

Andrzej RUDEŃSKI

METODY NIEDETERMINISTYCZNE W OPTYMALIZACJI SILNIKÓW INDUKCYJNYCH JEDNOFAZOWYCH

STRESZCZENIE *W artykule przedstawiono zastosowanie metod niedeterministycznych w optymalizacji jednofazowych silników indukcyjnych. Testowano trzy odmiany strategii ewolucyjnych oraz algorytm genetyczny z częściową wymianą populacji i zmiennoprzecinkową reprezentacją genów w chromosomach. Na podstawie eksperymentów obliczeniowych zbadano wpływ liczby osobników w populacji, liczby pokoleń oraz zastosowanych operatorów genetycznych na uzyskane wyniki tj. na: średnią wartość funkcji przystosowania, odchylenie standardowe oraz czas obliczeń. Optymalizację przeprowadzono przy 12-tu zmiennych niezależnych oraz 9-ciu ograniczeniach. Założono ciągłą zmienność wszystkich zmiennych niezależnych. Jako funkcję celu przyjęto koszt materiałów czynnych oraz koszt eksploatacji silnika w zadanym czasie pracy. Przedstawiono wyniki obliczeń optymalizacyjnych sześciu silników indukcyjnych jednofazowych z kondensatorem pracy oraz rozruchowym o mocy znamionowej od 1,1 kW do 11 kW, przy użyciu testowanych metod oraz wnioski z przeprowadzonych eksperymentów obliczeniowych.*

Słowa kluczowe: *silniki indukcyjne jednofazowe, optymalizacja, metody ewolucyjne*

dr inż. Andrzej RUDEŃSKI

e-mail: iel.nme@iel.waw.pl

Zakład Maszyn Elektrycznych
Instytut Elektrotechniki

1. WPROWADZENIE

Niedeterministyczne metody optymalizacji można podzielić na stochastyczne oraz ewolucyjne.

Metody stochastyczne polegają na losowym generowaniu punktów w n -wymiarowej przestrzeni, w których jest obliczana wartość funkcji celu.

Charakteryzują się one powolną zbieżnością i pomimo stosowania różnych prób jej zwiększenia, nie znajdują praktycznego zastosowania w projektowaniu obiektów o złożonej strukturze, jakimi są np. maszyny elektryczne.

Metody ewolucyjne polegają na wykorzystaniu do celów optymalizacji elementów teorii ewolucji gatunku, tzn. doboru naturalnego, selekcji oraz dziedziczenia. Odznaczają się one następującymi cechami, które odróżniają je od pozostałych metod optymalizacji:

- poszukiwanie poprawy rozwiązań rozpoczyna się nie z pojedynczego punktu jak w metodach deterministycznych, ale z ich zbioru nazywanego populacją – przy czym zbiór ten może w szczególnym przypadku zawierać wielokrotne kopie jednego osobnika;
- korzysta się w nich tylko z funkcji celu lub jej modyfikacji nazywanej funkcją przystosowania, a nie z jej pochodnych;
- w większości z nich stosuje się stochastyczne, a nie deterministyczne sposoby wyboru parametrów określających strategię realizacji algorytmu.

Generowanie populacji początkowej, a także przeszukiwanie przestrzeni rozwiązań, odbywa się metodami stochastycznymi, a modyfikacji poszczególnych osobników populacji w kolejnych pokoleniach dokonuje się za pomocą operatorów rekombinacji (krzyżowania chromosomów) oraz mutacji (modyfikacji genów). Podstawową zaletą metod ewolucyjnych jest możliwość zwiększenia liczby zmiennych niezależnych optymalizacji nawet do kilkuset oraz znalezienia rozwiązania w punkcie optimum globalnego (przy właściwej strukturze algorytmu ewolucyjnego). Ponadto ich zastosowanie pozwala rozszerzyć klasę analizowanych funkcji celu, poprzez włączenie funkcji nieróżniczkowalnych, nieciągłych a nawet zawierających zmienne dyskretne. Dlatego są rozszerzane badania nad ich zastosowaniem do obiektów o coraz bardziej złożonej strukturze.

Ze względu na złożoność obliczeniową oraz duży nakład pracy, związane z wykorzystaniem metod stochastycznych, przy zastosowaniu procedur ewolucyjnych trzeba posługiwać się zaawansowanym sprzętem informacyjnym.

Powtarzając obliczenia przy takich samych danych początkowych i takiej samej strategii algorytmu, otrzymuje się za każdym razem inne rozwiązanie. Wadą metod ewolucyjnych jest zatem ich mniejsza dokładność w porównaniu z metodami deterministycznymi. To znaczy nie można jednoznacznie obliczyć składowych wektora w punkcie optymalnym. Skłania to do poszukiwania skutecznych metod hybrydowych, tj. łączących zalety obu metod ewolucyjnych i deterministycznych.

Algorytmy realizujące zadania optymalizacji metodami ewolucyjnymi można podzielić na: strategie ewolucyjne; programowanie ewolucyjne; algorytmy genetyczne [2].

Podział ten wynika z kierunków ich historycznego rozwoju. Pierwsza z wymienionych grup była rozwijana w Niemczech, druga na zachodnim, a trzecia na wschodnim wybrzeżu USA [2]. W pierwszych dwóch grupach stosuje się zmiennoprzecinkową reprezentację genów w chromosomach (odpowiadających poszczególnym zmiennym niezależnym). Ostatnia grupa metod początkowo bazowała na binarnym kodowaniu genów, jednak związane z tym niedogodności spowodowały, że stosuje się w nich także zmiennoprzecinkową reprezentację genów.

W ostatnim dwudziestoleciu na skutek dynamicznego rozwoju omawianych metod oraz coraz szerszego ich zastosowania w różnych dziedzinach, powstało wiele nowych odmian algorytmów ewolucyjnych oraz wyspecjalizowanych metod służących np. do rozwiązywania zadań optymalizacji w obszarze ograniczonym. Niektóre z nich wymagają stosowania specjalnych operatorów rekombinacji i mutacji [12; 13]. Takie okoliczności występują przy projektowaniu maszyn elektrycznych, a w szczególności silników indukcyjnych.

Zastosowanie algorytmów ewolucyjnych do zadań optymalizacji w dziedzinie maszyn elektrycznych jest jeszcze w fazie początkowej. W zadaniach optymalizacji silników indukcyjnych główną trudność stanowi nieliniowość funkcji celu oraz funkcji ograniczeń. Ponadto zarówno funkcja celu jak i funkcje ograniczeń nie dają się wyrazić analitycznie w zależności od zmiennych niezależnych sterujących optymalizacją; wyznaczanie ich wartości musi się odbywać za pomocą algorytmów i złożonych obliczeń iteracyjnych [6]. Formalnie nie jest istotny sposób obliczania funkcji celu oraz funkcji ograniczeń pod warunkiem, że obliczenia iteracyjne dają wyniki jednoznaczne podobnie jak obliczenia funkcji wyrażonych analitycznie. Ma to jednak istotny wpływ na pracochłonność obliczeń.

Pierwsze próby zastosowania tych metod do optymalizacji przetworników elektromechanicznych, dotyczyły obiektów o prostej strukturze [10; 18], albo bazowały na uproszczonych modelach matematycznych maszyn [5]. W pracach [11; 18] zastosowano klasyczny algorytm genetyczny z prostym binarnym kodowaniem genów w chromosomach. Nie posłużono się natomiast algorytmami

wykorzystującymi zmiennoprzecinkową reprezentację genów w chromosomach. Jak wykazano m. in. w pracach [1; 13], pomimo braku matematycznego dowodu zbieżności takich algorytmów, są one przydatne do rozwiązywania zadań optymalizacji spotykanych w technice, a to dzięki naturalnej reprezentacji zmiennych decyzyjnych oraz możliwości stosowania wyspecjalizowanych operatorów genetycznych. Ponadto, w przeciwieństwie do algorytmów z naturalnym kodowaniem binarnym, nie wykazują one „fałszywych” dodatkowych ekstremów lokalnych, które pojawiają się na skutek tzw. efektu klifów Hamminga [1; 13].

