Skład zespołu:

Mateusz Winnicki

WSI ĆWICZENIE 2

Cel ćwiczenia

Tematem drugich ćwiczeń są algorytmy genetyczne i ewolucyjne. Zadaniem do wykonania jest implementacja algorytmu ewolucyjnego bez krzyżowania, z selekcją turniejową i sukcesją generacyjną.

Problemem, dla którego będziemy szukać najlepszego rozwiązania w ograniczonej liczbie iteracji, jest problem komiwojażera.

Algorytm będziemy badać w zależności od parametrów wejściowych, które określać będą:

- typ rozmieszczenia miast (jednolity/gromadowy/losowy),
- rozmiar populacji,
- wymagany próg mutacji (np. dla progu równego 0,8 mutacja zajdzie statystycznie w 20% przypadków),
- maksymalną liczbę iteracji.

Przebieg ćwiczenia

Na potrzeby ćwiczenia zakładamy stałą liczbę miast do odwiedzenia równą 30 oraz turnieje tylko między dwoma osobnikami z populacji. Mutacja, występująca w statystycznie (1 – próg mutacji) % osobników, polega na zamianie miejscami dwóch losowych miast w ścieżce. Wszystkie badania będą przeprowadzane na 15 próbach, by wyciągnąć z nich wartości takie jak średnia, czy odchylenie standardowe, z racji losowego charakteru algorytmów ewolucyjnych.

Algorytmy zostały zaimplementowane w Pythonie, z użyciem następujących bibliotek:

- *numpy,*
- *argparse,*
- *matplotlib,*
- *math,*
- *copy,*
- *sklearn.datasets (do generacji skupisk wielu miast).*

Aby uruchomić program należy to zrobić z parametrami określającymi: liczbę miast, typ rozmieszczenia miast – [*ud* | *cd* | *rd*], rozmiar populacji, rozmiar turnieju, wymagany próg mutacji oraz maksymalną liczbę iteracji.

Dodanie do linii *-plt* wykona program kreśląc grafy z przebiegiem najkrótszej drogi w populacjach co 1/5 maksymalnej liczby iteracji. Program zakończy się wyświetleniem wykresu najkrótszej ścieżki od danej iteracji.

Przykład:

> *py .\evolutionary_algorithms.py -cn 30 -cd rd -p 300 -t 2 -m 0.85 -i 10000 -plt*

Wyniki ćwiczenia i wnioski

Badania rozpoczniemy od wpływu progu mutacji na wyniki algorytmu. Zestaw pozostałych parametrów przy wszystkich pomiarach będzie stały i równy:

- Rozkład miast: 3 duże skupiska grup[1]
- Osobników w populacji: 300
- Liczba iteracji: 10000

Próg mutacji	Najkrótsza wyznaczona droga	Najdłuższa wyznaczona droga	Średnia długość wyznaczonych dróg	Odchylenie standardowe
0,1	227,98	277,70	261,16	15,14
0,2	226,71	279,24	257,17	12,65
0,33	215,77	279,64	252,81	19,69
0,5	204,17	281,41	249,55	23,38
0,7	214,63	277,41	247,55	16,82
0,85	218,44	279,64	252,44	15,91
0,875	195,72	278,26	245,24	22,9
0,9	232,48	278,28	254,71	10,62
0,95	219,99	278,65	256,19	12,83
0,99	222,78	280,76	254,71	14,42

Próg mutacji	Najszybsze wyznaczenie [generacja]	Najpóźniejsze wyznaczenie [generacja]	Średnia iteracja wyznaczenia [generacja]	Odchylenie standardowe [generacja]
0,1	1953	8866	5238	2368
0,2	1089	9641	5765	2545
0,33	18	8977	4518	3055
0,5	12	9854	4336	3517
0,7	157	9860	4467	3114
0,85	644	9543	5407	2995
0,875	475	9220	5553	2910
0,9	20	7731	4664	2209
0,95	924	9440	6019	2555
0,99	59	8282	5172	2374

[1] Być może lepszym wyborem byłby rozkład jednolity, a badania kończyłby się porównaniem go z pozostałymi typami rozmieszczenia miast, jednak pomysł ten przyszedł do mnie już po wykonaniu wielu eksperymentów, których wyniki były dosyć sensowne. Rozkład „gromadowy” (czyli 3 grupki skupionych miast) i tak jest bardziej regularny niż rozkład zupełnie losowy.

Jak widać wyniki charakteryzują się dużą losowością. Możemy jednak zauważyć kilka zależności, zwracając szczególną uwagę na średnie wartości otrzymanych długości oraz czasów ich odkrycia.

Najlepsze wyniki, zarówno pod względem średniej długości najkrótszej ścieżki, jak i średniej generacji wyznaczenia, otrzymujemy dobierając próg mutacji nie będący zbyt duży jak i zbyt mały. Jak widać w tabeli, najkrótsze ścieżki[2] udało nam się znaleźć dla progów mutacji równych <0,33; 0,85>. Algorytm odkrywał je również wcześniej.

