

Algorytmy Metaheurystyczne - Algorytm genetyczny

Piotr Maciejończyk

25 stycznia 2024

1 Algorytm genetyczny

Algorytm genetyczny to metaheurystyka inspirowana procesem ewolucji biologicznej, która znajduje zastosowanie w rozwiązywaniu problemów optymalizacyjnych. Inspirację czerpie z genetyki, selekcji naturalnej oraz dziedziczenia genetycznego, a jego celem jest znalezienie najlepszego rozwiązania danego problemu w przestrzeni poszukiwań.

Algorithm 1 Algorytm genetyczny

```
1: procedure ALGORYTMGENETYCZNY
2:   while Warunek stopu nie jest spełniony do
3:     WYZNACZENIE POPULACJI POZATKOWEJ
4:     EWALUACJA I SELEKCJA
5:     KRZYZOWANIE
6:     MUTACJA
7:   end while
8: end procedure
```

W algorytmie genetycznym, jednostki populacji są kodowane za pomocą genów, które reprezentują potencjalne rozwiązania problemu. Proces ewolucji obejmuje ocenę przystosowania każdego osobnika, selekcję rodziców, krzyżowanie genetyczne, mutację genów oraz zastępowanie starej populacji nową populacją. Proces ten powtarzany jest do momentu spełnienia warunku stopu, który może być związany z osiągnięciem satysfakcjonującego rozwiązania lub przekroczeniem ustalonej liczby iteracji.

2 Moja implementacja

Podczas wyznaczania populacji początkowej posłużyłem się algorytmem wyspowym, gdzie co pewien okres czasu umożliwiam im wymianę informacji pomiędzy sobą. Moja implementacja procesu selekcji rodziców opiera się na 'ruletce' i lepiej przystosowane osobniki mają większą szansę zostać wybranymi. Krzyżowanie testowałem na dwóch jego rodzajach: PMX (Partially Mapped Crossover) oraz OX (Order Crossover). Podczas procesu mutacji wykonuję dwie procedury: invert pomiędzy dwoma wierzchołkami oraz swap dwóch losowych wierzchołków. Dodatkowo, mój algorytm zmodyfikowałem o dodanie metaheurystyki przeszukiwania lokalnego *LocalSearch* z losowym sąsiedztwem.

Parametrami, które obrałem są następująco:

1. Liczba wysp: 8
2. Rozmiar populacji: 32
3. Liczba epok: 5
4. Liczba iteracji w epoce: 100
5. Prawdopodobieństwo krzyżowania: 0.8
6. Prawdopodobieństwo mutacji: 0.05

3 Porównanie dwóch sposobów krzyżowania

Skorzystałem z dwóch sposobów krzyżowania - PMX (Partially Mapped Crossover) oraz OX (Order Crossover). Poniżej przedstawiam wyniki dla obu z nich:

	PMX		OX	
Filename	The best solution weight	Average solution weight	The best solution weight	Average solution weight
xql131	662	685.56	648	681.45
xqg237	1202	1232.38	1197	1228.57
pma343	1707	1750.35	1697	1746.31
pka379	1644	1713.11	1655	1710.26
bcl380	1815	1880.88	1822	1881.08
pbl395	1469	1530.17	1489	1532.02
pbk411	1554	1597.76	1556	1597.46
pbn423	1588	1628.8	1578	1629.17
pbm436	1674	1725.18	1653	1722.25
xql662	2948	3049.41	2949	3047.36
xit1083	4329	4436.53	4352	4444.33
icw1483	5367	5531.95	5468	5580.57
djc1785	7780	7933.48	7795	7969.56
dcb2086	8677	8858.83	8744	8936.75
pds2566	10454	10741	10645	10855.9

Tabela 1: Wyniki dla PMX oraz OX

Suma wartości najlepszych rozwiązań podzielonych przez ich odpowiednią liczbę wierzchołków dla każdego ze sposobów wynosi:

1. $PMX = 63.971$
2. $OX = 64.051$

Zatem w moich rozważaniach to PMX zwracał lepsze rozwiązania, przez co skorzystałem z niego podczas tworzenia statystyk.

4 Zestawienie wyników dla zaimplementowanych algorytmów

Wyniki zestawilem dla największych przykładów:

Filename	Simulated Annealing	Tabu Search	Local Search	Genetic Algorithm
xit1083	4157	4256	4173	4329
icw1483	4918	5075	5010	5367
djc1785	6954	7159	6997	7780
dcb2086	7508	7729	7543	8677
pds2566	8693	9041	8862	10454

Tabela 2: Najlepsze otrzymane rozwiązania dla zadanych algorytmów

Filename	Simulated Annealing	Tabu Search	Local Search	Genetic Algorithm
xit1083	4257.21	4375.2	4271.19	4436.53
icw1483	5056.45	5218.19	5122.12	5531.95
djc1785	7100.03	7312.91	7129.9	7933.48
dcb2086	7667.51	7905.26	7722.32	8858.83
pds2566	8913.45	9222.67	9011.12	10741

Tabela 3: Średnia wartość otrzymywanych rozwiązań dla zadanych algorytmów

Obliczyłem dla wszystkich wyników sumę wartości znalezionych najlepszych rozwiązań podzielonych przez ich liczbę wierzchołków i tak oto prezentują się wyniki:

1. Simulated Annealing - 18.037
2. Local Search - 18.221
3. Tabu Search - 18.591
4. Genetic Algorithm - 20.208