Autorzy prac zwykle pomijają szczegóły zastosowanych metod uważając, że samo sformułowanie "metody ewolucyjne" jest wystarczające. Jest to niesłuszne, ponieważ stale się pogłębia "róznicowanie" tych metod. W literaturze [13; 14] podkreśla się konieczność prowadzenia prac badawczych ukierunkowanych na dostosowanie algorytmów ewolucyjnych do specyfiki rozwiązywanych zadań. Dopiero takie postępowanie umożliwia znalezienie metod najbardziej odpowiednich do rozpatrywanego zagadnienia.

Próby zastosowania algorytmów ewolucyjnych do optymalizacji w maszynach elektrycznych są także przedstawione w pracach [11; 16]. Rozpoznawcze eksperymenty obliczeniowe z wykorzystaniem algorytmu ewolucyjnego do optymalizacji silnika jednofazowego zostały przedstawione w pracy [16]. Wynika z nich m.in. potrzeba bardziej wnikliwej analizy ich właściwości oraz sklasyfikowania pod względem przydatności do rozwiązywania zadań optymalizacji w procesie projektowania silników indukcyjnych, co jest celem tej pracy.

2. ALGORYTM OBLICZANIA FUNKCJI PRZYSTOSOWANIA

W tej pracy zastosowano programy testowe przystosowując zmodyfikowane algorytmy obliczania funkcji celu stosowane w optymalizacji deterministycznej przedstawione w opracowaniu [15]. Przedmiotem optymalizacji jest silnik indukcyjny jednofazowy z kondensatorem pracy oraz włączanym na czas rozruchu kondensatorem rozruchowym. Jako kryterium optymalizacji przyjęto minimum sumy kosztów materiałów czynnych użytych do produkcji oraz kosztów eksploatacji silnika w określonym czasie (do obliczeń przyjęto 5000 godzin). Funkcja celu zależy od 12-tu zmiennych, a optymalizacja jest przeprowadzana przy 9-ciu ograniczeniach. Zbiór zmiennych może być zmniejszony, tzn. część zmiennych może nie zostać uaktywniona, ale w tej pracy eksperymenty obliczeniowe wykonano przy pełnym zbiorze zmiennych niezależnych. Są nimi: śred-

nica wewnętrzna rdzenia stojana D ; długość rdzenia stojana L ; szerokość żłobka stojana b_{Qs} ; wysokość zęba stojana h_{ds} ; szerokość żłobka wirnika b_{Qr} ; wysokość zęba wirnika h_{dr} ; liczba zwojów uzwojenia głównego N_A ; liczba zwojów uzwojenia pomocniczego N_B ; średnica drutu uzwojenia głównego d_A ; średnica drutu uzwojenia pomocniczego d_B ; szerokość pierścienia zwierającego klatki wirnika a_{er} ; pojemność kondensatora pracy C_p .

Jako ograniczenia przyjęto: minimalną sprawność η_{\min} ; minimalny współczynnik mocy $(\cos\varphi)_{\min}$; minimalny względny początkowy moment rozruchowy $m_{k\min}$; maksymalny względny początkowy prąd rozruchowy $i_{k\max}$; minimalny względny moment krytyczny $m_{b\min}$; – maksymalną indukcję w zębach stojana $B_{d\max}$; maksymalną gęstość prądu w uzwojeniach stojana j_{\max} ; maksymalny współczynnik wypełnienia żłobków $k_{z\max}$; maksymalne napięcie na kondensatorze pracy przy biegu jałowym $U_{c0\max}$.

Ponieważ celem tej pracy jest głównie ocena przydatności różnych algorytmów ewolucyjnych w optymalizacji, to obliczenia wykonano przy założeniu ciągłości wszystkich zmiennych niezależnych, nie wykonując po zakończeniu obliczeń "urealnienia" otrzymanych rozwiązań.

W algorytmach ewolucyjnych osobników ocenia się na podstawie wartości przypisywanej im funkcji przystosowania. Funkcja ta jest uzależniona od wartości funkcji celu oraz od zbioru funkcji ograniczeń. Przy czym w odróżnieniu od metod deterministycznych, w których zwykle przyjmuje się kary wewnętrzne, w metodach niedeterministycznych stosuje się kary zewnętrzne – tj. zależne od przekroczenia ograniczeń.

Przy takich założeniach funkcja przystosowania ma postać:

$$f(\mathbf{x}) = f_c + \sum_{i=1}^m w_i \cdot g(\mathbf{x}) \quad (1)$$

przy czym:

- \mathbf{x} – wektor zmiennych niezależnych;
- f_c – wartość funkcji celu, zależnej od kryterium optymalizacji;
- w_i – współczynnik wagowy i -tego ograniczenia;
- $g(\mathbf{x})$ – funkcja ograniczeń;
- m – liczba ograniczeń.

Funkcja $g(\mathbf{x})$ może być liniowa, kwadratowa lub logarytmiczna. Także współczynniki wagowe dla poszczególnych ograniczeń są na ogół różne. Ich właściwy dobór w istotny sposób wpływa na przebieg procesu ewolucyjnego i jest trudnym zadaniem. W opracowanych programach narzędziowych przewidziano możliwość zmian zarówno funkcji kary jak i współczynników wagowych poszczególnych ograniczeń.

Każdy składnik p_i sumy we wzorze (1), w zależności od tego, jaka funkcja kary została wybrana, jest obliczany wg jednej z zależności:

$$p_i = \begin{cases} w_i \cdot f_c \cdot [g_i(\mathbf{x}) - g_{i,d}] & \text{dla } g_i(\mathbf{x}) > g_{i,d} \\ 0 & \text{dla } g_i(\mathbf{x}) \leq g_{i,d} \end{cases} \quad (2a)$$

$$p_i = \begin{cases} w_i \cdot f_c \cdot [g_i(\mathbf{x}) - g_{i,d}]^2 & \text{dla } g_i(\mathbf{x}) > g_{i,d} \\ 0 & \text{dla } g_i(\mathbf{x}) \leq g_{i,d} \end{cases} \quad (2b)$$

$$p_i = \begin{cases} w_i \cdot f_c \cdot \ln \left[\frac{g_i(\mathbf{x})}{g_{i,d}} \right] & \text{dla } g_i(\mathbf{x}) > g_{i,d} \\ 0 & \text{dla } g_i(\mathbf{x}) \leq g_{i,d} \end{cases} \quad (2c)$$

przy czym: $g_i(\mathbf{x})$ – obliczona wartość i -tego parametru podlegającego ograniczeniu; $g_{i,d}$ – wartość dopuszczalna i -tego ograniczenia. Wzory (2a)...(2c) dotyczą ograniczeń od dołu; dla ograniczeń od góry zmienia się kolejność składników w nawiasach kwadratowych we wzorach (2a) i (2b) oraz licznik z mianownikiem pod znakiem logarytmu we wzorze (2c).

Znanych jest kilka strategii kar, przedstawionych m.in. w pracach [13; 14] polegających na: różnicowaniu wartości nakładanych kar w zależności od stopnia zaawansowania poszukiwań, tj. od numeru pokolenia, albo na doborze kar w taki sposób, aby osobnik niedopuszczalny zawsze miał gorszą wartość przystosowania od osobnika dopuszczalnego. Może się bowiem zdarzyć, że rozwiązanie, w którym zostało nieznacznie naruszone jedno z graniczeń, ma lepszą wartość przystosowania od rozwiązania dopuszczalnego.

