**Ministerul Educației și Cercetării al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

Text

Description automatically generated

**Departamentul Ingineria Software și Automatica**

Raport

Lucrare de control

**Matematică discretă**

Varianta 4

|  |  |  |
| --- | --- | --- |
| **A efectuat:** | Student grupa TI-231 FR | Apareci Aurica |
| **A verificat:** | Lector universitar | Ceban Gherghe |

**Chișinău**

**2024**

**Problema 1.** Este dată funcția logică f(*x1,x2,x3,x4*) prin setul de valori a argumentelor pentru care primește valoarea 1

1. De alcătuit tabelul de adevăr
2. De obținut forma canonică disjunctivă normală (FCDN) și forma canonică conjunctivă normală (FCCN)
3. De minimizat FCDN prin 3 metode: Quine, Quine-McKluskey, Karnaugh
4. De implementat schema logică în baza ȘI-NU, SAU-NU
5. De construit diagrama temporară pentru funcția f(*x1,x2,x3,x4*)

f(*x1,x2,x3,x4*)=∨(2,3,6,7,8,14,15)

|  |  |  |  |
| --- | --- | --- | --- |
| Tabelul de adevăr | FCDN | | FCCN |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **X1** | **X2** | **X3** | **X4** | **f** | | **0** | 0 | 0 | 0 | 0 | 0 | | **1** | 0 | 0 | 0 | 1 | 0 | | **2** | 0 | 0 | 1 | 0 | 1 | | **3** | 0 | 0 | 1 | 1 | 1 | | **4** | 0 | 1 | 0 | 0 | 0 | | **5** | 0 | 1 | 0 | 1 | 0 | | **6** | 0 | 1 | 1 | 0 | 1 | | **7** | 0 | 1 | 1 | 1 | 1 | | **8** | 1 | 0 | 0 | 0 | 1 | | **9** | 1 | 0 | 0 | 1 | 0 | | **10** | 1 | 0 | 1 | 0 | 0 | | **11** | 1 | 0 | 1 | 1 | 0 | | **12** | 1 | 1 | 0 | 0 | 0 | | **13** | 1 | 1 | 0 | 1 | 0 | | **14** | 1 | 1 | 1 | 0 | 1 | | **15** | 1 | 1 | 1 | 1 | 1 | | |  | | --- | | ∨ | | ∨ | | ∨ | | ∨ | | ∨ | | ∨ | | ∨ | | | |  | | --- | | (∨∨∨)^ | | (∨∨∨)^ | | (∨∨∨^ | | (∨∨∨)^ | | (∨∨∨)^ | | (∨∨∨)^ | | (∨∨∨)^ | | (∨∨∨)^ | | (∨∨∨)^ | |
| Minimizarea FCDN - Quine | | | |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **FCDN:** |  | **I Alipire** |  | **II Alipire** | **Implicanți primi:** | | |  | | --- | | ∨ | | ∨ | | ∨ | | ∨ | | **∨** | | ∨ | | ∨ | | ~~1~~  ~~2~~  ~~3~~  ~~4~~  **5**  ~~6~~  ~~7~~ | |  | | --- | | ∨ | | ∨ | | ∨ | | ∨ | | ∨ | | ∨ | | ∨ | | ~~1~~  ~~2~~  ~~3~~  ~~4~~  ~~5~~  ~~6~~  ~~7~~ | |  | | --- | | ∨ | | ∨ | |  | | ∨ | | A:  B:  C:  𝐹CM=A∨B∨C  𝐹CM=∨∨ |   **Tabela de acoperire:**   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  | |  | 1 | 1 | 1 | 1 | 0 | 0 | 0 | |  | 0 | 0 | 1 | 1 | 0 | 1 | 1 | |  | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | | | |
| Minimizarea FCDN – Quine-McKluskey | | | |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | 0010 | | 0011 | | 0110 | | 0111 | | 1000 | | 1110 | | 1111 | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Nivelul 0** | 0010  1000 | | **Nivelul 1** | 0011  0110 | | **Nivelul 2** | 0111  1110 | | **Nivelul 3** | 1111 | | | |  |  | | --- | --- | | **Nivelul 0** | 001\_  0\_10 | | **Nivelul 1** | 0\_11  011\_  \_110 | | **Nivelul 2** | \_111  111\_ | | |  |  | | --- | --- | | **Nivelul 0** | 0\_1\_  0\_1\_ | | **Nivelul 1** | \_11\_  \_11\_ | | **Implicanți primi:**  A: 0\_1\_  B: \_11\_  C: 1000 |   **Tabela de acoperire:**   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | 0010 | 1000 | 0011 | 0110 | 0111 | 1110 | 1111 | | 0\_1\_ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | | \_11\_ | 0 | 0 | 1 | 1 | 0 | 1 | 1 | | 1000 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | | | |
| Minimizarea FCDN – Karnaugh | | Minimizarea FCCN – Karnaugh | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | 00 | 01 | 11 | 10 | | 00 |  |  |  | 1 | | 01 |  |  |  |  | | 11 | 1 | 1 | 1 |  | | 10 | 1 | 1 | 1 |  |   𝐹DM=∨∨ | | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | 00 | 01 | 11 | 10 | | 00 | 0 | 0 | 0 |  | | 01 | 0 | 0 | 0 | 0 | | 11 |  |  |  | 0 | | 10 |  |  |  | 0 |   𝐹CM= | |
| **ȘI-NU** | | | |
| 𝐹CM=∨∨ 𝐹CM=(( | | | |
| **SAU-NU** | | | |
| 𝐹DM= 𝐹DM= | | | |
| Schema logică a funcției **ȘI-NU** | | | |
|  | | | |
| Schema logică a funcției **SAU-NU** | | | |
|  | | | |
| Diagrama temporară a funcției | | | |
| **A black and white rectangular object with red lines  Description automatically generated** | | | |

