Marketing Przemysłowy:
Problem Marszrutyzacji

Realizacja zagadnienia VRP z użyciem algorytmu pszczelego, mrówkowego i genetycznego

Łukasz Bandała
Maciej Fabia
Rafał Gibas
Marcin Godlewski
Tomasz Gryzio
Piotr Jarmołowicz
Michał Kołodziej

Problem VRP

Problem VRP rozwiązywany jest w różnych przedsiębiorstwach mających do czynienia z dostawą towarów w określone miejsca. Przykładami są:

- firmy kurierskie
- poczta
- firmy wysyłkowe z własną flotą dostawczą (np. pizzerie, sklepy z meblami)
- firmy dostawcze

Definicja

Mamy firmę kurierską, która musi zrealizować zlecenia w postaci dostarczenia przesyłek do klientów rozmieszczonych w różnych odległościach od magazynu. Przesyłki mają określoną wielkość, a ciężarówki określoną pojemność. Do klientów można dotrzeć różnymi drogami. Drogi są połączeniami między klientami i znamy koszt ich przebycia (odległość, czas pokonania, lub koszt spalenia benzyny na odcinku).

Dodatkowe ograniczenia

Dodatkowymi ograniczeniami mogą być:

- Maksymalna trasa dla ciężarówki - ciężarówka bez paliwa daleko nie ujedzie: trzeba zdefiniować stacje benzynowe w systemie
- Godziny pracy - kurier nie pracuje dłużej niż 8h
- Maksymalna pojemność ciężarówki
- Okna czasowe - klient życzy sobie dostać towar o określonej porze

Spośród wymienionych wyżej ograniczeń przyjęliśmy maksymalną trasę i maksymalną długość jazdy. W zależności od rozpatrywania macierzy kosztów, są to wartości równoważne. Pojemność ciężarówki - ilość odwiedzonych klientów (przesyłki są takie same) jest rozpatrywana w jednej z możliwych funkcji celu.

W projekcie nie rozważaliśmy problemu okien czasowych.

Uproszczenia

Problem rozładunku dla uproszczenia zostanie pominięty. Normalnie, oprócz samej dostawy, kurier musi rozładować towar, podpisać pokwitowania, uregulować płatności. Trwa to niezerowy czas, czasem wymaga nakładu energii (rozładowanie ciężkich materiałów dźwigiem), czasem jest niedeterministyczne (klient wolno schodzi po schodach). Z uwagi na to, że rozładunek można np wliczyć w koszt, rozwiązaniem może być aktualizacja wag grafu połączeń z uwzględnieniem dodatkowych kosztów dotarcia do klienta. W takiej sytuacji macierz połączeń staje się niesymetryczna, a realizacja algorytmu nie ulega zmianom. Podsumowując, graf zostanie zgodnie z założeniami zaktualizowany zerowymi kosztami dostawy. Innymi słowy - nie ulegnie zmianie.

Dodatkowo zakładamy, że ciężarówki są jednakowe - ich pojemność jest taka sama i palą tyle samo benzyny. Ma to uzasadnienie w generowaniu rozwiązania za pomocą różnych algorytmów.

Cel

Naszym celem jest znalezienie optymalnej trasy dla każdej z ciężarówek, minimalizującej koszt dostarczenia wszystkich przesyłek. W celu znalezienia rozwiązania posłużymy się algorytmami:

- pszczelim
- mrówkowym
- genetycznym

Model

Magazyn i klienci

Mamy dany magazyn `M`, oraz punkty dostawy S_1, S_2, … S_n. Odpowiadają im węzły kolejno 0, 1, …, n.

Punkt	M	S_1	S_2	…	S_n
Węzeł grafu	0	1	2	…	n

Wagi połączeń zdefiniowane są za pomocą macierzy:

	0	**1**	**2**	**3**
0	-	4	5	2
1	4	-	1	-
2	5	1	-	1
3	2	-	1	-

Wagi ”-” oznaczają nieskończoność. W modelu nie muszą występować wartości nieskończone, można je zastąpić wystarczająco dużą liczbą. Liczba ta powinna być większa niż maksymalna możliwa długość cyklu. W zależności od algorytmu - można włączyć taką krawędź do cyklu, uzyskując przy tym gorszą wartość funkcji celu; można również uznać taki cykl za nieprawidłowy i zastosować korekcję lub wygenerować inny.

