Paweł Biel

nr indeksu 225949

Sprawozdanie

**Algorytm genetyczny dla problemu komiwojażera**

1. **Informacje teoretyczne o algorytmie genetycznym**

Algorytm genetyczny – rodzaj heurystyki przeszukującej przestrzeń alternatywnych rozwiązań problemu w celu wyszukania rozwiązań najlepszych. Sposób działania algorytmów genetycznych nieprzypadkowo przypomina zjawisko ewolucji biologicznej, ponieważ ich twórca John Henry Holland właśnie z biologii czerpał inspiracje do swoich prac.

Każde środowisko ma ograniczone zasoby - określa to maksymalną liczbę osobników (populacja) mogących utrzymać się w nim przy życiu. Liczba osobników   
zmienia się w czasie - najczęściej rośnie, ponieważ osobniki rozmnażają się -> populacja się powiększa.

Liczba osobników musi być regulowana - środowisko określa warunki przeżycia, osobniki chcąc przeżyć muszą się do tych warunków przystosować. Z tego wynika, że najlepiej przystosowane osobniki mają największą szansę na przeżycie i rozmnażanie.

W procesie rozmnażania osobniki najczęściej przekazują swoje „przystosowanie” potomstwu - nie jest to regułą i możliwe jest wydanie potomstwa gorzej niż rodzice przystosowanego do warunków określonych przez środowisko.

Oprócz cech poprawiających przystosowanie, u potomstwa mogą występować losowe zmiany (mutacje), które mogą prowadzić do zmian ich przystosowania (lepsze lub gorsze przystosowanie), więc jeśli środowisko nie zmieni diametralnie warunków przystosowania to taki sposób postępowania prowadzi (w czasie) do coraz lepszego przystosowania populacji.

1. **Informacje wstępne**

**W mojej implementacji algorytmu rozpoczynam od wygenerowania populacji początkowej na zasadzie zbliżonej do metody zachłannej, losuje dwa lub trzy wierzchołki a następnie resztę ścieżki wybieram na zasadzie metody zachłannej, następnie zostaje oceniana populacja.**

**Kolejnym etapem jest selekcja, którą wymieszałem przez początkową część działania programu selekcja jest wybierana na zasadzie wyboru np. 100 najlepszych osobników, wraz z czasem działania programu i kolejnymi iteracjami metodą selekcji jest selekcja turniejowa, podczas selekcji jest zawsze wybierana stała ilość osobników.**

**Dalej osobniki są krzyżowane metodą PMX, a rodzicami są osobniki iterowane po kolei tzn. Pierwszy z drugim, drugi z trzecim, trzeci… itd. Zostaje jeszcze mutacja, to czy mutacja zajdzie na pojedynczym genie decyduje określone wcześniej prawdopodobieństwo, jeśli dojdzie do mutacji to mutacja przebiega na zasadzie tzw. losowego swapa.**

**Warunkiem zatrzymania algorytmu jest określona ilość populacji bez poprawy wyniku.**

1. **Lista kroków algorytmu**

1. wybór populacji początkowej chromosomów (losowy)

2. ocena przystosowania chromosomów

3. sprawdzanie warunku zatrzymania

a. selekcja chromosomów - wybór populacji macierzystej (ang. mating pool )

