AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Praca dyplomowa inżynierska

Optymalizacja trasy z wykorzystaniem algorytmu przybliżonego

Autor: Rafał Siniewicz
Kieunek studiów: Automatyka i automatyka
Opiekun: dr inż. Joanna Kwiecień

Kraków, 2019/2020r.

Spis treści

1. Wstęp

Transport od zawsze pełnił bardzo ważną rolę w życiu społeczeństw. W przeciągu wieków zmieniały się środki transpotu, jego metody, a także szybkość przemieszaczania ludzi oraz towarów. Za każdym jednak razem wyzwanie stanowiło jak najoptymalniejsze przeprowadzenie procesu transportu. Szczególnie w dzisiejszych czasach zaplanowanie tras w taki sposób, aby zminimalizować czas ich pokonania- co wiąże się z mniejszymi kosztami oraz emisjami spalin- jest bardzo cenne. Również zwyklejsze sytuacje, jak np. zaplanowanie dojazdu do pracy (komunikacją miejską lub samochodem) wymagają od wielu osób skorzystania z odpowiednich narzędzi (świadczy o tym chociażby popularność google maps czy aplikacji jakdojade) przeszukujących wiele możliwych tras i wybierających tą najlepszą. Widać zatem jak dużą rolę odgrywa w życiu niemalże każdego dobranie odpowiedniej trasy dla danej podróży.
Problem optymalizacji trasy jest bardzo powszechny i wielokrotnie studiowany, zwłaszcza w badaniach operacyjnych, teorii grafów czy logistyce i dziedzinach pokrewnych. W teorii najczęściej dąży się do uzyskania jak najkrótszej trasy, choć problem można rozwinąć o wiele różnych czynników, które wpływają na przebieg trasy, np. jakość nawierzchni czy określona pojemność pojazdów.

1.1. Cel pracy

Praca ma na celu stworzenie aplikacji komputerowej wraz z interfejsem graficznym służącej do obliczania optymalnej trasy na podstawie odpowiednich danych. Optymalna trasa zostanie obliczona przy pomocy odpowiednich algorytmów opisanych w poniższym wstępie teoretycznym. Program będzie również wyświetlał przebiegi tras w formie graficznej wraz z zaznaczeniem poszczególnych punktów na mapie. Całość ma stanowić ujednolicony i łatwy w użyciu system, w którym można ustalać parametry, takie jak, np. ilość pojazdów i ich pojemności, współrzędne geograficzne punktów, a także uzyskać informacje o długości trasy, jej przebiegu i innych statystykach dotyczących rozwiązywanego problemu.

1.2. Zakres pracy

W pracy zagadnienie optymalizacji trasy rozważono dla problemu marszrutyzacji, czyli VRP (Vehicle Routing Problem) oraz CVRP (Capacity Vehicle Routing Problem). Projekt skupia się głównie na dwóch czynnikach decydujących o optymalności: długości trasy oraz zapełnieniu pojazdów ładunkiem.
Przy rozwiązywaniu problemu marszrutyzacji korzystano z algorytmów przybliżonych, a konkretniej algorytmu genetycznego, przede wszystkim ze względu na złożoność problemu, gdyż przy problemach bardziej złożonych dobrze sprawdzają się algorytmy przybliżone. Algorytm genetyczny wybrano również ze względu na jego uniwersalność, możliwość uzyskania stosunkowo dobrych wyników w krótkim czasie oraz prostą koncepcję podejścia do rozwiązania.

1.3. Wstęp teoretyczny

1.3.1. Problem poszukiwania optymalnej trasy

Na podstawie [1]:
Optymalizacja- proces polegający na poszukiwaniu przy pomocy metod matematycznych najlepszego (ze względu na wybrane kryteria) rozwiązania pewnego problemu, z uwzględnieniem określonych ograniczeń.
Problem poszukiwania optymalnej trasy należy do ważnych zagadnień z zakresu planowania transportu. Optymalizacja trasy może skupiać się na różnych czynnikach, takich jak: długość trasy, koszty, czas podróży, itp. Zazwyczaj optymalizacja ma na celu znalezienie najkrótszej trasy.

