

1. Algorytm poszukiwania harmonii

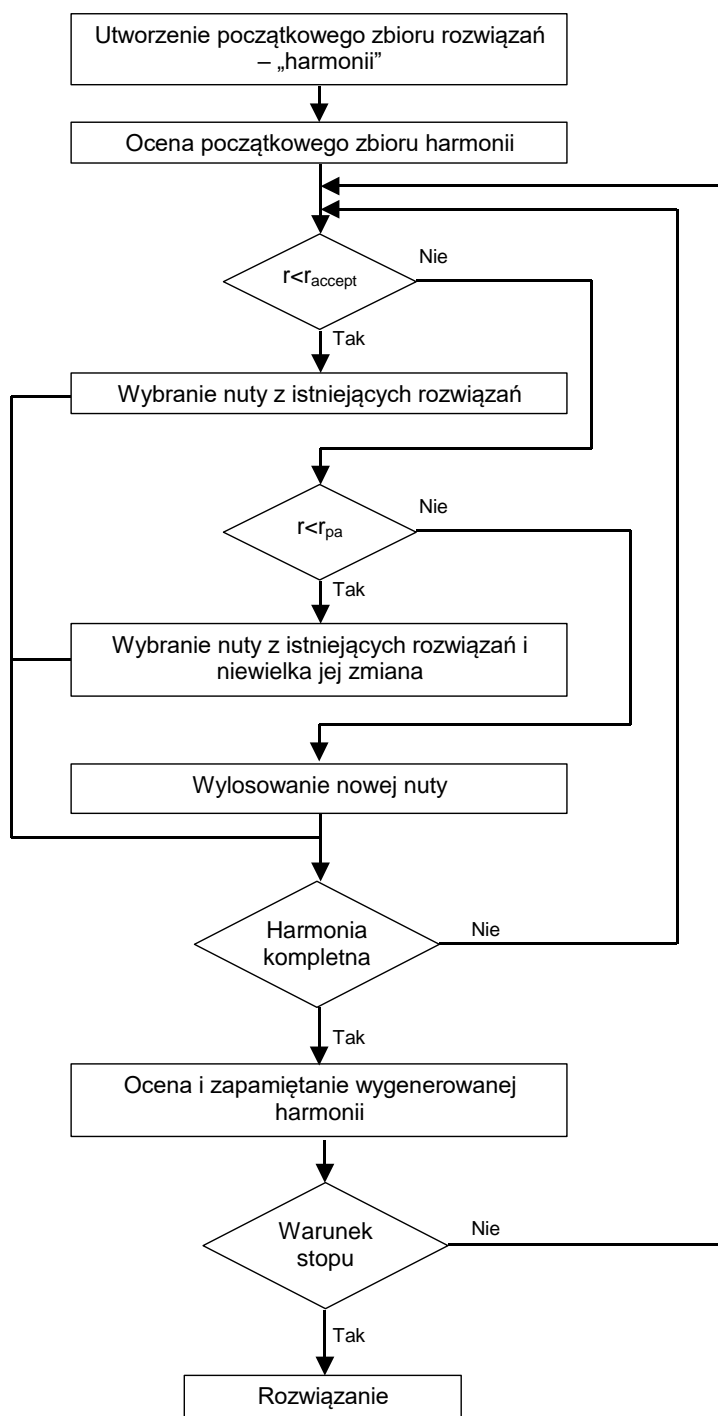
Charakterystyka

Algorytmy poszukiwania harmonii (Harmony Search – HS) są stosunkowo młodą grupą metod. Za pierwszą publikację opisującą ich ideę wskazuje się dzieło Geem'a i współpracowników z 2001 roku. Obecnie istnieje już wiele odmian algorytmów HS i wiele układów hybrydowych, w których wykorzystuje się ich własności oraz oczywiście powstało wiele ich implementacji, zarówno w zadaniach kombinatorycznych jak i zadaniach o zmiennych ciągłych.