b. krzyżowanie chromosomów z populacji rodzicielskiej

c. mutacja

d. ocena przystosowania chromosomów

e. utworzenie nowej populacji

4. wyprowadzenie ,,najlepszego’’ rozwiązania

**Powyższa lista została zaczerpnięta z prezentacji Dr. Kapłona**

1. **Wyniki**

**Dla każdej konfiguracji zostało przeprowadzone 10 prób oraz następne uśrednienie wyników.**

* 1. **Wyniki dla macierzy asymetrycznych oraz ich wykresy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rodzaj instancji** | **Rozmiar populacji** | **Prawdopodobieństwo** | **Czas [s]** | **Wynik** |
| br17.atsp | 500 | 0,01 | 3,211973316 | 39 |
| br17.atsp | 500 | 0,04 | 3,260722748 | 39 |
| br17.atsp | 500 | 0,08 | 3,218631105 | 39 |
| br17.atsp | 500 | 0,1 | 3,231885673 | 39 |
| br17.atsp | 1000 | 0,01 | 11,55566039 | 39 |
| br17.atsp | 1000 | 0,04 | 11,50841928 | 39 |
| br17.atsp | 1000 | 0,08 | 11,54377105 | 39 |
| br17.atsp | 1000 | 0,1 | 11,66180183 | 39 |
| br17.atsp | 1500 | 0,01 | 25,24254281 | 39 |
| br17.atsp | 1500 | 0,04 | 25,59180866 | 39 |
| br17.atsp | 1500 | 0,08 | 24,91856657 | 39 |
| br17.atsp | 1500 | 0,1 | 25,53177949 | 39 |
|  |  |  | **MIN** | **39** |
| ftv33.atsp | 500 | 0,01 | 5,24830133 | 1459,2 |
| ftv33.atsp | 500 | 0,04 | 4,038356214 | 1420,2 |
| ftv33.atsp | 500 | 0,08 | 4,501955411 | 1446,4 |
| ftv33.atsp | 500 | 0,1 | 4,789745436 | 1473,2 |
| ftv33.atsp | 1000 | 0,01 | 13,4542274 | 1434,2 |
| ftv33.atsp | 1000 | 0,04 | 13,30859363 | 1421,2 |
| ftv33.atsp | 1000 | 0,08 | 13,42832249 | 1439 |
| ftv33.atsp | 1000 | 0,1 | 13,31141111 | 1439,6 |
| ftv33.atsp | 1500 | 0,01 | 27,83212203 | 1400,6 |
| ftv33.atsp | 1500 | 0,04 | 28,54450737 | 1408 |
| ftv33.atsp | 1500 | 0,08 | 27,40761204 | 1398,8 |
| ftv33.atsp | 1500 | 0,1 | 27,80482609 | 1411,8 |
|  |  |  | **MIN** | **1398,8** |
| ftv47.atsp | 500 | 0,01 | 8,135766628 | 2094,6 |
| ftv47.atsp | 500 | 0,04 | 7,583633576 | 2053,2 |
| ftv47.atsp | 500 | 0,08 | 6,586493776 | 2085,8 |
| ftv47.atsp | 500 | 0,1 | 6,560823257 | 2093,8 |
| ftv47.atsp | 1000 | 0,01 | 16,68440357 | 2091,8 |
| ftv47.atsp | 1000 | 0,04 | 18,49750203 | 2080 |
| ftv47.atsp | 1000 | 0,08 | 16,41088833 | 2069,4 |
| ftv47.atsp | 1000 | 0,1 | 17,51350525 | 2074 |
| ftv47.atsp | 1500 | 0,01 | 30,95538522 | 2084,8 |
| ftv47.atsp | 1500 | 0,04 | 31,33840031 | 2088,2 |
| ftv47.atsp | 1500 | 0,08 | 32,14235625 | 2080,6 |
| ftv47.atsp | 1500 | 0,1 | 30,92286586 | 2066 |
|  |  |  | **MIN** | **2053,2** |
| ftv70.atsp | 500 | 0,01 | 6,237221025 | 2339,4 |
| ftv70.atsp | 500 | 0,04 | 8,500165046 | 2314,8 |
| ftv70.atsp | 500 | 0,08 | 11,14474283 | 2291,4 |
| ftv70.atsp | 500 | 0,1 | 11,25912599 | 2307,4 |
| ftv70.atsp | 1000 | 0,01 | 20,11305529 | 2305,2 |
| ftv70.atsp | 1000 | 0,04 | 24,31564462 | 2290,4 |
| ftv70.atsp | 1000 | 0,08 | 19,78961401 | 2330 |
| ftv70.atsp | 1000 | 0,1 | 21,83760778 | 2300,6 |
| ftv70.atsp | 1500 | 0,01 | 34,73630009 | 2304,6 |
| ftv70.atsp | 1500 | 0,04 | 43,50632264 | 2281,4 |
| ftv70.atsp | 1500 | 0,08 | 38,28437035 | 2296,4 |
| ftv70.atsp | 1500 | 0,1 | 38,6688449 | 2294,8 |
|  |  |  | **MIN** | **2281,4** |
| ftv170.atsp | 500 | 0,01 | 20,37199012 | 3607,2 |
| ftv170.atsp | 500 | 0,04 | 26,2213409 | 3585,8 |
| ftv170.atsp | 500 | 0,08 | 37,31953617 | 3564,6 |
| ftv170.atsp | 500 | 0,1 | 31,352796 | 3608,6 |
| ftv170.atsp | 1000 | 0,01 | 56,94163613 | 3572,8 |
| ftv170.atsp | 1000 | 0,04 | 53,20009581 | 3557 |
| ftv170.atsp | 1000 | 0,08 | 89,05271679 | 3559,2 |
| ftv170.atsp | 1000 | 0,1 | 114,9658001 | 3506,6 |
| ftv170.atsp | 1500 | 0,01 | 76,64094636 | 3561,4 |
| ftv170.atsp | 1500 | 0,04 | 125,9137177 | 3512,4 |
| ftv170.atsp | 1500 | 0,08 | 183,9891512 | 3531 |
| ftv170.atsp | 1500 | 0,1 | 160,0308579 | 3506,6 |
|  |  |  | **MIN** | **3506,6** |

