

Problem komiwojadzera dla algorytmów genetycznych

Szymon Skoczylas, 250072

11 lutego 2021

1 Problem komiwojadzera

Problem komiwojadzera (ang. travelling salesman problem, TSP) jest to problem optymalizacyjny który zazwyczaj jest przedstawiany następująco- mamy n miast, musimy odwiedzić je wszystkie dokładnie raz, zaczynając i kończąc w tym samym mieście. Celem jest znalezienie takiej kolejności odwiedzania miast, żeby całkowita droga była jak najmniejsza. Niech odległość między miastem i i j będzie zadana jako $d_{i,j}$, minimalizowaną funkcją [2] celu będzie:

$$\sum_{i=0}^n \sum_{j=0, j \neq i}^n d_{i,j} x_{i,j}$$

gdzie,

$$x_{i,j} = \begin{cases} 1 & \text{jeśli ścieżka z miasta } i \text{ do } j \\ 0 & \text{oth.} \end{cases}$$

Formalnie można to też przedstawić jako szukanie cyklu Hamiltona o minimalnej wadze w pełnym grafie ważonym, gdzie wierzchołki to miasta a krawędzie to drogi z odpowiadającą sobie wagą. Problem ten jest NP-trudny. Istnieją również różne modyfikacje TSP. Symetryczna wersja zakłada, że $d_{i,j} = d_{j,i}$, podczas gdy wersja niesymetryczna nie daje nam takiej gwarancji.

2 Algorytm Genetyczny

Algorytmy genetyczne to grupa heurystyk ewolucyjnych. Jego działanie wzorowane jest na procesie selekcji naturalnej. Algorytmy takie stosuje się gdy szukanie dokładnego algorytmu lub rozwiązania jakiegoś problemu jest zbyt trudne do osiągnięcia w sposób analityczny. Zastosowanie algorytmu genetycznego do rozwiązania jakiegoś problemu nie daje nam gwarancji na znalezienie globalnie najlepszego rozwiązania, ale umożliwia nam zbliżenie się do niego z, zazwyczaj, akceptowalnym marginesem błędu. Przedstawmy podstawowe operatory stochastyczne algorytmów genetycznych [2]:

1. Selekcja- jest to wybór osobników które przechodzą do następnego pokolenia z uwagi na swoją wartość, czyli miarę tego jak dobrze są dostosowane.
2. Rekombinacja- jest to krzyżowanie dwóch osobników z populacji w celu otrzymania nowego osobnika który dziedziczy część cech po swoich rodzicach.
3. Mutacja- jest to losowe modyfikowanie osobnika w sposób niezależny od innych. Pozwala to zwiększyć różnorodność w populacji, oraz w razie wpadnięcia w lokalne optimum, wydostać się z niego.

Istotną kwestią jest również dobór kodowania rozwiązania, czyli sposób przedstawienia osobnika. Kodowanie powinno umożliwiać dokładne przedstawienie rozwiązania. Przykłady możliwych kodowań: ciąg bitowy, permutacje elementów zbioru dopasowań. Dla ustalonej formy osobnika możemy stosować różne formy rekombinacji takie jak

na przykład rekombinacja jednopunktowa lub dwupunktowa. Mutacji możemy dokonywać za pomocą permutacji lub transpozycji takich ciągów. Przedstawmy sobie ogólny schemat algorytmu genetycznego [2]:

1. Dobór parametrów takich jak: ilość osobników w populacji, warunek końca etc.
2. Ustalenie rozwiązań startowych- początkowa populacja
3. Symulacja ekosystemu aż do osiągnięcia ustalonego warunku końca:
 - (a) Ocena wartości każdego osobnika za pomocą funkcji celu
 - (b) Wybór części osobników która będzie krzyżowana
 - (c) Krzyżowanie wybranych najsilniejszych osobników
 - (d) Dokonanie ewentualnych losowych mutacji
 - (e) Ocena nowopowstałych osobników i generacja nowej populacji
4. Wybór najlepszego rozwiązania

W kontekście TSP, zazwyczaj przyjmuje się, że funkcją oceny osobników, która ma być minimalizowana, jest ta przedstawiona w sekcji 1. Populacja, zazwyczaj, składa się z osobników które przedstawiają premutacje zbioru miast do odwiedzenia.

