IE 680 – Special Topics in Production Systems: Networks, Routing and Logistics*

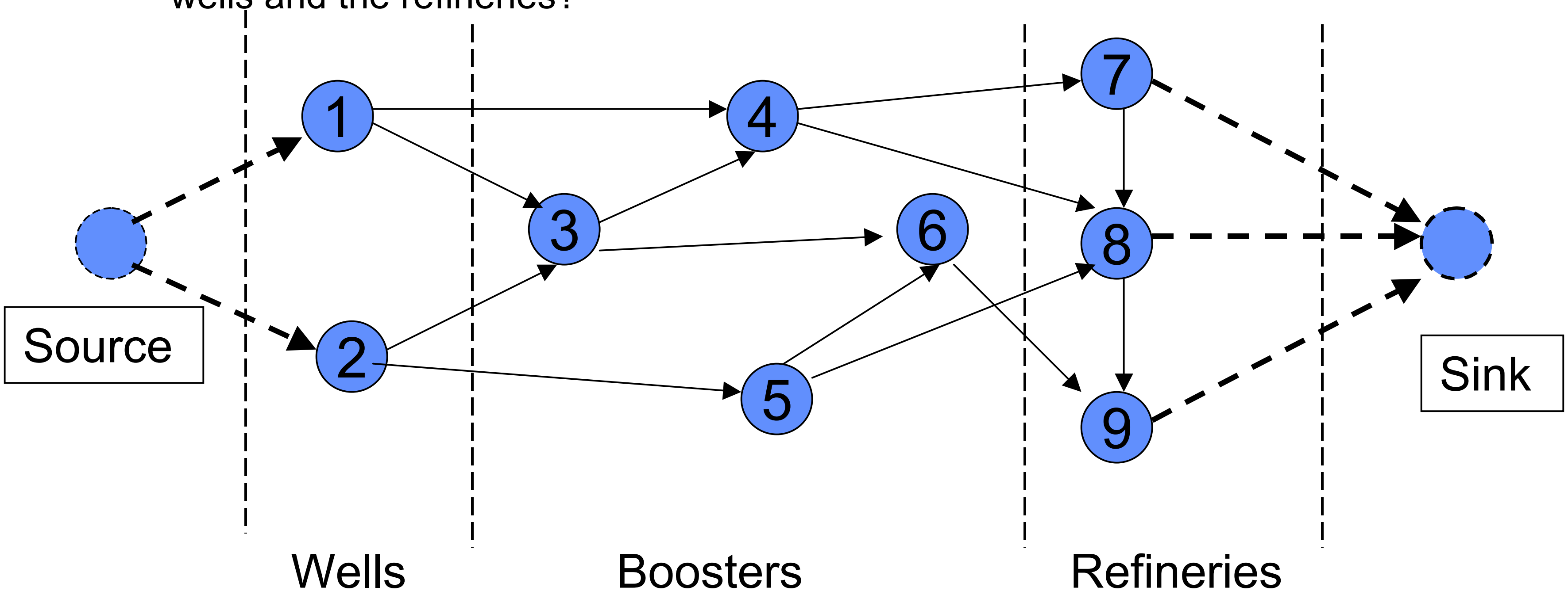
Rakesh Nagi
Department of Industrial Engineering
University at Buffalo (SUNY)

***Lecture notes from Network Flows by Ahuja, Magnanti and Wong (1993); Dr. Joy Bhadury**

Maximal Flow Problem

<u>Background of the problem:</u>

Consider a network of pipelines that transports crude oil from wells to refineries.Intermediate booster and pumping stations are installed at appropriate distances to move the crude in the network. Each pipeline has a finite maximum capacity. How can we determine the maximum capacity of the network between wells and the refineries?



Maximal Flow Problem

The Maximum Flow problem is stated as follows:

In a capacitated network, we wish to send as much flow as possible between two special nodes, a source node s and a sink node t, without exceeding the capacity of any arc

LP formulation of the Maximum flow problem:

Maximize υ

Subject to:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \begin{cases} \upsilon & \text{for } i = s \\ 0 & \text{for all } i \in N - \{S \text{ and } t\} \\ -\upsilon & \text{for } i = t \end{cases}$$

$0 =< x_{ij} <= u_{ij}$ for each $(i,j) \in A$

Maximal Flow Problem

- <u>Maximal Flow Algorithm:</u> (aka Labeling algorithm):
  - <u>Node label:</u>

$$(Pf , \pm i) = \begin{cases} Pf = \text{Potential} \pm \text{flow at node } i \\ i = \text{node} \end{cases}$$

  - <u>Arc Label:</u>

$$(f , CAP) = \begin{cases} f = \text{Actual flow through arc.} \\ CAP = \text{Capacity through arc} \end{cases}$$

The idea of the Maximal flow algorithm is to find a **<u>breakthrough path</u>** with net *positive* flow that links the source and the sink nodes.

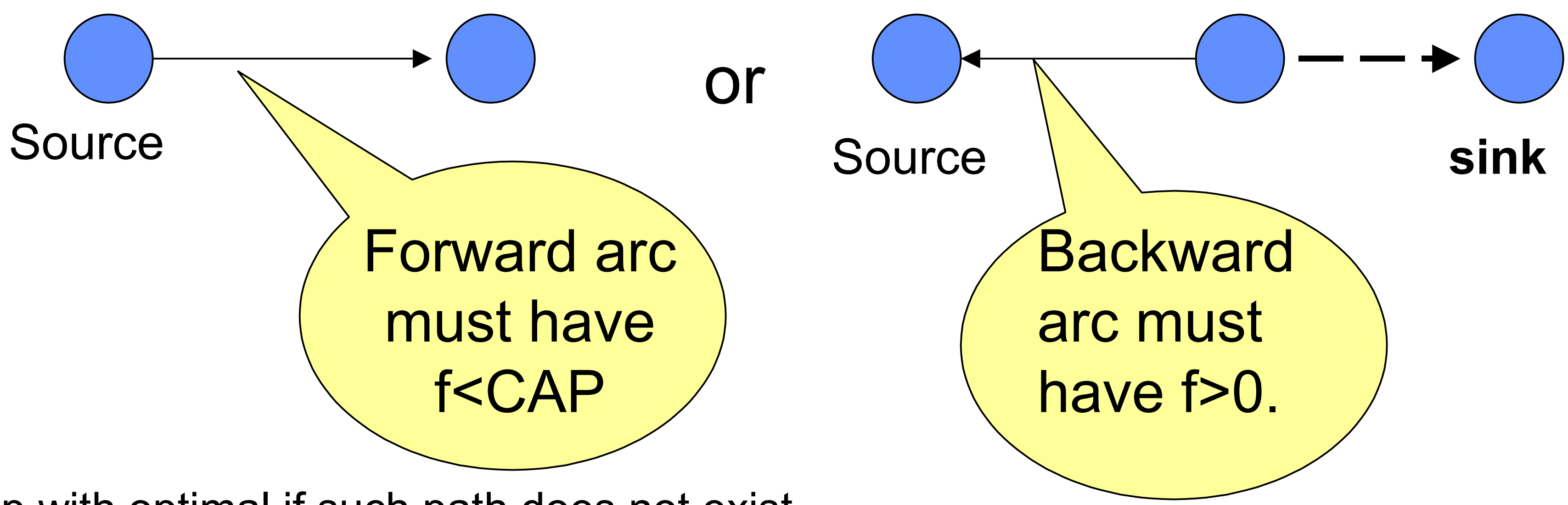
Maximal Flow Problem

- <u>Step 1:</u>

  Label source with (∞, -)

- <u>Step 2:</u>

  Find a path P, from the source to the sink having the following characteristics:



Stop with optimal if such path does not exist.

Maximal Flow Problem

- <u>Step 3:</u>

  Let (i,j) be an edge in P, then label j as follows:

  a) If (i,j) is forward in P , then label j as,

$$\left[\text{MIN}\left\{ \text{CAP-f of arc (i,j)} , \text{Pf at i} \right\} , +\text{i} \right]$$

  b) If (i,j) is backward in P , then label j as,

$$\left[\text{MIN}\left\{ \text{f of edge (i,j)} , \quad \text{Pf at i} \right\} , -\text{i} \right]$$

Maximal Flow Problem

- <u>Step 4:</u>

  Let f* be the Pf of the sink. Update edge (i,j) as follows:

  a) If label on j is [, + i] then label (i,j) becomes:

  [f + f* , CAP]

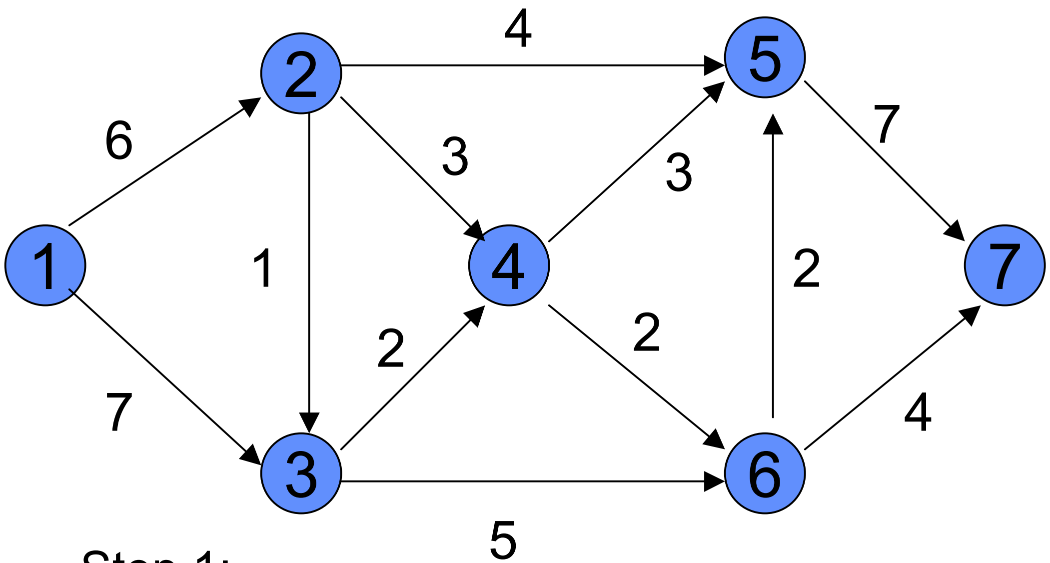
  b) If label on j is [, - i] then label (i,j) becomes:

  [f - f* , CAP]