Ciężarówki

Mamy ciężarówki C_1, …, C_m. Każda ma:

- pojemność V_i
- aktualny załadunek L_i
- współczynnik kosztu przejazdu (odpowiadający zużyciu paliwa) c_i[1)]
- długość przejechanej trasy s_i

Do podstawowego wariantu problemu (ogranicza nas tylko pojemność ciężarówki) wystarczą nam tylko trzy pierwsze wartości. W przypadku posiadania ciężarówek niezróżnicowanych wyraźnie pod względem kosztu ich eksploatacji, współczynniki kosztów można przyjąć jako równe 1. W naszym problemie przyjmujemy takie same współczynniki dla wszystkich ciężarówek.

Graf połączeń

Graf połączeń uwzględniający większą ilość ciężarówek może być modyfikacją wcześniej przedstawionego grafu. Zobaczmy to na przykładzie:

X	**M_1**	**M_2**	**S_1**	**S_2**	**S_3**
M_1	-	0	4	5	2
M_2	0	-	4	5	2
S_1	4	4	-	1	-
S_2	5	5	1	-	1
S_3	2	2	-	1	-

Powyższa macierz opisuje model z dwoma ciężarówkami. Węzły M_1 i M_2 odpowiadają magazynowi. Ciężarówka C_1 wystartuje z magazynu M_1, a po dojechaniu do magazynu M_2, magazyn opuści kolejna ciężarówka[2)].

Podsumowując, tworzymy w modelu m magazynów, z czego każdy magazyn zaopatruje jedną ciężarówkę. Ciężarówka wyjeżdża ze swojego magazynu, podróż kończąc w magazynie kolejnej ciężarówki. Wszystkie magazyny są zlokalizowane w tym samym miejscu, co sprawia, że mamy do czynienia cały czas z jednym magazynem.

Dodatkowo dla naszych algorytmów generujemy macierz z wylosowanych punktów na płaszczyźnie, więc nie używamy krawędzi nieskończonych. Nie ma to znaczącego wpływu na rozwiązanie, za to pozwala na lepszą wizualizację wyników.

Funkcja oceny

Funkcja oceny stanowi kluczową część algorytmu. W zależności od definicji problemu i jego ograniczeń, funkcja oceny przyjąć może różne formy. Zdefiniujemy tutaj funkcję celu pasującą do określonych wcześniej ograniczeń problemu.

Każdy wektor rozwiązania ma postać:

M1 S1 S2 … Sn M2 Sn+1 …

Gdzie:

- M1,M2… oznaczają zmianę ciężarówki w magazynie,
- S1,S2… oznaczają kolejny odwiedzony punkt.

Mając dany wektor kodujący pełny cykl, potrafimy obliczyć koszt przejazdu ciężarówek przez wszystkie węzły grafu. Algorytm liczący funkcję oceny F polega na wyznaczeniu sumy kosztów przejazdu i uwzględnienie kar, gdzie kary naliczane są za przeładowanie ciężarówek, przekroczenie czasu jazdy, itp..

Obecnie istnieją 3 możliwe funkcje celu:

1. Suma wartości wszystkich przejść między poszczególnymi punktami.
2. Suma wartości wszystkich przejść między poszczególnymi punktami oraz dodatkowo za każdym razem dodawana jest rosnąca z każdym ruchem (liniowo) wartość (kara za zwiększanie czasu przejazdu).
3. Suma wartości wszystkich przejść między poszczególnymi punktami oraz dodatkowo po przekroczeniu dozwolonej wartości ruchów, dodawana jest nieskończoność (uwzględnienie maksymalnej ładowności.

Dodatkowo, w wariancie przekroczenia maksymalnej trasy, do funkcji celu dodawana jest wartość drogi powrotnej do magazynu. Funkcja celu jest minimalizowana. Wszystkie funkcje celu zakładają nierozróżnialność ciężarówek.

