Lab 3c,
Programowanie matematyczne

Piotr Onyszczuk, gr. C

15 XI 2021

1 Treść zadania

Celem zadania jest rozwiązanie za pomocą funkcji **linprog** oraz za pomocą własnej implementacji algorytmu **symplex** następującego zagadnienia.

$$\max_{x \in \Omega} c^T x \tag{1}$$

$$\Omega : \begin{cases} Ax \leq b & \text{, b>0} \\ |x| \leq g & \text{, g>0} \end{cases} \tag{2}$$

$$c, x \in R^n; b \in R^m; g \in R^n; A \in R^{mxn}; m = n \tag{3}$$

1.1 Dane testowe

Do testów należy użyć losowe wektory (n=5, m=5).

- c i A zawierają wartości całkowite z przedziału [-2,2]
- b i g zawierają wartości całkowite z przedziału [1,5]

2 Algorytm

Algorytm opiszę na jednym z wylosowanych przykładów (Wycinek 1)

2.1 Sprowadzenie do postaci standardowej

Pierwszym etapem jest sprowadzenie układu do postaci standardowej. Etap ten składa się z kilku kroków.

1. Z racji tego że x mogą przyjmować wartości całkowite, pierwszym krokiem jest rozbicie każdej zmiennej x_i na nieujemne ${x_i}^+$ oraz ${x_i}^-$. Do wektora c

```
A =

     2     0    -2    -2    -2
     1     2     1     0     0
    -1    -1    -1    -2     1
     1    -1    -1     0     0
    -2    -2     0     1    -2


b =

     3     4     3     5     3


c =

     2    -2    -2     2    -1


g =

     1     5     2     4     4
```

Wycinek 1: Wylosowany przykład

oraz macierzy A dopisujemy ich kopie ze zmienionym znakiem. Pozostałe wektory się nie zmieniają.

$$x_1 + ... + x_5 \leq b^1 => {x_1}^+ + {x_1}^- ... + {x_5}^+ + {x_5}^- \leq b^1, itd. \quad (4)$$

(Wycinek 2)

2. Drugim krokiem jest zamiana nierówności na równości. W tym celu dodajemy 5 zmiennych dopełniających. Po jednej do każdego równania. Zmieniają się znowu A i c.

$$ {x_1}^+ + {x_1}^- ... + {x_5}^+ + {x_5}^- \leq b^1 => {x_1}^+ ... + {x_5}^- + x_6 = b^1, itd. \quad (5)$$

(Wycinek 3)

3. W ostatnim kroku musimy uwzględnić ograniczenia na x. Z racji tego, że w pierwszym kroku rozbiliśmy nasze zmienne na część dodatnią i ujemną, poradziliśmy sobie od razu z wartością bezwzględną. Teraz wystarczy, że nasze 10 zmiennych z pierwszego kroku będą $\leq$ od swoich odpowiednich ograniczeń. Od razu dodajemy również nowe zmienne dopełniające, aby zmienić nierówności na równania. Dochodzi nam zatem 10 nowych równań i 10 zmiennych. (Wycinki 4, 5)

$$ {x_1}^+ + x_{11} = g^1, {x_1}^- + x_{12} = g^1, {x_2}^+ + x_{13} = g^2, itd \quad (6)$$

```
c =

    -2     0     2     0     1     2     0    -2     0    -1

A =

     0    -2     0     1     2     0     2     0    -1    -2
     1    -2     1     0     2    -1     2    -1     0    -2
    -2    -1    -1     0    -1     2     1     1     0     1
    -1    -1     2    -2     1     1     1    -2     2    -1
    -2     0    -2    -2     2     2     0     2     2    -2
```

Wycinek 2: Krok 1

```
c =

    -2     0     2     0     1     2     0    -2     0    -1     0     0     0     0     0

A =

     0    -2     0     1     2     0     2     0    -1    -2     1     0     0     0     0
     1    -2     1     0     2    -1     2    -1     0    -2     0     1     0     0     0
    -2    -1    -1     0    -1     2     1     1     0     1     0     0     1     0     0
    -1    -1     2    -2     1     1     1    -2     2    -1     0     0     0     1     0
    -2     0    -2    -2     2     2     0     2     2    -2     0     0     0     0     1
```

