Network Simplex Method

Fatme Elmoukaddem

Jignesh Patel

Martin Porcelli

Outline

- Definitions
- Economic Interpretation
- Algebraic Explanation
- Initialization
- Termination

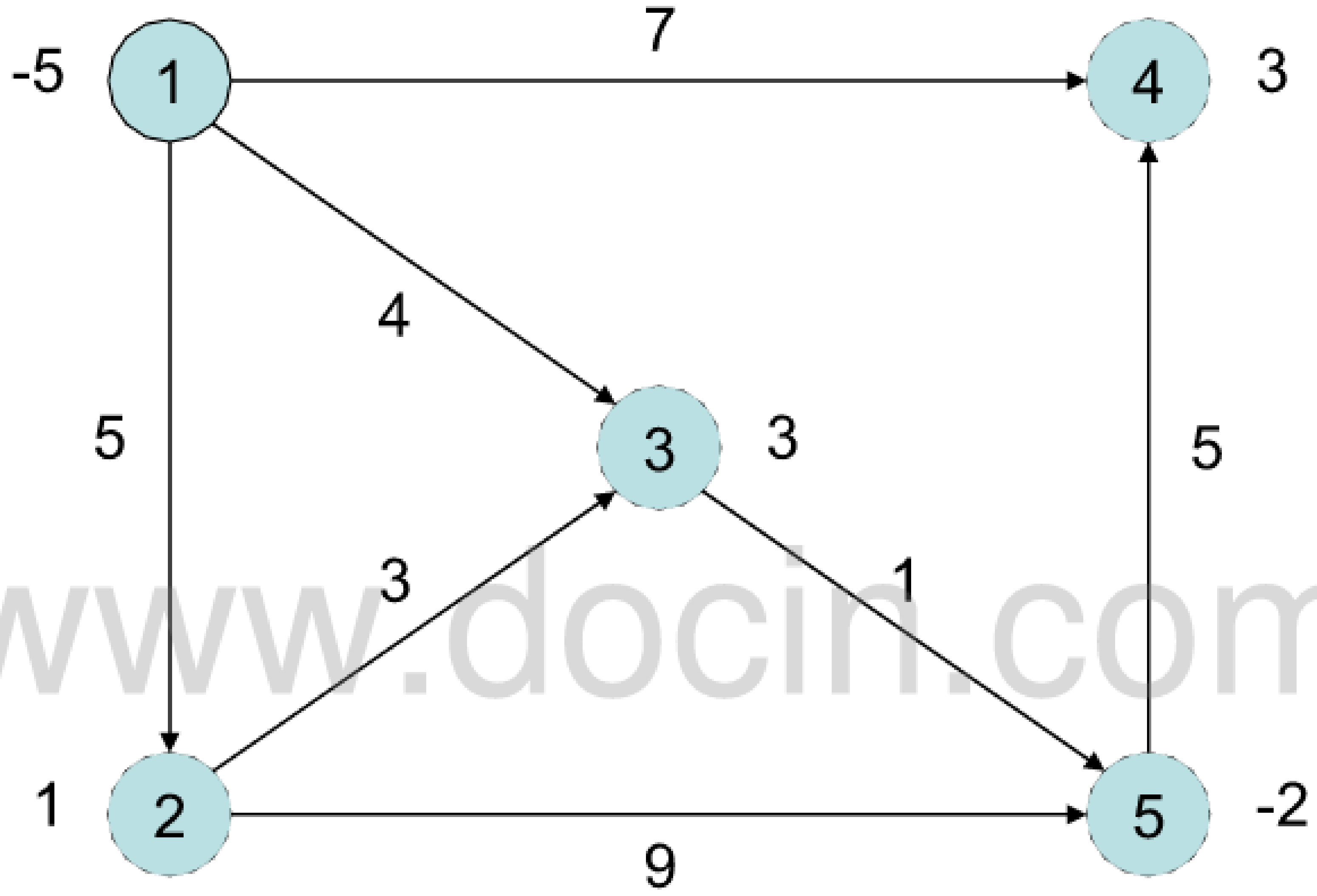
Transshipment Problem

- Find the cheapest way to ship prescribed amounts of a commodity from specified origins to specified destinations through a transportation network

Network

- A *network* is a collection of *nodes* connected by *arcs*
- Each node has a *demand* for the commodity
  - Nodes that are sources of the commodity have a negative demand
  - The sum of all the demands is zero
- Each arc has a *cost* to ship a unit of commodity over it

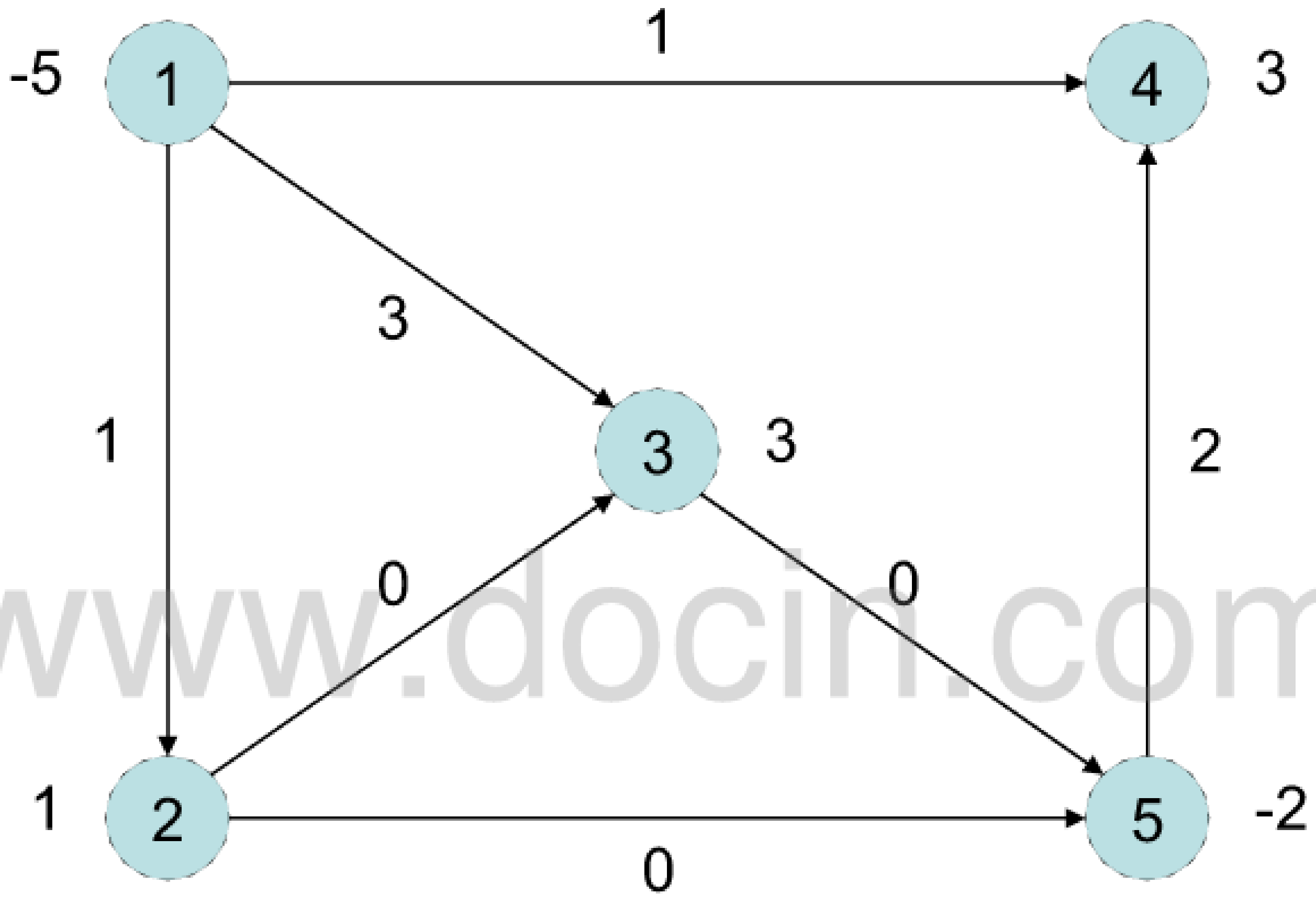
Example



Schedule

- A *schedule* describes how much of the commodity is shipped over each arc
- Requirements
  - The amount entering a node minus the amount leaving it is equal to its demand
  - The amount shipped over any arc is nonnegative

