

Dokumentacja projektu MDVRPTW-Europe

Bartosz Młynarski 2025

Sterowanie całym projektem zajmuje się plik „Program.cs” w języku C# znajdujący się w folderze głównym projektu. Korzysta on z innych napisanych plików napisanych w tym samym języku znajdujących się w folderze src. Są to:

- „Algorithm.cs” – główny plik algorytmu ewolucyjnego i algorytmu własnego,
- „Colour.cs” – prosty plik umożliwiający dodawanie kolorowych napisów w konsoli,
- „CreateTest.cs” – plik odpowiadający za generowanie testów,
- „Data.cs” – plik zajmujący się przetwarzaniem map OSM w potrzebne dane,
- „Graph.cs” – plik zajmujący się obsługą grafu na podstawie map OSM,
- „MakeTests.cs” – plik wykonujący testy do przetestowania algorytmu ewolucyjnego,
- „PreparingData.cs” – plik pobierający potrzebne dane do uruchomienia algorytmu,
- „Visualization.cs” – plik zajmujący się wizualizacją graficzną działania algorytmów.

Pobranie i przygotowanie map

By móc pierwszy raz odpalić algorytm, należy pobrać potrzebne mapy na dysk urządzenia własnego. W pliku „Program.cs” znajduje się zmienna o nazwie `countriesList`, która odpowiada za to, na jakich krajach będziemy wykonywać operację. Obok nazw regionów znajduje się przybliżona ilość pamięci potrzebna do przygotowania danych dla danego miejsca. Domyślnie od komentowane są Węgry i Słowacja, przez stosunkowo małą wymaganą ilość pamięci do ich obsługi, bo około 0,5 GB na jeden z tych krajów. Za pierwszym uruchomieniem programu należy od komentować wybrane przez siebie kraje i od komentować linijkę kodu:

```
await pData.PD(countriesList);
```

Powyższa komenda pobiera wymagane mapy i przygotowuje je do algorytmu. Proces może trwać trochę czasu, w zależności od wybranych danych, szybkości łącza internetowego oraz samego komputera. Nie należy go przerywać przed pojawieniem się zielonego komunikatu w konsoli o treści „**----DANE PRZYGOTOWANE----**”. Aby uruchomić program, należy wejść w konsoli w folder „/MDVRPTW-Europe/” i wpisać w niej komendę „dotnet run”. Następnie po pobraniu map można z powrotem za komentować podaną linijkę, gdyż w wypadku używania tych samych krajów w algorytmie, nie będzie trzeba przygotowywać danych kolejny raz.

Uruchomienie algorytmu

W kolejnej części pliku „Program.cs” znajdują się przykładowe dane do uruchomienia algorytmu i ta część dokumentacji będzie poświęcona dokładnemu opisowi, co należy zrobić, żeby ten algorytm uruchomić. Całym algorytmem zajmuje się klasa o nazwie Algorithm i to jej konstruktor odpowiada za uruchomienie całej procedury w zależności od dobranych przez nas parametrów, którymi są:

- numberPopulation – parametr odpowiadający za wielkość populacji w AE,
- numberEpochs – liczba epok w AE,
- percentLeft - % elity w AE [1 – 100],
- mutationSwapPower – moc pierwszego operatora mutacji [0 - 1],
- mutationScramblePower – moc drugiego operatora mutacji [0 - 1],
- crossoverSwapPower – moc pierwszego operatora krzyżowania [0 – 1],
- mutationSwapFreq – częstotliwość pierwszego operatora mutacji [0 – 1],
- mutationScrambleFreq – częstotliwość drugiego operatora mutacji [0 – 1],
- crossoverSwapFreq – częstotliwość pierwszego operatora krzyżowania [0 – 1],
- crossoverLinearOrderFreq – częstotliwość drugiego operatora krzyżowania [0 – 1],

Suma wszystkich częstotliwości mutacji musi być równa 1!

- packagesPerDriver – liczba paczek przypadająca na pierwszego kierowcę,
- mode – tryb pracy algorytmu, do wyboru to:
 - „warehouses” – tryb z AE do połączeń między klientami a magazynami,
 - „sortOffices” – tryb z algorytmem własnym między magazynami a sortowniami,
 - „all” – oba powyższe tryby uruchomione naraz.
- showWarehousesMap – pokazanie mapy wizualnej dla wyniku AE,
- showSortOfficesMap – pokazanie mapy wizualnej dla wyniku algorytmu własnego,
- warehousesMaxHours – parametr określający ograniczenie czasowe kierowcy dla AE,
- sortOfficesMaxHours – ograniczenie czasowe dla kierowcy dla algorytmu własnego,
- testName – nazwa wybranego testu, obecnie dostępne to:
 - „Test1kP10C” – paczka zaprogramowana dla 10 krajów dla 1000 paczek,
 - „Test2kP10C” – paczka zaprogramowana dla 10 krajów dla 2000 paczek,
 - „Test3kP10C” – paczka zaprogramowana dla 10 krajów dla 3000 paczek,
 - „Test4kP10C” – paczka zaprogramowana dla 10 krajów dla 4000 paczek.
 - „Test5kP10C” – paczka zaprogramowana dla 10 krajów dla 5000 paczek.

Aby uruchomić algorytm należy odmentować liniijkę:

```
Algorithm algorithm = new Algorithm( ... )
```

Tworzenie własnych danych testowych

Zainicjowany program posiada również możliwość tworzenia własnych danych testowych. Funkcjonalność ta jest wykonywana dzięki klasie CreateTest. Aby utworzyć nowy test należy wywołać metodę CreateNewTest(testName, nop, countriesList), gdzie:

- testName – nazwa nowo tworzonych danych testowych podana w string,
- nop - liczba paczek w nowym teście,
- countriesList – omówiona wcześniej lista krajów.

Proces tworzenia nowych danych testowych również może być złożony czasowo, więc dopóki całość się nie wykona należy nie przerywać. Szablon tworzenia nowego testu zaprezentowany na dole pliku „Program.cs”.