WSI, Ćwiczenie 2

Michał Kwarciński

Polecenie:

Zaimplementować klasyczny algorytm ewolucyjny bez krzyżowania z selekcją turniejową i sukcesją elitarną. Dostępny budżet to 1000 ewaluacji funkcji celu. Optymalizujemy funkcje numeru 4 i 5 z CEC 2017 w 10 wymiarach. Ograniczenia kostkowe przestrzeni to -100, 100.

Dane w tabelkach są średnią z 25 uruchomień.

Funkcja f4

Zmienna: siła mutacji
Stałe: liczebność populacji = 20; liczebność elity = 1:

Siła mutacji	Wartość średnia	Odchylenia standardowe	Najlepsza wartość	Najgorsza wartość
0,5	415,35	22,37	400,14	466,55
0,75	412,88	18,37	400,67	468,19
1	413,97	21,34	400,36	473,96
1,25	410,98	17,54	400,49	472,06
1,5	408,9	13,42	400,73	473,63
2	409,99	12,53	402,99	470,81

Zmienna: liczebność populacji
Stałe: siła mutacji = 1,5; liczebność elity = 1:

Liczebność populacji	Wartość średnia	Odchylenia standardowe	Najlepsza wartość	Najgorsza wartość
5	413,46	21,92	400,26	475,61
10	414,49	21,72	400,69	473,67
15	411,31	17,8	400,67	472,65
25	414,27	20,77	401,32	470,96
30	412,17	19,23	400,76	481,39
40	411,76	16,6	401,06	473,82

Zmienna: liczebność elity
Stałe: siła mutacji = 1,5; liczebność populacji = 15:

Liczebność elity	Wartość średnia	Odchylenia standardowe	Najlepsza wartość	Najgorsza wartość
0	413,23	15,89	405,56	490,48
2	405,44	2,51	400,79	408,5
2	411,34	17,71	401,28	472,43
3	418,59	16,51	400,59	472,33
5	406,12	4,78	400,44	427,12
5	414,38	24,3	401,73	492,02
10	408,43	13,17	401,74	472,5

Najlepsza średnia wartość przy 25 uruchomieniach: 405,44
dla parametrów: siła mutacji = 1,5; liczebność populacji = 15; liczebność elity = 2.

Najmniejsza znaleziona wartość funkcji f4: 400,14

Dla dwóch testów otrzymałem wyniki wyróżniające się na tle reszty (dla zmiennej liczebności elity = 2 i 3). Powtórzyłem te pomiary i okazało się, że najprawdopodobniej wylosowały się „dobre" punkty startowe i duża część mutacji była w „odpowiednim" kierunku.

Funkcja f5

Zmienna: siła mutacji
Stałe: liczebność populacji = 20; liczebność elity = 1:

Siła mutacji	Wartość średnia	Odchylenia standardowe	Najlepsza wartość	Najgorsza wartość
0,25	582,25	31,59	535,87	641,33
0,5	602,03	35,45	546	669,47
0,75	586,04	24,75	536,52	637,88
1	598,06	27,13	562,77	661,61
1,25	585,28	26,49	524,84	637,58
1,5	580,12	27,3	518,92	629,71
1,75	577,39	28,87	533,09	633,82
2	571,97	25,51	538,79	628,96
2,5	572,39	16,09	539,4	599,9

Zmienna: liczebność populacji
Stałe: siła mutacji = 2; liczebność elity = 1:

Liczebność populacji	Wartość średnia	Odchylenia standardowe	Najlepsza wartość	Najgorsza wartość
10	596,55	30,62	542,66	689,61
15	594,17	28,25	533,83	648,13
25	574,68	25,92	537,55	650,37
30	570,31	19,83	532,05	604,64
40	568,17	17,2	537,51	609,06
50	556,07	16,17	520,92	601,11
100	548,73	15,14	526,55	579,44
250	543,45	10,76	523,73	566,35
500	548,77	9	528,36	562,83

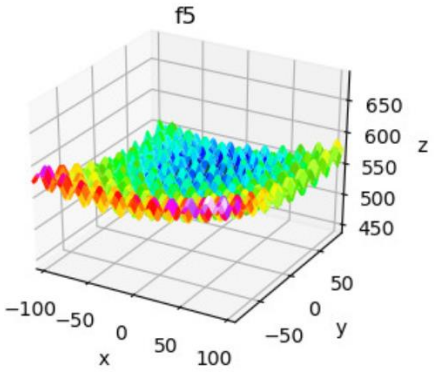
Zmienna: liczebność elity
Stałe: siła mutacji = 2; liczebność populacji = 30:

Liczebność elity	Wartość średnia	Odchylenia standardowe	Najlepsza wartość	Najgorsza wartość
0	556,65	15,45	527,06	584,6
2	567,73	22,19	527,21	643,13
3	584,15	20,2	544,15	617,77
5	578,77	20,98	533,09	618,32
10	580,96	30,0	535,99	676,73

Najlepsza średnia wartość przy 25 uruchomieniach: 543,45
dla parametrów: siła mutacji = 2, liczebność populacji = 250, liczebność elity = 1.

Najmniejsza znaleziona wartość funkcji f5: 518,92

Zauważyłem, że dla f5 algorytm osiągnął najlepszy średni wynik dla dużej populacji (co skutkowało mniejszą liczbą iteracji, gdyż budżet był stały i wynosił 10000 ewaluacji funkcji celu). Może to wynikać ze skomplikowanej budowy omawianej funkcji. Dla owej funkcji algorytm z mniejszą licznością populacji łatwo blokował się w minimach lokalnych, więc im większa populacja startowa, tym lepsze przeszukanie przestrzeni.



Zdjęcie z pliku README z cec2017 na github

Wnioski

Parametry są zależne od przeszukiwanej funkcji. Z racji losowości populacji początkowej analizowanie poszczególnych kombinacji parametrów jest pozbawione większego sensu, ponieważ przy kolejnym uruchomieniu programu z takimi samymi parametrami może dać on zupełnie różne rezultaty (np. tabelka dla liczebności elity w f4 i dwa powtórzone wyniki). Natomiast podczas analizy trendów doszedłem do następujących wniosków:

- Zarówno zbyt mała jak i zbyt duża wartość siły mutacji zazwyczaj nie wpływa pozytywnie na algorytm. Przy zbyt małej wartości algorytm nie będzie eksplorował przestrzeni, przez co będzie szukał tylko w okolicach punktów startowych. Przy zbyt dużej wartości nie będzie on skupiać się na eksploatacji dobrych punktów tylko na przeszukiwaniu przestrzeni.
- Wpływ elity zależy od stosunku jej liczebności do wielkości całej populacji. Mała wartość nie będzie mieć większego wpływu na końcowy wynik, natomiast za duża może albo poprawić (algorytm szybciej znajdzie minimum globalne), albo zaszkodzić (algorytm będzie miał tendencje do blokowania się w słabym minimum lokalnym) - zależnie od wylosowanych punktów początkowych.
- W sytuacji, gdy ogranicza nas ilość ewaluacji funkcji celu, większa populacja dla nieskomplikowanych funkcji (np. f4) może spowodować, że algorytm nie osiągnie minimum przez zbyt małą liczbę iteracji. Dla bardziej złożonych funkcji (np. f5) możliwość większego przeszukania przestrzeni skutkuje lepszym rezultatem.