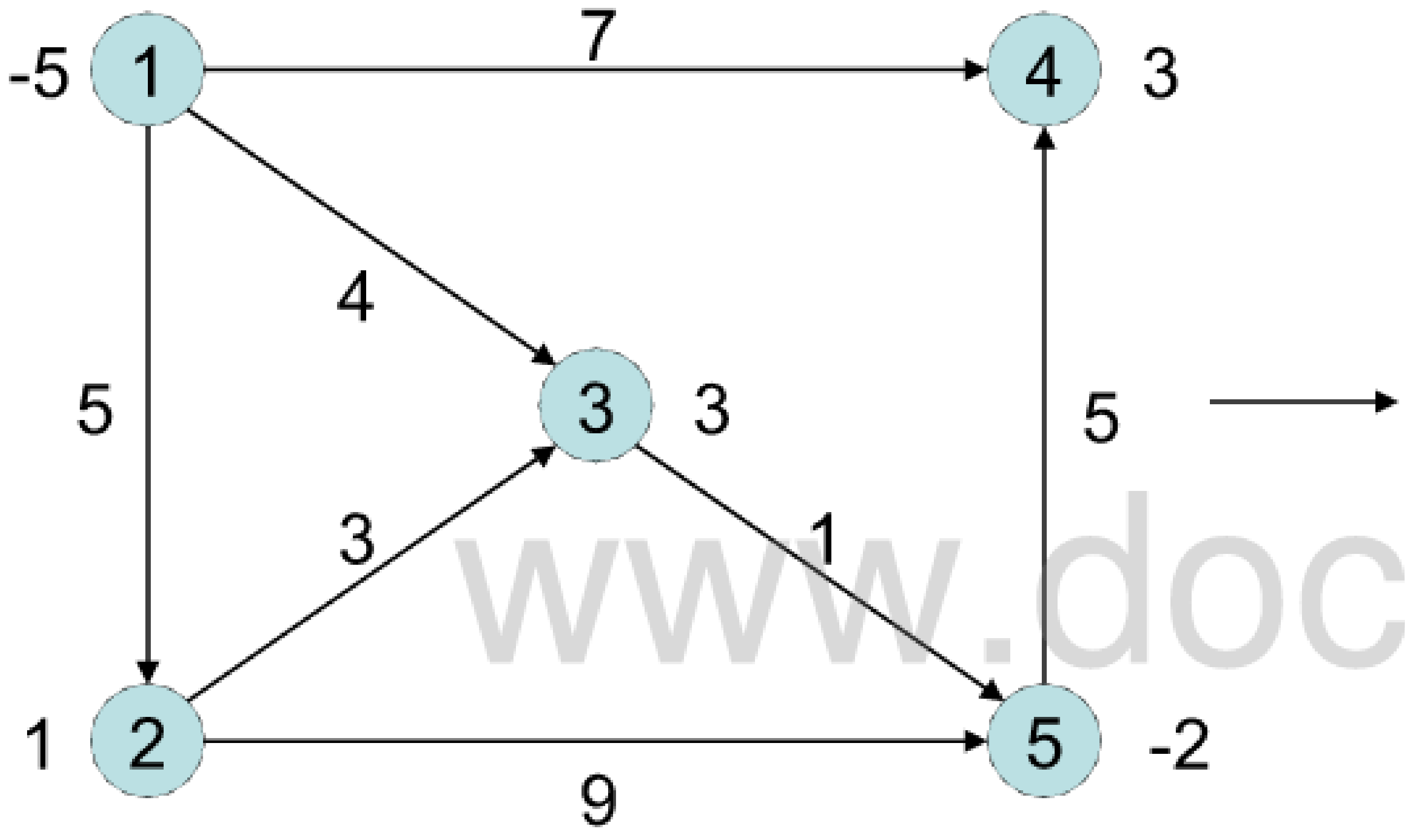
Example



LP Formulation

- Let **c** be a row vector and **x** a column vector indexed by the set of arcs
  - c_{ij} is the cost of shipping over *ij*
  - x_{ij} is the amount to ship over *ij*
- Let **b** be a column vector indexed by the set of nodes
  - b_i is the demand at *i*

Example



$$\mathbf{c} = \begin{bmatrix} 5 & 4 & 7 & 3 & 9 & 1 & 5 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{25} \\ x_{35} \\ x_{54} \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -5 & 1 & 3 & 3 & -2 \end{bmatrix}$$

LP Formulation

minimize $\mathbf{cx} = \sum_{ij} c_{ij} x_{ij}$ subject to

(ij) $x_{ij} \geq 0$

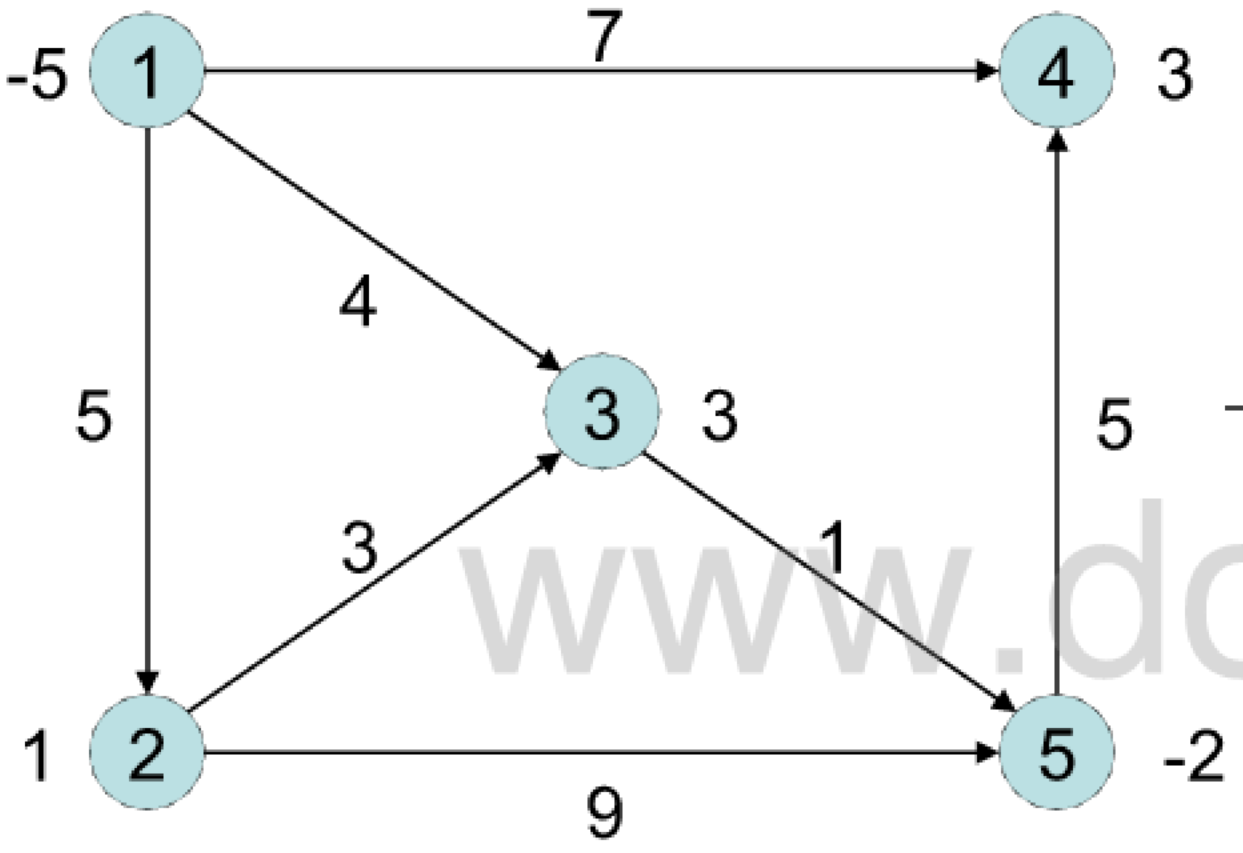
(i) $\sum_{ji} x_{ji} - \sum_{ij} x_{ij} = b_i$

$$\sum_{i} b_i = 0$$

LP Formulation (2)

- Let **A** be the matrix indexed by the set of nodes *x* the set of arcs
  - $A_{i,jk}$ is either
    - -1 if $i=j$
    - 1 if $i=k$
    - 0 otherwise
- **A** is known as the *incidence matrix* of the network

Example



$\longrightarrow$

$$
\begin{array}{c|ccccccc}
 & 12 & 13 & 14 & 23 & 25 & 35 & 54 \\
\hline
1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\
3 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\
4 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
5 & 0 & 0 & 0 & 0 & 1 & 1 & -1
\end{array}
$$

LP Formulation (2)

minimize $\mathbf{cx}$ subject to

$$\forall (ij) \quad x_{ij} \geq 0$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\sum_i b_i = 0$$

