G}(p)$ , gdzie

$$(3.1) \quad \mathcal{G}(p) = \frac{1}{f(p)} \mathbf{U} \mathbf{S} p.$$

Te  $r$  elementów to nasza nowa populacja  $q$ .

Oznaczmy przez  $W$  zbiór wszystkich możliwych populacji  $r$ -elementowych złożonych z elementów wybranych ze zbioru  $Z$ , przy czym elementy w populacji mogą się powtarzać. Zbiór ten jest skończony i jego moc oznaczamy przez  $M$ . Można wykazać, że

$$(3.2) \quad M = \binom{s+r-1}{s-1} = \binom{s+r-1}{r}.$$

Teraz w dowolny sposób numerujemy wszystkie populacje, tzn. mamy utożsamienie zbioru  $W$  z listą  $W = \{w^1, \dots, w^M\}$ . Każde  $w^k$ ,  $k = 1, 2, \dots, M$ , jest pewną populacją, dla której dotąd używaliśmy oznaczenia  $p$  w poprzednim rozdziale. Zgodnie z tym, co napisaliśmy w punkcie 2.1, populacji odpowiada jej wektor częstości bądź wektor probabilistyczny. To oznacza, że przy danej populacji  $p = w^k = (w_0^k, \dots, w_{s-1}^k)$ , liczba  $w_i^k$  dla  $i \in \{0, \dots, s-1\}$  oznacza prawdopodobieństwo wylosowania z populacji  $w^k$  komórki  $z_i$  (czyli względny udział komórki  $z_i$  w populacji  $w^k$ ).

Załóżmy, że zaczynamy naszą implementację algorytmu genetycznego od dowolnej populacji  $p = w^k$ . W następnym kroku każda z populacji  $w^1, \dots, w^M$  może wystąpić z prawdopodobieństwem, które jest wyznaczone dzięki dokonanej w poprzednim rozdziale analizie. Mianowicie, prawdopodobieństwo wystąpienia w następnym pokoleniu (kroku) populacji  $q$  o numerze, powiedzmy,  $l$ , tzn.  $w^l$ , jest równe

$$(3.3) \quad r! \prod_{j=0}^{s-1} \frac{(\mathcal{G}(p)_j)^{r q_j}}{(r q_j)!}.$$

Po dwóch krokach każda z populacji  $w^1, \dots, w^M$  będzie występować z pewnym prawdopodobieństwem, które jest dwukrotnym złożeniem powyższego wzoru; podobnie w trzecim kroku itd. Tak więc ma sens rozpatrywanie rozkładu prawdopodobieństwa, z jakim w kolejnych krokach pojawiają się odpowiednie populacje. Powyższy wzór daje możliwość efektywnego wyznaczenia wszystkich elementów macierzy określającej rozkład prawdopodobieństw pojawienia się  $M$  populacji w następnym kroku, gdy dany jest aktualny rozkład tych populacji. Przy tych oznaczeniach na numery populacji  $p$  i  $q$  element  $(l, k)$  tej macierzy będzie oznaczał prawdopodobieństwo przejścia w pojedynczym kroku populacji o numerze  $k$  w populację o numerze  $l$ . Dla naszych rozważań szczególnie ważne jest, że elementy macierzy są wyznaczone raz na zawsze, niezależnie od numeru kroku. Jest oczywiste, że różne

pary populacji opisywane są różnymi prawdopodobieństwami (por. (3.3)) (elementami macierzy).

Oznaczmy

$$\Gamma = \{x \in \mathbb{R}^M : \forall k \ x_k \geq 0 \text{ oraz } \|x\| = 1\},$$

gdzie  $\|x\| = x_1 + \dots + x_M$  dla  $x = (x_1, \dots, x_M)$ . Poszczególne współrzędne wektora  $x$  reprezentują prawdopodobieństwa występowania danej populacji z przestrzeni wszystkich  $M$  populacji  $W$ . Zbiór  $\Gamma$  składa się ze wszystkich możliwych rozkładów prawdopodobieństwa dla  $M$  populacji. Tak więc opisana przez nas implementacja przeprowadza w każdym kroku zbiór  $\Gamma$  w siebie.

