

Grzegorz Wojarnik

Uniwersytet Szczeciński

ZŁOŻONOŚĆ OBLICZENIOWA METOD EWOLUCYJNYCH – POSZUKIWANIE ALGORYTMU EWOLUCYJNEGO ROZWIĄZUJĄCEGO PROBLEM BIZNESOWY

Streszczenie: Istnieje wiele sposobów rozwiązywania problemów wielokryterialnych. Jednym z nich są algorytmy ewolucyjne. Ze względu na różnorodność parametrów sterujących tymi algorytmami można wyróżnić ich wiele rodzajów i odmian. Trudno jest stwierdzić *ad-hoc*, która z metod ewolucyjnych jest lepsza do rozwiązania postawionego problemu. Jednym z możliwych rozwiązań, zmierzających do uzyskania odpowiednich parametrów działania algorytmu ewolucyjnego, wydają się metaalgorytmy, które mogą być pomocne w wyborze algorytmu ewolucyjnego, niejako dopasowanego do rozwiązywania problemów danej klasy. Z tego punktu widzenia można zaproponować metaalgorytm, który pozwoli na rozwiązywanie różnorodnych problemów biznesowych przy użyciu algorytmów ewolucyjnych, co będzie przedmiotem dalszych badań autora artykułu w niedalekiej przyszłości.

Słowa kluczowe: metody ewolucyjne, sztuczna inteligencja, optymalizacja.

1. Wstęp

Życie przynosi wyzwania, których komplikacja bardzo często przekracza możliwości klasycznych metod rozwiązania problemów optymalizacyjnych. Czasami jest tak, że nie można wybrać metody dającej zadowalające rozwiązania lub jest ona nieznaną. Zdarzają się w końcu sytuacje, w których znana jest metoda rozwiązania danego problemu, tyle że nie wiadomo, ile czasu będzie potrzeba do wygenerowania zadowalającego rozwiązania, ze względu na bardzo dużą liczbę rozwiązań potencjalnych. Często też zdarza się tak, że sama metoda jest na tyle skomplikowana, że sam wybór jej wariantu, który byłby dostosowany do specyfiki problemu, jest zadaniem nietrywialnym.

Przykładem takich metod są algorytmy genetyczne, które same w sobie stanowią podzbiór metod sztucznej inteligencji charakteryzujących się dużą różnorodnością i zmiennością. Dlatego wydaje się, że warto przyrzeć się algorytmom ewolucyjnym pod kątem ich złożoności, a co za tym idzie – możliwości opanowania ich komplikacji celem takiego wyboru parametrów działania algorytmu genetycznego, aby był on możliwie najlepiej dostosowany do charakteru rozwiązywanego zadania.

2. Różnorodność metod ewolucyjnych

Najbardziej popularnym rodzajem metod ewolucyjnych są **algorytmy genetyczne**. Prekursorem takiego podejścia jest J.H. Holland, który opublikował w 1962 r. pracę *Outline for a Logical Theory of Adaptive Systems*. Holland przedstawił w niej podstawy systemów adaptacyjnych, które potrafią zmieniać się w reakcji z środowiskiem, w którym funkcjonują [De Jong, Fogel, Schwefel 1997, s. A2.3:4].

Działanie algorytmu genetycznego można przedstawić za pomocą następujących kroków [Kinnear, Smith, Michalewicz 1997, s. 147]:

1. Inicjalizacja populacji.
2. Obliczenie wartości funkcji dopasowania każdego osobnika z populacji.
3. Reprodukacja wybranych osobników w celu stworzenia nowej populacji.
4. Przeprowadzenie operacji krzyżowania i mutacji na nowej populacji.
5. Przejście do kroku 2, o ile nie zajdzie warunek kończący przetwarzanie.

Kolejnym przykładem metod ewolucyjnych są **strategie ewolucyjne**, które powstały w 1964 r. na Uniwersytecie Berlińskim jako wynik prac badawczych Bienera, Rechenberga i Schwefela. Z prac tych wyłoniła się pierwsza strategia ewolucyjna znana później pod nazwą (1+1).

