TEORIA WSPÓŁBIEŻNOŚCI 2023 / 2024

Laboratorium 7 - Zadanie Domowe 2

Współbieżna Eliminacja Gaussa z Wykorzystaniem Teorii Śladów

Wiktor Wilkusz

Spis treści

1. Wstęp

Ćwiczenie polega na stworzeniu współbieżnego algorytmu eliminacji Gaussa z wykorzystaniem Teorii Śladów. Algorytm na wejściu otrzymuje rozmiar macierzy *N*, macierz współczynników $M_{i,j}$ (gdzie *i* to numer wiersza, a *j* to numer kolumny) o rozmiarze *N x N* oraz wektor wyrazów wolnych *y* o rozmiarze *N*. Zadaniem algorytmu jest znalezienie wektora niewiadomych *x*. Rozpatrywane równanie macierzowe jest widoczne poniżej.

$$M \times x = y$$

$$\begin{bmatrix} M_{1,1} & M_{1,2} & \dots & M_{1,N} \\ M_{2,1} & M_{2,2} & \dots & M_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ M_{N,1} & M_{N,2} & \dots & M_{N,N} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

2. Niepodzielne operacje wykonywane przez algorytm

Wyróżniono następujące niepodzielne operacje wykonywane przez algorytm:

- $A_{k,i}$ – znalezienie $m_{k,i}$, czyli mnożnika dla wiersza *i*-tego, potrzebnego do odjęcia wiersza *i*-tego od wiersza *k*-tego. Wzór na $m_{k,i}$ jest widoczny poniżej

$$m_{k,i} = \frac{M_{k,i}}{M_{i,i}}$$

- $B_{k,i,j}$ - znalezienie $n_{k,i,j}$, czyli *j*-tego elementu wiersza *i*-tego pomnożonego przez mnożnik $m_{k,i}$, potrzebnego do odejmowania od *j*-tego elementu *k*-tego wiersza. Wzór na $n_{k,i,j}$ jest widoczny poniżej

$$n_{k,i,j} = M_{i,j} \cdot m_{k,i}$$

- $C_{k,i,j}$ - odjęcie $n_{k,i,j}$ od j-tego elementu wiersza k-tego. Operacja ta jest widoczna poniżej

$$M_{k,j} = M_{k,j} - n_{k,i,j}$$

3. Alfabet w sensie Teorii Śladów

Alfabet w sensie Teorii Śladów zdefiniowano następująco:

$$\Sigma = \{A_{k,i} \mid 1 \le i < N,\ i < k \le N\} \cup$$
$$\{B_{k,i,j} \mid 1 \le i < N,\ i < k \le N,\ i \le j \le N + 1\} \cup$$
$$\{C_{k,i,j} \mid 1 \le i < N,\ i < k \le N,\ i \le j \le N + 1\}$$

4. Relacja zależności opisująca algorytm

W celu wyznaczenia relacji zależności opisującej algorytm wprowadzono pomocniczą tabelę (Tabela 1.) przedstawiającą dostęp do macierzy i wprowadzonych w punkcie 2 zmiennych przez poszczególne operacje.

operacja	odczyt	modyfikacja
$A_{k,i}$	$M_{k,i}$, $M_{i,i}$	$m_{k,i}$
$B_{k,i,j}$	$M_{i,j}$, $m_{k,i}$	$n_{k,i,j}$
$C_{k,i,j}$	$M_{k,j}$, $n_{k,i,j}$	$M_{k,j}$

Tabela 1. Dostęp do macierzy or wprowadzonych w punkcie 2 zmiennych przez poszczególne operacje

