Porównanie metod ewolucyjnych

Piotr Kostrzeński, Natan Orzechowski

Grudzień 2020

1 Wstęp

Przedmiotem niniejszego artykułu jest porównanie dwóch różnicowych metod ewolucyjnych: klasycznej (DE - Differential Evolution) oraz ulepszonej (BBDE). Ulepszona wersja od klasycznej różni się z perspektywy użytkownika ilością wprowadzanych parametrów. Jednym z problemów klasycznej wersji jest to, że należy przeprowadzić wiele eksperymentów dla różnych wartości parametrów. Oczywiście, są ogólne wartości które powinny być wystarczająco dobre, ale nie będą one najlepsze. Także poszukiwanie najlepszego rozwiązania wymaga czasochłonnej kalibracji.

Alternatywna wersja algorytmu eliminuje te problemy poprzez kilka zabiegów wymienionych w podanym nam artykule. Jest to podejście czarnoskrzynkowe, które zakłada, że nie mamy żadnych powodów żeby wstępnie wybrać kierunek w który rozwiązania powinny zmierzać. Inaczej mówiąc, nic nie wiemy o wewnętrznym działaniu procesu, dla którego próbujemy znaleźć najlepsze rozwiązanie.

1.1 Metodologia

Zbadano zachowanie obu algorytmów dla:

- Dwóch różnych funkcji testowych 10 wymiarowych
- Różnych liczebności populacji
- W przypadku algorytmu klasycznego:
    1. Wartości mutacji $mutation = 0.7$[1]
    2. Z prawdopodobieństwem krzyżowania $cross\_probability \in \{0, 0.5\}$[2]

Dla każdej funkcji testowej, zbadano działanie algorytmu klasycznego w schemacie **rand/1/bin** oraz **rand/1/exp**, aby porównywać możliwie najlepszą wersję

[1] Wyłonionej w trakcie testów jako optymalna wartość parametru

[2] Typowo stosuje się 50% prawdopodobieństwo; zamiarem autorów była weryfikacja konieczności stosowania krzyżowania w procesie reprodukcji

algorytmu klasycznego do jego ulepszonej wersji. W celu zapewnienia miarodajnych wyników, każdy scenariusz został poddany testom 25 razy. Wszystkie warianty korzystały z takiej samej populacji początkowej. Każdy algorytm miał na znalezienie optimum 2000 iteracji.

W eksperymentach użyliśmy dwóch wersji algorytmu BBDE, według wartości parametru zastępującego parametry F oraz K (według symboli użytych w artykule). Pierwsza to główna wersja zaproponowana przez autora, według której wynik losowania wartości tego parametru ma medianę 1.0:

$$F, K \sim exp(n_{i,g}(0, 1))$$

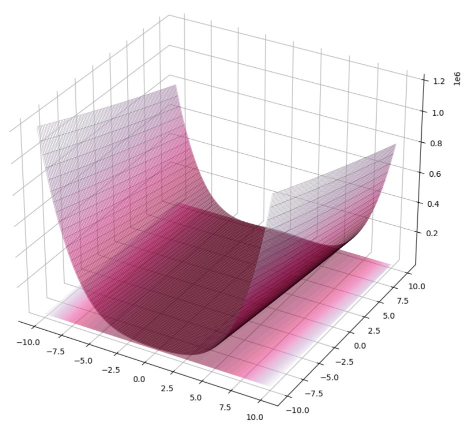
oraz druga wersja (alternatywna, jak podaje autor artykułu) ze średnią 1.0:

$$F, K \sim exp(n_{i,g}(0, 1) - 0.5)$$

Dwie wersje wynikają z tego, że dla pierwszej uzyskiwane wyniki dla niektórych eksperymentów były względnie słabe. Sprawdzając drugą, uzyskano w wielu przypadkach lepsze wyniki, dlatego wzięto ją pod uwagę.

1.2 Funkcje testowe: reprezentacja graficzna, charakterystyka

1.2.1 Funkcja Rosenbrocka

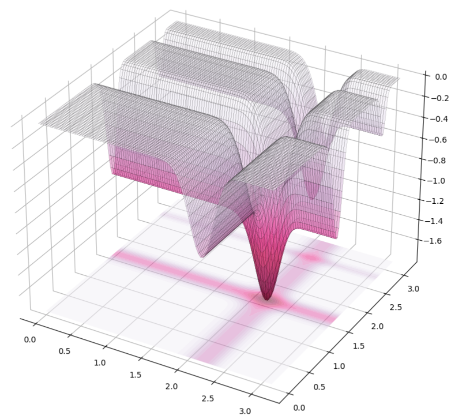


Rysunek 1: Funkcja Rosenbrocka

- Typowa dziedzina: $[-5; 10]$
- Minimum globalne: $f(x^*) = 0$

Niewielkie zróżnicowanie wartości funkcji wokół minimum globalnego sprawia, iż eksploatacja dla algorytmu staje się problematyczna.