Strategia (1+1) polega na perturbacji jedyne go chromosomu bazowego [Arabas 2004, s. 83], której wynikiem jest nowy chromosom. Perturbacja ta jest wynikiem dodania losowej modyfikacji rozkładem normalnym do każdego genu chromosomu bazowego. Następnie są porównywane wartości funkcji przystosowania dla chromosomu bazowego i nowego. Do dalszego przetwarzania jest pobierany ten chromosom, który charakteryzuje się większą wartością funkcji przystosowania. Ze strategią (1+1) jest związana reguła 1/5 sukcesów. W myśl tej reguły jeśli przez k generacji liczba mutacji zakończonych sukcesem jest większa niż 1/5 wszystkich mutacji, to należy zwiększyć zasięg mutacji; jeśli 1/5 mutacji zakończy się sukcesem, zasięg mutacji nie będzie się zmniejszał; jeśli w końcu mniej niż 1/5 zakończy się sukcesem, zasięg mutacji należy zmniejszyć [Arabas 2004, s. 85].

Większość strategii ewolucyjnych działa według następującego algorytmu [Kennedy, Eberhart, Shi 2001, s. 179]:

1. Inicjalizacja populacji.
2. Wykonanie rekombinacji przy użyciu μ rodziców w celu stworzenia λ dzieci.
3. Wykonanie mutacji na wszystkich dzieciach.
4. Ewaluacja λ lub $\mu + \lambda$ osobników populacji.
5. Wybór μ osobników do nowej populacji.
6. Przejście do kroku 2, o ile nie zajdzie warunek kończący przetwarzanie.

Rozwinięciem strategii (1+1) jest ($\mu + \lambda$), w której przetwarzaniu podlega populacja osobników stanowiących rozwiązanie danego problemu. W algorytmie owym wprowadzono również mechanizm krzyżowania. W strategii tej przetwarzana jest populacja bazowa zawierająca μ osobników, której osobniki stanowią połączenie dwóch chromosomów. Pierwszy to wektor wartości zmiennych niezależnych zwią-

zanych z rozwiązywanym problemem, drugi – to wektor zawierający wartości standardowych odchyłeń wykorzystywanych podczas mutacji. Z populacji tej generuje się populację potomną zawierającą λ osobników. Populacja potomna powstaje przez wykonanie operacji krzyżowania i mutacji na populacji bazowej. Nową populację bazową tworzy μ najlepszych osobników spośród populacji stanowiących złączenie populacji bazowej i potomnej.

W algorytmie $(\mu + \lambda)$ istotny jest proces podawania procesom genetycznym chromosomu odpowiedzialnego za przechowywanie wartości standardowych odchyłeń wykorzystywanych podczas mutacji. Mechanizm ten pozwala na automatyczną regulację zasięgu mutacji, tak aby minimalizować wartość oczekiwaną czasu dojścia do maksimum globalnego funkcji przystosowania [Arabas 2004, s. 87].

Algorytm (μ, λ) jest prostą modyfikacją strategii $(\mu + \lambda)$. Modyfikacja ta polega na tym, że nowa populacja bazowa jest wyłoniona z λ osobników potomnych wynikłych z przekształcenia osobników bazowych μ . Można zatem zauważyć, że w strategii tej w skład dalszego przetwarzania nie wchodzi osobniki rodzicielskie stanowiące bazę do przetwarzania za pomocą operatorów genetycznych.

Różnice pomiędzy strategiami ewolucyjnymi a algorytmami genetycznymi [Michalewicz 2003, s. 197-8] są następujące:

- strategie ewolucyjne rozwinięto jako metody optymalizacji numerycznej, gdy tymczasem algorytmy genetyczne stanowią metody adaptacyjnego przeszukiwania (ogólnego przeznaczenia). Oznacza to, że algorytmy genetyczne były używane w różnych zastosowaniach, w tym do optymalizacji numerycznej,
- w strategiach ewolucyjnych kolejna populacja bazowa jest tworzona m.in. przez usunięcie osobników najgorzej dopasowanych, z kolei w przypadku algorytmów genetycznych istnieje pewna szansa, aby do kolejnej populacji trafił nawet naj słabiej przystosowany osobnik,
- w strategiach ewolucyjnych proces selekcji następuje po operatorze rekombinacji, natomiast w algorytmach genetycznych kolejność tych kroków jest odwrotna,
- strategie ewolucyjne używają reprezentacji zmiennoprzecinkowej [Whitley 2001, s. 12], co powoduje, że operator mutacji jest zaimplementowany jako propagacja wartości wokół osobnika, który podlega mutacji,
- parametry reprodukcji (czyli prawdopodobieństwo krzyżowania oraz mutacji) w algorytmach genetycznych są stałe, a w strategiach ewolucyjnych zmienia się je w czasie,
- w przypadku strategii ewolucyjnych, jeśli potomek nie spełnia założonych ograniczeń, jest od razu odrzucany, a więc nie pojawi się w nowej populacji i jeśli stosunek generowania nieprawidłowych potomków jest zbyt duży, zmieniają parametry sterujące. W przypadku algorytmów genetycznych na osobniki naruszające ograniczenia nakładana jest funkcja kary, ponieważ dla problemów z nałożonymi licznymi ograniczeniami istnieje możliwość znacznego ograniczenia przetwarzanej populacji, a nawet całkowitej jej eliminacji, w sytuacji gdy wszystkie osobniki nie spełniałyby nałożonych ograniczeń.

Do metod ewolucyjnych zaliczyć można **programowanie ewolucyjne**, którego podstawowe zasady zostały sformułowane przez L.J. Fogela w 1960 r. Metody sztucznej inteligencji koncentrowały się w tym czasie wokół heurystyk oraz symulacji prostych sieci neuronowych. Było jasne dla Fogela, że obydwie podejścia były ograniczone, ponieważ skupiały się na ujęciu modelu człowieka, a nie na procesie, którego efektem jest powstawanie wszelkich istot, a więc ewolucja. Problem ewolucyjny był przez niego zdefiniowany jako rozwój algorytmu (szczególnie programu komputerowego), który będzie działał na sekwencji symboli tak, aby wyłonić symbol wyjściowy w taki sposób, żeby maksymalizować wydajność algorytmu [De Jong, Fogel, Schwefel 1997, s. A2.2:3].

Celem programowania ewolucyjnego jest odkrywanie gramatyki nieznanego języka, gdy znany jest zestaw symboli i dane są przykłady wyrażen syntaktycznie poprawnych. Gramatyka ta jest modelowana przez automat skończony, którego zbiór stanów, funkcja przejść i funkcja wyjść są przedmiotem poszukiwań.

W algorytmie programowania ewolucyjnego przetwarzana jest populacja bazowa o licznosci μ osobników. W jednym kroku każdy osobnik tworzy jednego potomka, który jest poddawany mutacji. W ten sposób powstaje populacja potomków o licznosci $\lambda = \mu$. Nowa populacja bazowa jest tworzona w ten sposób, że dla każdego osobnika populacji bazowej i potomnej określana jest ranga, a następnie do nowej populacji bazowej są wybierane osobniki, których ranga jest najwyższa.

Działanie algorytmu w programowaniu ewolucyjnym jest następujące [Kennedy, Eberhart, Shi 2001, s. 165]:

1. Inicjalizacja populacji.
2. Kalkulacja funkcji dopasowania dla każdego osobnika populacji.
3. Losowa mutacja każdego rodzica członka populacji.
4. Porównanie rodziców i dzieci.
5. Wybór członków nowej populacji.
6. Przejście do kroku 2, o ile nie zajdzie warunek kończący przetwarzanie.

Do najważniejszych różnic pomiędzy programowaniem ewolucyjnym a strategiami ewolucyjnymi można zaliczyć następujące aspekty [Michalewicz 2003, s. 201]:

- w programowaniu ewolucyjnym nie używa się operatorów rekombinacji,
- w programowaniu ewolucyjnym stosuje się selekcję turniejową, podczas gdy w strategiach ewolucyjnych wybiera się najlepsze osobniki do następnego pokolenia,
- w programowaniu ewolucyjnym wartości dopasowania uzyskuje się z funkcji celu przez ich skalowanie z ewentualnym wymuszaniem losowych zmian,
- odchylenie standardowe mutacji dla każdego osobnika oblicza się jako pierwiastek kwadratowy z liniowego przekształcenia jego własnej wartości dopasowania.