Wprowadźmy teraz na zbiorze  $\Gamma$  podstawowy dla dalszych rozważań probabilistyczny operator przejścia (ang. *transition operator*)

$$(3.4) \quad T(\cdot) : \mathbb{N} \times \Gamma \rightarrow \Gamma,$$

którego działanie określiliśmy opisowo powyżej. Konkretniej, jeśli  $u \in \Gamma$ , to przez  $T(t)u = ((T(t)u)_1, \dots, (T(t)u)_M)$  oznaczamy rozkład prawdopodobieństwa dla  $M$  populacji w kroku numer  $t$ , jeśli zaczynaliśmy naszą implementację prostego algorytmu genetycznego  $\mathcal{G}$  (por. (3.1)) od rozkładu prawdopodobieństwa dla  $M$  populacji równego  $u = (u_1, \dots, u_M) \in \Gamma$ , przy  $t$ -krotnym zastosowaniu powyższego rozumowania. Zauważmy, że  $(T(t)u)_k$  dla  $k \in \{1, \dots, M\}$  oznacza prawdopodobieństwo wystąpienia w kroku numer  $t$  populacji  $w^k$ . Ze względu na definicję  $\mathcal{G}(p)$  w (3.1), (3.3) oraz uwagę zrobioną na końcu punktu 2.2, poświęconego operatorowi selekcji, operator przejścia  $T(t)$  jest liniowy dla każdego naturalnego  $t$ .

Aby przybliżyć sposób działania algorytmów genetycznych, warto odwołać się do obrazu często występującego w błędzeniu losowym punktów po pewnym zbiorze [1], gdyż samo działanie (prostego) algorytmu genetycznego jest podobne do następującego schematu. W przestrzeni wszystkich możliwych populacji  $\bar{\Lambda}$  wędruje punkt, którego następne losowe położenie jest efektem działania SGA na bieżącej populacji. Wiemy, że w chwili początkowej punkt był jedną z możliwych populacji, ponumerowanych liczbami  $1, 2, \dots, M$ , odpowiednio z prawdopodobieństwami  $u_1, u_2, \dots, u_M$ . Wiemy też, że jeśli w chwili  $t$  (w pokoleniu o numerze  $t$ ) mieliśmy populację  $p$  o numerze  $k$ , tzn. populację  $w^k$ , to prawdopodobieństwo, że w chwili  $t+1$  (w pokoleniu o numerze  $t+1$ ) osiągniemy populację  $q$  o numerze  $l$ , tzn. populację  $w^l$ , wynosi  $p_{lk}$ , i prawdopodobieństwo to nie zależy od numeru kroku, w którym to przejście następuje. Przy tych właśnie oznaczeniach prawdopodobieństwo  $p_{lk}$  jest dane wzorem (3.3).

Utwórzmy nieujemną, kwadratową macierz  $\mathbf{T}$  o rozmiarze  $M$  i elementach  $p_{lk}$ ,  $l, k = 1, 2, \dots, M$ . Wówczas rozkład prawdopodobieństwa położe-

nia naszego punktu w kroku  $t$  dany jest wzorem

$$\mathbf{T}^t u, \quad t = 0, 1, 2, \dots$$

Wzór (3.3) definiuje nam każdy element macierzy opisującej prawdopodobieństwa przejścia między dowolnymi populacjami. Elementy te nie zależą od numeru kroku algorytmu. Wprowadzony powyżej operator przejścia  $T(t)$  jest związany z powyższą macierzą zależnością

$$T(t) = \mathbf{T}^t.$$

Tak określona macierz jest macierzą Markowa. Fakt ten pozwala na wykorzystanie dorobku teorii operatorów Markowa do analizy zbieżności algorytmów genetycznych.

Niech  $e_k \in \Gamma$  będzie wektorem, który na  $k$ -tym miejscu ma jedynkę oraz zero na pozostałych miejscach. Tak więc  $e_k$  jest rozkładem, w którym populacja  $w^k$  występuje z prawdopodobieństwem 1. Zapis  $T(t)w^k$  będziemy rozumieli jako

$$(3.5) \quad T(t)w^k = T(t)e_k.$$

Jest to więc przypadek, gdy nasze doświadczenie rozpoczynamy od konkretnej populacji  $w^k$ .

