Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 47

Politechnika Gdańska

Wydział Elektrotechniki i Automatyki

Gdańsk 2015

OPTYMALIZACJA NIECIĄGŁYCH FUNKCJI WIELOMODALNYCH Z WYKORZYSTANIEM KOOPERACYJNEGO ALGORYTMU KOEWOLUCYJNEGO

Łukasz KUCZKOWSKI[1], **Roman ŚMIERZCHALSKI**[2]

1. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki
tel.: 58 347 21-24 e-mail: lukasz.kuczkowski@pg.gda.pl
2. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki
tel.: 58 348 63-27 e-mail: roman.smierzchalski@pg.gda.pl

Streszczenie: W artykule przedstawiono algorytmy koewolucyjne, heurystyczną metodę rozwiązywania złożonych obliczeniowo problemów opartą na zasadzie korelacji oraz darwinowskiej teorii ewolucji. Opisano zalety algorytmu, możliwe zastosowania, sposób działania oraz niektóre z dotychczasowych implementacji. Następnie wybrano trzy wielomodalne lub nieciągłe funkcje testowe: Rosenbrocka, Styblinskiego-Tanga oraz Schaffer'a. Dokonano dekompozycji problemu wyznaczenia minimum globalnego funkcji i przeprowadzono optymalizację wykorzystując kooperacyjny algorytm koewolucyjny. Uzyskane wyniki pozwoliły na ocenę jakości działania algorytmu. Przeprowadzone testy i ich rezultaty są wstępem do szerszych badań nad algorytmami koewolucyjnymi.

Słowa kluczowe: algorytmy ewolucyjne, koewolucja, optymalizacja.

1. WSTĘP

Algorytmy ewolucyjne (AE) to heurystyczna metoda rozwiązywania złożonych obliczeniowo problemów, wykorzystująca mechanizmy biologiczne oparte na darwinowskiej teorii ewolucji. Algorytmy ewolucyjne zostały pomyślnie zastosowane przy rozwiązywaniu wielu złożonych problemów, których rozwiązanie jest trudne lub niemożliwe przy zastosowaniu metod analitycznych np. przewidywanie struktury białek [1], projektowanie układów regulacji i sterowania [2], projektowanie układów VLSI [3], wyznaczanie trasy obiektów ruchomych [4].

Działanie algorytmu ewolucyjnego polega na przetwarzaniu zbioru rozwiązań nazwanych populacją. Środowisko definiowane jest na podstawie problemu (funkcja celu, ograniczenia). Każdy osobnik (element populacji) reprezentuje rozwiązanie problemu. Na podstawie funkcji celu do osobnika przypisuje się wartość nazwaną przystosowaniem, która określa jakość reprezentowanego przez osobnika rozwiązania. W momencie startu algorytmu każdemu osobnikowi zostaje przypisana losowa wartość oraz wyliczone jego przystosowanie. Następnie zostają wykonane etapy: reprodukcji, operacji genetycznych, oceny oraz sukcesji. Reprodukcja polega na losowym wprowadzeniu do populacji tymczasowej osobników z populacji bazowej. Możliwe jest wprowadzenie więcej niż jednej kopii danego osobnika. Lepsza wartość przystosowania osobnika zapewnia większe prawdopodobieństwo powielenia jego kopii. Następnie populacja tymczasowa poddawana jest operacjom genetycznym, które dokonują modyfikacji na osobnikach. Powstały w ten sposób zbiór rozwiązań nazywamy populacją potomną, która podlega ocenie. Etap sukcesji polega na stworzeniu nowej populacji bazowej. Powyższe fazy algorytmu są powtarzane w pętli do momentu spełnienia warunku zatrzymania (np. określona liczba generacji).

Głównym celem pracy jest testowanie działania kooperacyjnego algorytmu koewolucyjnego. W tym celu wybrano trzy funkcje testowe na podstawie których przeprowadzono symulacje z uwzględnieniem nieciągłej i wielomodalnej przestrzeni poszukiwań. Na podstawie badań określono jakoś działania algorytmu. Artykuł zorganizowano następująco: w punkcie drugim opisano zasadę działania algorytmu koewolucyjnego z uwzględnieniem zagadnień kooperacji, w trzecim przedstawiono trzy funkcje testowe, w części czwartej zaprezentowano wyniki symulacji oraz opisano otrzymane rezultaty. W części końcowej podsumowano wyniki badań.