W literaturze można znaleźć opis takich zastosowań programowania ewolucyjnego, jak modelowanie i identyfikacja systemów, sterowanie, prognozowanie, uczenie sieci neuronowych, rozpoznawanie obrazów, problemy kombinatoryczne.

Kolejnym przykładem metod ewolucyjnych jest **programowanie genetyczne**, które powstało jako gałąź algorytmów genetycznych o silnie rozwiniętych strukturach danych. Jej twórcą jest J.R. Koza, który w 1989 r. opracował podstawowe zasady rządzące tym rodzajem algorytmów ewolucyjnych [Kinnear, Smith, Michalewicz 1997, s. B1.5:1]. Programowanie genetyczne można uznać za formę automatycznego programowania, a nie metodę optymalizacyjną.

Pierwotnie jako bazowego języka użyto LISP, w którym program jest przedstawiony w postaci drzewa – analogicznie jak dane. Dzięki temu w odróżnieniu od algorytmów genetycznych kodowanie binarne w przypadku programów genetycznych jest zastąpione drzewiastym. W związku z tym opracowano sposoby kodowania, które pozwalają operatorom genetycznym na modyfikację zarówno węzłów drzewa, jak i jego struktury. W programowaniu genetycznym operatory działające na strukturze chromosomu mogą zmieniać jego strukturę, powodując rozrastanie, zmniejszanie lub zmianę kształtu chromosomu.

W programowaniu genetycznym występują następujące kroki [Kennedy, Eberhart, Shi 2001, s. 180]:

1. Specyfikacja zestawu operatorów.
2. Specyfikacja zestawu funkcji.
3. Specyfikacja miary dopasowania.
4. Wybór parametrów kontroli systemu.
5. Specyfikacja warunków zakończenia.

Programowanie genetyczne można rozumieć jako metodykę, której celem jest ewolucja populacji programów, których zadaniem jest rozwiązanie określonego problemu w taki sposób, aby dzięki procesom ewolucyjnym został wyłoniony program najlepiej rozwiązujący dany problem.

Do najważniejszych różnic pomiędzy programowaniem genetycznym a algorytmami genetycznymi należą [Koza 1997, s. 637]:

- struktura adaptowanego osobnika, w przypadku której programowanie genetyczne używa populacji składającej się z hierarchicznych kompozycji funkcji ze zbioru funkcji i operatorów ze zbioru operatorów, natomiast w przypadku algorytmu genetycznego jest to ciąg znaków (w danym alfabecie) o stałej długości,
- operatory modyfikujące strukturę w przypadku programowania genetycznego, tj. reprodukcja i krzyżowanie, w przypadku algorytmów genetycznych dodatkowo występuje jeszcze mutacja,
- obydwa rodzaje metod ewolucyjnych różnią się parametrami kontrolnymi działania operatorów.

Programowanie genetyczne znajduje swoje zastosowanie w informatyce, nauce, inżynierii, sztuce i rozrywce [Banzhaf 2001, s. 791]. Dzięki możliwości manipulacji na strukturach symboli programowania genetyczne są heurystyczną metodą konstrukcji algorytmów. Do tych zaliczyć można algorytmy sortowania, cache'owania danych, generatory liczb losowych, algorytmy automatyzujące równoległe wykonywanie kodu. Typowe wykorzystanie programowania genetycznego w nauce to mo-

delowanie i odkrywanie wzorców. W inżynierii jest używane w konkurencji lub kooperacji z innymi metodami heurystycznymi, takimi jak sieci neuronowe oraz systemy rozmyte. Podstawowy cel w inżynierii to również modelowanie procesów, takich jak plany produkcji czy też klasyfikacja wyników procesów produkcyjnych. W sztuce i rozrywce programowanie genetyczne jest używane do generowania realistycznych animacji czy też grafik.