Muzyka, aby była akceptowalna przez większość słuchaczy wychowanych w danym kręgu kulturowym, musi spełniać pewne warunki. Głównym warunkiem jest osiągnięcie odpowiedniej harmonii, czyli odległości pomiędzy poszczególnymi dźwiękami współbrzmiącymi i po sobie następującymi. Poszukiwanie harmonii jest pewnego rodzaju procesem optymalizacyjnym. W przypadku algorytmów HS za kluczową inspirację należy uznać wyidealizowany proces improwizacji, w którym muzyk grając kolejny fragment utworu może: zagrać fragment znanego utworu ze swej pamięci lub zagrać melodię podobną do znanego utworu bądź też zagrać nową lub przypadkową melodię.

Na rysunku 3.19 przedstawiono ogólną postać algorytmu poszukiwania harmonii. Rozpoczyna się on od utworzenia, zwykle w sposób losowy, początkowego zbioru rozwiązań czyli początkowego zbioru „harmonii” (melodii). Metaforą nut w poszczególnych rozwiązaniach – harmoniach – są elementy rozwiązania, stanowiące – na przykładzie optymalizacji funkcji wielu zmiennych ciągłych – wartości poszczególnych zmiennych tej funkcji (czyli liczby rzeczywiste). Każde z rozwiązań początkowych poddaje się ocenie. Następnie przechodzi się do głównej pętli algorytmu, w której generowane są nowe rozwiązania. Nowe rozwiązanie jest budowane „nuta po nucie”. Oznacza to, że dla każdej nuty określa się wartość albo pobierając ją z któregoś z istniejących rozwiązań, ewentualnie dodatkowo nieznacznie ją modyfikując, albo losując zupełnie nową wartość. Decyzję, którą z możliwości przyjąć, podejmuje się na podstawie porównania wylosowanej wartości r z przedziału $(0,1)$. Jeżeli r jest mniejsze od wartości prawdopodobieństwa r_{accept} zaakceptowania nuty zapisanej w pamięci, wówczas pobiera się taką nutę z losowo wybranej harmonii. W przeciwnym przypadku sprawdza się, czy wartość r jest mniejsza od prawdopodobieństwa r_{pa} małej zmiany (dostrojenia w pewnym zakresie) istniejącej nuty z losowo wybranej harmonii. Jeżeli i ten warunek nie jest spełniony,

wówczas losuje się nową wartość dla danej nuty z przedziału zmienności danej zmiennej. Gdy skonstruowana jest w ten sposób cała harmonia (rozwiązanie), dokonuje się jej oceny. Jeżeli nowe rozwiązanie jest lepsze od najgorszego przechowywanego rozwiązania, to ono je zastępuje. Ostatnim elementem jest sprawdzenie warunku zatrzymania, który zwykle sprowadza się do sprawdzenia liczby wszystkich prób utworzenia nowego rozwiązania.



Rysunek 3.19 Ogólna postać algorytmu poszukiwania harmonii

Istotnymi parametrami wpływającymi na działanie algorytmu HS są, oprócz wymienionych prawdopodobieństw, także liczba przechowywanych w pamięci harmonii oraz maksymalna możliwa zmiana istniejącej w pamięci nuty. Wpływają one, podobnie jak parametry innych metod optymalizacyjnych, na uzyskiwane przez nie rezultaty. Badacze proponują także wersje algorytmów HS ze zmiennymi automatycznie dostrajającymi się parametrami.

Zastosowania

Pierwsze zastosowania metod bazujących na poszukiwaniu harmonii pojawiły się w elektroenergetyce pod koniec pierwszej dekady XXI wieku. Skutkuje to bardzo niewielką liczbą publikacji, poświęconych zasadniczo kilku obszarom. Metoda ta traktowana jest jako uniwersalna metoda optymalizująca, co jest zgodne z jej konstrukcją.

Optymalizacja

Poszukiwanie harmonii posiada, co jest naturalne, najwięcej zastosowań w optymalizacji. Zbiór występujących tu zadań jest podobny jak dla pozostałych metod optymalizacyjnych:

1. Optymalizacja ekonomicznego/ekologicznego rozdziału obciążeń (włącznie z harmonogramowaniem)
2. Optymalizacja rozptywu mocy czynnej
3. Optymalizacja konfiguracji sieci
4. Optymalizacja struktury sieci
5. Optymalizacja rozwoju sieci
6. Optymalizacja lokalizacji źródeł rozproszonych w sieci
7. Optymalizacja lokalizacji urządzeń FACTS w celu minimalizacji strat
8. Optymalizacja lokalizacji kompensatorów mocy biernej w sieci