Rolę ograniczeń spełniają także dolne i górne wartości poszczególnych zmiennych niezależnych, określające przestrzeń poszukiwań. Ich uwzględnienie w algorytmie jest proste, lecz ich obecność nakłada często trudne do spełnienia warunki na algorytm obliczania funkcji celu. Ze względu bowiem na stochastyczny sposób przeszukiwania przestrzeni rozwiązań może się zdarzyć, że wygenerowane wartości zmiennych będą prowadzić np. do bardzo dużych indukcyj w jednym z odcinków obwodu magnetycznego maszyny, co może doprowadzić do "utknięcia" algorytmu iteracyjnego. Takie rozwiązanie jest jednak z reguły technicznie niedopuszczalne, a więc można go pominąć, zastępując innym rozwiązaniem dopuszczalnym, co nie podważa istoty działania algorytmu ewolucyjnego. W opracowanych programach obliczeń zastosowano mechanizm "naprawy" takich osobników, polegający na ich wymianie na innego, pochodzącego z dodatkowej populacji osobników dopuszczalnych.

3. ZASTOSOWANE ALGORYTMY EWOLUCYJNE

Dla realizacji postawionego zadania opracowano w programowej formie obiektowej procedury realizujące obliczenia optymalizacyjne oparte na różnych metodach ewolucyjnych. Testowano trzy odmiany strategii ewolucyjnych oraz algorytm genetyczny ze zmiennoprzecinkową reprezentacją genów w chromosomach, w którym wykorzystano część rozwiązań zastosowanych w algorytmie GENOCOP przedstawionym w pracach [13; 14]. Podstawowe cechy tych algorytmów przedstawiono w następujących podpunktach.

Jako punkt startowy wprowadza się wektor zmiennych niezależnych (genów) pojedynczego, niekoniecznie dopuszczalnego osobnika, graniczne wartości wszystkich zmiennych niezależnych oraz zbiór ograniczeń.

W wieloosobniczych strategiach ewolucyjnych inicjalizacja populacji początkowej polega na wygenerowaniu żądanej liczby osobników, których chromosomy zawierają geny o wartościach wybranych losowo z przedziałów zmienności odpowiadających im zmiennych. Natomiast w algorytmie genetycznym z populacji początkowej eliminowane są osobniki, przy których algorytm iteracyjny ze wspomnianych w poprzednim punkcie względów mógł nieprawidłowo obliczyć ich przystosowanie. To samo dotyczy pojedynczego osobnika w jednoosobniczej strategii ewolucyjnej tzw. (1+1)-SE. Taki sposób postępowania wynika z zastosowanych mechanizmów selekcji.

3.1. Strategie ewolucyjne

Strategie ewolucyjne należą do najstarszych spośród stosowanych niedeterministycznych metod optymalizacji funkcji wielu zmiennych. Stosowana jest w nich selekcja deterministyczna, tzn. do następnego pokolenia przechodzą osobniki o lepszym przystosowaniu. W najprostszej strategii (1+1)-SE, populację stanowi pojedynczy osobnik, który tworzy jednego potomka poprzez mutację każdego z genów rodzica zgodnie z zależnością:

$$x_i' = x_i + N(0, \sigma) \quad (3)$$

przy czym:

- x_i – i -ta składowa wektora zmiennych niezależnych (gen);
- $N(0, \sigma)$ – zmienna losowa o rozkładzie Gaussa z wartością oczekiwaną 0 oraz odchyleniem standardowym σ .

W tej wersji algorytmu zastosowano prostą samoczynną adaptację zasięgu mutacji wykorzystując zaproponowaną przez Rechenberga, tzw. "regułę 1/5 sukcesów". Polega ona na zmianie odchylenia standardowego σ co k pokoleń w zależności od tego, czy mutacja w ostatnich k pokoleniach spowodowała poprawę funkcji przystosowania – tj. wg zależności:

$$\begin{aligned} \sigma' &= \sigma \cdot a && \text{jeżeli } s/k < 0,2 \\ \sigma' &= \sigma \cdot \frac{1}{a} && \text{jeżeli } s/k > 0,2 \\ \sigma' &= \sigma && \text{jeżeli } s/k = 0,2 \end{aligned} \quad (4)$$

przy czym:

- $a < 1$ – parametr metody; zgodnie z [8] przyjęto $a = 0,82$;
- s – liczba mutacji uwiecznionych sukcesem.

Potomek zastępuje swojego rodzica w następnym pokoleniu wtedy i tylko wtedy, jeśli odznacza się lepszym przystosowaniem. W obliczeniach przystosowania zrezygnowano z dodawania składnika kary do obliczonej wartości funkcji celu, przestając na wygenerowaniu pierwszego osobnika (pokolenie 0) spełniającego wszystkie ograniczenia oraz wprowadzając dodatkowo ten wymóg do warunku akceptacji potomka.

W tej wersji algorytmu nie stosowano operatora krzyżowania (rekombinacji).

Drugą testowaną odmianą strategii ewolucyjnej jest tzw. strategia $(\mu + \lambda)$ -SE, w której w każdym pokoleniu μ rodziców tworzy λ potomków. Pary rodziców są wybierane spośród μ osobników populacji w sposób losowy. Każda para rodziców tworzy dwóch potomków poprzez zastosowanie wybranego operatora krzyżowania. Operacja krzyżowania jest powtarzana tak długo, aż uzyska się λ potomków. Następnie każdy z potomków jest poddawany mutacji z samoczynną adaptacją zasięgu. Polega ona na wprowadzeniu dodatkowego chromosomu σ , który nie ma wpływu na wartość przystosowania osobnika, a zawiera jedynie parametry wpływające na zasięg mutacji – tzn. wartości odchylenia standardowych rozkładu prawdopodobieństwa. Chromosom ten podczas operacji genetycznych ulega modyfikacjom, jest wykorzystywany podczas generowania potomstwa i podlega dziedziczeniu. Na skutek działania mechanizmu selekcji, potomstwo dziedziczy nie tylko lepsze geny wpływające na wartość funkcji przystosowania, ale również bardziej adekwatny zasięg operatorów mutacji.

Znanych jest wiele operatorów krzyżowania, w tej pracy przewidziano możliwość wyboru jednego z dwóch operatorów. Są nimi:

1. Krzyżowanie równomierne, w którym dwóch rodziców \mathbf{x}_1 oraz \mathbf{x}_2 tworzy dwóch potomków \mathbf{y}_1 oraz \mathbf{y}_2 . Potomek \mathbf{y}_1 zawiera chromosomy o składowych $y_{1,i} = x_{1,i}$ oraz $\sigma'_{1,i} = \sigma_{1,i}$ albo $y_{1,i} = x_{2,i}$ oraz $\sigma'_{1,i} = \sigma_{2,i}$ dla $i = 1, \dots, n$ pochodzących od jednego lub drugiego rodzica z jednakowym prawdopodobieństwem. Drugi potomek jest tworzony przez zaprzeczenie decyzji, od którego z rodziców ma pochodzić i -ty gen.

$$\begin{aligned}\mathbf{x}_1 &= (x_{1,1}, \dots, x_{1,n}; \sigma_{1,1}, \dots, \sigma_{1,n}) \\ \mathbf{x}_2 &= (x_{2,1}, \dots, x_{2,n}; \sigma_{2,1}, \dots, \sigma_{2,n}) \\ \mathbf{y}_1 &= (y_{1,1}, \dots, y_{1,n}; \sigma'_{1,1}, \dots, \sigma'_{1,n}) \\ \mathbf{y}_2 &= (y_{2,1}, \dots, y_{2,n}; \sigma''_{2,1}, \dots, \sigma''_{2,n})\end{aligned}\quad (5)$$

2. Krzyżowanie proporcjonalne zwane również arytmetycznym, w którym dwóch rodziców \mathbf{x}_1 oraz \mathbf{x}_2 tworzy dwóch potomków \mathbf{y}_1 oraz \mathbf{y}_2 wg reguły:

$$\begin{aligned}y_{1,i} &= a \cdot x_{1,i} + (1-a) \cdot x_{2,i} \\ y_{2,i} &= a \cdot x_{2,i} + (1-a) \cdot x_{1,i} \\ \sigma_{1,i} &= a \cdot \sigma_{1,i} + (1-a) \cdot \sigma_{2,i} \\ \sigma_{2,i} &= a \cdot \sigma_{2,i} + (1-a) \cdot \sigma_{1,i}\end{aligned}\quad (6)$$

przy czym:

$a \in [0,1]$ – jest zmienną losową o rozkładzie równomiernym.