Tree Solution

- A *spanning tree* of a network is a network containing every node and enough arcs such that the undirected graph it induces is a tree
- A *feasible tree solution* **x** associated with a spanning tree T is a feasible solution with
  - $x_{ij} = 0$ if *ij* is not an arc of T

Network Simplex Method

- Search through feasible tree solutions to find the optimal solution
- Has a nice economic interpretation

Economic Interpretation

- Given a spanning tree T and an associated feasible tree solution **x**
- Imagine you are the only company that produces the commodity
- What price should you sell the commodity for at each node?
  - Assume that you ship according to **x**

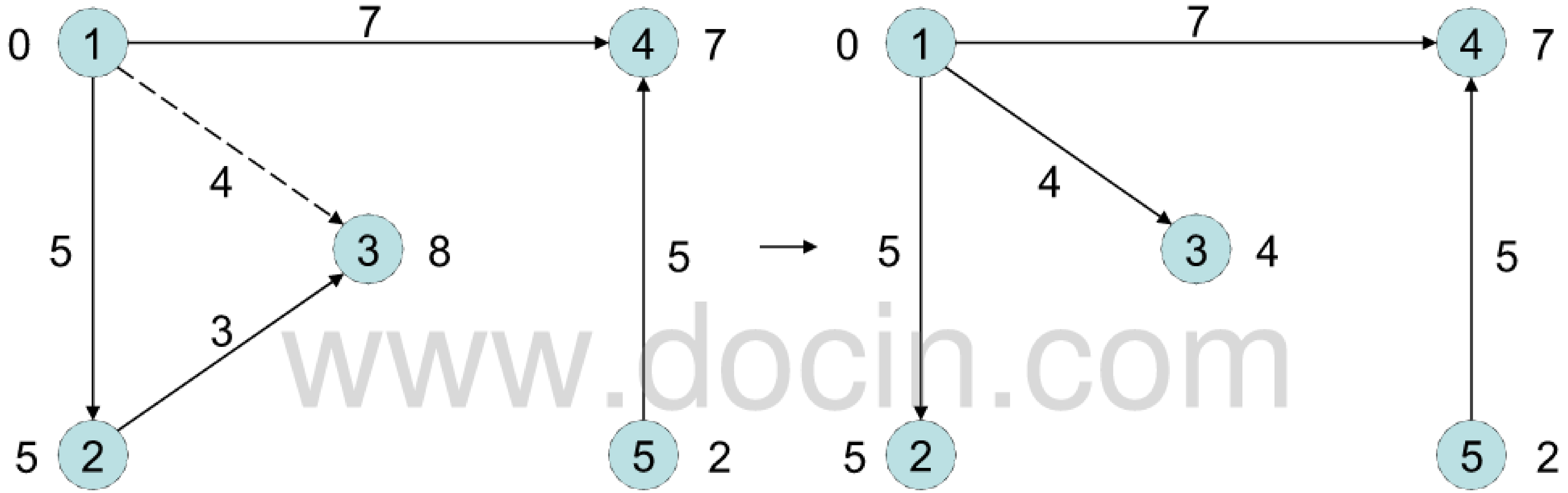
Price Setting

- You want to set the price y_i at node i
  - For all ji in T, $y_i = y_j + c_{ji}$
  - If the price was lower then you would lose money
  - If the price was higher then a competitor could undercut your price

Problem / Solution

- A competitor could still undercut your price
  - If there was an arc *ki* not in T with $y_i > y_k + c_{ki}$
- You don't want to lose business, so you also plan to ship over *ki*
  - You want to ship as much as possible
  - You must also adjust the rest of your schedule to conform with demand

Example



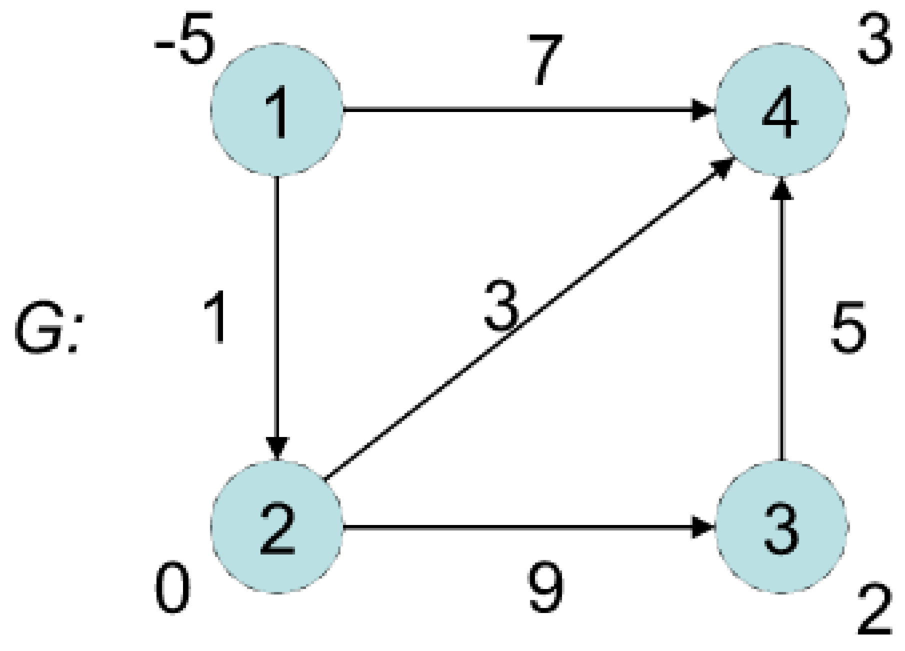
Optimality

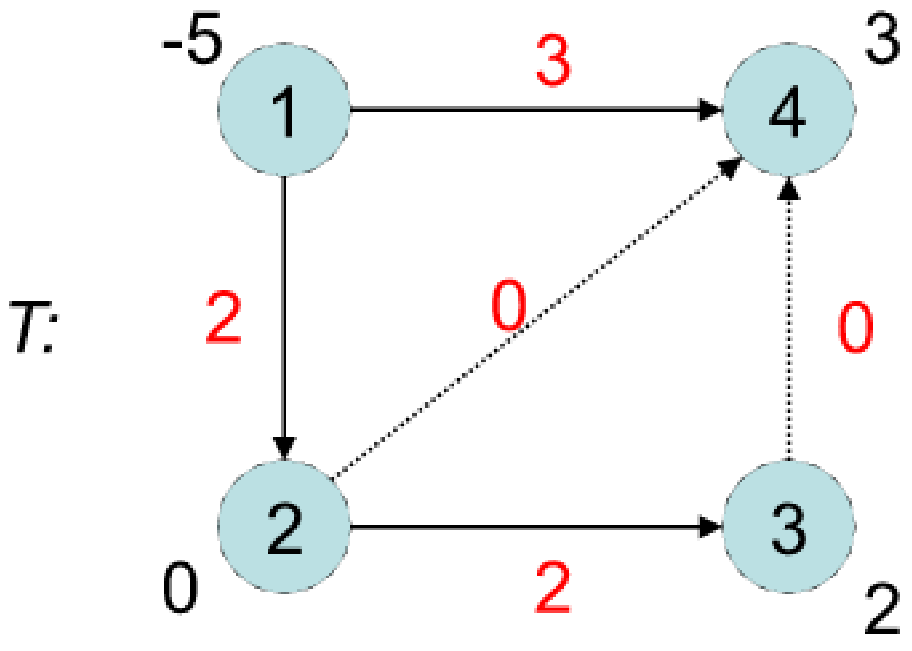
- If no arc like *ki* exists, then your prices can not be undercut
  - A competitor could break even at best

Algebraic Description (Step 1)

- Each step begins with a feasible tree solution **x** defined by a tree *T*.
  - **x** is a column vector with a flow value for each arc.
- In step 1 we calculate a value for each node such that $y_i + c_{ij} = y_j, \forall\ ij \in T$.
  - **y** is a row vector with value of each node.
- **c** is the cost (row) vector, **b** is the demand (column) vector, and **A** is the incidence matrix.

