

Algorytm Genetyczny dla Problemu Komiwożacza

Bohdan Tkachenko 256630

February 2, 2024

1 Wprowadzenie

W pracy przedstawiono analize algorytmu genetycznego zastosowanego do problemu komiwożacza (TSP). Celem badania jest ocena skuteczności algorytmu w kontekście różnych zestawów danych TSP. Algorytm genetyczny jest techniką metaheurystyczna, która wykorzystuje mechanizmy inspirowane naturalną ewolucją biologiczną, takie jak selekcja, krzyżowanie i mutacja, do efektywnego wyszukiwania rozwiązań w przestrzeni problemów optymalizacyjnych.

1.1 Implementacja Algorytmu

Implementacja algorytmu genetycznego dla problemu komiwożacza (TSP) wykorzystuje zaawansowane techniki ewolucyjne, algorytm wyspowy oraz zrównoleglenie obliczeń, aby efektywnie eksplorować przestrzeń rozwiązań. Proces implementacyjny obejmuje następujące etapy:

1. **Inicjalizacja początkowej populacji:** Każda wyspa inicjuje losową populację tras, gdzie każda trasa jest permutacją miast.
2. **Ocena osobników:** Każdy osobnik w populacji jest oceniany za pomocą funkcji celu, która mierzy całkowitą długość trasy.
3. **Selekcja:** Używając selekcji turniejowej, wybierani są rodzice, którzy będą uczestniczyć w krzyżowaniu. Selekcja turniejowa zapewnia, że lepiej przystosowane osobniki mają większą szansę na przekazanie swoich genów do kolejnych pokoleń.
4. **Krzyżowanie:** Stosowane są metody PMX i OX do generowania potomstwa z wybranych rodziców. Każda z tych metod zachowuje kluczowe cechy trasy od obu rodziców, jednocześnie wprowadzając nową różnorodność.
5. **Mutacja:** Losowa zamiana dwóch miast w trasie (mutacja przez zamianę) wprowadza dodatkową różnorodność, co pozwala na eksplorację nowych rozwiązań.

6. **Lokalna optymalizacja:** Algorytm 2-opt jest stosowany do każdego nowego osobnika w celu poprawy jego jakości poprzez lokalne przeszukiwanie przestrzeni rozwiązań.
7. **Algorytm wyspowy i wielowatkowość:**
 - (a) Każda wyspa (subpopulacja) jest przetwarzana równolegle w osobnym watku, co pozwala na niezależną ewolucję różnych części populacji.
 - (b) Po określonej liczbie epok przeprowadzana jest wymiana najlepszych osobników między wyspami. Wybrane osobniki są kopiowane do innych wysp, gdzie zastępują one niektóre z istniejących osobników, co zwiększa różnorodność genetyczną.
 - (c) Wymiana ta jest zsynchronizowana między watkami, aby upewnić się, że wszystkie wyspy ukończyły swoje iteracje przed rozpoczęciem wymiany.
8. Proces ewolucji jest **powtarzany** przez ustaloną liczbę pokoleń lub do osiągnięcia warunku stopu.

1.2 Parametryzacja Algorytmu

Parametry algorytmu genetycznego odgrywają kluczową rolę w jego działaniu i efektywności. W badaniu wykorzystano:

- **Rozmiar populacji (50):** Określa liczbę tras w każdej subpopulacji na wyspie. Większa populacja zwiększa różnorodność genetyczną, ale wymaga więcej obliczeń.
- **Liczba wysp (30):** Reprezentuje liczbę niezależnych subpopulacji. Dzielenie populacji na wyspy pomaga uniknąć przedwczesnej konwergencji i zachować różnorodność genetyczną.
- **Liczba epok (10) i Liczba iteracji w epoce (100):** Epoka to cykl, po którym następuje wymiana osobników między wyspami. W każdej epoce wykonuje się określoną liczbę iteracji algorytmu genetycznego, co pozwala na intensywną ewolucję w obrębie wyspy.
- **Prawdopodobieństwo krzyżowania (0.8):** Szansa na krzyżowanie dwóch osobników. Wysokie prawdopodobieństwo sprzyja wymianie materiału genetycznego i tworzeniu nowych rozwiązań.
- **Prawdopodobieństwo mutacji (0.3):** Określa częstość mutacji w populacji. Mutacja wprowadza nowe cechy do genotypu osobników, zwiększając różnorodność populacji.

