

Porównanie algorytmów optymalizacji: genetycznego i rojowego

Paulina Machcińska 191677

Wstęp

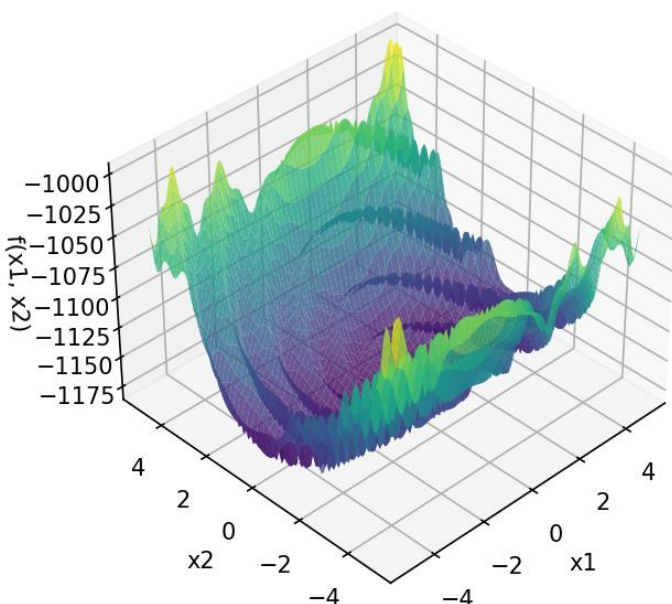
Celem mojej pracy jest porównanie dwóch algorytmów optymalizacyjnych - genetycznego i rojowego. Obie metody mają na celu znalezienie minimum funkcji skrótu. Chcę sprawdzić, który z tych algorytmów lepiej sobie poradzi z tym zadaniem. W tym celu przeprowadzę kilka testów dla różnych parametrów, a następnie porównam wyniki. Oczywiście, nie wszystkie funkcje mają takie same właściwości i nie zawsze jedna metoda będzie skuteczniejsza od drugiej. Niemniej jednak, moim celem jest znalezienie ogólnych wniosków dotyczących efektywności tych dwóch algorytmów w kontekście optymalizacji funkcji skrótu.

Algorytmy

Jako funkcję skrótu wykorzystałam wzór pochodzący z algorytmu genetycznego, który znalazłam w materiałach na stronie enauczanie.pl.

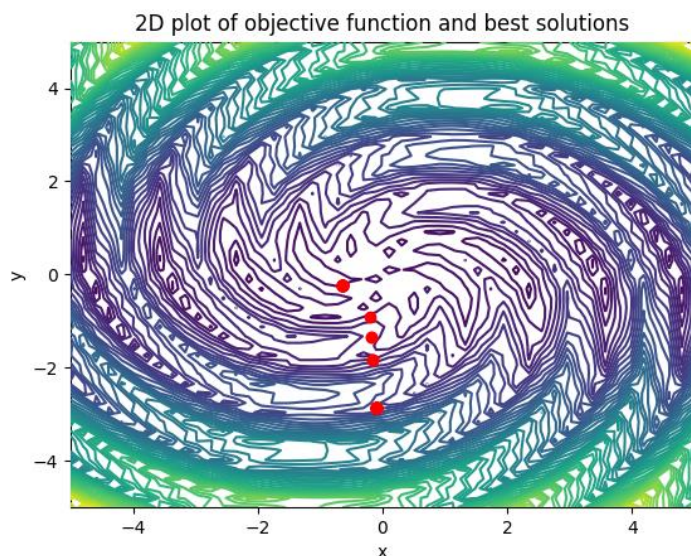
```
# Definicja funkcji celu
def funea(x1, x2):
    z = np.sqrt(x1**2 + x2**2)
    y = x1**2 + 5*x2**2 + 5*z*(np.sin(6*np.arctan2(x2, x1) + 5*z))**3 \
        - 100*g(x1+3, x2+3) - 125*g(x1-2, x2-2) - np.exp(x1*.005)
    return y
```

Dzięki wizualizacji możemy zauważyć, że funkcja ma zarówno minimum globalne, jak i wiele lokalnych minimów. W związku z tym wyniki uzyskane przez algorytmy optymalizacyjne mogą się różnić w zależności od tego, w którym z minimów algorytm się zatrzyma. W niektórych przypadkach algorytmy mogą nie "wpadać" do minimum globalnego, a zamiast tego zatrzymywać się w jednym z lokalnych minimów i nie być w stanie z niego wyjść. W takich sytuacjach ważne jest, aby wybrać odpowiedni algorytm i dobór parametrów tak, aby uniknąć wpadnięcia w lokalne minimum.

