

WSI – CWICZENIA 3

Zadanie

Zaimplementować klasyczny algorytm ewolucyjny z selekcją turniejową i sukcesją generacyjną, bez krzyżowania. Dostępny budżet to 10000 ewaluacji funkcji celu. Optymalizujemy funkcje numer 2 i 13 z CEC 2017 w 10 wymiarach. Ograniczenia kostkowe przestrzeni to -100, 100.

Uwaga

- Ponieważ algorytmy ewolucyjne wykorzystują losowość, nie wolno wyciągać wniosków na podstawie wyników pojedynczego uruchomienia. Należy porównywać średnie z co najmniej 25 uruchomień. W celu uzyskania pełnego obrazu, w tabelach z wynikami oprócz średniej zamieszcza się odchylenie standardowe, oraz najlepszy i najgorszy ze znalezionych wyników.
- Nie należy podawać zbyt wielu miejsc po przecinku.
- Liczbę iteracji wyliczmy w kodzie: $tmax = \text{budżet} / \mu$, gdzie μ to aktualnie przyjęta liczba osobników w populacji.
- Oczywiście optymalne wartości parametrów są zależne od siebie i analizując parametry oddzielnie możemy ich nie znaleźć. Na tym ćwiczeniu nie szukamy jednak optymalnych parametrów, szukamy rozsądnych, wyrabiamy podstawowe intuicje. Na początek przyjmujemy rozsądne wartości parametrów, np. $\sigma = 3$. Na podstawie eksperymentów znajdujemy najlepszy dla danego zadania rozmiar populacji (np. sprawdzamy kolejne potęgi 2). Mając rozmiar populacji badamy wpływ siły mutacji, tj. wykonujemy eksperymenty i tworzymy kolejną tabelę.

Pytania (odpowiedzi proszę umieścić w dokumencie tekstowym)

Zbadać wpływ:

- liczby osobników w populacji,
- siły mutacji,

na jakość uzyskanych rozwiązań.

Po dobraniu parametrów należy sprawdzić czy wyniki poprawią się po użyciu pięciokrotnie większego budżetu.

Jakie wnioski można wyciągnąć na podstawie wyników tego ćwiczenia?

Badanie wpływu liczby osobników w populacji

Na potrzeby ćwiczenia zaimplementowano algorytm ewolucyjny bez krzyżowania, z reprodukcją turniejową oraz sukcesją generacyjną. Działanie algorytmu testowano w pierwszym etapie dla dwóch funkcji z CEC2017 (odpowiednio f2 i f13) dla budżetu o wysokości 10000 z stałym parametrem SIGMA=3 oraz dla różnych rozmiarów populacji. Postanowiono wziąć pod obserwację kolejne potęgi liczby 2. Z uwagi na losowość, wnioski wyciągano na podstawie zagregowanych statystyk dla 30 uruchomień każdej kolejnej wartości parametru „mu”.

Wyniki przedstawiono w tabelach poniżej:

Eksperyment 1.1 : badanie wpływu rozmiaru populacji dla funkcji f2

	Maks. iteracji	Rozmiar populacji	Sigma	Minimum	Srednia	Od. Stand.	Maksimum
0	5000	mu=2	3	8,75177e+08	1,45009e+13	7,24205e+13	4,04164e+14
1	2500	mu=4	3	1,67477e+07	2,73477e+10	1,36129e+11	7,59939e+11
2	1250	mu=8	3	1,35509e+06	1,97241e+08	3,06262e+08	1,52925e+09
3	625	mu=16	3	806577	2,1466e+07	2,52269e+07	1,10628e+08
4	312	mu=32	3	157734	4,47611e+06	4,96749e+06	2,08493e+07
5	156	mu=64	3	50223,8	7,84496e+06	2,07841e+07	1,17939e+08
6	78	mu=128	3	20701,3	1,28075e+07	2,65386e+07	1,17641e+08

Jak można zauważyć, najlepiej dla funkcji f2 sprawdził się rozmiar populacji mu=32. Świadczy o tym najniższa średnia ($4,47611 \cdot 10^6$) a także najniższa wartość maksymalna ($2,0849 \cdot 10^7$). Wartość minimalna 157734 pozostawia jednakże wiele do życzenia, dla parametrów mu równych 64 i 128 można było uzyskać wyniki w tej statystyce niższe nawet 3- i 7-krotnie, aczkolwiek średnia dla tych parametrów były większe i tym samym dalej rozwiązania optymalnego. Trochę odmienną sytuację ukazują wyniki dla funkcji f13.

Eksperyment 1.2 : badanie wpływu rozmiaru populacji dla funkcji f13

	Maks. iteracji	Rozmiar populacji	Sigma	Minimum	Srednia	Od. Stand.	Maksimum
0	5000	mu=2	3	3,39943e+08	1,65363e+12	4,91488e+12	2,74248e+13
1	2500	mu=4	3	4,94895e+06	3,09095e+09	6,6232e+09	3,37417e+10
2	1250	mu=8	3	3,23419e+06	8,49016e+07	1,01297e+08	4,92785e+08
3	625	mu=16	3	45361,8	1,45286e+07	1,87543e+07	8,20638e+07
4	312	mu=32	3	14448,8	6,16907e+06	1,02497e+07	5,44752e+07
5	156	mu=64	3	40048	4,0604e+06	6,52839e+06	3,46358e+07
6	78	mu=128	3	45352,5	2,07432e+08	9,85961e+08	5,50951e+09

W przypadku funkcji f13, najlepiej sprawdził się rozmiar populacji o wysokości mu=64. Średnia dla tego parametru była najmniejsza ($4,0604 \cdot 10^6$), a co więcej biorąc pod uwagę wartość Minimum i Maksimum, były to również wartości najbliższe optimum (minimum globalnemu) spośród ich odpowiedników dla pozostałych rozmiarów populacji.