W dalszym ciągu będziemy zakładać, że zachodzi następujący warunek:

$$(3.6) \quad U_{jj} > 0 \quad \text{dla } j \in \{0, \dots, s-1\}.$$

Warunek ten, w przypadku mutacji binarnej, opisanej w punkcie 2.2 i danej wzorem (2.10), jest w rzeczywistości ograniczeniem na dopuszczalne wartości parametru  $\mu$ . Parametr ten charakteryzuje prawdopodobieństwo mutacji pojedynczego genu w chromosomie (elemente przestrzeni kodowej), nazywanym komórką. Spojrzenie na wzór (2.10) mówi, że spełnienie nierówności (3.6) jest możliwe przy warunku

$$(3.7) \quad 0 \leq \mu < 1.$$

Załóżmy teraz, że mamy dany dowolny rozkład prawdopodobieństwa dla  $M$  populacji  $u = (u_1, \dots, u_M) \in \Gamma$ . Łatwo obliczyć, że wtedy dla  $i \in \{0, \dots, s-1\}$  prawdopodobieństwo wylosowania komórki  $z_i$  wynosi

$$(3.8) \quad \sum_{k=1}^M w_i^k \cdot u_k,$$

gdzie  $w_i^k$  to prawdopodobieństwo wylosowania z  $k$ -tej populacji komórki  $z_i$ , a  $u_k$  prawdopodobieństwo wystąpienia  $k$ -tej populacji. *Populacją oczekiwaną* będziemy nazywać wektor z przestrzeni  $\mathbb{R}^s$ , którego  $i$ -ta współrzędna jest dana wzorem (3.8). Ponieważ  $u_k \geq 0$  i  $w_i^k \geq 0$  dla  $k \in \{1, \dots, M\}$ ,

$i \in \{0, \dots, s-1\}$  oraz

$$\sum_{i=0}^{s-1} \left( \sum_{k=1}^M u_k \cdot w_i^k \right) = \sum_{k=1}^M u_k \left( \sum_{i=0}^{s-1} w_i^k \right) = \sum_{k=1}^M u_k = 1,$$

więc nasz wektor należy do  $\bar{\Lambda}$ . Z (3.8) wynika, że populacja oczekiwana dana jest wzorem

$$(3.9) \quad \sum_{k=1}^M w^k \cdot u_k.$$

Oczywiście populacja oczekiwana może nie być żadną z możliwych populacji  $r$ -elementowych.

Dla każdego  $u \in \Gamma$  oraz dla dowolnego  $t$  mamy dany pewien rozkład prawdopodobieństwa dla  $M$  populacji  $T(t)u$ . Stąd wynika, że mamy daną również populację oczekiwaną w tym kroku.

Oznaczmy przez  $R(t)u = ((R(t)u)_0, \dots, (R(t)u)_{s-1})$  populację oczekiwaną w kroku  $t$ , jeśli zaczęliśmy nasze doświadczenie od rozkładu  $u \in \Gamma$ . Mamy oczywiście  $R(t)u \in \bar{\Lambda}$ .

DEFINICJA 3.1. Będziemy mówili, że model jest *asymptotycznie stabilny*, jeśli istnieje  $u^* \in \Gamma$  takie, że

$$(3.10) \quad T(t)u^* = u^* \quad \text{dla } t = 0, 1, \dots,$$

$$(3.11) \quad \lim_{t \rightarrow \infty} \|T(t)u - u^*\| = 0 \quad \forall u \in \Gamma.$$

Ponieważ dla  $k \in \{1, \dots, M\}$  mamy

$$(3.12) \quad |(T(t)u)_k - u_k^*| \leq \|T(t)u - u^*\|,$$

więc z (3.11) otrzymujemy

$$(3.13) \quad \lim_{t \rightarrow \infty} (T(t)u)_k = u_k^*.$$

To oznacza, że prawdopodobieństwo wystąpienia populacji  $w^k$  w kroku numer  $t$  zmierza do pewnej ustalonej liczby  $u_k^*$ , niezależnej od początkowego rozkładu  $u$ . Ma to miejsce również w szczególnym przypadku, gdy naszą implementację rozpoczęliśmy od jednej konkretnej populacji  $p = w^j$ .