1.2.2 Funkcja Michalewicza



Rysunek 2: Funkcja Michalewicza

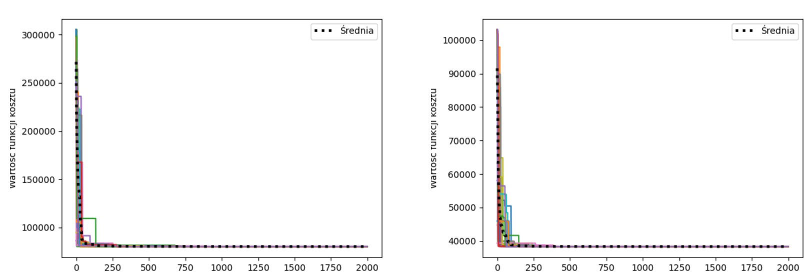
- Typowa dziedzina: $[0; \pi]$
- Minimum globalne dla $d = 10 : f(x^*) = -9.66015$

Głębokie, ostre doliny oraz płaska charakterystyka poza minimami powoduje, że algorytm sprawnie eksplorujący może nie wykryć minimum.

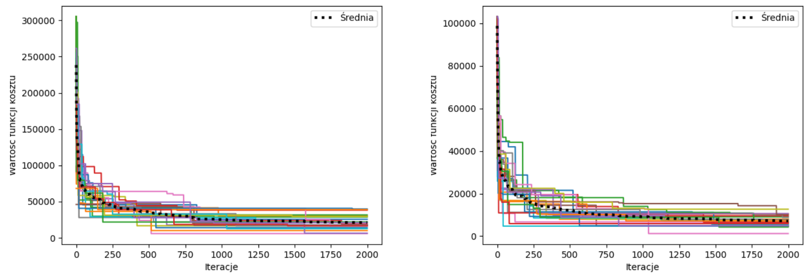
2 Wyniki badań

2.1 Funkcja Rosenbrocka

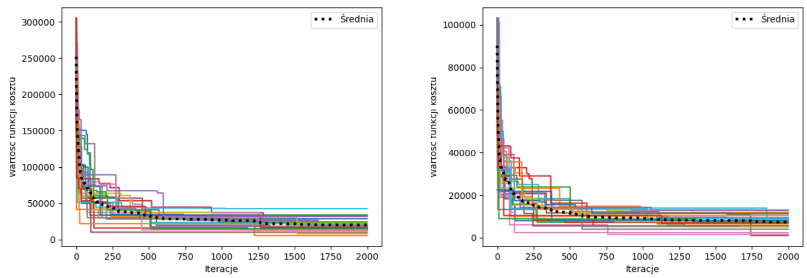
Według opisanej metodologii przeprowadzono eksperymenty, których rezultaty przedstawiono na następnej stronie. Pominęliśmy dla oszczędności miejsca wykresy populacji pośrednich i przedstawiliśmy skrajne - kolejno 20 oraz 120.



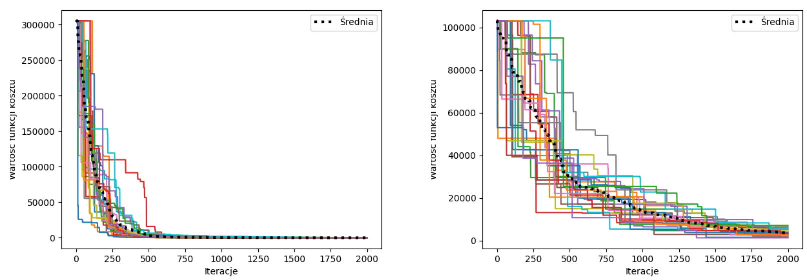
Rysunek 1: *DE rand/1*



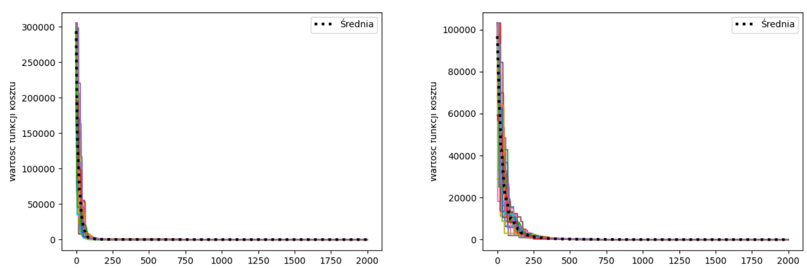
Rysunek 2: *DE rand/1/bin*



Rysunek 3: *DE rand/1/exp*



Rysunek 4: *BBDE normalne*



Rysunek 5: *BBDE alternatywne*

Dla uzupełnienia widoku na sprawę, dodajemy uśrednioniony wynik ostateczny dla każdego z wariantów:

Algorytm	Populacja - 20	Populacja - 120
DE rand/1/bin bez krzyżowania	80077.354	38310.211
DE rand/1/bin z krzyżowaniem	21230.487	7138.492
DE rand/1/exp	20087.274	7415.341
BBDE normalne	46.143	3713.091
BBDE alternatywne	25.143	1.679

Pierwszym, widocznym od razu wnioskiem jest to, że algorytm BBDE bez dyskusji wygrał z algorytmem DE w każdej wariacji. Wartość błędu dla obu wersji algorytmu BBDE była zdecydowanie mniejsza od algorytmu DE. Jedynie dla populacji 120, normalny algorytm BBDE zwrócił względnie podobny błąd (lecz dalej mniejszy).

Ogólnie, większość algorytmów prowadziła do uzyskania lepszego lub gorszego minimum. Jeżeli chodzi o dokładność, to tendencją jest wzrost dokładności wraz ze wzrostem populacji. Jedynym wyjątkiem jest tutaj wersja **BBDE normalna**. Najprawdopodobniej wynika to z tego, że ta wersja jest lepsza w eksploracji niżeli eksploatacji minimum, dla większej populacji rozwiązanie bardziej oscylowało wokół minimum niż je osiągało. Miałoby to sens, gdyż w tej wersji parametr mutacji jest średnio większy niż w alternatywnej. Stąd prawdopodobnie, dla tak płaskiej funkcji nie udało się dotrzeć do tego najważniejszego punktu.

Kolejnym spostrzeżeniem jest to, że o ile wraz ze wzrostem populacji rosła dokładność uzyskiwanych wyników, tak szybkość zbiegania do tego minimum była mniejsza. Dla mniejszych populacji to słabsze minimum osiągano bardzo szybko, ale trudno było z niego osiągnąć najbardziej optymalny wynik. Jest to wniosek o którym warto pamiętać, gdyż taką wersję można by użyć w scenariuszu posiadania słabych mocy obliczeniowych przy wynikach, które nie muszą być bardzo dokładne, ale wystarczające (szczególnie że dla mniejszych populacji fizyczny czas wykonywania każdej iteracji jest po prostu mniejszy).

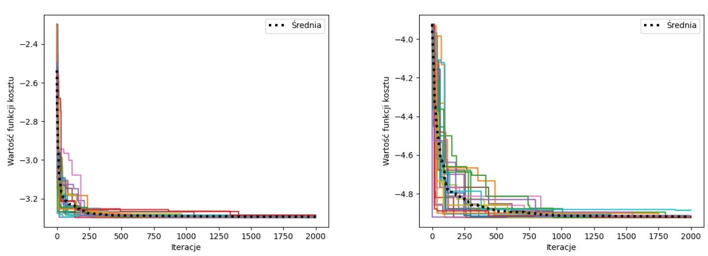
Wariacje algorytmu DE zawiodły dosyć mocno, prawdopodobnie dlatego, że algorytm DE jest ogólnie słabszy w eksploatacji tak płaskiego minimum. Najlepszym rezultatem może się ostatecznie pochwalić algorytm **BBDE alternatywny** (możliwe powody względem BBDE normalnego przytoczono wcześniej). Uzyskano dla niego najlepszą dokładność, szybkość zbiegania do minimum oraz powtarzalność przebiegów eksperymentów.

2.2 Funkcja Michalewicza

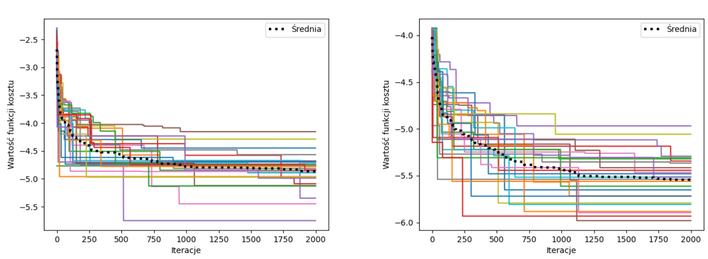
Algorytm klasyczny wykazał wzrost skuteczności wraz ze wzrostem populacji dla każdego schematu. Wniosek ten jest intuicyjny, ponieważ większa liczba testowych wektorów, zwiększa szanse na znalezienie takiego wektora, który znajdzie się w okolicy doliny funkcji. Ponadto, wersje z krzyżowaniem oraz bez dowodzą, iż w procesie poszukiwania wąskiej hiperprzestrzeni, w której znajduje się optimum, proces krzyżowania **jest korzystny**. Jest to spowodowane faktem, iż w ramach każdej iteracji, istnieje szansa na znalezienie odrobinę lepszych wektorów, niż te wyznaczone losowo, a w konsekwencji populacja szybciej wypełnia się wektorami z otoczenia minimum. Schematy rand/1/bin oraz rand/1/exp wykazują zbliżone zdolności eksploatacji minimum. Algorytm ulepszony, w przypadku nor wykazał podobną tendencję - wzrost populacji oznacza lepsze wyniki. Wersja alternatywna natomiast **była tym bliżej optimum globalnego, im mniejsza była populacja**. Wyniki osiągnięte przez ten wariant algorytmu są znacznie lepsze od zarówno wersji normalnej algorytmu BBDE, jak i klasycznej wersji algorytmu DE z krzyżowaniem. Nie znaleziono logicznego uzasadnienia tego zjawiska.

