

Zagadnienia symulacji procesu ewolucji w systemach wielo-agentowych

Rafał DREŻEWSKI *
e-mail: drezew@agh.edu.pl

Streszczenie

Klasyczne programy ewolucyjne, pomimo obfitującej w liczne sukcesy historii badań i zastosowań, charakteryzują się pewnymi ograniczeniami, które są przyczyną problemów z wprowadzaniem nowych, wzorowanych na ewolucji biologicznej, mechanizmów. Realizacja procesu ewolucji w systemach wielo-agentowych oferuje nam dwie podstawowe korzyści: 1) uniknięcie ograniczeń klasycznych programów ewolucyjnych oraz 2) umożliwienie populacji agentów adaptacji do zmian zachodzących w środowisku. *Ewolucyjne systemy wielo-agentowe (Evolutionary Multi-Agent Systems — EMAS)*, będące rezultatem realizacji procesu ewolucji w systemach wielo-agentowych, mogą być traktowane zarówno, jako nowe podejście do konstrukcji programów ewolucyjnych, jak i nowy kierunek rozwoju systemów wielo-agentowych. W pracy niniejszej przedstawione zostały wybrane zagadnienia związane z realizacją ewolucyjnych systemów wielo-agentowych. Zaprezentowana została metodologia symulacji procesu ewolucji w systemach wielo-agentowych z wykorzystaniem modelu opartego na koncepcji M-Agenta oraz systemu symulacyjnego SWARM. Istotną część pracy stanowi zastosowanie tej metodologii do realizacji ewolucyjnego wielo-agentowego systemu predykcyjnego. Przedyskutowano również wyniki symulacji przeprowadzonych z wykorzystaniem tego systemu.

1 Wstęp

Programy ewolucyjne mają za sobą już ponad 30 letnią historię badań. Zostały one z powodzeniem zastosowane w wielu dziedzinach, takich jak: optymalizacja funkcji, przetwarzanie obrazów, sztuczna inteligencja, genetyczne systemy uczące się (GBML — Genetic-based Machine Learning Systems), teoria gier [7, 11].

Klasyczne programy ewolucyjne (jaki zostały zaprezentowane między innymi w pracach [7, 11]) posiadają następujące — nierealistyczne z punktu widzenia ewolucji biologicznej, na której są przecież wzorowane — cechy:

*Katedra Informatyki, Akademia Górniczo-Hutnicza, Kraków.

1. Występuje tam jeden wspólny algorytm odpowiedzialny za selekcję, reprodukcję, krzyżowanie, mutację oraz inne operacje ewolucyjne. Prowadzi to do nieuchronnej centralizacji całego procesu ewolucji oraz do występowania jednego rytmu w procesie ewolucji (w kolejnych fazach biorą udział równocześnie wszystkie byty).
2. Byty biorące udział w procesie ewolucji uproszczone są do genomu, który manipulowany jest przez wspólny algorytm. Uniemożliwia to samodzielne podejmowanie decyzji, rywalizację, współdziałanie i tym samym czyni niemożliwym występowanie relacji społecznych.
3. W większości zastosowań programów ewolucyjnych mamy do czynienia wyłącznie z wykorzystywaniem podstawowych operatorów ewolucyjnych, takich jak *reprodukcja*, *krzyżowanie* i *mutacja*. Próby wprowadzenia bardziej zaawansowanych operatorów nie zakończyły się pełnym sukcesem, a wprowadzenie niektórych mechanizmów ewolucji nie powiodło się w ogóle. Właściwość ta wydaje się być nieuchronnie związana z poprzednimi dwoma.

Omówione powyżej, charakterystyczne dla klasycznych programów ewolucyjnych, cechy są oczywiście całkowicie niezgodne z tym, co obserwujemy w rzeczywistym procesie ewolucji [9, 14, 5, 13, 8]. W rzeczywistym procesie ewolucji dobór naturalny działa na organizmy, które są autonomicznymi bytami i które mają możliwość samodzielnego działania. Ograniczenie się w programach ewolucyjnych do samego genomu jest daleko idącym i powodującym przeróżne komplikacje uproszczeniem. Już nawet tak prosty operator ewolucyjny jak *reprodukcja* jest wprowadzany w sposób nienaturalny, gdyż byty biorące udział w procesie ewolucji nie są w stanie samodzielnie wyszukiwać sobie partnerów.

Rozmnażanie płciowe (i związane z nim istnienie diploidalnego zestawu chromosomów), którego występowanie niesie ze sobą olbrzymie korzyści dla gatunku oraz takie siły napędowe ewolucji jak „Hipoteza Czerwonej Królowej”, czy dobór płciowy nie są możliwe do wprowadzenia, gdy byty biorące udział w procesie ewolucji zredukowane są wyłącznie do genomu. „Hipoteza Czerwonej Królowej” wymaga istnienia wielu gatunków, których osobniki biorą udział w „wyścigu zbrojeń” w układach pasożyt–żywiciel, drapieżnik–ofiara lub o dostęp do ograniczonych zasobów. Dobór płciowy również zakłada zdolność rywalizacji oraz możliwość dokonywania wyborów przez organizmy [9].

Jeżeli chodzi o mechanizmy specjacji, to niezbędne wydaje się tutaj zarówno umożliwienie organizmom samodzielnego działania, jak i wprowadzenie środowisk z panującymi w nich różnymi warunkami wraz z możliwością migracji organizmów pomiędzy nimi. Możliwe stało by się wtedy powstawanie nowych gatunków dzięki wykorzystaniu mechanizmów specjacji *alloptrycznej* (czyli izolowania geograficznego małych peryferyjnych grup macierzystego gatunku) lub też specjacji *symptrycznej* poprzez zróżnicowanie warunków panujących w poszczególnych fragmentach siedliska [9].

Możliwość samodzielnego działania bytów biorących udział w procesie ewolucji doprowadzi również do pojawienia się możliwości wyboru pomiędzy rywalizacją a współpracą, co spowoduje powstawanie relacji społecznych.

Otwarcie się nowych perspektyw przed programami ewolucyjnymi związane jest niewątpliwie z urealnieniem symulowanego procesu ewolucji. W tym celu należy proces ewolucji zdecentralizować i dać bytom biorącym w nim udział możliwość samodzielnego działania i podejmowania decyzji. Umożliwi to wprowadzenie nowych mechanizmów napędowych ewolucji.

Jedną z możliwości prowadzących do tego celu jest realizacja procesu ewolucji w *systemie wielo-agentowym (Multi-Agent System — MAS)* [3]. Związane z tym korzyści są następujące:

1. W procesie tworzenia naszego systemu możemy korzystać z istniejących modeli systemu wielo-agentowego. Powoduje to, iż byty (agenci) biorące udział w procesie ewolucji, środowisko ich działania, relacje agent-agent oraz agent-środowisko są dobrze zdefiniowane.
2. Proces ewolucji jest zdecentralizowany i nie przebiega wspólnym dla wszystkich, biorących w nim udział, agentów rytmem.
3. Agenci mogą samodzielnie działać i oddziaływać na środowisko oraz na siebie nawzajem. Powoduje to, iż możliwe jest powstawanie relacji społecznych w populacji agentów.
4. Proces selekcji jest zdecentralizowany ponieważ agenci mogą rywalizować pomiędzy sobą o dostęp do ograniczonych zasobów środowiska.
5. Poprzez wprowadzenie procesu ewolucji do systemu wielo-agentowego dostarczamy populacji agentów najlepszy możliwy mechanizm adaptacji do zmieniających się warunków środowiska, w którym żyją i działają.

Ewolucyjne systemy wielo-agentowe (Evolutionary Multi-Agent Systems — EMAS), będące rezultatem realizacji procesu ewolucji w systemach wielo-agentowych, mogą być traktowane zarówno, jako nowe podejście do konstrukcji programów ewolucyjnych, jak i nowy kierunek rozwoju systemów wielo-agentowych.