TWIERDZENIE 3.1. *Jeśli model jest asymptotycznie stabilny, to*

$$(3.14) \quad \lim_{t \rightarrow \infty} \|R(t)u - p^*\| = 0 \quad \text{dla } u \in \Gamma,$$

gdzie  $p^* \in \bar{\Lambda}$  jest populacją oczekiwaną odpowiadającą rozkładowi  $u^*$ .  
W szczególności

$$(3.15) \quad \lim_{t \rightarrow \infty} \|R(t)p - p^*\| = 0 \quad \text{dla } p \in W.$$

Dowód. Z (3.9) mamy

$$R(t)u = \sum_{i=1}^M w^i \cdot (T(t)u)_i$$

oraz

$$p^* = \sum_{i=1}^M w^i \cdot u_i^* .$$

Tak więc

$$\begin{aligned} \|R(t)u - p^*\| &= \left\| \sum_{i=1}^M w^i \cdot (T(t)u)_i - \sum_{i=1}^M w^i \cdot u_i^* \right\| \\ &= \sum_{j=0}^{s-1} \left| \sum_{i=1}^M w_j^i \cdot (T(t)u)_i - \sum_{i=1}^M w_j^i \cdot u_i^* \right| \leq \sum_{j=0}^{s-1} \sum_{i=1}^M w_j^i |(T(t)u)_i - u_i^*| \\ &= \sum_{i=1}^M \left( \sum_{j=0}^{s-1} w_j^i \right) |(T(t)u)_i - u_i^*| = \|T(t)u - u^*\|. \end{aligned}$$

Stąd na podstawie (3.11) zachodzi (3.14). Biorąc pod uwagę oznaczenie (3.5), wzór (3.15) jest szczególnym przypadkiem (3.14). ■

Z twierdzenia 3.1 wynika, że jeśli model jest asymptotycznie stabilny, to populacja oczekiwana stabilizuje się, zmierzając do  $p^* \in \bar{\Lambda}$ , niezależnie od warunków początkowych.

Będziemy mówili, że z komórki  $z_a$  można otrzymać  $z_b$  w jednej mutacji (lub w jednym kroku) z dodatnim prawdopodobieństwem, jeśli  $U_{ba} > 0$ . Będziemy mówili, że z komórki  $z_a$  można otrzymać komórkę  $z_b$  z dodatnim prawdopodobieństwem w  $n$  mutacjach (lub w  $n$  krokach), jeśli istnieją komórki  $z_{i_0}, \dots, z_{i_n}$  takie, że  $z_{i_0} = z_a$ ,  $z_{i_n} = z_b$  oraz każdą komórkę  $z_{i_j}$  dla  $j = 1, \dots, n$  można otrzymać z komórki  $z_{i_{j-1}}$  w jednym kroku z dodatnim prawdopodobieństwem.

Twierdzenie 3.1 ma fundamentalne znaczenie dla analizy zbieżności algorytmów genetycznych. W pracy [14] pokazano asymptotyczną stabilność modelu Markowa prostego algorytmu genetycznego, tzn. poszerzonego w stosunku do obecnego przypadku o operator krzyżowania.

DEFINICJA 3.2. Model nazywamy *punktowo asymptotycznie stabilnym*, jeśli istnieje populacja  $w^j$  taka, że

$$(3.16) \quad \lim_{t \rightarrow \infty} (T(t)u)_j = 1 \quad \text{dla } u \in \Gamma.$$

Warunek (3.16) definicji 3.2 oznacza, że w kolejnych krokach prawdopodobieństwo pojawienia się populacji innej niż  $w^j$  zmierza do zera. Jest to szczególny przypadek asymptotycznej stabilności, gdzie

$$u^* = e_j.$$

**TWIERDZENIE 3.2.** *Model jest punktowo asymptotycznie stabilny wtedy i tylko wtedy, gdy istnieje dokładnie jedna komórka  $z_a$  o tej własności, że można ją otrzymać z dowolnej komórki w skończonej liczbie kroków z dodatnim prawdopodobieństwem. W takiej sytuacji populacja  $w^j$  składa się wyłącznie z komórek  $z_a$  oraz zachodzi równość*

$$(3.17) \quad T(t)w^j = w^j.$$

*Ponadto prawdopodobieństwo wystąpienia w kroku numer  $t$  populacji innej niż  $w^j$  zmierza do zera w postępie geometrycznym, tzn. istnieją  $\lambda \in (0, 1)$  i  $D \in \mathbb{R}_+$  takie, że*

$$(3.18) \quad \sum_{\substack{i=1 \\ i \neq j}}^M (T(t)u)_i \leq D \cdot \lambda^t. \blacksquare$$

Dowody twierdzeń i wspomagających je lematów można znaleźć w artykułach [14, 15, 16].