Wycinek 3: Krok 2

```
c =

    -2     0     2     0     1     2     0    -2     0    -1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

A =

     0    -2     0     1     2     0     2     0    -1    -2     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     1    -2     1     0     2    -1     2    -1     0    -2     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0
    -2    -1    -1     0    -1     2     1     1     0     1     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0
    -1    -1     2    -2     1     1     1    -2     2    -1     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0
    -2     0    -2    -2     2     2     0     2     2    -2     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1
```

Wycinek 4: Krok 3

```
b =

     5
     1
     1
     1
     5
     2
     4
     5
     1
     4
     2
     4
     5
     1
     4
```

Wycinek 5: Krok 3

2.2 Symplex

Następnie rozpoczynamy algorytm symplex. Z racji tego że mamy zagadnienie w postaci standardowej, możemy wybrać podstawową wersję. Jako bazę wybieramy 15 zmiennych dopełniających. Stanowią one bazowe dopuszczalne rozwiązanie. Stąd też późniejszy wniosek, że algorytm zawsze powinien znaleźć rozwiązanie. Odnosząc to do pierwotnej wersji zagadnienia, punktem startowym jest punkt złożony z samych zer.

Dla tego BRD zadanie jest w postaci kanonicznej. Wynika to z dwóch rzeczy. Po pierwsze w bazie mamy tylko zmienne dopełniające, dodawane pojedynczo do równań (z plusem), wówczas zawsze mamy postać kanoniczną. Po drugie, dość łatwo jest to zauważyć na Wycinku 4. Ostatnie 15 kolumn tworzy macierz jednostkową.

Następnie powtarzamy w pętli:

- Obliczamy z jako iloczyn współczynników z wektora c dla zmiennych bazowych z macierzą A
- Obliczamy $z - c$
- Znajdujemy minimalny element w $z - c$ (jego indeks i).
- W kolumnie macierzy A o indeksie i znajdujemy dodatni element $A(j, i)$ maksymalizujący wartość $b(j)/A(j, i)$.
- Wykonujemy odpowiednią eliminację Gaussa
- Aktualizujemy bazę zamieniając zmienną na pozycji j, zmienną x_i.

Pętlę powtarzamy dopóki $z - c$ zawiera wartości ujemne (z dokładnością do ustalonego epsilona).

2.3 Uzyskanie rozwiązania

Aby uzyskać rozwiązanie:

1. Tworzymy pusty wektor 25-elementowy - wynikowy
2. Uzupełniamy pola odpowiadające zmiennym z bazy, wartościami z wektora b
3. Wynikiem jest iloczyn pierwszych 10 elementów z wektora c z pierwszymi 10-cioma elementami z wynikowego wektora
4. Punkt wynikowy uzyskujemy dodając dwa wektory: wektor złożony z pierwszych pięciu elementów wekotra wynikowego oraz wektor złożony z kolejnych pięciu elementów. Odwracamy w ten sposób pierwszy krok przekształcenia zadania.

3 Wyniki i wnioski

Testując na 1000 przypadków, oba algorytmy zawsze zwracają wynik, (prawie) zawsze ten sam wynik (z różnicą co do znaku). Stąd też potwierdzenie sygnalizowanego wcześniej przypuszczenia, że zagadnienie zawsze posiada rozwiązanie. Niemożliwe jest w takim razie podanie przykładu bez rozwiązania.

Na 1000 przypadków:

- Moja implementacja zwraca wynik szybciej (mniej iteracji) niż linprog 11 razy
- Wyniki różnią się raz. Różnica jest dość mała (< 0.2). Zakładam błędy numeryczne. W załączonym pliku jest to zadanie dla $i = 246$
- Linprog wykonuje średnio ok. 3 razy mniej iteracji niż moja implementacja.
- Moja implementacja wykonuje maksymalnie 13 iteracji, podczas gdy linprog maksymalnie 6

4 Pliki

- Losowanie.m - skrypt losujący dane do zadania
- Rozwiazanie.m - główny skrypt, losuje zadania w pętli i uruchamia oba algorytmy
- my_simplex.m - zawiera implementację algorytmu symplex wraz z wypisywaniem wyniku

- my_linprog.m - zawiera funkcję przyjmującą parametry zgodne z poleceniem, przekształca zadanie i wykonuje funkcję my_simplex. Aby uzyskać wypisywanie stanu, należy przestawić flagę przekazywaną do my_simplex na 1.