Algebraic Description (Step 1)





$$\mathbf{y} \quad \begin{bmatrix} 0 & 1 & 10 & 7 \end{bmatrix}$$

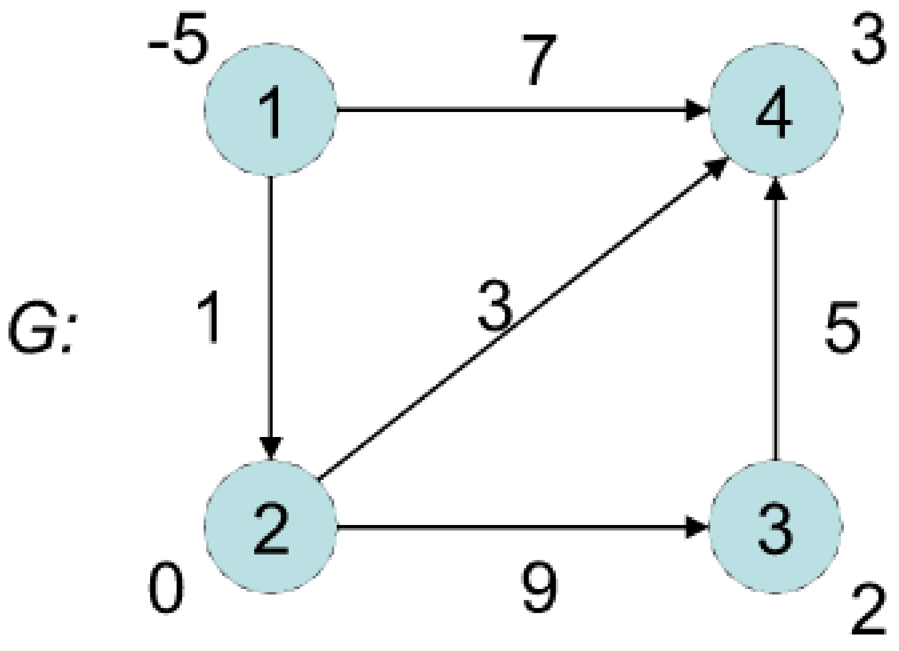
$$\mathbf{A} \quad \begin{array}{c|ccccc} & 12 & 14 & 23 & 24 & 34 \\ \hline 1 & -1 & -1 & 0 & 0 & 0 \\ 2 & 1 & 0 & -1 & -1 & 0 \\ 3 & 0 & 0 & 1 & 0 & -1 \\ 4 & 0 & 1 & 0 & 1 & 1 \end{array} \qquad \mathbf{x} \begin{bmatrix} 2 \\ 3 \\ 2 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{b} \begin{bmatrix} -5 \\ 0 \\ 2 \\ 3 \end{bmatrix}$$

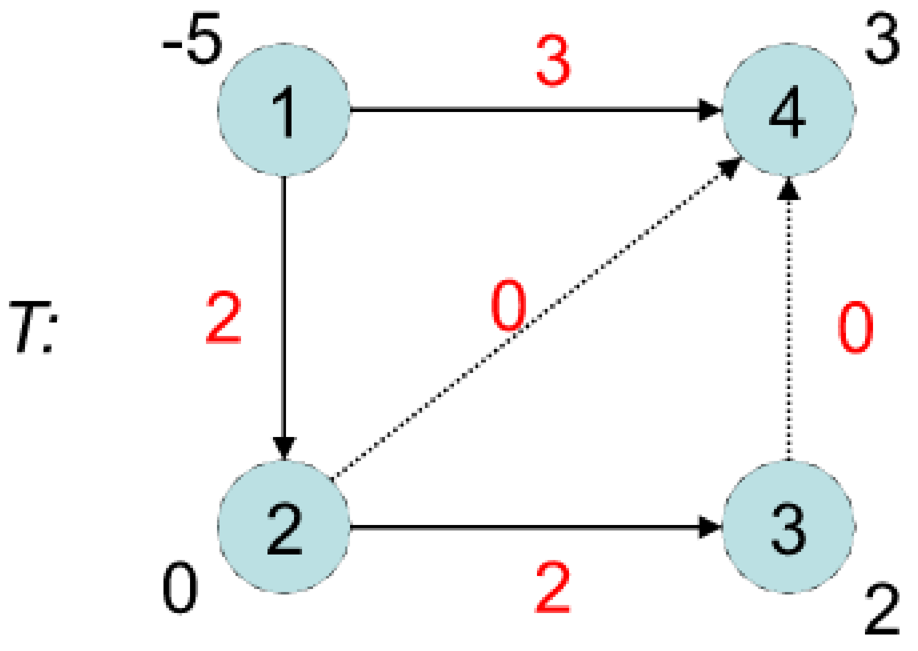
$$\mathbf{c} \quad \begin{bmatrix} 1 & 7 & 9 & 3 & 5 \end{bmatrix}$$

Algebraic Description (Step 1)

- We define **c’** = **c** – **yA**.
- **c’** is the difference between the cost of an arc and the value difference across the arc.
- If $ij \in T$ then $c'_{ij} = c_{ij} + y_i - y_j = 0$.
- If $ij \notin T$ and if $c'_{ij} < 0$ then *ij* is candidate for entering arc.
- Also if $ij \notin T$ then $x_{ij} = 0$, combining with above we get **c’x** = **0** ($\forall$ *ij*, either $c'_{ij} = 0$ or $x_{ij}=0$).

Algebraic Description (Step 1)





A **x** **b**

$$\begin{array}{c|ccccc} & 12 & 14 & 23 & 24 & 34 \\ \hline 1 & -1 & -1 & 0 & 0 & 0 \\ 2 & 1 & 0 & -1 & -1 & 0 \\ 3 & 0 & 0 & 1 & 0 & -1 \\ 4 & 0 & 1 & 0 & 1 & 1 \end{array} \qquad \begin{bmatrix} 2 \\ 3 \\ 2 \\ 0 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} -5 \\ 0 \\ 2 \\ 3 \end{bmatrix}$$

y

$$\begin{bmatrix} 0 & 1 & 10 & 7 \end{bmatrix}$$

c

$$\begin{bmatrix} 1 & 7 & 9 & 3 & 5 \end{bmatrix}$$

$$\mathbf{c}' \begin{bmatrix} 0 & 0 & 0 & -3 & 8 \end{bmatrix}$$

Algebraic Description (Step 1)

- For any feasible solution $\mathbf{x'}$ (i.e. $\mathbf{Ax'} = \mathbf{b}$, $\mathbf{x'} \geq \mathbf{0}$), its cost is

$$\begin{aligned} \mathbf{cx'} &= (\mathbf{c'} + \mathbf{yA})\mathbf{x'} && (\mathbf{c'} = \mathbf{c} - \mathbf{yA}) \\ &= \mathbf{c'x'} + \mathbf{yAx'} && \\ &= \mathbf{c'x'} + \mathbf{yb}. && (\mathbf{Ax'} = \mathbf{b}) \end{aligned}$$

- In particular for $\mathbf{x}$, its cost is

$$\mathbf{cx} = \mathbf{c'x} + \mathbf{yb} = \mathbf{yb}. \qquad (\mathbf{c'x} = \mathbf{0})$$

- Substituting for $\mathbf{yb}$ in the cost of $\mathbf{x'}$ we get

$$\mathbf{cx'} = \mathbf{cx} + \mathbf{c'x'} \qquad (1)$$

- So if $\mathbf{c'x'} < \mathbf{0}$ then $\mathbf{x'}$ is a better solution than $\mathbf{x}$.

Algebraic Description (Step 2)

- In step 2 we find an arc $e = uv$ such that

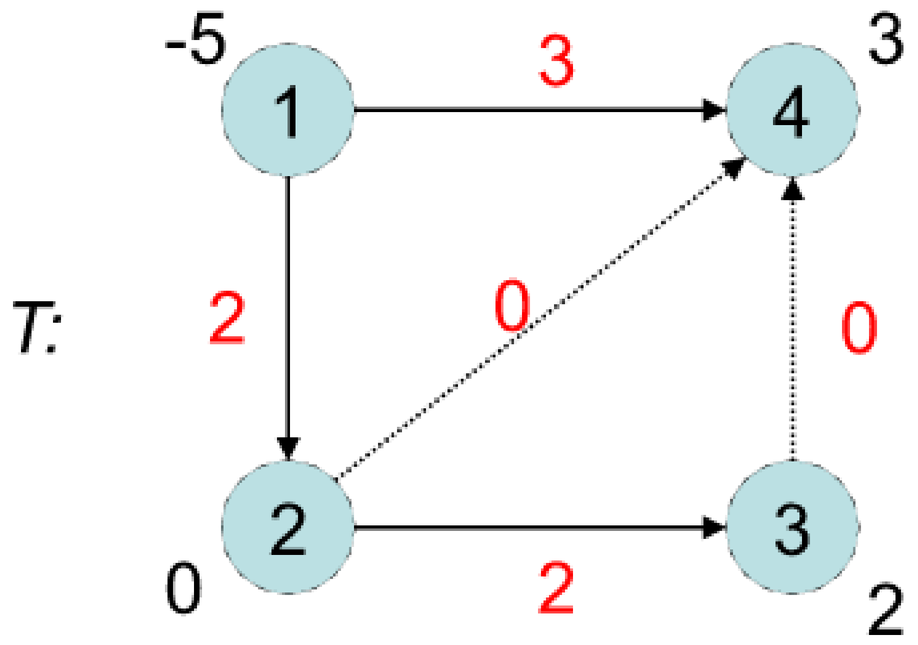
  $y_u + c_{uv} < y_v$ (i.e. $c'_{uv} < 0$).
- If no such arc exists then $\mathbf{c'} \geq \mathbf{0}$ and so $\mathbf{c'x'} \geq \mathbf{0}$.
- Hence equation (1) implies $\mathbf{cx'} \geq \mathbf{cx}$ for every feasible solution $\mathbf{x'}$, and so $\mathbf{x'}$ is optimal.
- If we find such an arc *e*, we it to the tree *T*.