Problem kombinatoryczny

Na podstawie [1]:
Problem kombinatoryczny jest to problem, który jest opisany przy pomocy skończonej ilości parametrów wejściowych, a także warunków wymaganych do spełnienia przez dane wyjściowe.
Problemy kombinatoryczne dzielimy na problemy:
- decyzyjne
- optymalizacyjne
- przeszukiwania

Problem decyzyjny jest to skończony zbiór parametrów, np. funkcje, zbiory, grafy oraz pytanie na które można odpowiedzieć "tak" lub "nie".

Problem optymalizacyjny to problem, który wymaga znalezienia ekstrumum określonej funkcji celu na podstawie podanego problemu kombinatorycznego.

Problem przeszukiwania jest to problem, który wymaga kompletnego rozwiązania lub odpowiedzi przeczącej jeśli rozwiązanie nie istnieje. Każdy problem decyzyjny i optymalizacyjny można przedstawić w postaci problemu przeszukiwania.

Złożoność obliczeniowa

Na podstawie [1]:
Problemy można podzielić na 2 główne klasy:
- **łatwe (P)**, tj. problemy, które można rozwiązać w czasie wielomianowym
- **trudne (NP)**, tj. problemy, które nie mają rozwiązań w czasie wielomianowym lub mniejszym, czyli zadania o złożoności wykładniczej lub większej

Istnieje jeszcze podklasa problemów trudnych (NP), tzw. problemy **NP-zupełne (NPC)**, tj. problemy zupełne w klasie NP ze względu na redukcje wielomianowe. Problemy NPC to problemy, które należą do klasy NP przy czym dowolny problem należący do NP może być do nich zredukowany w czasie wielomianowym. Zatem jeśli udałoby się uzyskać rozwiązanie problemu NP-zupełnego w czasie wielomianowym, to można uzyskać rozwiązanie w czasie wielomianowym wszystkich problemów NP. Problemy NP-zupełne można postrzegać jako najtrudniejsze problemy klasy NP ze względu na wielomianową rozwiązywalność.

Problem komiwojażera

Poblem wyboru optymalnej trasy historycznie wywodzi się od problemu komiwojażera (Travelling Salesman Problem). Na podstawie [1]:
Problem komiwojażera – problem optymalizacyjny mający na celu znalezienie minimalnego cyklu Hamiltona w grafie pełnym ważonym. Jest to problem z kategorii NP- trudnych.
Słowne przedstawienie problemu:
Komiwojażer ma za zadanie odwiedzić n lokalizacji. Znane są odległości (czasem także cena i czas podróży) między każdą parą lokalizacji. Celem komiwojażera jest odwiedzenie wszystkich punktów (tylko raz) w jak najkrótszym czasie/ najkrótszą drogą/ najmniejszym kosztem, zaczynając w ustalonym punkcie i do niego wracając.
Matematyczne ujęcie problemu zgodnie ze wzorami Dantziga-Fulkersona-Johnsona [1]:
Oznaczmy punkty do odwiedzenia numerami od 1 do n. Dalej oznaczmy:

$$x_{ij} = \begin{cases} 1 & \text{jeśli trasa idzie z punktu i do j} \\ 0 & \text{w przeciwnym wypadku} \end{cases}$$

Przyjmijmy c_{ij} jako odległość z punktu i do j. Wtedy problem komiwojażera może być zapisany jako problem programowania liniowego:

$$min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij} : \tag{1}$$

$$0 \leq x_{ij} \leq 1 \quad i, j = 1, ..., n \tag{2}$$

$$min \sum_{i=1, i \neq j}^{n} x_{ij} = 1 \quad j = 1, ..., n \tag{3}$$

$$min \sum_{j=1, j \neq i}^{n} x_{ij} = 1 \quad i = 1, ..., n \tag{4}$$

$$\sum_{i \in Q} \sum_{j \in Q} x_{ij} \leq |Q| - 1 \quad \forall Q \not\subseteq \{1, ..., n\}, |Q| \geq 2 \tag{5}$$

Ostatnie ograniczenie formuły Dantziga-Fulkersona-Johnsona zapewnia, że nie ma żadnych podtras między początkowymi wierzchołkami, więc zwrócone rozwiązanie to pojedyncza trasa, a nie połączenie kilku mniejszych tras.

