15.093 Optimization Methods

Lecture 10: Network Optimization
The Network Simplex Algorithm

Network Optimization — Why do we care?

- Networks and associated optimization problems constitute reoccurring structures in many real-world applications.
- The network structure often leads to additional insight and improved understanding.
- It also enable us to design more efficient algorithms.

Outline — Today's Lecture

- The Simplex Algorithm: A Reminder
- The Network Simplex: A Combinatorial View
- The Network Simplex: An Animated View
- The Network Simplex: An Algebraic View

The Simplex Algorithm — A Reminder — The Problem...

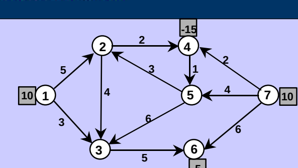
$$\begin{aligned} \min\ & c'x \\ \text{s.t.}\ & Ax = b \\ & x \geq 0 \end{aligned}$$

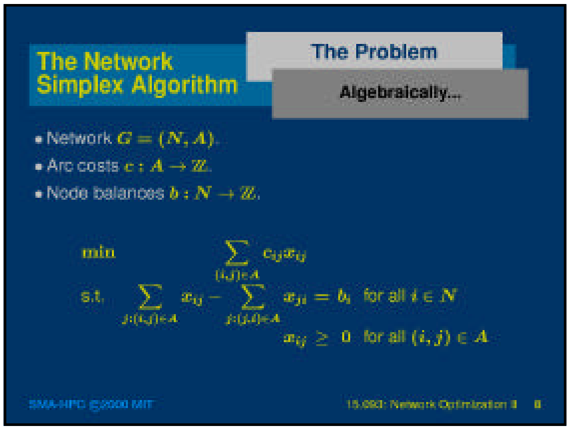
The Simplex Algorithm — A Reminder — The Algorithm

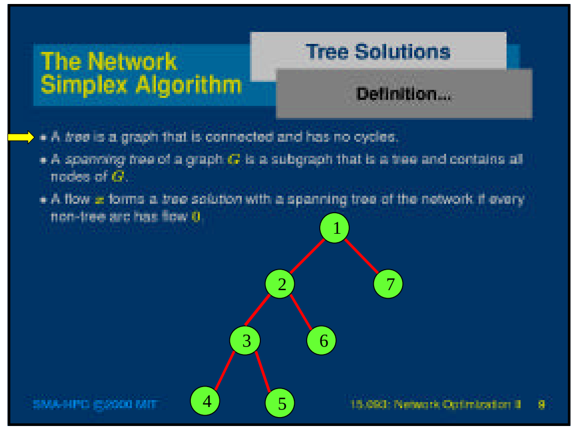
1. Start with basis $B = [A_{B(1)}, \ldots, A_{B(m)}]$ and BFS x.
2. Compute $\bar{c}_j = c_j - c'_B B^{-1} A_j$.
   - If $\bar{c}_j \geq 0$; x optimal; stop.
   - Select j such that $\bar{c}_j < 0$.
3. Compute $u = B^{-1} A_j$. $\theta^* = \min_{1 \leq i \leq m, u_i > 0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell}$.
4. Form a new basis by replacing $A_{B(\ell)}$ with A_j.
5. $y_j = \theta^*$; $y_{B(i)} = x_{B(i)} - \theta^* u_i$.

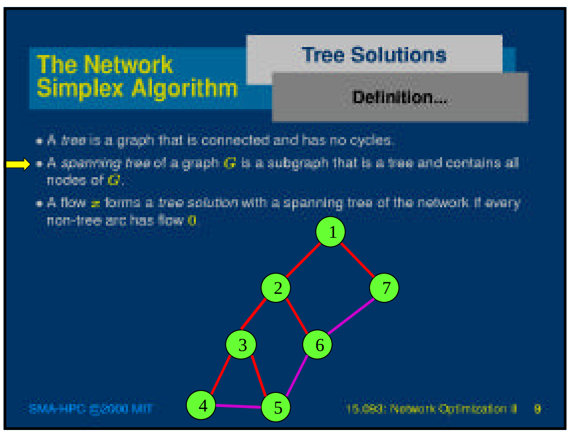
The Network Simplex Algorithm — The Problem — Combinatorially...