Mutacja przebiega w dwóch etapach. W pierwszym z nich następuje modyfikacja składowych wektora odchyłeń standardowych σ zgodnie z zależnością:

$$\sigma'_i = \sigma_i \cdot \exp[\tau_p N(0,1) + \tau N_i(0,1)]\quad (7)$$

przy czym:

τ_p – parametr, przyjęty wg zależności: $\tau_p = \frac{1}{\sqrt{2\sqrt{n}}}$;

τ – parametr, przyjęty wg zależności: $\tau = \frac{1}{\sqrt{2n}}$;

n – liczba zmiennych niezależnych.

Pierwszy składnik w nawiasie we wzorze (7) zawiera zmienną losową o rozkładzie Gaussa wspólną dla wszystkich genów (zmiennych) chromosomu.

Natomiast drugi składnik zawiera zmienną losową inną dla każdego genu. Przy braku tego składnika zmiany wartości genów następowałyby tylko wzdłuż kierunków osi n -wymiarowego układu współrzędnych. Obecność drugiego składnika powoduje dodatkowe odchylenia zmian od tych kierunków.

W drugim etapie mutacji zmieniano wartości genów wg zależności:

$$x'_i = x_i + \sigma'_i \cdot N(0,1) \quad \text{lub} \quad x'_i = x_i + \sigma'_i \cdot C \quad (8)$$

Pierwszy wzór jest stosowany do mutacji z rozkładem Gaussa, a drugi w przypadku rozkładu Cauchy'ego, przy czym C – zmienna losowa o rozkładzie Cauchy'ego.

Następnie oblicza się przystosowanie wszystkich λ osobników potomnych, z zastosowaniem funkcji kary zewnętrznej (opcynie: liniowej, kwadratowej lub logarytmicznej), po czym jest dokonywana selekcja polegająca na malejącym posortowaniu wszystkich $\mu + \lambda$ osobników (rodziców i potomków) oraz utworzeniu nowego pokolenia z pierwszych μ najlepszych osobników, które są po wybraniu poddane tasowaniu. Selekcja jest więc deterministyczna, co jest cechą charakterystyczną wszystkich odmian strategii ewolucyjnych.

Trzecią testowaną odmianą strategii ewolucyjnej jest tzw. strategia (μ, λ) -SE. Różni się ona od strategii (μ, λ) -SE tym, że do następnego pokolenia wybiera się μ najlepiej przystosowanych osobników tylko spośród λ potomków. Rodzice nie wchodzi do nowego pokolenia, a więc każdy osobnik "żyje" tylko przez jedno pokolenie.

W eksperymentach obliczeniowych przy zastosowaniu strategii ewolucyjnych $(\mu+\lambda)$ -SE oraz (μ, λ) -SE, zrezygnowano ze stosowania podanego w p. 2 mechanizmu wymiany osobników, z którymi "nie radzi sobie" algorytm obliczeń wartości przystosowania. Osobnikom takim przypisuje się wartość przystosowania gorszą od najgorszego dotychczas osobnika, a deterministyczny charakter selekcji zawsze powoduje ich eliminację. Ta okoliczność została potwierdzona w wykonanych eksperymentach obliczeniowych.

3.2. Zmodyfikowany algorytm genetyczny

Testowano także algorytm genetyczny zawierający elementy algorytmu o nazwie GENOCOP, przedstawionego w pracach [13, 14]. Jest to algorytm z częściową wymianą populacji, w którym zastosowano zmiennoprzecinkową reprezentację genów w chromosomach oraz zmodyfikowaną w porównaniu z klasycznym algorytmem procedurę selekcji. Każda nowa populacja jest tworzona w następujący sposób:

- wybiera się p rodziców spośród N osobników populacji z zastosowaniem selekcji turniejowej przy c uczestnikach turnieju lub selekcji rankingowej (liniowej lub nieliniowej). Obydwa rodzaje selekcji umożliwiają regulację nacisku selektywnego;
- wybiera się bez powtórzeń spośród wszystkich N osobników populacji $N - p$ osobników, które są kopiowane bez zmian do następnego pokolenia, przy czym p – jest liczbą rodziców. Wyboru dokonuje się losowo z zastosowaniem selekcji turniejowej – tzn. wybiera się k -krotnie zbiór $N - p$ osobników, a do następnego pokolenia kopiuje się ten zbiór, w którym suma przystosowań osobników jest lepsza. Poprzez dobór liczby uczestników turnieju k można wpływać na nacisk selektywny;
- p osobników wybranych na rodziców tworzy p potomków, przy zastosowaniu operatorów krzyżowania i mutacji a potomków kopiuje się do następnego pokolenia.

Przewidziano możliwość wyboru jednego spośród dwóch operatorów krzyżowania: równomiernego lub proporcjonalnego oraz trzecią, polegającą na losowym wyborze jednego z operatorów. Krzyżowanie przebiega zgodnie z wzorem (4) lub (5), jednak w tym algorytmie nie występują dodatkowe chromosomy zawierające wartości odchyłeń standardowych stosowanych podczas mutacji.

W algorytmie zastosowano 3 operatory mutacji: równomierny; nierównomierny; graniczny.

Mutacja równomierna polega na nadaniu losowo wybranemu genowi x_k losowej wartości z przedziału $[x_{k,\min}, x_{k,\max}]$, czyli z przedziału zmienności zmiennej reprezentowanej przez ten gen.

Mutacja nierównomierna polega na nadaniu losowo wybranemu genowi x_k jednej z wartości (z jednakowym prawdopodobieństwem):

$$x_k' = \begin{cases} x_k + \Delta(t, x_{k,\max} - x_k) \\ x_k - \Delta(t, x_k - x_{k,\min}) \end{cases} \quad (9)$$

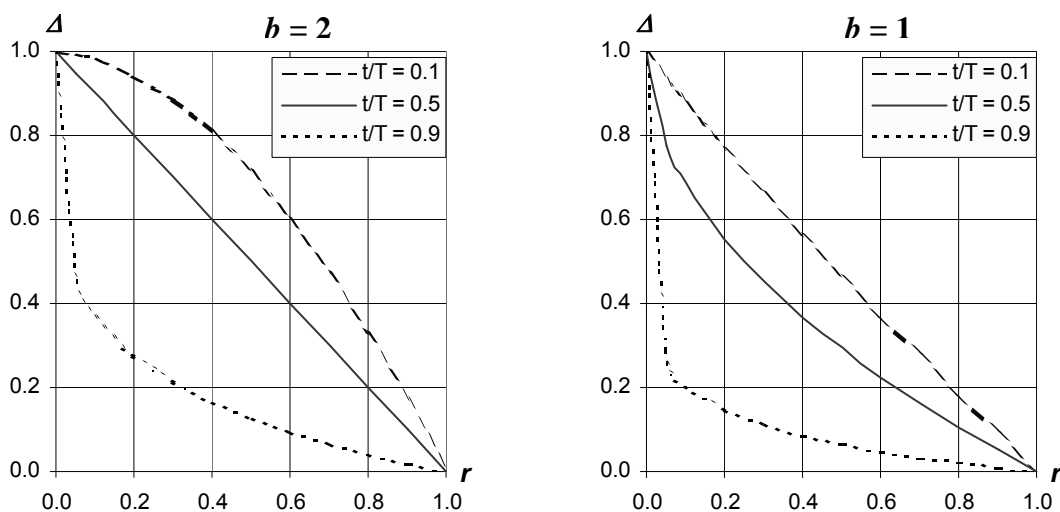
Funkcja $\Delta(t, y)$ przybiera wartości z przedziału $[0, y]$ i jest określona zależnością:

$$\Delta(t, y) = y \left[1 - r^{b \left(1 - \frac{t}{T} \right)} \right] \quad (10)$$

przy czym:

- y – szerokość przedziału $y = x_{k\max} - x_k$ lub $y = x_k - x_{k\min}$;
- r – liczba losowa z przedziału $[0, 1]$;
- t – aktualny numer pokolenia;
- T – najwyższy numer pokolenia;
- B – parametr określający stopień nierównomierności mutacji.

