

POLITECHNIKA WARSZAWSKA

WYDZIAŁ ELEKTRYCZNY

Michał Sut

5 grudnia 2017

1 Treść zadania

Napisać program rozwiązujący problem komiwojażera (minimalizacja drogi pomiędzy n miastami bez powtórzeń) przy pomocy algorytmu genetycznego. Zastosować reprodukcję przy użyciu nieproporcjonalnej ruletki, operator krzyżowania CX oraz mutację równomierną.

Program powinien umożliwiać użycie różnych wielkości populacji, liczby iteracji, prawdopodobieństwa mutacji.

Program powinien zapewnić wizualizację wyników w postaci wykresów średniego, maksymalnego i minimalnego przystosowania (długości funkcji) dla kolejnych populacji oraz 2 map (o wymiarach 10x10 punktów), na których będą wyświetlane miasta oraz drogi najdłuższa i najkrótsza.

Pokazać działanie programu na danych testowych składających się z 10 miast, opisanych za pomocą współrzędnych na mapie o wymiarach 10x10 punktów.

Dane testowe: miasta:

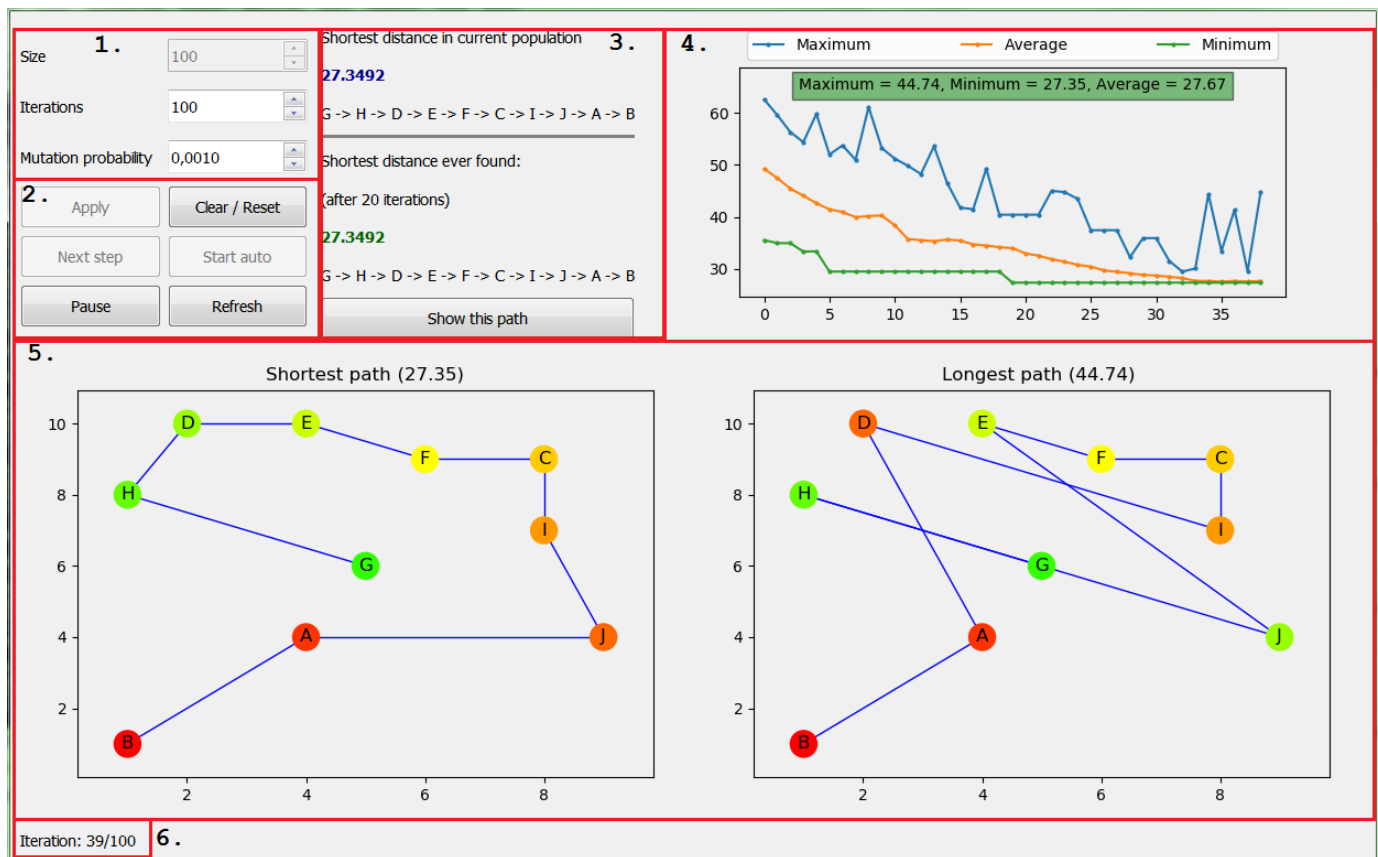
A(4,4), B(1,1), C(8,9), D(2,10), E(4,10), F(6,9), G(5,6), H(1,8), I(8,7), J(9,4)

