

Piotr Kadłuczka*, Wojciech Chmiel*, Jacek Piwowarczyk*

Strategia przetwarzania wiedzy w wielopopulacyjnym algorytmie ewolucyjnym

1. Wprowadzenie

W ramach prowadzonych badań nad efektywnością wielopopulacyjnych, równoległych algorytmów ewolucyjnych, powstała koncepcja algorytmu samoadaptacyjnego [1], którego inspiracją są inteligentne systemy wieloagentowe (*IMAS*). Należy podkreślić, że opis proponowanego systemu może być zrealizowany także w oparciu o pojęcia związane z algorytmami ewolucyjnymi (*AE*), programowaniem współbieżnym oraz systemami przetwarzania wiedzy. Przedstawienie systemu w obszarze pojęć inteligentnych systemów agentowych pozwala na prosty opis wielu możliwych strategii przeszukiwania i przetwarzania wiedzy realizowanych za pomocą algorytmów ewolucyjnych.

W konstrukcji *AE*, problemem jest mnogość możliwych koncepcji każdego elementu algorytmu. Należy określić: postać rozwiązania (reprezentacja, sposób kodowania) i funkcji oceny przystosowania, sposób przetwarzania populacji (generacja populacji startowej, pokolenie, selekcja, elita), typy i mechanizm użycia operatorów genetycznych (krzyżowanie, mutacji, dodatkowych np. optymalizacji lokalnej), kryteria zakończenia obliczeń, restartu, odświeżania populacji oraz wiele innych. Mechanizmy te opisane są wieloma parametrami (np. wielkość populacji, elity, prawdopodobieństwa użycia operatorów), których prawidłowy dobór wartości powinien być przeprowadzony dla reprezentatywnego zbioru zadań testowych (indywidualnie dla każdego zadania lub wszystkich razem). Wymagane jest też wielokrotne powtórzenie każdego eksperymentu, gdyż jedynie statystyczna ocena efektywności jest miarodajna dla metod losowych, do których należą EA.

Przedstawiona metodyka prowadzenia badań skutkuje dużym nakładem pracy nawet przy wprowadzeniu niewielkich zmian w algorytmie, ze względu na występującą „interferencję” wszystkich jego elementów. Próby stworzenia systemów realizujących samoadaptacyjne mechanizmy doboru elementów konstrukcyjnych, struktury i parametrów algorytmu były podejmowane już wcześniej (M. L. Cramer 1985 r. programowanie genetyczne – *Genetic Programming, GP*), chociaż wielokrotnie powracano do tej koncepcji (J. Koza, 1992), sła-

* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

bość teorii oraz ograniczenia wynikające z mocy obliczeniowej maszyn cyfrowych nie pozwoliły w pełni jej zrealizować.

Podstawową trudnością realizacji szeroko zakrojonej koncepcji *GP* jest znaczne rozszerzenie eksplorowanej przestrzeni. Jeśli rozważany problem jest NP-trudny (gdyż to on wymusza użycie metod przybliżonych), to licznosc przeszukiwanej przestrzeni rozwiązań jest dodatkowo multiplikowana przez liczbę możliwych do rozważenia elementów konstrukcyjnych (wymienne lub łącznie) w każdym miejscu algorytmu oraz liczbę wartości przyjmowanych przez każdy parametr (zależną od zakresu i dyskretyzacji).

2. Współbieżny, wielopopulacyjny algorytm ewolucyjny jako system wieloagentowy

Realizacja wielopopulacyjnego *AE* w większości implementacji sprowadza się do niezależnego przetwarzania zadanej liczby populacji (lub podpopulacji) i okresowej wymianie (migracji) rozwiązań (osobników) między populacjami. Podejście to w sposób naturalny doskonale nadaje się realizacji współbieżnej. Proponowane przez nas rozszerzenie podstawowej koncepcji wielopopulacyjnego *AE* łączy w sobie następujące cechy: autonomia wątków realizujących wyodrębnione *AE*, elastyczność, ukierunkowanie na osiąganie pewnych celów, umiejętność dostosowania, uczenia się (zdobywania wiedzy, wnioskowania) i porozumiewania się, a dzięki temu współdziałania.