3 TSP dla algorytmu genetycznego bazowanego na entropii

W [3] autorzy, w celu znalezienia rozwiązania problemu komiwojadzera, przedstawiają propozycję algorytmu genetycznego zmodyfikowanego w taki sposób, aby zapewnić większą różnorodność genetyczną populacji ("Entropy based genetic algorithm, EBGA"). Ma to zapobiec utknięciu w lokalnym optimum. Zjawisko takie może nastąpić kiedy różnorodność populacji wraz z kolejnymi pokoleniami maleje. Spadek różnorodności genetycznej pociąga za sobą pogorszenie efektów operacji genetycznych takich jak krzyżowanie. Aby zminimalizować ryzyko wystąpienia takich sytuacji autorzy użyli pojęcia entropii informacji. Wykorzystują je w celu określenia różnorodności populacji.

W EBGA rozwiązanie t_k , czy też osobnik, jest przedstawiane jako permutacja n miast które należy odwiedzić ($t_k : (c_1, c_2 \dots c_n)$). Operacja mutacji w tym algorytmie jest realizowana jako transpozycja dwóch pozycji w permutacji będącej osobnikiem. Rekombinację zrealizowano za pomocą krzyżowania cyklicznego (CX). Taka operacja znajduje liczbę cykli pomiędzy parą rodziców (x i y). Następnie do jednego z potomków pierwszy cykl jest kopiowany z rodzica x , potem drugi cykl z rodzica y , cykl trzeci z x i tak dalej. Przeprowadźmy krzyżowanie dwóch przykładowych permutacji tą właśnie metodą (gdzie x, y to rodzice, a x', y' to nowopowstałe osobniki):

$$x : (8, 4, 7, 3, 6, 2, 5, 1, 9, 0)$$

$$y : (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$$

Pierwszy znaleziony cykl to 8, 9, 0. Drugi to 4, 1, 7, 2, 5, 6. Trzeci cykl to sama 3. Następnie aby stworzyć potomstwo, kopiujemy wartości cyklu pierwszego z x do x' . Wartości z y będą skopiowane do y' . Wartości drugiego cyklu będą skopiowane z x do y' i z y do x' . Wartości cyklu 3 będą rozdysponowane podobnie tak jak te z cyklu pierwszego. Oto efekt:

$$x' : (8, 1, 2, 3, 4, 5, 6, 7, 9, 0)$$

$$y' : (0, 4, 7, 3, 6, 2, 5, 1, 8, 9)$$

Funkcja służąca do waluacji osobników została zrealizowana w następujący sposób:

$$eval(t_k) = \frac{1}{\sum_{i=1}^n d(c_i, c_{i+1}) + d(c_n, c_1)}$$

gdzie $d(c_i, c_{i+1})$ to dystans z miasta c_i do c_{i+1} , oraz $k \in \{1, 2 \dots pop.size\}$.

Określanie zróżnicowania populacji będzie wykorzystywać odpowiadające sobie pola we wszystkich permutacjach (dla $t_k : (c_1, c_2, \dots, c_n)$, każde c_i należące do dwóch różnych osobników będzie porównywane). Różnorodność pojedynczego pola c_i jest zdefiniowana następująco [3]:

$$H_i = \sum_{c \in C} pr_{ic} \ln pr_{ic}$$

Gdzie C to zbiór wszystkich miast, $pr_{ic} = \frac{na_{ic}}{pop.size}$, na_{ic} to ilość wystąpień miasta c w komórce permutacji pod indeksem i . Miara różnorodności pojedynczego pola została zapożyczona z koncepcji entropii informacyjnej. H_i dąży do największej wartości (tj. $\ln(pop.size)$) kiedy, każde $c \in C$ pojawia się jednolicie na tych samych miejscach w permutacji. H_i dąży do minimum (tj. 0), gdy liczba wystąpień danego c na określonym miejscu w permutacji jest znacznie większa niż liczba wystąpień innych. Ewaluacja różnorodności odbywa się w następujący sposób- porównujemy wartość H_i z $\ln 2$. Jeśli $H_i \leq \ln 2$ to w komórkach o indeksie i panuje niska różnorodność. Powtarzamy tę czynność dla wszystkich odpowiadających sobie komórek w permutacjach. Jeśli całkowita liczba komórek l w permutacji, dla których zachodzi $H_i \leq \ln 2$, jest większa, bądź równa $\frac{n}{a}$ to różnorodność populacji uznawana jest za zbyt niską.

Aby zwiększyć różnorodność autorzy zdefiniowali procedurę *Mod – Pop*. Gdzie a jest parametrem z zakresu $[2, 5]$. Dokładna wartość a jest określana na podstawie stosunku ilości miast i rozmiaru populacji. Wraz ze wzrostem wartości tego parametru, rośnie prawdopodobieństwo poprawy populacji. Przedstawmy sobie procedurę [3]:

Procedura *Mod – Pop*:

Krok 1: Wybierz m osobników z populacji.

m jest losową liczbą z zakresu $[\frac{pop.size}{a}, pop.size - 1]$

Krok 2: Dla każdego wybranego osobnika przeprowadź wymianę genów,
w każdej komórce dla której zachodzi $H_i < \ln 2$

Za każdym razem po zakończeniu procedury oceniamy różnorodność populacji jeszcze raz, tak jak opisano wyżej. Procedurę powtarzamy do momentu, aż osiągniemy zadawalające zróżnicowanie w populacji. Jako metodę selekcji osobników wykorzystano metodę ruletki. Metoda ta wybiera najsilniejszego osobnika na podstawie obszaru który zajmowałby na kole ruletki. Odbywa się to tak: wybieramy grupę osobników i na podstawie wartości funkcji celu dla każdego z nich przypisujemy adekwatny obszar na kole. Im osobnik silniejszy, tym większy obszar dostaje, co zwiększa jego szanse na wybranie jako rodzic.

Sam schemat EBGA wygląda następująco [3]:

Procedura *EBGA*:

Krok 1: Ustawienie parametrów takich jak: ilość miast n , rozmiar populacji $pop.size$, prawdopodobieństwo krzyżowania pr_c , prawdopodobieństwo mutacji pr_m , oraz maksymalną ilość generacji $maxgen$. Na początku ustawiamy także licznik generacji $gen = 0$, oraz najlepszej znalezionej wartości $maxeval = 0$.

Krok 2: Losowe generowanie populacji początkowej o rozmiarze $pop.size$.

Krok 3: Ocena początkowych rozwiązań:

Krok 3. 1: Wylicz wartość osobnika na podstawie równania

Krok 3. 2: Jeśli $maxeval < max\{eval(t_k)\}$
wtedy $s* = argmax\{eval(t_k)\}$ oraz $maxeval = max\{eval(t_k)\}$

Gdzie $argmax$ jest argumentem przechowywującym t_k mającego największą wartość oceny.

Krok 4: Krzyżowanie CX na osobnikach z prawdopodobieństwem pr_c .

Krok 5: Mutacja (transpozycja) na wybranych osobnikach z prawdopodobieństwem pr_m .

Krok 6: Selekcja $pop.size$ osobników ze starej populacji i nowopowstałych rozwiązań do następnej generacji za pomocą metody ruletki.

Krok 7: Polepszanie zróżnicowania populacji:

Krok 7. 1: Określenie zróżnicowania populacji.