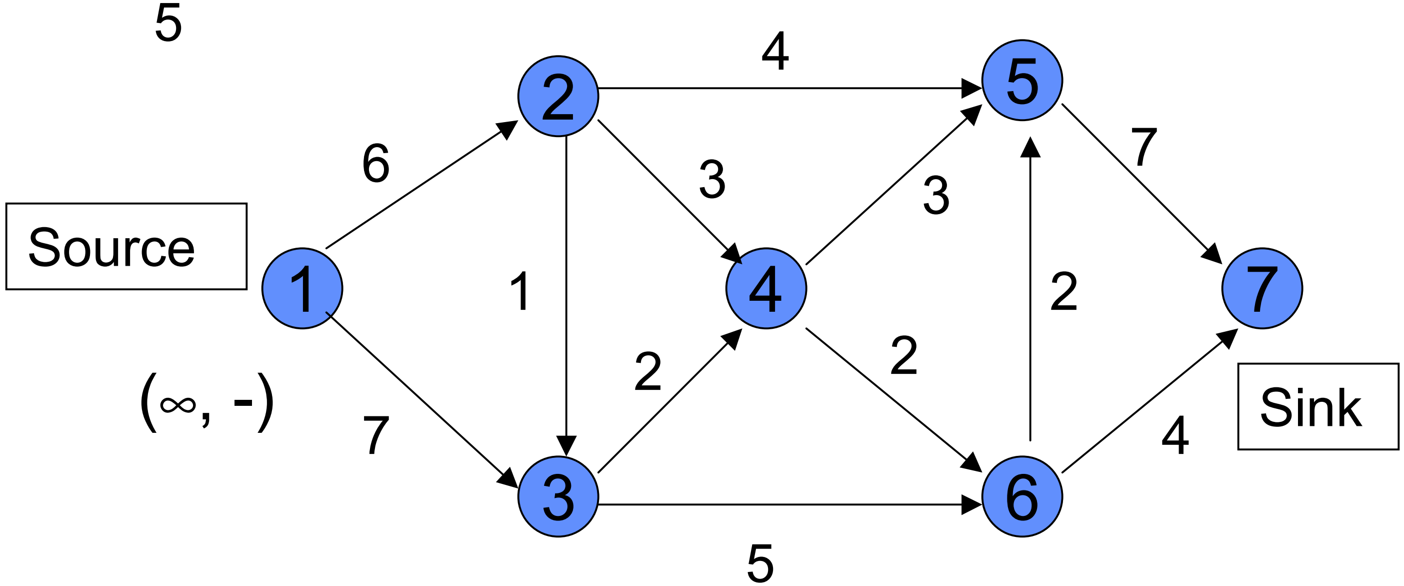
Go to Step 2.

Maximal Flow Problem

- Example: Find the maximal flow from node 1 to node 7.

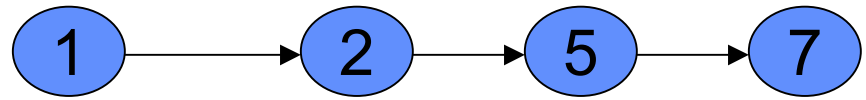


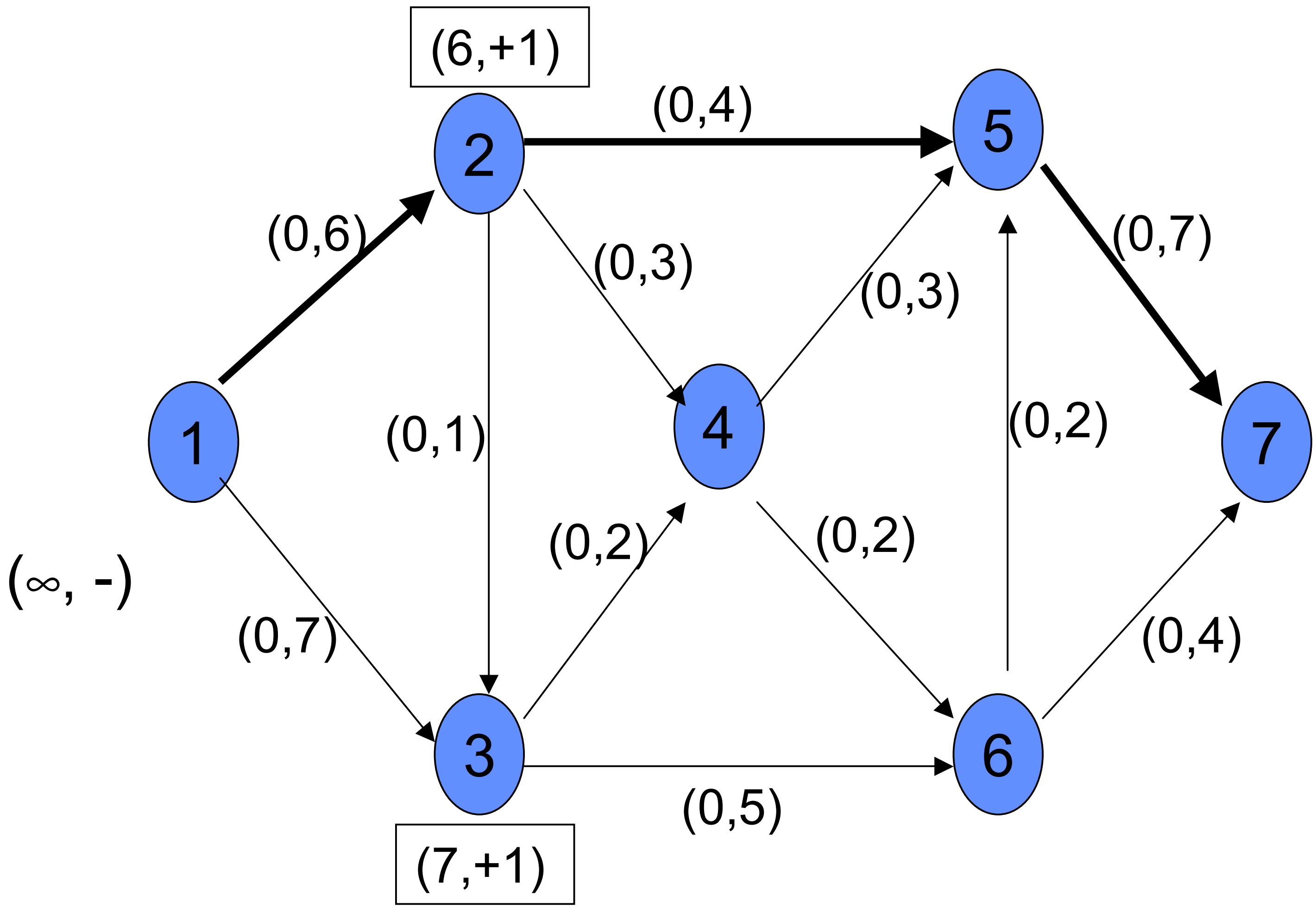
- Step 1:



Maximal Flow Problem

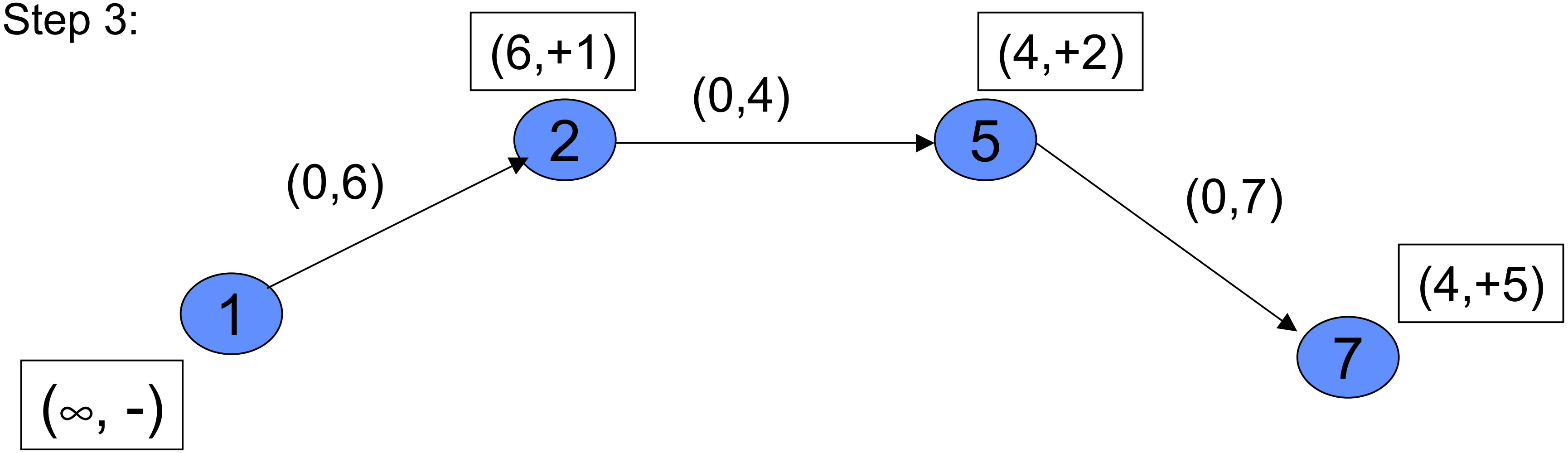
- Iteration 1:
- Step 2: Choose Path P





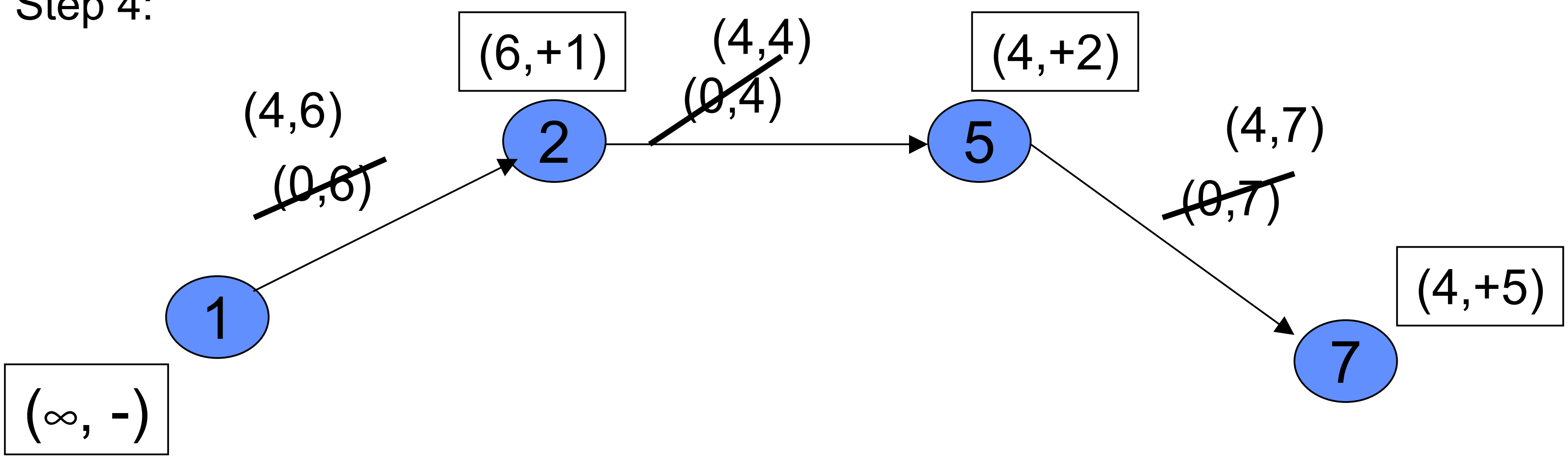
Maximal Flow Problem

- Step 3:



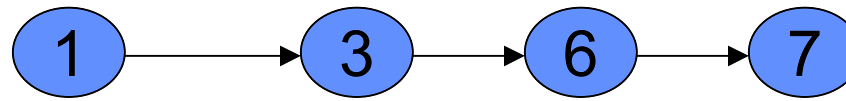
f * = 4; add +4 to each of the arc labels in the path.

