Advanced Algorithms

Floriano Zini
Free University of Bozen-Bolzano
Faculty of Computer Science

Academic Year 2013-2014

Lab 8 – Solution of assignments
Assignment 07

Exercise 7.10 page 224 DPV

For the following network, with edge capacities as shown, find the maximum flow from S to T. Use both Linear programming and the Ford-Fulkerson algorithm.

![Network Diagram]

**Solution with LP**

**Exercise 7.10 page 224 DPV**

<table>
<thead>
<tr>
<th>Variables:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a=SA, b=SB, c=SC, d=AB, e=CB, f=AD, g=AE, h=BE, i=CF, j=DE, k=EF, l=DG, m=EG, n=FT, o=GT</td>
</tr>
</tbody>
</table>

**Objective function**

maximize \( a + b + c + d + e + f + g + h + i + j + k + l + m + n + o \)

**Constraints**

- Capacity and non negativity:
  - \( 0 <= a <= 6 \)
  - \( 0 <= b <= 1 \)
  - \( 0 <= c <= 10 \)
  - \( 0 <= d <= 2 \)
  - \( 0 <= e <= 2 \)
  - \( 0 <= f <= 4 \)
  - \( 0 <= g <= 1 \)
  - \( 0 <= h <= 20 \)
  - \( 0 <= i <= 2 \)
  - \( 0 <= j <= 6 \)
  - \( 0 <= k <= 5 \)
  - \( 0 <= m <= 10 \)
  - \( 0 <= n <= 4 \)
  - \( 0 <= o <= 12 \)

- Flow conservation:
  - \( a = d + f + g \)
  - \( b + d + e = h \)
  - \( c + e + i \)
  - \( f = j + l \)
  - \( g + h + j = k + m \)
  - \( i + k = n \)
  - \( i + m = o \)
Assignment 07

Exercise 7.10 page 224 DPV
Solution with LP (cont.)

The solution is on the right and is:

\( a = 6, b = 1, c = 6, d = 2, \\
\( e = 2, f = 3, g = 1, h = 5, \\
\( i = 4, j = 2, k = 0, l = 1, \\
\( m = 8, n = 4, o = 9 \)

To obtain the maximum flow, we consider either the variables corresponding to the out-links of the source node or those corresponding to the in-links of the sink node.

Max flow = \( a + b + c = n + o = 13 \)

Assignment 07

Exercise 7.10 page 224 DPV
Solution with Ford-Fulkerson

We use the following notation for the edges in \( G \):

\( a = SA, b = SB, c = SC, d = AB, e = CB, f = AD, g = AE, \\
\( h = BE, i = CF, j = DE, k = EF, l = DG, m = EG, n = FT, o = GT \)

We apply the FF algorithm to the graph \( G \).

1. Take augmenting path \( c \rightarrow i \rightarrow n \). The residual capacity \( c^* = \min\{c, i, n\} = 4 \). We allocate 4 flow units from this path. Update \( e := e - 4 \) for every \( e \) along the path \( c \rightarrow i \rightarrow n \).
2. Take augmenting path \( a \rightarrow f \rightarrow l \rightarrow o \). The residual capacity \( c^* = \min\{a, f, l, o\} = 4 \). We allocate 4 flow units from this path. Update \( e := e - 4 \) every \( e \) along the path \( a \rightarrow f \rightarrow l \rightarrow o \).
3. Take augmenting path \( c \rightarrow e \rightarrow h \rightarrow m \rightarrow o \). The residual capacity \( c^* = \min\{c, e, h, m, o\} = 2 \). We allocate 2 flow units from this path. Update \( e := e - 2 \) every \( e \) along the path \( c \rightarrow e \rightarrow h \rightarrow m \rightarrow o \).
4. Take augmenting path \( a \rightarrow d \rightarrow h \rightarrow m \rightarrow o \). The residual capacity \( c^* = \min\{a, d, h, m, o\} = 2 \). We allocate 2 flow units from this path. Update \( e := e - 2 \) every \( e \) along the path \( a \rightarrow d \rightarrow h \rightarrow m \rightarrow o \).
5. Take augmenting path \( b \rightarrow h \rightarrow m \rightarrow o \). The residual capacity \( c^* = \min\{b, h, m, o\} = 1 \). We allocate 1 flow units from this path. Update \( e := e - 1 \) every \( e \) along the path \( b \rightarrow h \rightarrow m \rightarrow o \).
6. Now we cannot find any other valid path from source to sink \( \rightarrow \) STOP!
7. Total flow = 4 + 4 + 2 + 2 + 1 = 13.
Assignment 07

Exercise 7.17 page 226 DPV

1. The maximum flow $f=11$ can be found solving the LP model corresponding to graph $G$ or running the Ford-Fulkerson algorithm on $G$.

2. The residual graph $G_f$ is

In this residual network, vertices A and B reachable from $S$, and C is the vertex from which $T$ is reachable (by reachable we mean that there is still some flow that can be further allocated for $S$ to $A$ and $B$ or from $C$ to $T$).

Assignment 07

Exercise 7.17 page 226 DPV

Solution (cont.)

3. The bottlenecks are AC and BC

4. Examples of networks without bottlenecks are:

5. On the residual graph, check every edge $e$ that has remaining flow $= 0$. Check whether there is a path in the graph from source to sink such that the path contains $e$ and $e$ is the only edge in the path that has flow $= 0$. 

All the edges have the same capacity