2 Kilka uwag metodologicznych

Badania naukowe prowadzone z wykorzystaniem symulacji komputerowych mają często na celu zaobserwowanie i zbadanie zjawisk, dla których nie jesteśmy w stanie sformułować odpowiedniego modelu matematycznego. W przypadku badań takich, możliwe jest na ogół wyłącznie sformułowanie modelu części składowych badanego systemu (podsystemów) oraz zdefiniowanie pewnych podstawowych relacji pomiędzy nimi. Jednak relacje lub zjawiska będące naprawdę przedmiotem

naszego zainteresowania nie są nigdzie zakodowane *explicite*. *Wylaniają się* one raczej podczas trwania symulacji i stają się wtedy możliwe do zaobserwowania przez badacza. Wszystko to wynika z faktu, że w badaniach tego rodzaju mamy do czynienia z pewną ustrukturowaną całością, która jest czymś więcej niż tylko prostą sumą swoich części składowych. Charakterystyczne dla takiej struktury właściwości wylaniają się jako rezultat skomplikowanych interakcji pomiędzy jej częściami składowymi tylko wtedy, gdy części te umieszczone są w ramach całości. Natomiast specyficzne dla takiej struktury cechy lub zjawiska nie mogą zostać wydedukowane po rozłożeniu struktury na części składowe i sformułowaniu ich modeli matematycznych. Dzieje się tak, ponieważ będące przedmiotem naszego zainteresowania zjawiska wylaniają się na poziomie wyższym niż ten, na którym formułujemy nasze modele. Fakty te powodują, że zjawiska tego typu — czyli właściwości systemu będącego pewną ustrukturowaną całością — mogą być obserwowane i badane wyłącznie przy wykorzystaniu technik symulacji komputerowej.

Rozważania powyższe odnoszą się również do badań nad sztuczną inteligencją. Nie zostanie ona z pewnością nigdy osiągnięta poprzez skonstruowanie scentralizowanego, skomplikowanego i rozbudowanego algorytmu. Sztuczna inteligencja wyłoni się raczej jako efekt skomplikowanych interakcji pomiędzy ogromną ilością stosunkowo prostych bytów, które podlegają procesowi ewolucji i dzięki temu mogą adaptować się do zmieniających się warunków środowiska. Jest to bardzo naturalny sposób powstawania inteligencji i wydaje się, iż w taki właśnie sposób rodzi się inteligencja naturalna.

Ogólny program badań nad złożonymi zjawiskami dynamicznymi z wykorzystaniem symulacji komputerowych powinien spełniać następujące warunki [12]:

1. Zawsze musimy być świadomi faktu, że symulacja generuje dynamiczne zjawiska na poziomie wyższym niż ten, na którym opisywane są elementarne części składowe systemu i ich wzajemne interakcje.
2. Musimy posiadać metody identyfikacji elementów składowych systemu generującego zjawiska będące przedmiotem naszego zainteresowania.
3. Niezbędne jest sformułowanie modeli podsystemów (definiujących elementarne podsystemy oraz interakcje element-element oraz obiekt-obiekt).
4. Musimy stworzyć pewną strukturę, w ramach której osadzimy reprezentację naszego systemu, tak aby możliwe było wygenerowanie i analiza zjawisk będących przedmiotem naszego zainteresowania.

Wszystko to, co zostało powiedziane powyżej ogólnie o badaniach prowadzonych z wykorzystaniem symulacji komputerowych, odnosi się również do badań prowadzonych w dziedzinie ewolucyjnych systemów wielo-agentowych. W systemach tych stało się możliwe wprowadzenie i badanie nowych mechanizmów wzorowanych na ewolucji biologicznej. Można wymienić tu takie mechanizmy, jak specjacja, rywalizacja pomiędzy gatunkami, zachowania społeczne. Badanie wpływu

tych mechanizmów na przebieg procesu ewolucji w systemie wielo-agentowym może być przeprowadzone wyłącznie z wykorzystaniem symulacji komputerowych.

Z prowadzeniem takich badań związane są jednak różnorodne problemy:

1. Po pierwsze musimy stworzyć przestrzeń i czas, w których populacja agentów będzie żyć i działać. W przypadku ewolucyjnych systemów wielo-agentowych mamy zazwyczaj do czynienia z dyskretnym czasem i przestrzenią.
2. Następnie musimy stworzyć byty (agentów) oraz mechanizmy ich niskopoziomowych interakcji. Modele agentów oraz ich podstawowych interakcji są możliwe do sformułowania i są one zakodowane *explicite*. Przedmiotem naszego zainteresowania są oczywiście relacje wyższego poziomu, których modele nie są możliwe do sformułowania. Relacje te wyłonią się dopiero podczas trwania symulacji.
3. Aby mieć możliwość zbadania interesujących nas zjawisk, musimy dysponować odpowiednimi narzędziami do ich obserwacji.
4. Na koniec, musimy sporządzić czasowy schemat aktywności stworzonych przez nas obiektów.

Kluczową sprawą w procesie tworzenia naszego systemu jest problem czasu. Oczywiście jest, że chcemy zawsze utrzymać system w stanie synchronizacji. Przykładowo, chcemy aby uaktualnienie stanu pewnego rodzaju obiektów dokonało się przed uaktualnieniem stanu innego rodzaju obiektów ponieważ zależą one w pewien sposób od siebie. Problemy te wiążą się oczywiście z bardziej ogólnymi problemami dotyczącymi współbieżności. Wszystkie te sprawy są bardzo subtelne i czasami nawet ludzie przeprowadzający symulacje komputerowe nie są ich świadomi i nie wiedzą w jaki sposób wpływają one na uzyskiwane przez nich rezultaty.

Równie ważną sprawą jest kluczowy w nauce proces niezależnej weryfikacji uzyskanych wyników. Aktualnie jest to proces trudny do zrealizowania w badaniach naukowych prowadzonych z wykorzystaniem symulacji komputerowych. Dzieje się tak, ponieważ w procesie przeprowadzania symulacji wykorzystywane są różnorodne narzędzia symulacyjne, systemy operacyjne i sprzęt komputerowy, co utrudnia fizykalną interpretację uzyskanych wyników.

Poniżej przedstawione zostaną dwa narzędzia mające zastosowanie w dwóch fazach przygotowywania symulacji komputerowej: formułowania modeli matematycznych podsystemów oraz tworzenia struktury, w ramach której można osadzić reprezentacje części składowych systemu i umożliwić ich wzajemne interakcje.

2.1 Model formalny systemu wielo-agentowego

Jak zostało to już powiedziane wcześniej, w literaturze występuje kilka koncepcji formalnego opisu agenta i systemu wielo-agentowego, które możemy wykorzystać w procesie formułowania matematycznego modelu podsystemów. Modele te kładą nacisk na różne aspekty budowy oraz działania agenta i systemu wielo-agentowego. W celu sformułowania modeli niższego poziomu naszego systemu będziemy w dalszej części pracy używali jednego z takich modeli: modelu opartego na koncepcji architektury *M-Agenta* [1, 2, 4]. Model ten umożliwia analizę i opis systemów wielo-agentowych na wysokim poziomie abstrakcji, zapewniając jednocześnie łatwość przejścia do fazy implementacji bez zaburzenia struktury logicznej elementów systemu. Dla wygody czytelnika, przedstawione zostaną teraz podstawowe koncepcje tego modelu. Szczegóły tego podejścia można znaleźć w pracach [1, 2, 4].

2.1.1 Podstawowe elementy systemu

Definicja 2.1. System wielo-agentowy AW , możemy opisać następująco:

$$AW = (\mathcal{A}, \mathcal{E}) \quad (1)$$

gdzie \mathcal{A} jest zbiorem możliwych (dopuszczalnych) konfiguracji agentów oraz \mathcal{E} jest zbiorem możliwych (dopuszczalnych) postaci przestrzeni środowiska.

Definicja 2.2. Całe otoczenie agenta, łącznie z nim samym i innymi agentami oraz wszelkie relacje pomiędzy elementami systemu, stanowią środowisko $V \in \mathcal{V}$. Jego chwilowy stan definiują trzy elementy:

$$V = (E, A, C) \quad (2)$$

gdzie:

\mathcal{V} — zbiór wszystkich dopuszczalnych konfiguracji środowiska;

E — przestrzeń środowiska, w której działają agenci, $E \in \mathcal{E}$;

A — zbiór agentów ($a \in A$) działających w przestrzeni E , $A \in \mathcal{A}$;

\mathcal{C} — zbiór wszystkich dopuszczalnych relacji;

C — relacje pomiędzy agentami $a \in A$ a przestrzenią środowiska E i pomiędzy samymi agentami, $C \in \mathcal{C}$.