1) operacja $B_{k,i,j}$ odczytuje zmienną $m_{k,i}$, która jest modyfikowana przez operację $A_{k,i}$

$$D_1 = \{(A_{k,i}, B_{k,i,j}) \mid A_{k,i}, B_{k,i,j} \in \Sigma\}$$

2) operacja $C_{k,i,j}$ odczytuje zmienną $C_{k,i,j}$, która jest modyfikowana przez operację $B_{k,i,j}$

$$D_2 = \{(B_{k,i,j}, C_{k,i,j}) \mid B_{k,i,j}, C_{k,i,j} \in \Sigma\}$$

3) operacja $C_{k,i,j}$ modyfikuje zmienną $M_{k,j}$, która:

a) jest odczytywana przez operację $A_{g,h}$, o ile: $(g = k \wedge h = j) \vee (k = h \wedge j = h)$, co jest równoważne wyrażeniu: $(h = j) \wedge (g = k \vee k = j)$

$$D_{3a} = \{(C_{k,i,j}, A_{g,h}) \mid C_{k,i,j}, A_{g,h} \in \Sigma \wedge (h = j) \wedge (g = k \vee k = h)\}$$

b) jest odczytywana przez operację $B_{l,g,h}$, o ile: $g = k \wedge h = j$

$$D_{3b} = \{(C_{k,i,j}, B_{l,g,h}) \mid C_{k,i,j}, B_{l,g,h} \in \Sigma \wedge g = k \wedge h = j\}$$

c) jest odczytywana przez operację $C_{l,g,h}$, o ile: $l = k \wedge h = j$

$$D_{3c} = \{(C_{k,i,j}, C_{l,g,h}) \mid C_{k,i,j}, C_{l,g,h} \in \Sigma \wedge l = k \wedge h = j\}$$

Uwzględniając symetrię, przechodniość i identyczność, relacja zależności wygląda następująco:

$$D = sym\{\{D_1 \cup D_2 \cup D_{3a} \cup D_{3b} \cup D_{3c}\}^+\} \cup I_\Sigma$$

Relacja niezależności wygląda następująco:

$$I = \Sigma^2 - D$$

5. Algorytm eliminacji Gaussa w postaci ciągu symboli alfabetu

Stworzony zostanie ciąg operacji symbolizujący wykonanie algorytmu eliminacji Gaussa. W tym celu definiujemy Operację $S_{k,i}$ oznaczającą odjęcie i-tego wiersza od wiersza k-tego, której rezultatem jest wyzerowanie elementu $M_{k,i}$. Operacja $S_{k,i}$ to ciąg operacji zdefiniowanych w punkcie 2.

$$S_{k,i} = A_{k,i}, B_{k,i,i}, C_{k,i,i}, B_{k,i,i+1}, C_{k,i,i+1}, \ldots, B_{k,i,N}, C_{k,i,N}, B_{k,i,N+1}, C_{k,i,N+1}$$

Algorytm eliminacji Gaussa przyjmuje wtedy następującą postać:

$$S_{2,1}, S_{3,1}, S_{4,1}, \ldots, S_{N,1}, S_{3,2}, S_{4,2}, S_{5,2}, \ldots, S_{N,N-2}, S_{N,N-1}$$

6. Graf zależności Diekerta

Na podstawie relacji zależności otrzymanych w punkcie 4 zostanie stworzy graf zależności Diekerta.

$$E_1 = \{(A_{k,i}, B_{k,i,j}) \mid A_{k,i}, B_{k,i,j} \in \Sigma\}$$

$$E_2 = \{(B_{k,i,j}, C_{k,i,j}) \mid B_{k,i,j}, C_{k,i,j} \in \Sigma\}$$

$$E_{3a} = \{(C_{k,i,j}, A_{g,h}) \mid C_{k,i,j}, A_{g,h} \in \Sigma \land (h = j) \land (g = k \lor k = h) \land j = h - 1\}\}$$

$$E_{3b} = \{(C_{k,i,j}, B_{l,g,h}) \mid C_{k,i,j}, B_{l,g,h} \in \Sigma \land g = k \land h = j \land i = g - 1\}$$

$$E_{3c} = \{(C_{k,i,j}, C_{l,g,h}) \mid C_{k,i,j}, C_{l,g,h} \in \Sigma \wedge l = k \wedge h = j \wedge i = g - 1\}$$

Ostatecznie graf przyjmuje następującą postać (zbiór krawędzi):

$$E = E_1 \cup E_2 \cup E_{3a} \cup E_{3b} \cup E_{3c}$$

Do wygenerowania grafu zależności Diekerta dla macierzy o rozmiarze N = 3 wykorzystano program stworzony w zadaniu 1 z laboratorium 5. Program wraz z opisem (*readme.txt)* znajduje się w podkatalogu ‘*graph_foata_generator*’.