* 1. **Wyniki dla macierzy symetrycznych oraz ich wykresy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rodzaj instancji** | **Rozmiar populacji** | **Prawdopodobieństwo** | **Czas** | **Wynik** |
| gr17.tsp | 500 | 0,01 | 3,349909097 | 2085 |
| gr17.tsp | 500 | 0,04 | 3,335020345 | 2085 |
| gr17.tsp | 500 | 0,08 | 3,285508456 | 2085 |
| gr17.tsp | 500 | 0,1 | 3,25531591 | 2085 |
| gr17.tsp | 1000 | 0,01 | 11,70195283 | 2085 |
| gr17.tsp | 1000 | 0,04 | 11,77780464 | 2085 |
| gr17.tsp | 1000 | 0,08 | 11,54186043 | 2085 |
| gr17.tsp | 1000 | 0,1 | 11,9875306 | 2085 |
| gr17.tsp | 1500 | 0,01 | 27,59135218 | 2085 |
| gr17.tsp | 1500 | 0,04 | 27,77251697 | 2085 |
| gr17.tsp | 1500 | 0,08 | 27,14626194 | 2085 |
| gr17.tsp | 1500 | 0,1 | 27,43576768 | 2085 |
|  |  |  | MIN | 2085 |
| gr24.tsp | 500 | 0,01 | 3,569382371 | 1340,3 |
| gr24.tsp | 500 | 0,04 | 3,756667549 | 1342,7 |
| gr24.tsp | 500 | 0,08 | 3,583940878 | 1340,9 |
| gr24.tsp | 500 | 0,1 | 3,693527696 | 1340,4 |
| gr24.tsp | 1000 | 0,01 | 13,41993543 | 1326,6 |
| gr24.tsp | 1000 | 0,04 | 12,41204225 | 1330,2 |
| gr24.tsp | 1000 | 0,08 | 12,32866886 | 1325,7 |
| gr24.tsp | 1000 | 0,1 | 12,56873317 | 1335,2 |
| gr24.tsp | 1500 | 0,01 | 28,87329605 | 1312,8 |
| gr24.tsp | 1500 | 0,04 | 26,34644638 | 1329,3 |
| gr24.tsp | 1500 | 0,08 | 25,92571462 | 1329,3 |
| gr24.tsp | 1500 | 0,1 | 25,9961599 | 1331,4 |
|  |  |  | MIN | 1312,8 |
| gr48.tsp | 500 | 0,01 | 10,54175031 | 5647,1 |
| gr48.tsp | 500 | 0,04 | 9,515240867 | 5710,4 |
| gr48.tsp | 500 | 0,08 | 9,547081625 | 5646,9 |
| gr48.tsp | 500 | 0,1 | 8,354438816 | 5595,4 |
| gr48.tsp | 1000 | 0,01 | 25,0213679 | 5630,5 |
| gr48.tsp | 1000 | 0,04 | 26,25432747 | 5602,5 |
| gr48.tsp | 1000 | 0,08 | 21,92286045 | 5591,7 |
| gr48.tsp | 1000 | 0,1 | 21,34553543 | 5641,7 |
| gr48.tsp | 1500 | 0,01 | 39,51726475 | 5591,2 |
| gr48.tsp | 1500 | 0,04 | 44,34045889 | 5619 |
| gr48.tsp | 1500 | 0,08 | 37,17922511 | 5596,3 |
| gr48.tsp | 1500 | 0,1 | 36,04224542 | 5582,3 |
|  |  |  | MIN | 5582,3 |
| gr120.tsp | 500 | 0,01 | 19,6115199 | 8363,6 |
| gr120.tsp | 500 | 0,04 | 56,82292631 | 8216,2 |
| gr120.tsp | 500 | 0,08 | 56,26556135 | 8031,6 |
| gr120.tsp | 500 | 0,1 | 63,88745641 | 8080,6 |
| gr120.tsp | 1000 | 0,01 | 80,46851021 | 8191,6 |
| gr120.tsp | 1000 | 0,04 | 110,6754702 | 8024,2 |
| gr120.tsp | 1000 | 0,08 | 104,1079041 | 8039,4 |
| gr120.tsp | 1000 | 0,1 | 96,81932667 | 7961,4 |
| gr120.tsp | 1500 | 0,01 | 148,7991107 | 8053,8 |
| gr120.tsp | 1500 | 0,04 | 147,9927905 | 8001,6 |
| gr120.tsp | 1500 | 0,08 | 132,6678614 | 8018,6 |
| gr120.tsp | 1500 | 0,1 | 107,7154778 | 7968 |
|  |  |  | **MIN** | **7961,4** |