2. ALGORYTMY KOEWOLUCYJNE

Algorytm koewolucyjny (AK) jest rodzajem algorytmu ewolucyjnego (lub grupą algorytmów ewolucyjnych), w którym wartość przystosowania osobnika zależy od jego korelacji z innymi osobnikami. Algorytmy koewolucyjne można podzielić na algorytmy kooperacyjne oraz konkurencyjne. W algorytmach kooperacyjnych osobniki są nagradzane, kiedy lepiej współpracują ze sobą, w algorytmach konkurencyjnych promowane jest współzawodnictwo. Algorytmy koewolucyjne dedykowane są do konkretnych problemów i mają przewagę nad algorytmami ewolucyjnymi w przypadku:

- nieograniczonej lub nieskończonej przestrzeni poszukiwań,
- kiedy wymagana jest interakcja pomiędzy rozwiązaniami z różnych obszarów przestrzeni poszukiwań oraz dynamiczna adaptacja tych relacji,
- wyznaczania strategii gracza,
- dekompozycji problemu na mniejsze części (algorytm kooperacyjny),

- wymuszenia adaptacji osobników z jednej populacji względem osobników z innej (algorytm konkurencyjny).

Zasada działania algorytmów koewolucyjnych oparta jest na przetwarzaniu wielu populacji oraz ocenie osobników na podstawie przystosowania do środowiska oraz kooperacji z osobnikami z innych populacji. Ogólny schemat działania algorytmu koewolucyjnego przedstawiono na schemacie 1.

1. for populacja $p_s \in P$, wszystkie populacje
1.1 ***Inicjalizacja*** *populacji p_s*
2. for populacja $p_s \in P$, wszystkie populacje
2.1 ***Ocena*** *populacji p_s*
3. t := 0
4. do
4.1 for populacja $p_s \in P$, wszystkie populacje
4.1.1 ***Selekcja***
4.1.2 ***Operacje genetyczne***
4.1.3 ***Wybór kooperantów*** *z P*
4.1.4 ***Ocena osobników***
4.1.5 ***Sukcesja*** *do nowej populacji p_s*
4.2 t := t + 1
until ***kryterium zatrzymania****.*

Schemat 1. Schemat działania algorytmu koewolucyjnego

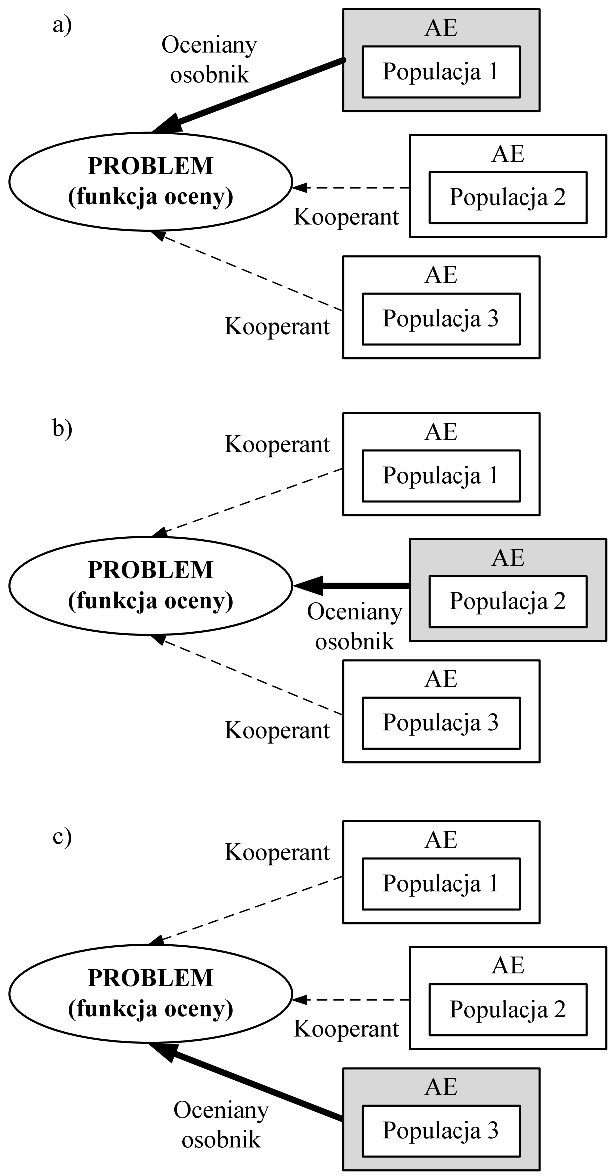
Zastosowanie algorytmów koewolucyjnych wymaga dekompozycji problemu oraz przypisania każdego komponentu do osobnej populacji. Każda z populacji ewoluuje osobno, a pojedynczy osobnik z danej populacji stanowi jedynie element potencjalnego rozwiązania. W celu oceny jego przystosowania z pozostałych populacji wybrani są tzw. kooperanci. Osobnik jest łączony z wybranymi kooperantami tworząc kompletne rozwiązanie problemu, które podlega ocenie. Możliwy jest wybór więcej niż jednego zestawu kooperantów w celu określenia wartości funkcji przystosowania. Idea oceny osobnika wraz z kooperantami została przedstawiona na rysunku 1.