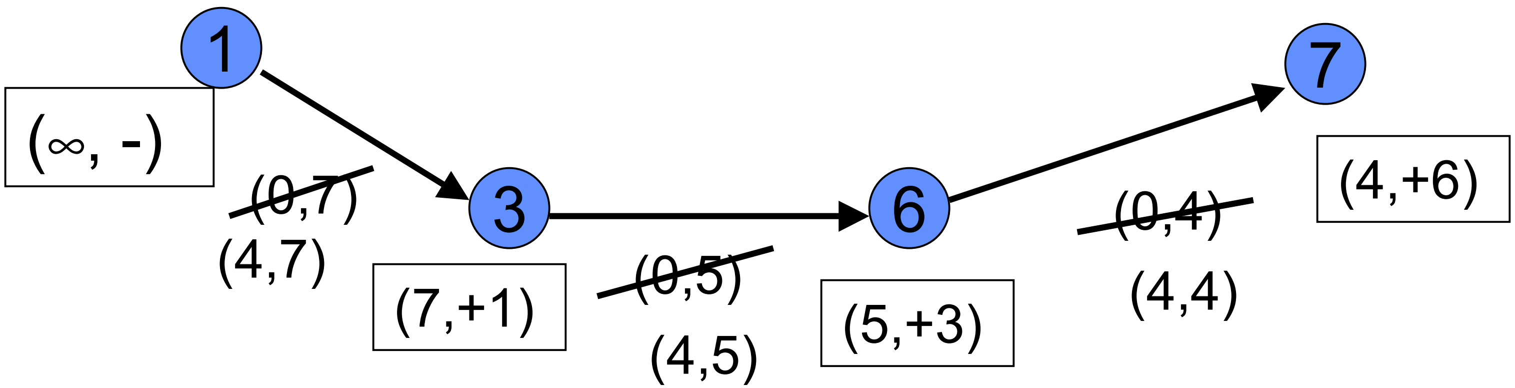
- Step 4:



Maximal Flow Problem

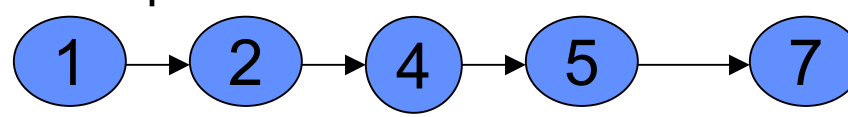
- <u>Iteration 2:</u> Step 2: Choose Path P

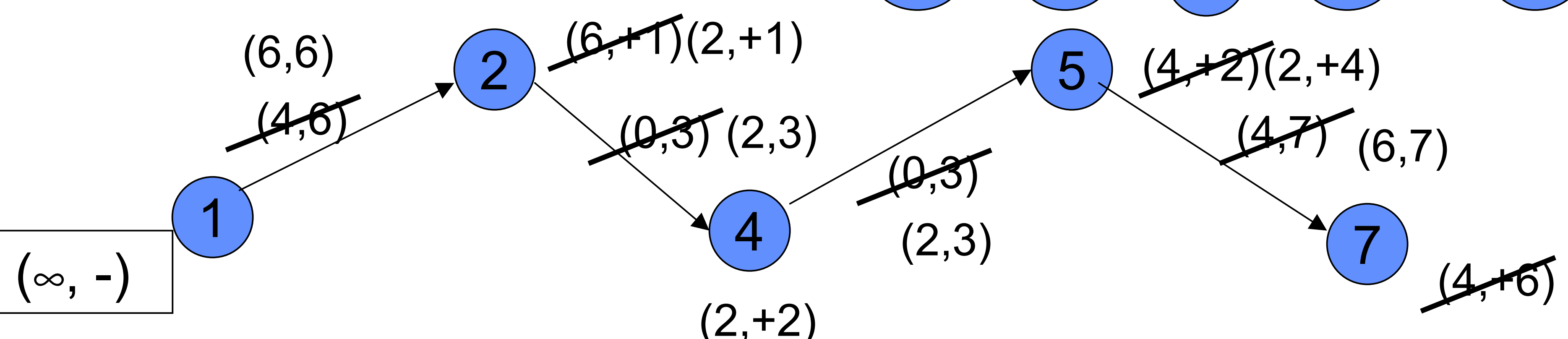




f * = 4; add +4 to each of the arc labels in the path.

- <u>Iteration 3:</u> Step 2: Choose Path P

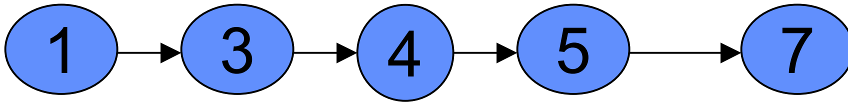


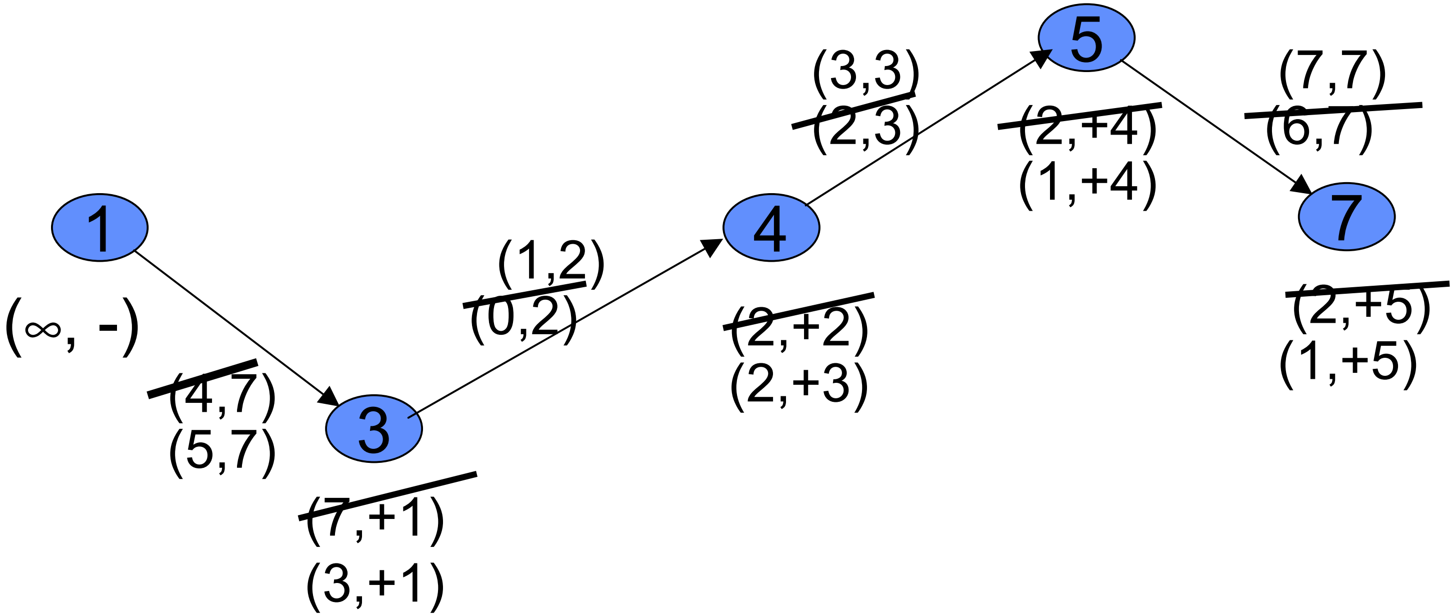


f * = 2; add +2 to each of the arc labels in the path.

Maximal Flow Problem

- <u>Iteration 4:</u> Step 2: Choose Path P



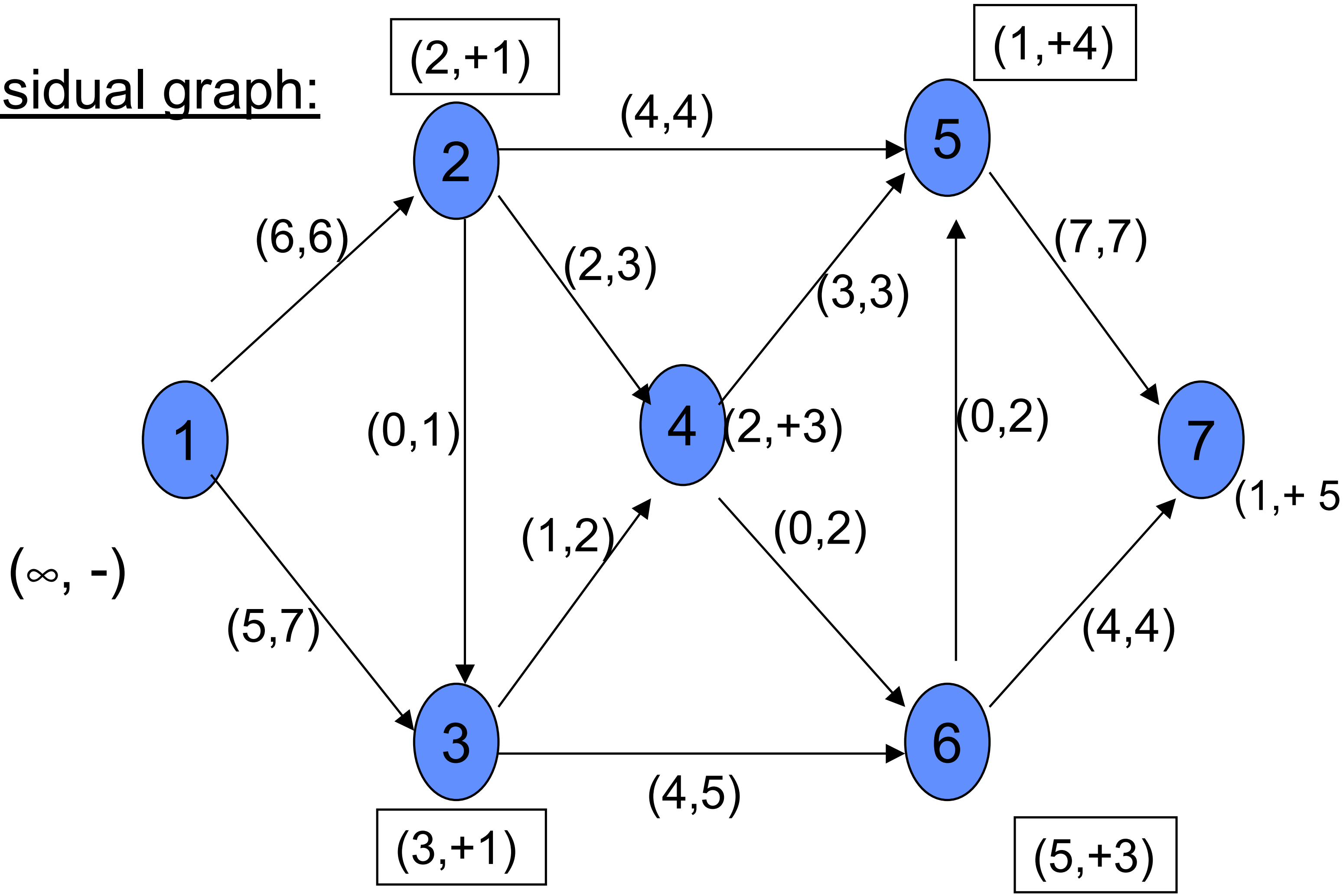


f * = 1; add +1 to each of the arc labels in the path.