Jak wynika z wzoru (10) prawdopodobieństwo, że wartość funkcji $\Delta(t, y)$ jest bliska zero wzrasta wraz ze wzrostem numeru pokolenia. Oznacza to, że na początku procesu ewolucyjnego mutacja ma duży zasięg, co pozwala na eksplorację większego obszaru, a pod koniec procesu zasięg mutacji jest mały, co pozwala dokładniej znaleźć optimum. Na rysunku 1 przedstawiono zależność wartości funkcji $\Delta(t, y)$ od zmiennej losowej r przy różnych wartościach t / T oraz przy dwóch wartościach parametru b we wzorze (10).



Rys. 1. Zasięg mutacji nierównomiernej przy parametrach $b = 1$ oraz $b = 2$

Mutacja graniczna polega na nadaniu losowo wybranemu genowi x_k jednej z wartości granicznych $x_{k, \max}$ lub $x_{k, \min}$ (z jednakowym prawdopodobieństwem), jakie może przybierać reprezentowana przez ten gen zmienna.

W programie testowym rodzaj selekcji rodziców (turniejowa; rankingowa liniowa; rankingowa nieliniowa), parametry selekcji: c – liczba uczestników turnieju przy wyborze puli rodziców; k – liczba uczestników turnieju przy wyborze nie zmienianej części populacji oraz b – parametr nierównomierności mutacji

mogą być zmieniane. Liczbę potomków, którzy mają podlegać mutacji określa się arbitralnie i wybiera się ich losowo bez zwracania (spośród p). Pozostałe osobniki są poddawane jednemu z operatorów krzyżowania. Ponadto część spośród osobników podlegających mutacji jest poddawana mutacji granicznej. Liczba ta maleje z prawdopodobieństwem p_{m-bord} malejącym wraz z postępem procesu ewolucyjnego zgodnie z zależnością:

$$p_{m-bord.} = 0,3 \cdot (1 - t/T)^3 \quad (11)$$

Pozostałe osobniki ze zbioru przeznaczonych do mutacji są poddawane, zależnie od wyboru, mutacji równomiernej lub nierównomiernej.

4. WYNIKI OBLICZEŃ

Za pomocą programów testowych umożliwiających zmianę omawianych w poprzednim punkcie parametrów zrealizowano dwa zbiory eksperymentów. Pierwszy z nich zawiera obliczenia zmierzające do wyznaczenia najbardziej korzystnych wartości parametrów algorytmów ewolucyjnych oraz doboru operatorów genetycznych. Na ich podstawie określono: liczebność populacji N (dla strategii ewolucyjnych liczby rodziców μ oraz liczby potomków λ); liczebność puli rodzicielskiej p dla algorytmu genetycznego oraz liczbę pokoleń T . Przebieg procesu ewolucyjnego (najlepsze, najgorsze oraz średnie przystosowanie w kolejnych pokoleniach) może być przedstawiony w postaci wizualnej, co pomogło m.in. określić konieczne maksymalne liczby pokoleń oraz badać wpływ wyboru różnych operatorów genetycznych na przebieg procesu, tzn. na szybkość zbieżności oraz różnorodność populacji. Ponadto, w programach są rejestrowane: numer pokolenia, z którego pochodzi osobnik o najlepszym przystosowaniu; liczba generowanych w procesie ewolucyjnym rozwiązań, z którymi "nie radzi sobie" algorytm obliczeń; oraz w przypadku, gdy rozwiązanie zawiera graniczną wartość dowolnej zmiennej niezależnej – jej numer.

Ze względu na stochastyczny charakter przeszukiwania przestrzeni rozwiązań, uzyskiwane wyniki nie są powtarzalne. Z tego powodu wszystkie obliczenia były powtarzane dwudziestokrotnie, a ich wyniki były zapisywane w pliku dyskowym. Dodatkowy program rejestrował rozwiązania najlepsze i najgorsze oraz obliczał średnie przystosowanie, odchylenie standardowe oraz średni czas obliczeń.

Do eksperymentów obliczeniowych wybrano silnik jednofazowy o mocy znamionowej 1,1 kW i $2p = 4$. Przy obliczaniu funkcji przystosowania (1) przyjęto logarytmiczną funkcję kary wg wzoru (2c). Obliczenia wykonywano na komputerze z procesorem Pentium IV 2,55 GHz. Badano wpływ liczby osobników w populacji oraz liczby pokoleń na uzyskane wyniki, tzn. na średnią wartość przystosowania f_{av} , odchylenie standardowe σ oraz czas obliczeń t . Tabela 1 zawiera wyniki obliczeń uzyskane przy zastosowaniu strategii ewolucyjnej (1+1)-ES. W tabeli 2 – przedstawiono wyniki testów przy strategiach ewolucyjnych $(\mu+\lambda)$ -ES oraz (μ,λ) -ES. W strategiach tych zastosowano operator krzyżowania równomiernego oraz następujący schemat mutacji:

- w pierwszych 60 % maksymalnej liczby pokoleń T stosuje się mutację z rozkładem Cauchy'ego;
- pomiędzy 60 % a 80 % tej liczby, rozkład mutacji jest wybierany pomiędzy rozkładami Cauchy'ego i Gaussa z jednakowym prawdopodobieństwem;
- w ostatnich 20 % pokoleń stosuje się mutację z rozkładem Gaussa.

Tabela 3 zawiera wyniki obliczeń otrzymane przy zastosowaniu algorytmu genetycznego, opisanego w punkcie 3.2, przy zastosowaniu operatorów krzyżowania równomiernego oraz mutacji nierównomiernej.

Następnie, dla wybranych parametrów wykonano dla porównania obliczenia przy zastosowaniu krzyżowania proporcjonalnego w strategiach ewolucyjnych oraz algorytmie genetycznym. Wyniki przedstawiono w tab. 4 oraz 5. Tabela 6 zawiera porównanie wyników uzyskanych przy zastosowaniu w algorytmie genetycznym operatorów mutacji nierównomiernej oraz równomiernej.

W tabeli 7 przedstawiono składowe pięciu wybranych wektorów rozwiązań otrzymanych w wyniku uruchomień algorytmu genetycznego wg p. 3.2 oraz składowe wektora rozwiązania otrzymanego przy użyciu metody deterministycznej Hooke'a-Jeevesa z zastosowaniem restartu procedury optymalizacyjnej opisanego w pracy [7]. Wybrano wektory (spośród 20), dla których otrzymano jak najbardziej zbliżone wartości przystosowania f .

Eksperymenty wykazały, że przy zastosowaniu strategii ewolucyjnej (1+1)-ES, najlepsze rozwiązanie osiągnęto nie później w 3900-nym pokoleniu.