2.1.2 Budowa środowiska

Definicja 2.3. Przestrzeń środowiska z określoną strukturą i zasobami można określić w następujący sposób:

$$E = (R, T), \quad E \in \mathcal{E}, \quad \mathcal{E} = \mathcal{R} \times \mathcal{T} \quad (3)$$

gdzie:

\mathcal{R} — zbiór wszystkich możliwych konfiguracji zasobów;

R — zbiór (konfiguracja) zasobów środowiska, $R \in \mathcal{R}$;

r — określony zasób, $r \in R$;

\mathcal{T} — zbiór wszystkich możliwych (dopuszczalnych) topologii;

T — topologia przestrzeni środowiska, która definiuje zbiór miejsc t ($t \in T$), w których agenci mogą żyć i działać, $T \in \mathcal{T}$.

2.1.3 Architektura agenta

Definicja 2.4. Agent zdefiniowany jest w następujący sposób:

$$a = \underbrace{(M, Q, S, I, X, L, m, q, s)}_{\text{profil intelektualny}}, \underbrace{(P, Q_p, S_p, X_p, q_p, s_p)}_{\text{profil energetyczny}} \quad (4)$$

gdzie:

\mathcal{M} — zbiór konfiguracji modeli agenta;

M — zbiór modeli środowiska reprezentujący wiedzę agenta o środowisku (konfiguracja modeli środowiska), $M \in \mathcal{M}$;

m — aktualny model środowiska ($m \in M$);

P — aktualny stan energetyczny;

Q, Q_p — uporządkowany zbiór celów agenta (konfiguracja celów agenta);

q, q_p — cel agenta aktualnie wybrany do realizacji ($q \in Q, q_p \in Q_p$), $q : M \times M \rightarrow \mathbb{R}$, $q(m, m') \in \mathbb{R}$, m' jest modelem nowego (przewidywanego), zmodyfikowanego środowiska;

\mathcal{S} — zbiór konfiguracji strategii;

S, S_p — konfiguracja strategii ($S, S_p \in \mathcal{S}$);

s, s_p — aktualnie wybrana do realizacji strategia ($s \in S, s_p \in S_p$), $s: M \rightarrow M$,
 $m' = s(m)$;

I — operator obserwacji, który przy użyciu zbioru M tworzy model środowiska
 $m, I: \mathcal{M} \times \mathcal{V} \rightarrow M, m = I(M, V)$;

X, X_p — operator realizacji strategii $s(s_p)$. Jego zastosowanie wywołuje zmiany
w środowisku, $X: S \times \mathcal{V} \rightarrow \mathcal{V}, V' = X(s, V), s \in S, V, V' \in \mathcal{V}, V'$ jest
nowym środowiskiem agenta a , stworzonym poprzez realizację przez niego
strategii s w środowisku $V = (E, A, C)$;

L — operator auto-adaptacji (uczenia się). Dostosowuje on agenta a do aktualnej
charakterystyki środowiska poprzez modyfikację zbiorów M oraz $S, L =$
 $\{L_M, L_S\}, L_M: \mathcal{M} \times M \times M \rightarrow \mathcal{M} (M' = L_M(M, m', m'')), L_S: \mathcal{S} \times$
 $M \times M \rightarrow \mathcal{S} (S' = L_S(S, m', m''))$.

2.2 System symulacyjny SWARM

System symulacyjny SWARM został opracowany w Instytucie Santa Fe [15, 16].
Może on być pomocny badaczom zajmującym się symulacjami komputerowymi,
szczególnie w obszarze systemów wielo-agentowych i sztucznego życia.

Głównym zamierzeniem twórców tego systemu było stworzenie takiego narzę-
dzia symulacyjnego, przy użyciu którego można by otrzymywać poprawne i
powtarzalne wyniki symulacji komputerowych. Aby osiągnąć ten cel, system
SWARM posiada następujące cechy:

1. Pisanie symulacji przeniesione zostaje na wyższy poziom abstrakcji. Wszy-
stkie aplikacje implementowane są w odniesieniu do standardowego zestawu
narzędzi symulacyjnych.
2. System SWARM zwalnia autora symulacji z konieczności posiadania całej
wiedzy informatycznej, która jest zazwyczaj niezbędna do zaimplementowa-
nia współbieżnego i rozproszonego systemu. Dodatkowo, SWARM wymusza
wyrażanie explicite przyjmowanych założeń dotyczących współbieżności.

System SWARM umożliwia autorowi symulacji zaimplementowanie modelu
symulowanego systemu tj. właściwości fizycznych i struktury przestrzeni, agentów
itd. Następnie definiowany jest czasowy plan aktywności tych obiektów.

Kolejną ważną cechą systemu jest to, iż dostarcza on różnorodnych narzędzi
niezbędnych do obserwacji stworzonego przez nas sztucznego świata. Możliwe
jest zarówno obserwowanie różnych parametrów symulacji na bieżąco, w czasie
jej trwania, jak i zapisywanie danych do pliku w celu późniejszej ich obróbki i
analizy.

System SWARM zaimplementowany został w obiektowo zorientowanym języku programowania Objective-C [10], jako biblioteki klas. Biblioteki te dostarczają użytkownikowi narzędzi niezbędnych w procesie tworzenia symulacji. Szczegóły ich opis można znaleźć w pracy [16].

3 Przykładowe zastosowanie: ewolucyjny wielo-agentowy system predykcyjny

W sekcji niniejszej przedstawione zostanie zastosowanie technologii ewolucyjnych systemów wielo-agentowych do konstrukcji systemu predykcyjnego [2, 3].

W systemie tym głównym celem populacji agentów jest przewidywanie zmian środowiska, w którym żyją i działają. W środowisku zdefiniowany jest parametr $\alpha \in \{0, 1\}$. Zmiany wartości parametru α w dyskretnych chwilach czasu mogą być przedstawione jako ciąg binarny $x(t)$, gdzie $x(t)$ jest wartością α w chwili czasu t . Każdy z agentów stara się przewidzieć wartość, którą parametr α przyjmie w chwili $t + 1$. Tak więc, agent a_i generuje ciąg binarny y_i , taki że $y_i(t) = \tilde{x}(t + 1)$, gdzie $\tilde{x}(t + 1)$ to przewidywana wartość parametru α w chwili $t + 1$.

3.1 Model systemu

3.1.1 Budowa środowiska

Zasobami dostępnymi dla wszystkich agentów są wartości parametru $\alpha = x(\tau)$, gdzie $\tau = 0, \dots, t$ oraz energia środowiska P_e . Tak więc, w naszym przypadku $R = \{\alpha, P_e\}$.

Topologia T środowiska określona jest jako graf, w którym każdy wierzchołek połączony jest ze swoimi ośmioma sąsiadami.

3.1.2 Wewnętrzna budowa agenta

W rozważanym systemie, każdy agent składa się z jednej lub więcej komórek. Każda z komórek składa się z automatu skończonego, informacji o swoim wieku oraz statystyk predykcji. Każda z komórek określa prognozę wartości parametru α w chwili $t + 1$ przy użyciu automatu skończonego, którego język wejścia-wyjścia składa się z symboli 0 i 1. Koncepcja przewidywania zmian w otoczeniu przez ewoluującą populację osobników, których chromosomy reprezentowane są przez automaty skończone, występuje w literaturze pod nazwą „programowania ewolucyjnego” [6, 7, 11]. Idea zastosowania automatu skończonego jako chromosomu agenta została zaprezentowana w pracy [3]. Konkretna realizacja tej koncepcji w systemie prezentowanym w niniejszej pracy, przedstawiona została schematycznie na rysunku 1.

Agenta można przedstawić jako parę $a_i = \langle \text{wyjście, akcja} \rangle$. W chwili t na wyjściu pojawia się symbol $y_i(t-1) = \tilde{x}(t)$. Gdy predykcja agenta a_i jest poprawna, uzyskuje on od środowiska nagrodę w postaci porcji energii. Akcja natomiast, to wszelkie możliwe działania agenta, takie jak reprodukcja czy ruch w pewnym kierunku. Decyzje dotyczące akcji podejmowanych przez agenta, uzależnione są od aktualnego poziomu jego energii. Genom danej komórki składa się z czterech chromosomów, które kodują jej automat skończony:

1. przejścia między stanami, gdy symbolem na wejściu jest 0;
2. przejścia między stanami, gdy symbolem na wejściu jest 1;
3. symbole wyjściowe, gdy symbolem na wejściu jest 0;
4. symbole wyjściowe, gdy symbolem na wejściu jest 1.