Podejście koewolucyjne zostało pomyślnie zastosowane do wielu problemów, zarówno w wariancie kooperacyjnym jak i konkurencyjnym takich jak: optymalizacja dużej skali [5], identyfikacja uszkodzeń [6], rozdział mocy [7], planowanie pracy sklepu [8], identyfikacja systemów nieliniowych [9] oraz z innymi metodami inteligentnymi: sztucznymi sieciami neuronowymi [10], logiką rozmytą [11] czy algorytmami rojowymi [12].

3. FUNKCJE TESTOWE

Na podstawie [13], [14], [15] wybrano 3 funkcje testowe przy pomocy których przeprowadzono badania w celu oceny działania algorytmu koewolucyjnego. Funkcja 1 jest nieciągła i unimodalna na badanym przedziale. Funkcja 2 jest ciągła i wielomodalna na badanym przedziale. Funkcja 3 jest nieciągła i wielomodalna na badanym przedziale.

Funkcja 1 (funkcja Rosenbrocka). Do celów badawczych funkcja została zmodyfikowana, zmienna x_2 należy do zbioru liczb całkowitych. Minimum globalne funkcji znajduje się wewnątrz parabolicznego wgłębienia w punkcie $f_1(1, 1, 1) = 0$.



Rys. 1. Przebieg procesu oceny osobników dla 3 populacyjnego algorytmu koewolucyjnego

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{2} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right], \boldsymbol{x} \in \Omega_1 \qquad (1)$$

gdzie $\Omega_1 = \{(x_1, x_2, x_3): x_1 \in [-10, 10] \wedge \mathbb{R}, x_2 \in [-10, 10] \wedge \mathbb{Z}, x_3 \in [-10, 10] \wedge \mathbb{R}\}$

Funkcja 2 (funkcja Styblinskiego-Tanga). Funkcja posiada minimum globalne w punkcie $f_2(-2{,}903534, -2{,}903534, -2{,}903534) = -117{,}49797$

$$f_2(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{3}\left[x_i^4 - 16x_i^2 + 5x_i\right], \boldsymbol{x} \in \Omega_2 \qquad (2)$$

gdzie $\Omega_2 = \{(x_1, x_2, x_3): x_1, x_2, x_3 \in [-5, 5] \wedge \mathbb{R}\}$

Funkcja 3 (funkcja Schaffer'a nr 4). Funkcja została zmodyfikowana do celów badawczych, zmienna x_1 należy do zbioru liczb całkowitych. Funkcja posiada minimum w punkcie $f_3(0, 1{,}25313) = 0{,}292579$.

$$f_3(\boldsymbol{x}) = \frac{1}{2} + \frac{\cos^2(\sin(|x_1^2 - x_2^2|)) - 0{,}5}{\left(1 + 0{,}001(x_1^2 + x_2^2)\right)^2}, \boldsymbol{x} \in \Omega_3 \quad (3)$$

gdzie $\Omega_3 = \{(x_1, x_2): x_1 \in [-100, 100] \wedge \mathbb{Z}, x_2 \in [-100, 100] \wedge \mathbb{R}\}$