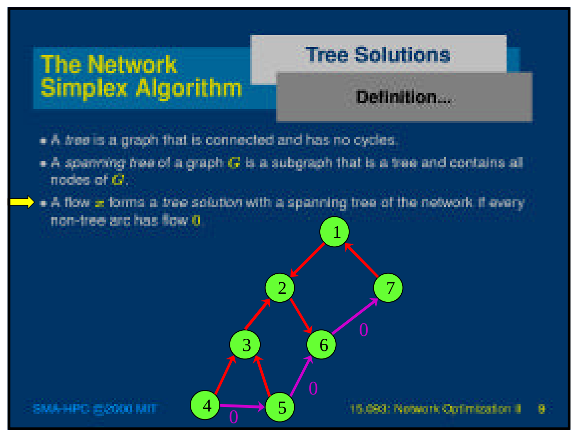
Determine a least cost shipment of a commodity through a network in order to satisfy demands at certain nodes from available supplies at other nodes. Arcs have costs associated with them.

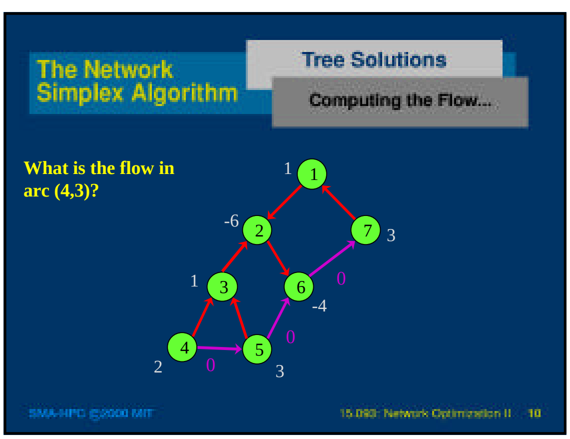


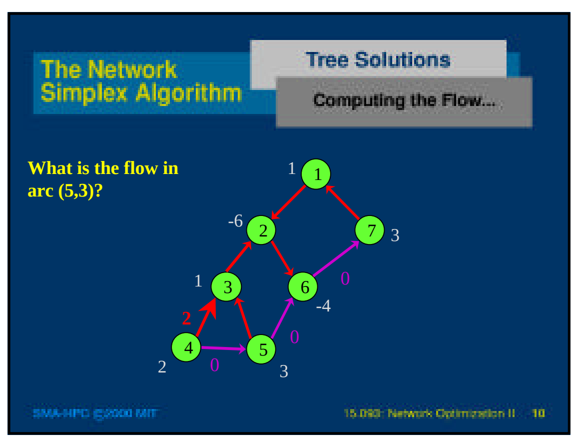

The Network Simplex Algorithm
The Problem
Algebraically...
• Network $G = (N, A)$.
• Arc costs $c : A \to \mathbb{Z}$.
• Node balances $b : N \to \mathbb{Z}$.
min $\sum_{(i,j) \in A} c_{ij} x_{ij}$
s.t. $\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = b_i$ for all $i \in N$
$x_{ij} \geq 0$ for all $(i,j) \in A$
SMA-HPC ©2000 MIT
15.093: Network Optimization II 8