Powyższe cechy wymieniane są jako charakterystyczne dla systemów agentowych, stąd uzasadnione jest przedstawienie algorytmu jako *MAS*.

2.1. Agent programowy bazujący na algorytmie ewolucyjnym

Przytaczając za [2] definicję, agentem Λ nazywamy trójkę:

$$\Lambda = (A, S, F \subset S \times A \times S) \quad (1)$$

gdzie:

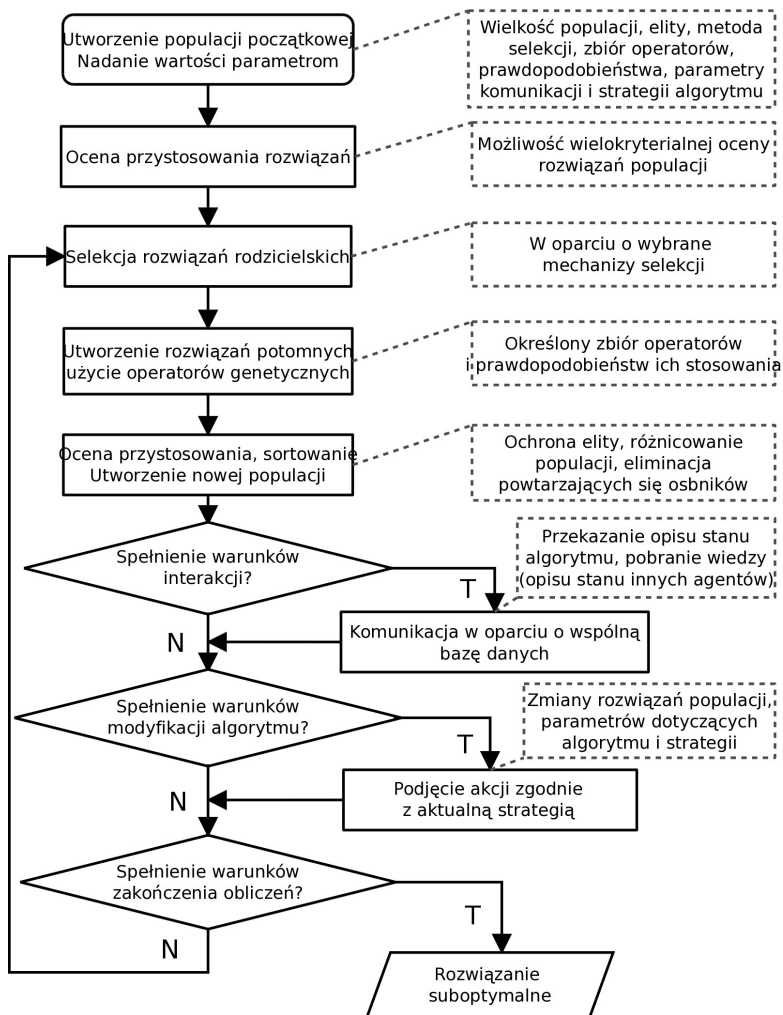
- A – zbiór akcji / decyzji (skończony),
- S – skończony zbiór stanów agenta,
- F – funkcja przejścia pomiędzy stanami.

Relacja F określa przejście pomiędzy stanami w wyniku wykonania akcji. Zachowanie związków przyczynowo-skutkowych (jednoznaczność) wymaga, aby prawdziwa była następująca implikacja: $(s, a, s_1) \in F \wedge (s, a, s_2) \in F \Rightarrow s_1 = s_2$.

Relacja F w sposób jednoznaczny, wiąże stany z akcjami, określając, do jakiego stanu prowadzi podjęcie danej decyzji w danym stanie, oraz wyznacza zbiór stanów osiągalnych ze stanu aktualnego.