Zadanie optymalizacji definiuje się następująco:

$$\min f_1(\boldsymbol{x}), \boldsymbol{x} \in \Omega_1 \quad (4)$$

$$\min f_2(\boldsymbol{x}), \boldsymbol{x} \in \Omega_2 \quad (5)$$

$$\min f_3(\boldsymbol{x}), \boldsymbol{x} \in \Omega_3 \quad (6)$$

Przyjęto model algorytmu koewolucyjnego kooperacyjnego. Dokonano dekompozycji problemu optymalizacji. Każdy argument funkcji został przypisany do osobnej populacji. Kooperanci do oceny osobnika są wybierani na podstawie ich przystosowania. Najlepiej przystosowany osobnik z populacji w danej generacji zostaje kooperantem dla pozostałych populacji. Jako mechanizm selekcji zastosowano ruletkę proporcjonalną. Operator krzyżowania zamienia wartości pomiędzy osobnikami. Operator mutacji zmienia wartość osobnika na losową z przedziału ograniczeń. Jako kryterium stopu przyjęto liczbę generacji.

4. WYNIKI BADAŃ

Dla każdej z funkcji testowych przeprowadzono serię 10 symulacji w dwóch wariantach zależnych od liczby generacji. Otrzymane rezultaty zestawiono w tabeli 1 dla funkcji Rosenbrocka, tabeli 2 dla funkcji Styblinskiego-Tanga oraz w tabeli 3 dla funkcji Schaffer'a nr 4. Tabela została skonstruowana w następujący sposób: wiersz Gen zawiera liczbę generacji, po której zakończono działanie algorytmu, kolumna L.p. oznacza liczbę porządkową kolejnej symulacji, kolumny x_1, x_2, x_3 zawierają wartości argumentów wyznaczone przez algorytm jako najlepsze, kolumna $f(\boldsymbol{x})$ zawiera wartość funkcji celu dla tych argumentów.

Tablica 1. Wyniki badań dla funkcji Rosenbrocka

Gen	200				500			
L.p.	x_1	x_2	x_3	$f(\boldsymbol{x})$	x_1	x_2	x_3	$f(\boldsymbol{x})$
1	0,991	1	1,005	0,036	1	1	0,998	0
2	0,999	1	1,002	0	0,589	0	0	128,9
3	0,583	0	0,004	128,9	1,004	1	0,999	0,008
4	0,992	1	1.001	0,03	0,998	1	1	0,001
5	0,995	1	1	0,008	1,001	1	1,002	0,001
6	0,999	1	1,004	0,002	0,999	1	1,005	0,002
7	1,34	2	4,002	115,7	1,363	2	3,996	115,7
8	1	1	1,008	0,007	1	1	0,997	0
9	0,604	0	0,004	129	1	1	1,001	0
10	0,998	1	1,001	0	1,003	1	0,993	0,01

Tablica 2. Wyniki badań dla funkcji Styblinskiego-Tanga

Gen	100				500			
L.p.	x_1	x_2	x_3	$f(\boldsymbol{x})$	x_1	x_2	x_3	$f(\boldsymbol{x})$
1	-2,89	-2,91	-2,91	-117,495	-2,9	-2,9	-2,9	-117,498
2	-2,9	-2,89	-2,88	-117,494	-2,9	-2,9	-2,9	-117,498
3	-2,9	-2,9	-2,89	-117,498	-2,9	-2,9	-2,9	-117,498
4	-2,91	-2,9	-2,9	-117,497	-2,9	-2,9	-2,9	-117,498
5	-2,9	-2,89	-2,89	-117,497	-2,9	-2,9	-2,9	-117,498
6	-2,9	-2,91	-2,9	-117,497	-2,9	-2,9	-2,9	-117,498
7	-2,9	-2,87	-2,91	-117,478	-2,9	-2,9	-2,9	-117,497
8	-2,9	-2,9	-2,9	-117,498	-2,9	-2,9	-2,9	-117,498
9	-2,9	-2,91	-2,88	-117,485	-2,9	-2,9	-2,9	-117,498
10	-2,89	-2,89	-2,92	-117,488	-2,9	-2,9	-2,9	-117,498

