

Projekt 1 - Podstawy sztucznej inteligencji

Aleksandra Kukawka (300246), Bartłomiej Binda (300188)

Instytut Informatyki, Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska

I. OPIS PROBLEMU

Nasze zadanie to znalezienie najkrótszego cyklu łączącego kilka europejskich miast stosując algorytm ewolucyjny na podstawie **tabeli** z odległościami między 37 wybranymi miastami w Europie. Problem ten jest w literaturze definiowany jako Problem Komiwojażera i polega na znalezieniu najkrótszego cyklu Hamiltona w pełnym grafie ważonym. Komiwojażer musi odwiedzić n miast (w naszym przypadku 37), przy czym każde tylko raz. Kończy swoją podróż w mieście, z którego wystartował. Koszt podróży z miasta do miasta, czyli waga krawędzi w grafie, jest liczbą nieujemną. Łączny koszt to suma kosztów poszczególnych krawędzi trasy, którą chcemy minimalizować.

Problem komiwojażera jest problemem NP-zupełnym, co oznacza, że nie znaleziono dotychczas algorytmu wielomianowego, który by ten problem rozwiązał. Już dla 25 miast mamy około 15.5 septyliona rozwiązań. Z tyloma możliwymi kombinacjami, niemożliwe jest wręcz znalezienie globalnego optymalnego rozwiązania. Zatem naszym celem nie jest znalezienie optimum globalnego, tylko jednego z wielu przybliżonych optymalnych rozwiązań. W każdej kolejnej iteracji chcemy znaleźć rozwiązanie, które jest lepsze od poprzedniego najlepszego. Dlatego używamy algorytmu ewolucyjnego, który na podstawie zdefiniowanej funkcji celu, szuka czołowego wyniku.

II. PRACA NAD PROJEKTEM

W tej sekcji opisany jest podział zadań, ważne decyzje dotyczące projektu, narzędzia i biblioteki oraz instrukcja uruchomienia programu tak, aby każdy mógł zobaczyć wizualizację dla kolejnych iteracji.

A. Podział zadań

1) *Aleksandra Kukawka*: projekt algorytmu, utworzenie repozytorium na GitHubie, implementacja mutacji i krzyżowań, praca nad estetyką dokumentacji, wykresy, opracowanie wniosków.

2) *Bartłomiej Binda*: projekt algorytmu, pobranie danych, utworzenie klas osobnik, populacja i travellingSalesman, praca nad estetyką i komentowanie kodu, testowanie, opracowanie wniosków.

B. Ważne decyzje projektowe

- 1) *Populację początkową* wyznaczamy w sposób losowy.
- 2) Używamy *reprodukcji rangowej (turniejowej)*.
- 3) *Krzyżujemy* metodą Order Crossover. Stosujemy krzyżowanie dwupunktowe. Wybieramy losowo dwa punkty i przepisujemy trasę, która jest między tymi punktami z rodzica P1 do potomka. Następnie przepisujemy wierzchołki z rodzica P2 do potomka pomijając wierzchołki, które potomek dostał już od rodzica P1. Drugiego potomka generujemy zamieniając rodziców P1 i P2.
- 4) *Mutujemy* z prawdopodobieństwem, które pokazało najlepsze wyniki w naszych eksperymentach. Używamy do mutacji dwóch sposobów - pierwszego, gdy losowa liczba z rozkładu $U(0, 1)$ jest mniejsza, bądź równa 0.5, drugiego - w przeciwnym przypadku. Pierwszy sposób (Inversion) polega na zamianie dwóch losowych wierzchołków i odwróceniu kolejności miast między tymi wierzchołkami. Drugi (Scramble) polega na permutacji podzbioru danego osobnika.
- 5) Stosujemy *sukcesję elitarną*.
- 6) Algorytm kończy się po ustalonej przez nas liczbie iteracji.

C. Wykorzystane narzędzia i biblioteki

- Rozwiązanie problemu w postaci programu jest napisane w języku Python.
- Biblioteki, które zostały użyte, to: NumPy, Pandas, Random oraz Seaborn i Matplotlib (do wizualizacji).
- Kod algorytmu został umieszczony w repozytorium na GitHubie.
- Dokumentacja została napisana w oprogramowaniu LaTeX.
- Pracę nad projektem usprawniała aplikacja Trello.