Dostarczony do programu plik wejściowy znajduje się na Rysunku 1, a część otrzymanego wyjścia (bez relacji zależności i niezależności) na Rysunku 2.

```
inputs > input.txt
(A21) m21 := M21 / M11
(A31) m31 := M31 / M11
(A32) m32 := M32 / M22
(B211) n211 := M11 * m21
(B212) n212 := M12 * m21
(B213) n213 := M13 * m21
(B214) n214 := M14 * m21
(B311) n311 := M11 * m31
(B312) n312 := M12 * m31
(B313) n313 := M13 * m31
(B314) n314 := M14 * m31
(B322) n322 := M22 * m32
(B323) n323 := M23 * m32
(B324) n324 := M24 * m32
(C211) M21 := M21 - n211
(C212) M22 := M22 - n212
(C213) M23 := M23 - n213
(C214) M24 := M24 - n214
(C311) M31 := M31 - n311
(C312) M32 := M32 - n312
(C313) M33 := M33 - n313
(C314) M34 := M34 - n314
(C322) M32 := M32 - n322
(C323) M33 := M33 - n323
(C324) M34 := M34 - n324

A = {A21, A31, A32, B211, B212, B213, B214, B311, B312, B313, B314, B322, B323, B324, C211, C221, C231, C241, C311,
C321, C231, C341, C322, C323, C324}

w = A21 B211 C211 B212 C212 B213 C213 B214 C214 A31 B311 C311 B312 C312 B313 C313 B314 C314 A32 B322 C322 B323 C323
B324 C324
```

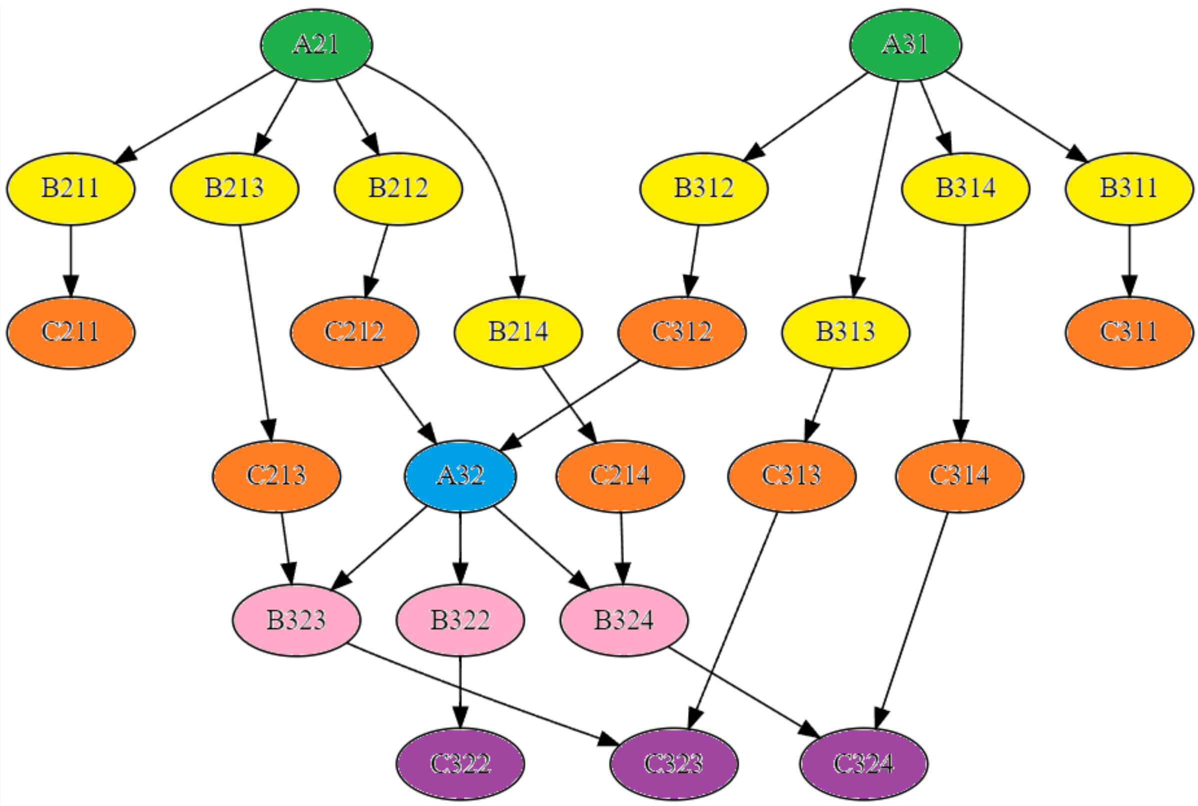
Rysunek 1. Plik wejściowy dostarczony do programu generującego graf zależności Diekerta i klasy Foaty

```
FNF: (A21A31)(B211B212B213B214B311B312B313B314)(C211C212C213C214C311C312C313C314)(A32)(B322B323B324)(C322C323C324)

Graph (dot): digraph "graph" {
    0 [label=A21]
    1 [label=B211]
    2 [label=C211]
    3 [label=B212]
    4 [label=C212]
    5 [label=B213]
    6 [label=C213]
    7 [label=B214]
    8 [label=C214]
    9 [label=A31]
    10 [label=B311]
    11 [label=C311]
    12 [label=B312]
    13 [label=C312]
    14 [label=B313]
    15 [label=C313]
    16 [label=B314]
    17 [label=C314]
    18 [label=A32]
    19 [label=B322]
    20 [label=C322]
    21 [label=B323]
    22 [label=C323]
    23 [label=B324]
    24 [label=C324]
    0 -> 1
    0 -> 3
    0 -> 5
    0 -> 7
    1 -> 2
    3 -> 4
    4 -> 18
    5 -> 6
    6 -> 21
    7 -> 8
    8 -> 23
    9 -> 10
    9 -> 12
    9 -> 14
    9 -> 16
    10 -> 11
    12 -> 13
    13 -> 18
    14 -> 15
    15 -> 22
    16 -> 17
    17 -> 24
    18 -> 19
    18 -> 21
    18 -> 23
    19 -> 20
    21 -> 22
    23 -> 24
}
```

