

Edward Nawarecki*, Leszek Siwik*

Agentowo-ewolucyjna platforma obliczeń optymalizacyjnych

1. Wstęp

Metodologie agentowe zyskują w ciągu ostatnich lat stale rosnącą popularność. Pojawiają się liczne doniesienia o coraz to nowych obszarach zastosowania tej klasy systemów (m.in. realizacja usług, systemy informacyjne, rozproszone systemy sterowania, modelowanie i symulacja). Akronim MAS (*multi agent system*) jest powszechnie stosowany, zarówno w literaturze naukowej, jak i w materiałach firmowych i popularyzatorskich.

Równocześnie, ze względu na bogactwo możliwości i dużą złożoność jakimi charakteryzuje się ta klasa systemów, wciąż jeszcze pozostaje szerokie pole dla prac badawczych i aplikacyjnych, co sprawia, że MAS znajduje się w centrum zainteresowania wielu ośrodków badawczych i firm komercyjnych.

Duża elastyczność i wieloaspektowość rozwiązań opartych na paradygmacie agentowym stwarza duże możliwości interpretacyjne i swobodę w tworzeniu koncepcji, wzorowanych m.in. na obserwacji procesów zachodzących w świecie rzeczywistym.

W artykule naszkicowano pewną koncepcję organizacji MAS opartą na zastosowaniu pojęcia energii życiowej agenta, odniesionego do podejmowanych przez niego działań. Bezpośrednim obszarem zastosowania tego podejścia jest zrealizowana platforma obliczeniowa, zorientowana na rozwiązywanie zadań optymalizacji wielokryterialnej. Opisano założenia realizacyjne i funkcjonalność tej platformy, podając wybrane przykłady rozwiązań uzyskanych przy jej użyciu.

2. System agentowy a rzeczywistość

Kontekst, w jakim rozważać można relację pomiędzy systemem agentowym a światem rzeczywistym, zależy głównie od roli, jaką ma spełniać konkretny system agentowy.

* Katedra Informatyki, Akademia Górniczo-Hutnicza w Krakowie

Z czysto użytkowego punktu widzenia wskazać można [1–5, 17–18]:

- Systemy agentowe działające w świecie rzeczywistym lub w bezpośrednim kontakcie z tym światem. Do tej klasy zaliczyć należy m.in. grupy inteligentnych robotów, inteligentne systemy sterowania ruchem pojazdów, różnego typu organizacje (społeczne, polityczne, bussinesowe).
- Systemy agentowe w środowiskach wirtualnych, stanowiące modele zjawisk zachodzących w świecie rzeczywistym (np. wykorzystywane do symulacji) lub realizujące procesy całkowicie abstrakcyjne w sposób dostosowany do środków i możliwości oferowanych przez systemy informatyczne (wirtualizacja działań i usług, inteligentne systemy obliczeniowe).

Odnosząc się do znanych realizacji wspomnianych wyżej klas systemów wirtualnych, a w tym agentowych, znamienny wydaje się fakt, że nawet przy rozwiązywaniu całkowicie abstrakcyjnych problemów (jak to ma miejsce w inteligencji obliczeniowej) wykorzystywane są różne analogie do świata rzeczywistego oraz zachodzących w nim procesów społecznych i biologicznych.

Obok sieci neuronowych, chyba najbardziej charakterystyczną klasą systemów wirtualnych, opartych na inspiracji biologicznej, stanowią algorytmy ewolucyjne.

Nie wdając się w ogólnie znane zasady działania tej klasy algorytmów, w dalszym ciągu artykułu starano się wyeksponować specyficzne możliwości, jakie występują przy agentowej realizacji tej klasy systemów obliczeniowych. Możliwości takie pojawiają się przy poszukiwaniu znaczenia dalej idących analogii do świata rzeczywistego, niż to wynika z samej selekcji populacji osobników i zastosowania klasycznych operatorów genetycznych.

Za punkt wyjścia do prowadzonej dalej dyskusji przyjęto ogólny model systemu agentowego wyrażony w postaci piątki:

$$\text{MAS} = \langle \text{Ag}, \text{Ev}, \text{Cm}, \text{Rs}, \text{Og} \rangle \quad (1)$$

gdzie:

- Ag* – zbiór agentów działających w MAS,
- Ev* – środowisko,
- Cm* – komunikacja,
- Rs* – zasoby,
- Og* – organizacja.

Przystępując do tworzenia koncepcji realizacji takiego systemu, należy wypracować pewną wizję jego funkcjonowania. Jeśli ma to być system wirtualny (jak to ma miejsce w przypadku systemu obliczeniowego), wówczas istnieje duża swoboda w definiowaniu i określaniu funkcjonalności poszczególnych jego komponentów. I tutaj właśnie, w sposób naturalny pojawiają się odniesienia do procesów znanych z obserwacji świata rzeczywistego.

