

Agentowe metody inteligencji obliczeniowej

Krzysztof Cetnarowicz, Rafał Dreżewski,

Aleksander Byrski, Leszek Siwik

Katedra Informatyki

Akademia Górniczo-Hutnicza w Krakowie

{cetnar,drezew,olekb,siwik}@agh.edu.pl

1. Wprowadzenie

Terminem *inteligencja obliczeniowa* (ang. *computational intelligence – CI*) określa się grupę heurystycznych technik obliczeniowych wykorzystywanych w przypadku zadań, dla których nie jest możliwe znalezienie rozwiązań metodami tradycyjnymi [33]. Do technik inteligencji obliczeniowej zalicza się obliczenia ewolucyjne, sieci neuronowe oraz logikę rozmytą, jak również systemy hybrydowe łączące w sobie wymienione techniki.

Koncepcja *ewolucyjnych systemów wieloagentowych* wywodzi się z badań mających na celu zdecentralizowanie obliczeń ewolucyjnych [12]. Decentralizacja, osiągnięta w wyniku realizacji procesów ewolucyjnych w systemie wieloagentowym, doprowadziła do powstania nowej klasy systemów obliczeniowych posiadających szereg cech i możliwości nieobecnych w „klasycznych” algorytmach ewolucyjnych. Jedną z ważniejszych cech tego typu systemów jest łatwość tworzenia systemów hybrydowych, łączących różnorodne techniki inteligencji obliczeniowej

w ramach jednego spójnego (agentowego) modelu, co w rezultacie prowadzi do *agentowych metod inteligencji obliczeniowej*.

Rozdział rozpoczyna się od wprowadzenia do zagadnień związanych z realizacją procesów ewolucyjnych w systemach wieloagentowych. Następnie omówiono podstawowe modele ewolucyjnych systemów wieloagentowych. W ostatniej części rozdziału przedstawione zostały wybrane zastosowania tej klasy systemów.

2. Procesy ewolucyjne w systemach wieloagentowych

Algorytmy ewolucyjne są przedmiotem badań od kilkudziesięciu lat i były z powodzeniem wykorzystywane jako punkt wyjścia do opracowania wielu praktycznych zastosowań [1, 20, 28]. Jednakże pomimo intensywnych badań i wprowadzanych licznych ulepszeń otrzymywane wyniki nie były w pełni zadowalające, a co więcej, napotymano często na poważne trudności w rozwiązywaniu pojawiających się problemów (np. definicja funkcji przystosowania, utrata różnorodności populacji).

Algorytmy genetyczne i programowanie ewolucyjne, które można znaleźć w literaturze [1, 20, 24, 25] posiadają następujące cechy, które mogą ograniczać ich przydatności w praktycznych zastosowaniach:

- Proces ewolucji jest procesem scentralizowanym, w którym jeden wspólny algorytm (proces) służy do realizacji procesu selekcji i generacji nowych pokoleń.
- Podmioty ewolucji są uproszczone do genów (układów genów) i pozbawione jakichkolwiek możliwości podejmowania decyzji

mających wpływ na proces ewolucji.

- Lista stosowanych operatorów ewolucyjnych (takich jak mutacja, krzyżowanie itp.) nie wydaje się być wystarczająca i można przypuszczać, że brak na niej pewnych operacji mających kluczowe znaczenie dla przebiegu ewolucji.

W konsekwencji powstają niedoskonałości i ograniczenia algorytmów genetycznych takie jak to, że uczestniczące w procesie obiekty są pozbawione możliwości pobierania informacji z otaczającej ich rzeczywistości, pozbawione zdolności podejmowania decyzji i realizacji własnych przedsięwzięć w otaczającym ich środowisku. Obiekty takie są niezdolne do rywalizacji oraz współzawodnictwa i nie mogą uczestniczyć w procesach i strukturach społecznych.

Ponadto ograniczenie listy operatorów genetycznych (do takich jak mutacja i krzyżowanie) zmniejsza znacznie możliwości procesu ewolucji do kreacji bardziej złożonych algorytmów.

Przy tych ograniczeniach znane algorytmy genetyczne i programowanie ewolucyjne przypominają bardziej proces dopasowania parametrów danego algorytmu do zadanych warunków niż twórczy proces tworzenia nowych rozwiązań. Wspomniane niedoskonałości są w pewnym zakresie dostrzeżone, o czym świadczą próby udoskonalenia algorytmów genetycznych i programowania ewolucyjnego [24, 25]. Jednakże w większości przypadków, brak jest całościowego spojrzenia na problem, a częściowe ulepszenia prowadzą do tego, że procesy algorytmów genetycznych stają się logicznie niespójne, a cały proces ewolucji trudny do

zinterpretowania.

Dlatego też wydaje się, że zastosowanie koncepcji świata autonomicznych agentów może pomóc w rozwiązaniu niektórych problemów występujących w dziedzinie algorytmów genetycznych i programowania ewolucyjnego. Proces ewolucji realizowany w systemie wieloagentowym czyli w „świecie agentów”, może być rozważany jako nowe podejście do obliczeń ewolucyjnych i dostarczyć nowych możliwości, implikowanych przez następujące spostrzeżenia:

- W procesie ewolucji uczestniczą agenci. Nie są oni biernymi obiektami przemian, ale uczestnikami procesu, w którym dobrze zdefiniowano relacje agent-agent i agent-środowisko [12].
- Proces ewolucji jest procesem zdecentralizowanym, dla każdego agenta przebiega z właściwą mu lokalnie szybkością. Agenci działają niezależnie i społeczne relacje mogą być tworzone w populacji agentów.
- Agenci mogą obserwować otaczające środowisko, podejmować niezależne decyzje i je realizować, co powoduje, że dzięki współzawodnictwu i rywalizacji można zrealizować proces selekcji i doboru w sposób zdecentralizowany.
- Podstawowe właściwości procesu ewolucji realizowane w formie zdecentralizowanej umożliwiają wprowadzenie nowych operacji ewolucyjnego tworzenia algorytmów.

2.1. Przesłanki procesu ewolucyjnego działania systemu wieloagentowego

Według definicji systemu wieloagentowego określającego „świat agentów” proces ewolucji zachodzi w społeczności agentów działających w danym środowisku. Stan tego środowiska zmienia się w rezultacie zdarzeń wewnętrznych (np. działania agentów) i zewnętrznych (np. interwencji zewnętrznej użytkownika). Zmiany zachodzące w środowisku (będące wynikiem zdarzeń) mogą być obserwowane przez agentów i być wykorzystane do podjęcia właściwego działania. Każda akcja wykonana przez dowolnego agenta zmienia relacje pomiędzy danym agentem a środowiskiem (w tym i w stosunku do innych agentów). Każdy agent w czasie swojego istnienia realizuje charakterystyczne dla swojego rodzaju cele, między innymi cel wspólny dla wszystkich agentów — cel przeżycia. Realizację tego celu z ogólnego punktu widzenia, agent może realizować dwiema drogami:

- przez przystosowanie siebie do warunków środowiska,
- przez modyfikację środowiska zgodnie ze swoimi potrzebami (zwykle współpracując z innymi agentami).

Zgodnie z powyższym, w celu rozwiązania danego problemu, stosując proces ewolucji agentów, należy:

- odwzorować dany problem w postaci środowiska z odpowiednimi cechami charakterystycznymi i zdarzeniami mogącymi zachodzić

w tym środowisku i zmieniającymi to środowisko,

- wprowadzić społeczność agentów mogących działać i ewoluować w tym środowisku,
- jako rozwiązanie problemu przyjęć odpowiednią interpretację zmian w środowisku dokonanych przez społeczność agentów działających i ewoluujących w tym środowisku.

Podstawową przyczyną zmuszającą agentów do udziału w procesie ewolucji jest relacja agent–środowisko. Jeżeli stan tej relacji jest niezadowalający dla danego agenta, ma on do wyboru dwie drogi działania:

1. zmienić siebie — przystosowując się do środowiska przez operacje mutacji i krzyżowania,
2. zmienić środowisko przez operacje reprodukcji, agregacji i ucieczki (migracji).

Proces ewolucji autonomicznych agentów może być zrealizowany za pomocą następujących operatorów oddziaływających bądź to na pojedynczego agenta, bądź na określoną grupę agentów:

- mutacja realizowana na pojedynczym agencie,
- krzyżowanie w ramach określonej grupy agentów,
- reprodukcja realizowana na pojedynczym agencie,
- agregacja realizowana na wybranej grupie agentów,
- migracja agenta ze środowiska do innego środowiska.

Operatory mutacji, krzyżowania i reprodukcji w środowisku agentów mają podobną postać do przedstawionych w literaturze [1, 20, 28]

z następującą różnicą: każdy agent samodzielnie podejmuje decyzję o swoich przedsięwzięciach takich jak rodzaj operatora ewolucyjnego i wybór partnera (partnerów) niezbędnych do tych operacji. W wyniku tego, proces ewolucji każdego agenta ma własny charakter indywidualny dla niego i zależny od jego decyzji, odbywający się według własnego niezależnego cyklu.