Tablica 3. Wyniki badań dla funkcji Schaffer'a nr 4

Gen	1000			3000		
L.p.	x_1	x_2	$f(\boldsymbol{x})$	x_1	x_2	$f(\boldsymbol{x})$
1	0	1,25431	0,292583	3	-1,07123	0,296087
2	0	-1,25431	0,292583	-7	0,54628	0,311038
3	0	-2,81075	0,296138	0	1,25431	0,292583
4	1	1,60832	0,293525	0	1,25431	0,292583
5	0	1,25431	0,292583	3	1,07121	0,296087
6	0	-1,25431	0,292583	0	1,24821	0,292647
7	-4	-1,36417	0,299166	4	-1,35197	0,299166
8	7	-0,55238	0,211021	4	-1,35197	0,299694
9	0	-1,26652	0,293094	0	1,23634	0,293403
10	0	-1,24821	0,292647	0	-1,25431	0,292583

Na podstawie przeprowadzonych badań stwierdzono, że kooperacyjny algorytm koewolucyjny poprawnie wyznacza globalne minimum funkcji Styblinskiego-Tanga. Niezależnie od przeprowadzonej próby i liczby generacji algorytm wyznacza rozwiązanie bliskie optymalnemu z pomijalnym błędem.

Dla funkcji Rosenbrocka, przy kryterium zatrzymania 200 generacji algorytm wyznaczył poprawne optimum globalne w 7 na 10 prób. Po zwiększeniu liczby generacji do 500 powtarzalność wzrosła do 8 na 10 prób. W pozostałych przypadkach niepoprawne rozwiązania wynikały z błędnego wyznaczenia wartości argumentu x_2 (odpowiadającego za nieciągłość funkcji). Uzyskany rezultat uznaje się za zadowalający.

Wyniki uzyskane dla funkcji Schaffer'a nr 4 oscylują blisko minimum globalnego. Wartości funkcji celu są zbliżone, jednak wyznaczone punkty różnią się. Wynika to z przebiegu funkcji w okolicy optimum. Funkcja jest silnie wielomodalna na wąskim przedziale oraz nieciągła względem argumentu x_1. Poprawne rozwiązanie uzyskano w 2 z 10 prób dla kryterium zatrzymania po 1000 generacji oraz w 4 z 10 prób dla 3000 generacji. Dalsze zwiększanie liczby generacji nie skutkowało poprawą rezultatów. Uzyskane wyniki, pomimo wyznaczenia wartości funkcji celu bliskiej optimum globalnemu uznaje się za niepoprawne ze względu na wyznaczenie w większości przypadków niepoprawnych wartości argumentów $\boldsymbol{x}$.

5. WNIOSKI KOŃCOWE

W artykule przedstawiono algorytm koewolucyjny, heurystyczną metodę wyznaczania globalnego optimum funkcji. Algorytm testowano wykorzystując trzy wielomodalne, nieciągłe funkcje celu. Uznano, że algorytm poprawnie wyznaczył minima dla dwóch badanych funkcji. Rozwiązania uzyskane dla funkcji trzeciej uznano za niepoprawne. W celu uzyskania lepszych rezultatów należy przeprowadzić precyzyjne strojenie algorytmu lub zaproponować metodę lub modyfikację pozwalającą na zawężenie (eksploatację) przestrzeni poszukiwań w okolicy optimum globalnego poprzez modyfikację nacisku selektywnego.

Przeprowadzone testy i ich rezultaty są wstępem do badań nad algorytmami koewolucyjnymi obejmującymi zagadnienia dekompozycji i reprezentacji problemu, doboru metod interakcji (kooperacji, konkurencji), w tym sposobu wyboru kooperantów, sposobu oceny, topologii i struktury populacji oraz możliwości praktycznego zastosowania metody.

