

Wprowadzenie do sztucznej inteligencji - ćwiczenie 2

Zadanie

Zaimplementować klasyczny algorytm ewolucyjny bez krzyżowania z selekcją turniejową i sukcesją elitarną. Dostępny budżet to 10000 ewaluacji funkcji celu. Optymalizujemy funkcję numer 4 i 5 z CEC 2017 w 10 wymiarach. Ograniczenia kosztowe 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
- W tabeli zmieniamy tylko jedną rzecz na raz. Przykładowo, chcąc zbadać wpływ zmiany rozmiaru populacji z 20 na 40 osobników należy stworzyć tabelę:

Parametr	min	śr	std	max
mu=20	1,23	2,34	0,11	3,45
mu=40	1,03	2,49	0,91	3,51

- Chcąc dalej zbadać wpływ siły mutacji przyjmujemy powyżej znaleziony rozmiar populacji i robimy kolejną tabelę. 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.
- Liczbę iteracji wyliczmy w kodzie: tmax=budżet/mu, gdzie mu to aktualnie przyjęta liczba osobników w populacji

Pytania (odpowiedzi proszę umieścić w dokumencie tekstowym)

Zbadać wpływ:

- Siły mutacji
- Rozmiaru elity
- Liczby osobników w populacji

na jakość uzyskanych rozwiązań.

Jakie wnioski można wyciągnąć na podstawie wyników tego ćwiczenia?

Wyniki:

Domyślne wartości:
rozmiar populacji = 20
rozmiar elity = 1
siła mutacji = 1
liczba iteracji dla każdego pomiaru = 50

F4

population size	minimum	maximum	average	standard deviation
5	400.16	468.45	406.35	9.16
10	400.64	470.72	409.42	15.27
20	400.66	473.48	411.5	18.63
50	400.95	560.55	425.91	34.01
100	401.02	723.73	481.54	64.24

elite size	minimum	maximum	average	standard deviation
0	400.8	477.53	410.02	16.26
1	400.27	486.33	417.64	27.1
2	401.71	477.47	411.06	16.36
3	400.29	492.99	412.85	22.49
5	400.45	491.86	417.14	25.42
10	400.3	472.88	409.33	16.02

sigma value	minimum	maximum	average	standard deviation
0.2	443.75	994.3	604.13	131.6
0.5	401.53	489.73	420.61	26.87
1	401.45	469.23	410.72	16.56
2	402.08	495.03	410	15.31
5	406.33	490.05	412.37	14.84
10	410.68	446.55	417.04	5.13

F5

population size	minimum	maximum	average	standard deviation
5	547.34	796.18	640.35	49
10	525.64	707.78	618.93	41.26
20	529.84	667.12	592.18	28.74
50	514.36	617.07	568.09	26.44
100	518.09	596.36	558.43	17.31

elite size	minimum	maximum	average	standard deviation
0	530.13	671.86	592.98	30.64
1	529.67	700.88	590.93	36.67
2	531.75	697.05	590.46	35.23
3	532.58	650.42	592.09	30.41
5	530.41	662.94	591.6	31.75
10	541.51	657.66	595.56	26.9

sigma value	minimum	maximum	average	standard deviation
0.2	521.94	673.17	590.19	33.89
0.5	544.06	648.6	592.28	27.86
1	530.04	650.28	587.7	28.66
2	532.02	628.42	580.28	23.56
5	519.06	549.6	533.36	7.41
10	528.45	554.04	540.91	5.44

Obserwacje i wnioski:

W przypadku funkcji f4 im większa populacja, tym gorsze wyniki co oznacza, że dla tej funkcji ważniejsza była liczba iteracji, niż zróżnicowanie populacji. Może to oznaczać, że aby zoptymalizować tą funkcję musieliśmy położyć nacisk na eksploatację. Potwierdzają to także wyniki pomiarów siły mutacji – dla wartości sigmy 10 otrzymujemy minimum 410, średnią zbliżoną do tego wyniku i małe odchylenie standardowe. Oznacza to, że algorytm często znajdował obszar, gdzie znajduje się minimum, ale nie potrafił go dokładnie wyznaczyć

Dla funkcji f5 nie udało mi się uzyskać optimum, prawdopodobnie dlatego, że funkcja ta ma dużo minimów lokalnych i ciężko jest znaleźć te właściwe. Potwierdza to fakt, że wartości parametrów sprzyjające eksploracji (większa populacja, większa siła mutacji) daje średnio lepsze wyniki.

Dla obu funkcji wielkość elity nie miała wielkiego znaczenia.