Idea *operacji agregacji* może być rozważana jako kreacja nowego środowiska agentów. Grupa agentów nie przystosowana do warunków aktualnego środowiska dokonuje porozumienia celem przejęcia kontroli nad częścią istniejącego środowiska. Kontrolując wybraną część środowiska, agenci zmieniają jego parametry tak, aby było lepiej dostosowane do ich wymagań. Taka grupa agentów, dzięki specjalizacji i współpracy utrzymuje założone (pożądane) parametry w utworzonym przez siebie środowisku i zachowuje się jak nowy agent o nowych charakterystycznych cechach, który powstał dzięki operacji agregacji. Operator agregacji umożliwia zmianę relacji agenci–środowisko przez zmianę parametrów środowiska.

Drugim operatorem, który umożliwia zmianę relacji agent–środowisko przez zmianę parametrów środowiska, jest *operator ucieczki (migracji)*. Załóżmy, że proces ewolucji ma miejsce w kilku środowiskach, między którymi mogą przemieszczać się agenci. Procesy ewolucji w poszczególnych środowiskach różnią się między sobą w pewnym zakresie swoich parametrów. Jeżeli w jednym ze wspomnianych środowisk powstanie w wyniku operacji mutacji, krzyżowania lub agregacji agent, który nie jest zbyt dobrze przystosowany do działania w tym środowisku, może przemieścić się do innego, w którym przy innych parametrach charakterystycznych może działać lepiej niż w poprzednim i stać się

początkiem nowego, wartościowego pokolenia.

2.2. Podstawowe modele ewolucyjnych systemów wieloagentowych

Kluczowym elementem opracowanego modelu agentowych obliczeń ewolucyjnych jest wprowadzenie mechanizmów ewolucyjnych do środowiska agentowego, czyli poddanie procesom ewolucyjnym populacji agentów. W efekcie otrzymujemy system posiadający zdolność samoczynnej adaptacji do warunków środowiska zarówno na poziomie poszczególnych agentów, jak i ich struktury organizacyjnej [30]. Proces ewolucyjny jest zatem prowadzony (zarządzany) przez populację autonomicznych agentów, reprezentujących jednostki ewolucji (osobniki) lub ich grupy (stada). Prezentowana koncepcja posiada wszystkie zalety *równoległego algorytmu ewolucyjnego* (ang. *parallel evolutionary algorithm*), jednocześnie pozwalając na dynamiczną adaptację systemu obliczeniowego do rozwiązywanego problemu oraz stosunkowo proste wprowadzanie nowych mechanizmów ewolucyjnych, których realizacja była trudna (lub niemożliwa) w „klasycznych” algorytmach ewolucyjnych.

Ze względu na zdecentralizowaną naturę ewolucyjnych systemów wieloagentowych i autonomię osobników (agentów) w tego rodzaju systemach nie jest możliwe wykorzystanie scentralizowanych mechanizmów sterowania przebiegiem procesów ewolucyjnych. Nie jest również możliwe zastosowanie żadnego z mechanizmów selekcji wykorzystywanych w klasycznych algorytmach ewolucyjnych.

W wyniku prowadzonych badań zaproponowana została koncepcja

mechanizmu selekcji oraz sterowania procesami ewolucji w systemie wieloagentowym w oparciu o wyczerpywalne zasoby [12]. Mechanizm wymiany zasobów pomiędzy agentami a środowiskiem wymaga zdefiniowania w systemie pewnej liczby zasobów, które umożliwiają ocenę agentów z punktu widzenia ich przydatności w kontekście rozwiązywanego problemu. W ewolucyjnych systemach wieloagentowych wystarczającym okazało się wykorzystanie jednego zasobu, który jest niezbędny ewoluującym osobnikom do wykonywania wszelkich czynności życiowych takich jak np. reprodukcja czy migracja w środowisku. W najprostszym przypadku środowisko może przyznawać agentom zasoby w ten sposób, że osobniki o wyższej wartości funkcji przystosowania dostają jej więcej. Mechanizm ten służy przede wszystkim do realizacji procesu selekcji w ewolucyjnych systemach wieloagentowych.

Podstawowym wariantem rozważanego modelu jest **ewolucyjny system wieloagentowy** (ang. *evolutionary multi-agent system* — *EMAS*), w którym agent pełni rolę podstawowej jednostki ewolucji [22]. Wymaga to, po pierwsze, wyposażenia go w umiejętność reprodukcji, której powinny towarzyszyć losowe zmiany dziedziczonych przez wygenerowanego agenta cech (mutacje, rekombinacje). Po drugie, koniecznym staje się wprowadzenie mechanizmów eliminacji agentów najgorzej spełniających pewne kryteria związane z realizowanymi zadaniami. Wynikiem tego jest asynchroniczny i zdecentralizowany proces, który, przy właściwym zdefiniowaniu mechanizmów selekcji, prowadzić powinien do samoczynnego osiągnięcia przez system pożądanego w aktualnej sytuacji stanu (konfiguracji).

Alternatywnym w stosunku do powyższego jest wariant **stadnego**

systemu wieloagentowego (ang. *flock-based multi-agent system — FMAS*) [23], w którym agent reprezentuje grupę osobników (stado). Podobnie jak w *migracyjnym (wyspowym)* modelu równoległego algorytmu ewolucyjnego [2], w ramach każdego stada realizowany jest proces ewolucji (np. klasyczny algorytm ewolucyjny), rozszerzony o możliwości migracji osobników. Jednakże w odróżnieniu od klasycznego modelu migracyjnego, osobniki migrują pomiędzy stadami w granicach jednej wyspy, natomiast stanowiące dodatkowy poziom organizacji systemu stada, mogą w całości migrować pomiędzy wyspami. Łączenie, rozdzielanie i usuwanie stad w zależności od ich stanu (a raczej stanu stanowiącej stado grupy osobników) daje możliwość dynamicznej rekonfiguracji systemu w trakcie procesu obliczeniowego.

Badania nad mechanizmami niszowania i specjacji doprowadziły do opracowania modelu **koewolucyjnego systemu wieloagentowego** (ang. *co-evolutionary multi-agent system — CoEMAS*) [16]. W systemach takich możliwe jest występowanie wielu oddziaływających na siebie gatunków agentów. Oddziaływania pomiędzy gatunkami polegają na wywieraniu wpływu przez agentów jednego gatunku na poziom zasobów u agentów drugiego gatunku, a przez to, pośrednio, na wartość ich funkcji przystosowania.

3. Zastosowania agentowych metod inteligencji obliczeniowej

3.1. Optimalizacja wielomodalna

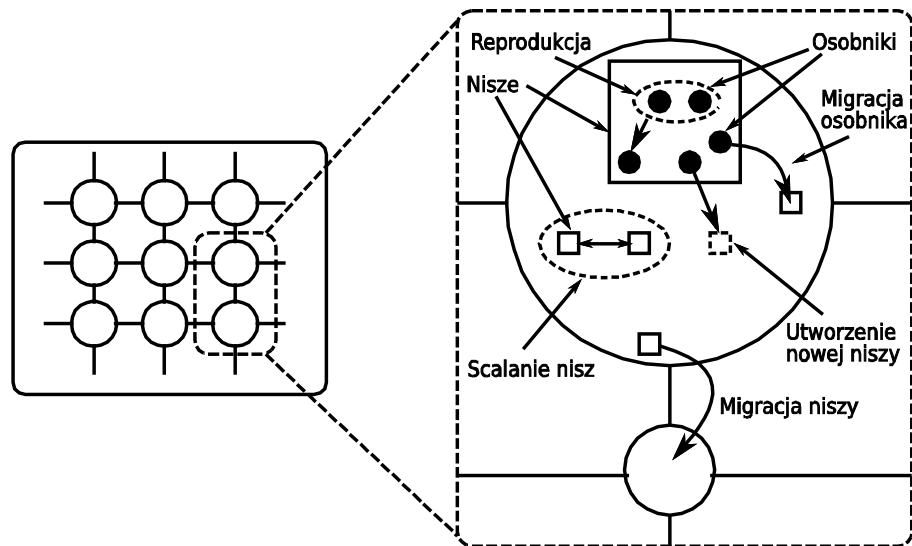
Zarówno analizy teoretyczne, jak i wyniki eksperymentów wskazują na

to, że w przypadku problemów optymalizacji wielomodalnej algorytm ewolucyjny znajdzie najczęściej tylko jedno rozwiązanie [26]. Jest to związane z szerszym zjawiskiem utraty różnorodności populacji przez algorytm ewolucyjny. Znalezienie wszystkich (lub większości) rozwiązań możliwe jest wyłącznie w przypadku zastosowania specjalnych technik optymalizacji wielomodalnej, zwanych *technikami niszowania i specjacji* (ang. *niching and speciation techniques*) [26].