Problem komiwojażera można rozpatrywać jako:
1) symetryczny- gdzie dla dowolnych punktów X i Y odległość z X do Y jest taka sama jak z Y do X
2) asymetryczny- gdzie powyższe odległości mogą się różnić

Trudność w rozwiązaniu problemu komiwojażera polega na dużej ilości danych- wszystkich możliwych kombinacji dla n punktów jest $(n-1)!/2$. Zatem już przy 15 punktach otrzymujemy ponad $43 \cdot 10^9$ (czterdzieści trzy miliardy) możliwych tras do rozpatrzenia. Rozwinięciem problemu komiwojażera jest problem marszrutyzacji.

1.3.2. VRP i CVRP

Problem marszrutyzacji

Na podstawie [1] oraz [2]:
Problem marszrutyzacji polega na wyznaczeniu optymalnych tras dla określonej liczby środków transportu. Jest on rozszerzeniem problemu komiwojażera czy problemu chińskiego listonosza. Jest to problem z kategorii NP- trudnych. Flota pojazdów ma za zadanie odwiedzenie zadanych punktów, z uwzględnieniem przyjętych ograniczeń. W ramach optymalizacji bierze się pod uwagę sumaryczny koszt pokonanych tras.

VRP

Matematyczne sformułowanie podstawowego problemu komiwojażera (VRP):
Rozpatrywany jest graf nieskierowany $G = (V, E)$, gdzie:
- V oznacza zbiór wierzchołków
- E oznacza zbiór krawędzi, do których przypisane są koszty przejazdu (również inne parametry, np. czas)

Zadanie polega na zminimalizowaniu funkcji celu:

$$minC = \sum_{r\in R} \sum_{f\in V} \sum_{g\in V} c_{fg} x_{fgr} \tag{6}$$

gdzie:
r- pojazd należący do zbioru pojazdów R
f,g- wierzchołki grafu między, którymi porusza się pojazd
c_{fg}- koszt przejazdu między wierzchołkadmi f i g
x_{fgr}- zmienna przyjmująca wartości 0 lub 1 w zależności od tego czy pojazd wykonuje przejazd między wierzchołkami f i g (1 oznacza, że wykonuje)

Ograniczenia:
- istnieje tylko jedna baza początkowa i końcowa, które odwiedza dokładnie jeden pojazd r. Dla wierzchołków pośrednich ilość pojazdów wjeżdżających równa się ilości pojazdów wyjeżdżających

$$\forall r \in R \sum_{g\in E} x_{0,g,r} = 1 \quad \text{dla bazy początkowej} \tag{7}$$

$$\forall r \in R \sum_{f\in E} x_{f,n+1,r} = 1 \quad \text{dla bazy końcowej} \tag{8}$$

$$\forall r \in R \wedge \forall f \in V \sum_{f\in E} x_{f,z,r} - \sum_{g\in E} x_{z,g,r} = 0 \quad \text{dla wierzchołków pośrednich} \tag{9}$$

- dla każdego punktu jest przypisany dokładnie jeden pojazd, który zabiera z niego cały ładunek, ładunki nie są rozdzielane

$$\forall f \in V \sum_{g\in E} \sum_{r\in R} x_{fgr} = 1 \tag{10}$$

$$\forall f \in E \wedge \forall g \in E \wedge \forall r \in R x_{fgr} \in \{0,1\} \tag{11}$$

CVRP

CVRP (Capacity Vehicle Routing Problem) jest rozwinięciem problemu komiwojażera poprzez dodanie określonej pojemności pojazdów. Matematyczny model dla CVRP na podstawie [7] oraz [8]:

Funkcja celu:

$$min = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} y_{ij} + KC_{DO} \tag{12}$$