Now, it is observed that there cannot be any breakthrough paths, since both the arcs leading to the sink are saturated, to their full capacity.Hence the iteration ends here, and finally we have:

Maximal Flow Problem

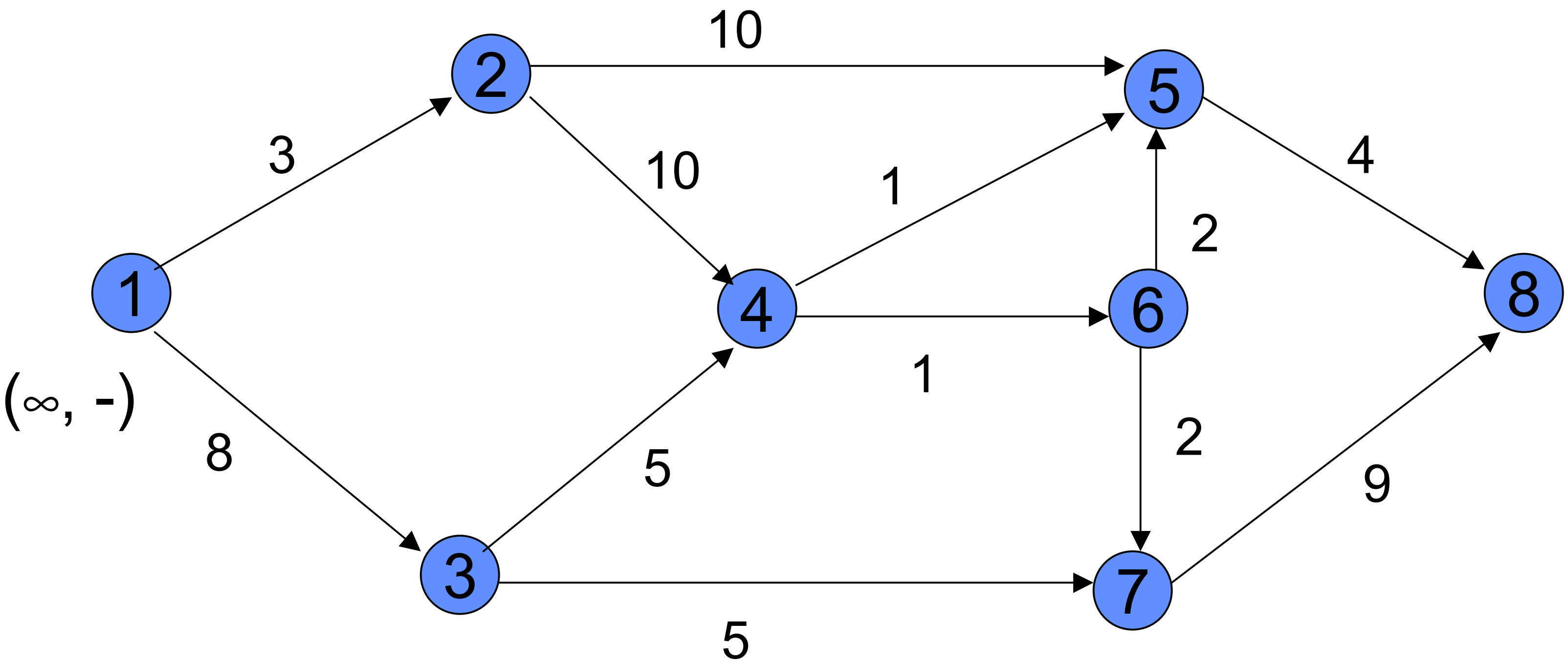
- Residual graph:



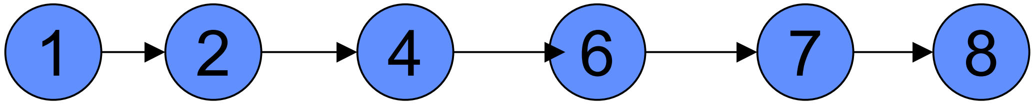
The maximal flow from node 1 to node 7 = 4+4+2+1= 11.

Maximal Flow Problem

- Example with an backward arc: Find the maximal flow from node 1 to node 8.

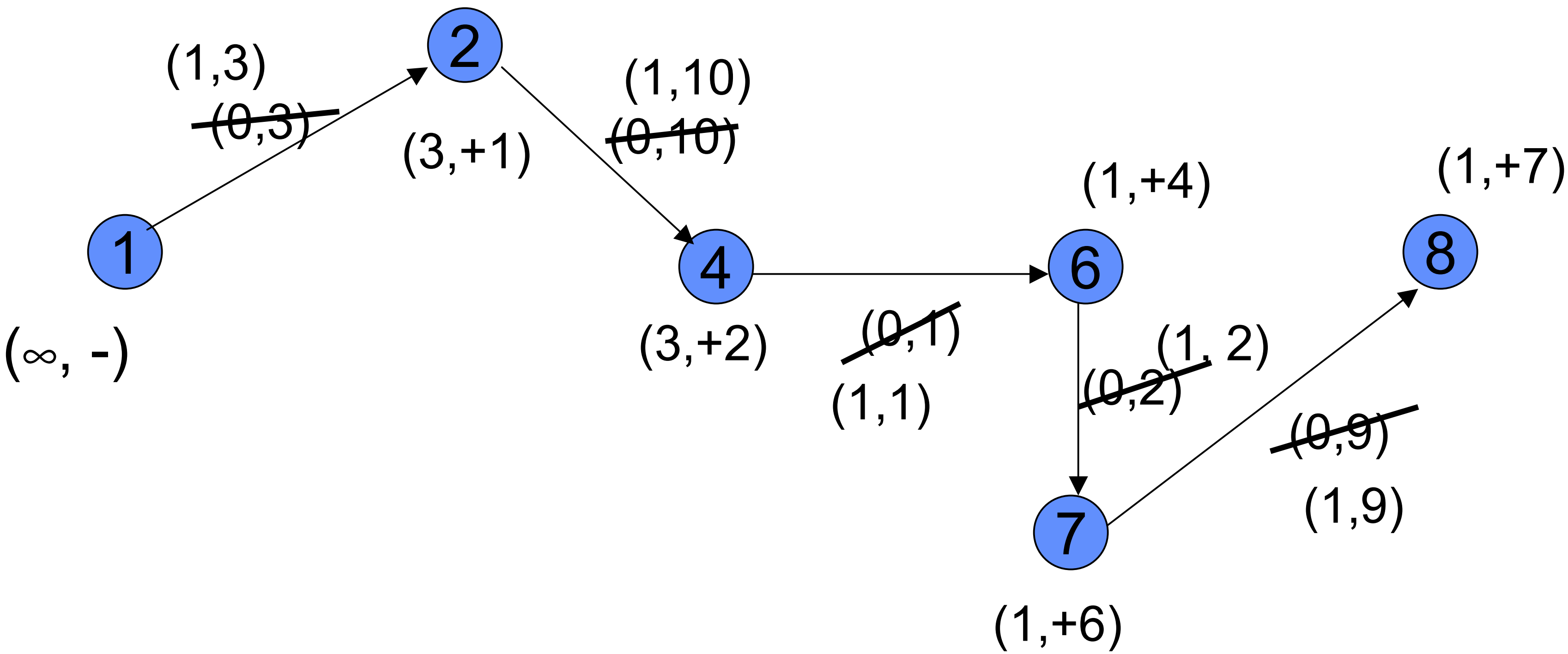


- Iteration 1:
- Step 2: Choose Path P



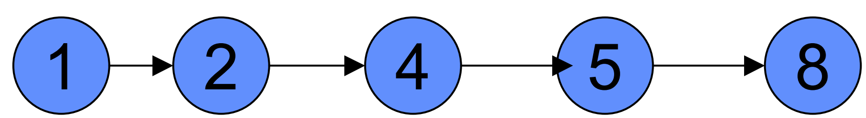
Maximal Flow Problem

*



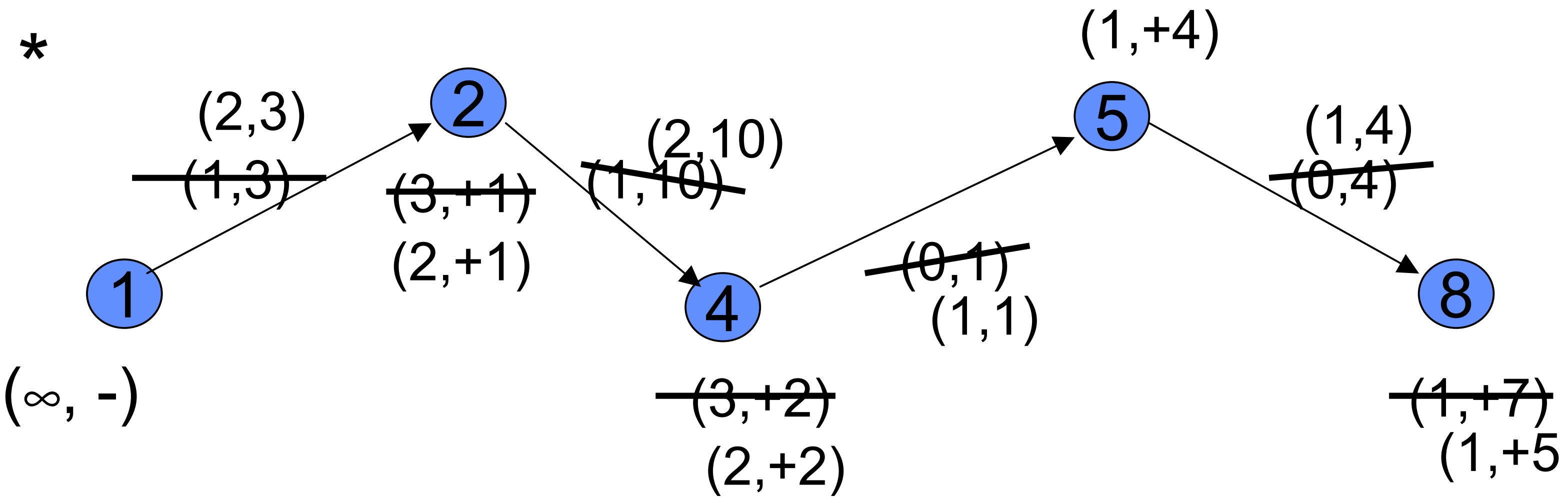
f * = 1; add +1 to each of the arc labels in the path.

