Algorytmy i metody optymalizacji (AMO)
Projekt I, zestaw AK, nr 4

Arkadiusz Piórkowski

Zadanie lokalizacji

Pewna sieć restauracji chce upowszechnić własną aplikację mobilną wskazującą swój najbliższy punkt, w oparciu o dane lokalne udostępniane przez operatora sieci komórkowej. Współrzędne punktów będą wyliczane na podstawie przybliżonych odległości od trzech najbliższych stacji bazowych. Współrzędne geograficzne stacji bazowych są znane.
Wypowiedzieć się w pobliżu jakiej miejscowości w Polsce znalazł się użytkownik aplikacji mobilnej.

Stacja	Dł. geogr.	Szer. geogr.	Odległość [km]
A	22°37'23” E	52°45`54” N	4
B	22°30'47” E	52°40`27” N	13.4
C	22°44'00” E	52°43`37” N	4.1

Należy:

1. Sformułować układ równań określający nasze położenie.
2. Sformułować zadanie optymalizacji bez ograniczeń stosując metodę najmniejszych kwadratów.
3. Wyznaczyć położenie rozwiązując sformułowane powyżej zadanie optymalizacji za pomocą:
   (a) pakietu AMPL na komputerze własnym oraz na serwerze NEOS,
   (b) wybranej gradientowej metody optymalizacji bez ograniczeń z Optimization Toolboxa programu MATLAB; wywołać korzystając z gradientu i nie; do liczenia gradientu użyć funkcji `gradient` z Symbolic Toolboxa (off-line, przed wywołaniem solwera optymalizacyjnego).
4. Sprawdzić wpływ doboru:
   - punktu startowego,
   - dokładności w teście STOP'u metody,
   - zaburzeń w danych

   na uzyskane wyniki.

Spis treści

Zadanie lokalizacji 1
1. Układ równań określający położenie użytkownika.. 3
2. Zadanie optymalizacji bez ograniczeń przy zastosowaniu metody najmniejszych kwadratów... 3
3. Rozwiązanie zadania optymalizacji.. 4
4. Uzyskane wyniki.. 5
Wpływ punktu startowego na uzyskiwane wyniki.. 5
Wpływ zaburzeń danych na uzyskiwane wyniki.. 6
Wpływ dokładności w teście STOP-u metody na uzyskiwane wyniki......................... 8

1. Układ równań określający położenie użytkownika

Do obliczenia odległości pomiędzy dwoma punktami wykorzystano formułę Haversine, bazującej na ortodomie(najkrótsza droga pomiędzy dwoma punktami na powierzchni kuli biegnąca po jej powierzchni). Argumentami wejściowymi dla tej metody są współrzędne geograficzne w radianach, należy zatem przejść ze współrzędnych w stopniach na radiany. Układ równań określający położenie użytkownika ma następującą postać:

$$\begin{cases} d_1 = 2 \cdot R \cdot asin\left(\sqrt{\sin^2\left(\frac{lat_1 - lat_x}{2}\right) + \cos(lat_x) \cdot \cos(lat_1) \cdot \sin^2\left(\frac{lon_1 - lon_x}{2}\right)}\right) \\ d_2 = 2 \cdot R \cdot asin\left(\sqrt{\sin^2\left(\frac{lat_2 - lat_x}{2}\right) + \cos(lat_x) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{lon_2 - lon_x}{2}\right)}\right) \\ d_3 = 2 \cdot R \cdot asin\left(\sqrt{\sin^2\left(\frac{lat_3 - lat_x}{2}\right) + \cos(lat_x) \cdot \cos(lat_3) \cdot \sin^2\left(\frac{lon_3 - lon_x}{2}\right)}\right) \end{cases}$$

, gdzie:

- $lon_{1...3}$ – długość geograficzna stacji (odpowiednio A,B,C) w radianach;
- $lat_{1...3}$ – szerokość geograficzna stacji (odpowiednio A,B,C) w radianach;
- $d_{1...3}$ – odległość od stacji (odpowiednio A,B,C) w kilometrach;
- R – promień Ziemi 6371 km;
- lon_x , lat_x – odpowiednio długość i szerokość geograficzna użytkownika w radianach.