W przyjętym rozwiązaniu, kluczową rolę w kształtowaniu zachowań agenta odgrywa jego parametr określany jako **energia życiowa**, która jest równocześnie podstawowym

(a często jedynym) zasobem (R_s) występującym w systemie. W chwili wprowadzenia agenta do systemu (populacja początkowa, reprodukcja) nadawany mu jest kwant energii, którego wartość zależy od oceny rozwiązania zawartego w genotypie agenta (wartości funkcji przystosowania).

W kolejnych krokach działania systemu wartość energii życiowej agenta może ulegać zmianie, w zależności od wykonywanych przez niego akcji. Można więc napisać:

$$\xi_{ai}(n+1) = \xi_{ai}(n) \pm e_{ai}(r_{ij}(n)) \quad (2)$$

gdzie:

- $\xi_{ai}(n)$ – energia życiowa agenta $a_i \in Ag$ w n -tym kroku algorytmu,
- $e_{ai}(n)$ – zmiana energii agenta, w danym kroku,
- $r_{ij}(n)$ – j -ta akcja agenta a_i wykonana w kroku n .

Wartość energii wpływa na możliwości wykonywania funkcji życiowych agenta, do których należą m.in.: poruszanie się w środowisku (spotkania, migracje), kreacja nowego agenta (reprodukcja) oraz śmierć agenta (usunięcie z systemu).

Do każdego z tych działań przypisany jest pewien zakres posiadanej przez agenta energii warunkujący możliwość jego realizacji. Można więc powiedzieć, że energia życiowa stanowi pewien rodzaj wielkości sterującej w sposób automatyczny działaniem agenta.

Równocześnie energia ta interpretowana być może jako pewna odpowiedź środowiska na ciąg działań agenta.

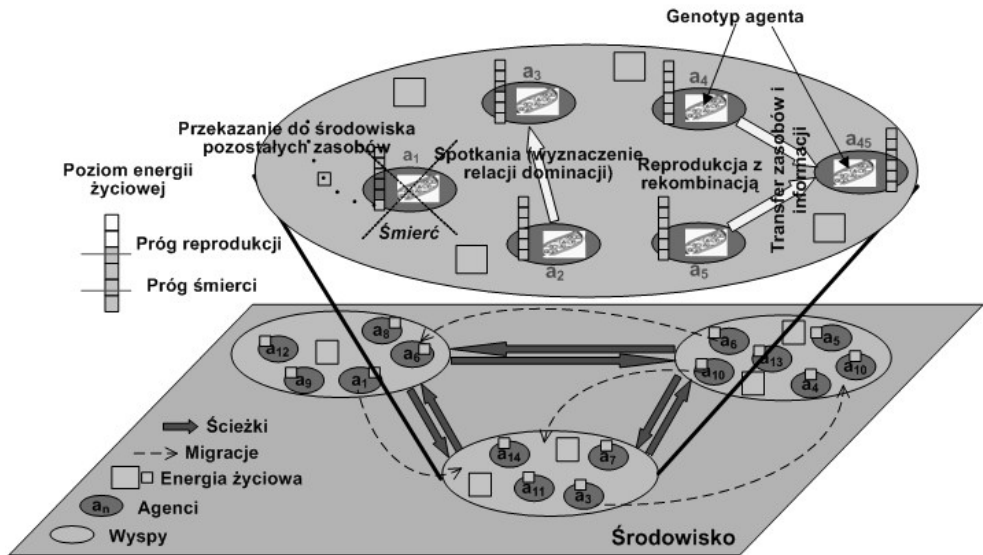
Przyjęcie opisanego trybu działania pojedynczego agenta umożliwia wprowadzenie pewnych zasad w zakresie organizacji systemu (Og). Wyodrębniono następujące relacje pomiędzy agentami, których zaistnienie uwarunkowane jest również stosunkiem ich energii życiowej:

- neutralność (gdy wartości energii agentów należą do strefy ich biernego zachowania);
- partnerstwo (jeżeli wartość energii pary agentów umożliwia reprodukcję, czyli wykrowanie nowego agenta, połączone z krzyżowaniem genotypów agentów „rodzicielskich”);
- dominacja (w przypadku, gdy występuje znaczna różnica wartości energii dla pary agentów, wówczas agent dominujący przejmuje część energii agenta zdominowanego).

Na organizację systemu składa się również przyjęta struktura środowiska (Ev). W opisywanej koncepcji wprowadzono strukturę grafową, przy czym agenci grupowani są w poszczególnych węzłach, zaś krawędzie określają możliwość migracji, czyli przechodzenia agentów pomiędzy węzłami.