Liczba genów w każdym z chromosomów odpowiada maksymalnej dopuszczalnej liczbie stanów automatu skończonego (n_{max}^s).

Dwa parametry pokazują jakość predykcji danej komórki agenta:

$\Psi_{ik}^{S_j(0)}$ — prawdopodobieństwo poprawnej predykcji k -tej komórki agenta a_i , gdy jej automat skończony znajduje się w stanie S_j , a symbolem na wejściu jest 0;

$\Psi_{ik}^{S_j(1)}$ — prawdopodobieństwo poprawnej predykcji k -tej komórki agenta a_i , gdy jej automat skończony znajduje się w stanie S_j , a symbolem na wejściu jest 1;

Kolejne dwa parametry pokazują jakość predykcji danego agenta:

$\Psi_i(t)$ — prawdopodobieństwo poprawnej predykcji agenta a_i .

$$\Psi_i(t) = \frac{\sum_{k=t-n_m}^t p_i(k)}{n_m} \quad (5)$$

gdzie n_m jest długością pamięci agenta oraz

$$p_i(k) = \begin{cases} 1 & \text{gdy agent } a_i \text{ wykonał poprawną predykcję w chwili } k \\ 0 & \text{w przeciwnym przypadku.} \end{cases} \quad (6)$$

$\Psi_{max}^i(t)$ — maksymalna wartość prawdopodobieństwa poprawnej predykcji w czasie całego życia agenta a_i .

$$\Psi_{max}^i(t) = \max \{ \Psi_i(k) : k = t_0^i, \dots, t, t_0^i \text{ jest czasem narodzin agenta } a_i \}. \quad (7)$$

Profil intelektualny. Model środowiska agenta a_i w chwili t

$$m_i(t) = \langle (Y_i(t), \Omega_i(t)), (Y_i(t+1), \Omega_i(t+1)), \dots \rangle, \quad (8)$$

gdzie

$$Y_i(u) = \{y_{i1}(u), \dots, y_{in}(u) : n \text{ jest liczbą komórek}\}, \quad (9)$$

$$\Omega_i(u) = \left\{ \Psi_{i1}^{S_1^u(x_1(u))}, \dots, \Psi_{in}^{S_n^u(x_n(u))} : n \text{ jest liczbą komórek} \right\}, \quad (10)$$

dla $u = t, t+1, \dots$

$$y_{ik}(u) = t_{ik}^{S_k^u}(x_k(u)) \quad (11)$$

jest symbolem pojawiającym się na wyjściu k -tego automatu skończonego przy przejściu przez symbol $x_k(u)$, S_k^u jest stanem k -tego automatu skończonego w chwili u ,

$$x_k(u) = \begin{cases} x(u) & \text{dla } u = t \\ y_{ik}(u-1) & \text{dla } u = t+1, t+2, \dots, \end{cases} \quad (12)$$

dla $k = 1, \dots, n$, n jest liczbą komórek agenta a_i .

Po zastosowaniu operatora realizacji strategii X , model środowiska agenta a_i

$$m'_i(t) = \langle (y_i(t), \omega_i(t)), (y_i(t+1), \omega_i(t+1)), \dots \rangle, \quad (13)$$

gdzie $y_i(u) = y_{ik}(u)$ dla $u = t, t+1, \dots$, przy czym k jest takie, że

$$\omega_i(u) = \Psi_{ik}^{S_k^u(z(u))} = \max \left\{ \Psi_{i1}^{S_1^u(z(u))}, \dots, \Psi_{in}^{S_n^u(z(u))} : n \text{ jest liczbą komórek} \right\} \quad (14)$$

dla $u = t, t+1, \dots$,

$$z(u) = \begin{cases} x(u) & \text{dla } u = t \\ y_i(u-1) & \text{dla } u = t+1, t+2, \dots \end{cases} \quad (15)$$

Celem profilu intelektualnego jest wykonywanie poprawnych predykcji:

$$q(m_i(t), m'_i(t)) = \begin{cases} 1 & \text{gd } y_i(t) = x(t+1) \\ 0 & \text{gd } y_i(t) \neq x(t+1). \end{cases} \quad (16)$$

Następnie, wykonywane jest przejście do kolejnej chwili czasu i operator obserwacji I generuje model

$$m_i''(t) = \langle (Y_i(t), \Omega_i(t)), (Y_i(t+1), \Omega_i(t+1)), \dots \rangle, \quad (17)$$

poprzez pobranie od środowiska aktualnej wartości parametru $x(t)$.

Operator auto-adaptacji $L = \{L_m\}$ składa się ze wszystkich operatorów ewolucyjnych.

Profil energetyczny. Profil energetyczny agenta a_i reprezentowany jest przez parametr $P_i(t) \in \mathbb{R}$, $P_i(t) \in [0, P_{max}]$, P_{max} jest maksymalnym dopuszczalnym poziomem energii agenta. Głównym celem agenta w tym profilu jest maksymalizacja wartości parametru $P_i(t)$. Aby osiągnąć ten cel, agent realizuje strategię energetyczną s_e poprzez stosowanie operatorów *reprodukcji*, *agregacji* oraz *ucieczki*. Każda akcja podejmowana przez agenta (przemieszczenie, reprodukcja) powoduje utratę pewnej ilości energii. Energia może być pozyskana od środowiska wyłącznie poprzez prawidłową predykcję. Gdy $P_i(t) = 0$ agent ginie.

Każda decyzja agenta dotycząca podejmowanej akcji (reprodukcji, krzyżowania, kierunku ruchu) zależy od aktualnego poziomu energii. Energia agenta służy również jako mechanizm eliminacji z systemu agentów o niskim poziomie przystosowania do warunków środowiska (tj. wykonujących nieprawidłowe predykcje).

Zmiany poziomu energii agenta a_i podczas jego życia, można przedstawić następująco:

$$P_i(t) = P_i(t_0^i) + \sum_{k=t_0^i+1}^t (\delta_i^p(k) + \delta_i^a(k)), \quad (18)$$

gdzie t_0^i jest czasem narodzin agenta a_i , $P_i(t_0^i)$ jest jego energią początkową,

$$P_i(t_0^i) = P_i^r(t_0^i) + P_i^c(t_0^i), \quad (19)$$

$P_i^r(t_0^i) \in [P_{min}^{rc}, P_{max}^{rc}]$ jest energią otrzymywaną od rodzica w procesie reprodukcji, natomiast $P_i^c(t_0^i) \in [P_{min}^{rc}, P_{max}^{rc}]$ jest energią otrzymywaną od drugiego rodzica (jeżeli dochodzi do procesu krzyżowania), $P_{min}^{rc}, P_{max}^{rc}$ są parametrami globalnymi,

$$\delta_i^p(k) = \begin{cases} \delta > 0 & \text{gd } y_i(k-1) = x(k) \\ \delta < 0 & \text{w przeciwnym przypadku,} \end{cases} \quad (20)$$

$$\delta_i^a(k) = \begin{cases} \delta > 0 & \text{jeżeli doszło do wchłonięcia innego agenta} \\ 0 & \text{w przeciwnym przypadku.} \end{cases} \quad (21)$$

Całkowita energia systemu w chwili t

$$P(t) = P_e(t) + \sum_{i=1}^{m(t)} P_i(t), \quad (22)$$

gdzie $P_e(t)$ jest energią środowiska, $P_i(t)$ jest energią i -tego agenta, $m(t)$ jest liczbą agentów znajdujących się w systemie w chwili t . Parametr ten zapobiega nadmiernemu wzrostowi liczby agentów działających w systemie.

3.1.3 Operatory ewolucyjne

W symulowanym procesie ewolucji, zachodzącym w systemie predykcyjnym, wykorzystywane są operatory *reprodukcji*, *krzyżowania*, *mutacji*, *agregacji* oraz *ucieczki*. Omówione zostanie teraz ich działanie.

Operatory *reprodukcji*, *krzyżowania* i *mutacji*. Na rysunku 2 pokazany został proces *reprodukcji* agenta, zachodzący w rozważanym przez nas systemie.