2. Zadanie optymalizacji bez ograniczeń przy zastosowaniu metody najmniejszych kwadratów

Sprowadzając układ równań określający położenie użytkownika do zadania optymalizacji bez ograniczeń przy zastosowaniu metody najmniejszych kwadratów otrzymujemy:

$$\min_{lon_x,\, lat_x} \sum_{i=1}^{3} \left(2 \cdot R \cdot asin\left(\sqrt{\sin^2\left(\frac{lat_i - lat_x}{2}\right) + \cos(lat_x) \cdot \cos(lat_i) \cdot \sin^2\left(\frac{lon_i - lon_x}{2}\right)}\right) - d_i\right)^2$$

3. Rozwiązanie zadania optymalizacji

Wymienione w drugim rozdziale zadanie optymalizacji rozwiązano za pomocą programów AMPL i MATLAB.

W programie MATLAB wykorzystano polecenie `lsqnonlin`, które rozwiązuje nieliniowe zadania najmniejszych kwadratów. Dla tego polecenia istnieją dwa algorytmy: "trust region reflective" and "Levenberg-Marquardt". Do rozwiązania zadania danego zadania wybrano metodę Levenberga-Marquardta, która łączy zalety metody najszybszego spadku i metody Gaussa-Newtona. Polecenie `lsqnonlin` umożliwia wybranie punktu startowego, określenie dokładności rozwiązania i użycie zdefiniowanego przez użytkownika gradientu. Użytkownik może zaimplementować gradient analitycznie lub wykorzystać do tego funkcje `gradient`. W tym przypadku według polecenia wykorzystano funkcję `gradient`. Umożliwienie wprowadzenie gradientu uzyskuje się poprzez ustawienie opcji `'SpecifyObjectiveGradient',true`. Dokładność rozwiązania określają głównie trzy parametry:

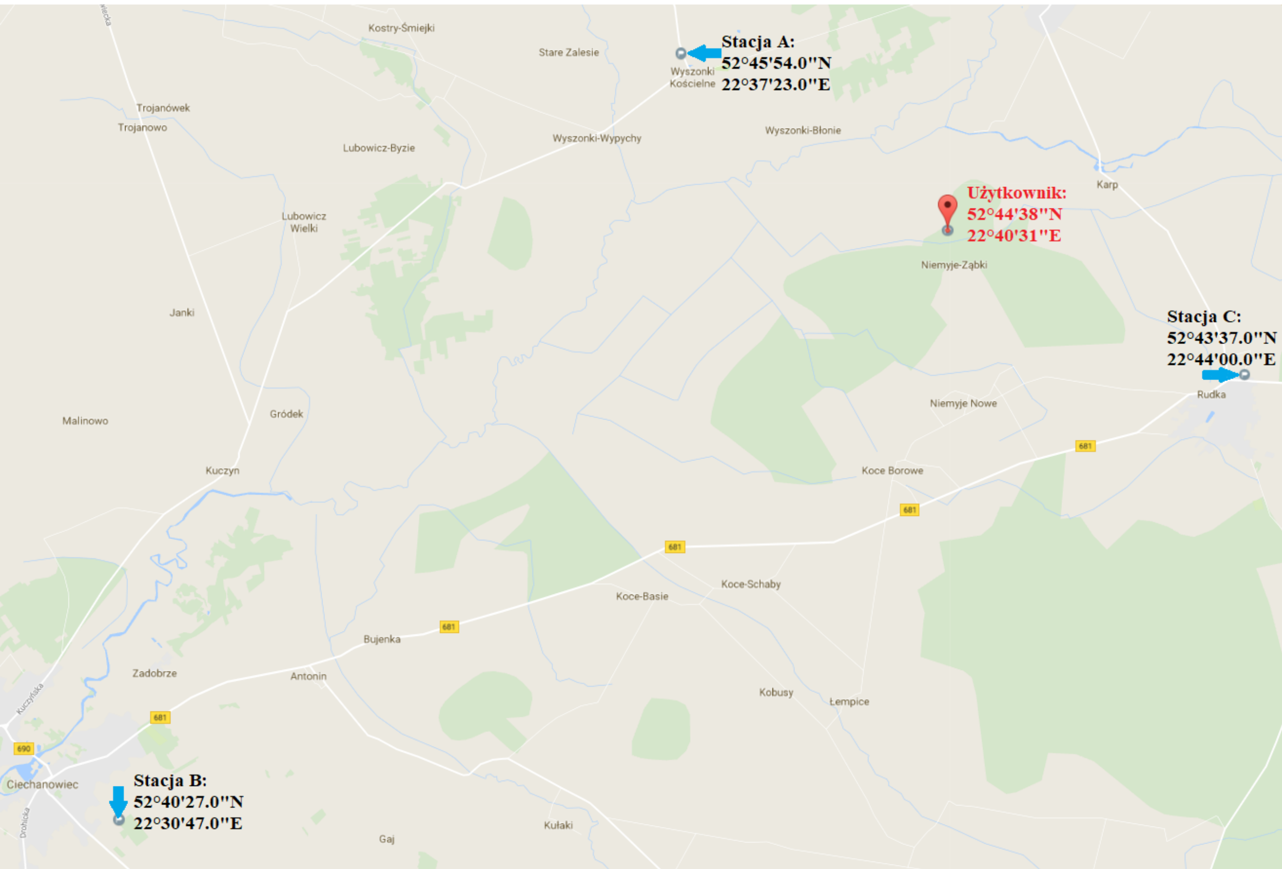
- `'FunctionTolerance',1e-6(default),`
- `'OptimalityTolerance',1e-6(default),`
- `'StepTolerance',1e-6(default),`