- <u>Iteration 2:</u>
- Step 2: Choose Path P



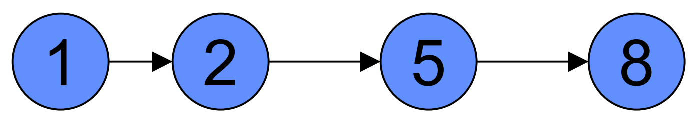
Maximal Flow Problem

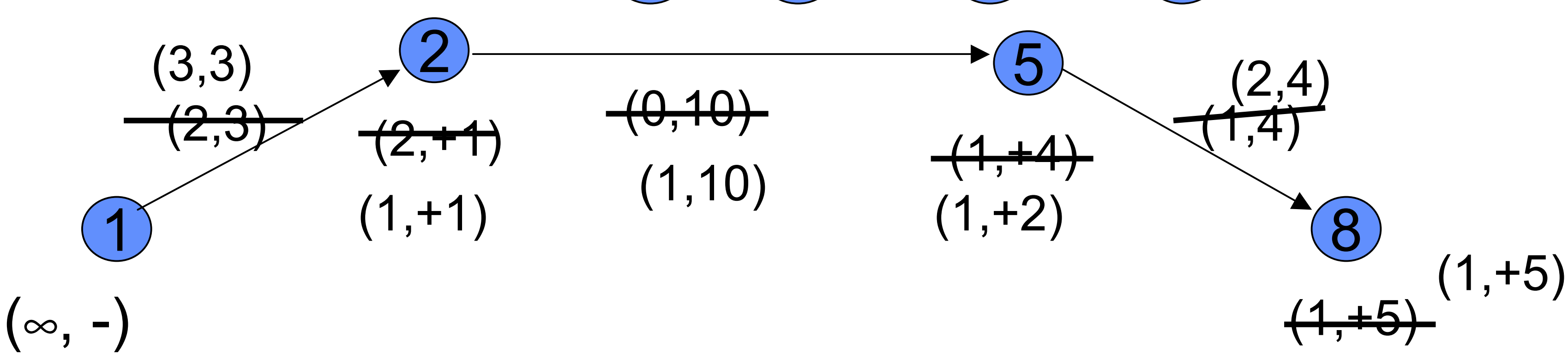
*



f * = 1; add +1 to each of the arc labels in the path.

- <u>Iteration 3:</u>
- Step 2: Choose Path P

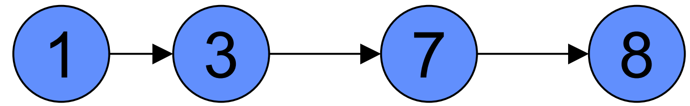


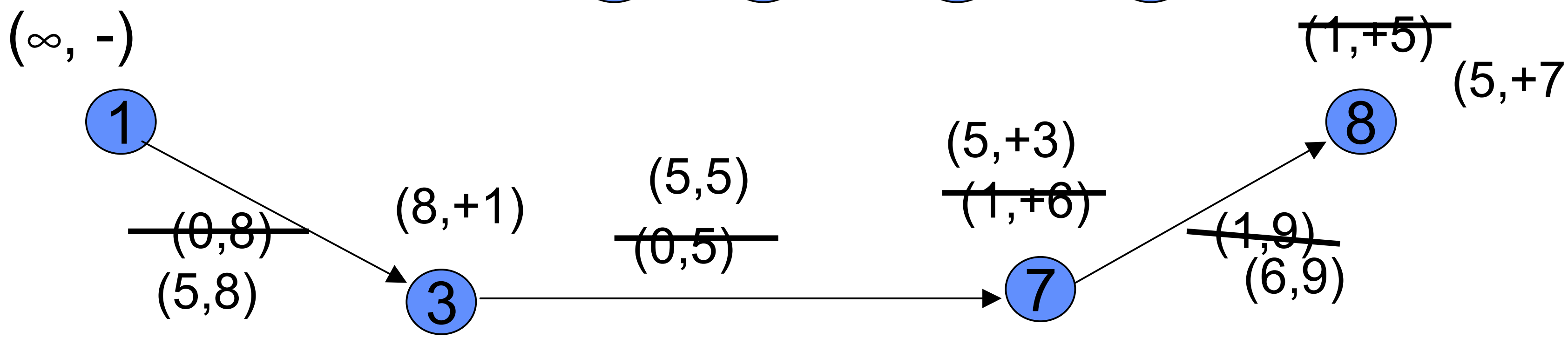


f * = 1; add +1 to each of the arc labels in the path.

Maximal Flow Problem

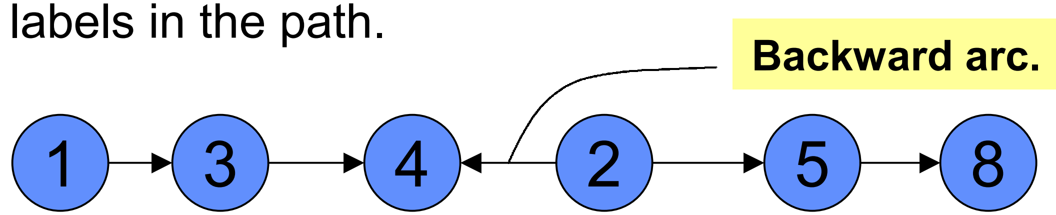
- Iteration 4:
- Step 2: Choose Path P

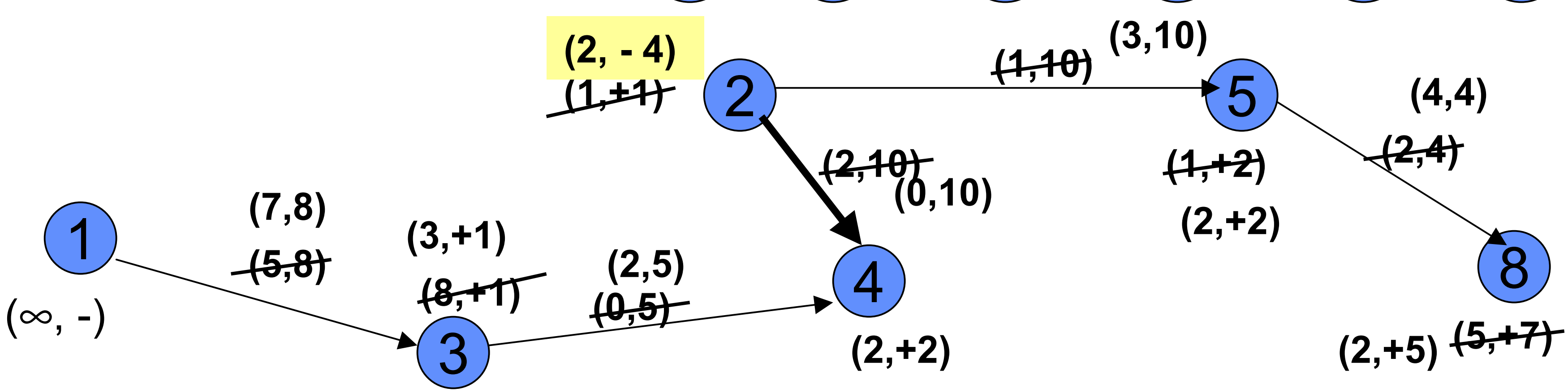




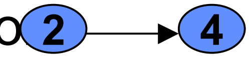
f * = 5; add +1 to each of the arc labels in the path.

- Iteration 5:
- Step 2: Choose Path P





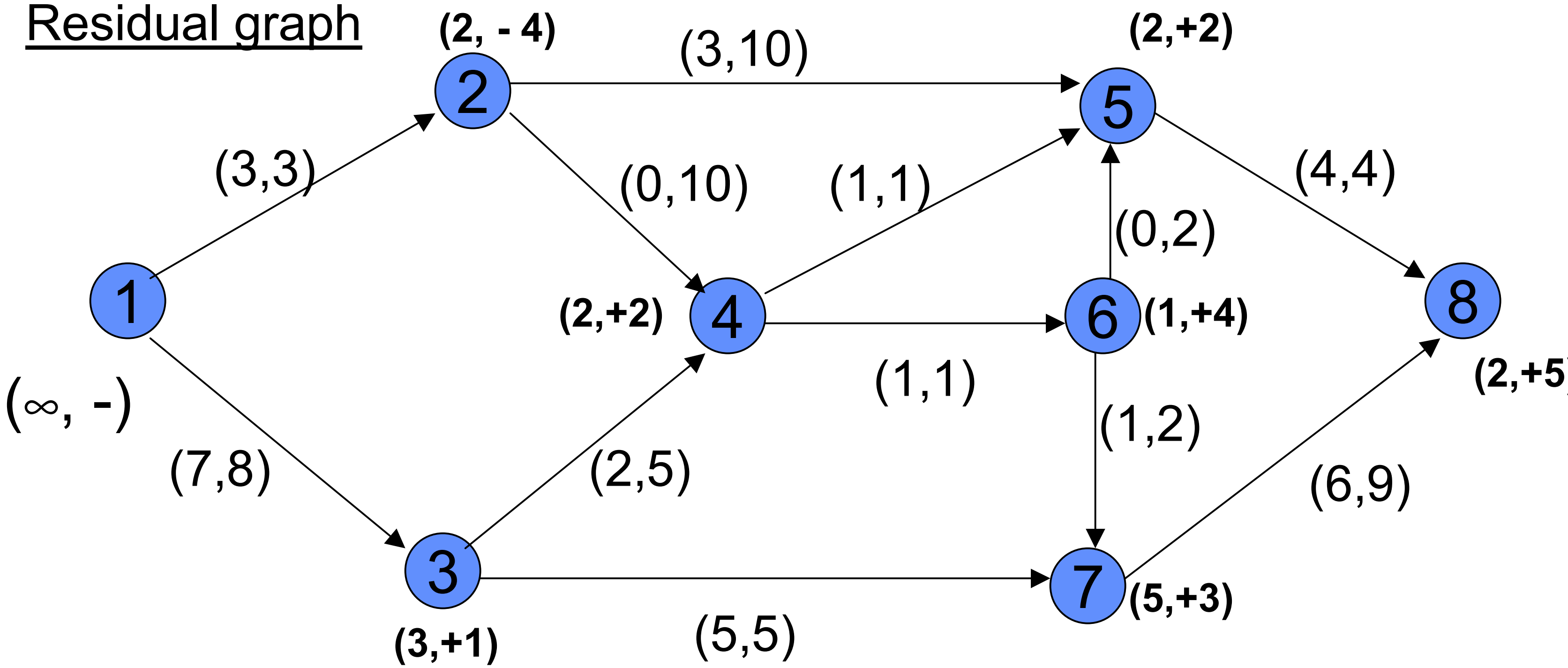
Maximal Flow Problem

f * = 2; add +2 to each of the arc labels in the path.(Except fo 

arc , since it is a backward arc, mark arc label as (f - f *) i.e. (2-2) = 0.