Przy ograniczeniach:

$$\sum_{i \in V} y_{ij} = 1, \forall i \in V \setminus \{O, D\}, i \neq j \tag{13}$$

$$\sum_{j \in V} y_{ij} = 1, \forall i \in V \setminus \{O, D\}, i \neq j \tag{14}$$

$$\sum_{j \in V} y_{Oj} = K \tag{15}$$

$$\sum_{i \in V} y_{iD} = K \tag{16}$$

$$\sum_{k \in K} x_{OD}^{k} = K \tag{17}$$

$$\sum_{k \in K} x_{Dj}^{k} = 0, \forall j \in V, j \neq 0 \tag{18}$$

$$U_i - U_j + q y_{ij} \leq q - d_j \quad \forall i, j \in V \setminus \{O, D\}, i \neq j \quad gdzie \quad d_i + d_j \leq q \tag{19}$$

$$d_i \leq u_i \leq q \quad \forall i \in V \setminus \{O, D\} \tag{20}$$

Dla zmiennych decyzyjnych:

$$y_{ij} \in \{0, 1\} \quad \forall i, j \in V \tag{21}$$

$$x_{ij}^{k} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K \tag{22}$$

gdzie:
i - miejsce odjazdu lub węzeł
j - węzeł przyjazdu
k - użyty pojazd, $k \in K$
|K| - ilość dostępnych pojazdów
V - zbiór węzłów
U - pojemność pojazdu
d_i - wymaganie klienta w węźle i
C_{ij} - koszt transportu lub dystans z węzła i do j
$x_{ij}^{k} = 1$ jeśli pojazd jest przypisany do przejścia
danej krawędzi z i do j (w przeciwnym wypadku $x_{ij}^{k} = 1$)
$y_{ij} = 1$ jeśli trasa z węzła i do j istnieje (w przeciwnym wypadku $y_{ij} = 0$)

q - pojemność pojazdu k
O - węzeł startowy pojazdów (baza) O_B
D - węzeł końcowy (magazyn) D_A
C_{DO} - koszt transportu lub dystans z D_A do O_B
KC_{DO} - koszt powrotu lub odległość wszystkich pojazdów z D_A do O_B

Inne warianty problemu marszrutyzacji

Ponadto istnieją różne warianty problemu marszrutyzacji, na przykład:
- Vehicle Routing Problem with Time Windows- uwzględnienie okien czasowych odbioru/ wysłania towaru
- Split Delivery Vehicle Routing Problem- możliwość obsługi jednego klienta przez kilka pojazdów
- Vehicle Routing Problem with Multiple Trips (VRPMT)- możliwość pokonania więcej niż jednej trasy przez jeden pojazd
- Vehicle Routing Problem with Pickup and Delivery (VRPPD)- wiele towarów należy przetransportować z niektórych miejsc odbioru do innych miejsc dostawy
- Vehicle Routing Problem with Multiple Trips (VRPMT)- pojazdy mogą pokonywać więcej niż jedną trasę
- Open Vehicle Routing Problem (OVRP)- nie jest wymagane, aby pojazdy wracały do onego miejsca (magazynu)
- inne

Powyższy problem można również rozwinąć o wiele innych czynników, jak np. kolejność odwiedzenia miejsc czy opcjonalnego odwiedzenia niektórych punktów lub funkcja kosztu może rozpatrywać różne parametry, np. czas wykonania zleceń czy ilość przewiezionego ładunku. Widać zatem, że problem marszrutyzacji jest zagadnieniem bardzo rozległym, a przy tym elastycznym, tzn. można go dostosować do wielu różnych problemów.