Opisane powyżej relacje pomiędzy agentami mogą zachodzić w ramach jednego węzła (wyspy), zaś zmiana węzła (migracja) następuje wówczas, gdy agent (spełniający odpowiednie uwarunkowania energetyczne) decyduje się na poszukiwanie bardziej dogodnego dla siebie obszaru działania.

Wreszcie ostatnim komponentem systemu, posiadającym wpływ na jego organizację, jest komunikacja (*Cm*). Zdolność do komunikacji umożliwia agentom pozyskiwanie informacji, które ułatwiają im podjęcie racjonalnych decyzji – szczególnie w odniesieniu do poszukiwania partnerów lub decyzji o migracji. W zależności od wariantu systemu, możliwości komunikacji mogą być zapewnione w ramach poszczególnych węzłów, lub także pomiędzy węzłami.



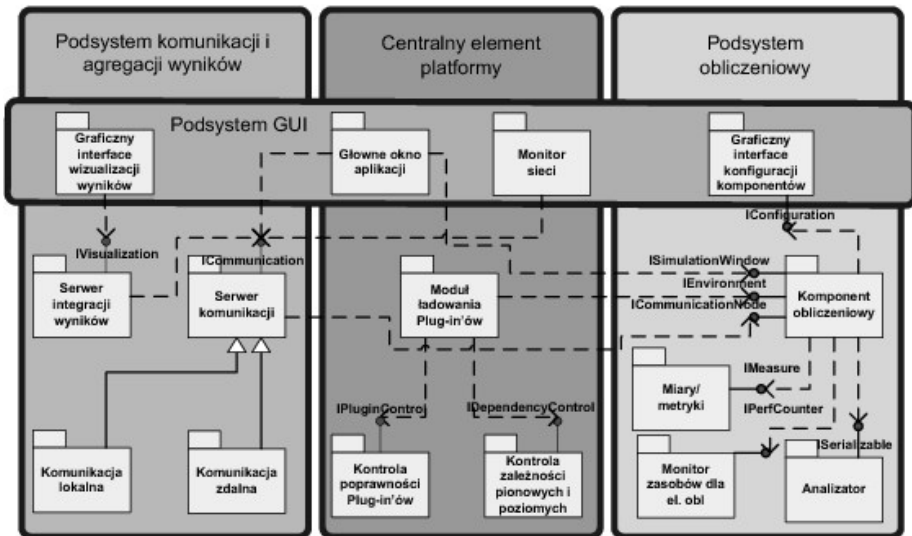
Rys. 1. Struktura i zarys koncepcji systemu EMAS

Zaprezentowana koncepcja, przedstawiona z konieczności w formie skróconej, może być wykorzystana przy realizacji dość różnorodnych systemów agentowo-ewolucyjnych (EMAS), stanowiących wirtualne środowisko obliczeniowe. Poglądowo, ilustrację tej koncepcji stanowi schemat przedstawiony rysunku 1.

3. Platforma NEvol

W trakcie prowadzonych badań konieczne stało się zaprojektowanie, a następnie zaimplementowanie agentowej (agentowo-ewolucyjnej) platformy obliczeniowej mającej charakter infrastrukturalny i umożliwiającej implementację i eksperymentalne zweryfikowanie zaproponowanych modeli obliczeniowych – platformie tej nadano kodową nazwę NEvol (*.Net Evolutionary Computing Platform*).

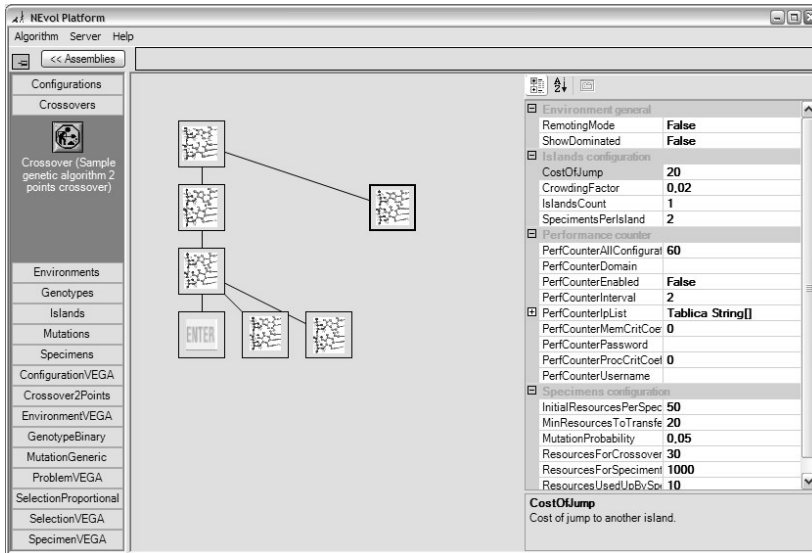
Uwzględniając wszystkie wymagania stawiane platformie, dokonano doprecyzowania modelu obliczeniowego jaki powinien być przez nią wspierany. Zgodnie z charakterystyką tworzonych systemów, przyjęto model hierarchiczny uwzględniający trzy, bądź cztery, poziomy.