Koncepcja inteligentnego agenta programowego bazuje na *AE*. Agentem jest algorytm (jeden z wielu możliwych), który wykorzystując wybrane przez siebie elementy konstruk-

cyjne i dobierając wartości parametrów, stara się poprawić swoją efektywność. Stosowane przez nas podstawowe pojęcia dotyczące opisu *AE* i jego komponentów można znaleźć w pracy [3] oraz [4]. Jako podstawowy schemat algorytmu (rys. 1) przyjęto schemat z zastępowaniem pokoleń (*generational replacement*), w którym przejście do następnego pokolenia powoduje automatyczne zastąpienie rozwiązań – rodziców przez ich potomków.



Rys. 1. Uproszczony schemat algorytmu realizowanego przez agenta

Dla rozpatrywanego zagadnienia testowego (komiwojażera – *TSP* – *traveling salesman problem*), jako typowego zagadnienia permutacyjnego przyjęto kodowanie całkowitoliczbowe (naturalne), ze względu na prostotę, i łatwiejsze, niż w przypadku innego kodowania

utrzymanie pewnych własności strukturalnych rozwiązań. Jako funkcję oceny przystosowania przyjęto funkcję celu dla *TSP*.

W celu uniknięcia problemu ze zbieżnością klasycznego podejścia zastępowania pokoleń, zastosowano w populacji elitę, tzn. część najlepszych rozwiązań populacji przepisuje się do następnej generacji, pozwalając konkurować rodzicom z ich potomstwem.

Faza selekcji jest realizowana w zasadzie przez dwa algorytmy: selekcji oraz wyboru (*sampling*). Pierwszy przyporządkowuje każdemu osobnikowi rzeczywistą liczbę, będącą wartością oczekiwaną liczby potomstwa. Następnie losowane są z prawdopodobieństwem selekcji rozwiązania, w ilości równej liczbie osobników, które mają zostać wymienione w populacji. W algorytmie wykorzystano różne strategii selekcji: selekcję proporcjonalną do przystosowania, bazującą na rankingu oraz selekcję turniejową.

Operatory genetyczne specjalizowane dla zagadnienia *TSP* sprowadzają się do znanych dla zagadnień permutacyjnych operatorów krzyżowania: *PMX*, *OX*, *CX*, dwóch operatorów mutacji oraz dodatkowo operatora optymalizacji lokalnej, poprawiającego zbieżność algorytmu.

Istotnymi parametrami algorytmu ewolucyjnego są wielkość populacji, wielkość elity, prawdopodobieństwa stosowania operatorów, całkowita liczba iteracji algorytmu.

Zbiorem akcji agenta są wszelkie działania dotyczące zmiany struktury algorytmu ewolucyjnego, wartości parametrów i strategii działania.

Przykładami akcji są:

- realizacja iteracji *AE* (np. na zbiorze aktualnych / częściowo / w całości zmienionych rozwiązań),
- dodanie lub usunięcie komponentu algorytmu (np. mechanizm selekcji, operator, elita),
- zmiana intensywności użycia komponentu algorytmu (np. krotność, prawdopodobieństwo),
- zmiana wartości parametrów (np. wielkość populacji, elity),
- zmiana konstrukcji komponentu (np. sposobu traktowania elity, powtórzenia osobników),
- zmiana strategii współdziałania agenta (np. komunikacja, interpretacja wiedzy).

Stan agenta jest określony przez aktualny zbiór rozwiązań optymalizowanego problemu, aktualnie realizowany algorytm (jego elementy i wartości parametrów) oraz przyjętą strategię współdziałania. Strategia współdziałania agenta definiuje warunki, częstość i sposób interakcji. Jest ona definiowana przez kilka elementów, których współwystępowanie w różnych konfiguracjach tworzy liczną przestrzeń możliwych strategii.

Funkcja przejścia określa wszystkie możliwe przejścia między stanami (w wyniku wykonania akcji) dla agenta realizującego aktualną strategię, czyli definiuje zbiór stanów osiągalnych z aktualnego stanu agenta.