Poniżej przedstawiono trzy systemy przeznaczone do optymalizacji funkcji wielomodalnych, opracowane z wykorzystaniem przedstawionego w rozdz. 2 modelu koewolucyjnego systemu wieloagentowego. We wszystkich systemach zastosowano reprezentację zmiennopozycyjną oraz krzyżowanie uśredniające i mutację z autoadaptacją [2].

3.1.1. Koewolucyjny system wieloagentowy z mechanizmem koewolucji gatunków

W koewolucyjnym systemie wieloagentowym z mechanizmem koewolucji gatunków (*NCoEMAS*) występuje gatunek nisz oraz pewna liczba gatunków osobników (p. rys. 1) [17]. Zadaniem gatunku nisz jest takie oddziaływanie z populacją osobników, aby nastąpił podział tej populacji na izolowane rozrodczo podpopulacje (gatunki) zlokalizowane w obszarach przyciągania różnych minimów lokalnych optymalizowanej funkcji.



Rysunek 1: System *NCoEMAS*

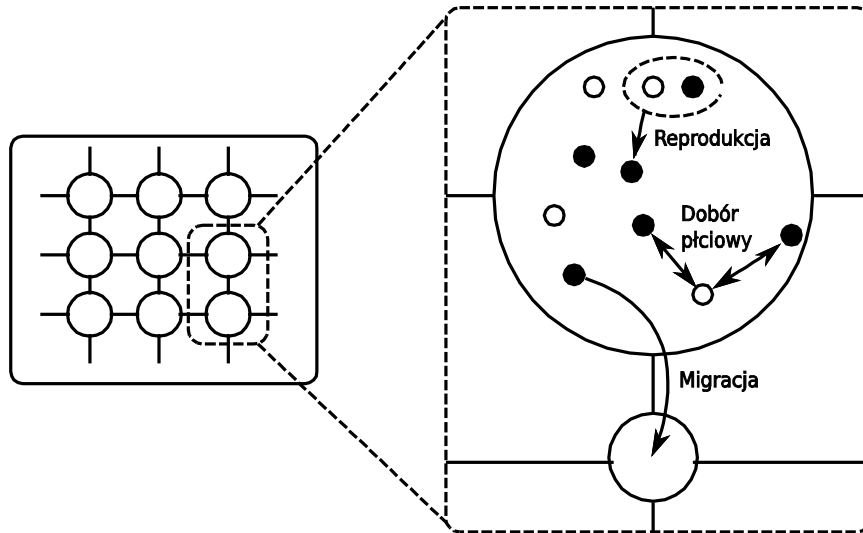
Główne mechanizmy stosowane w tym celu w systemie *NCoEMAS* to:

- Przyporządkowanie agentów z gatunków osobników do agentów z gatunku nisz (agenci z gatunku osobników „żyją wewnątrz” agentów z gatunku nisz — por. rys. 1).
- Możliwość migracji w środowisku wyłącznie agentów z gatunku nisz (agenci z gatunku osobników migrują w środowisku razem z tym agentem z gatunku nisz, do którego są aktualnie przyporządkowani).
- Możliwość migracji agentów z gatunków osobników pomiędzy agentami z gatunku nisz (por. rys. 1).
- Konkurencja pomiędzy agentami z gatunku nisz o ograniczone

zasoby (podział zasobów pomiędzy tych agentów przez środowisko).

- Konkurencja pomiędzy agentami z danego gatunku osobników o ograniczone zasoby (podział zasobów pomiędzy tych agentów przez agenta z gatunku nisz, do którego są oni aktualnie przyporządkowani).
- Scalanie gatunków osobników (czyli agentów z gatunku nisz — każdy z gatunków osobników jest przyporządkowany do jednego agenta z gatunku nisz) zlokalizowanych w obszarze przyciągania tego samego minimum lokalnego (por. rys. 1).
- Podział gatunku osobników (czyli agenta z gatunku nisz), którego agenci znajdują się w obszarach przyciągania różnych minimów lokalnych (por. rys. 1).
- Izolacja rozrodcza agentów różnych gatunków.
- Specjalny operator mutacji dla agentów z gatunku nisz, działający w oparciu o wyznaczenie środka ciężkości danego gatunku osobników.

3.1.2. Koewolucyjny system wieloagentowy z mechanizmem koewolucji płci



Rysunek 2: System *SCoEMAS*

Koewolucyjny system wieloagentowy z mechanizmem koewolucji płci i wynikającego stąd doboru płciowego (*SCoEMAS*) jest systemem, w którym występuje pewna liczba gatunków osobników [19]. W ramach każdego gatunku występują dwie płcie: samice i samce (p. rys. 2). Proces powstawania gatunków jest tu oparty o zjawisko konfliktu płci oraz wynikającą z niego koewolucję samic i samców. Samice preferują pewne cechy samców (dobór płciowy) w wyniku czego dochodzi do specjacji i powstania wielu gatunków osobników zlokalizowanych w obszarach

przyciągania różnych minimów lokalnych. Główne mechanizmy stosowane w systemie to:

- Konkurencja o ograniczone zasoby pomiędzy agentami tego samego i różnych gatunków (podział zasobu pomiędzy agentów przez środowisko).
- Zjawisko konfliktu i koewolucji płci (wyższe koszty reprodukcji w przypadku samic).
- Wynikający z powyższych zjawisk dobór międzypłciowy (wybieranie samców przez samice, preferowanie partnerów zlokalizowanych w obszarze przyciągania tego samego minimum lokalnego).
- Łączenie samców i samic w pary.

3.1.3. Koewolucyjny system wieloagentowy z mechanizmem specjacji allopatricznej

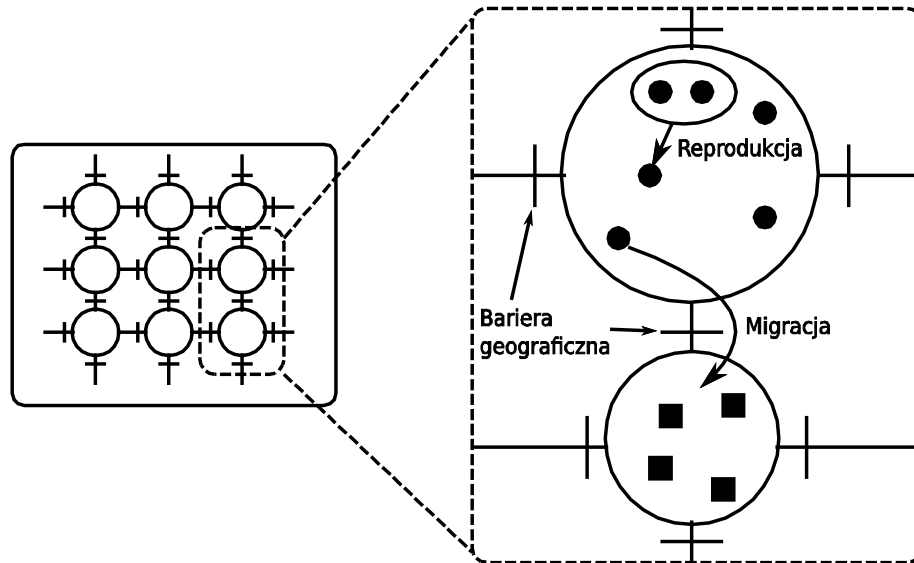


Figure 3: System *ACoEMAS*

W koewolucyjnym systemie wieloagentowym z mechanizmem specjacji allopatricznej (*ACoEMAS*) występuje pewna liczba gatunków osobników, z których każdy zlokalizowany jest w innym wierzchołku grafu stanowiącego strukturę środowiska (p. rys. 3) [18]. Proces powstawania gatunków jest tu oparty o geograficzną izolację podpopulacji (model specjacji allopatricznej). Główne mechanizmy stosowane w systemie to:

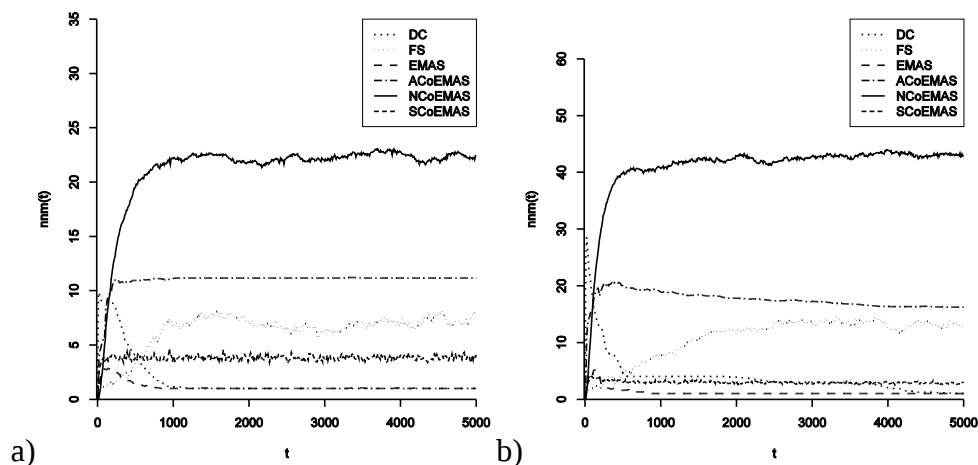
- Konkurencja o ograniczone zasoby pomiędzy agentami tego samego gatunku (podział zasobu pomiędzy agentów przez środowisko).

