

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Algorytmy heurystyczne

Specyfikacja zadania

Mateusz Dziwulski, Jakub Szczepański

Warszawa, 2017

1. Treść projektu

1.1. Zmodyfikowany algorytm ewolucji różnicowej

W ramach projektu należy przygotować implementację zmodyfikowanego algorytmu ewolucji różnicowej korzystającego z zapisanej historii populacji oraz punktu środkowego (metoda DES — dokładny opis w artykule). Kod trzeba przygotować w formie biblioteki, a przeprowadzić jego testy na wybranym zestawie zadań optymalizacji. Za wybranie projektu można otrzymać dodatkowe 5 pkt.

2. Specyfikacja zadania

2.1. Wyjaśnienie algorytmu

Algorytm DES w uproszczeniu można uznać za algorytm pośredni do CMA-ES oraz ewolucji różnicowej. Podobnie jak CMA-ES algorytm ten przemieszcza populację w kierunku wyznaczonym ze średniej populacji do średniej pewnej grupy najlepszych osobników, wykorzystuje też różnicę między wylosowanymi osobnikami z populacji podobnie do ewolucji różnicowej.

W skrócie algorytm można opisać w następujących punktach:

1. Inicjalizacja parametrów i populacji początkowej.
2. Obliczenie średniej z populacji.
3. Sortowanie osobników populacji zgodnie z funkcją celu.
4. Obliczenie średniej z μ najlepszych osobników.
5. Aktualizacja każdego z osobników w oparciu o obliczone średnie, parametry $\{c, F\}$, pewne zmienne losowe o danych rozkładach oraz zadany szum.
6. W wypadku nieosiągnięcia warunku stopu powrót do punktu drugiego.

W wypadku optymalizacji z ograniczeniami są one dodawane poprzez przypisanie punktom z poza obszaru poszukiwań największej napotkanej dotychczas wewnątrz obszaru wartości powiększonej o sumę kwadratów odległości od ograniczeń dla wymiarów, w których ograniczenia zostały przekroczone.

2.2. Opis zagadnienia

Zagadnienie polega na implementacji zmodyfikowanego algorytmu ewolucji różnicowej. Algorytm zostanie wstępnie zaimplementowany w języku R, jednakże w razie trudnych do przewidzenia na wstępie trudności możliwe, że końcowa implementacja zostanie wykonana w języku C++ bądź Python i jedynie skompilowana do pakietu R. Z pakietu dostępna będzie pojedyncza funkcja, która najprawdopodobniej będzie przyjmowała następujące parametry:

- funkcję celu q
- rozmiar populacji λ
- wielkość potomstwa μ
- współczynnik skalowania F - podany algorytm sugeruje wartość $\frac{1}{\sqrt{2}}$, możliwe jest więc jego pominięcie
- współczynnik migracji punktu środkowego populacji c
- długość horyzontu H
- natężenie szumu ϵ
- opcjonalne ograniczenia przestrzeni przeszukiwania

Warunek stopu będzie obliczany tak jak w artykule zgodnie z równaniem (14) i ogólnym opisem z sekcji V - kiedy zmiany w populacji będą głównie powodowane głównie współczynnikiem losowym, albo po wykonaniu pewnej maksymalnej liczbie ewaluacji funkcji celu.

Ograniczenia zostaną wprowadzone zgodnie z równaniami (10–11) z sekcji IV artykułu.

Dla zaprezentowania wyników oraz na potrzeby testów wybrany został zestaw benchmarkowy CEC 2017 (single objective bound constrained case). Taki wybór pozwoli nam na porównanie

otrzymanych wyników z zawartymi w artykule oraz ewentualnie innymi publikacjami zgłoszonymi do konkursu i bardziej poinformowaną dyskusję otrzymanych wyników. Ze względu na szeroki zakres dostępnych testów ich zakres może zostać ograniczony czasem obliczeń, ale przewidujemy taką możliwość jedynie w ostateczności.

2.3. Planowane testy

Na zestawie benchmarkowym CEC 2017 planowane jest sprawdzenie zarówno jakości otrzymywanych przez algorytm rozwiązań jak również uzyskiwanych przez niego czasów. Najprawdopodobniej jako interfejs do zestawu testów wykorzystany zostanie pakiet mgr. Jagodzińskiego `cec2017`.

Zestaw ten zawiera większość standardowych funkcji takich jak funkcje Rosebrocka, Zahkaro-va czy Levy'ego. Na potrzeby testów planowane jest wykorzystanie ok. 10 z dostępnych funkcji.