Rozwiązywanie problemu VRP

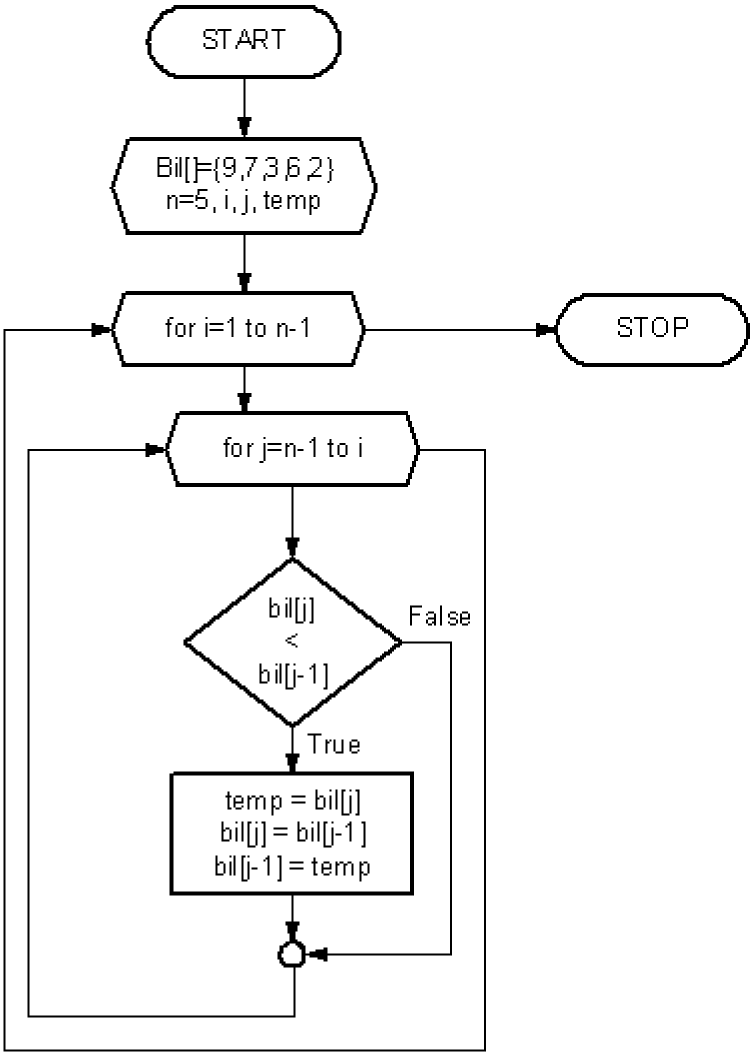
Ze względu na trudność w znalezieniu optymalnych rozwiązań dla problemów związanych z trasowaniem pojazdów w dużej skali znaczny wysiłek badawczy został poświęcony metaheurystyce, takiej jak algorytmy genetyczne, wyszukiwanie Tabu czy symulowane wyżarzanie. Niektóre z najnowszych i skutecznych metaheurystyk dotyczących problemów z trasowaniem pojazdów osiągają rozwiązania w granicach 0,5% lub 1% wartości optymalnej dla przypadków problemowych liczących setki lub tysiące punktów dostawy. Metody te są również bardziej niezawodne pod takim względem, że można je łatwiej dostosować do różnych ograniczeń bocznych. Stosowanie technik metaheurystycznych jest często uprzywilejowane w przypadku aplikacji na dużą skalę ze skomplikowanymi ograniczeniami i zestawami decyzji.

Na podstawie [1]:
Metaheurystykę stanowią ogólne ramy algorytmiczne, często inspirowane naturą (np. algorytm mrówkowy), które zostały utworzone w celu rozwiązywania złożonych problemów związanych z optymalizacją. Od kilku dekad stanowią coraz większy obszar badań związanych z optymalizacją. Metaheurystyka jest skuteczną alternatywą do bardziej klasycznych podejść w rozwiązywaniu problemów optymalizacyjnych, które zawierają w swoich wzorach matematycznych informacje nieokreślone konkretnie, stochastyczne i dynamiczne.

1.3.3. Algorytmy przybliżone

Algorytm

Na podstawie [3]:
Algorytm to dowolna poprawnie zdefiniowana procedura obliczeniowa, która przyjmuje pewną wartość lub zestaw wartości jako dane wejściowe i generuje pewną wartość lub zestaw wartości jak wynik. Algorytm jest zatem sekwencją kroków obliczeniowych, które przekształcają dane wejściowe do wyjściowych.
Można także postrzegać algorytm jako narzędzie do rozwiązywania odpowiednio określonego problemu obliczeniowego. Opis problemu określa ogólnie pożądaną relację między danymi wejściowymi, a wyjściowymi. Algorytm opisuje określoną procedurę obliczeniową do osiągnięcia tej relacji wejścia / wyjścia. Klasycznym przykładem, do którego wykorzystuje się algorytmy jest procedura sortowania, np. liczb. Sposób przedstawienia algoytmu może być różny- algorytm może być zapisany jako: schemat blokowy, pseudokod, lista kroków, itp. Przykładowy diagram blokowy ilustrujący algorytm sortowania bąbelkowego:



Rysunek1. Diagram blokowy algorytmu sortującego[1]

Algorytm jest poprawny, jeśli dla każdej instancji wejściowej kończy się z poprawnym wyjściem. Mówimy, że poprawny algorytm rozwiązuje dany problem obliczeniowy. Niepoprawny algorytm może w ogóle nie zostać zatrzymany dla niektórych instancji wejściowych lub może się zakończyć z niepoprawną odpowiedzią. Czasami jednak niepoprawne algorytmy mogą okazać się przydatne, jeśli możemy kontrolować ich poziom błędów. W dalszej części zostanie przybliżony pewien rodzaj algorytmów nazywany algorytmami przybliżonymi.

Algorytmy przybliżone

Na podstawie [3]:
Wiele problemów o znaczeniu praktycznym jest NP-zupełnych, czyli takich, dla których nie ma rozwiązań w czasie wielomianowym. Często są one jednak zbyt ważne, aby je porzucić. Z tego powodu szuka się rozwiązań przybliżonych. Istnieją co najmniej trzy sposoby obejścia NP-zupełności:
- jeśli rzeczywiste ilości danych wejściowych są małe, algorytm z wykładniczym czasem działania może być całkowicie zadowalający
- można wyodrębnić ważne specjalne przypadki, które można rozwiązać w czasie wielomianowym
- można zaproponować metody znalezienia prawie optymalnych rozwiązań w czasie wielomianowym (w najgorszym lub oczekiwanym przypadku)

W praktyce przybliżona optymalizacja jest na ogół wystarczająca. Algorytm, który zwraca prawie optymalne rozwiązanie nazywamy **algorytmem przybliżonym (aproksymacyjnym)**.

Współczynniki wydajności dla algorytmów przybliżonych

Na podstawie [3]:

Mówimy, że algorytm dla pewnego problemu ma wskaźnik aproksymacji $\rho(n)$, jeśli dla dowolnego wejścia wielkości n koszt C rozwiązania wytworzonego przez algorytm zawiera się w ramach $\rho(n)$ współczynnika kosztu C* optymalnego rozwiązania:

$$max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n) \tag{23}$$

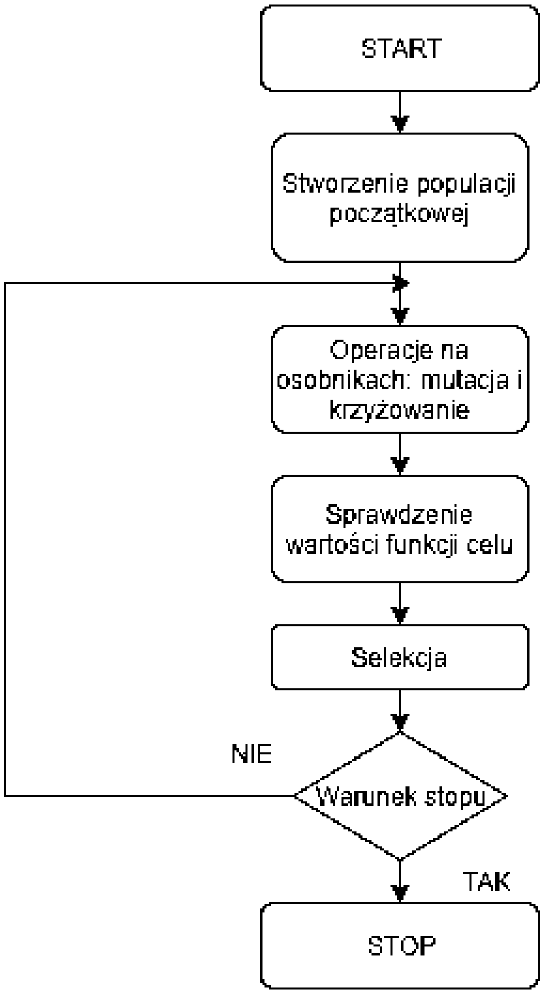
Jeśli algorytm osiąga wskaźnik aproksymacji $\rho(n)$, nazywamy go algorytmem $\rho(n)$- aproksymacyjnym. Definicja wskaźnika aproksymacji algorytmu $\rho(n)$- aproksymacyjnego stosuje się zarówno dla problemów, w których jest minimalizcja jak i maxymalizacja funkcji celu.
Dla maxymalizacji jest: $0 \leq C \leq C^*$ oraz wskaźnik C^*/C daje czynnik mówiący o ile koszt rozwiązania optymalnego jest większy od kosztu rozwiązania przybliżonego.
Podobnie dla minimalizacji jest: $0 \leq C^* \leq C$ oraz wskaźnik C/C^* daje czynnik mówiący o ile koszt rozwiązania optymalnego jest większy od kosztu rozwiązania przybliżonego.
Ponieważ zakładamy, że wszystkie rozwiązania mają dodatni koszt, współczynniki te są dobrze zdefiniowane. Wskaźnik aproksymacji algorytmu przybliżonego jest nie mniejszy niż 1, skoro $C/C^* \leq 1$ implikuje $C/C^* \geq 1$.
Stąd algorytm przybliżony o wskaźniku aproksymacji wynoszącym 1 daje optymalne rozwiązanie, natomiast algorytm przybliżony o dużym wskaźniku aproksymacji może zwrócić rozwiązanie, które jest o wiele gorsze od optymalnego.