Rys. 2. Schemat architektury platformy NEVol

Biorąc pod uwagę wszystkie stawiane konstruowanej platformie wymagania, zdecydowano się na jej komponentową i plug-inową budowę, zawierającą następujące podsystemy (por. rys. 2):

- Centralny element platformy – odpowiedzialny za poprawne załadowanie plug-in'ów (komponentów obliczeniowych) oraz ich części. Podsystem ten dokonuje kontroli poprawności i zgodności ładowanego plug-inu oraz kontroli zależności poziomych i pionowych wczytanego modułu z pozostałymi, załadowanymi elementami. Twórca komponentu (plug-inu) na etapie jego definiowania określa jakich i ile elementów podrzędnych oraz równorzędnych (to właśnie tzw. zależności pionowe oraz poziome) wymaga on do prawidłowego działania.
- Podsystem komunikacji i agregacji wyników – to podsystem umożliwiający komunikację pomiędzy poszczególnymi elementami środowiska obliczeniowego. Komunikacja w trybie rozproszonym jest przezroczysta dla elementów platformy. Podsystem ten umożliwia również gromadzenie informacji (integrację wyników) ze wszystkich węzłów obliczeniowych i przekazywanie ich do wyższych warstw aplikacji, np. do podsystemów GUI.
- Podsystem obliczeniowy – jest to element zdolny do przeprowadzenia zadanych obliczeń.
- Podsystem GUI – podsystem warstwy prezentacji udostępniający użytkownikowi w sposób czytelny (wykresy) i przejrzysty (optymalizacja i agregacja) wyniki obliczeń. Udostępnia również mechanizmy konfiguracji komponentu obliczeniowego oraz podsystemów zależnych, takich jak np. podsystem komunikacji.



Rys. 3. Okno komponowania algorytmu obliczeniowego platformy NEvol

Na rysunku 3 przedstawiono okno graficznego komponowania algorytmu obliczeniowego platformy NEvol. Lista zakładek po lewej stronie okna zawiera zgrupowane, dostępne w systemie (realizowane jako plug-iny) gotowe komponenty realizujące wybrane fragmenty/elementy ewolucyjnego bądź agentowo-ewolucyjnego modelu obliczeniowego (np. określony schemat reprodukcji, rekombinacji, selekcji, krzyżowania etc.).

Twórca komponentów może definiować wiele plug-inów realizujących określoną funkcjonalność w odmienny sposób (krzyżowanie jedno-, dwu- lub wielopunktowe, selekcja proporcjonalna, turniejowa etc.), czy też definiujących w odmienny sposób określony element struktury obliczenia (środowiska realizujące różne strategie zarządzania wyspami, wyspy realizujące różne schematy zarządzania agentami etc.). Każdy taki element, implementujący odpowiedni interfejs programowy oraz definiujący określone zależności pionowe oraz poziome, dostępny jest z punktu widzenia biernego użytkownika platformy jako jeden z elementów w odpowiedniej zakładce okna komponowania algorytmów.

Użytkownik platformy odpowiedzialny za przeprowadzenie obliczeń (decydent lub badacz/analityk) na etapie komponowania algorytmu (przeciągania określonych ikon reprezentujących wybrane fragmenty algorytmu bądź środowiska do obszarów aktywnych okna komponowania) informowany jest o tym, jakie inne elementy wymagane są przez określony element do prawidłowego działania. I tak np. po przeciągnięciu do okna komponowania algorytmu ikony reprezentującej środowisko, użytkownik informowany jest o tym, iż wybrane środowisko do swojego poprawnego działania wymaga zdefiniowania wyspy, z którą powinno współpracować oraz podania konfiguracji środowiska (por. rys. 3).

Przecignięcie z okna zakładki ikony reprezentującej określoną wyspę czy konfigurację do aktywnego obszaru widocznego w oknie kompozycji algorytmu oznacza dokonanie przez użytkownika wyboru określonej realizacji (implementacji) wybranego elementu. Określona kompozycja algorytmu może zostać zapisana, a następnie wczytana w dowolnym momencie przez użytkownika jako gotowy do użycia i skonfigurowany algorytm obliczeniowy.

Platforma NEvol pozwala na monitorowanie, gromadzenie, agregację oraz wizualizację bieżących (a także archiwalnych) wyników obliczeń – bez względu na to, czy obliczenia odbywają się lokalnie (wszystkie węzły obliczeniowe pracują na jednej fizycznej stacji obliczeniowej) czy też w sposób rozproszony (węzły obliczeniowe (wyspy) uruchamiane są na (wielu) różnych fizycznych stacjach obliczeniowych).