Poniżej zamieszczono minima znalezione przez poszczególne warianty algorytmów oraz wykresy odpowiadające poszczególnym wierszom oraz kolumnom tabeli wyników. Przedstawiają one wartości osiągane przez algorytmy w kolejnych iteracjach dla 25 prób oraz ich średnią.

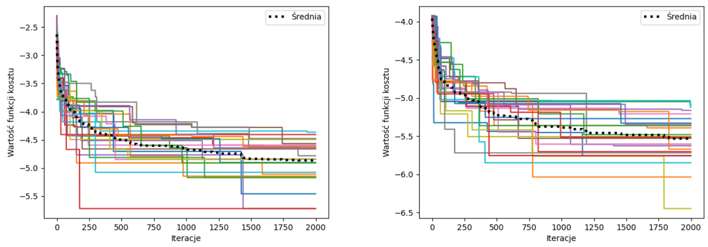
Algorytm	**Populacja - 20**	**Populacja - 120**
DE rand/1	-3.294	-4.919
DE rand/1/bin	-4.865	-5.543
DE rand/1/exp	-4.866	-5.537
BBDE normalne	-3.908	-4.329
BBDE alternatywne	-6.4	-4.329



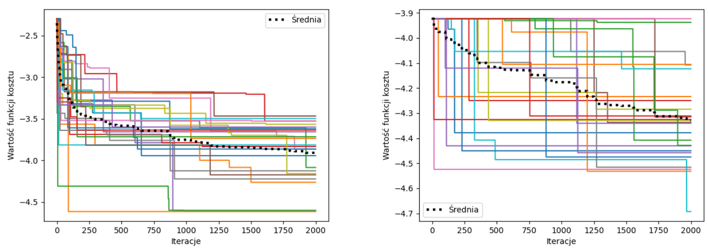
Rysunek 1: *DE rand/1*



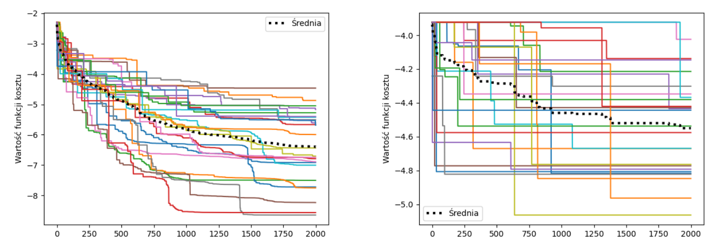
Rysunek 2: *DE rand/1/bin*



Rysunek 3: *DE rand/1/exp*



Rysunek 4: *BBDE*



Rysunek 5: *BBDE alternatywna*

3 Podsumowanie

Patrząc na zagregowane wyniki badań, można wysnuć konkluzję, iż zmodyfikowana wersja algorytmu, zwłaszcza z nastawami alternatywnymi, jest lepsza zarówno w eksploracji jak i eksploatacji[1].

W przypadku funkcji Rosenbrocka, klasyczny DE nie był w stanie znaleźć minimum globalnego - ostateczna wartość funkcji kosztu była rzędu tysięcy przy minimum globalnym wynoszącym 0. W przypadku funkcji Michalewicza, różnica również była znaczna na korzyść algorytmu zmodyfikowanego, zwłaszcza

[1] W przypadku testowanych funkcji.

w wersji alternatywnej. Ten wariant osiągnął wartości bardzo bliskie minimum globalnego - podczas jednego w przebiegów aż ok. -8.7. Wersja klasyczna z krzyżowaniem osiągnęła również dobre wyniki, bo zbliżone do wersji normalnej algorytmu BBDE.

Ostatecznie ocenia się nowszą wersję algorytmu za lepszą jeżeli chodzi o efektywność, zarówno dla funkcji płaskich jak i tych o silnie zmiennym przebiegu. Nowsza wersja jest także lepsza jeżeli chodzi o wygodę użytkowania ze względu na mniejszą liczbę parametrów do skalibrowania.

Dziękujemy za przeczytanie
Piotr Kostrzeński
Natan Orzechowski