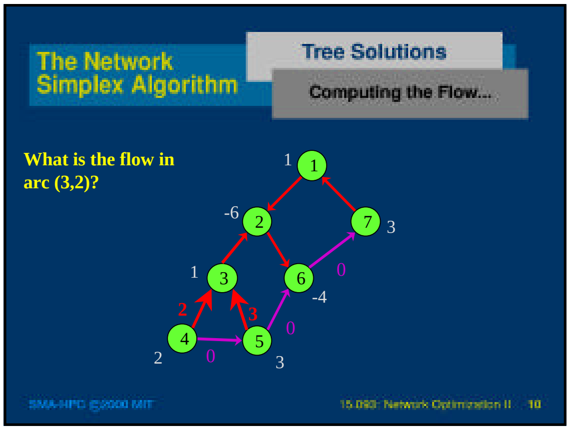

The Network Simplex Algorithm
Tree Solutions
Definition...
• A *tree* is a graph that is connected and has no cycles.
• A *spanning tree* of a graph G is a subgraph that is a tree and contains all nodes of G.
• A flow x forms a *tree solution* with a spanning tree of the network if every non-tree arc has flow 0.
1
2
7
3
6
4
5
SMA-HPC ©2000 MIT
15.093: Network Optimization II 9

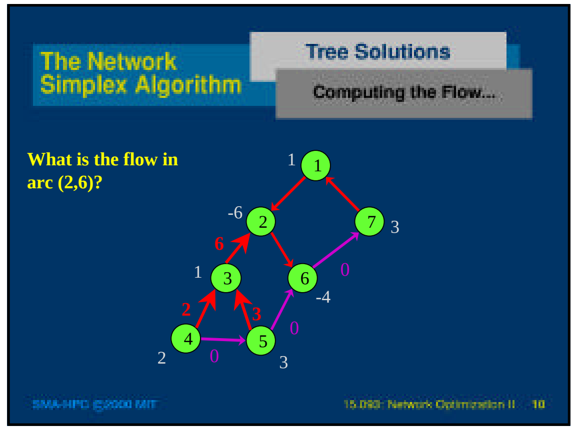

The Network Simplex Algorithm
Tree Solutions
Definition...
• A *tree* is a graph that is connected and has no cycles.
• A *spanning tree* of a graph G is a subgraph that is a tree and contains all nodes of G.
• A flow x forms a *tree solution* with a spanning tree of the network if every non-tree arc has flow 0.
1
2
7
3
6
4
5
SMA-HPC ©2000 MIT
15.093: Network Optimization II 9

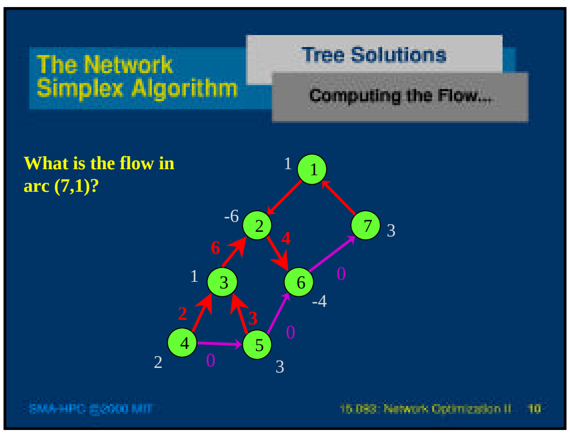

The Network Simplex Algorithm
Tree Solutions
Definition...
• A *tree* is a graph that is connected and has no cycles.
• A *spanning tree* of a graph G is a subgraph that is a tree and contains all nodes of G.
• A flow x forms a *tree solution* with a spanning tree of the network if every non-tree arc has flow 0.
1
2
7
0
3
6
0
4
0
5
SMA-HPC ©2000 MIT
15.093: Network Optimization II 9