Do reprodukcji agenta a_i w chwili t dochodzi z prawdopodobieństwem p_r (jest to parametr globalny), gdy spełniony jest następujący warunek:

$$P_i^r(t) \leq \frac{P_i(t)}{P_{max}} \quad (23)$$

gdzie $P_i^r(t) \in [P_{min}^r, 1.0]$ jest pseudolosową liczbą rzeczywistą, P_{min}^r jest natomiast parametrem określającym minimalny poziom energii agenta niezbędny do przeprowadzenia procesu reprodukcji. Jeżeli agent a_i w chwili t składa się z więcej niż jednej komórki, wtedy do reprodukcji wybierana jest komórka k , taka że zachodzi

$$\Psi_{ik}^{S_k^t(x(t))} = \max \left\{ \Psi_{i1}^{S_1^t(x(t))}, \dots, \Psi_{in}^{S_n^t(x(t))} : n \text{ jest liczbą komórek agenta } a_i \right\} \quad (24)$$

Operator *mutacji* może zmienić symbol wyjściowy oraz przejście pomiędzy danymi stanami automatu skończonego. Do *mutacji* dochodzi z prawdopodobieństwem p_m , które jest parametrem globalnym symulacji.

Do operacji *krzyżowania* homologicznych chromosomów agentów a_i oraz a_j dochodzi z prawdopodobieństwem p_c (parametr globalny), gdy agent a_i (inicjujący proces reprodukcji) spełnia warunki (23) oraz

$$\exists a_j, \text{ taki że } d(a_i, a_j) \leq d_{max}^c \wedge P_j^c(t) \leq \frac{P_j(t)}{P_{max}} \quad (25)$$

gdzie $d(a_i, a_j)$ jest długością najkrótszej ścieżki pomiędzy tymi węzłami grafu, w których przebywają agenci a_i oraz a_j , d_{max}^c jest maksymalną długością najkrótszej ścieżki, $P_j^c(t) \in [P_{min}^c, 1.0]$ jest pseudolosową liczbą rzeczywistą, P_{min}^c jest natomiast parametrem określającym minimalny poziom energii agenta niezbędny do przeprowadzenia operacji krzyżowania.

Operator *agregacji*. Operator ten umożliwia powstawanie relacji społecznych w populacji agentów. Koncepcja operatora *agregacji* po raz pierwszy została zaprezentowana w pracy [3]. Poniżej przedstawiona zostanie konkretna realizacja tej koncepcji w opisywanym systemie predykcyjnym.

W skład agenta wchodzi jedna lub więcej komórek:

$$a_i = \{c_1, \dots, c_n : 1 \leq n \leq n_{max}^c\}, \quad (26)$$

gdzie c_i jest komórką oraz n_{max}^c jest maksymalną liczbą komórek mogących wchodzić w skład agregatu.

Do sformowania agregatu przez agentów a_i oraz a_j w chwili t dochodzi z prawdopodobieństwem p_a (parametr globalny) jeżeli

$$d(a_i, a_j) \leq d_{max}^a \wedge n_i^c + n_j^c \leq n_{max}^c \wedge \Psi_i(t) \geq \Psi_j(t) \wedge \Psi_{max}^i(t) \leq \Psi_{max}^j(t) \quad (27)$$

gdzie d_{max}^a jest maksymalną długością najkrótszej ścieżki, n_i^c i n_j^c są liczbami komórek, odpowiednio, agenta a_i oraz a_j .

Sformowany agregat składa się z komórek obydwu agentów, natomiast poziom energii agregatu (po wchłonięciu przez agenta a_i agenta a_j) jest następujący:

$$P_i(t+1) = \begin{cases} P_i(t) + P_j(t) & \text{jeżeli } P_i(t) + P_j(t) < P_{max} \\ P_{max} & \text{w przeciwnym przypadku.} \end{cases} \quad (28)$$

Operator *ucieczki*. Operator ten umożliwia migrację agentów pomiędzy środowiskami, w których panują różne warunki, tzn. w których agenci muszą przewidywać wartości parametru α pobierane z różnych ciągów pseudolosowych. Ponieważ istnienie wielu środowisk powoduje izolację geograficzną poszczególnych grup agentów, możliwe staje się wystąpienie procesu specjacji w populacji.

Jeżeli agent a_i znajduje się w chwili t w zasięgu agenta a_j^α (tj. gdy $d_{ij}(t) \leq r_j$; r_j jest zasięgiem oddziaływania agenta a_j^α), podającego kolejne wartości parametru α w środowisku j (wartości te tworzą ciąg $x_j(t)$), to ich wzajemna odległość zmienia się w następujący sposób:

$$d_{ij}(t) = \begin{cases} d_{ij} < d_{ij}(t-1) & \text{jeżeli } y_i(t-1) = x_j(t) \\ d_{ij} > d_{ij}(t-1) & \text{jeżeli } y_i(t-1) \neq x_j(t). \end{cases} \quad (29)$$

Jeżeli agent a_i wyjdzie poza zasięg agenta a_j^α (tj. gdy $d_{ij}(t) > r_j$), wtedy porusza się on w sposób losowy dotąd, aż poziom jego energii P_i nie spadnie poniżej pewnego poziomu P_{min}^m . W takiej sytuacji agent zaczyna przemieszczać się w kierunku najbliższego agenta a_k^α (przy czym $k \neq j$).

Wprowadzenie operatora *ucieczki* stanowi istotną zmianę jakościową w procesie ewolucji, w stosunku do rozwiązania opisanego w [3].

3.2 Wyniki badań symulacyjnych

W sekcji niniejszej przedstawione zostaną wyniki badań symulacyjnych przeprowadzonych z wykorzystaniem opisanego powyżej systemu predykcyjnego. Wpływ prawdopodobieństwa reprodukcji, krzyżowania i mutacji nie był w tych badaniach brany pod uwagę. Miały one na celu raczej przebadanie wpływu nowych operatorów ewolucyjnych (*agregacji* i *ucieczki*) na przebieg procesu ewolucji w systemie wielo-agentowym.

3.2.1 Przebieg procesu ewolucji z wykorzystaniem operatorów klasycznych

Badania symulacyjne, których wyniki widoczne są na rysunku 3, miały na celu sprawdzenie, jak będzie przebiegał proces ewolucji z wykorzystaniem wyłącznie klasycznych operatorów (*reprodukcji*, *krzyżowania* oraz *mutacji*). Populacja agentów musiała przewidywać zmiany wartości parametru α , który przyjmował wartości kolejnych elementów pseudolosowego ciągu $x(t)$ o długości l . Ciąg ten był powtarzany cyklicznie dając w wyniku ciąg nieskończony. W przypadku opisywanej tutaj symulacji, $l = 10$ oraz maksymalna liczba stanów automatu skończonego $n_{max}^s = 10$.

Jak można zaobserwować na rysunku 3, w pierwszym kroku symulacji skuteczność predykcji całej populacji agentów wynosi 50% (jest to wartość początkowa nadawana wszystkim agentom). Następnie występuje spadek liczby agentów w systemie, gdyż eliminowani są ci z nich, którzy wykonują dużo błędnych predykcji. W późniejszym czasie można zaobserwować wzrost liczby agentów z coraz wyższym procentem poprawnych predykcji. Dzieje się tak, ponieważ kolejne pokolenia składają się z potomków tych agentów, którzy przewidywali najlepiej kolejne wartości parametru α . Natomiast agenci przewidujący błędnie zmiany środowiska wymarli bezpotomnie. Pod koniec symulacji można zaobserwować pojawienie się w populacji grupy agentów wykonujących 95-100% poprawnych predykcji.

3.2.2 Badania symulacyjne z wykorzystaniem operatora *agregacji*

Badania wpływu operatora *agregacji* na przebieg procesu ewolucji przeprowadzone zostały z wykorzystaniem tak samo skonstruowanego ciągu $x(t)$, z tym że $l = 20$ oraz $n_{max}^s = 5$. Liczba elementów ciągu jest tutaj większa niż liczba stanów automatu skończonego. Powoduje to, że agenci z jedną komórką nie są w stanie przewidywać kolejnych wartości parametru α ze skutecznością lepszą niż 75-80% (rysunek 4).