Algorytm pszczeli

Jest to algorytm heurystyczny z 2004 roku, bazujący na obserwacji zachowania pszczół w ich naturalnym środowisku. Informacje zaczerpnięte zostały ze strony The Bees Algorithm website.

Opis zachowania pszczół w rzeczywistości

Rój w celu zapewnienia pożywienia, w pierwszej kolejności wysyła pszczoły - zwiadowców. Zwiadowcy po znalezieniu źródła pokarmu wracają do ula. Następnie za pomocą tańca przekazywane są informacje w jakim kierunku, w jakiej odległości i w jakiej ilości znaleziono źródło. Następnie zwiadowcy odlatują, a wraz z nimi zwerbowane robotnice, których ilość zależy od wykonanego tańca. Robotnice na bieżąco monitorują stan pożywienia. Jeżeli wciąż jest to obiecujące miejsce, to po powrocie do ula odbywają kolejny taniec rekomendujący to miejsce.

Opis algorytmu

Pseudo kod dla wykorzystanego algorytmu:

```
1. Inicjalizacja populacji za pomocą losowych rozwiązań.
2. Wyliczenie funkcji celu dla populacji.
3. For (ilość kroków) //Tworzenie nowej populacji.
    4. Wybór najlepszych miejsc do przeszukiwania sąsiedztwa.
    5. Werbunek pszczół dla wybranych miejsc (proporcjonalnie do najlepszych miejsc + stała wartość 1)
    6. Badanie przez pozostałe pszczoły sąsiedztwa ( ponowne wyliczanie funkcji celu )
    7. Nowi zwiadowcow i usuniecie powtorzonych tras
    8. Wybór najlepsze trasy.
9. End FOR.
```

Algorytm mrówkowy

Algorytm mrówkowy został opracowany przez Marco Dorigo. Jest stosowany głównie do wyszukiwania optymalnych ścieżek w grafach. Inspiracją do jego stworzenia było zachowanie kolonii mrówek podczas poszukiwania źródła pożywienia. Wyruszając z mrowiska w poszukiwaniu pożywienia, początkowo wybierają trasę losowo. Jeśli odnajdą pokarm, to powracając do mrowiska zostawiają feromon. Kolejne mrówki wybierają daną trasę tym chętniej, im więcej pozostawiono na niej feromonu. Ostatecznie dochodzi do wyodrębnienia jednej lub co najwyżej kilku ścieżek.

Kroki algorytmu mrówkowego

```
1. Przygotuj zerową macierz feromonu populacji i feromonu ogólnego
2. DOPÓKI ostatnia poprawa nastąpiła nie wcześniej niż przed PP iteracjami
    1. DOPÓKI numer aktualnej mrówki jest mniejszy niż liczebność populacji
        1. Przygotuj nową mrówkę
        2. DOPÓKI jest możliwe wykonanie kroku przez mrówkę
            1. Oblicz wagi dla krawędzi wychodzących z aktualnie zajmowanego węzła
            2. Wylosuj węzeł docelowy uwzględniając wagi
            3. Przejdź do węzła docelowego
        3. Inkrementuj numer aktualnej mrówki
        4. JEŚLI mrówka wykonała pełną trasę
            1. Oblicz wartość funkcji celu
            2. Oblicz ilość zostawianego feromonu
            3. Zostaw feromon w macierzy feromonu populacji
            4. JEŚLI wartość celu lepsza od najlepszej dotychczasowej
                1. Zapamiętaj wartość funkcji celu jako najlepszą
                2. Zapamiętaj aktualny numer iteracji jako ostatnią iterację poprawy
    2. Dodaj feromon populacji do feromonu ogólnego i wyzeruj feromon populacji
    3. Inkrementuj licznik iteracji
3. Zwróć najlepsze rozwiązanie
```

Zastosowanie w przypadku problemu VRP

W typowym algorytmie mrówkowym każda mrówka przechowuje listę węzłów grafu, które już odwiedziła. W przedstawionej implementacji dodatkowo przechowuje informacje:

- Liczbę odwiedzonych już węzłów właściwego grafu, tzn. w wyłączeniem ciężarówek,
- Liczbę zmian ciężarówek, wliczając w to wybór pierwszej ciężarówki,
- Liczbę odwiedzonych węzłów właściwego grafu od zmiany ostatniej ciężarówki,
- Niewykorzystaną ładowność aktualnej ciężarówki.