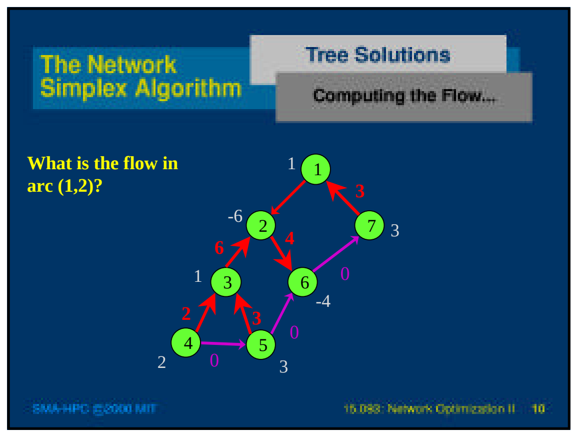

The Network Simplex Algorithm
Tree Solutions
Computing the Flow...
What is the flow in arc (4,3)?
1
1
-6
2
7
3
1
3
6
0
-4
0
4
5
2
0
3
SMA-HPC ©2000 MIT
15.093: Network Optimization II 10

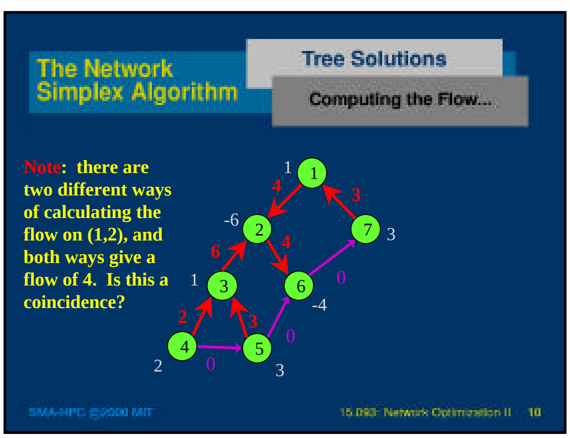

The Network Simplex Algorithm
Tree Solutions
Computing the Flow...
What is the flow in arc (5,3)?
1
1
-6
2
7
3
1
3
6
0
2
-4
0
4
5
2
0
3
SMA-HPC ©2000 MIT
15.093: Network Optimization II 10

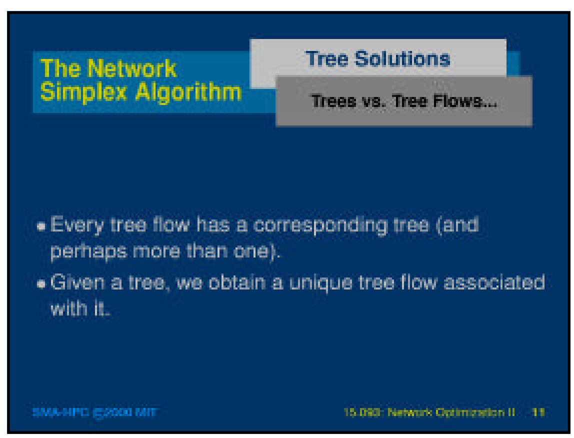

The Network Simplex Algorithm
Tree Solutions
Computing the Flow...
What is the flow in arc (3,2)?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 10

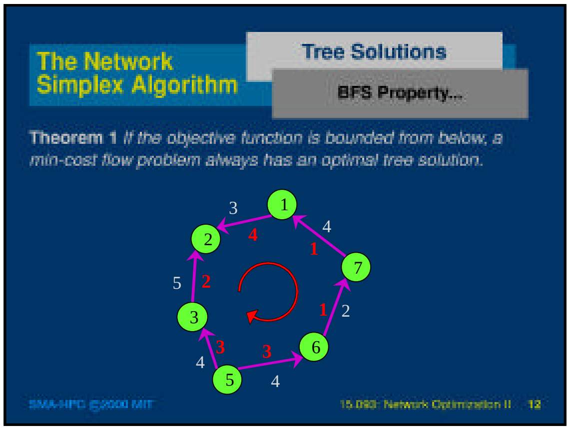

The Network Simplex Algorithm
Tree Solutions
Computing the Flow...
What is the flow in arc (2,6)?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 10