- Izolacja geograficzna podpopulacji (wysoki koszt migracji w środowisku) i wynikający stąd proces powstawania gatunków.

3.1.4. Eksperymentalne porównanie wybranych technik

W rozdziale niniejszym przedstawione zostaną wyniki eksperymentów, mających na celu porównanie przebiegu procesów powstawania gatunków oraz ich stabilnego utrzymywania w trakcie ewolucji w porównywanych systemach przeznaczonych do optymalizacji funkcji wielomodalnych. W eksperymentach wykorzystano ewolucyjny system wieloagentowy (*EMAS*), system *NCoEMAS*, system *ScoEMAS* oraz system *ACoEMAS*. W celu porównania wyników z algorytmami znanymi z literatury, wykorzystano również system z mechanizmem ścisku deterministycznego (*DC*, [26]) oraz system z mechanizmem podziału przystosowania (*FS*, [15]).

W eksperymentach użyte zostały dwie wielomodalne funkcje testowe: Rastrigin'a oraz Schwefel'a [32]. Do porównania systemów wykorzystany został wskaźnik $n_{nm}(t)$, którego wartość w chwili t odpowiada liczbie zlokalizowanych otoczeń minimów lokalnych funkcji celu — otoczenie danego minimum lokalnego jest uznawane za zlokalizowane w chwili t , gdy znajduje się w nim co najmniej n_{min} osobników. Promień otoczenia przyjęto jako równy 0.05 dla funkcji Rastrigin'a oraz 10.0 dla funkcji Schwefel'a.



Rysunek 4: Liczba zlokalizowanych otoczeń minimów lokalnych funkcji Rastrigin'a (a) oraz Schwefel'a (b) dla $nmin=5$. Porównanie dla różnych systemów

Najlepsze rezultaty osiągnięto w przypadku systemu *NCoEMAS*, który lokalizował średnio otoczenia około 22 minimów lokalnych w przypadku funkcji Rastrigin'a oraz około 40 minimów lokalnych funkcji Schwefel'a (p. rys. 4). System *ACoEMAS* lokalizował otoczenia około 10 minimów lokalnych funkcji Rastrigin'a i 15 minimów lokalnych funkcji Schwefel'a. System *FS* osiągnął rezultaty gorsze od systemu *NCoEMAS* oraz *ACoEMAS* (p. rys. 4). Również gorsze wyniki uzyskał system *SCoEMAS*, co związane jest ze zjawiskiem polegającym na tym, że utrzymywała się w nim stosunkowo duża różnorodność populacji w ramach poszczególnych gatunków i osobniki były zlokalizowane często w dużej odległości od minimum lokalnego — były rozrzucone po całych obszarach przyciągania minimów lokalnych. System *EMAS* nie był w stanie zlokalizować więcej niż

jedno otoczenie minimum lokalnego, natomiast *DC* początkowo lokalizował dość dużą liczbę otoczeń minimów lokalnych, ale nie utrzymywał stabilnie gatunków i wszystkie z nich, za wyjątkiem jednego, stopniowo zanikały (p. rys. 4).

3.2. Optymalizacja wielokryterialna

3.2.1. Definicja problemu

Proces wielokryterialnego podejmowania decyzji (jeden z najbardziej naturalnych i najczęstszych procesów decyzyjnych) opiera się najczęściej o procesy i mechanizmy związane z rozwiązywaniem problemów optymalizacji wielokryterialnej. Podstawowe mechanizmy analizy wielokryterialnej zostały co prawda zaproponowane w XIX wieku — jednak ich właściwy rozwój w rozwiązywaniu problemów optymalizacji wielokryterialnej (*multi-objective optimization problems* — *MOOP*) nastąpił po roku 1906 kiedy to Vilfredo Pareto sformułował swoją teorię optymalności (tzw. optymalność w sensie Pareto). Zgodnie z definicjami zaproponowanymi w [14] problem optymalizacji wielokryterialnej w swojej ogólnej postaci formułowany jest następująco:

$$MOOP \equiv \left\{ \begin{array}{l} \text{Minimalizacja / Maksymalizacja : } f_m(\bar{x}), \quad m = 1, 2, \dots, M \\ \text{Przy uwzględnieniu } g_j(\bar{x}) \geq 0 \quad j = 1, 2, \dots, J \\ h_k(\bar{x}) = 0 \quad k = 1, 2, \dots, K \\ x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad i = 1, 2, \dots, N \end{array} \right\}$$

Zbiór ograniczeń — zarówno „równościowych” (equalities) jak i „nierównościowych” (inequalities) a także ograniczenia nakładane na zmienne decyzyjne (lower bounds, upper bounds) definiują tak zwaną

przestrzeń przeszukiwań (feasible alternatives) (D).

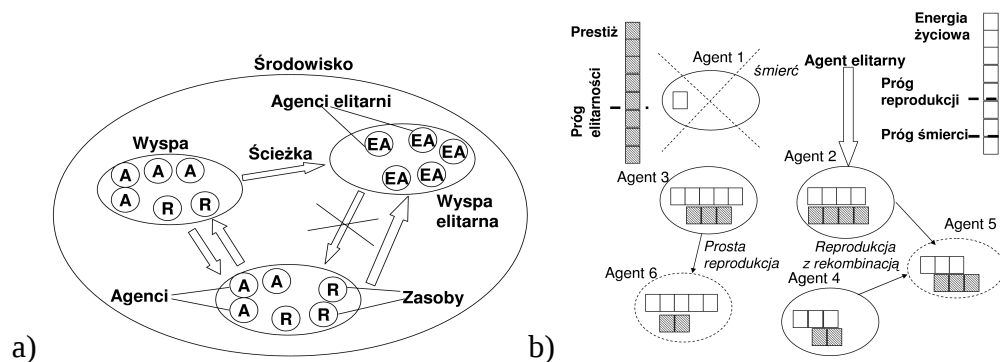
Rozwiązanie w sensie Pareto problemu optymalizacji wielokryterialnej oznacza poszukiwanie wszystkich niezdominowanych alternatyw ze zbioru D .

3.2.2. Systemy agentowo-ewolucyjne w optymalizacji wielokryterialnej

Punktem wyjścia do rozważań na temat zastosowania systemów agentowo-ewolucyjnych do rozwiązywania problemów optymalizacji wielokryterialnej stanowi ewolucyjny system wieloagentowy (patrz rozdz. 2),

w którym agent pełni rolę jednostki ewolucji i w którym wprowadzono mechanizm przekazywania zasobów przez agentów zdominowanych agentom dominującym w trakcie ich „spotkań”. Dotychczasowe — przeprowadzone przez autorów — eksperymenty związane z wykorzystaniem paradygmatu obliczeń agentowo-ewolucyjnych w zadaniach optymalizacji wielokryterialnej potwierdzają, iż prowadzić to może to uzyskania niezwykle obiecujących rezultatów zarówno w zadaniach optymalizacji ciągłej jak i dyskretnej (w tym kombinatorycznej). Niestety zastosowanie tego paradygmatu w rozwiązywaniu problemów optymalizacji wielokryterialnej napotyka na ograniczenia z których za najistotniejsze przyjąć należy niebezpieczeństwo zaistnienia stagnacji procesu ewolucyjnego. Jak pokazują wstępne testy, jednym ze sposobów zapobiegania występowania tych ograniczeń, może być wprowadzenie do systemu tzw. operatorów elitarnych — a więc wyposażenie agentów w akcje, które dostępne są wyłącznie „dla elity” społeczeństwa agentowego ([35, 36].

