Politechnika Wrocławska

Projektowanie efektownych algorytmów

Dawid Słodkowski

1. Wstęp teoretyczny

Algorytmy populacyjne do podjęcia decyzji o kolejnej próbce funkcji, wykorzystują informację z kilku poprzednich próbek.

Przykład (Operator krzyżowania) - Operator krzyżowania w algorytmie genetycznym uśrednia 2 próbki o wysokiej jakości funkcji, w nadziei że pomiędzy nimi funkcja również będzie wysokiej jakości.

Celem tego projektu jest rozwiązanie omawianego w poprzednich projektach problemu komiwojażera za pomocą algorytmu populacyjnego.

Algorytm populacyjny jest stworzony aby odwzorowywał proces selekcji naturalnej czyli przetrwanie bytów najbardziej pod to pasujących. Standardowy algorytm populacyjny jest podzielony na pięć faz:

1. Tworzenie pierwotnej populacji
2. Obliczenie kondycji
3. Wybieranie najlepszych genów
4. Przechodzenie dalej
5. Mutacja aby wprowadzić inne warianty.

Algorytm ten może być używany przy różnych problemach (takich jak właśnie problem komiwojażera). W tym algorytmie miasta są uznawane jako geny, string jest nazywany chromosomem, a kondycja jest równa ścieżce wszystkich wspomnianych miast.

2. Opisanie algorytmu

1. Zainicjowanie populacji
2. Określić kondycje chromosomu
3. Wykonywać aż do zakończenia pracy programu:
    1. Wybór rodziców
    2. Wykonanie przejścia i mutacja
    3. Obliczenie kondycji nowej populacji
    4. Dodać ją do puli genów.

Pseudokod:

```
    Initialize procedure GA{
  Set cooling parameter = 0;
  Evaluate population P(t);
  While( Not Done ){
      Parents(t) = Select_Parents(P(t));
      Offspring(t) = Procreate(P(t));
      p(t+1) = Select_Survivors(P(t), Offspring(t));
      t = t + 1;
  }
}
```

3. Czas pracy programu

Program może wyszukać rozwiązania dla dowolnych plików. W implementacji użyłem pliku „br17.atsp” a do pomiaru czasu użyłem biblioteki <ctime>

```
clock_t poczatek = clock();
kod(tab);
clock_t koniec = clock();
```

Pomiar	Czas w sekundach
1	0,058
2	0,039
3	0,031
4	0,039
5	0,031
6	0,039
7	0,039
8	0,031
9	0,031
10	0,031

4. Wnioski

W pracy został zaproponowany równoległy algorytm populacyjny dla problemu komiwojażera.
Z przeprowadzonych eksperymentów wynika, że zastosowanie algorytmu równoległego pozwala na kilku krotne przyspieszenie obliczeń w stosunku do metody podziału i ograniczeń oraz programowania dynamicznego.

5. Bibliografia

https://www.geeksforgeeks.org/traveling-salesman-problem-using-genetic-algorithm/?ref=gcse