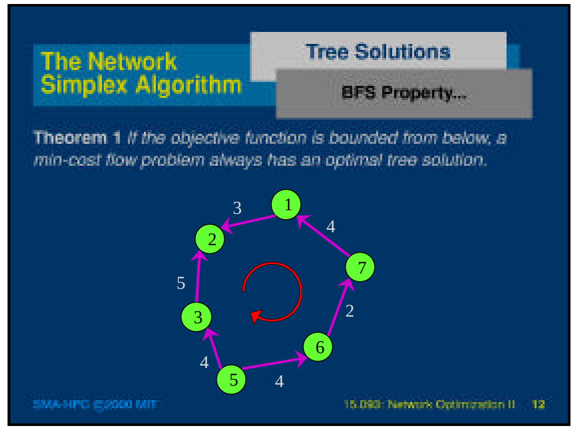

The Network Simplex Algorithm
Tree Solutions
Computing the Flow...
What is the flow in arc (7,1)?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 10

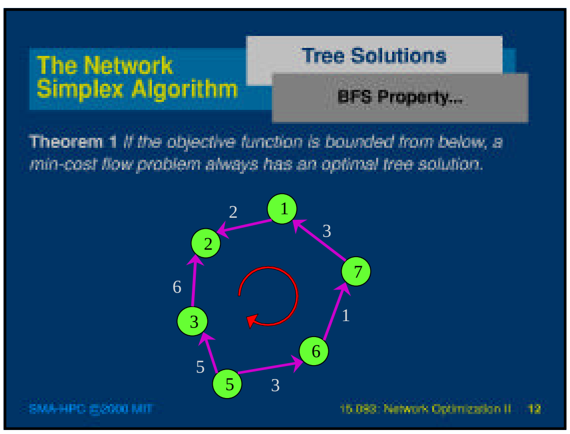

The Network Simplex Algorithm
Tree Solutions
Computing the Flow...
What is the flow in arc (1,2)?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 10

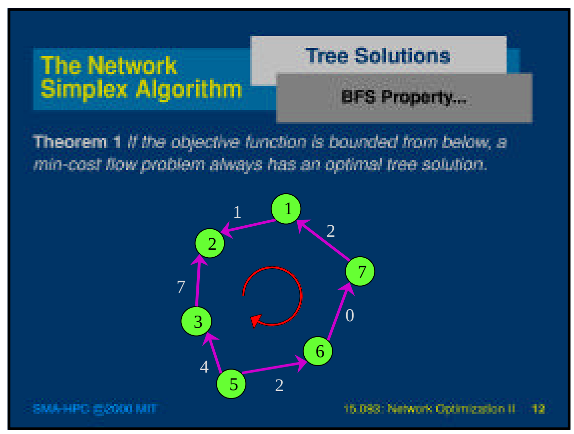

The Network Simplex Algorithm
Tree Solutions
Computing the Flow...
Note: there are two different ways of calculating the flow on (1,2), and both ways give a flow of 4. Is this a coincidence?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 10

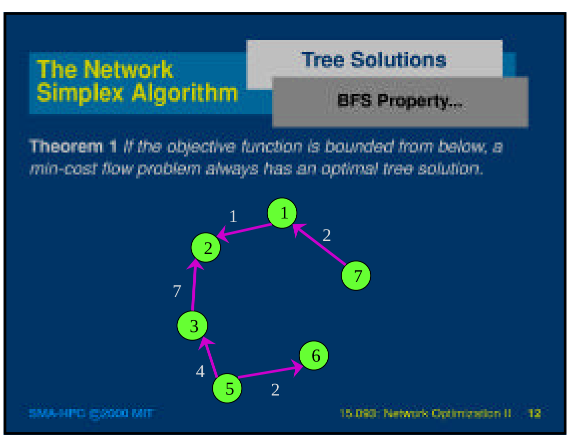

The Network Simplex Algorithm
Tree Solutions
Trees vs. Tree Flows...
• Every tree flow has a corresponding tree (and perhaps more than one).
• Given a tree, we obtain a unique tree flow associated with it.
SMA-HPC ©2000 MIT
15.093: Network Optimization II 11