2.2. System wieloagentowy

Systemem wieloagentowym (*MAS – multi-agent system*) w ogólnym pojęciu określamy wszelkie rodzaje systemów złożonych z wielu autonomicznych komponentów (encji). Systemem agentowym Ω nazywany następującą dwójką [2]:

$$\Omega = \left\{ \left[\left\{ \Lambda^i \right\}_{i=1, \dots, N}, I \right] \right\} \quad (2)$$

którą stanowi skończony zbiór agentów systemu $\left\{ \Lambda^i \right\}_{i=1, \dots, N}$ oraz relacja zwana relacją współdziałania o postaci

$$I = \bigcup_{\substack{i, j=1, \dots, N \\ i \neq j}} I^{ij},$$

przy czym: $A^i \times A^j \supset I^{ij} \ni (a^i, a^j) : A^i \in \Lambda^i; A^j \in \Lambda^j; A^i, A^j \in \Omega$.

Relacja I jest symetryczna $(a^i, a^j) \in I \Rightarrow (a^j, a^i) \in I$ i określa potencjalne możliwości współdziałania agentów w obrębie systemu. Współdziałanie następuje, gdy agenci wykonują akcje będące w relacji I . Oczywiście obie akcje nie mogą zostać wykonane niezależnie.

2.3. Strategia przetwarzania wiedzy i komunikacja w systemie agentowym

Strategia ta nierozzerwalnie wiąże się z interakcją agentów. Typowymi wzorcami interakcji w *MAS* są kooperacja (osiąganie wspólnego celu), koordynacja (organizacja współpracy) oraz negocjacja (osiąganie kompromisu).

Interakcja agentów, do której dochodzi za pośrednictwem bazy danych, realizuje głównie dwa pierwsze wzorce. Dokonywana jest ona w oparciu o implementację protokołu komunikacyjnego Γ , będącą różnowartościową funkcją [2]:

$$I_{\Gamma} : O \in \Gamma \rightarrow \left\{ \left[\left\{ \Lambda^i \right\}_{i=1, \dots, N} \right] \right\} \in \Omega \quad (3)$$

gdzie O jest zbiorem (skończonym) stron w procesie komunikacji.

Komunikaty implementowanego protokołu przekazywane są za pomocą akcji σ_{\bullet} – nadania i ρ_{\bullet} – odbioru komunikatu odpowiednich agentów wchodzących w interakcję. Dla dowolnych $o_1, o_2 \in O$ oraz agentów Λ^1, Λ^2 i komunikatu m zachodzi:

$$(o_1, m, \bullet), (o_2, \bullet, m) \in P \Rightarrow (o_1, \Lambda^1), (o_2, \Lambda^2) \in I_{\Gamma} \wedge (\Lambda^1, \Lambda^2) \in C \quad (4)$$

gdzie C i P są odpowiednio relacją komunikacji w *MAS* i relacją określającą możliwy przebieg procesu komunikacji.

Podjęcie interakcji jest wewnętrzną akcją agenta i może być realizowane:

- cyklicznie – co zadaną liczbę iteracji,
- asynchronicznie – w przypadku wystąpienia określonego zdarzenia – np. poprawa najlepszego rozwiązania, pogorszenie średniej wartości funkcji celu populacji o zadaną wartość progową lub w wyniku wnioskowania na podstawie zdobytej wiedzy.

Komunikacja może być realizowana w oparciu o wiele protokołów komunikacyjnych. Komunikat zawiera fragmentaryczny opis stanu agenta (zewnątrzny opis stanu) co ma celu ograniczenie wielkości strumienia komunikacji przy uwzględnieniu:

- kierunku przepływu informacji,
- przekazywania jedynie istotnej wiedzy,
- ziarnistości informacji (wiedza szczegółowa, uogólniona, ogólna).

W bazie danych, za pośrednictwem której komunikują się agenci, zapisany jest ich aktualny stan. Opis stanu zawiera jedynie informacje uznane przez agenta za istotne i przekazane zgodnie z formatem komunikatów przy ustalonym protokole komunikacji.