TABELA 1Strategia ewolucyjną (1+1)-ES. Wyniki obliczeń przy różnych liczbach pokoleń g

Liczba pokoleń g	Średnie przystosowanie f_{av}	Odchylenie standartowe σ [%]	Średni czas obliczeń [s]
$g = 500$	365,10	0,77	1,5
$g = 1000$	365,06	0,75	2,4
$g = 2000$	365,03	0,57	4,3
$g = 5000$	364,41	0,75	11,1

TABELA 2Strategie ewolucyjne $(\mu + \lambda)$ -ES oraz (μ, λ) -ES. Wyniki obliczeń przy różnych liczbach rodzi-
ców μ , potomków λ oraz różnych liczbach pokoleń g

Algorytm	$(\mu + \lambda)$ -ES			(μ, λ) -ES		
	Średnie przystosowanie f_{av}	Odchylenie standartowe σ [%]	Średni czas obliczeń [s]	Średnie przystosowanie f_{av}	Odchylenie standartowe σ [%]	Średni czas obliczeń [s]
Parametry						
$\mu = 50$ $\lambda = 100$ $g = 50$	362,94	0,40	8,3	367,37	0,80	8,2
$\mu = 100$ $\lambda = 200$ $g = 50$	361,39	0,31	16,7	365,57	0,61	16,2
$\mu = 100$ $\lambda = 200$ $g = 100$	359,78	0,22	32,8	364,93	0,39	31,6
$\mu = 50$ $\lambda = 300$ $g = 50$	359,10	0,30	24,5	361,95	0,35	23,9
$\mu = 50$ $\lambda = 300$ $g = 100$	359,57	0,18	48,7	360,82	0,18	48,2

TABELA 3

Algorytm genetyczny wg p. 3.2. Wyniki obliczeń przy różnych liczbach osobników w populacji N , liczbach rodziców p , oraz różnych liczbach pokoleń g

Parametry algorytmu	Średnie przystosowanie f_{av}	Odchylenie standardowe σ [%]	Średni czas obliczeń [s]
$N = 100; p = 50$ $g = 50$	366,69	0,61	4,8
$N = 100; p = 50$ $g = 200$	363,86	0,50	16,6
$N = 200; p = 100$ $g = 100$	364,7	0,35	17,4
$N = 200; p = 100$ $g = 200$	362,31	0,51	33,5
$N = 100; p = 80$ $g = 200$	368,77	0,68	27,0
$N = 100; p = 40$ $g = 200$	364,36	0,54	13,3
$N = 100; p = 50$ $g = 1000$	361,30	0,62	82,6
$N = 1000; p = 500$ $g = 100$	362,35	0,23	87,0

TABELA 4

Strategie ewolucyjne $(\mu+\lambda)$ -ES oraz (μ,λ) -ES. Porównanie wyników obliczeń przy zastosowaniu krzyżowania równomiernego oraz proporcjonalnego. $\mu = 100; \lambda = 200; g = 100$

Algorytm	$(\mu+\lambda)$ -ES			(μ,λ) -ES		
	Średnie przystosowanie f_{av}	Odchylenie standardowe σ [%]	Średni czas obliczeń [s]	Średnie przystosowanie f_{av}	Odchylenie standardowe σ [%]	Średni czas obliczeń [s]
Operator krzyżowania						
Równomierne	359,78	0,22	32,8	364,93	0,39	31,6
Proporcjonalne	365,66	0,40	32,7	372,15	0,58	34,6

TABELA 5

Algorytm genetyczny wg p. 3.2. Porównanie wyników obliczeń przy zastosowaniu krzyżowania równomiernego oraz proporcjonalnego $N = 100$; $p = 50$; $g = 200$

Rodzaj operatora krzyżowania	Średnie przystosowanie f_{av}	Odchylenie standardowe σ [%]	Średni czas obliczeń [s]
Równomierne	363,86	0,50	16,6
Proporcjonalne	369,16	0,75	17,3

TABELA 6

Algorytm genetyczny wg p. 3.2. Porównanie wyników obliczeń przy zastosowaniu mutacji równomiernej oraz nierównomiernej $N = 100$; $p = 50$; $g = 200$

Rodzaj operatora mutacji	Średnie przystosowanie f_{av}	Odchylenie standardowe σ [%]	Średni czas obliczeń [s]
Nierównomierna	363,86	0,50	16,6
Równomierna	365,15	0,48	17,1

TABELA 7

Porównanie składowych 5 wybranych wektorów rozwiązań uzyskanych w wyniku optymalizacji silnika $P_N = 1,1$ kW; $2p = 4$ przy użyciu algorytmu genetycznego dla $N = 200$; $p = 100$; $g = 200$, oraz w wyniku optymalizacji metodą deterministyczną Hooke'a-Jeevesa z restartem procedury optymalizacyjnej.

Zmienna	Jedn.	Rozwiązanie 1	Rozwiązanie 2	Rozwiązanie 3	Rozwiązanie 4	Rozwiązanie 5	Rozwiązanie H-J + restart
D	mm	85,19	85,65	86,18	85,26	85,74	85,57
l	mm	119,02	118,20	119,9	115,15	116,71	119,68
b_{Os}	mm	6,02	6,04	6,08	6,07	6,01	6,05
h_{ds}	mm	14,27	13,79	13,24	13,65	13,83	14,57
b_{Or}	mm	4,67	4,82	4,66	4,59	4,59	4,67
h_{dr}	mm	17,80	16,52	17,71	17,19	17,07	17,29
N_A	–	336,76	343,96	329,82	337,01	328,91	337,97
N_B	–	367,37	376,09	375,84	370,14	363,96	363,94
d_A	mm	0,501	0,520	0,489	0,504	0,476	0,514
d_B	mm	0,613	0,561	0,594	0,585	0,633	0,596
a_{er}	mm	7,60	7,96	7,15	7,43	7,77	7,22
C_p	μF	30,87	27,66	28,28	29,27	31,78	30,69
f	zł	360,56	360,89	360,96	362,59	362,79	363,50

Obliczenia optymalizacyjne wykonano także dla sześciu silników jedno-fazowych o mocach znamionowych: 1,1 kW; 4,0 kW; 5,5 kW; 7,5 kW oraz 11 kW. W tabelach 8a i 8b przedstawiono wyniki obliczeń przy użyciu różnych algorytmów ewolucyjnych. Dla porównania w ostatnich wierszach tablic podano wyniki obliczeń optymalizacyjnych przy użyciu metody deterministycznej Hooke'a-Jeevesa z zastosowaniem restartu procedury optymalizacyjnej [7].

TABELA 8a

Porównanie wyników obliczeń optymalizacyjnych przy użyciu różnych algorytmów ewolucyjnych oraz procedury deterministycznej dla 3 silników. Wartości górne – silnik o mocy znamionowej $P_N = 1,1$ kW; $2p = 4$; wartości środkowe – silnik o mocy znamionowej $P_N = 4,0$ kW; $2p = 4$; wartości dolne – silnik o mocy znamionowej $P_N = 5,5$ kW; $2p = 4$;

Algorytm	Parametry algorytmu	Najlepsze znalezione rozwiązanie	Średnia wartość przystosowania f_{av} z 20 uruchomień	Odchylenie standardowe σ [%]	Średni czas obliczeń t [s]
Strategia ewolucyjna ($I + I$)-ES	liczba pokoleń $g = 5000$	360,34	364,41	0,75	11,1
		960,34	970,23	0,68	14,2
		1155,97	1177,02	0,82	11,5
Strategia ewolucyjna ($\mu + \lambda$)-ES	$\mu = 100; \lambda = 200$ l. pokoleń $g = 100$	358,61	359,78	0,22	32,8
		952,37	955,72	0,32	36,8
		1135,61	1146,08	0,73	34,6
Strategia ewolucyjna (μ, λ)-ES	$\mu = 50; \lambda = 300$ l. pokoleń $g = 100$	359,68	360,82	0,18	48,2
		948,49	950,63	0,19	58,2
		1129,26	1132,57	0,21	54,5
Algorytm genetyczny wg 3.2	l. osob. $N = 200$ l. rodz. $p = 100$ l. pokoleń $g = 200$	358,92	362,31	0,51	33,5
		955,98	960,47	0,30	39,2
		1136,73	1150,67	0,78	36,3
Metoda deterministyczna	H-J + restart	363,50	–	–	11,1
		955,04	–	–	16,6
		1146,71	–	–	12,8