Natomiast w sytuacji, gdy możliwe jest formowanie agregatów, w populacji pojawiają się agenci przewidujący z dużo lepszą skutecznością. Przy odpowiedniej liczbie komórek możliwe jest nawet uzyskanie w populacji osobników przewidujących poprawnie w 95-100%. Jak można zaobserwować na wykresie,

zwiększanie parametru n_{max}^c powoduje występowanie w populacji coraz większej liczby agentów ze skutecznością predykcji rzędu 95-100%. Należy jednak zaznaczyć, że nadmierne zwiększanie wartości tego parametru prowadzi do znacznego wydłużenia czasu trwania symulacji (dochodzenia do poprawnej predykcji).

3.2.3 Symulacje z operatorem *ucieczki*

Symulacje, których wyniki przedstawione są w niniejszej sekcji, miały na celu zbadanie wpływu operatora *ucieczki* na przebieg procesu ewolucji w systemie predykcyjnym.

Maksymalna liczba stanów automatu skończonego $n_{max}^s = 5$. W trakcie symulacji istniały trzy przestrzenie środowiska: E_1, E_2 oraz E_3 . W przestrzeni E_2 populacja agentów musiała przewidywać zmiany parametru α , który przyjmował wartości kolejnych elementów powtarzanego cyklicznie, pseudolosowego ciągu $x_2(t)$, o długości $l_2 = 10$. W przestrzeni E_3 parametr α przyjmował wartości kolejnych elementów pseudolosowego ciągu $x_3(t)$ o długości $l_3 = 15$ również powtarzanego cyklicznie. Natomiast ciąg $x_1(t)$ w przestrzeni E_1 powstał w wyniku naprzemiennego powtarzania $k_2(t)$ razy ciągu $x_2(t)$ oraz $k_3(t)$ razy ciągu $x_3(t)$, gdzie liczby powtórzeń poszczególnych ciągów są liczbami pseudolosowymi, losowanymi za każdym razem gdy dochodzi do zmiany ciągu, $k_2(t) \in [1, k_{max}]$, $k_3(t) \in [1, k_{max}]$. W przypadku opisywanej tutaj symulacji $k_{max} = 3$.

W przypadku działania operatora *ucieczki*, w trakcie pierwszej części symulacji, agenci przebywali w środowiskach E_2 i E_3 . Dzięki temu możliwe było powstanie dwóch odrębnych gatunków, specjalizujących się w rozpoznawaniu dwóch różnych ciągów. Następnie obydwie gatunki zostały zmuszone do przeniesienia się do środowiska E_1 , poprzez ustawienie wartości parametrów $r_2 = 0$ i $r_3 = 0$. Po wejściu w zasięg nowego środowiska, skuteczność predykcji agentów osiągała wartość 50%, co można zaobserwować na rysunkach 5 i 6.

W przypadku symulacji, w których operator *ucieczki* był wyłączony agenci przez cały czas trwania symulacji przebywali w środowisku E_1 .

W symulacjach, w których włączony był operator *agregacji*, w środowisku E_1 agenci mogli formować agregaty złożone z maksymalnie $n_{max}^c = 2$ komórek.

Operatory klasyczne włączone były we wszystkich symulacjach. Poszczególne symulacje różnią się pomiędzy sobą tylko występowaniem (lub nie występowaniem) poszczególnych nowych operatorów ewolucyjnych.

Na rysunkach 5 i 6 widać, odpowiednio, wykresy najlepszego i średniego procentu poprawnych predykcji w populacji agentów dla poszczególnych rodzajów operatorów ewolucyjnych. Jak widać, najlepsze wyniki można uzyskać poprzez zastosowanie zarówno operatora *ucieczki* jak i *agregacji*. Najwyższy udział procentowy poprawnych predykcji w populacji przekracza wtedy 90%, a średni dochodzi do 80%. Zdecydowanie najgorsze wyniki występują, gdy włączony jest operator *ucieczki* bez możliwości tworzenia agregatów. Spowodowane to jest tym, iż wąsko wyspecjalizowane gatunki agentów przenoszone są do środowiska z naj-

trudniejszym do przewidywania ciągiem, bez możliwości wzajemnej współpracy. Rezultat jest wtedy znacznie gorszy niż gdy agenci od początku przebywają w środowisku, w którym panują najtrudniejsze warunki.

Na rysunku 7 widać rozkład skuteczności predykcji w populacji w ostatnim kroku symulacji, dla różnych operatorów ewolucyjnych. Tutaj również zdecydowanie najlepiej działa wariant z *ucieczką* i *agregacją*. Możliwe było w tym przypadku uzyskanie agentów ze skutecznością predykcji 90-95%. Na drugim miejscu znajduje się wariant z samą *agregacją*. W populacji występuje wtedy znacznie więcej agentów z wyższym udziałem procentowym poprawnych predykcji niż w przypadku zastosowania samych operatorów klasycznych. Najgorsze wyniki uzyskano w przypadku zastosowania samego operatora *ucieczki*. Przyczyny tego zostały już omówione powyżej.

4 Podsumowanie

W pracy niniejszej przedstawione zostały wybrane zagadnienia związane z realizacją ewolucyjnych systemów wielo-agentowych. Systemy te, których działanie opiera się na wprowadzeniu procesu ewolucji w systemach wielo-agentowych, mogą być traktowane zarówno jako nowe podejście do konstrukcji programów ewolucyjnych, jak i nowy kierunek rozwoju systemów wielo-agentowych. Proces ewolucji realizowany w systemach wielo-agentowych oferuje nam dwie podstawowe korzyści:

1. Uniknięcie ograniczeń klasycznych programów ewolucyjnych.
2. Umożliwienie populacji agentów adaptacji do zmian zachodzących w środowisku.

Dla badań prowadzonych w dziedzinie ewolucyjnych systemów wielo-agentowych (oraz ogólnie dla wszystkich badań prowadzonych z wykorzystaniem symulacji komputerowych) charakterystyczne jest odniesienie do zjawisk, dla których nie jest możliwe sformułowanie ścisłego modelu matematycznego. Program takich badań można sprowadzić do dwu zasadniczych etapów:

1. Sformułowanie modeli matematycznych części składowych systemu oraz ich wzajemnych relacji na niższym poziomie.
2. Stworzenie środowiska, w którym podsystemy oraz ich wzajemne relacje mogą być symulowane, tak aby możliwym stało się zaobserwowanie i zbadanie zjawisk wyższego poziomu będących przedmiotem naszego zainteresowania.

W pracy niniejszej, dla pierwszej części badań, zaproponowano model systemu wielo-agentowego oparty na koncepcji *M-Agenta*. Jako narzędzie dla realizacji

części drugiej zaproponowany został system SWARM, który umożliwia otrzymanie poprawnych i sprawdzalnych wyników symulacji. Metodologia ta została zastosowana do konstrukcji ewolucyjnego wielo-agentowego systemu predykcyjnego.

Punktem wyjścia do przeprowadzonych badań była praca [3], w której zaprezentowano technologię ewolucyjnych systemów wielo-agentowych, jej zastosowanie do konstrukcji systemu predykcyjnego oraz wprowadzono operator *agregacji*, umożliwiający powstanie relacji społecznych w populacji agentów.

W niniejszej pracy zaprezentowany został nowy operator *ucieczki*. Umożliwia on migrację agentów pomiędzy środowiskami, w których panują odmienne warunki. Istnienie wielu środowisk powoduje izolację geograficzną poszczególnych grup agentów, dzięki czemu możliwe jest wystąpienie procesu specjacji w populacji agentów.

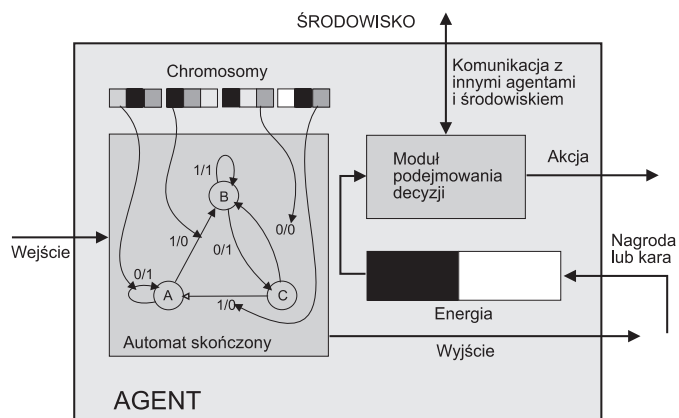
Na podstawie wyników przeprowadzonych symulacji pokazane zostało, że zastosowanie operatora *ucieczki* w połączeniu z operatorem *agregacji* przyczynia się do znacznego poprawienia rezultatów predykcji wykonywanej przez populację agentów.