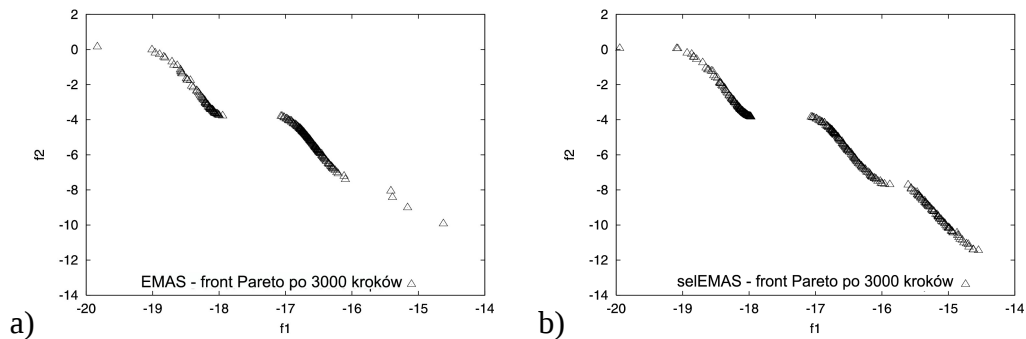
Jedną z takich akcji może być prawo do migracji na wyróżnioną — elitarną — wyspę. W przypadku, w którym aktywność agenta elitarnego na wyspie elitarniej ogranicza się do sprawdzania czy na wyspie tej nie pojawił się osobnik go dominujący, lub też czy on nie dominuje (w sensie relacji dominacji) innych osobników, zgodnie z proponowaną terminologią mamy do czynienia z tzw. semi-elitarnym ewolucyjno-agentowym systemem optymalizacji wielokryterialnej (selEMAS). Zastosowanie w ramach systemu agentowo-ewolucyjnego operatorów semi-elitarnych [36] pozwala — poza innymi zaletami — na wprowadzenie do systemu dalszych dodatkowych mechanizmów odpowiedzialnych za równomierne rozproszenie osobników niezdominowanych wzdłuż całego frontu Pareto (co stanowi jeden z wyznaczników jakości otrzymanego rozwiązania problemu optymalizacji wielokryterialnej, a co jest szczególnie istotne w przypadku trudnych problemów charakteryzujących się nieciągłym frontem Pareto (a także zbiorem Pareto), w przypadku problemów kombinatorycznych czy też w przypadku problemów wielokryterialnych-wielomodalnych).



Rysunek 5: Struktura środowiska oraz progi energetyczne w semi-elitarnym systemie agentowo-ewolucyjnym

Na rysunku 5 przedstawiona została struktura oraz progi zasobów występujące w semi-elitarnym systemie agentowo-ewolucyjnym — selEMAS (zastosowanym do optymalizacji wielokryterialnej).

Jednym ze wspomnianych wcześniej dodatkowych mechanizmów, który z powodzeniem może zostać zaproponowany w ramach systemu selEMAS jest mechanizm, który nazwać można *rozproszonym współczynnikiem zatłoczenia* (ang. distributed frontier crowding), który to mechanizm może zostać zrealizowany następująco. Podczas spotkań każdy z agentów (np. agent A) ma możliwość określenia czy rozwiązanie reprezentowane przez spotykanych agentów (np. agenta A) zlokalizowane jest w jego bezpośrednim otoczeniu (innymi słowy czy odległość pomiędzy rozwiązaniami reprezentowanymi przez agentów A oraz B jest mniejsza niż wartość dodatkowego parametru promienia sąsiedztwa ε). Dodatkowo, wymieniając pomiędzy sobą odpowiednie informacje agenci mają możliwość stopniowego (w miarę kolejnych spotkań) zgromadzenia wiedzy o średnim zagęszczeniu fragmentów przestrzeni kryteriów reprezentowanych przez kolejnych agentów. Dzięki takiej wiedzy agent, który staje się agentem elitarnym może uzależnić decyzję o wykonaniu akcji dostępnych dla agentów elitarnych (w szczególności o migracji na wyspę elitarną) od relacji pomiędzy średnim zagęszczeniem otoczenia kolejnych agentów a ilością agentów zlokalizowanych w bezpośrednim sąsiedztwie agenta podejmującego decyzję. Na rysunku 6 zaprezentowano wpływ zastosowania omawianych mechanizmów na jakość otrzymywanych przybliżeń frontu Pareto.

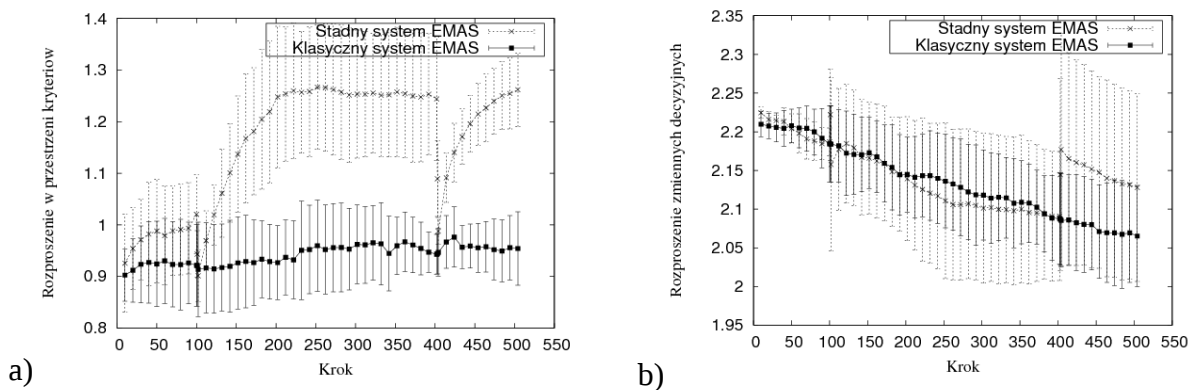


Rysunek 6: Przybliżenia frontu Pareto dla problemu Kursawe otrzymane przez systemy EMAS (a) oraz selEMAS (b) po 3000 kroków symulacji

Alternatywną do opisaną powyżej modyfikacji podstawowego modelu EMAS może być stadny system agentowo-ewolucyjny do optymalizacji wielokryterialnej. W modelu takim agent reprezentuje grupę osobników — stado. W kontekście optymalizacji wielokryterialnej procesy ewolucji zachodzące wewnątrz każdego takiego stada, odpowiedzialne są przede wszystkim za „dryfowanie” rozwiązań w kierunku wzorcowego frontu Pareto, zaś wprowadzone na poziomie agentów-stad operatory stadne (takie jak łączenie, dzielenie, tworzenie i usuwanie stad z systemu) odpowiadać powinny za utrzymanie różnorodności ewoluujących populacji, a także równomierne obsadzenie przez osobniki niezdominowane całego frontu Pareto (a zatem za otrzymanie rozwiązań realizujących trzy podstawowe cele optymalizacji wielokryterialnej — bliskość do wzorcowego frontu Pareto, równomierne rozdystrybuowanie osobników niezdominowanych wzdłuż

całego frontu, a także znalezienie jak największej liczby osobników niezdominowanych). Biorąc pod uwagę tak zdefiniowane cele — operatory stadne mogą być zrealizowane w następujący sposób: podczas spotkań agenta (z osobnikami z tego samego stada) agenci mogą wymienić informacje o swojej lokalizacji (w przestrzeni kryteriów bądź w przestrzeni zmiennych decyzyjnych) — dzięki temu agenci mogą zgromadzić (częściową) wiedzę o odległości (zgodnie z przyjętą metryką) dzielącej ich od pozostałych osobników. Następnie, jeśli taka (częściowa) średnia odległość do innych agentów jest większa niż zadana wartość dodatkowego parametru — która to wartość może zmieniać się (i w zaimplementowanych realizacjach zmienia się) adaptacyjnie — agent może podjąć decyzję o utworzeniu nowego stada (istnieją bowiem przesłanki ku temu, aby stwierdzić, iż agent reprezentuje słabo obsadzony a jednocześnie „obiecujący” fragment przestrzeni. Po podjęciu decyzji o utworzeniu nowego stada — agent wykonuje taką operację, migruje z dotychczasowego do stada nowoutworzonego, a następnie inicjalizuje nowe stado. Operacja inicjalizacji w zaproponowanej realizacji polega na wykonaniu operacji prostej reprodukcji (klonowanie oraz mutacja genotypu o niewielkim zasięgu — np. poprzez zmutowanie najmniej znaczących bitów genotypu). Operacja taka pozwala na próbkowanie z dużym prawdopodobieństwem najbliższego sąsiedztwa agenta tworzącego nowe stado — co, zgodnie z tym co zostało powiedziane wcześniej, może okazać się korzystne z punktu widzenia próbkowania słabo obsadzonego a jednocześnie obiecującego obszaru przestrzeni przeszukiwań. Operacja łączenia stad w podstawowej wersji może być realizowana następująco. Agenci-stad podejmują decyzję o

połączeniu jeśli „środek ciężkości” osobników należących do obu stad oraz różnica w długości promienia obu stad są mniejsze od zadanego parametru (oczywiście wartości obu tych parametrów mogą zmieniać się adaptacyjnie, co skutkuje zwiększającą się precyzją przeszukiwania przestrzeni kryteriów lub zmiennych decyzyjnych — w zależności od tego, w której przestrzeni wykonywane są obliczenia). Aby zaprezentować wpływ omawianych operatorów na zachowanie systemu, a w szczególności na „różnorodność” populacji — podczas eksperymentów zaprezentowanych na rysunku 7 operatory tworzenia nowych stad uruchomione zostały w setnym oraz czterechsetnym kroku symulacji. Jak można zauważyć — wprowadzenie operatorów stadnych wpływa bardzo pozytywnie na różnorodność populacji, a w konsekwencji także na jakość otrzymywanych przybliżeń frontów Pareto.