Przyjęto, że ciężarówki mają taki sam koszt przebycia jednostkowej długości (takie samo zużycie benzyny) i taką samą pojemność

Przyrost feromonu dla krawędzi (i,j) należącej do ścieżki s jest obliczany z wykorzystaniem wspólnej dla wszystkich trzech algorytmów funkcji oceny: $\mathbf{delta_{fer}[i,j] = 1/f_{oceny}(s)}$

Realizacja

Problem dominacji jednej ścieżki

W celu zredukowania problemu dominacji jednej ścieżki nad innymi, używamy zmodyfikowanych wzorów do obliczania wag prawdopodobieństwa służących do wylosowania następnego węzła grafu (moment podjęcia decyzji przez mrówkę). Modyfikacja polega na dodaniu składnika „1", który powoduje, że nawet krawędzie o zerowej ilości feromonu otrzymują niezerowe wagi. Do ustalenia korzystnych zależności pomiędzy krawędziami posiadającymi i nie posiadającymi feromonu służą współczynniki alfa1 i alfa2.

Najczęściej spotykanym rozwiązaniem tego problemu jest okresowe nanoszenie dodatkowego feromonu na wszystkie krawędzie oraz obliczanie prawdopodobieństw tylko na podstawie ilości feromonu.

Wagi krawędzi dla prostej funkcji celu

Poniższe wzory służą do obliczania wag prawdopodobieństwa w przypadku prostej funkcji celu - sumy wszystkich krawędzi ścieżki.

Krawędzie o niezerowej długości

a[i,j] = (1 + alfa1*f[i,j]^{alfa2}) / (d[i,j]^{beta})
alfa1, alfa2 - parametry
f[i,j] - feromon ogólny znajdujący się na krawędzi (i,j)
d[i,j] - długość krawędzi (i,j)

Krawędzie o zerowej długości (zmiany ciężarówek)

a[i,j] = 1 + gamma2*f[i,j]^{gamma3}
f[i,j] - feromon ogólny znajdujący się na krawędzi (i,j)
gamma2, gamma3 - parametry

Narastająca funkcja kosztu

Możliwe jest skorzystanie z bardziej skomplikowanego wariantu funkcji oceny - na węzły nakładane są rosnące wykładniczo kary. Im później po zmianie ciężarówki został odwiedzony dany węzeł, tym kara jest większa. Gdy wybierzemy tą opcję, wzory na wagi ulegają pewnym zmianom, mającym na celu głównie uwzględnienie, ile węzłów odwiedzono po ostatniej zmianie ciężarówki.

Krawędzie o niezerowej długości

a[i,j] = (1 + alfa1*f[i,j]^{alfa2}) / (d[i,j]$^{beta\text{^}(z+1)}$)

Krawędzie o zerowej długości (zmiany ciężarówek)

a[i,j] = gamma1^{z}(1 + gamma2*f[i,j]^{gamma3})

z - liczba odwiedzonych węzłów po ostatniej zmianie ciężarówki

Parowanie i zwiększanie feromonu ogólnego

W pamięci przechowywane są jednocześnie dwa zestawy feromonów - ogólny i populacyjny. Podczas poruszania się (podejmowania decyzji) bierzemy pod uwagę tylko ogólny. Feromon odkładany na ścieżkach przez kolejne mrówki jest zapamiętywany jako populacyjny. Kiedy przez graf przejdzie cała populacja, zostaje on dodany do feromonu ogólnego i wyzerowany.

$$f_{nowy} = (1 - ro) * f + f_{pop}$$

f_{nowy} - nowe wartości feromonu ogólnego
f - feromon ogólny
f_{pop} - feromon populacji
ro - parametr

Algorytm genetyczny

Jednym z opracowanych przez nas algorytmów jest algorytm genetyczny. Zaimplementowana została prosta wersja algorytmu, rozwiązująca problem VRP przy zadanych ograniczeniach.