Now, it is observed that there cannot be any breakthrough paths, hence the iteration ends here, and finally we have:

- Residual graph



- The maximal flow from node 1 to node 8 = 1+1+1+5+2= 10.

Maximal Flow Problem

Recall the discussion on Cuts :

s-t Cut: is defined w.r.t. two distinguished nodes s and t and is a cut [S, Š] satisfying the property that $s \in S$ and $t \in Š$.

- In other words, a cut is a set of arcs which when deleted from the network, disconnects the source completely from the sink.

- Capacity of a cut: is defined as the sum of the capacities of the forward arcs in the cut.
- Minimum cut : We refer to an s-t cut whose capacity is minimum among all s-t cuts as a minimum cut.

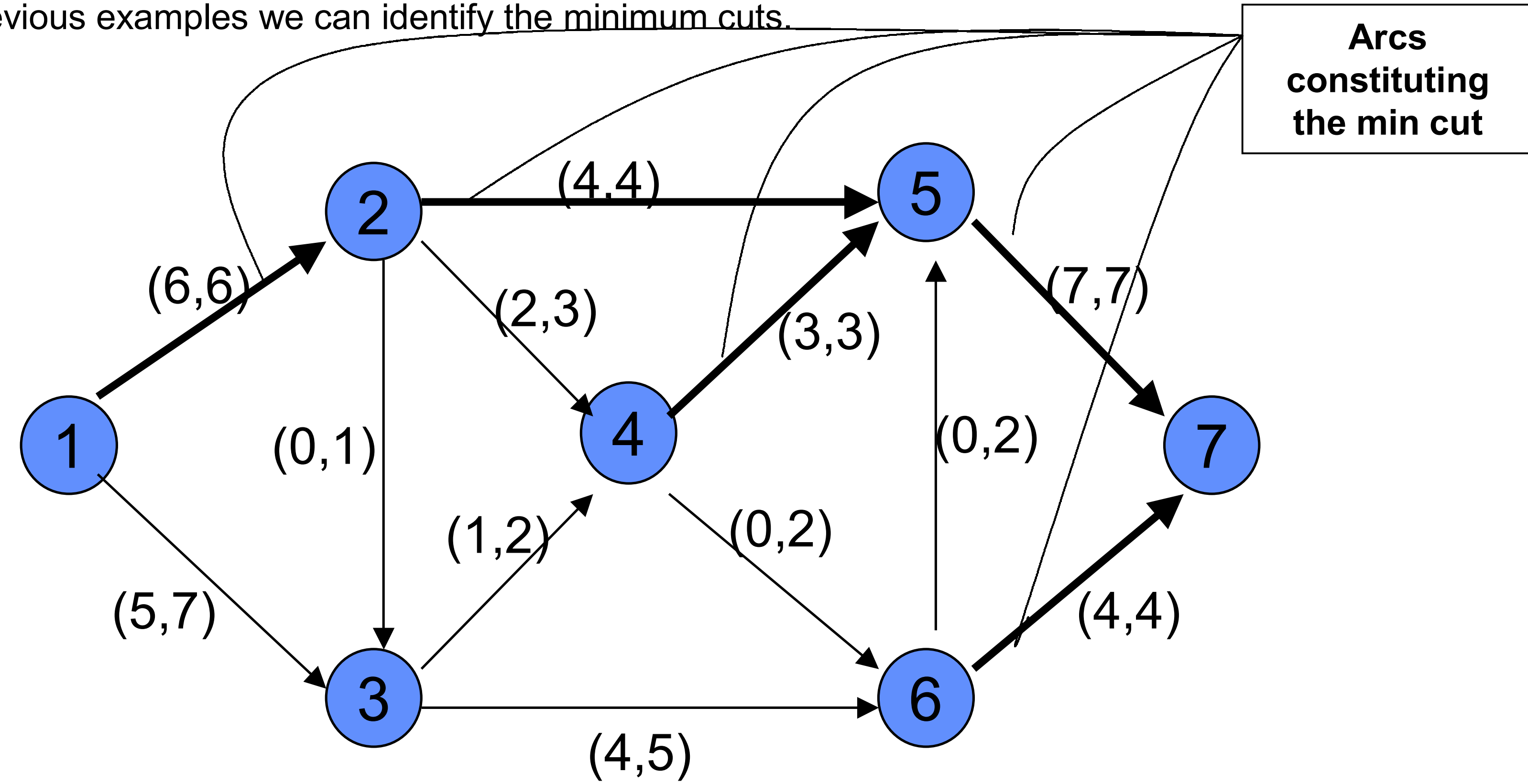
☛ Max-Flow Min-Cut Theorem:

The maximum value of the flow from a source node s to a sink node t in a capacitated network equals the minimum capacity among all the s-t cuts.

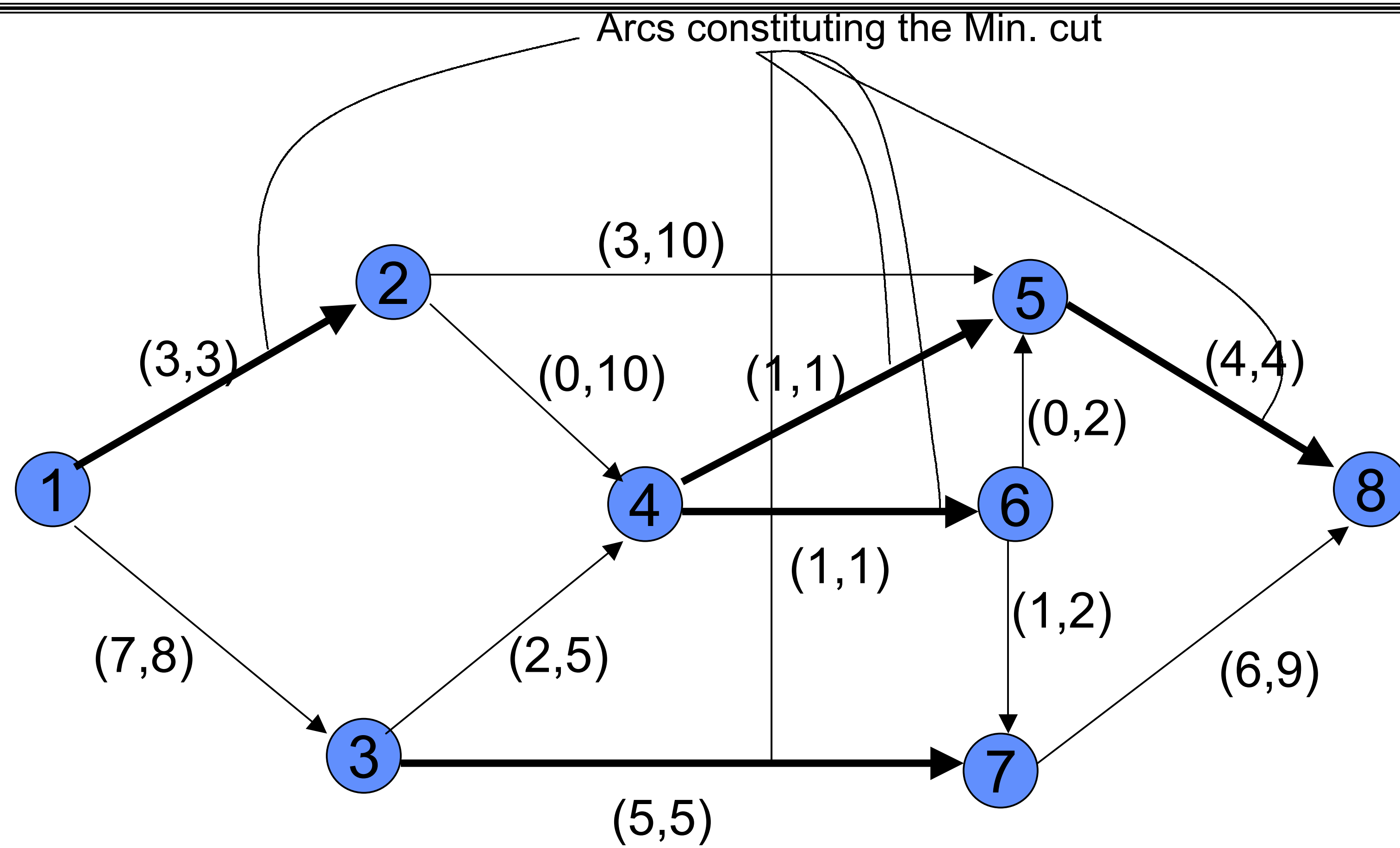
Maximal Flow Problem

The proof of the Max-Flow Min-Cut Theorem shows that when the labeling algorithm terminates,it has also discovered a minimum cut.

In the previous examples we can identify the minimum cuts.



Maximal Flow Problem



Maximal Flow Problem

- <u>Maximal flow in networks with lower bounds:</u>

  In some networks, the lower bounds may be strictly positive and we may be interested in finding the maximum and minimum flow in the network.