Rysunek 7: Różnorodność w przestrzeni kryteriów (a) oraz różnorodność w przestrzeni zmiennych decyzyjnych (b) otrzymane podczas rozwiązywania problemu ZDT-1 przez system stadny.

Zbliżonymi — bo także wielopopulacyjnymi — do wspomnianego

powyżej modelu stadnego są prezentowane w rozdz. 3 — modele koewolucyjne. Badania nad zastosowaniem tego typu podejścia w problemach optymalizacji wielokryterialnej potwierdzają, iż modele te w skuteczny sposób mogą zapobiegać przedwczesnej utracie różnorodności populacji, co jest szczególnie istotne właśnie w problemach wielomodalnych oraz wielokryterialnych. Ponieważ jednakże modele koewolucyjne stanowiły już przedmiot rozważań niniejszej monografii ich szersze omówienie zostanie w tym miejscu pominięte.

3.3. Hybrydowe systemy optymalizacji parametrów sieci neuronowych

Znalezienie parametrów sieci neuronowej, dzięki którym proces uczenia oraz jej działanie będą optymalne, wymaga szeregu prób lub głębokiej wiedzy eksperckiej. Algorytmy ewolucyjne mogą wspomóc poszukiwanie optymalnych sieci neuronowych, gdyż należą do grupy uniwersalnych algorytmów optymalizacji globalnej [34, 6, 32, 37, 38, 3]. Jednym z ciekawych, choć nie jedynym problemem, w którym zastosowanie sieci neuronowych, oraz ich optymalizacja może dać interesujące efekty jest problem predykcji (przewidywania) ciągów czasowych (definicja problemu - patrz [5, 29]).

Do rozwiązywania problemu predykcji można użyć wielu rodzajów sieci neuronowych, takich jak sieci *feed forward* (np. perceptron wielowarstwowy), sieci rekurencyjne (np. sieć Hopfielda), sieci samoorganizujące się (SOM) i inne. W opisywanym podejściu wybrano jako

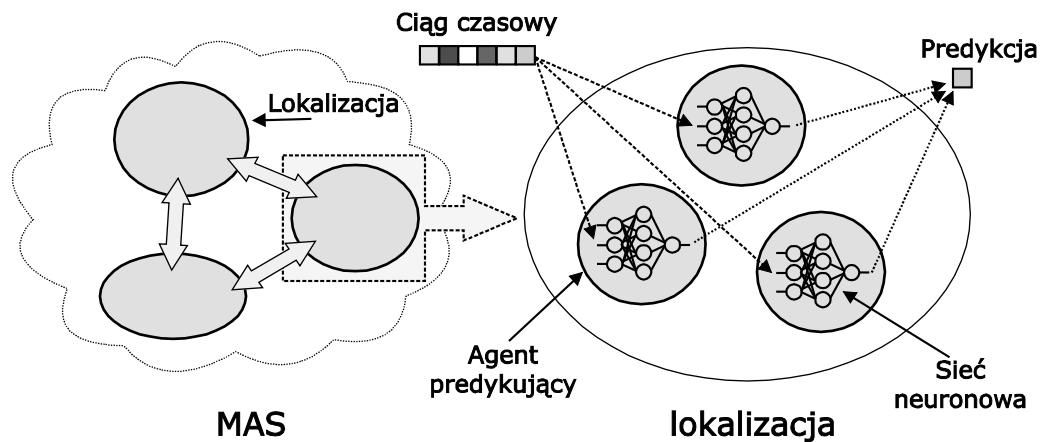
podstawową strukturę sieci neuronowej perceptron wielowarstwowy (predykcja za pomocą sieci neuronowych – patrz [13, 4, 31]).

Problem predykcji ciągów czasowych został wykorzystany jako przykład zastosowania wieloagentowego systemu predykcyjnego, którego zadaniem jest optymalizacja parametrów uczenia perceptronów wielowarstwowych.

W tekście przedstawiono koncepcję zastosowania EMAS do optymalizacji parametrów predykujących sieci neuronowych, immunologicznej metody selekcji w EMAS wprowadzanej w celu przyspieszenia obliczeń oraz wstępne wyniki eksperymentów.

3.3.1 Hybrydowy system optymalizacji parametrów predykujących sieci neuronowych

Rozważany system (skonstruowany wg modelu EMAS, patrz rozdz 3) składa się z grupy agentów, z których każdy jest wyposażony w sieć neuronową. Każdy agent przeprowadza predykcję kolejnego wyrazu ciągu czasowego, a następnie konieczne jest ustalenie zbiorczej odpowiedzi systemu na podstawie odpowiedzi otrzymanych od poszczególnych agentów (patrz rys. 8) [21, 11, 12, 9, 7, 8, 10].



Rysunek 8. Agentowy system predykcji ciągów czasowych

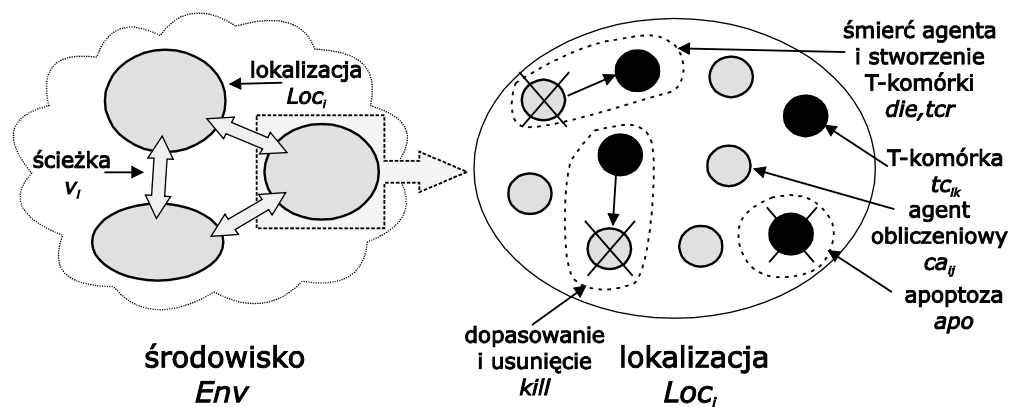
W tak zdefiniowanym systemie procesy ewolucyjne mogą służyć do polepszenia skuteczności predykcji. W rozważanym przypadku optymalizacji podlegają parametry sieci neuronowych używanych przez agentów w celu predykcji. Głównym celem systemu jest więc wykształcenie najlepszych sieci predykujących dla zadanego problemu.

Zadaniem agenta predykującego jest przewidywanie kolejnego wyrazu ciągu czasowego na podstawie pewnej liczby ostatnich jego wartości (zgodnie z neuronowym modelem predykcji NARX [31]). W tym celu, w trakcie swojego życia przeprowadza uczenie własnej sieci neuronowej, wykonując od jednego do kilku kroków uczenia na każdą iterację swojego życia (patrz rys. 8). Na podstawie jednej z metod szacowania błędu predykcji (patrz np. [27]) agent wylicza wartość swojej funkcji dostosowania.

3.3.2. Koncepcja mechanizmu selekcji immunologicznej

Koncepcja rozszerzenia EMAS o mechanizmy immunologiczne opiera

się na spostrzeżeniu, że nieprawidłowe (charakteryzujące się niską wartością funkcji dopasowania) fenotypy agentów pochodzą od nieprawidłowych genotypów, przejętych przez agentów w procesie reprodukcji. Dlatego proponuje się wprowadzenie do systemu dodatkowej populacji agentów immunologicznych zwanych T-komórkami, których głównym zadaniem będzie usuwanie z populacji osobników, które są nośnikami nieprawidłowych rozwiązań problemu. Określona w ten sposób koncepcja nosi nazwę iEMAS (immunological Evolutionary Multi-Agent System).



Rysunek 9: Zasada działania iEMAS

Podstawowym elementem iEMAS są przeciwciała — T-komórki, których działanie wzorowane jest na działaniu limfocytów obecnych w systemie odpornościowym kręgowców. Cykl życia T-komórki przebiega według następującego schematu (patrz rys. 9):

1. Stworzenie — w trakcie usunięcia agenta obliczeniowego w systemie powstaje jedna lub więcej T-komórek. Każda ze stworzonych

T-komórek otrzymuje wzorzec rozwiązania usuniętego agenta (wzorzec ten może być wcześniej zmutowany). Dzięki temu T-komórki stają się nośnikami nieprawidłowych rozwiązań.

