**Projektowanie Efektywnych Algorytmów - Projekt 3**

**Temat: Algorytm oparty na metodzie algorytmu populacji**

**Opracował: Miłosz Siofer**

**Nr. Indeksu: 256794**

**Prowadzący: Mgr Inż. Łukasz Kamiński**

1.Wstęp

Algorytm genetyczny

**Algorytm genetyczny** jest heurystyką przeszukującą przestrzeń alternatywnych rozwiązań bazującą na zjawisku ewolucji biologicznej. Zadaniem algorytmu jest symulowanie populacji danego rozwiązania dążącego do jak największego przystosowania się do otoczenia, czyli do uzyskania jak najlepszego rozwiązania badanego problemu. Wykorzystując naturalne mechanizmy takie jak rozmnażanie czy mutację osobników staramy się przystosować kolejne pokolenia rozwiązań by coraz bardziej zbliżać się do optymalnego rozwiązania (jednak nie mamy żadnej gwarancji na jego odnalezienie).

Podstawą algorytmu jest **populacja**, czyli grupa osobników danego problemu. To na jej podstawie przebiega dalsze dostosowywanie się osobników do otoczenia, gdyż jest ona bazą dla wykonywanego rozmnażania. Początkowa populacja generowana jest zazwyczaj losowo. Kiedy istnieje już populacja, następuje właściwa część algorytmu zaczynając od **selekcji**. Polega ona na wybraniu jak najbardziej optymalnych osobników do późniejszego rozmnożenia. Zazwyczaj są to osobniki najlepiej przystosowane, jednak nie jest to regułą.

Ostatnim elementem algorytmu jest wybór **nowej populacji**, będącej następną generacją algorytmu. Polega ona na stworzeniu nowej grupy osobników, z do której należy część poprzedniej populacji oraz nowo wygenerowane dzieci skrzyżowanych osobników.

Po wybraniu nowej populacji powtarzana jest cała sekwencja dla nowych osobników, w celu utworzenia kolejnej generacji rozwiązań. Całość algorytmu powtarza się aż do spełnienia warunków zakończenia, wybierając jako wynik końcowy najlepiej przystosowanego osobnika, czyli rozwiązanie z najlepszym uzyskanym wynikiem.

1. Implementacja algorytmu

Napisany algorytm został napisany na podstawie standardowego schematu Algorytmu genetycznego:

Algorytm po wczytaniu danych z plików testowych generuje wektor **nowej populacji**. Populacja wygenerowana zostaje w ilości określonej przez użytkownika całkowicie losowo (tworząc drogę z losowo wybranych miast), po czym obliczana jest droga każdego wygenerowanego rozwiązania. Na końcu populacja jest sortowana pod kątem długości drogi, czyli najlepszego wyniku wylosowanej drogi. Każda droga zaczyna się od wierzchołka startowego o indeksie 0.

Po utworzeniu pierwszej populacji, następuje pętla działając aż nie zostanie spełniony warunek końcowy działania algorytmu. W programie zastosowane zostało **ograniczenie czasowe** ustalane w programie. Wewnątrz tej pętli znajduje się główna część algorytmu.

1. Procedura testowania

Podczas przeprowadzania testów nie były uruchomione żadne dodatkowe aplikacje. A wszystkie procesy w tle zostały ograniczone do minimum poprzez wyłączenie aplikacji w tle oraz odłączenie komputera od Internetu w celu uzyskania jak najlepszych wyników.

Jako danych wejściowych użyłem danych z plików dostępnych na stronie: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/atsp/index.html>. Testowane były trzy pliki TSP: br17.atsp, ftv\_35.atsp, ft\_70.atsp

Każdy z plików posiada inny zestaw wierzchołków, w różnej liczbie oraz o różnej odległości względem siebie.

Parametry do testowania algorytmu były ustawiane na stałe dla każdej instancji problemu takie same. Parametry wynosiły odpowiednio:

* **Czas działania**: 60 sekund;
* **Rozmiar populacji początkowej**: 40, 80, 120,160,200,240;
* **Wartość mutacji**:

Wyniki pomiarów przedstawiają wartość uzyskaną dla badanej kombinacji. Poniżej zamieściłem wykresy oraz tabele otrzymanych wyników.

3.Wyniki moich badań oraz testowań

br17.atsp czas 60s

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lp. | Wielkość populacji | Koszt trasy |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 40 | 147 |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 80 | 132 |  |  |  |  |  |  |  |  |  |  |  |
| 3 | 120 | 120 |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 160 | 110 |  |  |  |  |  |  |  |  |  |  |  |
| 5 | 200 | 103 |  |  |  |  |  |  |  |  |  |  |  |
| 6 | 240 | 95 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ftv35.atsp czas 60s | |  |  |  |  |  |  |  |  |  |  |  |  |
| Lp. | Wielkość populacji | Koszt trasy |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 40 | 4322 |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 80 | 4254 |  |  |  |  |  |  |  |  |  |  |  |
| 3 | 120 | 4103 |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 160 | 4076 |  |  |  |  |  |  |  |  |  |  |  |
| 5 | 200 | 3929 |  |  |  |  |  |  |  |  |  |  |  |
| 6 | 240 | 3837 |  |  |  |  |  |  |  |  |  |  |  |
| ft70.atsp czas 60s | |  |  |  |  |  | | | | | | | |
| Lp. | Wielkość populacji | Koszt trasy |  |  |  |
| 1 | 40 | 67698 |  |  |  |
| 2 | 80 | 67389 |  |  |  |
| 3 | 120 | 67273 |  |  |  |
| 4 | 160 | 67132 |  |  |  |
| 5 | 200 | 67061 |  |  |  |
| 6 | 240 | 66824 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  | | | | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

5.Wnioski

Podczas analizy wyników głównie skupiłem się na wpływie wielkość populacji na odnalezienie optymalnej drogi, podczas porównywania wykresów można zauważyć, że zwiększanie wielkości populacji ma duże znaczenie na odnalezienie trasy o najniższym koszcie. Ponadto zaobserwowałem znaczący wpływ na odnalezienie trasy o najniższym koście badając pliki (mapy) o dużym zagęszczeniu.