D. Instrukcja uruchomienia algorytmu

Po uruchomieniu terminala w środowisku Linux w katalogu, w którym znajdują się pliki programu, należy wpisać:

```
python main.py
```

Na ekranie będą wyświetlać się numery iteracji wraz z najlepszą dotychczas znaną trasą w kilometrach. Po zakończeniu programu wyświetli się wykres podsumowujący i znaleziona najlepsza trasa.

III. CELE I WNIOSKI

A. Główny cel i teza

Naszym *głównym celem* jest minimalizacja długości trasy przebytej przez komiwojażera. W naszym rozwiązaniu zamieniamy minimalizację na maksymalizację, ponieważ naszą funkcją celu jest 1 dzielone przez długość trasy. W reprodukcji lepszym osobnikiem jest osobnik z większą wartością funkcji celu, czyli z krótszą trasą.

B. Eksperymenty i ich wyniki

Z powodu losowości naszego algorytmu zdecydowaliśmy się dla każdej mierzonej wartości wykonywać algorytm 5 razy, podając w raporcie średnią długość wyliczonej drogi (zaokrągloną do najbliższej liczby całkowitej) oraz długość najkrótszej i najdłuższej znalezionej drogi. Eksperymenty wykonywane są w ustalonej poniższej kolejności. W każdym następnym eksperymencie uwzględniamy najlepsze wartości danych parametrów z poprzednich eksperymentów. Początkowe parametry ustalone przez nas do pierwszego eksperymentu to:

- Liczba permutacji w mutacji Scramble: 5,
- Prawdopodobieństwo mutacji: 0.2,
- Prawdopodobieństwo krzyżowania: 0.8,
- Liczba osobników w reprodukcji turniejowej: 2,
- Sukcesja elitarna z $k=1$,
- Rozmiar populacji: 15,
- Liczba iteracji: 2000.

1) Badanie wpływu rodzaju mutacji:

Zaczelismy od eksperymentu z mutacjami. Celem tego eksperymentu jest zbadanie wpływu rodzaju mutacji na wynik optymalizacji. Zakładamy, że algorytm będzie zachowywał się najlepiej przy połączonych dwóch mutacjach (obie z prawdopodobieństwem 0.5).

Rodzaj mutacji	Średni wynik	Najlepszy wynik	Najgorszy wynik
Inversion	18 206	17 408	18 785
Scramble	18 242	17 712	18 952
Dwie mutacje (razem)	17 790	17 469	18 071

Tablica I

WPLYW ZMIANY PRAWDOPODOBIEŃSTWA MUTACJI.

Potwierdziliśmy naszą tezę. Okazało się, że najlepszą średnią długość trasy uzyskujemy przy używaniu dwóch rodzajów mutacji. Dzieje się tak dlatego, że korzystanie z dwóch różnych mutacji zapewnia większą różnorodność, więc algorytm jest mniej podatny na optima lokalne.

2) Badanie wpływu zmiany prawdopodobieństwa mutacji:

W tym momencie używamy już dwóch rodzajów mutacji z poprzedniego eksperymentu. Celem tego eksperymentu jest zbadanie wpływu prawdopodobieństwa mutacji na wynik. Zakładamy, że prawdopodobieństwo mutacji powinno zawierać się w przedziale od 0.2 do 0.5, ale sprawdzamy też mniejsze liczby.

Prawdopodobieństwo mutacji	Średni wynik	Najlepszy wynik	Najgorszy wynik
0.05	17 803	17 452	18 018
0.1	17 577	17 075	18 234
0.2	17 790	17 469	18 071
0.3	18 177	17 662	19 015
0.4	18 498	17 933	19 163

Tablica II

WPLYW ZMIANY PRAWDOPODOBIEŃSTWA MUTACJI.

Mimo naszych założeń co do przedziału mutacji okazało się, że najlepsze wyniki dostajemy, gdy prawdopodobieństwo jest poniżej wyznaczonego przez nas przedziału. Najlepszy średni wynik uzyskaliśmy dla prawdopodobieństwa 0.1. Od tego momentu w dalszych eksperymentach będziemy używać właśnie prawdopodobieństwa 0.1.

Liczba osobników	Średni wynik	Najlepszy wynik	Najgorszy wynik
2	16 989	16 924	17 085
3	16 935	16 596	17 461
5	16 936	16 721	17 087

Tablica III

WPLYW ZMIANY LICZBY OSOBNIKÓW W REPRODUKCJI TURNIEJOWEJ.