Liczby  $\lambda$  i  $D$  dla konkretnego modelu możemy wyznaczyć. Będzie to tematem następnych artykułów. Wzór (3.18) głosi, że gdybyśmy w rzeczywistości postępowali według opisanego przez nas algorytmu, to populacja  $w^j$  pojawi się praktycznie po skończonej liczbie kroków. Ze wzoru (3.17) wynika, że z populacji  $w^j$  otrzymujemy  $w^j$  z prawdopodobieństwem równym 1. Jeśli więc  $w^j$  raz się pojawi, to od tego momentu będziemy mieli stale populację  $w^j$ .

Z twierdzenia 3.2 wynika, że powyższa zbieżność do jednej populacji może pojawić się tylko przy bardzo szczególnych założeniach. To uzasadnia sens badania asymptotycznej stabilności, jak w definicji 3.1.

**DEFINICJA 3.3.** Przez *komórkę osiągalną* rozumiemy taką komórkę  $z_a \in Z$ , którą można otrzymać z dowolnej innej w skończonej liczbie kroków z dodatnim prawdopodobieństwem. Oznaczmy przez  $Z^*$  zbiór komórek  $z_a$  o tej własności.

**TWIERDZENIE 3.3.** *Model jest asymptotycznie stabilny wtedy i tylko wtedy, gdy  $Z^* \neq \emptyset$ .  $\blacksquare$*

**TWIERDZENIE 3.4.** *Załóżmy, że model jest asymptotycznie stabilny. W tej sytuacji zachodzi następująca równoważność:  $u_k^* > 0$  wtedy i tylko wtedy, gdy populacja  $w^k$  składa się wyłącznie z komórek należących do zbioru  $Z^*$ .*

**WNIOSEK.** *Jeśli  $Z^* = Z$ , to  $\forall_{k \in \{1, \dots, M\}} u_k^* > 0$ .  $\blacksquare$*

Podsumujmy nasze wyniki:

1.  $Z^* = \emptyset \Rightarrow$  brak asymptotycznej stabilności;
2.  $Z^* \neq \emptyset \Rightarrow$  asymptotyczna stabilność, przy czym:
3.  $Z^*$  jest zbiorem jednoelementowym  $\Rightarrow$  punktowa asymptotyczna stabilność (zbieżność w pewnym sensie do jednej populacji);

4.  $Z^*$  zawiera więcej niż jeden element  $\Rightarrow$  asymptotyczna stabilność, ale brak punktowej asymptotycznej stabilności (stabilizuje się prawdopodobieństwo otrzymania poszczególnych populacji w kolejnych krokach, lecz brak zbieżności do jednej populacji).

UWAGA. Dla konkretnego modelu spełniającego założenia twierdzenia 3.4 można otrzymać efektywne oszacowanie  $u_k^*$  i  $p_k^*$  z dołu.

**4. Wnioski końcowe.** W ostatnich latach gwałtownie wzrasta zainteresowanie uniwersalnymi algorytmami optymalizacyjnymi, które wymagają tylko ograniczonej wiedzy o rozwiązywanym problemie. Inspiracją dla tego typu algorytmów jest obserwacja procesów optymalizacyjnych zachodzących w naturze. Szeroki zakres zastosowania tej metody rozwiązywania złożonych problemów optymalizacyjnych sprawia, że istnieje duża potrzeba uzasadnienia teoretycznego ich własności. Szczególnie palące jest zbadanie zbieżności tych algorytmów, ich uwarunkowań oraz ograniczeń ich stosowalności.

Przedstawione w pracy podejście pozwala na zastosowanie dobrego rozwiniętego aparatu teorii operatorów Markowa do badania asymptotycznego zachowania iteracji operatorów. Okazuje się, że w badanych przez nas problemach operatory te są zadane macierzami Markowa.

Uzyskane wyniki dowodzą zbieżności prostego algorytmu genetycznego w sensie probabilistycznym. Zdaniem autorów istnieje możliwość uogólnienia obecnych rezultatów na przypadek odmiennej kolejności operatorów mutacji-selekcji. W ten sposób, choć powoli, przybliżamy się do wyjaśnienia mechanizmów optymalizacji realizowanych przez algorytmy genetyczne. Pozwala to na uzasadnione podejmowanie dalszych badań nad wpływem wielkości parametrów algorytmu, takich jak liczebność populacji, prawdopodobieństwo mutacji, liczba krzyżowań, na zachowanie asymptotyczne i zbieżność algorytmów.

