

Algorytm genetyczny jest heurystyką inspirowaną ewolucją biologiczną. W każdej iteracji algorytmu tworzona jest nowa populacja osobników, która jest następnie poddawana krzyżowaniu i mutacji.

Populacje rozdzielone są na wyspy, które są od siebie izolowane. Co pewną liczbę epok populacje są wymieniane między wyspami. W ten sposób algorytm może uniknąć zatrzymania się w lokalnym optimum.

Dodatkowo rozwiązania mogą być ulepszone przez algorytm lokalnego przeszukiwania, np. algorytm Local Search.

Implementacja algorytmu genetycznego:

Wczytywanie danych: Wczytywane są pliki zawierające zestawy punktów z współrzędnymi (x, y) . Plik zawiera informacje o numerze punktu oraz jego współrzędnych (x, y) .

Inicjalizacja grafu: Na podstawie wczytanych punktów tworzony jest graf pełny, gdzie każdy wierzchołek odpowiada jednemu z punktów, a krawędzie między nimi mają wagę równą odległości euklidesowej między nimi.

Algorytm genetyczny: Dla każdego zestawu punktów (wczytanego z pliku) przeprowadzany jest algorytm genetyczny.

- Inicjalizacja populacji: Tworzona jest populacja początkowa, gdzie każde rozwiązanie (cykl) jest losowo generowane.
- Ewaluacja populacji: Dla każdego rozwiązania w populacji obliczana jest wartość funkcji celu, czyli suma wag krawędzi w cyklu.
- Krzyżowanie (OX): Wybierani są rodzice z populacji, następnie stosowany jest operator krzyżowania Order Crossover (OX) do wygenerowania potomstwa. Operator polega na wybraniu losowego fragmentu jednego z rodziców i przepisaniu go do dziecka. Następnie wypełniane są brakujące wierzchołki z drugiego rodzica w kolejności występowania w nim, ale bez powtórzeń.
- Mutacja: Losowo wybrane rozwiązania podlegają mutacji, która polega na odwróceniu kolejności pewnych elementów w cyklu.
- Lokalne przeszukiwanie: Dla każdego rozwiązania w populacji przeprowadzane jest lokalne przeszukiwanie w celu poprawy jakości rozwiązania.
- Selekcja: Wybierane są najlepsze rozwiązania, które zostaną przekazane do kolejnej iteracji.

Wyniki: Dla każdego zestawu danych wyświetlane są wyniki działania algorytmu genetycznego, takie jak: średnia kosztów, minimalny koszt znaleziony w populacji oraz najlepsze znalezione rozwiązanie.

Parametry:

- **Liczba iteracji algorytmu genetycznego (iter):** Ustawiono na wartość 100. Oznacza to, że algorytm genetyczny będzie wykonywany 100 razy, czyli przeprowadzi 100 generacji populacji.
- **Rozmiar populacji (populationSize):** Ustawiono na wartość 100. Określa liczbę rozwiązań (cykli) w każdej populacji.
- **Prawdopodobieństwo krzyżowania (crossProbability):** Ustawiono na wartość 0.8. Określa szansę, że dwa wybrane rozwiązania zostaną skrzyżowane (operator krzyżowania OX).
- **Prawdopodobieństwo mutacji (mutationProbability):** Ustawiono na wartość 0.1. Określa szansę, że dane rozwiązanie zostanie poddane mutacji.

Wyniki dla algorytmu genetycznego z krzyżowaniem OX i LocalSearch:

	Minimum Cost	Average Cost
xqf131	603	646.32
xqg237	1279	1322.09
pma343	1757	1860.83
pka379	1715	1841.36
bcl380	2075	2246.68
pbl395	1654	1777.11
pbk411	1756	1906.28
pbn423	1816	1931.81
pbm436	2039	2171.53
xql662	3427	3715.13
xit1083	5522	5904.52
icw1483	6878	7475.47
djc1785	9590	10258.8
dcb2086	11040	11954
pds2566	13249	14086.5

Populacja początkowa jest zatem tworzona przez wygenerowanie losowych permutacji wierzchołków i obliczenie dla każdego osobnika kosztu cyklu. To stanowi bazę dla algorytmu genetycznego, który następnie będzie ewoluował tę populację w kierunku coraz lepszych rozwiązań.

Selekcja turniejowa: Losowo wybierane są pary osobników (rodziców) z populacji posortowanej. Pary te stanowią "turnieje" z udziałem kilku osobników. Wybierany jest osobnik o najniższym koszcie z każdego turnieju. Pary te są następnie używane do krzyżowania.

Operator krzyżowania Order Crossover (OX) jest techniką używaną w algorytmach genetycznych do krzyżowania dwóch osobników (rozwiązań) w celu uzyskania potomstwa.

Kroki, jakie wykonuje operator OX:

1. **Wybór dwóch punktów cięcia:** Losuje się dwa punkty cięcia w chromosomie rodzica. Warto zauważyć, że OX jest operatorem dla permutacji, a chromosom reprezentuje permutację kolejności wierzchołków.
2. **Kopiowanie fragmentów rodzica 1:** Kopiowane są elementy znajdujące się pomiędzy dwoma punktami cięcia z rodzica 1 do potomka.
3. **Kopiowanie reszty elementów z rodzica 2:** Pozostałe miejsca w potomku są uzupełniane elementami z rodzica 2 w kolejności, w jakiej występują, zaczynając od punktu cięcia końcowego i kontynuując cyklicznie.
4. **Uzyskanie potomka 2:** Analogicznie do kroku 2 i 3, uzyskuje się drugiego potomka przez kopiowanie fragmentów rodzica 2 i uzupełnianie ich elementami z rodzica 1.

Operator OX pomaga zachować informacje o kolejności elementów w chromosomie, co jest istotne w przypadku problemu komiwojażera, gdzie kolejność wierzchołków ma znaczenie.

Mutacja jest realizowana w następujący sposób:

1. **Losowy wybór dwóch punktów w chromosomie:** Mutacja polega na odwróceniu kolejności fragmentu chromosomu. Dlatego losuje się dwa różne punkty w chromosomie, które będą określać fragment podlegający mutacji.
2. **Odwrócenie fragmentu chromosomu:** Elementy pomiędzy wybranymi punktami są odwracane, zmieniając ich kolejność. Dzięki temu wprowadza się lokalną zmianę w rozwiązaniu, co może pomóc w uniknięciu utknięcia w lokalnym minimum.

Lokalne przeszukiwanie w algorytmie genetycznym dla problemu komiwojażera jest używane w celu dalszej poprawy jakości rozwiązania. Po krzyżowaniu i mutacji, stosowanie lokalnego przeszukiwania pozwala na eksplorację lokalnych obszarów przestrzeni rozwiązań w poszukiwaniu lepszych konfiguracji.

Warunek stopu dla całego algorytmu genetycznego może być zdefiniowany jako określona liczba iteracji lub jako czas trwania algorytmu. Kod działa przez określoną liczbę iteracji, co jest warunkiem stopu.