Identyfikacja

Niewielka liczba zastosowań HS w identyfikacji skupia się wokół zagadnień klasyfikacyjnych. W pewien sposób wynika to ze specyfiki metody, która bazuje na przeszukiwaniu przestrzeni rozwiązań, korzystając z jednej strony z podobieństw pomiędzy poszczególnymi wzorcami, a z drugiej strony z ich różnorodności. Za przykłady takich zagadnień mogą posłużyć:

1. Analiza zachowania sieci w przypadku wyłączeń
2. Klasyfikacja jednoczesnych zaburzeń jakości energii.
3. Identyfikacja zwarć w liniach przesyłowych

Sterowanie

Sposób wykorzystania HS w sterowaniu nie odbiega od wykorzystania innych metod optymalizacyjnych. Zastosowano je do stabilizacji systemu elektroenergetycznego, koordynacji nastaw zabezpieczeń nadprądowych i optymalizacji lokalizacji urządzeń FACTS w celu poprawy stabilności systemu

2. Funkcje testowe

- Funkcja Ackley'a

$$f(x) = -20 \exp \left[-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right] - \exp \left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right] + 20 + e$$

gdzie: $n = 1, 2, \dots$; and $-32.768 \leq x_i \leq 32.768$ for $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnęte jest dla $x_* = (0, 0, \dots, 0)$.

- Funkcja Rosenbrocka:

$$f(x) = \sum_{i=1}^{n-1} \left[(x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2 \right]$$

gdzie: $n = 1, 2, \dots$; and $-30.0 \leq x_i \leq 30.0$ for $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnęte jest dla $x_* = (1, 1, \dots, 1)$.

- Funkcja DeJonga:

$$f(x) = \sum_{i=1}^n x_i^2$$

gdzie: $n = 1, 2, \dots$; oraz $-100.0 \leq x_i \leq 100.0$ dla $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnęte jest dla $x_* = (0, 0, \dots, 0)$.

- Funkcja Griewanka

$$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (13)$$

gdzie: $n = 1, 2, \dots$; oraz $-600.0 \leq x_i \leq 600.0$ dla $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnęte jest dla $x_* = (0, 0, \dots, 0)$.

- Funkcja Rastrigina :

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \text{ gdzie: } n = 1, 2, \dots; \text{ oraz } -5.12 \leq x_i \leq 5.12 \text{ for } i$$

$= 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnęte jest dla $x_* = (0, 0, \dots, 0)$.

- Funkcja Schwefela:

$$f(x) = -\sum_{i=1}^n x_i \sin(\sqrt{|x_i|})$$

gdzie: $n = 1, 2, \dots$; oraz $-500 \leq x_i \leq 500$ dla $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* \approx -418.9829 \cdot n$ osiągane jest dla $x_* = (420.9687, 420.9687, \dots, 420.9687)$.

W celu zapewnienia dodatniości funkcji dla wszystkich numerów wymiarów oraz zmiennych, następująca modyfikacja funkcji powinna podlegać zastosowaniu.

$$f(x) = -\frac{1}{n} \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}) + 500$$

Minimum globalne funkcji $f_* \approx 81.0171$ występuje w tym samym punkcie, co dla funkcji oryginalnej, tj. $x_* = (420.9687, 420.9687, \dots, 420.9687)$.

Celem optymalizacji każdej funkcji jest jej minimalizacja. Algorytmy były jednak budowane tak, aby maksymalizować funkcję przystosowania. Konieczne było więc przejście między funkcją celu a przystosowania, dokonywane docelowo za pomocą:

$$F_{fitness} = \frac{1}{F_{obj} + 1}$$

3. Przebieg ćwiczenia

- Oprócz funkcji domyślnej zespół wybiera 3 inne funkcje testowe
- W pliku prototype.goi należy ustawić odpowiednią dla danej funkcji testowej liczbę zmiennych oraz zakresy zmiennych wejściowych
- Wybór kombinacji parametrów wejściowych w pliku auto_HS.bxt, umożliwiającym zbadanie wpływu każdej zmiennej na efekty symulacji. Kombinacje pozostają takie same dla każdej funkcji testowej.
- Porównanie wyników na podstawie plików report_HS_auto.txt, w tym ocena jakości rozwiązania znalezionego oraz analiza zbieżności