3) Badanie wpływu liczby osobników w reprodukcji turniejowej:

Celem tego eksperymentu jest zbadanie wpływu liczby osobników, które biorą udział w reprodukcji turniejowej, na wynik optymalizacji. Zakładamy, że najlepszy wynik uzyskamy przy $k=2$.

Średnie wyniki są do siebie bardzo zbliżone, więc nie możemy być pewni przewagi jednego z wyborów parametru k nad innymi. Jednakże trzeba wziąć pod uwagę, że uśredniamy tylko na 2000 iteracji i przy takim samym rozmiarze populacji, a większej ilości iteracji, dla 5 nasz algorytm byłby bardziej podatny na minima lokalne, niż w przypadku k równego 2 i 3. Dlatego od następnego eksperymentu używamy $k=3$, które uzyskało również najlepszy pojedynczy wynik.

4) Badanie wpływu sukcesji elitarniej:

Celem tego eksperymentu jest zbadanie wpływu sukcesji elitarniej na wynik optymalizacji. Zakładamy, że algorytm będzie zachowywał się najlepiej dla sukcesji elitarniej z $k=1$.

Sukcesja elitarna	Średni wynik	Najlepszy wynik	Najgorszy wynik	Średni czas [s]
0	25 005	23 321	27 036	916.14
1	16 912	16 629	17 112	1 056.01
2	16 946	16 815	17 092	1 203.51
5	17 162	16 902	17 299	1 699.53

Tablica IV

WPLYW SUKCESJI ELITARNEJ.

Tabela dokładnie opisuje, jak korzystnie sukcesja elitarna wpływa na końcowy wynik. Przy nieużywaniu jej, najlepszy wynik, który udało się uzyskać to 23 321, który na pewno nie jest blisko optimum globalnego. Z sukcesją elitarną uzyskujemy o wiele lepsze wyniki, gdyż nie gubimy wtedy naszego najlepszego osobnika. Różnice w najlepszych średnich wynikach są minimalne, więc wybieramy wartość $k=1$, gdyż w tym wariancie algorytm wykonuje się krócej.

5) Badanie wpływu zmiany rozmiaru populacji:

Celem tego eksperymentu jest zbadanie wpływu zmiany rozmiaru populacji na wynik optymalizacji. Zakładamy, że wraz ze zwiększeniem rozmiaru populacji (oraz później - wraz ze zmianą liczby iteracji) będziemy mieli coraz krótszą trasę.

Rozmiar populacji	Średni wynik	Najlepszy wynik	Najgorszy wynik	Średni czas [s]
5	17 722	17 672	17 780	352.95
15	17 206	17 032	17 492	1045.03
25	17 144	17 033	17 298	1723.17
50	16 990	16 788	17 225	3469.79
100	16 970	16 734	17 265	6927.66

Tablica V

WPLYW ZMIANY ROZMIARU POPULACJI.

Zgodnie z naszymi oczekiwaniami - im większa ilość osobników w populacji, tym krótsza średnia długość trasy. Wynika to z tego, że przy większej liczbie osobników, mamy o wiele większą różnorodność. Jednakże zwróćmy uwagę na czas wykonania. Mierzyliśmy go dla 2000 iteracji (tak jak podano w założeniach początkowych). Czas jest liniowo zależny od rozmiaru populacji. Dla populacji składającej się ze 100 osobników jest to około dwóch godzin. Różnica między średnimi wynikami dla populacji 100 i 50 wynosi tylko 20 km, a algorytm wykonuje się 2 razy dłużej. Postanowiliśmy zostać do końca przy 15 osobnikach w populacji, gdyż w ten sposób możemy uzyskać bardzo dobre wyniki w rozsądnym czasie.

6) Badanie wpływu zmiany liczby iteracji:

Celem tego eksperymentu jest zbadanie wpływu liczby na wynik. Zakładamy, że wraz ze zwiększeniem liczby iteracji, wyniki będą coraz lepsze.

Potwierdzamy naszą tezę. Liczba iteracji bezpośrednio wpływa na najlepszą trasę. Zaskoczyło nas to, że dla zarówno dla iteracji 5000 jak i 10000 najlepszy uzyskany wynik ma identyczną wartość - 16514 km. Czy to może oznaczać, że znaleźliśmy optimum globalne? Niestety nie możemy wyciągnąć takiego wniosku, ale skoro algorytm dwukrotnie utknął w tym miejscu, to możemy powiedzieć, że jesteśmy już blisko.