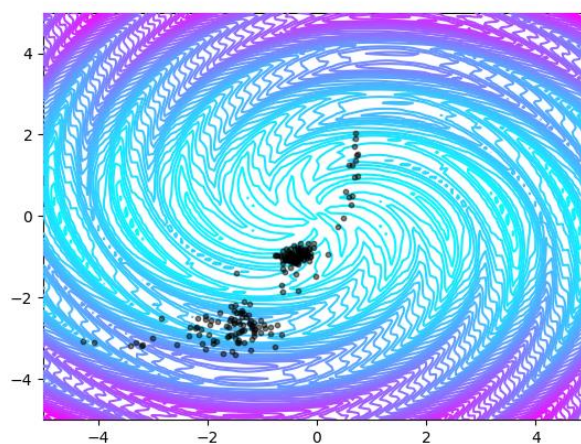
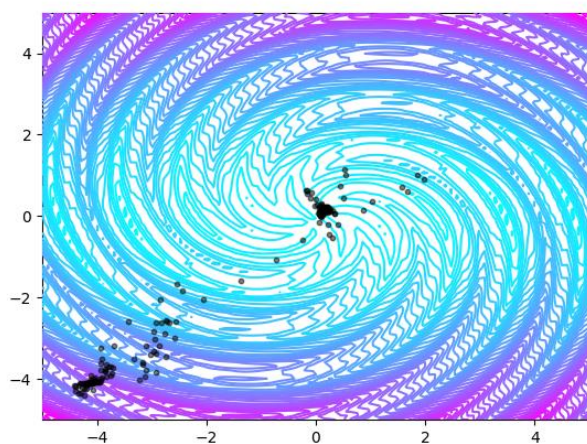


Porównując algorytmy genetyczny i rojowy w celu optymalizacji danej funkcji, należy podkreślić, że oba algorytmy zostały uruchomione z gotowymi implementacjami i dobrze dobranymi parametrami, co pozwoliło uzyskać najlepsze możliwe wyniki dla wybranej funkcji.

Algorytm genetyczny radził sobie dobrze w znalezieniu globalnego minimum funkcji w ciągu 100 generacji. Najlepsze osobniki zbliżały się stopniowo do minimum globalnego, ale w niektórych przypadkach zatrzymywały się w lokalnych minimach. Mimo, że było wiele populacji, nie po każdej był zmieniany najlepszy osobnik, dlatego widzimy tylko 5 czerwonych kropek, czyli najlepszych rezultatów jakie uzyskał algorytm.



Algorytm rojowy również dobrze radził sobie z znajdowaniem globalnego minimum funkcji, a czarne kropki na wykresie pokazują, jak przemieszczał się po funkcji rój. Algorytm rojowy wykazywał większe natężenia w miejscach, gdzie wynik był wystarczająco dobry, co oznacza, że trudno było przejść do lepszego rozwiązania.



Podsumowanie

Można określić, że algorytm rojowy i genetyczny są skutecznymi narzędziami optymalizacji, ale ich skuteczność zależy od wielu czynników, takich jak funkcja celu, liczba iteracji, rozmiar populacji, liczba wymiarów, itp.

W przypadku analizowanej funkcji i wybranych parametrów, algorytm rojowy okazał się bardziej skuteczny, osiągając lepsze wyniki i krótszy czas wykonywania w porównaniu z algorytmem genetycznym. Niemniej jednak, w innych sytuacjach, algorytm genetyczny może być bardziej odpowiedni do optymalizacji. Podczas wielu testów, algorytm genetyczny potrafił znaleźć lepsze dopasowanie niż algorytm rojowy.

Warto podkreślić, że czas wykonania algorytmu rojowego jest nawet o 100 razy mniejszy, co może być szczególnie korzystne w przypadku dużych zbiorów danych lub złożonych funkcji celu.

```
GA execution time: 14.157567977905273

Best solution found:
[-0.21399904 -0.8096609 ]
Fitness: 921.7746971328761
Number of iterations: 100
2023-04-19 20:07:18,787 - pyswarms.single.global_best - INFO - Optimize for 100 iters with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|100/100, best_cost=0.000585
2023-04-19 20:07:18,845 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0005851555365167807, best pos: [3.42976517 0.0685244 ]
PSO execution time: 0.05905914306640625
Best solution found:
x1 = 3.429765
x2 = 0.068524
Fitness = 1708.947344
```

Podsumowując, algorytm rojowy okazał się być szybszy niż algorytm genetyczny w tym konkretnym przypadku. Algorytm rojowy jest bardziej zoptymalizowany pod kątem szukania optimum w wielowymiarowej przestrzeni i lepiej radzi sobie z unikaniem lokalnych minimów. W niektórych przypadkach algorytm genetyczny był w stanie znaleźć lepsze rozwiązania, jednakże wymagał więcej czasu i zasobów obliczeniowych.

Wiele jednak zależy od czynników, takich jak kształt funkcji celu, ilości wymiarów, czasu wykonania itp. Po przeanalizowaniu problemu można dojść do wniosku, że najlepszym rozwiązaniem może okazać się zastosowanie hybrydowych algorytmów, łączących w sobie różne techniki optymalizacji.