Krok 7. 2: Niech $cout = 0$ oraz $I = \{\emptyset\}$

Porównaj H_i dla $i \in \{1, 2, \dots, n\}$ z wartością $ln2$.

Jeśli $H_i \leq ln2$ wtedy dodaj indeks i do zbioru I i $coutn = count + 1$.

Powtarzaj ten krok dla $i = 1, \dots, n$.

Krok 7. 3: Jeśli $\frac{n}{a} \leq count$ przejdź do **Krok 7.4**,
w przeciwnym wypadku **Krok 8**

Krok 7. 4: Weź losową liczbę $rand$ z zakresu $[\frac{pop.size}{a}, pop.size - 1]$.

Wybierz $rand$ losowych osobników z populacji.

Dla wybranych osobników losowo wymień geny wyspecyfikowane w I (Procedura: Mod-Pop).

Idź do **Krok 7.1**.

Krok 8: Jeśli $gen = maxgen$ to zwróć $s*$ i zakończ działanie,
w przeciwnym wypadku niech $gen = gen + 1$ i idź do **Krok 3**

Przedstawmy wyniki otrzymane za pomocą tego algorytmu. Eksperymenty zostały przeprowadzone dla parametrów:

1. Rozmiar populacji: $pop.size = 20$
2. Maksymalna liczba populacji: $maxgen = 20000$
3. Prawdopodobieństwo krzyżowania: $pr_c = 0.8$
4. Prawdopodobieństwo mutacji: $pr_m = 0.2$
5. Parametr a : $a = 3$

Algorytm *EBGA* został uruchomiony 20 razy. Dla porównania, przeprowadzono testy dla zwykłego algorytmu genetycznego (schemat taki sam jak *EBGA*, ale z wyłączeniem Kroku 7). Okazało się, że dla takiego zestawu danych wejściowych *EBGA* potrzebuje znacznie mniejszego czasu, aby znaleźć tak samo dobre rozwiązanie jak zwykły algorytm genetyczny [3].

4 TSP dla algorytmu genetycznego i optymalizacji wielokryterialnej

W [1] autorzy przedstawiają nam wielokryterialną optymalizację problemu komiwojadzera za pomocą algorytmu genetycznego (ang. Multi-Objective TSP- MOTSP). Optymalizowanymi wartościami są całkowity koszt i długość przebytej trasy.

Optymalizacja paru wartości jednocześnie odbywać się będzie w sensie Pareta. Celem jest znalezienie takiego optimum, aby polepszenie jednej wartości nie mogło się odbyć inaczej niż kosztem pogorszenia drugiej. Formalnie, minimalizowana funkcja, wygląda tak [1]:

$$f(x) = \begin{cases} f_1(x) = c_1^{x(n),x(1)} + \sum_{i=1}^{n-1} c_1^{x(i),x(i+1)} \\ f_2(x) = c_1^{x(n),x(1)} + \sum_{i=1}^{n-1} c_1^{x(i),x(i+1)} \\ f_3(x) = c_1^{x(n),x(1)} + \sum_{i=1}^{n-1} c_1^{x(i),x(i+1)} \end{cases}$$

Gdzie n to liczba miast, c_j wartość macierzy kosztu dla j , oraz x to cykliczna permutacja n miast. Miasta są indeksowane kolejnymi liczbami naturalnymi: $\{1, 2, \dots, n\}$, a odległość między miastami i, j jest zdefiniowana jako $distance[i][j]$ ($i \neq j$), koszt natomiast wyraża się jako: $cost[i][j]$ ($i \neq j$). Autorzy rozważaniu poddali symetryczną wersję TSP, gdzie $distance[i][j] = distance[j][i]$. W takim wypadku, dla n miast, wszystkich możliwych rozwiązań jest:

$$\frac{(n-1)!}{2}$$

Macierze kosztu i dystansu są określone za pomocą 2-wymiarowej metryki euklidesowej. Dystans dla punktów (x_1, y_1) i (x_2, y_2) jest określany w następujący sposób [1]:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