Może to wynikać z faktu, że w przypadku rzadkich mutacji (duże wartości progu) populacja nie jest w stanie szukać nowych rozwiązań zbyt efektywnie, często musimy czekać wiele generacji, zanim nieliczne mutacje dadzą nam oczekiwany efekt. Natomiast zbyt częste mutacje poruszają się po przestrzeni zbyt chaotycznie, znalezienie satysfakcjonującego rozwiązania zaczyna zależeć przede wszystkim od szczęścia. Dobrze obrazuje to również najwcześniejsza generacja wyznaczenia najkrótszej ścieżki dla progów mutacji równych 0,1 i 0,2 – jest ona zdecydowanie wyższa niż w pozostałych przypadkach.

W poprzednim badaniu w większości przypadków algorytm często znajdował najlepszą drogę w generacjach rzędu 9000. Wybierzemy teraz jeden z optymalnych progów mutacji i zaczniemy zwiększać liczbę iteracji, by program miał szansę znaleźć jeszcze lepsze rozwiązania.

- Rozkład miast: 3 duże skupiska grup
- Osobników w populacji: 300
- Próg mutacji: 0,7

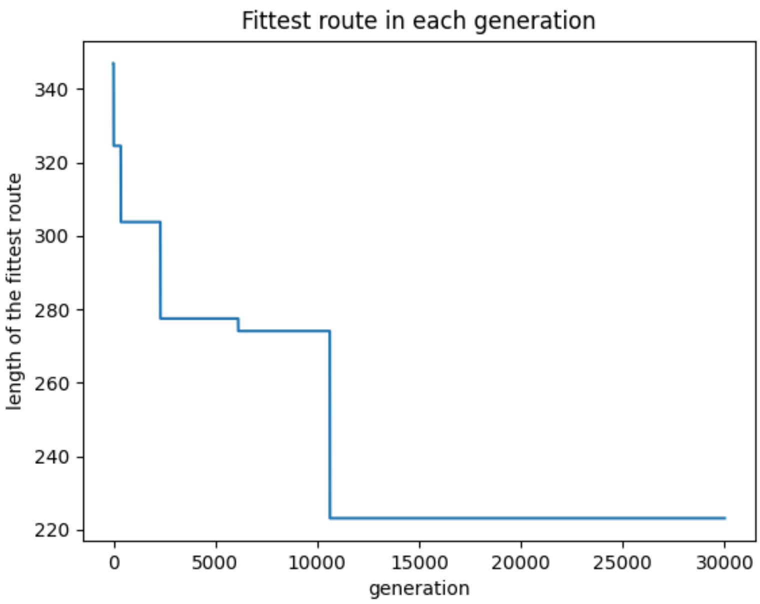
Liczba iteracji	Najkrótsza wyznaczona droga	Najdłuższa wyznaczona droga	Średnia długość wyznaczonych dróg	Odchylenie standardowe
10000	214,63	277,41	247,55	16,82
15000	214,08	274,52	243,12	17,64
20000	211,28	261,77	242,35	14,35
30000	217,27	259,09	244,76	11,93

[2] Co prawda najkrótszą ścieżkę oraz średnią długość ścieżek udało nam się odnaleźć dla progu mutacji równego 0,875, jednak wartość odchylenia standardowego jest bardzo duża, dlatego wynik ten nie jest najbardziej rzetelny.

Liczba iteracji	Najszybsze wyznaczenie [generacja]	Najpóźniejsze wyznaczenie [generacja]	Średnia iteracja wyznaczenia [generacja]	Odchylenie standardowe [generacja]
10000	157	9860	4467	3114
15000	17	14967	4565	5204
20000	1384	16313	7731	5554
30000	104	17766	7695	4876

Okazuje się, że najbardziej optymalnym wyborem jest 20000 iteracji. Udało nam się znaleźć najkrótszą drogę ze wszystkich dotychczasowych, której długość wynosi 211,28 oraz średnią i odchylenie standardowe, z których można wywnioskować, że wyniki w każdym przebiegu były dosyć stabilne, a nie losowe. Zwiększenie liczby iteracji do 30000 poskutkowało jedynie zmniejszeniem przedziału znalezionych rozwiązań (długości od 217,27 do 259,09). Nie udało się jednak znaleźć drogi lepszej niż w poprzednim przypadku, co więcej najpóźniejsze wyznaczenie najkrótszej drogi wystąpiło w 17766. generacji. Wykonywanie 15 przebiegów przy takiej liczbie iteracji trwało 9 minut i 10 sekund, co może skłaniać do wniosków, że nie ma sensu zwiększać dalej liczby badanych pokoleń.

Chyba że przyjrzymy się poniższemu wykresowi najlepszego rozwiązania od numeru generacji. Wtedy możemy dojść do konkluzji (zgodnej zresztą z wiedzą wykładową), że kolejne rozwiązania odkrywamy „epokowo". Często zanim znajdziemy kolejne rozwiązanie, które będzie lepsze od dotychczasowych, musimy czekać przez wiele pokoleń. Być może, gdybyśmy poczekali jeszcze 10000 generacji (po tych 30 tysiącach), udałoby nam się znaleźć jeszcze krótszą ścieżkę pomiędzy miastami. Musimy się jednak zastanowić – czy faktycznie będzie to rozwiązanie na tyle lepsze, że jesteśmy skorzy czekać jeszcze dłużej?