Pewnym rozwinięciem wyżej opisanych metod jest idea algorytmów koewolucyjnych, którą przedstawili Pollack i Blair w 1996 r. w pracy *Coevolution of a Backgammon Player*. Metoda ta charakteryzuje się lokalnym oddziaływaniem operatorów selekcji i krzyżowania w odróżnieniu od standardowych algorytmów ewolucyjnych, w których oddziaływania te charakteryzują się globalnością zasięgu.

Wyróżnia się dwa podstawowe rodzaje algorytmów koewolucyjnych:

- Algorytmy wyspowe, w których występuje wiele populacji ewoluujących niezależnie od siebie. W algorytmach takich wymiana osobników pomiędzy tymi populacjami następuje sporadycznie. Dzięki takiemu podejściu na poziomie pojedynczej populacji dochodzi do jej zasilania przez osobniki z zewnątrz przy jednoczesnej możliwości jej opuszczenia przez wybrane osobniki.
- Algorytmy komórkowe (masowo równoległe, dyfuzyjne), w których dla każdego osobnika jest zdefiniowane pewne sąsiedztwo i osobnik taki podlega niezależnej ewolucji. Ewolucja polega na tym, że osobnik taki poddawany jest interakcji przez operatory selekcji, mutacji oraz krzyżowania z osobnikami znajdującymi się w jego sąsiedztwie.

Jak zostało zaprezentowane, można stwierdzić, że wielość podejść do obliczeń ewolucyjnych stanowi nie lada wyzwanie dla tych wszystkich, którzy chcieliby wybrać do rozwiązania określonych problemów określoną metodę. Dodatkowo należy pamiętać, że metoda taka, aby była użyteczna, musi być odpowiednio skonfigurowana, żeby można było mówić o jej dostosowaniu do danej dziedziny problemów.

3. Metody dostosowania operatorów algorytmów ewolucyjnych do charakteru rozwiązywanych zadań

W doborze odpowiedniej metody ewolucyjnej do rozwiązania postawionych problemów pomocne mogą się okazać same algorytmy ewolucyjnej. Rozwinęły się dwa główne nurty wspierające dostosowanie operatorów algorytmów ewolucyjnych do rodzaju rozwiązywanego zadania.

Pierwszym z nich są **algorytmy samoadaptujące**, które czerpią z koncepcji przyświecających działaniu strategii ewolucyjnych, z tym że w odróżnieniu od nich w chromosom włączony jest nie tylko operator mutacji, ale również inne. Dzięki definicji genów powiązanych z poszczególnymi operatorami, algorytm samoadaptujący pozwala nie tylko na dobór osobników najlepiej przystosowanych, w kontekście rozwiązywanego problemu, ale również wybór sposobu działania takiego algorytmu, przez dobranie takich parametrów pracy operatorów ewolucyjnych, które na

danym etapie działania algorytmu będą skutkowały najszybszymi zmianami osobników w kierunku rozwiązań najbardziej bliskich funkcji celu [Meyer-Nieberg, Beyer 2007, s. 47-75].

Drugim sposobem są **metaalgorytmy ewolucyjne**, które wykorzystują ideę sterowania parametrami wykonania algorytmu ewolucyjnego. Pomysł na takie działanie wywodzi się z konstatacji, iż liczba różnych rodzajów poszczególnych operatorów oraz parametrów decydujących o działaniu algorytmu ewolucyjnego jest na tyle duża, że samo poznanie postaci tych parametrów najlepiej przystających i odpowiednich do danej klasy rozwiązywanych problemów jest na tyle trudne, że pomocny przy określaniu tych parametrów może być algorytm zwany metaalgorytmem ewolucyjnym [Bäck 2001]. Standardowym sposobem dobierania parametrów sterujących działaniem algorytmu ewolucyjnego jest po pierwsze wybór jego rodzaju (np. spośród omówionych w niniejszej opracowaniu) oraz, po tym wyborze, dobór odpowiednich operatorów w ramach wybranego już rodzaju. Możliwych operatorów krzyżowania można wymienić np. co najmniej 340 [Gwiazda 2007a], a operatorów mutacji 130 [Gwiazda 2007b].