**Problema 2.** Utilizând algoritmul Ford și Bellman-Kalaba de aflat drumurile de valoare minimă și maximă între vârfurile 1 și 8 în graful dat.

A diagram of a network

Description automatically generated

**Algoritmul Ford**

Permite determinarea drumului minim care începe cu un vârf iniţial xi până la oricare vârf al grafului G. Dacă prin Pij se va nota ponderea arcului (xi, xj) atunci algoritmul conţine următorii paşi:

1. Fiecărui vârf xj al grafului G se va ataşa un număr foarte mare Hj(∞). Vârfului iniţial i se va ataşa Ho = 0;
2. Se vor calcula diferenţele Hj - Hi pentru fiecare arc (xi, xj). Sunt posibile trei cazuri:

a) Hj - Hi < Pij,

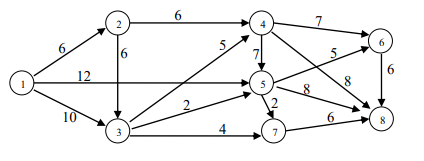
b) Hj - Hi = Pij,

c) Hj - Hi > Pij. Cazul "c" permite micşorarea distanţei dintre vârful iniţial şi xj din care cauză se va realiza Hj = Hi + Pij.

1. Pasul 2 se va repeta atâta timp cât vor mai exista arce pentru care are loc inegalitatea “c”. La terminare, etichetele Hi vor defini distanţa de la vârful iniţial până la vârful dat xi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | XiXj | Pij | Hj-Hi | Hj-Hi |
|  | (1,2) | 6 | H2-H1 = ∞ - 0 > 6; H2 = H1 + 6 = 6 | H2-H1 = 6 - 0 = 6 |
|  | (1,3) | 10 | H3-H1 = ∞ - 0 > 10; H3 = H1 + 10 = 10 | H3-H1 = 10 - 0 = 10 |
|  | (1,5) | 12 | H5-H1 = ∞ - 0 > 12; H5 = H1 + 12 = 12 | H5-H1 = 12 - 0 = 12 |
| H1 = 0 | (2,3) | 6 | H3-H2 = 10 - 6 < 6; nu se schimbă | H3-H2 = 10 - 6 < 6 |
| H2 = ∞/6 | (2,4) | 6 | H4-H2 = ∞ - 6 > 6; H4 = H2 + 6 = 12 | H4-H2 = 12 - 6 = 6 |
| H3 = ∞/10 | (3,4) | 5 | H4-H3 = 12 - 10 < 5; nu se schimbă | H4-H3 = 12 - 10 < 5 |
| H4 = ∞/12 | (3,5) | 2 | H5-H3 = 12 - 10 = 2; nu se schimbă | H5-H3 = 12 - 10 = 2 |
| H5 = ∞/12 | (3,7) | 4 | H7-H3 = ∞ - 10 > 4; H7 = H3 + 4 = 14 | H7-H3 = 14 - 10 = 4 |
| H6 = ∞/19/17 | (4,5) | 7 | H5-H4 = 12 - 12 < 7; nu se schimbă | H5-H4 = 12 - 12 < 7 |
| H7 = ∞/14 | (4,6) | 7 | H6-H4 = ∞ - 12 > 7; H6 = H4 + 7 = 19 | H6-H4 = 17 - 12 < 7 |
| H8 = ∞/20 | (4,8) | 8 | H8-H4 = ∞ - 12 > 8; H8 = H4 + 8 = 20 | H8-H4 = 20 - 12 = 8 |
|  | (5,6) | 5 | H6-H5 = 19 - 12 > 5; H6 = H5 + 5 = 17 | H6-H5 = 17 - 12 = 5 |
|  | (5,7) | 2 | H7-H5 = 14 - 12 = 2; nu se schimbă | H7-H5 = 14 - 12 = 2 |
|  | (5,8) | 8 | H8-H5 = 20 - 12 = 8; nu se schimbă | H8-H5 = 20 - 12 = 8 |
|  | (6,8) | 6 | H8-H6 = 20 - 17 < 6; nu se schimbă | H8-H6 = 20 - 17 < 6 |
|  | (7,8) | 6 | H7-H5 = 20 - 14 = 6; nu se schimbă | H7-H5 = 20 - 14 = 6 |