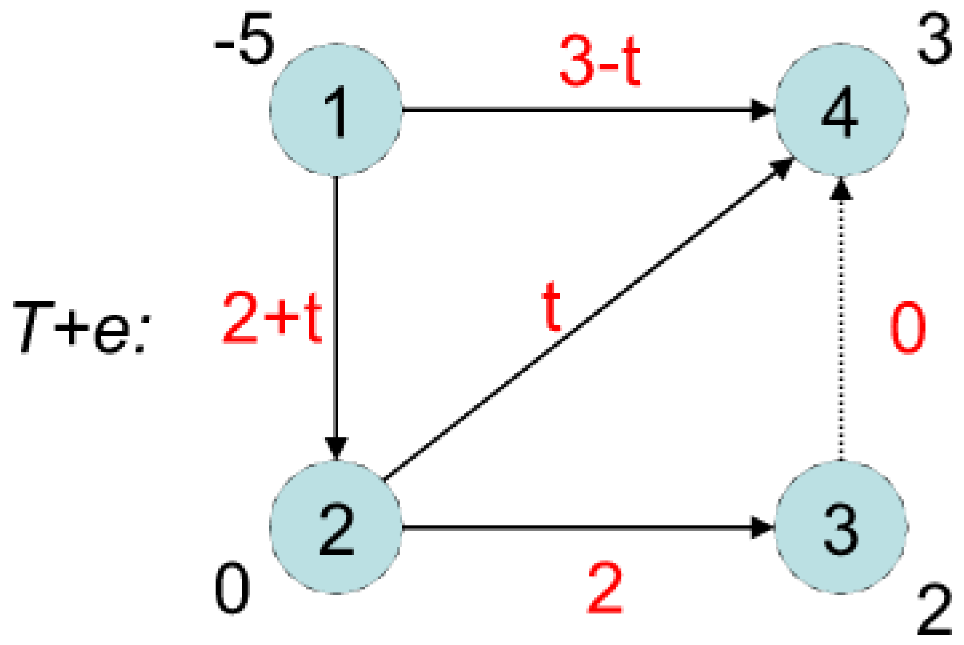
Algebraic Description (Step 3)

- For step 3, *T* + *e* has a unique cycle.
- Traversing the cycle in the direction of *e* we define *forward arcs* as arcs pointing in the same direction as *e* and *reverse arcs* as arcs pointing in the opposite direction.
- Then we set

$$\bar{x}_{ij} = \begin{cases} x_{ij} + t & \text{if } ij \text{ is a forward arc,} \\ x_{ij} - t & \text{if } ij \text{ is a reverse arc,} \\ x_{ij} & \text{if } ij \text{ is not on the cycle.} \end{cases}$$

Algebraic Description (Step 3)





$$\mathbf{c}' \quad \begin{bmatrix} 0 & 0 & 0 & -3 & 8 \end{bmatrix}$$

$$\mathbf{x} \quad \begin{bmatrix} 2 \\ 3 \\ 2 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{x}' \quad \begin{bmatrix} 2+t \\ 3-t \\ 2 \\ 0+t \\ 0 \end{bmatrix}$$

Algebraic Description (Step 3)

- Now, **Ax'** = **Ax** = b, because for each node of the cycle the extra $\pm t$ cancel each other.
- So if we choose *t* such that **x'** ≥ **0**, then **x'** is feasible.
- Since *e* is the only arc with $c'_{ij} \neq 0$ and $x'_{ij} \neq 0$, we have **c'x'** = $c'_e x'_e = c'_e t$.
- Substituting in equation (1) we get **cx'** = **cx** + $c'_e t$.
- We want to choose *t* such that **x'** is feasible and which minimizes **cx'**.

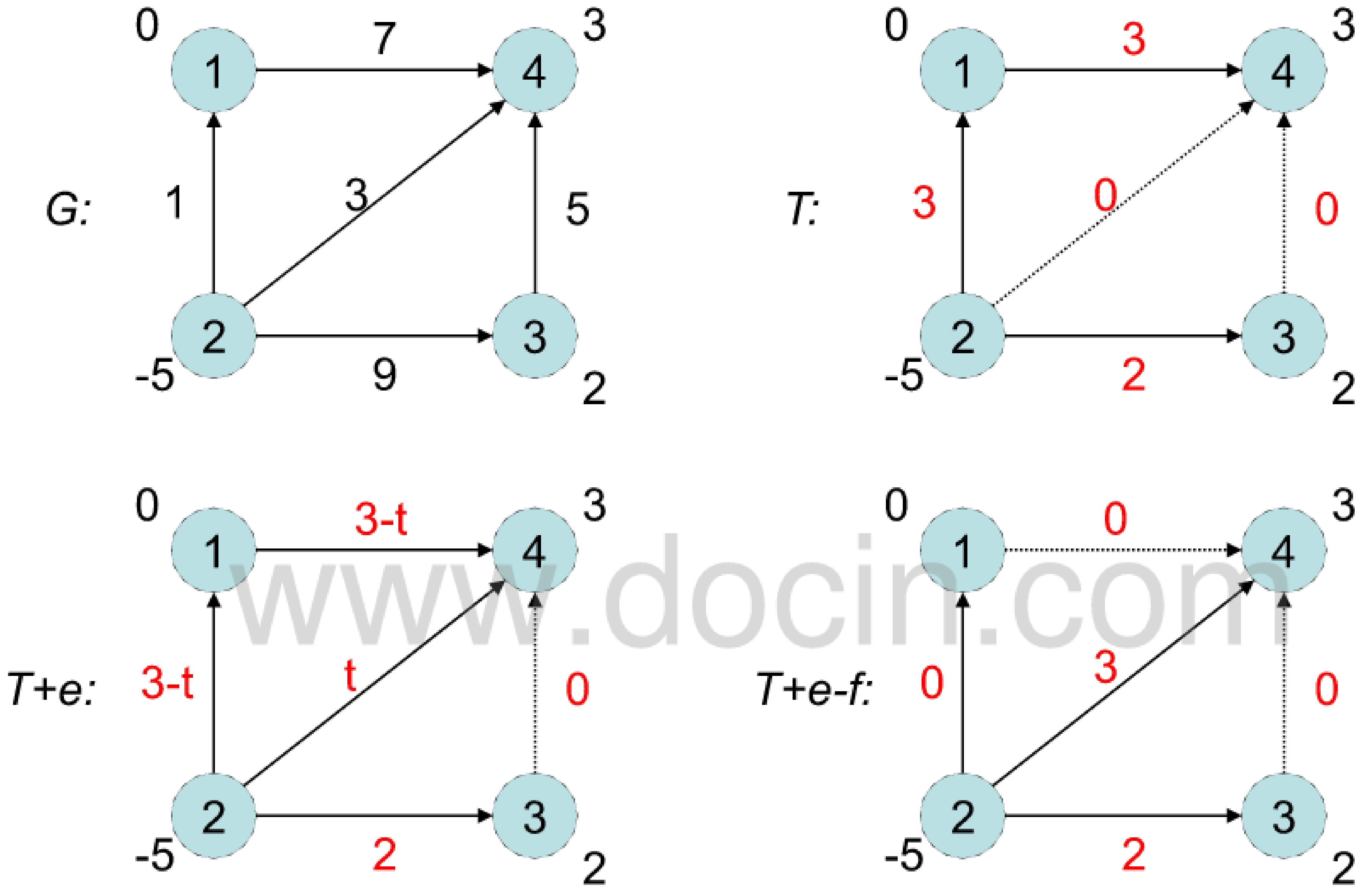
Algebraic Description (Step 3)

- We have, $\mathbf{cx'} = \mathbf{cx} + c'_e t$.
- Since $c'_e < 0$, to minimize **cx'**, we have to maximize t.
- To make sure **x'** is feasible (i.e. $\mathbf{x'} \geq \mathbf{0}$), we find a reverse arc f with minimal value x_f, and let $t = x_f$.
- The new feasible solution **x'** has $x'_f = 0$, so f is the leaving arc.
- Removing f breaks the only cycle in $T + e$, so $T + e – f$ is the new tree that defines the new feasible tree solution **x'**.