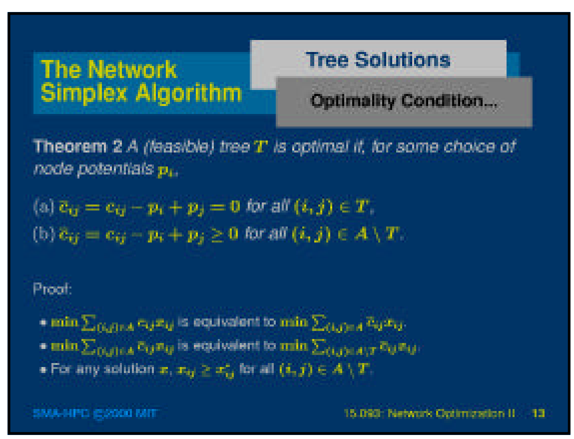

The Network Simplex Algorithm
Tree Solutions
BFS Property...
Theorem 1 *If the objective function is bounded from below, a min-cost flow problem always has an optimal tree solution.*
SMA-HPC ©2000 MIT
15.093: Network Optimization II 12

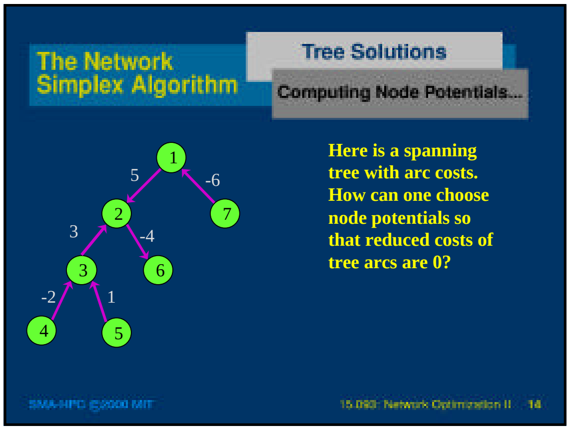

The Network Simplex Algorithm
Tree Solutions
BFS Property...
Theorem 1 *If the objective function is bounded from below, a min-cost flow problem always has an optimal tree solution.*
SMA-HPC ©2000 MIT
15.093: Network Optimization II 12

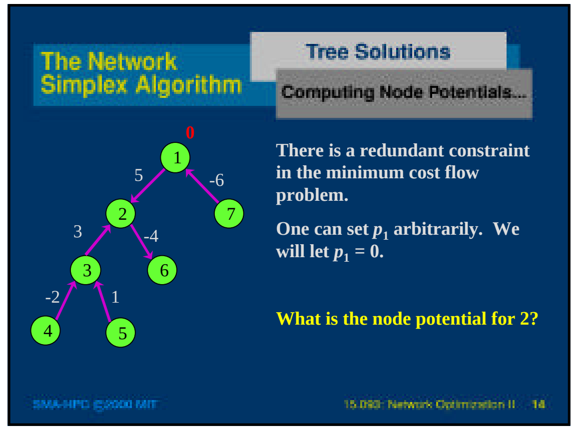

The Network Simplex Algorithm
Tree Solutions
BFS Property...
Theorem 1 *If the objective function is bounded from below, a min-cost flow problem always has an optimal tree solution.*
SMA-HPC ©2000 MIT
15.093: Network Optimization II 12