TABELA 8b

Porównanie wyników obliczeń optymalizacyjnych przy użyciu różnych algorytmów ewolucyjnych oraz procedury deterministycznej dla 3 silników. Wartości górne – silnik o mocy znamionowej $P_N = 5,5$ kW; $2p = 2$; wartości środkowe – silnik o mocy znamionowej $P_N = 7,5$ kW; $2p = 4$; wartości dolne – silnik o mocy znamionowej $P_N = 11,0$ kW; $2p = 4$;

Algorytm	Parametry algorytmu	Najlepsze znalezione rozwiązanie	Średnia wartość przystosowania f_{av} z 20 uruchomień	Odchylenie standardowe σ [%]	Średni czas obliczeń t [s]
Strategia ewolucyjna (1 + 1)-ES	liczba pokoleń $g = 5000$	1103,55	1123,43	0,83	12,5
		1467,95	1486,03	0,90	11,5
		1954,07	1972,84	0,91	32,5
Strategia ewolucyjna ($\mu + \lambda$)-ES	$\mu = 100$; $\lambda = 200$ l. pokoleń $g = 100$	1087,06	1091,25	0,27	39,9
		1460,32	1465,87	0,21	33,5
		1943,55	1948,31	0,20	39,7
Strategia ewolucyjna (μ, λ)-ES	$\mu = 50$; $\lambda = 300$ l. pokoleń $g = 100$	1074,52	1082,16	0,43	57,8
		1458,82	1468,75	0,29	54,4
		1943,12	1949,23	0,22	58,6
Algorytm genetyczny wg 3.2	l. osob. $N = 200$ l. rodz. $p = 100$ l. pokoleń $g = 200$	1097,48	1110,57	0,84	41,8
		1467,31	1475,78	0,32	37,8
		1952,96	1963,93	0,26	37,3
Metoda deterministyczna	H-J + restart	1094,60	–	–	16,8
		1478,84	–	–	28,7
		1957,71	–	–	16,4

5. WNIOSKI

Eksperymenty obliczeniowe wykazały, że właściwie użyte metody ewolucyjne mogą być efektywnie zastosowane do optymalizacji w procesie projektowania silników indukcyjnych jednofazowych. W szczególności wykazano że:

- optymalizacja przy zastosowaniu strategii ewolucyjnej (1+1)-SE, pomimo małego nakładu obliczeń jest mało efektywna i prowadzi do wyników gorszych od uzyskanych przy użyciu innych procedur ewolucyjnych oraz metod deterministycznych (tab. 1);
- zastosowanie strategii ewolucyjnych wieloosobniczych tzn. ($\mu + \lambda$)-ES oraz (μ, λ)-ES daje dobre wyniki przy odpowiednio dużej liczbie osobników w populacji (rodziców i potomków). Pomimo braku powtarzal-

ności, uzyskane wyniki wykazują bardzo małe odchylenia standardowe (tab. 2). Średnie wartości przystosowania (uzyskane z 20 uruchomień algorytmu) zależą od liczby rodziców oraz potomków, kosztem zwiększonego czasu obliczeń. Najkorzystniejszy stosunek liczby potomków do liczby rodziców wynosi dla strategii $(\mu+\lambda)$ -ES około 2, natomiast dla strategii (μ,λ) -ES jest większy i wynosi ok. 6. Strategia (μ,λ) -ES daje wyniki porównywalne z uzyskanymi przy strategii $(\mu+\lambda)$ -ES dopiero przy większych liczbach pokoleń. Biorąc pod uwagę czas obliczeń można przyjąć, że najbardziej odpowiednia do optymalizacji w procesie projektowania silników jednofazowych jest strategia $(\mu+\lambda)$ -ES, przy zastosowaniu krzyżowania równomiernego (tab. 4);

- algorytm genetyczny wg p. 3.2 (tab. 3) daje nieco gorsze wartości przystosowania oraz odchyłeń standardowych niż strategie ewolucyjne. Najlepsze wyniki uzyskano przy wymianie w każdym pokoleniu ok. 50 % osobników populacji. Dostateczna liczba osobników w populacji wynosi 200. Zwiększanie tej liczby prowadzi do poprawy wyników, jednak w nieco mniejszym stopniu, niż zwiększanie liczby pokoleń. Duża liczba osobników w populacji wpływa natomiast korzystnie na zmniejszenie odchylenia standardowego. Bardziej korzystne jest krzyżowanie równomierne (tab. 5) oraz mutacja nierównomierna, która odznacza się bardziej dokładnym poszukiwaniem pod koniec procesu ewolucyjnego (tab. 6).

Podczas wstępnych eksperymentów obliczeniowych zaobserwowano, że w algorytmie genetycznym, dla zapewnienia różnorodności populacji należy stosować minimalny nacisk selektywny poprzez ustalenie liczby uczestników turnieju $k = 2$ przy wyborze kopiowanej do następnego pokolenia części populacji o liczebności $N - p$. To samo dotyczy selekcji przy wyborze puli rodzicielskiej p . Stwierdzono również, że zastosowanie selekcji turniejowej (zarówno liniowej jak i nieliniowej) przy wyborze puli rodzicielskiej prowadzi do gorszych wyników zarówno pod względem uzyskiwanych wartości przystosowania jak i odchyłeń standardowych.

Przy zastosowaniu operatora mutacji nierównomiernej w algorytmie genetycznym, najlepsze wyniki uzyskano dla współczynnika nierównomierności mutacji $b = 1$ oraz $b = 2$, natomiast gorsze dla współczynnika $b = 0,5$.

Przy zastosowaniu strategii ewolucyjnych (μ,λ) -ES osobniki o najlepszym przystosowaniu pojawiały się już w niższych numerach pokoleń, a deterministyczny charakter selekcji powodował ich przetrwanie aż do końca procesu ewolucyjnego. Na przykład, podczas testowania operatorów krzyżowania (tab. 4), na 100 pokoleń najlepszy osobnik w strategii (μ,λ) -ES pojawiał się przeciętnie w 35 pokoleniu, podczas gdy w strategii $(\mu+\lambda)$ -ES – dopiero w 96 pokoleniu.

Wyniki przedstawione w tab. 7 wskazują, że w wyniku optymalizacji zarówno ewolucyjnej jak i deterministycznej uzyskuje się zbliżone wymiary obwodu magnetycznego silnika. Natomiast większe różnice występują w liczbach zwojów oraz pojemności kondensatora pracy.

Przedstawione w tabelach 8a i 8b wyniki obliczeń optymalizacyjnych sześciu silników jednofazowych wskazują, że stosowanie metod ewolucyjnych jest możliwe i uzasadnione w procesie projektowania silników indukcyjnych. Uzyskane wyniki są zbliżone, a często nawet lepsze niż przy zastosowaniu metod deterministycznych. Pomimo większego nakładu pracy obliczeniowej, algorytmy ewolucyjne w porównaniu z metodami deterministycznymi odznaczają się większą stabilnością oraz są niewrażliwe na właściwy dobór takich parametrów jak np. wstępna długość kroku lub parametry metody SUMT (wstępny współczynnik kary oraz współczynnik redukcji kary). Te właściwości umożliwiają korzystanie z nich przez użytkowników nie mających dużego doświadczenia. Ponadto, co jest szczególnie przydatne – nie jest konieczne rozpoczynanie obliczeń od dozwolonego, tzn. spełniającego wszystkie ograniczenia punktu startowego.