W algorytmie autorów operacja mutacji jest realizowana jako transpozycja, krzyżowanie za pomocą rekombinacji jednopunktowej, selekcja jako turniej (z populacji wybieramy losowo grupę osobników, z niej tego o najlepszym dostosowaniu i to on przechodzi do krzyżowania- proces powtarza się parę razy). Schemat algorytmu genetycznego użytego przez autorów [1]:

Procedura *GA*:

Krok 1: Losowanie populacji P początkowych rozwiązań

Krok 2: Dla każdego osobnika w P określ jego przystosowanie

Krok 3: Dla każdego i od 1 do maksymalnej liczby generacji:

Krok 3. 1: Wybierz losowo operację o : mutacja lub krzyżowanie

Krok 3. 2: Jeśli $o ==$ krzyżowanie, wybierz losowo dwóch rodziców z P i przeprowadź krzyżowanie.
Jeśli $o ==$ mutacja, wybierz losowo jednego osobnika z P i zastosuj na nim mutację

Krok 3. 3: Dla nowopowstałego osobnika przeprowadź jego waluację.

Krok 3. 4: Jeśli nowopowstały osobnik jest lepszy od aktualnie najgorszego wprowadź go do populacji zamiast najgorszego

Krok 4: Zwróć najlepsze znalezione rozwiązanie.

Autorzy przeprowadzili eksperymenty używając benchmarkowego zbioru danych z TSPLIB. Testowane instance problemu to instance Krolak dla 100, 150, oraz 200 miast. Testowane również były instance Euclid dla 100, 300 i 500 miast. Wyniki uzyskane [1] przez algorytm zaprezentowany przez autorów były bardzo zbliżone do najlepszych rozwiązań, w sensie Pareta, znalezionych dotychczas dla każdego ze zbiorów testowych. Wszystkie testowe uruchomienia w których zbliżenie się do optimum było osiągane algorytm kończył w akceptowalnym czasie.

5 Wnioski

Powołując się na powyższe przykłady, można stwierdzić, że jedną z poważniejszych trudności w szukaniu rozwiązania dla problemu komiwojadzera za pomocą algorytmu genetycznego jest utrzymanie wystarczającej różnorodności genetycznej populacji. Gdy różnorodność spada bardzo zmniejszają się nasze szanse na znalezienie lepszego rozwiązania. Rozwiązaniem tego problemu może być dobranie odpowiednio wysokiego prawdopodobieństwa mutacji, może to znaczenie zwiększyć zróżnicowanie genetyczne osobników. Jeśli jednak współczynnik mutacji będzie za duży, nasz algorytm genetyczny może generować zbyt losowe pokolenia.

W [3] autorzy zastosowali mechanizm różnicowania populacji bazujący na entropii informacji, jak pokały ich doświadczenia, osiągnięcie dobrych wyników zajmowało mniej czasu niż zwykłemu algorytmowi genetycznemu. Wszystko to za sprawą zwiększenia różnorodności populacji.

Zaproponowany przez twórców w [1] algorytm genetyczny do rozwiązania wielokryterialnego problemu komiwojadzera, jak można zauważyć analizując wyniki testów, wykazał się dużą wydajnością i dokładnością. Osiągając wyniki zbliżone do tych najlepszych znalezionych dotychczas na używanym zbiorze testowym.

Literatura

- [1] Tintu George and T Amudha. Genetic algorithm based multi-objective optimization framework to solve traveling salesman problem. In *Advances in Computing and Intelligent Systems*, pages 141–151. Springer, 2020.
- [2] P. Syga. Algorytmy Metaheurystyczne. https://cs.pwr.edu.pl/syga/courses/amh/amh_lec.html, 2020. Odwiedzono: 15-06-2020.
- [3] Y. Tsujimura and M. Gen. Entropy-based genetic algorithm for solving tsp. In *1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No.98EX111)*, volume 2, pages 285–290 vol.2, 1998.