W rzeczywistości jednak zaimplementowany algorytm Levenberga-Marquardta nie korzysta z wartości parametru `'OptimalityTolerance'`, przyjmując jego wartość równą iloczynowi wartości parametru `'FunctionTolerance'` i liczby 1e-4. Określając dokładność w teście STOP-u metody posłużono się parametrem `'StepTolerance'`.

Pakiet AMPL umożliwia rozwiązywanie zadań optymalizacji zarówno na własnym komputerze jak i na serwerze NEOS. Wykorzystując serwer NEOS uruchamiamy nasz problem optymalizacji na własnym komputerze jednak zamiast wybierać solver zainstalowany na własnym komputerze wywołuje się Kestrel, będący programem typu klient, który wysyła zadanie optymalizacji do solvera uruchomionego na zdalnym komputrzerze serwera NEOS. Na serwerze NEOS występuje znacznie większa liczba solverów, niż dostępna w podstawowym pakiecie AMPL. Dane zadanie rozwiązano korzystając z solvera MINOS(domyślny) na własnym komputerze oraz z solvera LANCELOT na serwerze NEOS. Obydwa solvery służą do rozwiązywania zadań nieliniowych optymalizacji. Tak jak w przypadku MATLABa możliwe jest określenie punktu startowego oraz dokładności rozwiązania poprzez ustawienie odpowiednich parametrów. Dla solvera MINOS jest to `'optimality_tolerance=1.0e-6(default)'`, zaś dla solvera LANCELOT `'gtol=1e-5(default)'`.

4. Uzyskane wyniki

W tym rozdziale sprawdzono wpływ doboru punktu startowego, dokładności(wartości jednego z parametrów) oraz zaburzeń w danych na uzyskane wyniki zarówno dla programu MATLAB oraz AMPL. Rozwiązując zadanie optymalizacji stwierdzono, iż użytkownik znajduje się w pobliżu miejscowości Niemyje-Ząbki, co reprezentuje Rysunek 1.



Rysunek 1: Mapa ilustrująca położenie stacji oraz wyliczone położenie użytkownika

- Wpływ punktu startowego na uzyskiwane wyniki

Poniżej zaprezentowano wpływ doboru punktu startowego x_0 na uzyskane wyniki dla programu AMPL i MATLAB przy braku zaburzeń w danych i domyślnych wartościach dokładności (Tabela 1). Pierwszy punkt startowy ma współrzędne (00°00'00" E 00°00'00" N) , które utożsamiane są z początkiem układu współrzędnych. Jest to domyślna wartość dla użytych metod. W kolejnych krokach przyjęto jako punkt początkowy współrzędne stacji, jednak z pewną modyfikacją. Do każdej współrzędnej dodano 01" ze względu na błędy jakie mogą pojawić się podczas liczenia gradientu(występowanie dzielenia przez pierwiastek, który bez modyfikacji w punkcie startowym przyjmowałby wartość 0). Ostatni punkt ma współrzędne będące średnią arytmetyczną współrzędnych trzech stacji.

Położenie punktu startowego x_0 :	Metoda rozwiązania			
	MATLAB (funkcja `lsqnonlin`, metoda Levenberga-Marquardta)		AMPL	
	Wykorzystanie gradientu	Niekorzystanie z gradientu	Komputer lokalny: solver MINOS	Serwer NEOS: solver LANCELOT
00°00'00" E 00°00'00" N	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 12	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 12	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 10	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 8
22°37'24" E 52°45'55" N - w pobliżu stacji A	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 5	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 5	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 6	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 15
22°30'48" E 52°40'28" N - w pobliżu stacji B	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 5	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 5	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 7	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 20
22°44'01" E 52°43'38" N - w pobliżu stacji C	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 6	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 6	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 7	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 18
22°37'23" E 52°43'19" N -średnie położenie między stacjami	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 5	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 5	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 7	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 10

Tabela 1: Wpływ doboru punktu startowego na otrzymane wyniki

Otrzymane wyniki dla programu AMPL mają te same wartości niezależnie od punktu startowego. Różne punkty startowe w programie MATLAB wpłynęły nieznacznie na wynik obliczeń. Różnica wynosiła 01" szerokości geograficznej, która jest równa około 31 m.