Działanie algorytmu

Algorytm genetyczny został zaimplementowany według następującego schematu:

```
1. Inicjalizacja populacji
2. Dopóki nie wykonano zadanej liczby iteracji wykonuj:
  a. Operacje krzyżowania
  b. Operacje mutacji:
     * zmiana kolejności genów
     * przesunięcie pojedynczego genu
     * odwrócenie fragmentu chromosomu
  c. Selekcja najlepszych osobników
  d. Redukcja wielkości populacji
```

Konstrukcja chromosomu

Gen

Genem będzie odwiedzone w danym momencie miejsce (identyfikator wierzchołka grafu, np. 1, 4 albo 8).

Chromosom

Chromosom będzie ciągiem genów. Będzie oznaczał przebyty w problemie cykl.

Przykładowy chromosom:

1 3 4 2 1

Oznacza to, że wychodząc z punktu 1 odwiedzono kolejno punkty 3, 4, 2 i nastąpił powrót do punktu 1.

Ograniczenia:

- zawsze startujemy w punkcie 1 i kończymy w tym punkcie[3)] → pierwszy i ostatni gen cyklu równy 1
- wartości pozostałych genów wynoszą od 2 do n
- wartości genów nie powtarzają się

Operacje genetyczne

Krzyżowanie

Ta operacja stwarza problem w postaci dublowania się węzłów.

Np. krzyżując:

1 3 2 4 5
3 4 5 **1 2**

Otrzymamy chromosom, w którym powtarzają się geny 1 i 2:

1 3 2 1 2

Zauważmy, że każdy gen może wystąpić 0, 1 lub 2 razy.

Rozwiązaniem może być:

- wylosowanie nowych wartości dla genów powtórzonych
- wybranie najlepszego cyklu eliminującego powtórzenia

Bardziej sensowna wydaje się pierwsza opcja, ponieważ zapewnia losowość algorytmu. Zapobiega to przedwczesnej zbieżności do minimum lokalnego funkcji celu.

Metoda losowa eliminacji powtórzeń

W celu głębszej analizy problemu pozwolę sobie zadać kilka pytań, po kolei na nie odpowiadając.

1. Czy wszystkie powtórzone geny rozlosować od nowa?

- + szybka implementacja (zaznaczamy i losujemy)
- - zmniejszone podobieństwo do rodziców

2. Czy może gdy gen się powtarza, wybrać gen drugiego rodzica?

1. Czy może to spowodować powtórzenie się następnych genów?
   - Tak, np.[4]:

3 3 **5** 1 2

1. Czy może to spowodować niewybranie niektórych genów?
   - Tak, wiąże się z tym poprzednia odpowiedź.

3. Który gen (pierwszy czy drugi) zastąpić wylosowanym?

- można przyjąć z góry
- można wybrać losowo (+ lepsze zróżnicowanie)

W celu eliminacji powtórzeń zastosowano ponowne rozlosowanie genów w miejscach powtórzeń, z dostępnej, nieużywanej puli genów.

Mutacja

Mutacje mają na celu wprowadzanie zaburzeń genów, które wytrącają osobniki z minimów lokalnych i zapewniają większą różnorodność populacji, a co za tym idzie, umożliwiają prawidłową adaptację populacji poprzez powstawanie osobników o nowych cechach.

Mutacja jest zwykle punktową zmianą chromosomu, obejmującą pojedyncze geny. W naszym problemie mutacja będzie polegać na:

- przesunięciu położenia pojedynczego genu
- odwróceniu fragmentu chromosomu
- zmianie kolejności K losowo wybranych genów.

Gdzie liczba K jest losowa i przyjmuje wartości nie mniejsze niż 2 oraz nie większe niż długość chromosomu.