Przedstawione wyniki eksperymentów nie wyczerpują w pełni możliwości badawczych. Interesujące jest zbadanie wpływu wprowadzenia sukcesji elitarniej na działanie algorytmu genetycznego, a także większej liczby oraz odmian operatorów genetycznych, w tym wyspecjalizowanych operatorów związanych z charakterem zadania. Potrzeba badań zmierzających w tym kierunku jest podkreślana także w literaturze [13; 14]. Na przykład, bardzo interesujące jest zbadanie zachowania się algorytmów ewolucyjnych w zadaniach mieszanych, tzn. zawierających zarówno zmienne ciągłe jak i dyskretne, występujące w zadaniach optymalizacji maszyn elektrycznych oraz opracowanie specjalnych operatorów genetycznych przystosowanych do takich zadań. Algorytmy ewolucyjne dzięki zbieżności do optimum globalnego, rokują w tej dziedzinie większe możliwości niż metody deterministyczne. Wstępne wyniki są przedstawione w pracach [11; 18]. Szczególnie trudne zadanie występuje w silnikach jednofazowych o większej mocy znamionowej z uzwojeniami o nierozdzielonych strefach oraz optymalizowanym rozkładzie zwojów – łatwiejsze natomiast w silnikach trójfazowych z symetrycznymi uzwojeniami.

LITERATURA

1. Arabas J.: *Wykłady z algorytmów ewolucyjnych*. WNT, Warszawa, 2001.
2. Bäck T., Schwefel H.P.: *An Overview of Evolutionary Algorithms for Parameter Optimization*. *Evolutionary Computation*, Vol. 1, nr 1, 1993, s. 1-23.

3. Bianchi N., Bolognani S.: *Design optimization of electric motors by genetic algorithms*. IEE Proc.- Electr. Power Appl. Vol 145, nr 5, wrzesień 1998, s. 475-483.
4. Boldea I., Nasar S. A.: *The Induction Machine Handbook*. Boca Raton, London, New York, Washington, CRC Press, 2002.
5. Chellapilla K.: *Combining Mutation Operators in Evolutionary Programming*. I.E.E.E. Trans. on Evolutionary Computation, Vol. 2, nr 3, wrzesień. 1998, s. 91- 96.
6. Dąbrowski M.: *Projektowanie maszyn elektrycznych prądu przemiennego*. Warszawa, WNT 1994.
7. Dąbrowski M., Rudeński A.: *Porównanie efektywności deterministycznych procedur optymalizacji w zadaniach syntezy silników indukcyjnych jednofazowych*. Proc. of XL International Symposium on Electrical Machines SME-2004, s. 107-115.
8. Eiben A.E., Hinterding R., Michalewicz Z.: *Parameter Control in Evolutionary Algorithms*. I.E.E.E. Trans. on Evolutionary Computation, Vol. 3, nr 2, lipiec 1999, s. 124-141.
9. Goldberg D.E.: *Algorytmy genetyczne i ich zastosowania*. WNT, Warszawa, 1995.
10. Kowalski K.: *Zastosowanie algorytmów genetycznych w procesie optymalizacji przetworników elektromagnetycznych*. Materiały VII Konferencji Naukowo-Technicznej: „Zastosowania komputerów w elektrotechnice”, Poznań / Kiekrz, kwiecień 2002, s. 701-704.
11. Göll Ö., Wieczorek J.P.: *A Comparison of Deterministic and Stochastic Methods in Induction Motor Design*. Proc. International Conference on Electrical Machines (ICEM 98), Sept. 1998, Istambul, s. 1472-1476.
12. Kozieł S., Michalewicz Z.: *Evolutionary Algorithms, Homomorphous Mapping, and Constrained Parameter Optimization*. Evolutionary Computation, nr 7, 1999, s. 19-44.
13. Michalewicz Z.: *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. WNT, Warszawa, 1996.
14. Michalewicz Z., Schoenauer M.: *Evolutionary Algorithms for Constrained Parameter Optimization Problems*. Evolutionary Computation, 1996, nr 1, s. 1-37.
15. Rudeński A.: *Algorytmizacja oraz programowanie w zadaniach syntezy optymalnych silników indukcyjnych jednofazowych*. Rozprawa doktorska, Instytut Elektrotechniki, Warszawa 2003 r.
16. Rudeński A.: *Zastosowanie metod deterministycznych oraz ewolucyjnych w zadaniach optymalizacji szczególnej i ogólnej maszyn elektrycznych*. Proc. of XL International Symposium on Electrical Machines SME-2004, s. 200-204.
17. Teixeira S., Pacheco X.: *Delphi 4: Vademecum profesjonalisty*. Tom 1 i 2, Gliwice, Wydawnictwo Helion, 1999.
18. Wieczorek J.P., Göll Ö., Michalewicz Z.: *An Evolutionary Algorithm for the Optimal Design of Induction Motor*. I.E.E.E. Trans. on Magnetics, Vol. 34, nr 6, listopad 1998, s. 3882-3887.
19. Wurtz F., Richomme M., Bignon J., Sabonnadiere J.C.: *A Few Results for using Genetic Algorithms in the Design of Electrical Machines*. I.E.E.E. Trans. of Magnetics, Vol. 33, nr 2, marzec 1997.
20. Yao X., Liu Y., Lin G.: *Evolutionary Programming Made Faster*. I.E.E.E. Trans. on Evolutionary Computation, Vol. 3, nr 2, lipiec 1999, s. 82-102.

Rękopis dostarczono, dnia 15.10.2004 r.

Opiniował: prof. dr hab. inż. Jan Sikora

APPLICATION OF NON-DETERMINISTIC METHODS TO OPTIMIZATION OF SINGLE-PHASE INDUCTION MOTORS

Andrzej RUDEŃSKI

ABSTRACT *In the paper an application of the non-deterministic methods to single-phase induction motors optimization is presented. Three kinds of evolution strategies and a genetic algorithm with partial population exchange and floating point representation of genes in chromosomes has been tested. On the basis of calculation experiments the influence of specified number of individuals in a population of the number of generations and genetic operators onto the obtained results, i. e.: the average value of the fitness function, standard deviation and computation time is examined. Optimization was executed with 12 independent variables and 9 constraints. The continuous variation of all independent variables has been assumed. As objective function the total cost of active materials and exploitation cost of the motor in given operation time has been considered. Optimization results for six single-phase induction motors with run and starting capacitors, and motor rated power from 1.1 kW to 11 kW, using the tested methods, and conclusions drawn from the computation experiments are presented.*

Dr inż. Andrzej Rudeński urodził się w roku 1946. Dyplom mgr inż. uzyskał w 1970 r. na Wydziale Elektrycznym Politechniki Warszawskiej i w tym też roku rozpoczął pracę w Instytucie Elektrotechniki. Początkowo zajmował się zagadnieniami ciepłno-wentylacyjnymi, a potem przez wiele lat (do końca 1982 r.) zagadnieniami zjawisk pasożytniczych w silnikach indukcyjnych, będąc autorem i współautorem wielu prac naukowo-badawczych oraz kilku publikacji. W latach 1983...2000 pracował kolejno w WSK PZL Warszawa II, Zakładzie Badawczo-Rozwojowym Dźwigów Osobowych i Zakładach Remontowych Energetyki Warszawa, zajmując się głównie problematyką maszyn elektrycznych oraz zastosowaniem technik informacyjnych do obliczeń. W 2000 r. powrócił do pracy w Instytucie Elektrotechniki, gdzie zajmuje się zagadnieniami analizy i syntezy silników indukcyjnych jednofazowych oraz optymalizacji. W roku 2003 uzyskał stopień doktora nauk technicznych. Obecnie zajmuje się zastosowaniem metod ewolucyjnych w projektowaniu oraz optymalizacji elektromechanicznych przetworników energii. Od roku 2000 jest autorem i współautorem 19 publikacji.