2 Instrukcja działania programu

2.1 Uruchomienie

Program został napisany w języku Python, z wykorzystaniem bibliotek zawartych w dystrybucji Anaconda. W związku z tym, należy uruchamiać program w tym środowisku. Można to zrobić poprzez zaimportowanie projektu do programu Visual Studio 2017, wybranie interpretera: Anaconda 4.2 i uruchomienie aplikacji. Plikiem startowym jest `ga.py`

2.2 Okno główne programu



2.3 Opis okna programu

1. Parametry algorytmu
2. Przyciski sterowania
3. Informacje o najkrótszych znalezionych trasach
4. Wykres wartości średnich, maksymalnych i minimalnych dla kolejnych generacji.
5. Mapy z miastami i trasami: po lewej – najkrótsza trasa, po prawej – najdłuższa (Kolory oznaczają kolejność miast. Punkt początkowy jest najbardziej zielony, a punkt końcowy - najbardziej czerwony)
6. Informacja o numerze aktualnej iteracji

2.4 Zmiana ustawień programu

Ustawienia programu znajdują się w pliku `settings.py`. Zmianą podlegają następujące elementy:

1. **CITIES** - Nazwy miast
2. **POSITIONS** - Współrzędne miast
3. **COLORS** - Kolory jakimi zaznaczane są kolejne miasta

2.5 Użycie

2.5.1 Podstawowe operacje

1. Ustawić parametry populacji

2. Zainicjować populację klikając przycisk
3. W tym momencie dostępne są trzy możliwości:
 - (a) - Wykonanie tylko jednej generacji
 - (b) - Wykonywanie kolejnych n generacji, gdzie n jest wybraną liczbą iteracji
 - (c) - Wyczyszczenie populacji i zresetowanie programu
4. Rozpoczęty proces automatycznych generacji można wstrzymać przyciskiem
5. Aby zobaczyć najkrótszą kiedykolwiek znalezioną trasę należy brać przycisk

2.6 Warunek stopu

Program zatrzymuje się po określonej liczbie iteracji, którą użytkownik określa w panelu ustawień parametrów algorytmu. Liczbę iteracji można zwiększać w trakcie działania programu.

3 Opis eksperymentów

Eksperymenty przeprowadzone były dla miast o następujących współrzędnych:

$$A(4, 4), B(1, 1), C(8, 9), D(2, 10), E(4, 10), F(6, 9), G(5, 6), H(1, 8), I(8, 7), J(9, 4)$$

3.1 Różna liczebność populacji

W celu przeprowadzenia tych eksperymentów, kilkakrotnie uruchomiony zostanie algorytm z następującymi parametrami: prawdopodobieństwo mutacji – 0,001, liczba iteracji – 200.