Na etapie prac projektowych oraz implementacyjnych zachowano ogólność zastosowania platformy – w szczególności uniknięto ograniczenia jej zastosowania wyłącznie do rozwiązywania zadań optymalizacji wielokryterialnej. Niemniej, ponieważ w przyjętej perspektywie, zasadniczym celem prowadzonych badań było rozwiązywanie problemów poli-optymalizacji, po zaimplementowaniu rdzenia platformy, skoncentrowano się na realizacji jej warstwy użytkowej pozwalającej na monitorowanie i wizualizację najważniejszych charakterystyk związanych właśnie z oceną działania algorytmów optymalizacji wielokryterialnej.

Do najważniejszych jej elementów w tym zakresie należą m.in.: wizualizacja (dwuwymiarowych) frontów Pareto (w funkcji czasu oraz funkcji kroku obliczeniowego), wizualizacja (dwuwymiarowych) zbiorów Pareto (w funkcji czasu oraz funkcji kroku obliczeniowego), a także wyznaczanie i wizualizacja (w funkcji czasu oraz funkcji kroku obliczeniowego) najważniejszych metryk pozwalających oceniać i porównywać jakość otrzymanywnych wyników. Wśród zaimplementowanych metryk znalazły się m.in. takie jak: Hypervolume (HV) [6], Hypervolume Ratio (HVR) [6], Spread [7], Error Ratio (ER) [6] etc.

4. Wybrane wyniki badań eksperymentalnych

Badania eksperymentalne, weryfikujące słuszność przyjętych koncepcji przeprowadzano m.in. z wykorzystaniem zestawu problemów referencyjnych ZDT (Zitzler-Deb-Thiele) [8].

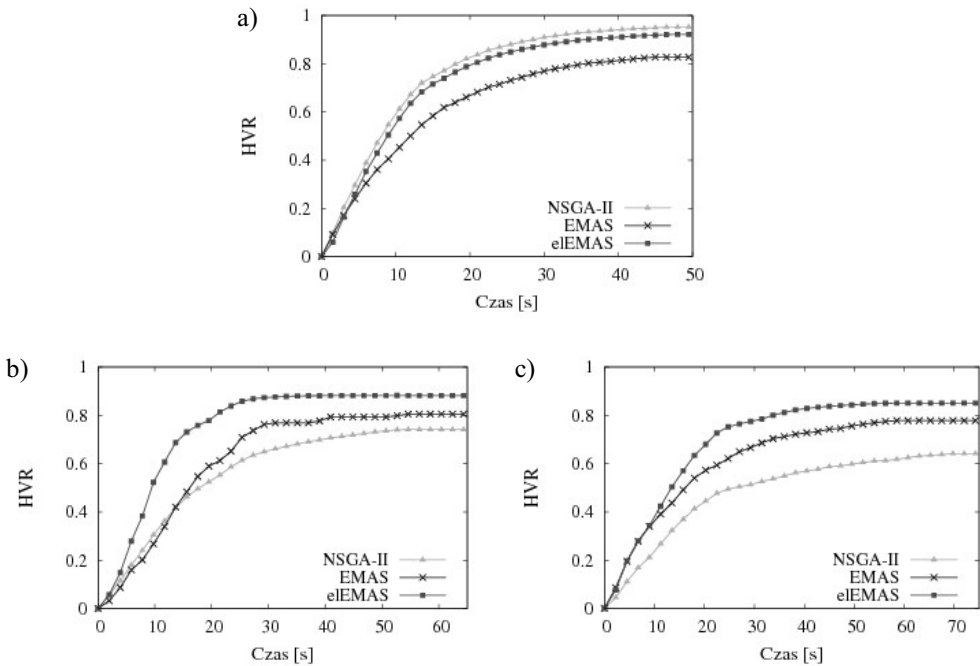
Jednym z istotnych aspektów prowadzonych badań eksperymentalnych było uzyskanie oceny efektywności działania testowanych algorytmów, w charakterystycznych i ważnych dla rzeczywistych zadań optymalizacyjnych, środowiskach zaszumionych. W środowiskach takich uzyskiwane i przetwarzane przez algorytm optymalizacyjny dane, a także sam proces obliczeniowy, mogą być przedmiotem szeregu zniekształceń i zakłóceń [9]. Podczas prowadzonych badań eksperymentalnych, zdecydowano, iż zaszumienie środowiska modelowane będzie poprzez zniekształcenie z określonym prawdopodobieństwem

wyniku porównania jakości reprezentowanych przez agentów rozwiązań (co ma miejsce podczas realizowanych przez agentów spotkań). Ponieważ podczas badań skoncentrowano się na rozwiązywaniu zadań optymalizacji wielokryterialną w sensie Pareto [10] – porównane jakości (i zniekształcanie otrzymywanego rezultatu podczas modelowania środowiska zaszumionego) realizowane było w oparciu o słabą relację dominacji [11].