W odróżnieniu od samoadaptujących algorytmów ewolucyjnych zadaniem metaalgorytmu ewolucyjnego jest wybór osobnika najlepiej przystosowanego do funkcji celu przez wielokrotne wykonanie procesu ewolucyjnego dla różnych konfiguracji parametrów sterujących poszczególnym procesem ewolucyjnym.

4. Koncepcja metodyki wykorzystania metaalgorytmu do poszukiwania algorytmu ewolucyjnego umożliwiającego generowanie zadowalających rozwiązań problemów biznesowych

Należy zwrócić uwagę, iż zarówno w przypadku algorytmów ewolucyjnych samoadaptujących, jak i w przypadku metaalgorytmu ewolucyjnego problemem jest efektywność tych rozwiązań. W pierwszym przypadku znacznemu zwiększeniu ulega chromosom przez wzbogacenie go o geny odpowiedzialne za możliwość zmian operatorów genetycznych, natomiast w drugim przypadku działanie pojedynczego algorytmu ewolucyjnego zostaje zwielokrotnione ze względu na konieczność uruchomienia procesu ewolucyjnego z różnymi ustawieniami operatorów ewolucyjnych.

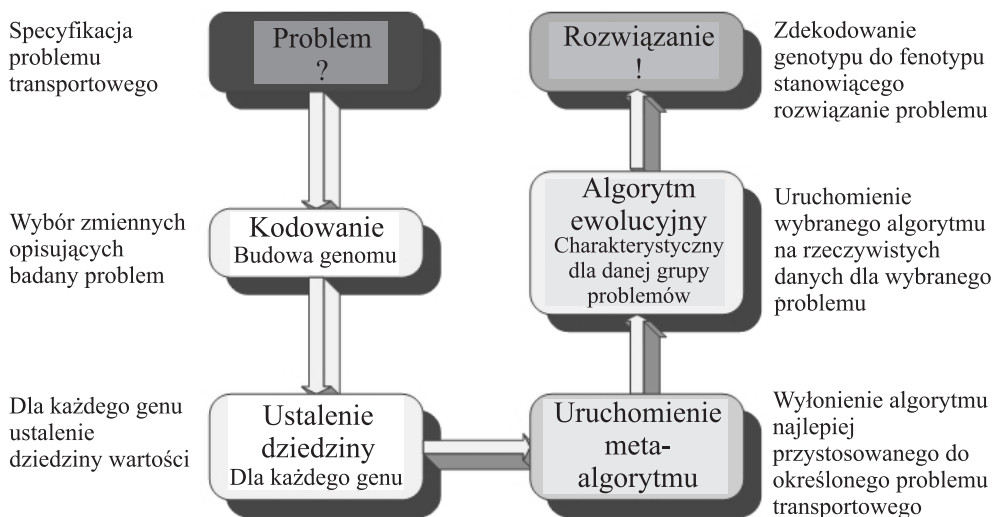
Na podstawie przedstawionego wcześniej opisu można stwierdzić, że metody ewolucyjne pozwalają na znalezienie rozwiązania (rozwiązań) problemów, które charakteryzuje się:

- możliwością zakodowania problemu do postaci akceptowalnej dla działającego algorytmu ewolucyjnego,
- istnieniem funkcji dopasowania, która pozwala zmierzyć jakość rozwiązania,
- populacją rozwiązań problemu zawierającą taką liczbę możliwych rozwiązań, która nie pozwala na efektywne obliczenie funkcji dopasowania dla poszczególnego osobnika tej populacji.

Zadaniem algorytmu ewolucyjnego jest wyłonienie rozwiązań możliwie najbliższych rozwiązaniu optymalnemu. Istnieje też wiele różnych rodzajów algorytmów ewolucyjnych (algorytmy genetyczne, strategie ewolucyjne, programowanie ewolucyjne, programowanie genetyczne). W ramach algorytmu ewolucyjnego można wykorzystać wiele operatorów genetycznych (np. krzyżowanie, mutację, selekcję), które z kolei mogą charakteryzować się wieloma parametrami działania. Podsumowując, można dostrzec lukę w obecnym stanie wiedzy, którą jest efektywny wybór odpowiedniego algorytmu i jego konfiguracja do danej dziedziny problemów. W związku z tym pojawiła się propozycja metodyki, która pozwoli na wyłonienie algorytmu ewolucyjnego dostosowanego do zadanego problemu. Jej ogólny zarys jest zobrazowany na rys. 2.