Kiedy K=2, mutacja polega po prostu na zamianie miejscami dwóch genów. W przypadku większej liczby K powinniśmy uporać się z prawdopodobnym wylosowaniem identycznej permutacji genów, czyli sytuacji, gdzie mutacja nic nie zmienia. To zjawisko jest na szczęście bardzo mało prawdopodobne (1/K!), ale tylko dla dużych K. Sprawdzanie poprawności (a następnie ponowne losowanie) powinno się przeprowadzać dla odpowiednio małego K[5)]. Na przykład sprawdzenie poprawności dla k< =7, daje mniejszą niż 1 na 1000 szansę, że chromosom się powtórzy.

Metody selekcji

Selekcja ma na celu wybranie spośród populacji osobników, które:

1. Przetrwają
2. Wydadzą potomstwo
3. Zmutują.

Pierwszy punkt obrazuje, jak środowisko wpływa na przetrwanie populacji. W wyniku selekcji zostaną wyeliminowane osobniki o najgorszych wartościach funkcji przystosowania. Na ilość przetrwanych osobników wpływ będzie mieć pojemność środowiska Vs. W naszym algorytmie eliminować będziemy Ws najgorszych osobników, gdzie Vs + Ws = Ps, Ps oznacza wielkość populacji.

Selekcja osobników do rozrodu opisuje mechanizmy wyboru rodziców, których geny zostaną skrzyżowane. Rodzice będą wybierani losowo, a wpływ na prawdopodobieństwo wylosowania będzie miała wielkość funkcji przystosowania. Potomstwo wydadzą

Mutacja w przyrodzie jest zjawiskiem, które dotknąć może każdego osobnika niezależnie od jego przystosowania, jakkolwiek szkodliwe warunki wpływają na wzrost ich prawdopodobieństwa. Przykładowo, w rejonach silnie zanieczyszczonych można zaobserwować większą liczbę anomalii w budowie osobników. W algorytmie zastosujemy jeden z trzech scenariuszy. Prawdopodobieństwo mutacji będzie równe dla każdego osobnika. Jako mutacje będziemy rozpatrywać wszystkie operacje zmieniające pojedynczy chromosom. Należeć będą do nich:

- permutacja fragmentu
- przesunięcie genu
- odwrócenie części cyklu.

Aplikacja

Projekt został zrealizowany w środowisku Matlab. Każdy z algorytmów może być wywołany ze wspólnej aplikacji interfejsu użytkownika, pozwalającej na modyfikację jego parametrów. Wyniki generowane są w postaci wykresu funkcji celu, wykresu czasów działania oraz diagramów reprezentujących najlepsze marszruty dla każdego algorytmu.

Wnioski

Zaimplementowane przez nas algorytmy pozwoliły na znalezienie dobrych rozwiązań problemu VRP przy względnie niedługim czasie działania. W zależności od ustawień parametrów, osiągane są różne rezultaty każdego z algorytmów.

Zaobserwowaliśmy, że ze znalezieniem najlepszej marszruty najlepiej radził sobie algorytm pszczeli, choć wymagał przy tym znacznego czasu na wykonanie. Nieco gorzej wypadał algorytm genetyczny, choć znacznie mniej czasu procesora zajmowało jego działanie. Algorytm mrówkowy z kolei działał znacznie szybciej od pozostałych, jednak nie osiągał znakomitych wyników.

Z analizy przypadków wynika, że najlepszym kompromisem wydaje się być algorytm genetyczny - ze względu na szybkość, dobrą zbieżność i łatwość implementacji. Z drugiej strony algorytm pszczeli pozwala znaleźć bardzo dobre rozwiązania przy akceptowalnym czasie.

[1)] W problemie przyjmujemy wszystkie równe 1.
[2)] Oczywiście to tylko w celu zamodelowania problemu. W rzeczywistości ciężarówki wyruszą równocześnie
[3)] Oznacza to, że podróż zaczynamy w magazynie i powracamy do magazynu po zakończeniu dostawy.
[4)] Pogrubiono podmienione.
[5)] Prawdopodobnie nie sprawi problemu sprawdzenie poprawności dla każdego K.