Do zbadania został nam ostatni parametr, jakim jest wielkość populacji. Skorzystamy ze znalezionej wartości optymalnej progu mutacji oraz liczby mutacji równej 15000.

- Rozkład miast: 3 duże skupiska grup
- Liczba iteracji: 20000
- Próg mutacji: 0,7

Liczba osobników w populacji	Najkrótsza wyznaczona droga	Najdłuższa wyznaczona droga	Średnia długość wyznaczonych dróg	Odchylenie standardowe
100	227,43	260,62	247,06	11,12
200	213,55	260,44	246,79	13,84
300	211,28	261,77	242,35	14,35
400	222,16	264,24	244,39	13,27
500	224,12	271,14	241,31	12,26

Liczba osobników w populacji	Najszybsze wyznaczenie [generacja]	Najpóźniejsze wyznaczenie [generacja]	Średnia iteracja wyznaczenia [generacja]	Odchylenie standardowe [generacja]
100	90	19930	11770	6148
200	371	19543	11410	5831
300	1384	16313	7731	5554
400	1829	15331	9659	4505
500	72	18627	8007	5891

Populacja musi być odpowiednio duża, żeby algorytmowi udało się odnaleźć zadowalające wyniki. Oczywiście wiąże się to z czasem wykonywania programu. Liczba ścieżek w populacji równa 100 okazała się niewystarczająca, natomiast rozwiązania dla 200 i 300 osobników były zadowalające. Dalsze zwiększanie tego parametru wiąże się z długim[3] czasem wykonywania – dla 500 osobników było to już 10 minut i 22 sekundy. Co więcej, jak widać nie jest to gwarantem lepszych wyników.

[3] Oczywiście jak na to laboratorium, dla rozwiązań problemów jakkolwiek zbliżonych do komercyjnych każde z prowadzonych powyżej badań byłoby prawdopodobnie niesłychanie krótkie.

Na koniec zobaczymy jak z optymalnym zestawem parametrów dla rozkładu miast w 3 aglomeracjach, poradzą sobie dwa pozostałe rozkłady – jednolity i losowy.

- Osobników w populacji: 300
- Liczba iteracji: 20000
- Próg mutacji: 0,7

Rozkład miast	Najkrótsza wyznaczona droga	Najdłuższa wyznaczona droga	Średnia długość wyznaczonych dróg	Odchylenie standardowe
jednolity	710,91	803,46	761,96	22,27
gromadowy	211,28	261,77	242,35	14,35
losowy	315,81	344,23	330,24	7,51

Rozkład miast	Najszybsze wyznaczenie [generacja]	Najpóźniejsze wyznaczenie [generacja]	Średnia iteracja wyznaczenia [generacja]	Odchylenie standardowe [generacja]
jednolity	1708	19439	11129	5897
gromadowy	1384	16313	7731	5554
losowy	1141	18111	7535	4911

Rozkład jednolity znajdował najlepsze rozwiązanie w generacjach późniejszych niż rozkład gromadowy – prawdopodobnie kalibrując liczbę iteracji, bylibyśmy w stanie odkryć rozwiązania lepsze od uzyskanych przy 20000 iteracji.
Rozkład losowy, którego długości dróg były bliższe drogom w rozkładzie gromadowym, posiada średnią iterację wyznaczenia zbliżoną do drugiego typu dystrybucji, dlatego prawdopodobnie maksymalna liczba generacji rzędu 20000 jest stosunkowo dobra. Znalezione długości najkrótszej ścieżki cechują się bardzo małym odchyleniem standardowym, dlatego przeprowadzimy jeszcze jeden pomiar dla tego przypadku, z progiem mutacji równym 0,5 (co drugi osobnik mutuje).

Najkrótsza wyznaczona droga	Najdłuższa wyznaczona droga	Średnia długość wyznaczonych dróg	Odchylenie standardowe
312,17	349,99	328,04	9,28
Najszybsze wyznaczenie [generacja]	Najpóźniejsze wyznaczenie [generacja]	Średnia iteracja wyznaczenia [generacja]	Odchylenie standardowe [generacja]
687	19805	9015	5860

Podsumowanie

Algorytmy ewolucyjne to potężne narzędzia do szukania rozwiązań optymalnych, które z zestawem odpowiednich parametrów wejściowych są w stanie dać nam satysfakcjonujące wyniki w rozsądnym czasie. Jeżeli jesteśmy w stanie dobrze dopasować zestaw wejściowy algorytmu, zadania o dużej złożoności obliczeniowej, takie jak problem komiwojażera, stają się wykonywalne, przy założeniu stosunku jakość rozwiązania/czas jego odkrycia, które zadowoli użytkownika. Na pewno pomaga również zrównoleglanie procesów, którego w tym ćwiczeniu nie wykorzystałem.