Degeneracy and Cycling

- If there is more than one candidate for the leaving arc, then for each candidate arc *ij*, $x'_{ij} = 0$.
- Only one of the candidate arcs leaves the tree, so the new solution has $x'_{ij}=0$ for at least one of its tree arcs.
- Such a solution is called a *degenerate* solution.
- They could lead to pivots with $t = x_f = 0$, that is no decrease in the cost.
- Degeneracy is necessary but not sufficient for cycling.

Degeneracy and Cycling



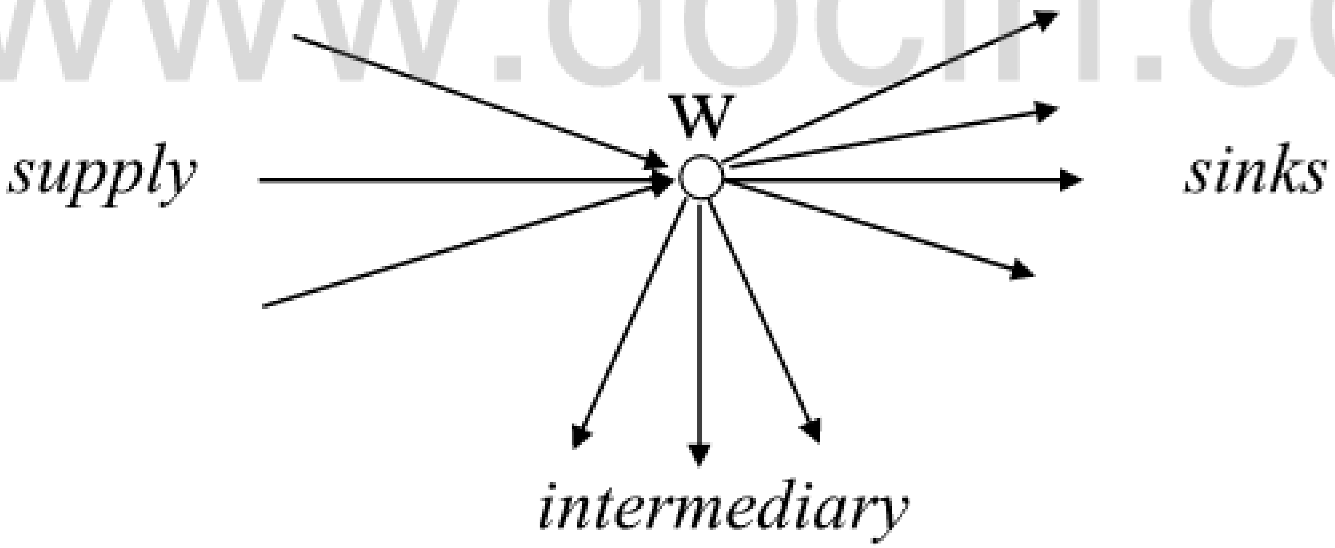
Degeneracy and Cycling

- Cycling is very rare. No practical example with cycling has been found.
- The first artificial example was constructed 13 years after the appearance of the network simplex method.
- Cycling can be avoided by proper choice of the leaving arc. We will see this later.

Initialization

- If there is a node w | there is an arc:
  - – from every supply node to w
  - – from w to every sink/intermediary node

Then there is an initial feasible solution



Initialization

- If no such w, add artificial arcs
- Associate a penalty p_{ij} for using these arcs ij:
  - – $p_{ij} = 0$ for original arcs
  - – $p_{ij} = 1$ for artificial arcs

- Solve auxiliary problem: Min $\Sigma p_{ij} x_{ij}$

Initialization

i. T contains an artificial arc ij with $x_{ij} > 0$

=> Original problem has no feasible sol

ii.T contains no artificial arc

=> T is a feasible sol. for original problem

iii. Every artificial arcs ij in T has $x_{ij} = 0$

=> original problem has a feasible sol. But not a feasible tree sol.

Decomposition

- For each set S and feasible solution **x**:

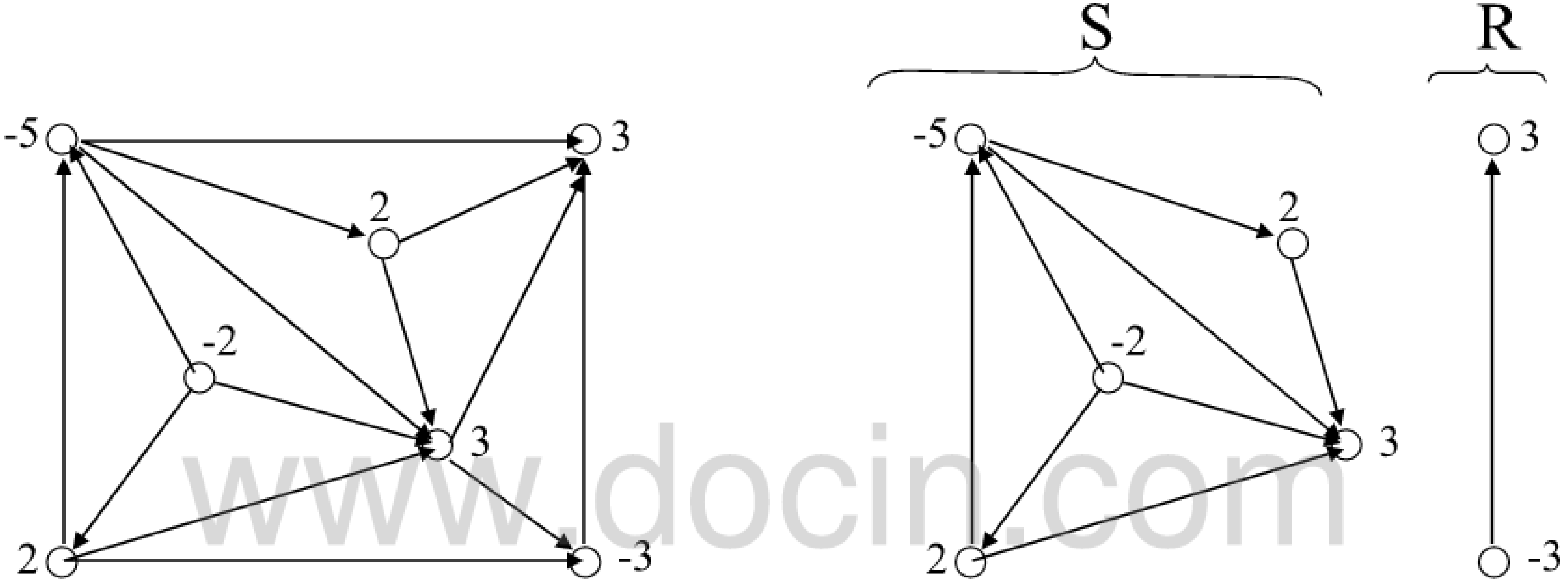
$$\sum_{\substack{i \notin S \\ j \in S}} x_{ij} - \sum_{\substack{i \in S \\ j \notin S}} x_{ij} = \sum_{k \in S} b_k$$

  - Import – Export = Net demand
  - In Ax = b, sum equations corresponding to nodes in S

Decomposition

- Assume there is a partition R and S of the nodes such that
  - $\sum_{k \in S} b_k = 0$
  - there is no arc ij with i in R and j in S
- S cannot afford to export
  - If i in S and j in R then $x_{ij} = 0$

Decomposition



Decomposition

Decompose optimal tree T of auxiliary problem same way:

Take arbitrary artificial arc uv

k in R if $y_k \leq y_u$ and k in S otherwise

Decomposition

- In the solution of the auxiliary problem:

$$\sum_{\substack{i \notin S \\ j \in S}} x_{ij} - \sum_{\substack{i \in S \\ j \notin S}} x_{ij} = \sum_{k \in S} b_k$$