Z oczywistych względów, zaprezentowanie wszystkich otrzymanych charakterystyk nie jest w tym miejscu możliwe, dlatego też ograniczono się do zaprezentowania charakterystyk czasowych metryki HVR [6] uzyskiwanych podczas rozwiązywania problemu ZDT4 odpowiednio w środowisku niezaszumionym, środowisku zaszumionym z prawdopodobieństwem wystąpienia szumu równym 10% oraz środowisku zaszumionym z prawdopodobieństwem wystąpienia szumu równym 15%.

Jednym z celów prowadzonych badań było potwierdzenie tezy, zgodnie z którą w oparciu o paradygmat obliczeń agentowo-ewolucyjnych możliwe jest skonstruowanie metod polioptymalizacji, które w określonych warunkach charakteryzowałyby się większą efektywnością niż klasyczne (nieagentowe) algorytmy ewolucyjne optymalizacji wielokryterialnej. Dlatego też na wykresach z rysunku 4 prezentowane są w ujęciu porównawczym m.in. przebiegi czasowe metryki HVR uzyskiwane przez uznawany za jeden z najbardziej efektywnych (i z pewnością najpowszechniej wykorzystywany zarówno w literaturowych rozważaniach teoretyczno-analitycznych, jak i w praktycznych zastosowaniach) algorytm Non-Dominated Sorting Genetic Algorithm II (NSGA-II) zaproponowany w 2001 r. przez zespół kierowany przez K. Deba [7].

Ponieważ podczas prowadzonych prac okazało się (zwłaszcza dla eksperymentów w środowiskach niezaszumionych), że podstawowy model EMAS znacząco ustępował jakością uzyskiwanych rezultatów rezultatom uzyskiwanym z wykorzystaniem wspomnianego (referencyjnego) algorytmu NSGA-II (co wynikało w głównej mierze z obserwowanej w przypadku podstawowego modelu EMAS tendencji do stosunkowo szybkiej stagnacji zachodzących w systemie procesów ewolucyjnych) – w wyniku dalszych prowadzonych prac zaproponowano model wykorzystujący m.in. mechanizmy przypominające operatory elitarne z klasycznych algorytmów ewolucyjnych (aczkolwiek realizowane oczywiście w sposób zdecentralizowany i rozproszony – a zatem zgodny z duchem podejścia agentowego). Mechanizmy te pozwoliły na znaczącą (w stosunku do modelu podstawowego) poprawę efektywności uzyskiwanych rezultatów, a zaproponowany model określony został mianem modelu eEMAS (*elitist Evolutionary Multi-Agent System*) [12]. Aby możliwe było zaprezentowanie wpływu mechanizmów zaproponowanych w modelu eEMAS, a także potwierdzenie postawionej tezy o możliwości skonstruowania systemów agentowo-ewolucyjnych dedykowanych do rozwiązywania zadań polioptymalizacji charakteryzujących się – w określonych warunkach – efektywnością lepszą niż ta, obserwowana w przypadku podejść klasycznych (nieagentowych), na wykresach z rysunku 4 zamieszczono obok charakterystyk algorytmu NSGA-II także przebiegi czasowe metryki HVR uzyskane z wykorzystaniem zarówno podstawowego modelu EMAS, jak i wspomnianego modelu elitarnego eEMAS.

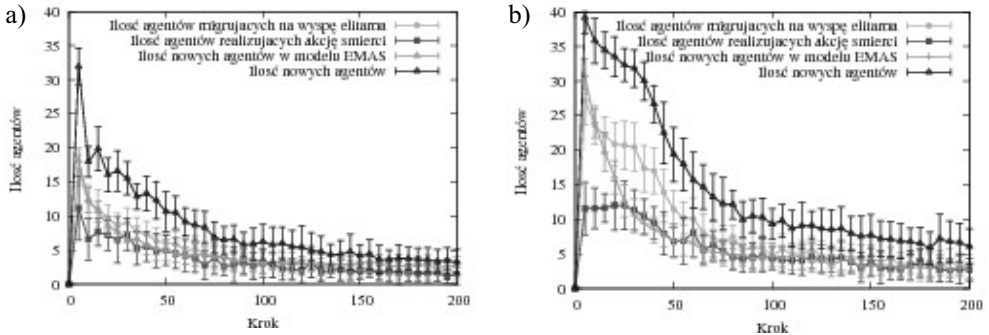


Rys. 4. Charakterystyki czasowe metryki HVR dla problemu ZDT4 otrzymywane przez systemy EMAS, eLEMAS oraz algorytm NSGA-II w środowisku niezaszumionym (a) zaszumionym z prawdopodobieństwem wystąpienia szumu równym 10% (b) oraz zaszumionym z prawdopodobieństwem wystąpienia szumu równym 15%