* 1. **Porównanie algorytmów**

Z racji że w pierwszym projekcie nie udało mi się uruchomić pomiarów dla macierzy symetrycznych, a bruteforce dla tak dużych instancji wykonuje się bardzo długo lub też instancja problemu jest na tyle duża, że jest to nie możliwe dlatego zamieszczam wykres tylko z zakresu obecnego i poprzedniego projektu.

* 1. **Tabela jakości otrzymanych wyników**

|  |  |  |  |
| --- | --- | --- | --- |
| Rodzaj instancji | Otrzymany wynik | Najlepszy znany wynik | Błąd |
| gr17 | 2085 | 2085 | 0% |
| br17 | 39 | 39 | 0% |
| gr24 | 1312 | 1272 | 3% |
| ftv33 | 1398 | 1286 | 9% |
| ftv47 | 2053 | 1776 | 16% |
| gr48 | 5582 | 5046 | 11% |
| ftv70 | 2281 | 1950 | 17% |
| gr120 | 7961 | 6942 | 15% |
| ftv170 | 3506 | 2755 | 27% |

1. **Wnioski** 
   1. **Czym większa populacja tym większa szansa na znalezienie lepszego wyniku, lecz czas wykonywania się algorytmu wydłuża się.**
   2. **Dla małych instancji o wielkościach 17 czy też 33 algorytm jest w stanie znaleźć rozwiązanie w dużo lepszym czasie niż metody B&B czy BruteForce natomiast porównywalnie lub czasem i szybciej niż TabuSearch**
   3. Dla dużych instancji tak jak tabu search znajduje szybciej rozwiązanie, z prawdopodobieństwem błędu, więc jeżeli nasz problem pozwala na założenie, że nasze wyniki nie są dokładne, ale są oszacowane i mieszczą się w zakresie dla nas dopuszczalnego błędu, to tabu search i algorytm genetyczny są dobrymi metodami na znalezienie rozwiązań w rozsądnym czasie.
   4. Porównując projekt 2 z obecnym to mój algorytm genetyczny znajduje lepsze wartości wyników niż mój TabuSearch
   5. Z wykresu można wywnioskować, że złożoność algorytmu genetycznego wynosi O(n2)
   6. **Wpływ prawdopodobieństwa zajścia mutacji, ma nie wielki wpływ na wynik czasem może polepszyć rozwiązanie czasem pogorszyć i nie można znaleźć reguły, która by jednoznacznie powiedziała, czy mutacja to zjawisko sprzyjające w polepszaniu wyniku, natomiast można stwierdzić, że czym większe prawdopodobieństwo zajścia mutacji tym czas znalezienia rozwiązania może się zwiększyć.**