Artykuł powstał we współpracy, w której dwóch współautorów (W.K., S.K.) było wspieranych projektem KBN Nr 3 T11 C007 28. Autorzy Ci są wdzięczni profesorowi Zbigniewowi Michalewiczowi za cenne uwagi i dyskusje, zaś Recenzentowi za inspirujący komentarz do wstępnej wersji pracy.

#### Bibliografia

- [1] A. Lasota, *Asymptotyczne własności półgrup operatorów Markowa*, Mat. Stos. 3 (2002), 39–51.
- [2] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [3] R. B. Hollstien, *Artificial Genetic Adaptation in Computer Control Systems*, Ph.D. Thesis, University of Michigan, 1971.
- [4] P. Kieś, Z. Michalewicz, *Podstawy algorytmów genetycznych*, Mat. Stos. 1 (2000), 68–91.



- [5] Z. Michalewicz, *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, WNT, Warszawa, 1996.
- [6] D. E. Goldberg, *Algorytmy genetyczne i ich zastosowania*, WNT, Warszawa, 1995.
- [7] J. Cytowski, *Algorytmy genetyczne: podstawy i zastosowania*, Seria: Problemy Współczesnej Nauki — Teoria i Zastosowania Nr 18, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1996.
- [8] M. D. Vose, *The Simple Genetic Algorithm: Foundation and Theory*, MIT Press, Cambridge, MA, 1999.
- [9] A. Lasota, J. A. Yorke, *Exact dynamical systems and the Frobenius–Perron operator*, Trans. Amer. Math. Soc. 273 (1982), 375–384.
- [10] J. E. Rowe, *The dynamical system models of the simple genetic algorithm*, w: Theoretical Aspects of Evolutionary Computing, L. Kallel i in. (red.), Springer, 2001, 31–57.
- [11] R. Schaefer, *Podstawy genetycznej optymalizacji globalnej*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2002.
- [12] R. Rudnicki, *On asymptotic stability and sweeping for Markov operators*, Bull. Polish Acad. Sci. Math. 43 (1995), 245–262.
- [13] J. Socała, *Asymptotic behaviour of the iterates of nonnegative operators on a Banach lattice*, Ann. Polon. Math. 68 (1998), 1–16.
- [14] S. Kotowski, J. Socała, W. Kosiński, Z. Michalewicz, *Markovian model of simple genetic algorithms and its asymptotic behaviour*, przesłana do publikacji do Fund. Inform., 2005.
- [15] J. Socała, *Markovian approach to genetic algorithms*, przesłana do publikacji, 2005.
- [16] S. Kotowski, *Klasyfikacja algorytmów genetycznych*, w przygotowaniu, 2005.

Jolanta Socała  
 Instytut Matematyki, Uniwersytet Śląski  
 ul. Bankowa 14, 40-007 Katowice  
 E-mail: jsocala@ux2.math.us.edu.pl

Stefan Kotowski  
 Zakład Sterowania i Dynamiki Układów  
 Instytut Podstawowych Problemów Techniki PAN  
 ul. Świętokrzyska 21, 00-049 Warszawa  
 E-mail: skot@ippt.gov.pl

Witold Kosiński  
 Centrum Badawcze  
 Polsko-Japońska Wyższa Szkoła  
 Techniki Komputerowych  
 ul. Koszykowa 86, 02-008 Warszawa  
 E-mail: wkos@pjwstk.edu.pl  
 Instytut Mechaniki Środowiska  
 i Informatyki Stosowanej  
 Uniwersytet Kazimierza Wielkiego  
 ul. Chodkiewicza 30, 85-064 Bydgoszcz

---

**Abstract.** The simple genetic algorithm (SGA) and its convergence analysis are main subjects of the article. The SGA is defined on a finite multi-set of potential problem solutions (individuals) together with crossover, mutation and selection operators, each with prescribed probability. The selection operation acts on the basis of the fitness function defined on potential solutions (individuals), and is fundamental for the problem considered. Generation of a new population from the given one is realized by the action of the composition of those operators. The composition is written in the form of a transition operator acting on probability vectors which describe probability distributions of each population. The transition operator is a Markov one. Thanks to the well-developed theory of Markov operators [9, 12, 13] new conditions for stability of the transition operator are formulated. The results obtained are related to the class of genetic operators.

**Key words:** population, crossover, mutation, selection, Markov operator, stability, convergence, limit distribution, simple genetic algorithm.