Podsumowując rezultaty prezentowane na rysunku 4, stwierdzić z pewnością należy iż:

- W środowisku niezaszumionym (por. rys. 4a) algorytm NSGA-II jest najbardziej efektywnym algorytmem spośród podejść poddanych testom. Wartości metryki HVR uzyskiwane przez model eLEMAS zbliżyły się do tych otrzymywanych przez algorytm NSGA-II, zaś wartości metryki HVR uzyskiwane przez podstawowy model EMAS bardzo wyraźnie ustępowały dwóm pozostałym podejściom.
- Wprowadzenie do środowiska zaszumienia, z 10-procentowym prawdopodobieństwem jego wystąpienia (por. rys. 4b), spowodowało pogorszenie charakterystyk czasowych dla wszystkich trzech testowanych podejść, jednak najbardziej wyraźny negatywny wpływ obserwowany był w przypadku algorytmu NSGA-II. Podczas tej serii eksperymentów najbardziej efektywnym podejściem okazał się model eLEMAS, nieco gorsze rezultaty uzyskiwane były przez model EMAS, zaś najmniej efektywnym podejściem okazał się algorytm NSGA-II.
- Zwiększenie stopnia zaszumienia środowiska poprzez zwiększenie prawdopodobieństwa jego wystąpienia do 15%, doprowadziło do dalszego pogorszenia charakterystyk

czasowych metryki HVR dla wszystkich trzech analizowanych systemów (por. rys. 4c), i dalsze podtrzymanie trendu zgodnie z którym wprowadzanie do środowiska szumów ma wyraźnie najbardziej negatywny wpływ na efektywność algorytmu NSGA-II.



Rys. 5. Ilość agentów: nowych, nowych w modelu EMAS, realizujących akcję śmierci oraz migrujących na wyspę elitarną dla problemów MaxEx (a) oraz Kursawe (b) po wprowadzeniu do podstawowego modelu EMAS wysp elitarnych i związanych z nimi migracji

Aby zaprezentować pozytywny wpływ mechanizmów wprowadzanych w modelu elitarnym eEMAS (a nieobecnych w podstawowym modelu EMAS), na rysunku 5 zaprezentowano charakterystyki związane z ilością nowych pojawiających się w systemie osobników, ilością osobników migrujących na wyspy elitarne oraz ilością osobników wykonujących akcję śmierci. Aby ułatwić analizę, na wykresach tych zamieszczono także charakterystykę średniej ilości nowych osobników wprowadzanych do populacji w podstawowym modelu EMAS. Charakterystyki te potwierdzają pozytywny wpływ na dynamikę populacji agentów wprowadzenia do podstawowego modelu EMAS wysp elitarnych i związanych z nimi migracji. Obniżenie średniego wieku agentów skutkuje zwiększeniem dynamiki procesu ewolucyjnego, mierzonej ilością nowych agentów wprowadzanych do populacji i pozwala na zwiększenie jej zdolności do intensywniejszej eksploracji przestrzeni poszukiwań.

5. Wnioski

Uzyskane rezultaty obliczeń polioptymalizacyjnych wskazują, że zrealizowane algorytmy charakteryzują się dużą efektywnością – szczególnie w warunkach występowania zakłóceń przy określaniu wartości funkcji przystosowania. Potwierdza to zasadność zastosowanych rozwiązań, jak też funkcjonalność platformy NEvol.

Pełna ocena możliwości prezentowanego środowiska obliczeniowego wymaga oczywiście przeprowadzenia szeregu badań eksperymentalnych dotyczących m.in. czasów obli-

czeń, skalowalności oraz wpływu poszczególnych parametrów algorytmu na dokładność uzyskanych rezultatów¹. Wydaje się jednak, że już dotychczasowe wyniki w pełni potwierdzają nowe możliwości jakie stwarza zastosowanie proponowanych mechanizmów w ewolucyjnych środowiskach obliczeniowych klasy EMAS. Słuszność przyjętych kierunków badawczych oraz decyzji projektowych potwierdza także bogaty zbiór innych (nieomawianych z konieczności w tym artykule) modeli agentowo-ewolucyjnych, których prototypy z powodzeniem zostały zrealizowane (i wstępnie przebadane) z wykorzystaniem omawianych koncepcji [13, 14] i prezentowanej platformy [12, 15–16].