Liczba iteracji	Średni wynik	Najlepszy wynik	Najgorszy wynik	Średni czas [s]	Średni czas 2* [s]
1 000	17 307	16 721	17 671	530.28	718.11
2 000	17 153	16 766	17 395	1 052.93	1424.76
5 000	16 757	16 514	16 887	2 627.18	3 561.64
10 000	16 624	16 514	16 844	5 230.43	7 321.26

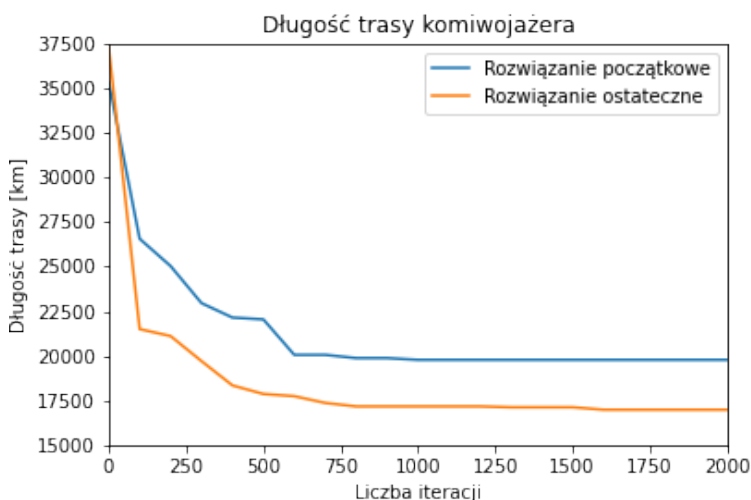
Tablica VI
WPŁYW ZMIANY LICZBY ITERACJI.

*Czas mierzony na drugim komputerze, analiza we wnioskach(III-C).

Ciekawym zaobserwowanym przez nas zjawiskiem jest także znacząca różnica w pomiarze czasu na dwóch różnych komputerach, którą opiszemy w następnej sekcji.

C. Wnioski

Dostosowując najlepsze algorytmy, udało nam się z wartości ponad 19000 uzyskać w najlepszym przypadku wartość 16514 km. Przykładowe porównanie szukania najlepszej ścieżki dla algorytmów z optymalnymi i nieoptymalnymi hiperparametrami jest pokazane poniżej. Nasze eksperymenty i ukazany wykres udowadniają jak ważne dla algorytmów ewolucyjnych jest dostosowanie hiperparametrów do danego problemu. Jednak, podczas procesu należy pamiętać o losowości tych algorytmów i dlatego nie można się kierować pojedynczymi wynikami, tylko agregować kilka lub kilkanaście iteracji na przykład za pomocą średniej wartości ścieżki.



Do mierzenia czasu używaliśmy dwóch komputerów stacjonarnych. Oba korzystają z tego samego systemu operacyjnego - Linux OS (Ubuntu 20.04.2.0 LTS). Czas uruchomienia algorytmu genetycznego jest przede wszystkim zależny od procesora, więc dzięki naszym wynikom możemy porównać procesory (1 - Ryzen 5 5600X i 2 - Ryzen 5 3600). Ryzen 5 3600 jest średnio 35% wolniejszy. Co więcej, dzięki 6 rdzeniowej konstrukcji procesorów mogliśmy wykonywać kilka eksperymentów naraz bez zauważalnych różnic w czasach działania algorytmów.

IV. PODSUMOWANIE

Podczas realizacji projektu dowiedzieliśmy się, do czego i jak używa się algorytmów ewolucyjnych. Nauczyliśmy się implementować taki algorytm i zapoznaliśmy się z całą jego strukturą. Mogliśmy przełożyć całą swoją wiedzę teoretyczną z wykładu na praktykę. Projekt był dla nas bardzo ciekawy, ponieważ mogliśmy bezpośrednio zaobserwować, że nawet niewielkie zmiany parametrów mogą znacząco wpłynąć na wynik i bardzo zaangażowaliśmy się w sprawdzanie różnych konfiguracji parametrów oraz w prowadzenie eksperymentów.