distmin(1,8)=20



1→2→4→8

1→3→5→7→8

1→3→5→8

1→3→7→8

1→5→7→8

1→5→8

**Algoritmul Ford**

Permite determinarea drumului maxim care începe cu un vârf iniţial xi până la oricare vârf al grafului G. Dacă prin Pij se va nota ponderea arcului (xi, xj) atunci algoritmul conţine următorii paşi:

1. Fiecărui vârf xj al grafului G se va ataşa un număr foarte mic Hj(-∞). Vârfului iniţial i se va ataşa Ho = 0;
2. Se vor calcula diferenţele Hj - Hi pentru fiecare arc (xi, xj). Sunt posibile trei cazuri:

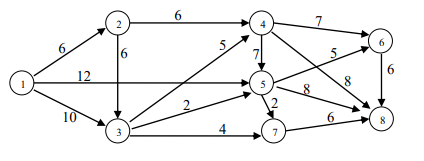
a) Hj - Hi > Pij,

b) Hj - Hi = Pij,

c) Hj - Hi < Pij. Cazul "c" permite mărirea distanţei dintre vârful iniţial şi xj din care cauză se va realiza Hj = Hi + Pij.

1. Pasul 2 se va repeta atâta timp cât vor mai exista arce pentru care are loc inegalitatea “c”. La terminare, etichetele Hi vor defini distanţa de la vârful iniţial până la vârful dat xi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | XiXj | Pij | Hj-Hi | Hj-Hi |
|  | (1,2) | 6 | H2-H1 = -∞ - 0 < 6; H2 = H1 + 6 = 6 | H2-H1 = 6 - 0 = 6 |
|  | (1,3) | 10 | H3-H1 = -∞ - 0 < 10; H3 = H1 + 10 = 10 | H3-H1 = 12 - 0 > 10 |
|  | (1,5) | 12 | H5-H1 = -∞ - 0 < 12; H5 = H1 + 12 = 12 | H5-H1 = 24 - 0 > 12 |
| H1 = 0 | (2,3) | 6 | H3-H2 = 10 - 6 < 6; H3 = H2 + 6 = 12 | H3-H2 = 12 - 6 = 6 |
| H2 = -∞/6 | (2,4) | 6 | H4-H2 = -∞ - 6 < 6; H4 = H2 + 6 = 12 | H4-H2 = 17 - 6 > 6 |
| H3 = -∞/10/12 | (3,4) | 5 | H4-H3 = 12 - 12 < 5; H4 = H3 + 5 = 17 | H4-H3 = 17 - 12 = 5 |
| H4 = -∞/12/17 | (3,5) | 2 | H5-H3 = 12 - 12 < 2; H5 = H3 + 2 = 14 | H5-H3 = 24 - 12 > 2 |
| H5 = -∞/12/14/24 | (3,7) | 4 | H7-H3 = -∞ - 12 < 4; H7 = H3 + 4 = 16 | H7-H3 = 26 - 12 > 4 |
| H6 = -∞/24/29 | (4,5) | 7 | H5-H4 = 14 - 17 < 7; H5 = H4 + 7 = 24 | H5-H4 = 24 - 17 = 7 |
| H7 = -∞/16/26 | (4,6) | 7 | H6-H4 = -∞ - 17 < 7; H6 = H4 + 7 = 24 | H6-H4 = 29 - 17 > 7 |
| H8 = -∞/25/35 | (4,8) | 8 | H8-H4 = -∞ - 17 < 8; H8 = H4 + 8 = 25 | H8-H4 = 35 - 17 > 8 |
|  | (5,6) | 5 | H6-H5 = 24 - 24 < 5; H6 = H5 + 5 = 29 | H6-H5 = 29 - 24 = 5 |
|  | (5,7) | 2 | H7-H5 = 16 - 24 < 2; H7 = H5 + 2 = 26 | H7-H5 = 26 - 24 = 2 |
|  | (5,8) | 8 | H8-H5 = 26 - 8 > 8; nu se schimbă | H8-H5 = 35 - 24 > 8 |
|  | (6,8) | 6 | H8-H6 = 25 - 29 < 6; H8 = H6 + 6 = 35 | H8-H6 = 35 - 29 = 6 |
|  | (7,8) | 6 | H7-H5 = 35 - 26 > 6; nu se schimbă | H7-H5 = 35 - 26 > 6 |