Analiza wiedzy dokonywana jest przez agenta na zasadzie badania efektywności stosowanego algorytmu i porównywaniu skutków wprowadzonych w nim zmian w stosunku do wcześniejszej wersji oraz osiągnięć innych agentów. Wnioskowanie realizowane jest w oparciu o indywidualny, podlegający modyfikacji zbiór reguł, zdefiniowany dla każdego agenta. Różnice efektywności analizowane są pod kątem wyszukania elementów, które różnią się lub są zgodne dla porównywanych algorytmów. Na tej podstawie określa się status zdobywanej wiedzy dotyczącej: konstrukcji algorytmu, przyjętych wartości parametrów *AE* oraz parametrów dotyczących komunikacji, wnioskowania i modyfikacji agenta. Jako status wiedzy wyróżniono następujące sytuacje:

- Parametr stanu został nadany jako wartość początkowa.
- Aktualna wartość parametru została ostatnio zmodyfikowana (wartość bieżąca).
- Wartość parametru uznana jako istotna z punktu widzenia agenta – potwierdzona efektywnością algorytmu – określony kierunek zmian, przedział zmienności, istotna wartość zmiany parametru.
- Wartość parametru potwierdzona przez system (zadaną liczbę agentów).

Jako podstawowe strategie wykorzystania wiedzy przez agenta możemy wyróżnić:

- Strategię powielającą – ustalenie parametrów stanu potwierdzonych przez system.
- Strategię ukierunkowaną – zmiana parametrów ukierunkowana poprawiającą się efektywnością algorytmu.
- Strategię poszukującą – nieukierunkowana zmiana wartości parametrów mająca na celu znalezienie pożądanego kierunku zmian.
- Strategię różnicującą – nadane parametrom wartości różniących się od dotychczas stosowanych przez innych agentów (eksploracja przestrzeni).

Strategia działania agenta opiera się na doborze elementów *AE* wraz z wartościami ich parametrów oraz na pozyskiwaniu wiedzy, poprzez komunikację i sposób wnioskowania, na podstawie której podejmowane są działania. Istotne jest zmniejszenie kosztów komunikacji pomiędzy współbieżnymi wątkami, realizowane poprzez rozproszoną inteligencję i autonomię agentów.

3. Implementacja systemu agentowego

Implementację *MAS* zrealizowano w języku Ada [5], mającym w składni języka konstrukcje współbieżne oraz rozszerzenie w standardzie języka pozwalające na programowanie rozproszone. Przyjęty przyrostowy sposób tworzenia oprogramowania zakłada stworzenie podstawowej struktury systemu *MAS* (zawierającej bazową wersję algorytmu ewolucyjnego), która jest stopniowo wzbogacana o kolejne elementy takie jak: nowe komponenty algorytmu, strategie postępowania nadzorcy i agentów, sposoby wnioskowania. Prace, dzięki wykorzystaniu podejścia wielowątkowego, ukierunkowano na wykorzystania procesorów wielordzeniowych, a w przyszłości systemów rozproszonych. Zaproponowana koncepcja implementacji jest niezależna od realizacji sprzętowej, w tym liczby procesorów jak i docelowego systemu operacyjnego.

W oparciu o przedstawioną koncepcję zaimplementowano wstępną wersję systemu wieloagentowego. Przeprowadzone w oparciu o nią testy są obiecujące i zostaną przedstawione w kolejnych publikacjach dotyczących przedstawionego w artykule zagadnienia.

Literatura

- [1] Kadłuczka P., Piwowarczyk J., Chmiel W., *Współbieżny algorytm ewolucyjny wykorzystujący mechanizm samoadaptacji*. Automatyka (półrocznik AGH), Kraków, 2007, 105–112.
- [2] Dobrowolski G., *Technologie agentowe w zdecentralizowanych systemach informacyjno-decyzyjnych*. UWND AGH, Kraków 2002.
- [3] Goldberg D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, 1989; tłum. *Algorytmy genetyczne i ich zastosowania*. WNT, Warszawa 1995.
- [4] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 1995, tłum. *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. WNT, Warszawa 1996.
- [5] *Ada Reference Manual*, ISO/IEC 8652:2007(E) Ed. 3.
- [6] <http://www.tsp.gatech.edu/data/index.html>.
- [7] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.