2. Proces selekcji negatywnej — stworzone niedojrzałe T-komórki przez pewien czas testują agentów, jeśli rozpoznają agenta o wysokiej wartości funkcji dopasowania — są usuwane z systemu. Po ukończeniu procesu ich działania nie podlegają dalszym ograniczeniom.
3. Rozpoznanie agenta — T-komórka posiada wzorzec nieprawidłowego rozwiązania (jego struktura jest identyczna, jak struktura genotypu opisującego rozwiązywany problem) oraz korzysta z pewnej miary podobieństwa poszukując podobnych rozwiązań w systemie w celu ich usunięcia lub penalizacji (w najprostszych przypadkach funkcja ta oparta może być bezpośrednio na odległości Hamminga lub odległości euklidesowej — w zależności od postaci genotypu — odpowiednio binarnej lub zmiennoprzecinkowej).
4. Apoptoza — T-komórka po zakończeniu procesu selekcji negatywnej może przebywać w systemie jeszcze przez pewien, ustalony czas. Po jego zakończeniu, T-komórka zostaje usunięta z systemu, czas ten można jednak wydłużyć, nagradzając ją za prawidłowe rozpoznania agentów.

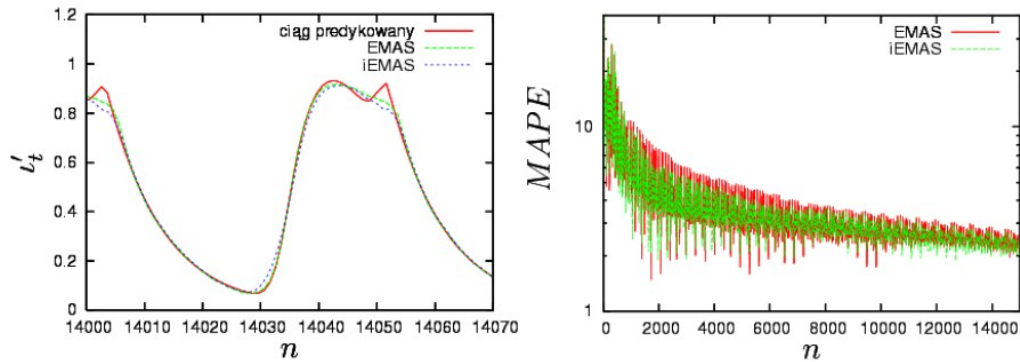
Wprowadzony sposób wspomagania selekcji prowadzi do obniżenia liczby agentów w populacji, dzięki wczesnemu usunięciu nieprawidłowych rozwiązań. Podniesiona w ten sposób presja selekcyjna nie powinna wpływać negatywnie na materiał genetyczny populacji, dzięki istnieniu procesu

selekcji negatywnej, który nie dopuszcza niezweryfikowanych T-komórek do oddziaływania na populację.

Usuwanie z populacji rozwiązań, które z punktu widzenia bieżącego celu są nieprawidłowe, bazując na zbiorze wzorców rozwiązań nieprawidłowych, kojarzyć się może z algorytmami poszukiwania z tabu [1]. Należy jednak zauważyć, iż w proponowanym podejściu nie istnieje globalnie dostępny zbiór rozwiązań będących tabu, zamiast tego wprowadzone T-komórki działają lokalnie, poszukując nieprawidłowych rozwiązań w swoim bezpośrednim sąsiedztwie. Można mówić tutaj o wprowadzeniu swoistego „zmiękczenia” poszukiwań z tabu.

3.3.3. Wyniki badań eksperymentalnych

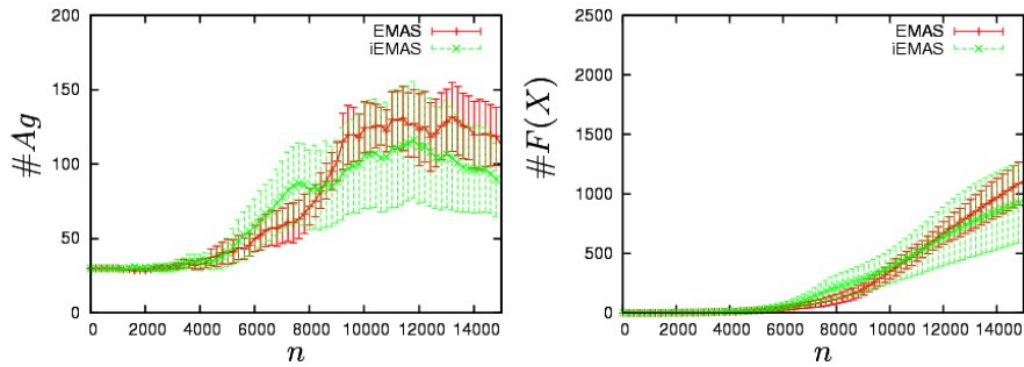
System został zaimplementowany w technologii Java, w oparciu o platformę agentową AgE (<http://age.iisg.agh.edu.pl>). Zadaniem systemu jest optymalizacji predykujących sieci neuronowych. Rozważanym problemem była predykcja chaotycznego ciągu Mackeya-Glassa często stosowanego do testowania systemów predykcyjnych, natomiast osobniki zostały wyposażone w jedne z najbardziej uniwersalnych sieci neuronowych – perceptrony wielowarstwowe.



Rysunek 10. Wyniki predykcji EMAS i iEMAS

Na rysunku 10, przedstawiono ciąg predykowany oraz rezultaty predykcji EMAS i iEMAS (w postaci nałożonych ciągów predykowanego oraz wyniku predykcji, oraz wykresu zależności błędu MAPE od numeru iteracji). Otrzymane wyniki wskazują na to, iż system jest w stanie poprawnie nauczyć się predykcji chaotycznego ciągu Mackeya-Glassa. Błąd predykcji zarówno w przypadku EMAS jak i iEMAS po 20000 kroków spada do wartości około 2%. W przypadku iEMAS obserwuje się również obniżenie liczby agentów (patrz rys. 11), natomiast efekty działania są porównywalne z osiągniętymi w przypadku EMAS. Obniżona została również liczba nowych agentów generowanych w systemie (patrz rys. 11). Dzieje się tak dlatego, gdyż w neuronowym systemie predykcyjnym obliczanie funkcji dopasowania jest rozciągnięte w czasie. Sieć neuronowa może więc zostać w pełni oceniona (pod względem zdolności do predykcji) po zakończeniu nauczania. Efektem zastosowania immunologicznej selekcji jest obniżenie liczby nowych osobników generowanych w systemie przy zachowaniu podobnej zdolności do predykcji. Jest to korzystne szczególnie z

tego względu, że tworzenie nowego osobnika wiąże się z koniecznością przeprowadzenia uczenia jego sieci neuronowej, co jest procesem kosztownym.



Rysunek 11. Stabilność populacji EMAS i iEMAS

4. Podsumowanie

Ewolucyjne systemy wieloagentowe stanowią nowy paradygmat obliczeń ewolucyjnych. Główną koncepcją leżącą u jego podstaw jest poddanie procesowi ewolucji populacji agentów występujących w systemie wieloagentowym.

Ewolucyjne systemy wieloagentowe charakteryzują się pewnym potencjałem nowych możliwości w porównaniu z klasycznymi obliczeniami ewolucyjnymi:

1. Agentowa architektura systemu zapewnia łatwość tworzenia systemów hybrydowych, łączących różnorodne techniki inteligencji obliczeniowej w ramach jednego spójnego modelu (*agentowe metody inteligencji obliczeniowej*).

2. Osobniki (agenci) nie są biernymi obiektami przemian ewolucyjnych, ale uczestnikami procesu ewolucyjnego, dzięki czemu w prosty sposób można modelować oddziaływania koewolucyjne oraz relacje społeczne.
3. Maksymalna decentralizacja procesu ewolucji (agenci jako jednostki ewolucji, brak scentralizowanego mechanizmu selekcji), dzięki której uzyskujemy relaksację synchronizacji obliczeń, a w efekcie zmniejszenie narzutów powodowanych przez mechanizmy synchronizacji.
4. Możliwość wprowadzenia w sposób naturalny mechanizmów i operatorów ewolucyjnych, których wykorzystanie w klasycznych obliczeniach ewolucyjnych napotyka na duże trudności (lub w ogóle nie jest możliwe). Jako przykład można tu podać operator *agregacji*.
5. Możliwość modelowania specjacji allopatrycznej (powstawanie gatunków w wyniku istnienia barier geograficznych) w oparciu o strukturę przestrzenną systemu *EMAS*, oraz specjacji sympatrycznej (powstawanie gatunków bez konieczności istnienia barier geograficznych, np. w wyniku oddziaływań koewolucyjnych) w oparciu o koewolucję gatunków i płci w systemie *CoEMAS*.
6. Mechanizm adaptacji systemu do rodzaju zadania oraz stopnia trudności problemu — przykładem może tu być adaptacja liczebności populacji do stopnia trudności problemu optymalizacji wielomodalnej w systemie *NCoEMAS* ([18]).

Ewolucyjne systemy wieloagentowe znalazły zastosowanie, między innymi, w optymalizacji wielomodalnej i optymalizacji wielokryterialnej.