1.3 Metody Krzyżowania

PMX (Partially Mapped Crossover) i **OX (Order Crossover)** to dwie wykorzystane metody krzyżowania, które mają na celu utworzenie nowego osobnika z cech obu rodziców, zachowując przy tym optymalną kolejność odwiedzania miast.

PMX działa poprzez wybór segmentu genów z jednego rodzica i mapowanie pozostałych miast z drugiego rodzica, zachowując ich względne pozycje, co zapobiega duplikacji miast w trasie.

OX również wybiera segment z jednego rodzica, ale wypełnia pozostałe miejsca miastami z drugiego rodzica w kolejności ich występowania, co również utrzymuje niepowtarzalność miast.

1.4 Mutacja

Mutacja **Swap** polega na losowej zamianie dwóch miast miejscami w trasie, co ma na celu wprowadzenie dodatkowej różnorodności i eksplorację nowych obszarów przestrzeni rozwiązań.

1.5 Lokalna Optimalizacja

2-opt jest technika lokalnej optymalizacji stosowana do każdego nowo wygenerowanego lub zmodyfikowanego osobnika. Polega na systematycznym sprawdzaniu każdej pary krawędzi w trasie i zamianie ich, jeśli prowadzi to do skrócenia długości trasy.

2 Wyniki

Instancja	Najlepsze	Średnia
xit1083	3837	3983.5
icw1483	4664	4857.9
djc1785	6568	6718.7
Dcb2086	7089	7239.3
Pds2566	8154	8337.2

Table 1: Wyniki algorytmu memetycznego dla problemu TSP

3 Podsumowanie Algorytmów

3.1 Podsumowanie Algorytmów

Algorytm Genetyczny(Memetyczny): Ten algorytm zapewnia najdokładniejsze rozwiązanie ze wszystkich przetestowanych metod. Dzięki wykorzystaniu obliczeń równoległych, algorytm nie potrzebuje aż tak dużo czasu. Ale jednak jest wolny.

Teoretycznie, elastyczność w parametryzacji pozwala na dostosowanie go do specyficznych zestawów danych, co może wpłynąć na jakość wyników.

Symulowane Wyżarzanie: Jest to algorytm, który w rozsądnym czasie oferuje rozwiązania wystarczająco dobre. Najszybszy z metaheurystyk.

Taboo Search: Czasami dostarcza wyniki trochę lepsze w porównaniu z Symulowanym Wyżarzaniem, ale wymaga więcej czasu na ich uzyskanie. Jego deterministyczny charakter sprawia, że jest to metoda przewidywalna w swoich rezultatach.

LocalSearch: Wśród porównywanych metaheurystyk, LocalSearch generuje rozwiązania o najniższej dokładności i wymaga stosunkowo dłuższego czasu na wykonanie, co czyni go mniej efektywnym wyborem dla problemu TSP.

Cykl TSP na podstawie MST: Ta metoda szybko dostarcza rozwiązania, które są gwarantowane 2-OPT. Jej główną zaletą jest szybkość działania.

3.2 Tabelka koncowa

Instancja	Optymalne	LocalSearch	SA	TS	Genetyczny (Memetyczny)
xil1083	3558	4343.45	4200.8	4180	3983.5
lwc1483	4416	5388.82	5158.2	5010.2	4857.9
djc1785	6115	7384.34	7177.7	7151.3	6718.7
Dcb2086	6600	8183.4	7767.0	7723.2	7239.3
Pds2566	7643	9396.42	9031.3	8950.7	8337.2

Table 2: Porównanie średnich wyników różnych algorytmów dla problemu TSP