distmax(1,8)=35

1→2→3→4→5→6→8

**Algoritmul Bellman-Calaba**

Etapa iniţială presupune ataşarea grafului dat G a unei matrice ponderate de adiacenţă, care se va forma în conformitate cu următoarele:

1. M(i,j) = Pij, dacă există arcul (xi, xj) de pondere Pij;

2. M(i,j) = ∞, unde ∞ este un număr foarte mare/mic, dacă arcul (xi, xj) este lipsă;

3. M(i,j) = 0, dacă i = j.

La etapa a doua se va elabora un vector V0 în felul următor:

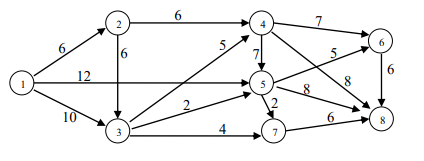
1. V0(i) = Pin, dacă există arcul (xi, xn), unde xn este vârful final pentru care se caută drumul minim, Pin este ponderea acestui arc;

2. V0(i) = ∞, dacă arcul (xi, xn) este lipsă;

3. V0(i) = 0, dacă i = j. Algoritmul constă în calcularea iterativă a vectorului V în conformitate cu următorul procedeu:

1. Vk(i) = min{Vk-1; Pij+Vk-1(j)}, unde i = 1, 2,…, n - 1, j = 1, 2,..., n; i<>j; 2. Vk(n) = 0. Când se va ajunge la Vk = Vk-1 - STOP. Componenta cu numărul i a vectorului Vk cu valoarea diferită de zero ne va da valoarea minimă a drumului care leagă vârful i cu vârful n.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | | X1 | 0 | 6 | 10 | ∞ | 12 | ∞ | ∞ | ∞ | | X2 | ∞ | 0 | 6 | 6 | ∞ | ∞ | ∞ | ∞ | | X3 | ∞ | ∞ | 0 | 5 | 2 | ∞ | 4 | ∞ | | X4 | ∞ | ∞ | ∞ | 0 | 7 | 7 | ∞ | 8 | | X5 | ∞ | ∞ | ∞ | ∞ | 0 | 5 | 2 | 8 | | X6 | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | ∞ | 6 | | X7 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 6 | | X8 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | | V0 | ∞ | ∞ | ∞ | 8 | 8 | 6 | 6 | 0 | | V1 | 20 | 14 | 10 | 8 | 8 | 6 | 6 | 0 | | V2 | 20 | 14 | 10 | 8 | 8 | 6 | 6 | 0 |   distmin(1,8)=20  1→2→4→8  1→3→5→7→8  1→3→5→8  1→3→7→8  1→5→7→8  1→5→8 | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | | X1 | 0 | 6 | 10 | -∞ | 12 | -∞ | -∞ | -∞ | | X2 | -∞ | 0 | 6 | 6 | -∞ | -∞ | -∞ | -∞ | | X3 | -∞ | -∞ | 0 | 5 | 2 | -∞ | 4 | -∞ | | X4 | -∞ | -∞ | -∞ | 0 | 7 | 7 | -∞ | 8 | | X5 | -∞ | -∞ | -∞ | -∞ | 0 | 5 | 2 | 8 | | X6 | -∞ | -∞ | -∞ | -∞ | -∞ | 0 | -∞ | 6 | | X7 | -∞ | -∞ | -∞ | -∞ | -∞ | -∞ | 0 | 6 | | X8 | -∞ | -∞ | -∞ | -∞ | -∞ | -∞ | -∞ | 0 | | V0 | -∞ | -∞ | -∞ | 8 | 8 | 6 | 6 | 0 | | V1 | 20 | 14 | 13 | 15 | 11 | 6 | 6 | 0 | | V2 | 23 | 21 | 20 | 18 | 11 | 6 | 6 | 0 | | V3 | 30 | 26 | 23 | 18 | 11 | 6 | 6 | 0 | | V4 | 33 | 29 | 23 | 18 | 11 | 6 | 6 | 0 | | V5 | 35 | 29 | 23 | 18 | 11 | 6 | 6 | 0 | | V6 | 35 | 29 | 23 | 18 | 11 | 6 | 6 | 0 |   distmax(1,8)=35  1→2→3→4→5→6→8 |
| distmin(1,8)=20  1→2→4 |  |