Konstruowane w oparciu o model EMAS hybrydowe, agentowe systemy inteligencji obliczeniowej zostały natomiast zastosowane w predykcji ciągów czasowych i ewolucji sieci neuronowych. Oprócz tego interesującymi i obiecującymi obszarami badawczymi są zastosowania w problemach optymalizacji funkcji niestacjonarnych, problemach optymalizacji transportu, problemach poszukiwania optymalnych strategii dla robotów mobilnych i sztucznym życiu.

Niniejszy rozdział niewątpliwie nie wyczerpuje całej tematyki związanej z agentowymi metodami inteligencji obliczeniowej i ewolucyjnymi systemami wieloagentowymi. Wiele zagadnień nadal pozostaje otwartych i jest przedmiotem aktualnie prowadzonych badań.

Bibliografia

- [1] Arabas J.: *Wykłady z algorytmów ewolucyjnych*, WNT, Warszawa, 2001.
- [2] Bäck T., Fogel D., Michalewicz Z. (red.): *Handbook of Evolutionary Computation*, IOP Publishing and Oxford University Press, 1997.
- [3] Balakrishan K., Honavar V.: *Evolutionary design of neural architectures a preliminary taxonomy and guide to literature*, Raport ISU CS-TR 95-01, Artificial Intelligence Research Laboratory, 1995.
- [4] Basso M., Giarre L., Groppi S., Zappa G.: NARX models of an industrial power plant gas turbine, *IEEE transactions on control systems technology*, 13(4):599–604, 2005.
- [5] Box G. E. P., Jenkins G. M.: *Analiza szeregów czasowych*, PWN

Warszawa, 1983.

[6] Branke J.: Evolutionary algorithms in neural network design and training – A review, [w:] Alander J. T. (red.), *Proc. of the First Nordic Workshop on Genetic Algorithms and their Applications (1NWGA)*, str. 145–163, Vaasa, Finland, 1995.

[7] Byrski A., Bałamut J.: Evolutionary neural networks in collective intelligent predicting system, [w:] Rutkowski L., et.al. (red.), *Proc. of the Artificial Intelligence and Soft Computing ICAISC 2004 : 7th International Conference*, str. 823–828, Springer Verlag, 2004.

[8] Byrski A., Kisiel-Dorohinicki M.: Evolving RBF networks in a multi-agent system, *Neural Network World*, 12(2):433–440, 2002.

[9] Byrski A., Kisiel-Dorohinicki M.: Collective intelligence from a population of evolving neural networks, [w:] Mieczysław A. Kłopotek, Sławomir T. Wierzchoń K. T. (red.), *Proc. of the IIS: IIPWM'03 conference: Zakopane, Poland, June 2-5 2003*, str. 401–410, Springer Verlag, 2003.

[10] Byrski A., Kisiel-Dorohinicki M.: Immune-based optimization of predicting neural networks, [w:] et. al. V. S. S. (red.), *Proc. of the Computational Science ICCS 2005 : 5th International Conference, Atlanta, GA, USA*, Lecture Notes in Computer Science, LNCS 3516, str. 703–710, Springer Verlag, 2005.

[11] Byrski A., Kisiel-Dorohinicki M., Nawarecki E.: Agent-based evolution of neural network architecture, [w:] Hamza M. (red.), *Proc. of the IASTED Int. Symp.: Applied Informatics*, str. 242–247, IASTED/ACTA Press, 2002.

- [12] Cetnarowicz K., Kisiel-Dorohinicki M., Nawarecki E.: The application of evolution process in multi-agent world to the prediction system, [w:] Tokoro M. (red.), *Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS 1996)*, AAAI Press, Menlo Park, CA, 1996.
- [13] de A. Barreto G., Araújo A. F. R.: A self-organizing NARX network and its application to prediction of chaotic time series, [w:] *Proc. of the IEEE-INNS Int. Joint Conf. on Neural Networks, (IJCNN'01)*, tom 3, str. 2144–2149, Washington, D.C., 2001.
- [14] Deb K.: *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, 2001.
- [15] Deb K., Goldberg D. E.: An investigation of niche and species formation in genetic function optimization, [w:] Schaffer J. D. (red.), *Proceedings of the 3rd International Conference on Genetic Algorithms*, str. 42–50, Morgan Kaufmann, 1989.
- [16] Dreżewski R.: A model of co-evolution in multi-agent system, [w:] Marik V., Müller J., Pechoucek M. (red.), *Multi-Agent Systems and Applications III*, tom 2691 serii LNCS, str. 314–323, Springer-Verlag, Berlin, Heidelberg, 2003.
- [17] Dreżewski R.: Co-evolutionary multi-agent system with speciation and resource sharing mechanisms, *Computing and Informatics*, 25(4):305–331, 2006.
- [18] Dreżewski R.: Koewolucyjne techniki optymalizacji funkcji wielomodalnych z zastosowaniem technologii agentowej. Rozprawa

doktorska, AGH University of Science and Technology, 2005.

[19] Dreżewski R., Cetnarowicz K.: Niching techniques based on sexual conflict in co-evolutionary multi-agent system, [w:] Lefranc G. (red.), *Proceedings of the Conference on Management and Control of Production and Logistics (MCPL 2004)*, str. 307–312, IFAC/IEEE/ACCA, Universidad de Las Américas, 2004.

[20] Goldberg D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, N. Y., 1989.

[21] Kisiel-Dorohinicki M., Klapper-Rybicka M.: Ewolucja sieci neuronowych w srodowisku wieloagentowym, [w:] *Zeszyty Naukowe Uniwersytetu Jagiellockiego*, UJ, 1998.

[22] Kisiel-Dorohinicki M.: Agent-oriented model of simulated evolution, [w:] Grosky W. I., Plasil F. (red.), *SofSem 2002: Theory and Practice of Informatics*, tom 2540 serii *Lecture Notes in Computer Science*, Springer-Verlag, 2002.

[23] Kisiel-Dorohinicki M.: Flock-based architecture for distributed evolutionary algorithms, [w:] Rutkowski L., Siekmann J., Tedeusiewicz R., Zadeh L. (red.), *Artificial Intelligence and Soft Computing – ICAISC 2004*, tom 3070 serii *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2004.

[24] Koza J. R.: *Genetic Programming. On the Programming of Computers by Means of Natural Selection*, A Bradford Book. The MIT Press, Cambridge, Massachusetts, 1992.

[25] Koza J. R.: *Genetic Programming II. Automatic Discovery of reusable Programs*, A Bradford Book. The MIT Press, Cambridge,

Massachusetts, 1994.

[26] Mahfoud S. W.: *Niching methods for genetic algorithms*, Rozprawa doktorska, University of Illinois at Urbana-Champaign, Urbana, IL, USA, 1995.

[27] Masters T.: *Neural, Novel & Hybrid Algorithms for Time Series Prediction*, Wiley & Sons, 1995.

[28] Michalewicz Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, Springer -Verlag, 1996.

[29] Mulawka J.: *Systemy Ekspertowe*, WNT Warszawa, 1996.

[30] Nawarecki E., Kisiel-Dorohinicki M., Dobrowolski G.: Organisations in the particular class of multi-agent systems, [w:] Dunin-Keplicz B., Nawarecki E. (red.), *From Theory to Practice in Multi Agent Systems*, tom 2296 serii *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2002.

[31] Parlos A., Rais O., Atiya A.: Multi-step-ahead prediction using dynamic recurrent neural networks, *Neural Networks*, 13(7):765–786, 2000.

[32] Potter M. A.: *The Design and Analysis of a Computational Model of Cooperative Coevolution*, Rozprawa doktorska, George Mason University, Fairfax, Virginia, 1997.

[33] Rutkowski L.: *Metody i techniki sztucznej inteligencji*, PWN, 2006.

[34] Schaefer R.: *Podstawy genetycznej optymalizacji globalnej*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków, 2002.

[35] Siwik L., Kisiel-Dorohinicki M.: Elitism in agent-based evolutionary multiobjective optimization, *Inteligencia Artificial, Special issue: New trends*

on Multiagent systems and soft computing, 9(28), 2005.

[36] Siwik L., Kisiel-Dorohinicki M.: Semi-elitist evolutionary multi-agent system for multiobjective optimization, [w:] Alexandrov V. N., van Albada G. D., Sloot P., Dongara J. (red.), *Proceedings of International Conference on Computational Science ICCS 2006*, tom 3993 serii LNCS, Springer-Verlag, 2006.

[37] Yao X.: *A Review of Evolutionary Artificial Neural Networks*, Commonwealth Scientific and Industrial Research Organization., Victoria, Australia, 1992.

[38] Yao X.: Evolutionary artificial neural networks, [w:] Kent A., Williams J. G. (red.), *Encyclopedia of Computer Science and Technology*, tom 33, str. 137–170, Marcel Dekker Inc., 1995.