  <u>LP formulation of the Maximum flow problem with lower bounds:</u>

  Maximize υ

  Subject to:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:\ (j,i)\in A\}} x_{ji} = \begin{cases} \upsilon & \text{for } i = s \\ 0 & \text{for all } i \in N - \{S \text{ and } t\} \\ -\upsilon & \text{for } i = t \end{cases}$$

$$l_{ij} =< x_{ij} <= u_{ij} \text{ for each } (i,j) \in A$$

Maximal Flow Problem

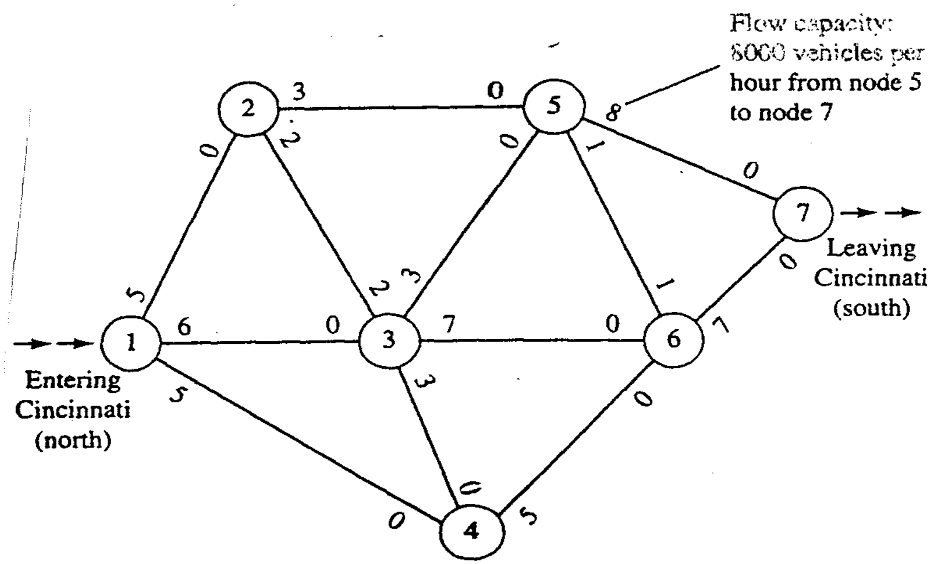
The maximum flow problem with zero lower bounds always has a feasible solution (since zero flow is feasible), the problem with non-negative lower bound could be infeasible.

Any Maximum flow algorithm for networks with non-negative lower bounds has two objectives:

- To determine whether the problem is feasible.
- If so, to establish a maximum flow.

Max-Flow Problem

Question: What Is The Maximum Traffic Flow Across The City ?



MAX-FLOW ALGORITHM

Step 1 Find any path from the source (input) node to the sink (output) node that has flow capacities in the direction of the flow greater than zero for all arcs on the path. If no path is available, the optimal solution has been reached.

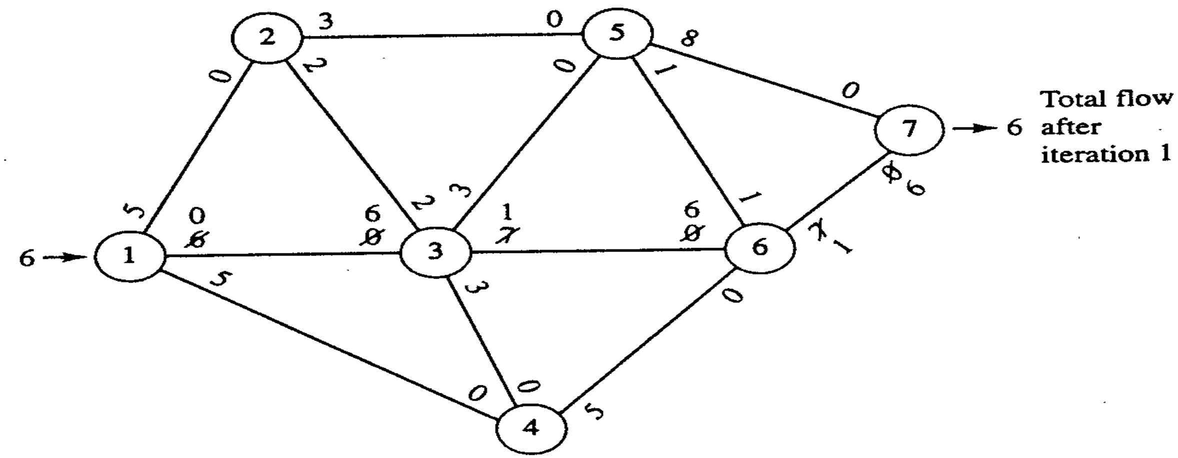
Step 2 Find the smallest arc capacity, P_f, on the path selected in step 1. Increase the flow through the network by sending an amount P_f over the path selected in step 1.

Step 3 For the path selected in step 1, reduce all arc flow capacities in the direction of flow by P_f, and increase all arc flow capacities in the reverse direction by P_f. Go to step 1.

Max Flow Problem - Numerical Example

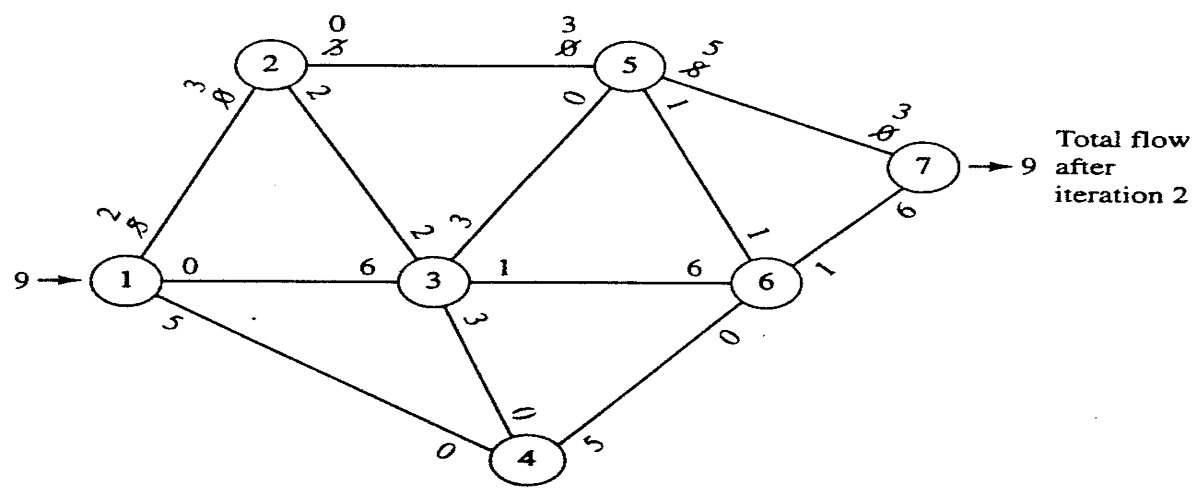
Iteration 1

The path selected is 1–3–6–7; P_f, determined by arc 1–3, is 6. The revised network is as follows:



Iteration 2

The path selected is 1–2–5–7; P_f, determined by arc 2–5, is 3. The revised network is as follows:



Max Flow Problem - Numerical Example

Iteration 3

The path selected is 1–2–3–5–7; P_f, determined by arc 1–2 (or 2–3), is 2.

Iteration 4

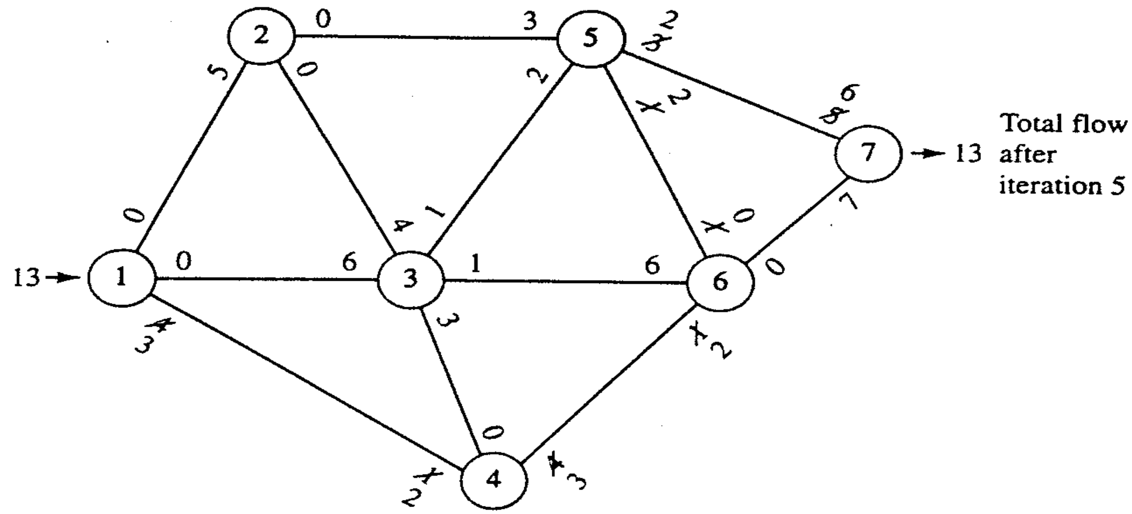
The path selected is 1–4–6–7; P_f, determined by arc 6–7, is 1.

Iteration 5

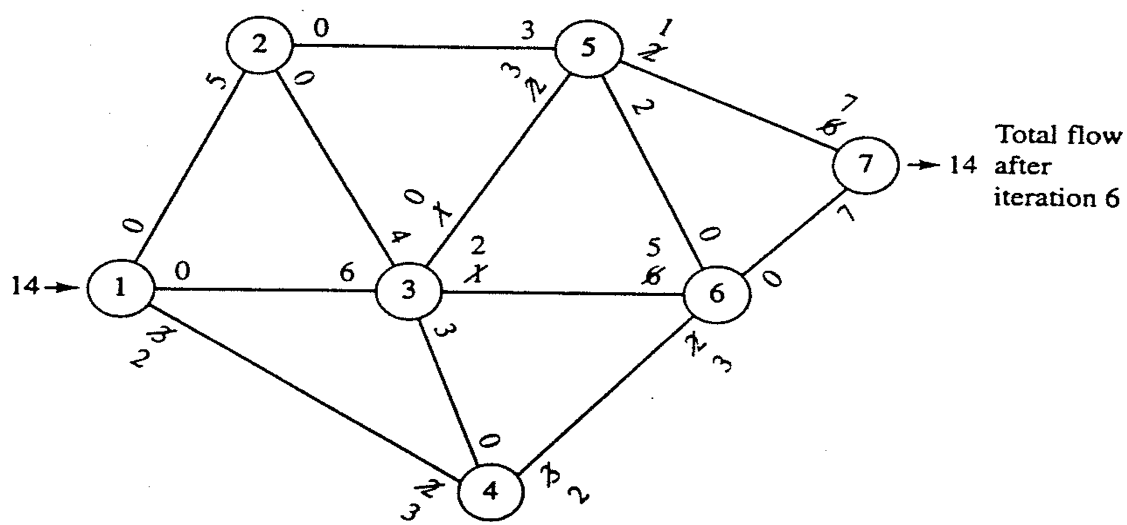
The path selected is 1–4–6–5–7; P_f, determined by arc 6–5, is 1.