Rysunek 2. Otrzymane wyjście programu generującego graf zależności Diekerta i klasy Foaty

Za pomocą strony Graphviz Online wygenerowano wizualizację otrzymanego grafu. Widoczna jest ona na Rysunku 3, na którym zaznaczono również otrzymane klasy Foaty.



Rysunek 3. Wizualizacja otrzymanego grafu Diekerta dla N=3 z pokolorowanymi klasami Foaty

7. Postać normalna Foaty

Na podstawie klas Foaty otrzymanych dla macierzy o rozmiarze N=3, wyprowadzono ich uogólnioną postać.

Klasy Foaty rozdzielają się na 3 podgrupy:

- F_{Ax} - znalezienie wszystkich mnożników *m* potrzebnych do znalezienia współczynników *n* potrzebnych do wyzerowania kolumny o indeksie *x* - operacje *A*

$$F_{Ax} = \{A_{k,x} \mid x < k \leq N,\ 1 \leq x < N - 1\}$$

- F_{Bx} - znalezienie wszystkich współczynników *n* potrzebnych do wyzerowania kolumny o indeksie *x* - operacje B

$$F_{Bx} = \{B_{k,x,j} \mid x < k \le N \wedge x \le j \le N + 1, 1 \le x < N - 1\}$$

- F_{Cx} - odejmowanie współczynnika *n* w celu wyzerowania kolumny o indeksie x

$$F_{Cx} = \{C_{k,x,j} \mid x < k \le N \wedge x \le j \le N + 1, 1 \le x < N - 1\}$$

Postać normalna Foaty przyjmuje następującą postać:

$$FNF = F_{A1}F_{B1}F_{C1}F_{A2}F_{B2}F_{C2} \dots F_{AN-1}F_{BN-1}F_{CN-1}$$

8. Implementacja algorytmu

Algorytm został zaimplementowany w języku Java. Instrukcja uruchomienia znajduje się w pliku *readme.txt* w katalogu *concurrent_gaussian_elimination*.