# WSI – sprawozdanie

Pryimak Andrii-Stepan 336173

# Algorytm ewolucyjny

## Wprowadzenie

Celem było zbadanie klasycznego algorytmu ewolucyjnego w optymalizacji funkcji. Algorytm miał za zadanie dostosować się do problemów wysokowymiarowych (10D), z ograniczeniami w przestrzeni poszukiwań do przedziału [-100, 100]. W eksperymencie zastosowano selekcję turniejową i sukcesję generacyjną, z budżetem 10000 ewaluacji funkcji celu. Badane były efekty liczby osobników w populacji oraz siły mutacji (sigma), a także wpływ zwiększonego budżetu na jakość rozwiązań.

## Analiza wyników

F2 (sigma = 0.5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Populacja | Średnia | Najlepszy | Najgorszy | Odchylenia std |
| 2 | 739.08 | 213.57 | 4733.05 | 938.49 |
| 4 | 717.10 | 207.98 | 2502.16 | 591.45 |
| 8 | 4029.48 | 230.58 | 20976.13 | 5833.66 |
| 16 | 2040503.60 | 573.126 | 27108177.76 | 6050426.52 |
| 32 | 7648472285.85 | 5280.004 | 122078178316.71 | 25570635533.33 |
| 64 | 89760202156.16 | 278436.696 | 523195856429.74 | 151845496971.11 |
| 128 | 870011497775.02 | 1241304.684 | 8426349071074.13 | 1984391980674.18 |

Dla funkcji najpierw zrobiłem że najlepszy to najlepszy z ostatniej generacji i to nie działało poprawni bo na końcu zwracana najlepsza wartość często była strącana a wyniki były od 200 do 20000

Z tabeli widać że po populacji >8 wyniki się pogorszają wynika to z tego że dla takiej wielkiej liczby osobników liczba ewaluacji jest jeśli zwiększyć liczbę ewaluacji w 5 raz to wynik jest bardzo lepszy. Ale dla liczby osobników 8 wynik polepszył się jeszcze bardziej

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Populacja(F2) | Średnia | Najlepszy | Najgorszy | Odchylenia std |
| 8 | 256.297 | 200.838 | 344.627 | 46.07 |
| 128 | 2202792.419 | 822.183 | 40258573.52 | 7831736.43 |

Zimina sigmy (populacja = 8)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sigma | Średnia | Najlepszy | Najgorszy | Odchylenia std |
| 0.2 | 200.290 | 200.013 | 205.836 | 1.13 |
| 0.5 | 237.34 | 200.806 | 347.894 | 40.19 |
| 1 | 894.41 | 203.371 | 3732.417 | 990.84 |
| 2 | 8181.54 | 1092.31 | 31693.26 | 8237.091 |
| 3 | 38943.43 | 3603.82 | 149573.71 | 31859.509 |
| 5 | 440012.48 | 31783.53 | 2422884.29 | 607652.90 |

Widać że przy zwiększeniu sigmy wyniki się pogarszają bo sigma decyduje o długości wektora mutacji co powoduje że obszar sprawdzenia się zwiększa i odpowiednio algorytm nie może dokładni trafić w punk a jest tylko o obszarze tego punku bo ma wielki włócznik eksploracji. Ale jeśli sigma jest zbyt mała np. 0.01 to liczba ewaluacji jest niewystarczająca żeby dojść do tego punktu

F13 (sigma=0,5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Populacja | Średnia | Najlepszy | Najgorszy | Odchylenia std |
| 4 | 14240.820 | 2316.685 | 31735.50 | 9727.72 |
| 8 | 14862.482 | 2542.980 | 56916.59 | 11833.93 |
| 16 | 15160.565 | 2329.226 | 48172.98 | 12046.195 |
| 32 | 14384.51 | 1994.586 | 35812.98 | 11791.409 |
| 64 | 18183.248 | 1986.078 | 68293.85 | 16705.551 |

Można zauważyć że wynik przy zwiększeniu do 64 się polepsza tym czasem średni wynik populacji oraz odchylenia jest najlepsze dla populacji 32

populacja = 32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sigma | Średnia | Najlepszy | Najgorszy | Odchylenia std |
| 0,2 | 12258.28 | 2092.027 | 45663.54 | 11531.53 |
| 0,5 | 14384.51 | 1994.586 | 35812.98 | 11791.40 |
| 1 | 14501.93 | 2496.455 | 31993.19 | 10017.07 |
| 2 | 18583.38 | 1973.829 | 54700.70 | 10752.47 |
| 3 | 22517.57 | 4539.613 | 54596.92 | 10660.31 |

Ty podobnie ja dla f2 z zwiększeniem sigmy wyniki się pogarszają ale dla 0.5 najlepsze bo f13 ma dużo różnych obrazów poziomice wyglądają tak jak duża liczba takich kółek i dlatego potrzebna trochę większa exproloracji żeby algorytm mógł przeskoczyć z jednego obszaru do innego

Dla tej funkcji zwiększenia liczby ewaluacji nic nie zmienia jedyne że istnie 5 krotnie większa szansa przypadkiem trafić w punk minimalny populacja = 32 sigma =0.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15468.23 | 2276.760 | 52538.61 | 12767.00 |  |

### Podsumowanie

Sukcesja generacyjna, choć jest metodą prostą i pozwala na skuteczną eksplorację przestrzeni rozwiązań, niesie ze sobą pewne ryzyko – może prowadzić do utraty najlepszego osobnika z poprzedniej generacji. W związku z tym, w algorytmach optymalizacyjnych, sukcesja elitarna często okazuje się lepsza, ponieważ umożliwia zachowanie najlepszych rozwiązań między kolejnymi generacjami, co sprzyja stabilnemu doskonaleniu jakości rozwiązań.

|  |  |
| --- | --- |
| Sukcesja elitarna(F2) | Sukcesja generyczna(F2) |
| A graph with numbers and lines  Description automatically generated |  |

Z drugiej strony, sukcesja generacyjna ma lepsze właściwości eksploracyjne, co może być pomocne na początkowych etapach poszukiwania, gdy algorytm przeszukuje szeroki obszar przestrzeni rozwiązań. Jednak zauważyłem, że z perspektywy znalezienia globalnego minimum, klasyczny algorytm ewolucyjny z mutacją i selekcją turniejową może być niewystarczający. Dobrym rozwiązaniem mogłoby być wprowadzenie dynamicznej zmiany parametrów, takich jak zmniejszanie wartości sigmy w miarę upływu czasu, co pozwoliłoby na początkową eksplorację szerokiego zakresu przestrzeni rozwiązań, a następnie bardziej precyzyjne poszukiwanie w obszarach potencjalnych minimów.

Alternatywnie, można by zastosować strategię polegającą na identyfikacji obiecujących obszarów za pomocą algorytmu ewolucyjnego, a następnie, po osiągnięciu wystarczająco dobrych wyników, przejść do dokładnego poszukiwania za pomocą metod gradientowych, które mogłyby działać w oparciu o wcześniej znalezione punkty startowe. Taka hybryda algorytmu ewolucyjnego i metody gradientowej mogłaby pozwolić na precyzyjne osiągnięcie optimum globalnego, wykorzystując zalety obu podejść.