Max Flow Problem - Numerical Example

AFTER ITN. 5



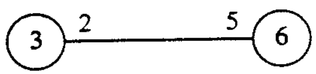
AFTER ITN. 6



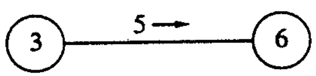
ADJUSTING FLOWS

Initial Capacities

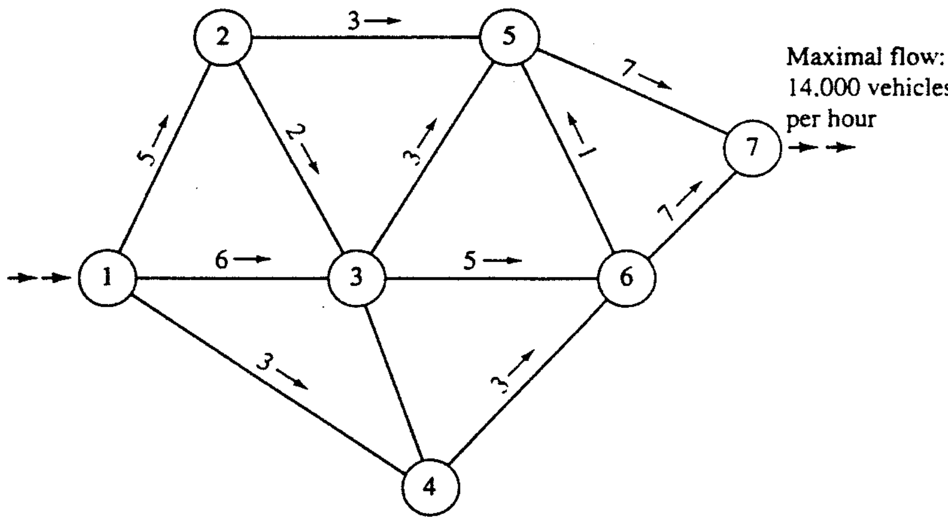
Final Capacities



Since the final flow capacity in the 3–6 direction is less than the initial flow capacity, the arc has a flow of 7 – 2 = 5 in the 3–6 direction. This arc flow is summarized as follows:



FINAL MAX-FLOW PATTERN



Application of Max-Flow Model to Assignment Problems

Dining Model

- *n* families go out to a dinner together to a restaurant. Assume that the number of members in the families is given by a(1), a(2), a(3), …a*(n)* members respectively.
- At the restaurant there are *m* tables, and that the seating capacity at the tables are given by c(1), c(2), c(3), …c*(m)* respectively.
- To increase their social interaction, they would like to sit at tables *so that no more than k members of the same family are seated at the same table*. Hence, if *k* = 1, there is atmost one member form the same family at a table.
- **Objective:** Is it possible to obtain a seating arrangement that satisfies this requirement ? If so, describe it.

In practical applications,

Families are employees with different backgrounds

Tables are projects.

And the objective is to assign people to projects such that each team is as "heterogeneous" as possible with regards to the backgrounds of its family members.

This problem can be formulated as a Max Flow Problem

Application of Max-Flow Model to Assignment Problems

PERGAMON

Omega 28 (2000) 143–153

omega
The International Journal of Management Science

www.elsevier.com/locate/orms

Maximizing workforce diversity in project teams: a network flow approach

Joyendu Bhadury[a,*], E. Joy Mighty[b], Hario Damar[b]

[a]*Department of Management and Finance, School of Business and Economics, California State University — Hayward, Hayward, CA 94542, USA*
[b]*Faculty of Administration, University of New Brunswick, Fredericton, NB, Canada E3B 5A3*

Received 1 May 1998; accepted 1 June 1999

Abstract

Several social, economic and political factors have contributed to the increasing diversity of today's workforce. In addition, in an era when organizations are continuously redesigning their work and restructuring their operations to achieve their goals with fewer resources, performing work in teams has become commonplace. These trends have increased the need for managing diverse work teams effectively. There are several existing models in the management science literature that help managers to assign employees to work groups in order to maximize the groups' diversity and hence, facilitate their effectiveness. This paper introduces a new model that recasts the problem of managing diversity in a different way: it is assumed that the population comes partitioned into 'families' with a high degree of intra-familial similarity and inter-familial dissimilarity. The objective of the assignment then is to disperse these family members as evenly into the workgroups as possible. A little known network flow problem, known as the *dining problem*, is used to develop an efficient algorithm to produce solutions to this new model. This is followed by a report on an experimental application of the developed model to assign Master of Business Administration students in a business school to different projects in a course. As a part of this empirical report, an attractive feature of this model is also demonstrated; namely, how to conduct sensitivity analysis to determine the optimal levels of diversity in the presence of resource constraints. Finally, the paper concludes by discussing limitations of this new model and how they may be addressed in future research on this topic.

Keywords: Decision making; Personnel/human resources; Manpower planning; Set partitioning; Optimization; Dining problem

1. Introduction

Among the many environmental trends affecting organizations in the 1990s is the rapidly changing composition of the workforce, a phenomenon known as *workforce diversity*. The word 'diversity' refers to differences in a range of human qualities among individuals [17]. The traditional view of an organization characterized by workforce diversity is one in which there are increasing numbers of nondominant or minority social groups based on gender, race, ethnicity or nationality, resulting in heterogeneity in socio-cultural perspectives, world views, life styles, language and behavior [17]. However, recent approaches have also attempted to extend the concept of diversity to include

* Corresponding author. Tel.: +1-510-885-4176; fax: +1-510-885-2165.
E-mail address: jbhadury@csuhayward.edu (J. Bhadury).

0305-0483/00/$ - see front matter
PII: S0305-0483(99)00037-7

Application of Max-Flow Model to Assignment Problems

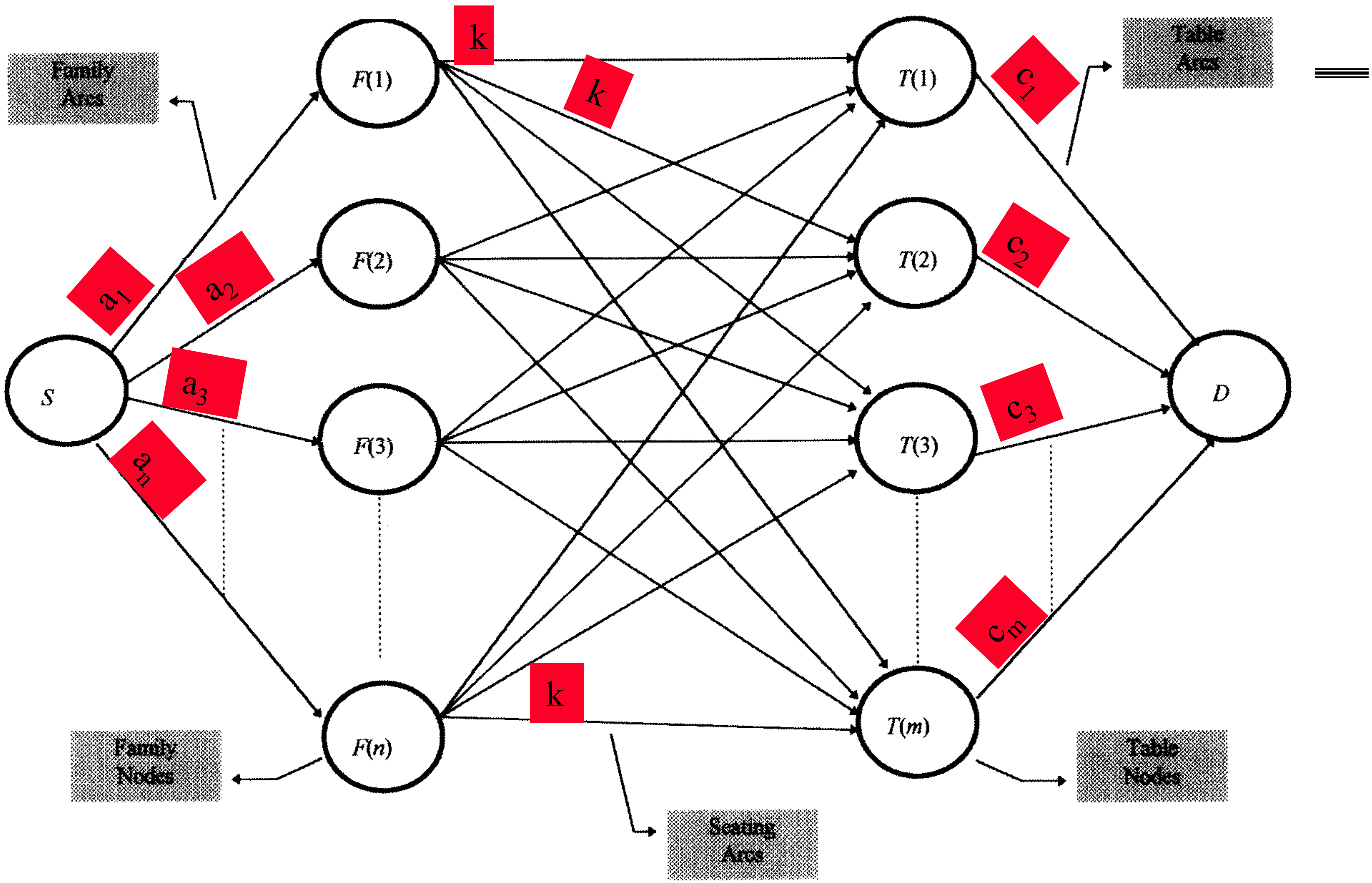


Fig. 1. The graph G.

B

Application of Max-Flow Model to Assignment Problems

•In the graph, the nodes are as follows:

1. The source node S and the destinations/sink node D
2. n Family nodes F(1), F(2), …F(n) where each node F(i) represents family i.
4. M Table nodes T(1). T(2), T(3), …, T(m), where each node T(j) represents table j.

•In the graph, the arcs are as follows:

1. The family arcs go from source node S to each of the family nodes F(i). The capacity of a family arc is assumed to be equal to the number of members in that family. Hence, the capacity of arc (S, F(1)) is a(1).
2. The table arcs go from each table node T(j) to the destination node D. The capacity of a table arc is assumed to be equal to the seating capacity of that table. Hence, the capacity of the table arc (T(1), D) is c(1).
3. The seating arcs that go form each family node F(i) to each table node T(j). The capacity of each of these arcs is set at k, which is the maximum number of members form each family that are permitted to be at the same table.

•As one can see, a flow of x units on the seating arc (F(i), T(j)) indicates that x members form family i should be seated at table j.

•Now solve the Max.-Flow problem form S to D. If the max flow is equal to (a(1) + a(2) + a(3) + …+ a(n)), then we are satisfied that all family members can be seated with the above constraint.

•Else, relax the parameters (increase value of k or increase the number of tables, or increase capacity of the tables etc.) and solve the problem again.