Zbliżanie punktu startowego w okolice rozwiązania powoduje zazwyczaj zmniejszenie liczby iteracji. Ciekawą zależność zauważono dla solvera LANCELOT, w którym zbliżanie się punktu startowego do rozwiązania powoduje zwiększenie liczby iteracji. Może to być związane z algorytmem jaki wykorzystuje ten solver. Jest to "band matrix precond. CG".

W przypadku programu MATLAB nie widać różnicy w wynikach pomiędzy metodą wykorzystującą bądź nie gradient funkcji. Położenie oraz liczba iteracji w obydwu przypadkach jest taka sama. Różnicę można zauważył obserwując parametr `funcCount` (liczba obliczeń funkcji), który jest znacznie mniejszy przy korzystaniu z gradientu.

- Wpływ zaburzeń w danych na uzyskiwane wyniki

W tym punkcie zaprezentowano wpływ zaburzeń danych na uzyskiwane wyniki dla obydwu programów. Dla wszystkich algorytmów założono punkt startowy pokrywający się z początkiem układu współrzędnych (00°00'00" E 00°00'00" N). Wartości parametrów odpowiedzialnych za dokładność w teście stopu została pozostawiona bez zmian. Zaburzenia w danych określono jako różnice w odległościach od stacji ±1 km (Tabela 2).

Zaburzenie danych:	Metoda rozwiązania			
	MATLAB (funkcja `lsqnonlin`, metoda Levenberga-Marquardta)		AMPL	
	Wykorzystanie gradientu	Niekorzystanie z gradientu	Komputer lokalny: solver MINOS	Serwer NEOS: solver LANCELOT
Bez zaburzeń	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 12	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 12	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 10	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 8
d_A +1 km	Położenie: 22°40'46" E 52°44'19" N Liczba iteracji: 13	Położenie: 22°40'46" E 52°44'19" N Liczba iteracji: 13	Położenie: 22°40'46" E 52°44'19" N Liczba iteracji: 9	Położenie: 22°40'46" E 52°44'19" N Liczba iteracji: 8
d_A -1 km	Położenie: 22°40'14" E 52°44'52" N Liczba iteracji: 12	Położenie: 22°40'14" E 52°44'52" N Liczba iteracji: 12	Położenie: 22°40'14" E 52°44'52" N Liczba iteracji: 10	Położenie: 22°40'14" E 52°44'52" N Liczba iteracji: 14
d_B +1 km	Położenie: 22°40'58" E 52°45'05" N Liczba iteracji: 12	Położenie: 22°40'58" E 52°45'05" N Liczba iteracji: 12	Położenie: 22°40'58" E 52°45'05" N Liczba iteracji: 10	Położenie: 22°40'58" E 52°45'05" N Liczba iteracji: 16
d_B -1 km	Położenie: 22°40'06" E 52°44'14" N Liczba iteracji: 13	Położenie: 22°40'06" E 52°44'14" N Liczba iteracji: 13	Położenie: 22°40'06" E 52°44'14" N Liczba iteracji: 10	Położenie: 22°40'06" E 52°44'14" N Liczba iteracji: 8
d_C +1 km	Położenie: 22°40'13" E 52°44'51" N Liczba iteracji: 14	Położenie: 22°40'13" E 52°44'51" N Liczba iteracji: 14	Położenie: 22°40'13" E 52°44'51" N Liczba iteracji: 10	Położenie: 22°40'13" E 52°44'51" N Liczba iteracji: 11
d_C -1 km	Położenie: 22°40'51" E 52°44'28" N Liczba iteracji: 15	Położenie: 22°40'51" E 52°44'28" N Liczba iteracji: 15	Położenie: 22°40'51" E 52°44'28" N Liczba iteracji: 9	Położenie: 22°40'51" E 52°44'28" N Liczba iteracji: 9

Tabela 2: Wpływ zaburzeń w danych na otrzymane wyniki

Wpływ zaburzeń danych ma taki sam wpływ na wyniki uzyskane przez każdy z algorytmów. Wprowadzając zaburzenie do danych zwiększa się liczba iteracji algorytmów. Największą różnicę w wyniku otrzymano zwiększając odległość do stacji B o 1km. Wynikowa lokalizacja użytkownika w tym przypadku jest oddalona o około 970metrów od pierwotnej(bez zaburzeń). Zaburzenia w danych wprowadzają zatem duże zmiany uzyskiwanych wyników.