W związku z powyższym można wysnuć wniosek, że zbyt duże populacje, ale też i zbyt małe, pogarszają jakość otrzymanego rozwiązania. Dla wybadanych rozmiarów populacji przeprowadzono następnie badanie wpływu siły mutacji poprzez manipulowanie wartością parametru Sigma.

Badanie wpływu siły mutacji

Eksperyment 2.1 : badanie wpływu siły mutacji dla funkcji f2

	Maks. iteracji	Rozmiar populacji	Sigma	Minimum	Srednia	Od. Stand.	Maksimum
0	312	mu=32	0,1	3,26277e+08	4,90546e+13	1,18796e+14	4,36868e+14
1	312	mu=32	1	3608,44	1,04054e+06	2,88508e+06	1,40384e+07
2	312	mu=32	3	175033	5,2833e+06	5,9581e+06	2,81056e+07
3	312	mu=32	5	891710	2,92945e+08	3,14947e+08	1,36101e+09
4	312	mu=32	10	5,64207e+07	7,93829e+10	1,13759e+11	4,07571e+11
5	312	mu=32	50	4,72552e+14	1,13312e+17	3,2152e+17	1,8064e+18
6	312	mu=32	100	1,0318e+17	1,62808e+19	2,99139e+19	1,29317e+20

W przypadku funkcji f2, algorytm ewolucyjny z stałą populacją mu=32 najlepsze wyniki uzyskał dla parametru sigma = 1. Średnia, minimum oraz maksimum były stosunkowo najbliższe minimum globalnemu.

Eksperyment 2.2 : badanie wpływu siły mutacji dla funkcji f13

	Maks. iteracji	Rozmiar populacji	Sigma	Minimum	Srednia	Od. Stand.	Maksimum
0	156	mu=64	0,1	5,85799e+08	6,48015e+13	1,88198e+14	9,63695e+14
1	156	mu=64	1	6092,18	2,33064e+09	7,24432e+09	3,74461e+10
2	156	mu=64	3	181034	5,5697e+06	6,74443e+06	3,12429e+07
3	156	mu=64	5	3,99443e+06	9,37521e+07	1,4494e+08	8,01515e+08
4	156	mu=64	10	1,1562e+08	3,3849e+10	7,54671e+10	3,49813e+11
5	156	mu=64	50	1,25337e+14	8,22647e+15	1,30057e+16	4,97763e+16
6	156	mu=64	100	1,29319e+16	4,90194e+18	9,39618e+18	4,59158e+19

Jednakże zmiana parametru Sigma dla funkcji f13 nie przyniosła poprawy rezultatów. Zmiana wartości na wyższą lub niższą dawała wartości funkcji celu średnio wyższe niż dla sigma = 3.

Sprawdzanie 5-krotnie większego budżetu

Postanowiono zatem przetestować jak na otrzymane w ramach testów kombinacje rozmiaru populacji i parametru Sigma wpłynie wykorzystanie pięciokrotnie wyższego budżetu do przeprowadzenia oceny. Wyniki były bardzo zadowalające, uzyskano rezultaty o zdecydowanie lepszej jakości. Przedstawiają to zestawienia w tabelach poniżej.

Eksperyment 3.1 : Funkcja f2 z budżetem 50 000, $\mu=32$ i $\sigma=1$

	Maks. iteracji	Rozmiar populacji	Sigma	Minimum	Srednia	Od. Stand.	Maksimum
0	1562	$\mu=32$	1	661,00	3710,93	2236,85	10406,10
1	312	$\mu=32$	1	3608,44	1,04054e+06	2,88508e+06	1,40384e+07

Zdecydowana poprawa wyników na skutek zwiększenia budżetu (maksymalnej liczby iteracji) dla obydwu funkcji. Im dłużej pozwolono na ewoluowanie rozwiązania tym wyniki były bliżej minimum (optimum).

Eksperyment 3.2 : Funkcja f13 z budżetem 50 000, $\mu=64$ i $\sigma=3$

	Maks. iteracji	Rozmiar populacji	Sigma	Minimum	Srednia	Od. Stand.	Maksimum
0	781	$\mu=64$	3	26456,4	1,76024e+06	1,35699e+06	5,05364e+06
2	156	$\mu=64$	3	181034	5,5697e+06	6,74443e+06	3,12429e+07

Wnioski końcowe:

Na podstawie powyższego ćwiczenia można wysnuć następujące wnioski:

- algorytm ewolucyjny stanowi bardzo prosty w implementacji i skuteczny w stosowaniu algorytm,
- stosując go należy wykazać się doskonałą umiejętnością doboru rozmiaru populacji, nie za dużej i nie za małej.
- wysokość parametru sigma, który zwiększa wpływ mutacji powinna być rozsądnie mała i raczej unikana. Zalecane jest bardziej konfigurowanie rozmiaru populacji i dostępnego budżetu,
- używanie krzyżowania nie jest wymagane.