Przyszłe badania będą miały na celu wprowadzenie nowych mechanizmów wzorowanych na ewolucji biologicznej oraz zastąpienie automatu skończonego siecią neuronową. Umożliwi to zastosowanie ewolucyjnych systemów wielo-agentowych do rozwiązywania bardziej złożonych problemów.

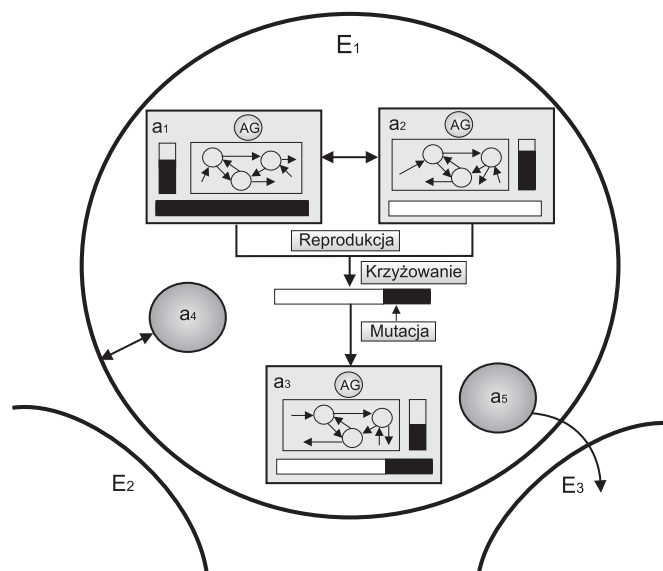
Bibliografia

- [1] Cetnarowicz E., Nawarecki E., Cetnarowicz K.: Agent oriented technology of decentralized systems based on the m-agent architecture. W *Proceedings of the Management and Control of Production and Logistics Conference – MCPL '97*, Sao Paulo, Brazil, 1997. IFAC, PERGAMON.
- [2] Cetnarowicz K.: M-agent architecture based method of development of multi-agent systems. W *Proceedings of the 8-th Joint EPS-APS International Conference on Physics Computing*, Kraków, Poland, 1996. ACC Cyfronet.
- [3] Cetnarowicz K., Kisiel-Dorohinicki M., Nawarecki E.: The application of evolution process in multi-agent world to the prediction system. W *Proceedings of the Second International Conference on Multi-Agent Systems – ICMAS'96*, Osaka, Japan, 1996. AAI Press.
- [4] Cetnarowicz K., Nawarecki E., Zabinska M.: M-agent architecture and its application to the agent oriented technology. W *Proceedings of the International Workshop – Distributed Artificial Intelligence and Multi-Agent Systems – W-DAIMAS'97*, St. Petersburg, Russia, 1997.
- [5] Dawkins R.: *Ślepy zegarmistrz*. Warszawa, PIW, 1997.

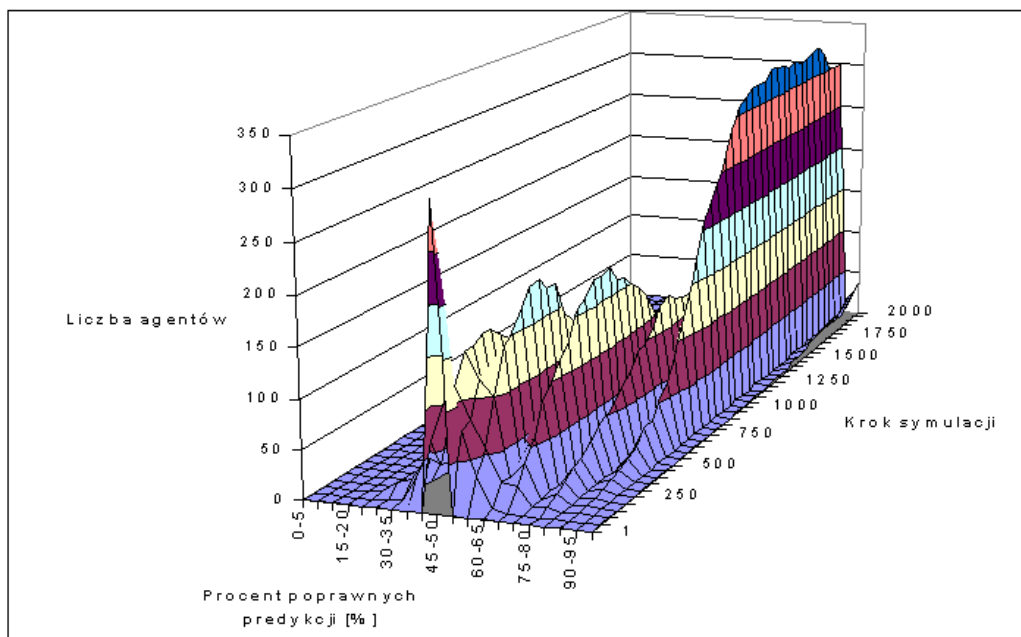
- [6] Fogel L. J., Owens A. J., Walsh M. J.: *Artificial Intelligence Through Simulated Evolution*. Chichester, UK, John Wiley, 1966.
- [7] Goldberg D.: *Algorytmy genetyczne i ich zastosowania*. Warszawa, WNT, 1995.
- [8] Gould S. J.: *Niewczesny pogrzeb Darwina*. Warszawa, PIW, 1991.
- [9] Krzanowska H., Łomnicki A., Rafiński J., Szarski H., Szymura J.: *Zarys mechanizmów ewolucji*. Warszawa, PWN, 1997.
- [10] Larkin D., Wilson G.: *Object-Oriented Programming and the Objective-C Language*. NeXT Software, Inc., 1995.
- [11] Michalewicz Z.: *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. Warszawa, WNT, 1999.
- [12] Rasmussen S., Barrett C. L.: Elements of a theory of simulation. W *ECAL'95, Lecture Notes in Computer Science*. Springer-Verlag, 1995.
- [13] Reichholf J.: *Twórczy impuls*. Warszawa, PWN, 1996.
- [14] Smith J. M.: *Problemy biologii*. Warszawa, PWN, 1992.
- [15] Swarm Development Group: *Brief Overview of Swarm*. Santa Fe Institute, Santa Fe, USA, 1999. <http://www.santafe.edu/projects/swarm>.
- [16] Swarm Development Group: *Reference Guide for Swarm 1.4.1*. Santa Fe Institute, Santa Fe, USA, 1999. <http://www.santafe.edu/projects/swarm>.



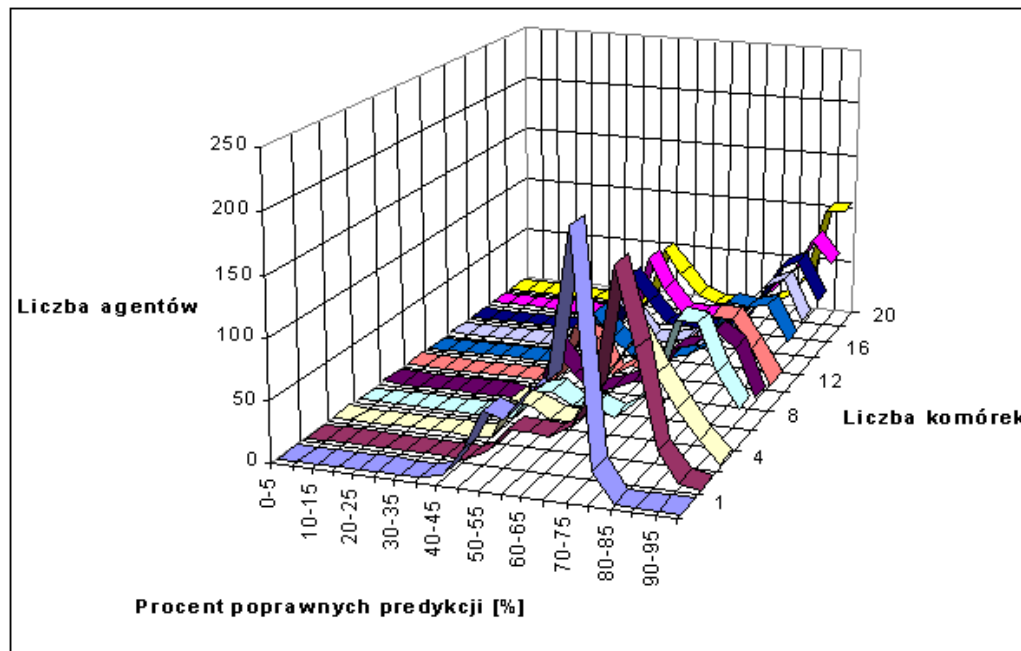
Rysunek 1: Wewnętrzna budowa agenta (z jedną komórką).