**Problema 3.** Determina-ți valoarea fluxului maximal în rețeaua de transport conform algoritmului Ford-Fulkersson.

A diagram of a number

Description automatically generated

fb = 20+9+11+23+1+3+3 = 70

*X = {0, 1, 2, 3, 4, 5, 6, 7}*

*A= {0, 3, 5}*

*X\A = {1, 2, 4, 6, 7}*

*(0,1); (0,2); (3,6); (5,4); (5,7);*

*20 + 23 + 1 + 3 + 23 = 70*

*Flux maxim = 70*

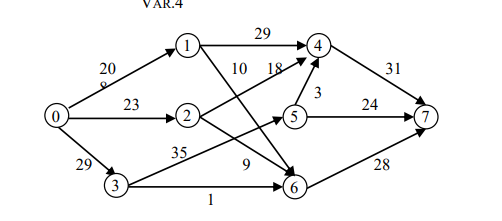
VII (0, 3, 5, 4, 1, 6, 7)

[+] +0 +3 +5 -4 +1 +6

e1 = minv+{29-24, 35-23, 3-0, 10-3, 28-13}=3

e2 = minv-{17}=17

e = min{3, 17}=3



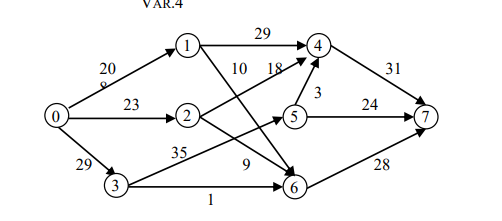
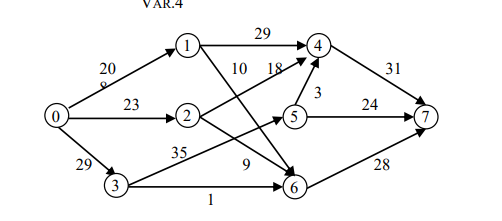
e = min{3, 20}=3

e2 = minv-{20}=20

e1 = minv+{23-20, 18-11, 10-0, 18-10}=3

[+] +0 +2 -4 +1 +6

VI (0, 2, 4, 1, 6, 7)



[+] +0 +3 +6

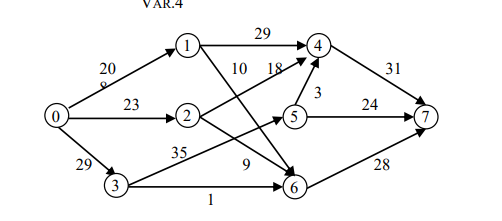
V (0, 3, 6, 7)

e1 = minv+{29-23, 1-0, 28-9}=1

IV (0, 3, 5, 7)

[+] +0 +3 +5

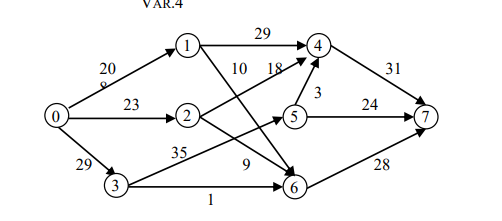
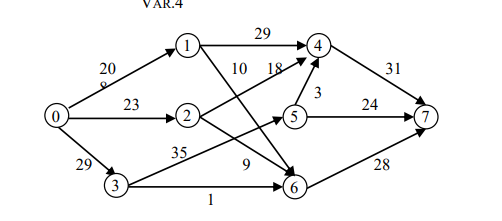
e1 = minv+{29-0, 35-0, 23-0}=23



e1 = minv+{23-9, 18-0, 31-20}=11

III (0, 2, 4, 7)

[+] +0 +2 +4



II (0, 2, 6, 7)

e1 = minv+{23-0, 9-0, 28-0}=9

[+] +0 +2 +6

[+] +0 +1 +4

I (0, 1, 4, 7)

e1 = minv+{20-0, 19-0, 31-0}=20