3.1.1 Populacja: 10 osobników

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	31,63	34,72	33,48	31,63
2	35,36	35,36	35,36	35,36
3	35,99	35,99	35,99	34,15
4	31,08	31,08	31,08	31,08
5	31,96	31,96	31,96	31,96
	avg. min:	avg. max:	avg. mean:	avg. best:
	33,20	33,82	33,57	32,84

3.1.2 Populacja: 50 osobników

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	30.87	30.87	30.87	30.87
2	28.28	37.84	28.47	28.28
3	32.38	34.34	32.5	32.38
4	31.76	31.76	31.76	31.76
5	27.87	32.28	27.96	27.87
	avg. min:	avg. max:	avg. mean:	avg. best:
	30,14	33,42	30,31	30,11

3.1.3 Populacja: 120 osobników

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	24.59	24.59	24.59	24.59
2	27.34	34.43	27.45	27.34
3	24.59	28.66	24.62	24.59
4	24.59	28.83	24.64	24.59
5	24.59	24.59	24.59	24.59
	avg. min:	avg. max:	avg. mean:	avg. best:
	25.14	28.22	25.17	25.14

3.1.4 Populacja: 600 osobników

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	27.35	44.16	27.5	27.35
2	24.59	32.96	24.74	24.59
3	24.59	41.4	24.75	24.59
4	24.59	38.92	24.7	24.59
5	24.59	41.4	24.69	24.59
	avg. min:	avg. max:	avg. mean:	avg. best:
	25.14	39.77	25.28	25.14

3.2 Różna liczba iteracji

Eksperymenty przeprowadzone będą dla populacji 250 osobników, z prawdopodobieństwem mutacji równym 0.001.

3.2.1 Liczba iteracji: 50

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	27.35	40.03	28.09	27.35
2	24.59	35.61	24.85	24.59
3	29.59	36.61	29.81	29.59
4	28.99	46.97	29.12	28.99
5	24.59	40.6	24.76	24.59
	avg. min:	avg. max:	avg. mean:	avg. best:
	27.02	39.96	27.33	27.02

3.2.2 Liczba iteracji: 125

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	24.59	40.65	24.79	24.59
2	24.59	38.34	28.76	24.59
3	27.93	29.35	27.93	27.93
4	27.35	42.12	27.54	27.35
5	24.59	37.26	26.25	24.59
	avg. min:	avg. max:	avg. mean:	avg. best:
	25.81	37.54	27.05	25.81

3.2.3 Liczba iteracji: 250

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	24.59	41.61	24.84	24.59
2	27.73	35.34	27.89	27.73
3	27.35	40.4	27.48	27.35
4	27.93	34.07	27.97	27.93
5	27.35	44.37	27.49	27.35
	avg. min:	avg. max:	avg. mean:	avg. best:
	26.99	39.16	27.13	26.99

3.2.4 Liczba iteracji: 500

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	29.03	39.89	29.17	29.03
2	24.59	37.24	24.76	24.59
3	29.76	43.78	29.9	29.76
4	24.59	36.91	27.39	24.59
5	29.03	35.8	29.1	29.03
	avg. min:	avg. max:	avg. mean:	avg. best:
	27.4	38.72	28.06	27.4

3.3 Różne prawdopodobieństwa mutacji

Eksperymenty przeprowadzone będą dla populacji 250 osobników. Liczba iteracji: 200

3.3.1 Mutacja: prawdopodobieństwo = 0,000

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	30.11	30.11	30.11	30.11
2	26.19	26.19	26.19	26.19
3	28.88	28.88	28.88	28.88
4	29.03	29.03	29.03	29.03
5	26.37	26.37	26.37	26.37
	avg. min:	avg. max:	avg. mean:	avg. best:
	28.12	28.12	28.12	28.12

3.3.2 Mutacja: prawdopodobieństwo = 0,001

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	24.59	34.09	24.72	24.59
2	30.71	43.21	30.85	30.71
3	27.35	40.81	27.71	27.35
4	24.59	28.51	25.01	24.59
5	27.35	40.18	27.58	27.35
	avg. min:	avg. max:	avg. mean:	avg. best:
	26.92	37.36	27.17	26.92

3.3.3 Mutacja: prawdopodobieństwo = 0,004

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	24.59	42.02	25.39	24.59
2	24.59	38.92	25.08	24.59
3	24.59	39.19	25.28	24.59
4	29.35	46.96	30.26	29.35
5	24.59	42.56	25.29	24.59
	avg. min:	avg. max:	avg. mean:	avg. best:
	25.54	41.93	26.26	25.54