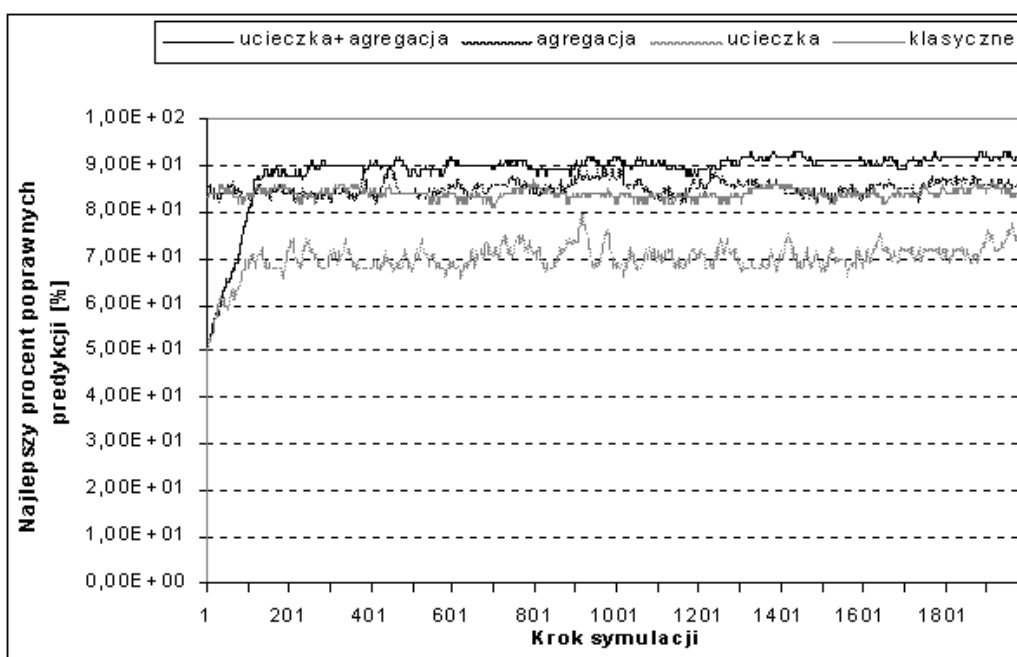
Rysunek 2: Proces reprodukcji agenta.



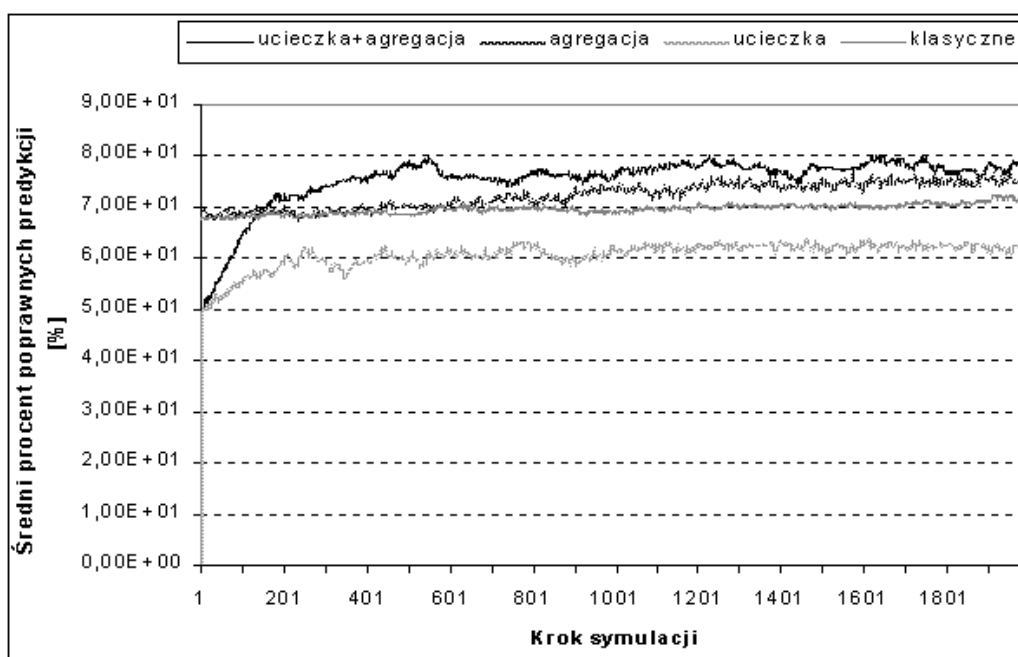
Rysunek 3: Rozkład skuteczności predykcji w populacji podczas trwania symulacji (wyłącznie operatory *klasyczne*).



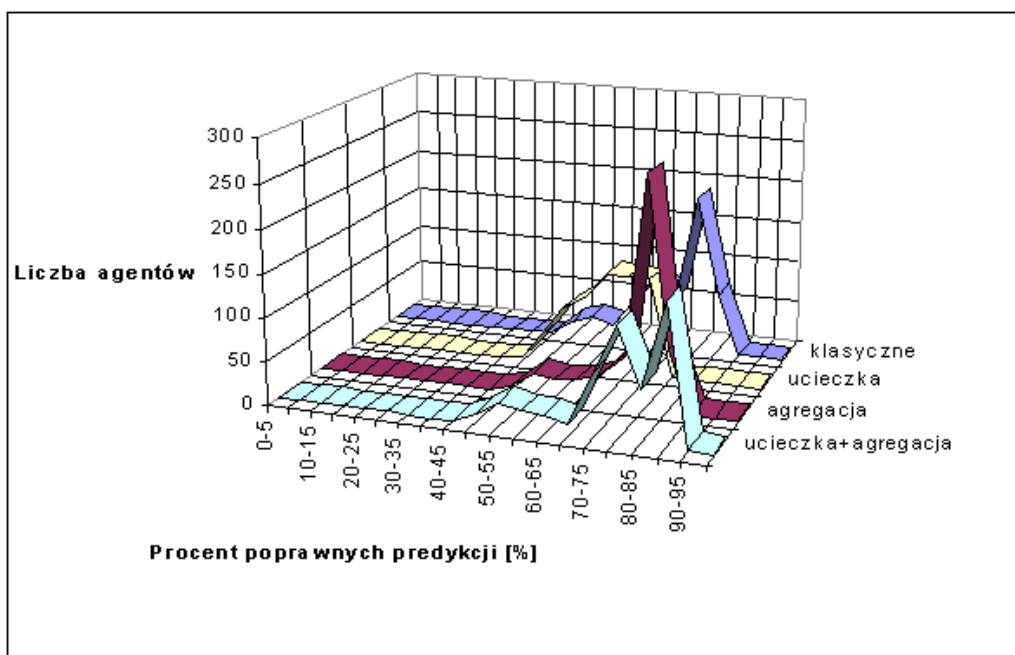
Rysunek 4: Rozkład skuteczności predykcji w populacji w ostatnim kroku symulacji (włączony operator *agregacji*).



Rysunek 5: Najlepszy procent poprawnych predykcji w populacji dla różnych operatorów ewolucyjnych.



Rysunek 6: Średni procent poprawnych predykcji w populacji dla różnych operatorów ewolucyjnych.



Rysunek 7: Rozkład skuteczności predykcji w populacji w ostatnim kroku symulacji dla różnych operatorów ewolucyjnych.