Oczywiste jest, że dostępne są metody doboru parametrów działania algorytmu ewolucyjnego (metaalgorytmy, algorytmy samoadaptacyjne), jednak ich mankamentem jest wydajność, dlatego powstał pomysł na przeprowadzenie badań zmierzających do określenia możliwości metaalgorytmów ewolucyjnych w wyborze algorytmu ewolucyjnego najlepiej przystosowanego do rozwiązywania problemów o zadanym genotypie (kodującym dany rodzaj problemów). Stąd koncepcja procedury, która z wykorzystaniem metod ewolucyjnych będzie sprzyjała ustrukturalizowanemu podejściu do rozwiązywania różnorodnych problemów życia gospodarczego dla danej grupy zastosowań.

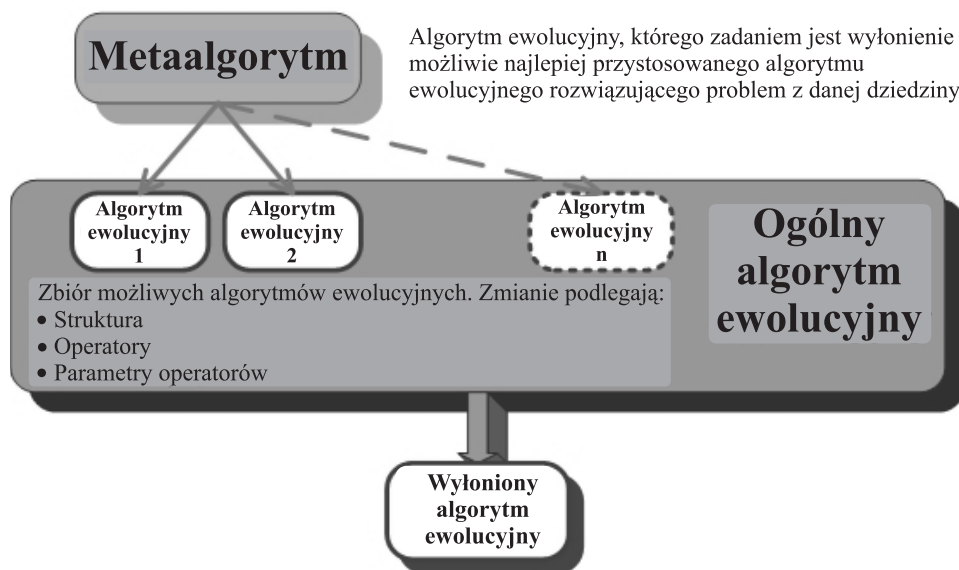
Kluczowym elementem proponowanej procedury jest wykorzystanie metaalgorytmu. Użycie metaalgorytmu sprowadza się do wyszukania dla danej klasy proble-



Rys. 1. Propozycja procedury wykorzystania metaalgorytmu do rozwiązywania problemów w danej grupie zastosowań

Źródło: opracowanie własne.

mów algorytmu o tak dobranej strukturze oraz parametrach operatorów, które zapewniają rozwiązanie najlepsze z punktu widzenia ustalonej funkcji celu spośród wyłonionych algorytmów (na rys. 2 algorytmy te są ponumerowane 1, 2, ... n). Podejście takie daje zysk w postaci czasu potrzebnego na uzyskanie algorytmu przystosowanego do danej klasy problemu. Różnica w stosunku do klasycznego podejścia wykorzystania algorytmów ewolucyjnych wyszukiwanych dzięki metaalgorytmowi opiera się na tym, że w proponowanym podejściu dla danej klasy problemów wykonuje się tylko jeden pełny przebieg w poszukiwaniu algorytmu ewolucyjnego na bazie algorytmu ogólnego. Rozwiązanie kolejnych problemów odbywa się przez uruchomienie wyłonionego algorytmu, którego parametry działania wynikają z operatorów, ich parametrów oraz struktury ogólnego algorytmu ewolucyjnego, którego model zostanie zaproponowany w trakcie dalszych badań (rys. 2).