Literatura

- [1] Turek W., Marcjan R., Cetnarowicz K., *Agent-based mobile robots navigation framework*. [w:] International Conference on Computational Science, ICCS 2006, LNCS, 3993, 775–782, Springer-Verlag 2006.
- [2] Koźlak J., *Agent approach for transportation modelling and optimisation*. [w:] 9th IFAC workshop on Intelligent Manufacturing Systems (IMS'08), Szczecin 2008.
- [3] Koźlak J., Nawarecki E., *Agent environment for supply chain modelling and management*. [w:] 9th IFAC workshop on Intelligent Manufacturing Systems (IMS'08), Szczecin 2008.
- [4] Turek W., Marcjan R., Cetnarowicz K., *A universal tool for multirobot system simulation*. [w:] Knowledge – driven computing : knowledge engineering and intelligent computations/eds. Carlos Cotta, Simeon Reich, Robert Schaefer, Antoni Ligęza, 289–303, Studies in Computational Intelligence, Springer-Verlag 2008.
- [5] Cetnarowicz K., Rojek G., *Etyczno-społeczne podejście do detekcji zagrożeń w systemach agentowych*. [w:] Rozwój informatycznych systemów wieloagentowych w środowiskach społeczno-gospodarczych, Warszawa 2008.
- [6] Van Veldhuizen D., *Multiobjective evolutionary algorithms: classifications, analyses and new innovations*. Rozprawa doktorska, Graduate School of Engineering of the Air Force Institute of Technology, 1999.
- [7] Deb K., Pratap A., Agarwal S., Meyarivan T., *A fast and elitist multi-objective genetic algorithm: NSGA-II*. IEEE Transaction on Evolutionary Computation, 6(2): 2002, 181–197.
- [8] Zitzler E., Deb K., Thiele L., *Comparison of multiobjective evolutionary algorithms: Empirical results (revised version)*. Raport wewnętrzny: Institut für technische Informatik und Kommunikationsnetze, Zurich 1999.
- [9] Bui L.T., Abbas H.A., Essam D., *Localization and fitness inheritance in solving noisy multiobjective optimization problems*. The Artificial Life and Adaptive Laboratory, School of Information Technology and Electrical Engineering, University of New South Wales, Raport Wewnętrzny: TR-ALAR-200605011.
- [10] Deb, K., *Multi-objective optimization using evolutionary algorithms*. J.Wiley and Sons 2001.
- [11] Coello C., Van Veldhuizen D., Lamont G., *Evolutionary algorithms for solving multi-objective problems*. 2nd edition, Genetic and evolutionary computation, Springer Verlag 2007.

¹ Przyczynek do tego typu pogłębionej analizy jak też szerszą dyskusję aspektów realizacyjnych platformy NEvol zawarto w raporcie wewnętrznym Katedry Informatyki nr 10/2007 pt. *Mechanizmy i narzędzia agentowo-ewolucyjnej optymalizacji wielokryterialnej*, przygotowanym przez zespół w składzie: E. Łopacz, M. Arkit, G. Domagała, B. Piec, P. Powroźnik, pod kierunkiem mgr. inż. L. Siwika oraz dr. inż. M. Kisiel-Dorohinickiego.

- [12] Siwik L. Natanek S., *Elitist evolutionary multi-agent system in solving noisy multiobjective optimization problems*. [w:] Proceedings of 2008 IEEE World Congress on computational intelligence, WCCI 2008, Hong Kong 2008, 3318–3325.
- [13] Cetnarowicz K., Kisiel-Dorohinicki M., Nawarecki E., *The application of evolution process in multi-agent world to the prediction system*. [w:] Proc. of the 2nd Int. Conf. on Multi-Agent Systems (ICMAS'96).
- [14] Dobrowolski G., Kisiel-Dorohinicki M., Nawarecki E., *Evolutionary multi-agent system for multiobjective balancing of production lines*. [w:] IFAC'02: preprints of the 15th triennial world congress of the International Federation of Automatic Control, Elsevier Science, Barcelona, 2002.
- [15] Dreżewski R., Siwik L., *Co-evolutionary multi-agent system for portfolio optimization*, [w:] A. Brabazon, M. O'Neil, Natural Computations in Computational Finance, Springer-Verlag, 2008, 273–303.
- [16] Dreżewski R., Siwik L., *Agent-based co-evolutionary techniques for solving multi-objective optimization problems*. [w:] Advances in evolutionary algorithms, W. Kosinski (Ed.), Vienna, IN-TECH, 2008, 231–260.
- [17] Kluska-Nawarecka S., Polcik H., Nawarecki E., Dobrowolski G., *Virtual teams and computational intelligence in foundry and metallurgical processes*. [w:] Virtual design and automation, Poznań 2005, 177–185.
- [18] Koźlak J., Dobrowolski G., Kisiel-Dorohinicki M., Nawarecki E., *Agent-based modeling of city traffic for crisis management in transportation systems*. [w:] Proceedings of the 16th International Conference on Systems Science, Wrocław 2007, 421–430.