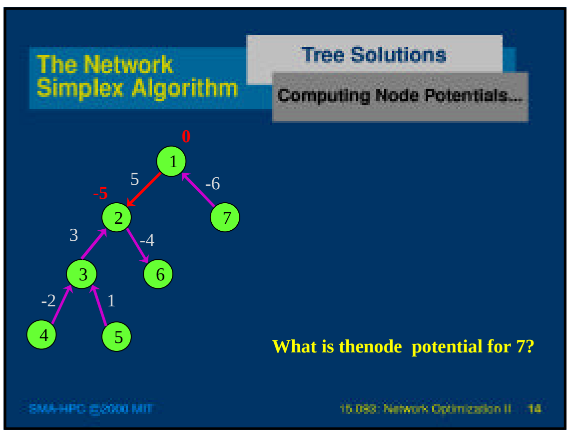

The Network Simplex Algorithm
Tree Solutions
BFS Property...
Theorem 1 *If the objective function is bounded from below, a min-cost flow problem always has an optimal tree solution.*
SMA-HPC ©2000 MIT
15.093: Network Optimization II 12

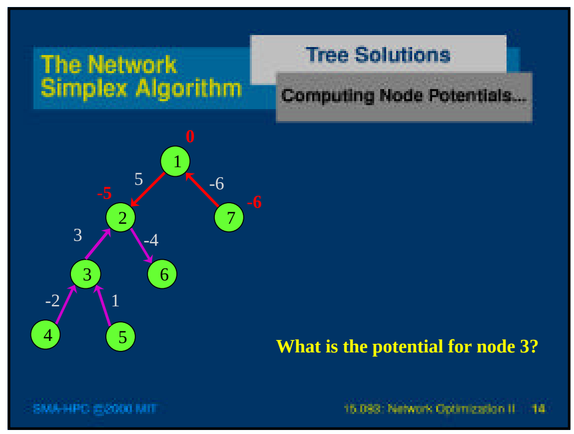

The Network Simplex Algorithm
Tree Solutions
BFS Property...
Theorem 1 *If the objective function is bounded from below, a min-cost flow problem always has an optimal tree solution.*
SMA-HPC ©2000 MIT
15.093: Network Optimization II 12

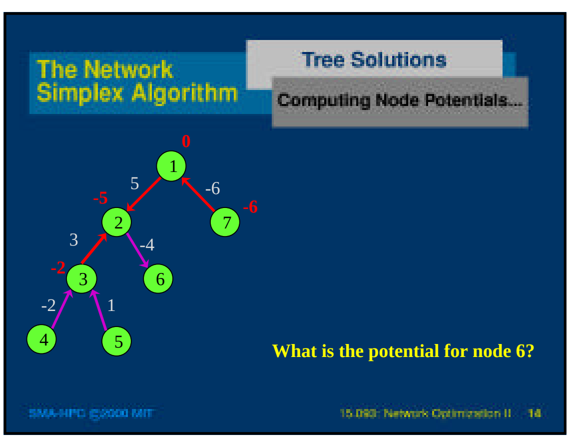

The Network Simplex Algorithm
Tree Solutions
Optimality Condition...
Theorem 2 *A (feasible) tree T is optimal if, for some choice of node potentials p_i,*
(a) $\bar{c}_{ij} = c_{ij} - p_i + p_j = 0$ *for all* $(i,j) \in T$,
(b) $\bar{c}_{ij} = c_{ij} - p_i + p_j \geq 0$ *for all* $(i,j) \in A \setminus T$.
Proof:
• $\min \sum_{(i,j)\in A} c_{ij}x_{ij}$ is equivalent to $\min \sum_{(i,j)\in A} \bar{c}_{ij}x_{ij}$.
• $\min \sum_{(i,j)\in A} \bar{c}_{ij}x_{ij}$ is equivalent to $\min \sum_{(i,j)\in A\setminus T} \bar{c}_{ij}x_{ij}$.
• For any solution x, $x_{ij} \geq x^*_{ij}$ for all $(i,j) \in A \setminus T$.
SMA-HPC ©2000 MIT
15.093: Network Optimization II 13