Rys. 2. Wykorzystanie metaalgorytmu do wyszukania algorytmu ewolucyjnego

Źródło: opracowanie własne.

Jak widać na podstawie przeprowadzonego przeglądu i analizy dostępnych możliwości w dziedzinie algorytmów ewolucyjnych, ich bogactwo wpływa na szanse w rozwiązywaniu licznych zadań, które wynikają z dużej złożoności rzeczywistości. Niniejsza praca stanowi propozycję punktu wyjścia do dalszych dociekań, których celem jest określenie sposobów wykorzystania odpowiednich metod do różnych problemów, które czekają na swoje rozwiązanie.

Proponowana procedura stanowi punkt wyjścia do planowanych przez autora badań. W ich ramach zostanie przede wszystkim zaproponowany metaalgorytm oraz zdefiniowany model ogólnego algorytmu ewolucyjnego, który będzie podlegał

zmianom sterowanym przez ten metaalgorytm. Istnieje duża nadzieja na to, że zostanie pozytywnie zweryfikowana hipoteza, iż dla danej klasy problemów można wyłonić algorytm ewolucyjny, który będzie lepiej przystosowany od innych algorytmów ewolucyjnych obdarzonych innymi parametrami, operatorami oraz strukturą.

Literatura

- Arabas J., *Wykłady z algorytmów ewolucyjnych*, WNT, Warszawa 2004.
- Bäck Th., *Evolutionary Computation, Overview and a CFD Application*, [w:] J. Periaux, P. Joly, O. Pironneau, E. Onate (red.), *Innovative Tools for Scientific Computation in Aeronautical Engineering*, CIMNE, Barcelona 2001.
- Banzhaf W., *Artificial Intelligence: Genetic Programming, International Encyclopedia of the Social & Behavioral Sciences*, Elsevier Science Ltd., 2001.
- De Jong K., Fogel D.B., Schwefel H.P., *A History of Evolutionary Computation w Handbook of Evolutionary Computation*, Oxford University Press, Oxford 1997.
- Gwiazda T.D., *Algorytmy genetyczne. Kompendium. Operator krzyżowania dla problemów numerycznych*, Wydawnictwo Naukowe PWN, Warszawa 2007a.
- Gwiazda T.D., *Algorytmy genetyczne. Kompendium. Operator mutacji dla problemów numerycznych*, Wydawnictwo Naukowe PWN, Warszawa 2007b.
- Kennedy J., Eberhart R.C., Shi Y., *Swarm Intelligence*, Morgan Kaufman Publishers, San Francisco 2001.
- Kinnear K.E., Jr, Smith R.E., Michalewicz Z., *Derivative Methods w Handbook of Evolutionary Computation*, Oxford University Press, Oxford 1997.
- Koza J.R., *Future Work and Practical Applications of Genetic Programming*, [w:] *Handbook of Evolutionary Computation*, Oxford University Press, Oxford 1997.
- Meyer-Nieberg S., Beyer H.G., *Self-Adaptation in Evolutionary Algorithms*, [w:] F. Lobo, C. Lima, Z. Michalewicz, (red.), *Parameter Setting in Evolutionary Algorithm*, Springer, Berlin 2007.
- Michalewicz Z., *Algorytmy genetyczne + struktury danych = programy genetyczne*, WNT, Warszawa 2003.
- Whitley D., *An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls*, Computer Science Department, Colorado University, 2001.

COMPUTATIONAL COMPLEXITY OF EVOLUTIONARY METHODS – BUSINESS PROBLEM SOLUTION SEARCHING THANKS TO EVOLUTIONARY ALGORITHMS

Summary: There are a lot of approaches to solving multi-objective problems. One of them are evolutionary algorithms. There are many different kinds of them. It is difficult to say which evolutionary method is better to solve a given problem. A possible solution to get proper properties of evolutionary method seems to be meta-algorithm that will help to select evolutionary algorithm matching a given problem. A proposed procedure of meta-algorithm which controls properties of evolutionary algorithm used in solving business problems, that will be researched by the author of the article in near future, starts from this point of view.