- No original arc has i in R and j in S

- x_{ij} for artificial arcs in 0

None of these arcs is in T

$\sum_{k \in S} b_k = 0$ and no arc ij with i in R and j in S

Updating nodes

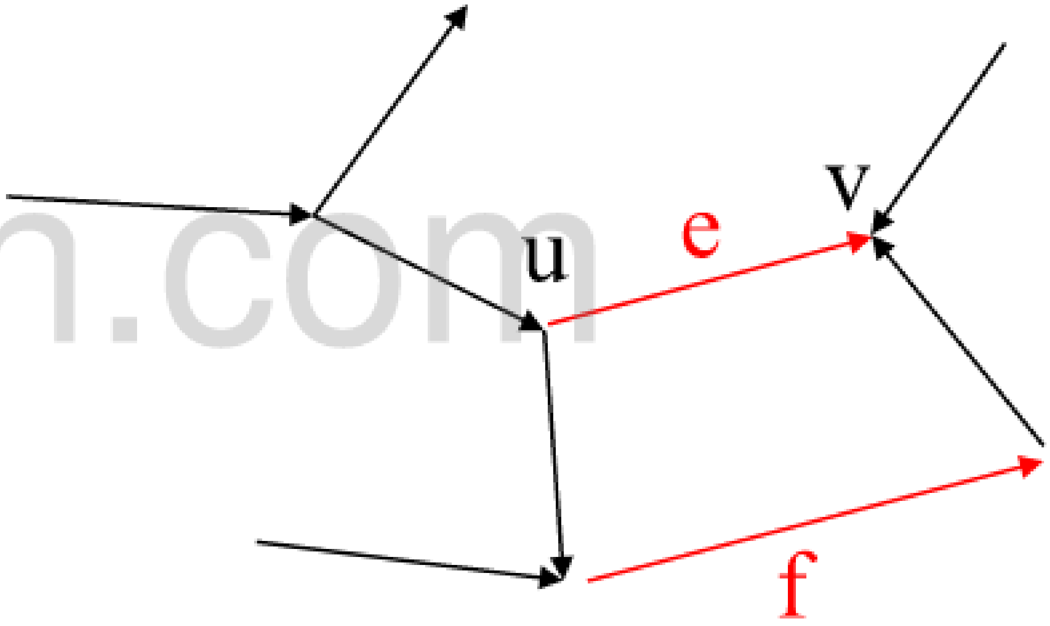
In T: $y_i + c_{ij} = y_j, \forall$ ij in T

Goal: $y'_i + c_{ij} = y'_j, \forall$ ij in T + e – f

Define:

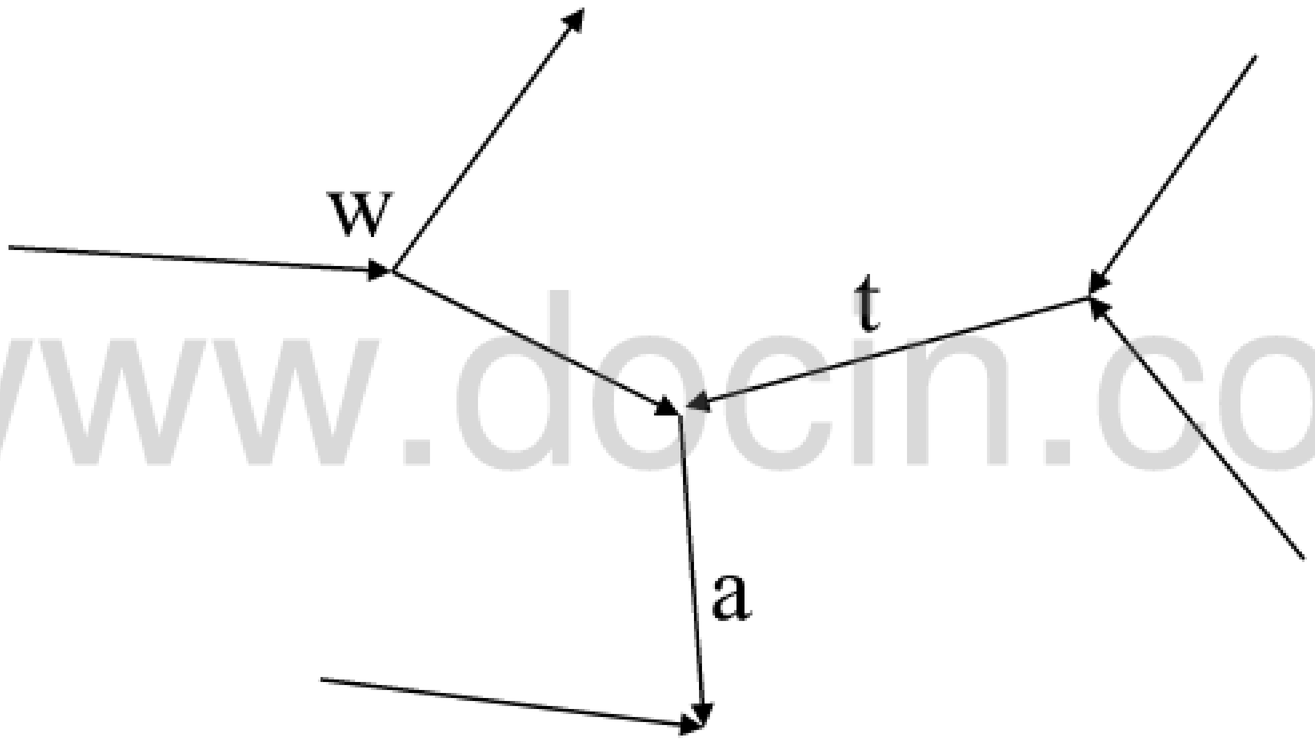
$$y'_k = \begin{cases} y_k & (k \text{ in } T_u) \\ y_k + c'_e & (k \text{ in } T_v) \end{cases}$$

$$c'_e = c_e + y_u - y_v$$



Avoid Cycling

Direction of an arc with respect to root



Avoid Cycling

Thm: If each degenerate pivot leads from T to T + e – f such that e is directed away from the root of T + e – f => no cycling

Avoid Cycling

Define: $g(T) = cx$

$h(T) = \Sigma_k(y_k - y_w)$

- $g(T_i) \geq g(T_{i+1})$
- $h(T_{i+1}) = \Sigma_k(y'_k - y'_w) = \Sigma_k(y_k - y_w) + c_e|T_v|$

$g(T_i) = g(T_{i+1}) \Rightarrow h(T_i) > h(T_{i+1})$

Avoid Cycling

$T_i = T_j, \ i < j \ \Rightarrow$

$$g(T_i) = g(T_{i+1}) = \dots = g(T_j)$$

$$h(T_i) > h(T_{i+1}) > \dots > h(T_j)$$

Contradicting $h(T_i) = h(T_j)$

Avoid Cycling

Strongly feasible: all arcs ij in T | $x_{ij} = 0$ are directed away from the root