1.3.4. Algorytm genetyczny

Na podstawie [3]:
Termin algorytm genetyczny po raz pierwszy wprowadził John Holland, bazując na koncepcji Darwina- teorii ewolucji. Jest to metaheurystyka (metoda poszukiwania rozwiązań, która nie gwarantuje znalezienia optymalnego rozwiązania) bazująca na zjawisku naturalnej selekcji (teorii ewolucji gatunków), a także dziedziczności. Należy do szerszej grupy- algorytmów ewolucyjnych. Algorytm ten polega na znajdowaniu rozwiązania naśladując zjawiska występujące w środowisku naturalnym, takie jak: mutacje, krzyżowania gatunków, a także selekcja.

Algorytm genetyczny w krokach:

1. Utworzenie początkowej populacji
2. Przeprowadzanie mutacji i krzyżowania
3. Sprawdzenie funkcji celu dla osobników
4. Wybór najlepszych osobników poprzez selekcję
5. Powtarzanie kroków 2-4 aż do spełnienia warunku stopu
6. Koniec algorytmu i wybór najlepszego osobnika



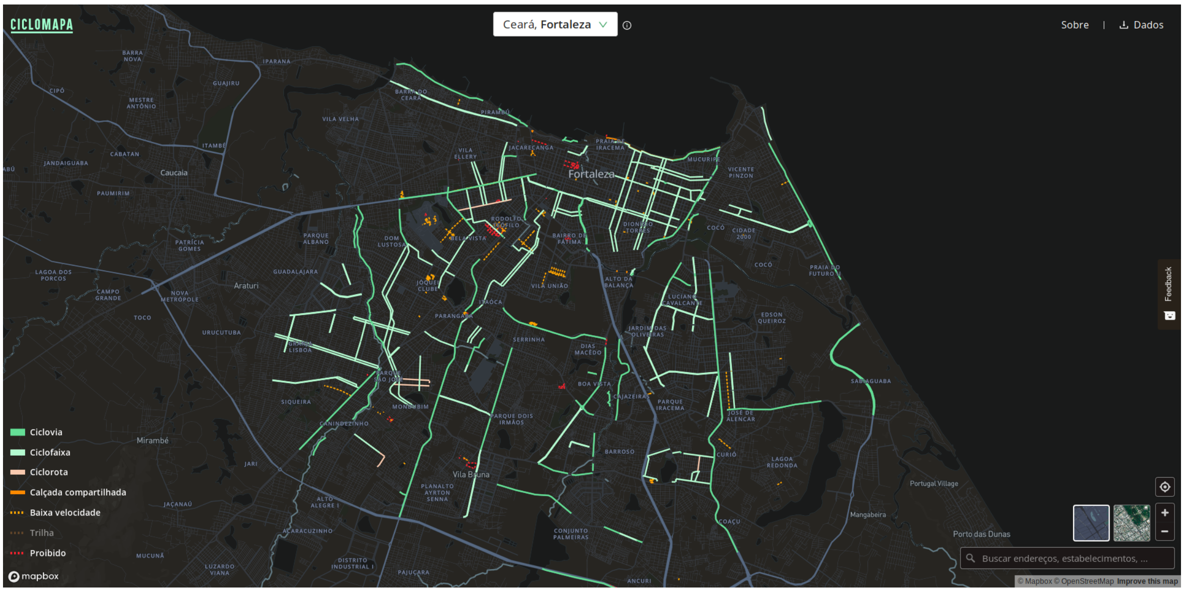
Rysunek1. Schemat algorytmu genetycznego