- Wpływ dokładności w teście STOP-u metody na uzyskiwane wyniki

W tym punkcie zaprezentowano wpływ dokładności w teście STOP-u dla każdego z algorytmów(Tabela 3, Tabela 4). Parametry określające dokładność dla każdego z algorytmów zostały wymienione w rozdziale 3. Ponownie wybrano punkt startowy pokrywający się z początkiem układu współrzędnych (00°00'00" E 00°00'00" N). Nie wprowadzano zaburzeń w danych.

Dokładność w teście STOP-u: wartość parametru `'optimality_tolerance'`	AMPL Komputer lokalny: solver MINOS	Dokładność w teście STOP-u: wartość parametru `'gtol'`	AMPL Serwer NEOS: solver LANCELOT
1.0e-6(default)	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 10	1.0e-5(default)	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 8
1.0e+1	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 8	1.0e+2	Położenie: 22°40'31" E 52°44'39" N Liczba iteracji: 7
1.0e+2	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 7 (31 m od prawidłowego rozwiązania)	1.0e+3	Położenie: 22°40'32" E 52°44'39" N Liczba iteracji: 6 (19 m od prawidłowego rozwiązania)
1.0e+4	Położenie: 22°40'26" E 52°44'26" N Liczba iteracji: 5 (412m od prawidłowego rozwiązania)	1.0e+5	Położenie: 22°42'18" E 52°40'51" N Liczba iteracji: 3 (7.3km od prawidłowego rozwiązania)
1.0e+6	Położenie: 23°13'27" E 52°43'28" N Liczba iteracji: 3 (37 km od prawidłowego rozwiązania)	1.0e+6	Położenie: 22°43'40" E 52°31'29" N Liczba iteracji: 2 (24.6km od prawidłowego rozwiązania)

Tabela 3: Wpływ dokładności w teście STOP-u metody na otrzymane wyniki - AMPL

Ze względu na inne parametry określające dokładność metody dla solvera MINOS i LANCELOT ich wpływ na rozwiązanie jest również inny. Można jednak zauważyć, iż zwiększając wartość tych parametry zmniejsza się liczba iteracji, ale dokładność wyniku również się zmniejsza. Dla wartości parametrów większych o 10^7 razy od wartości domyślnych nie widać wpływu na dokładność rozwiązania – maleje wyłącznie liczba iteracji. Zwiększenie jeszcze 10-krotnie tej wartości powoduje niewielkie zmiany w otrzymanym wyniku, zaś dalsze zwiększenia powoduje duże zmiany w uzyskanych wynikach, które nie są akceptowalne.

Dokładność w teście STOP-u: wartość parametru `'StepTolerance'`	Metoda rozwiązania	
	MATLAB (funkcja `lsqnonlin`, metoda Levenberga-Marquardta)	
	Wykorzystanie gradientu	Niekorzystanie z gradientu
1.0e-6(default)	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 12	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 12
1.0e-4	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 10	Położenie: 22°40'31" E 52°44'38" N Liczba iteracji: 10
1.0e-3	Położenie: 22°40'31" E 52°44'47" N Liczba iteracji: 9	Położenie: 22°40'31" E 52°44'47" N Liczba iteracji: 9
1.0e-2	Położenie: 22°38'48" E 52°47'05" N Liczba iteracji: 7	Położenie: 22°38'48" E 52°47'05" N Liczba iteracji: 7

Tabela 4: Wpływ dokładności w teście STOP-u metody na otrzymane wyniki - MATLAB

Wartość parametru `'StepTolerance'` ma taki sam wpływ na algorytm z gradientem i nie. Ze wzrostem wartości parametru zmniejsza się liczba iteracji, jednak kosztem dokładności wyniku (zwiększana jest wartość tolerancji kolejnych kroków). Przy wartości o 10 większej nie widać wpływu na otrzymane wyniki, więc skutecznie zmniejszono liczbę iteracji. Zwiększając dalej wartość tego parametru otrzymuje się coraz gorsze wyniki(dla 1e-2 odległość wynikowego punktu od pierwotnego wynosi około 5km).