3.3.4 Mutacja: prawdopodobieństwo = 0,01

Wyniki prezentują się następująco:

Lp	Min	Max	Mean	Best
1	29.24	49.74	30.65	29.24
2	24.59	42.6	26.44	24.59
3	29.03	53.71	31.79	29.03
4	24.59	55.74	26.69	24.59
5	24.59	45.19	26.35	24.59
	avg. min:	avg. max:	avg. mean:	avg. best:
	26.4	49.4	28.38	26.4

4 Wnioski

Przeprowadzone eksperymenty pokazują, że zmiana parametrów populacji wpływa na znajdowanie maksimum funkcji dopasowania danej populacji.

4.1 Prawdopodobnie najlepsze rozwiązanie

Po przeprowadzeniu wszystkich eksperymentów, czyli uruchomieniu algorytmu 60 razy, można przypuszczać że najmniejsza z wartości, jaką podczas wszystkich prób udało się uzyskać, jest rozwiązaniem optymalnym lub przynajmniej jest zbliżona do wartości optymalnej. W tym przypadku, najmniejsza długość trasy, jaką udało się znaleźć, wynosiła: **24,5853**, a trasa ta to: **B -> A -> G -> D -> E -> F -> C -> I -> J**

4.2 Dokładność algorytmu

Algorytm nie zawsze znajduje najkrótszą możliwą drogę. W większości przypadków znajdowane były wartości bliskie oczekiwanej, czyli trasy, o długości zbliżonej do optymalnej. Na fakt znalezienia lub nie znalezienia optymalnego rozwiązania wpływają parametry populacji.

4.3 Wpływ wielkości populacji na wynik działania algorytmu

Przeprowadzone eksperymenty pokazują, że statystycznie lepsze wyniki uzyskiwane są dla większych populacji. Im większa populacja, tym większa szansa na znalezienie optymalnego rozwiązania. Przeprowadzone eksperymenty pokazały, że w większej populacji, wartość oszacowana jako najlepsza (w punkcie „4.1.”) była znajdowana najczęściej. Dodatkowo, im większa jest populacja, tym więcej iteracji potrzeba, aby populację zdominowało najlepsze rozwiązanie.

4.4 Wpływ liczby iteracji na wynik działania algorytmu

Przeprowadzone badania pokazały, że im dłużej działa algorytm, tym lepsze wartości są znajdowane. Jest to szczególnie widocznie, przy małych populacjach, w których pośród początkowych wartości nie ma wartości optymalnych. W tym przypadku mutacje i krzyżowania muszą zajść więcej razy, aby udało się wygenerować coraz lepsze rozwiązania.

Ponadto, z każdą kolejną iteracją etap reprodukcji zwiększa liczbę najlepszych rozwiązań w populacji, a zmniejsza liczbę tych najgorszych, co zwiększa szanse na znalezienie optymalnej wartości z każdą kolejną iteracją.

4.5 Wpływ prawdopodobieństwa mutacji na wynik działania algorytmu

Prawdopodobieństwo mutacji ma różny wpływ na działanie algorytmów w zależności od innego z parametrów: liczby osobników w populacji. Dla małych populacji, zwiększenie omawianego parametru powoduje, że generowane jest wiele nowych rozwiązań, wśród których mogą znajdować się osobniki bardziej przystosowane od obecnie istniejących w populacji. Jednakże, wraz ze wzrostem wielkości populacji, prawdopodobieństwo mutacji powinno się zmniejszać, aby uzyskiwać dobre wyniki. Dzieje się tak, ponieważ w większej populacji jest większa szansa na wystąpienie lepszych osobników. Mutacja może jednak wyeliminować te najlepsze wyniki, szczególnie w przypadku, gdy jest ich niewiele w populacji.

Z drugiej strony, brak mutacji powoduje, że nowe rozwiązania znajdowane są znacznie rzadziej, ponieważ nowe rozwiązania powstają tylko w procesie krzyżowania. Dla zerowego prawdopodobieństwa mutacji, populacja zostaje szybko stosunkowo szybko zdominowana przez osobników o takim samym przystosowaniu.