6. BIBLIOGRAFIA

1. Wong K. C., Leung K. S., Wong M. H.: Protein structure prediction on a lattice model via multimodal optimization techniques, Proceedings of the 12th annual conference on Genetic and evolutionary computation, Nowy Jork, USA, 2010, s. 155-162.
2. Corn M., Atanasijević-Kunc M.: Designing model and control system using evolutionary algorithms, IFAC-PapersOnLine, Tom 48, Nr 1, 2015, s. 526-531.
3. Corno F., Prinetto P., Rebaudengo M., Sonza Reorda M.: Exploiting competing subpopulations for automatic generation of test sequences for digital circuits, Applications Of Evolutionary Computation Evolutionary Computation In Electrical, Electronics, And Communications Engineering, LNCS Tom 1141, 2005, s. 791-800.
4. Kuczkowski Ł., Śmierzchalski R.: Impact of Initial Population on Evolutionary Path Planning Algorithm, Aktualne problemy automatyki i robotyki, Monografie Komitetu Automatyki i Robotyki Polskiej Akademii Nauk, Tom 20, 2014, s. 806 – 814.
5. Yang Z.Y., Tanga K., Yao X.: Large scale evolutionary optimization using cooperative coevolution, Information Sciences, Vol. 178, Issue: 15, 2008, s. 2985-2999.
6. Kouchmeshky B., Aquino W., Bongard J.C., Lipson H.: Co-evolutionary algorithm for structural damage identification using minimal physical testing, International Journal for Numerical Methods in Engineering, Vol. 69, Issue: 5, 2007, s. 1085-1107.
7. Chen H.Y., Wang X.F.: Cooperative Coevolutionary Algorithm for unit commitment, IEEE Transactions On Power Systems, Vol. 17, Issue: 1, 2002, s. 128-133.
8. Gu J.W., Gu M.Z., Cao C.W., Gu X.S.: A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem, Computers & Operations Research, Vol. 37, Issue: 5, 2010, s. 927-937.
9. Bongard J.C., Lipson H.: Nonlinear system identification using coevolution of models and tests, IEEE Transactions on Evolutionary Computation, Vol. 9, Issue: 4, 2005, s. 361-384.
10. Garcia-Pedrajas N., Hervas-Martinez U., Ortiz-Boyer D.: Cooperative coevolution of artificial neural network ensembles for pattern classification, IEEE Transactions on Evolutionary Computation, Vol. 9, Issue: 3, 2005, s. 271-302.
11. Pena-Reyes C.A., Sipper M.: Fuzzy CoCo: A cooperative-coevolutionary approach to fuzzy modeling, IEEE Transactions on Fuzzy Systems, Vol. 9, Issue: 5, 2001, s. 727-737.
12. Li X.D., Yao X.: Cooperatively Coevolving Particle Swarms for Large Scale Optimization, IEEE Transactions on Evolutionary Computation, Vol. 16, Issue: 2, 2012, s. 210-224.
13. Bäck T.: Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Oxford University Press, Oxford 1995, s. 328.
14. Haupt R. L., Haupt, S. E.: Practical genetic algorithms with DC-Rom, Wiley, New York 2004.
15. Oldenhuis R.: Many test functions for global optimizers, Mathworks 2012.

OPTIMIZATION OF DISCONTINUOUS AND MULTIMODAL FUNCTIONS USING COOPERATIVE COEVOLUTIONRY ALGORITHM

In this paper a brief study of coevolutionary algorithm is presented. The coevolutionary algorithm (CA) is an evolutionary algorithm (or collection of evolutionary algorithms) in which the fitness of an individual depends on the relationship between that individual and other individuals. CA can be divided into two fundamental sub-types. In cooperative algorithms, individuals are rewarded when they work well with other individuals and punished when they perform poorly. In competitive algorithms, however, individuals are rewarded at the expense of those with which they interact. The principle of operation of CA is quite similar to traditional evolutionary algorithm. The main deference lies in a fact that CA operate on multi-populations and evaluate individual based on its collaboration with individuals (collaborators) from other populations. Applying CA requires decomposition of the problem into components and assigning each component to a population. This article presents an optimization of discontinuous and multimodal functions using cooperative coevolutionry algorithm. The modified testing functions: Rosenbrocka, Styblinskiego-Tanga and Schaffer'a are decomposed and minimize using coevolutionary algorithm. Obtained results allow to evaluate the quality of the algorithm and will be used for further research on the topic.

Keywords: evolutionary algorithms, coevolution, optimization.