1. initial solution is strongly feasible

2. if T is strongly feasible, then T + e – f is strongly feasible

=> No cycling

Avoid Cycling

1. Starting with a strongly feasible:

Bad arc: directed toward the root and $x_{arc} = 0$

i. Start with T

ii. Remove bad arc uv: T_v and T_u

iii. If there is an arc ij | i in T_v and j in T_u, add ij to T

=> T' with less bad arcs

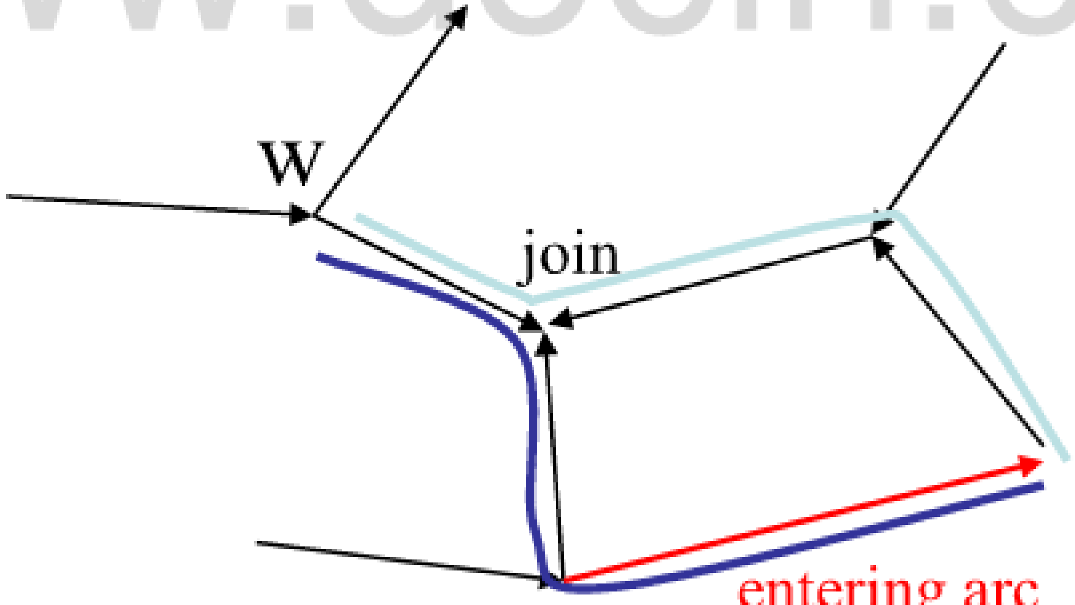
else decompose into two subproblems

Avoid Cycling

2. Arrive at a strongly feasible solution:

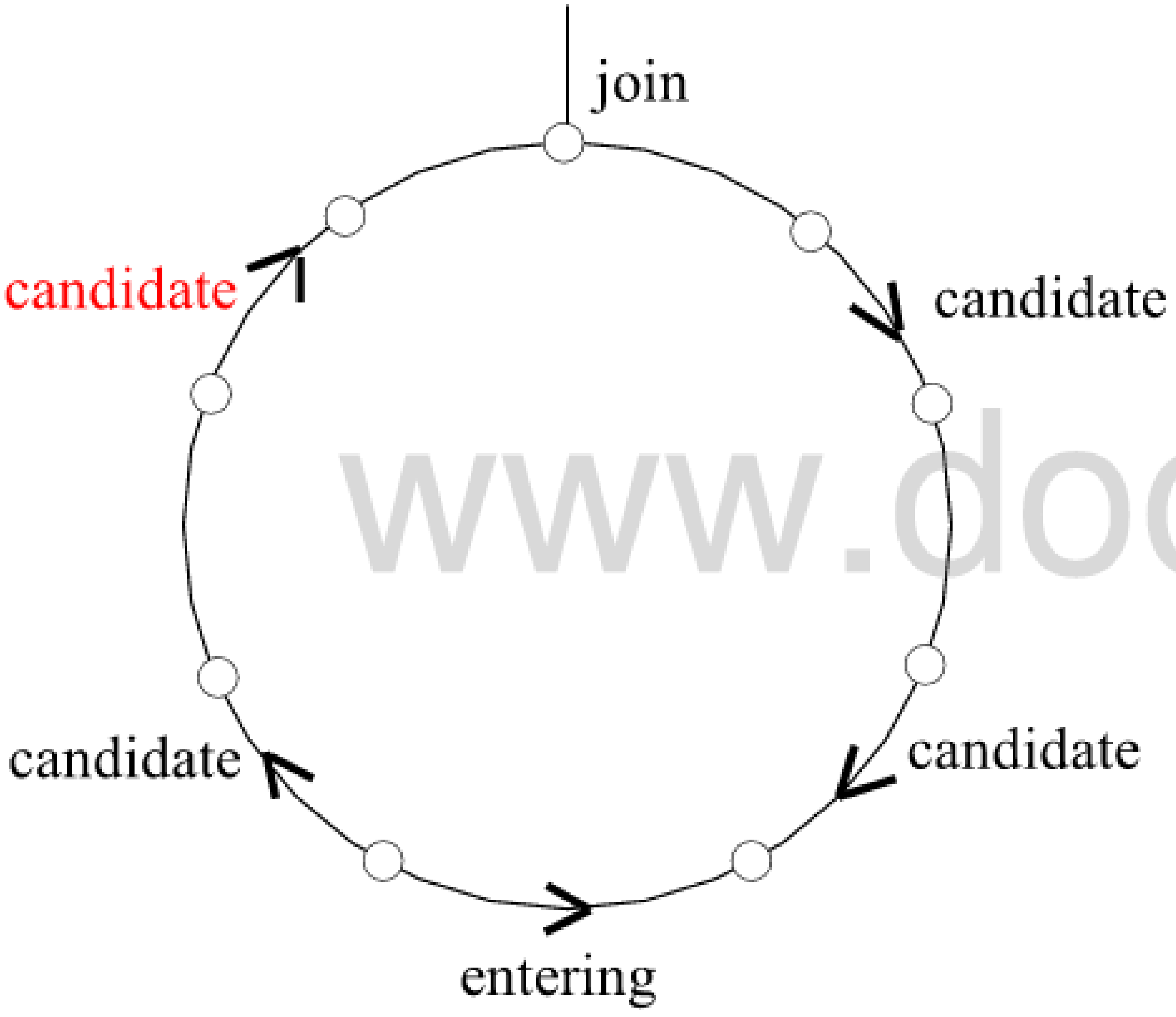
Entering arc: any candidate

Leaving arc: first candidate while traversing C in direction of e starting at the join



Avoid Cycling

Case 1: the pivot is non-degenerate



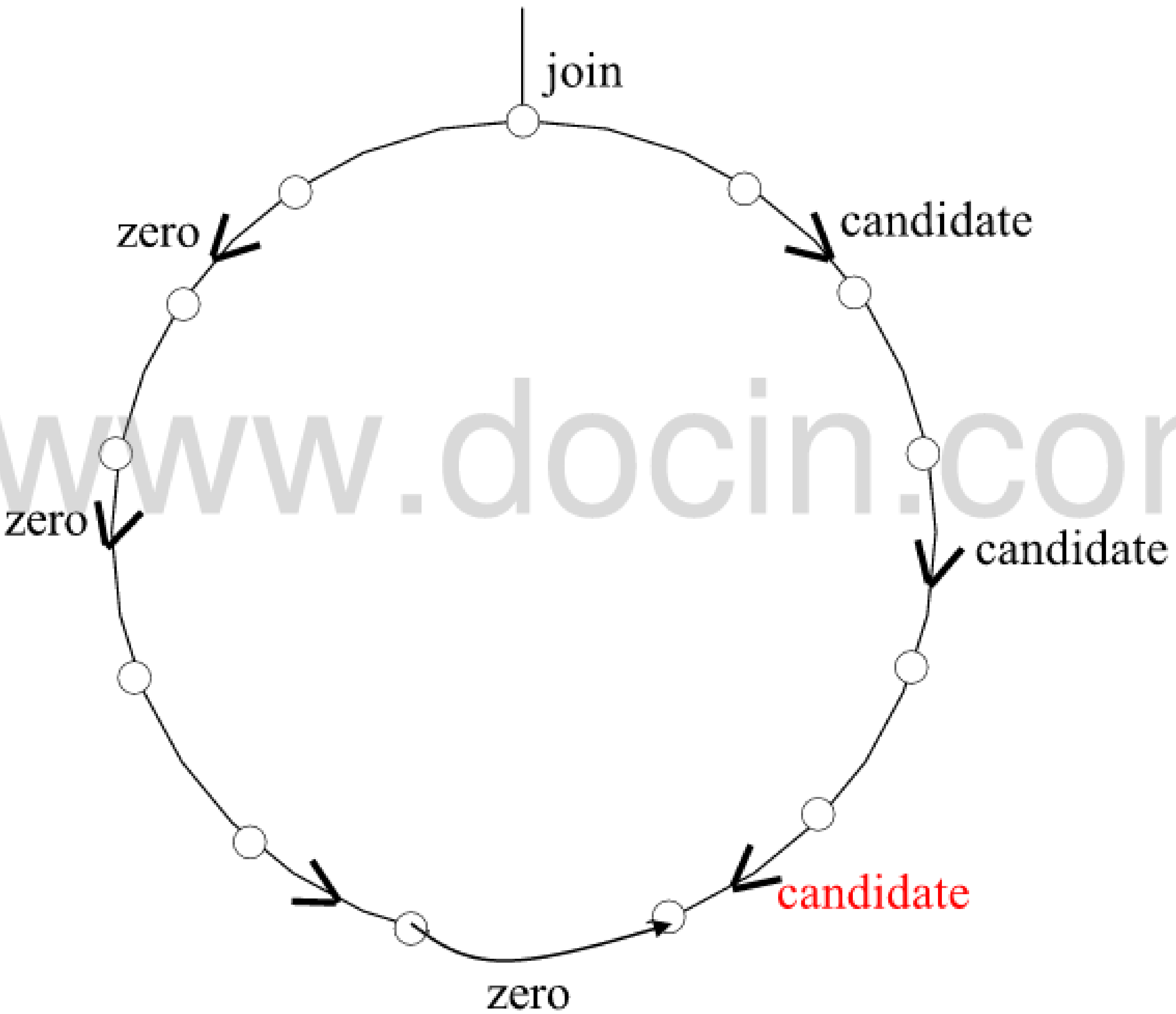
Candidate arcs ij:

- will have $x_{ij} = 0$ in the new sol.
- will be directed away from w

Avoid Cycling

Case 2: the pivot is degenerate

No reverse zero arcs



No forward zero arcs

Questions?