2. Opis projektu

2.1. Użyte technologie

Na podstawie [4]:
Python- jest językiem programowania ogólnego przeznaczenia. Jest to język wysokiego poziomu. Może być skutecznie wykorzystywany przy budowie w zasadzie każdego programu. Nie potrzebuje bezpośredniego dostępu do sprzętu komputerowego. Python nie jest optymalny dla programów, które mają duże ograniczenia niezawodności lub są zbudowane i utrzymywane przez wiele osób lub przez długi czas. Ma on jednak kilka zalet w porównaniu z innymi językami programowania, m.in.:
- jest łatwy w nauce
- posiada bardzo dużą liczbę darmowych i powszechnie dostępnych bibliotek, zapewniających rozszerzoną funkcjonalność

Na podstawie [5]:
OSM (OpenStreetMap)- jest to darmowa, edytowalna mapa całego świata, tworzona przez wolontariuszy, wydawana na licencji open-content. Licencja OpenStreetMap pozwala na bezpłatny dostęp do obrazów i podstawowych danych map. Projekt ma na celu promowanie nowych zastosowań tych danych. W pracy OSM zostało wykorzystane do zwizualizowania przebiegów tras i pokazania ich na rzeczywistej mapie. Przykład obrazu uzyskanego dzięki OSM [6]:



Rysunek2. Przykładowy screen z osm

Biblioteki:

Źródła

[1] https://sjp.pwn.pl/slowniki/optymalizacja.html

[2] http://www.cs.put.poznan.pl/arybarczyk/TeoriaZlozonosc.pdf

[3] http://home.agh.edu.pl/ horzyk/lectures/pi/ahdydpiwykl8.html

[4] Sysło M.M., Deo N., Kowalik J.S., Algorytmy optymalizacji dyskretnej, wyd. drugie, Warszawa: Wydawnictwo Naukowe PWN, 1995, ISBN 83-01-11818-0

[5] Dantzig, George B. (1963), Linear Programming and Extensions, Princeton, NJ: PrincetonUP, pp. 545–7, ISBN 0-691-08000-3, sixth printing, 1974.

[6] Dantzig, George Bernard; Ramser, John Hubert (October 1959). "The Truck Dispatching Problem", https://andresjaquep.files.wordpress.com/2008/10/2627477-clasico-dantzig.pdf

[7] T. Ralphs, J. Hartman and M. Galati. "Capacitated Vehicle Routing and Some Related Problems". Some CVRP Slides. Rutgers University. 2001

[8] https://www.researchgate.net/publication/326129926_Capacitated_vehicle_routing_problem_model_for_carriers

[9] Bianchi, Leonora; Marco Dorigo; Luca Maria Gambardella; Walter J. Gutjahr (2009). "A survey on metaheuristics for stochastic combinatorial optimization"

[10] Thomas H. Cormen; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein (2009). Introduction To Algorithms (3rd ed.). MIT Press. ISBN 978-0-262-03384-8.

[11] https://javarevisited.blogspot.com/2014/08/bubble-sort-algorithm-in-java-with.html

[12] D. E. Goldberg: Algorytmy genetyczne i ich zastosowania. Warszawa: WNT, 1998. (pol.)

[13] Guttag, John V. (12.08.2016). Introduction to Computation and Programming Using Python: With Application to Understanding Data. MIT Press. ISBN 978-0-262-52962-4.

[14] https://wiki.openstreetmap.org/wiki/About_OpenStreetMap

[15] https://wiki.openstreetmap.org/wiki/File:Screenshot_2019-10-23_CicloMapa.png