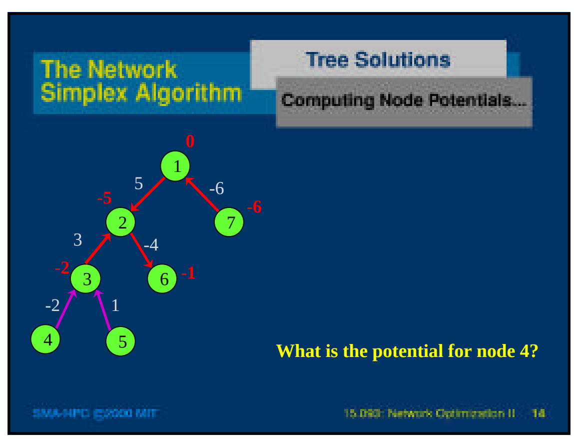

The Network Simplex Algorithm
Tree Solutions
Computing Node Potentials...
Here is a spanning tree with arc costs. How can one choose node potentials so that reduced costs of tree arcs are 0?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 14

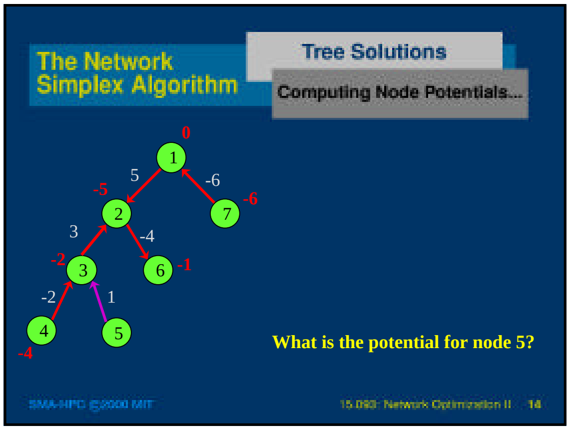

The Network Simplex Algorithm
Tree Solutions
Computing Node Potentials...
There is a redundant constraint in the minimum cost flow problem.
One can set p_1 arbitrarily. We will let $p_1 = 0$.
What is the node potential for 2?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 14

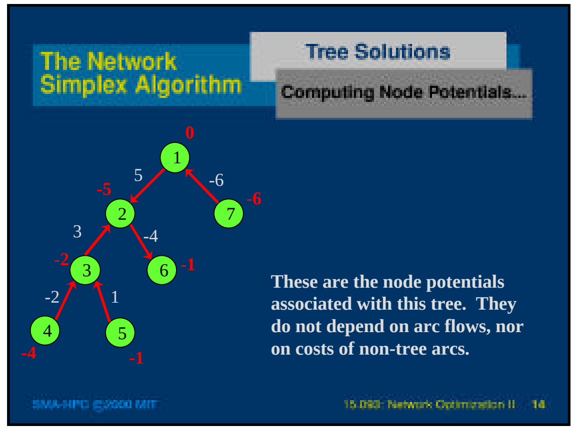

The Network Simplex Algorithm
Tree Solutions
Computing Node Potentials...
What is thenode potential for 7?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 14

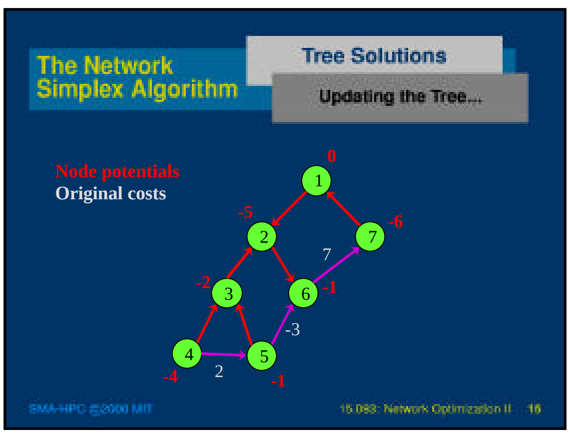

The Network Simplex Algorithm
Tree Solutions
Computing Node Potentials...
What is the potential for node 3?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 14

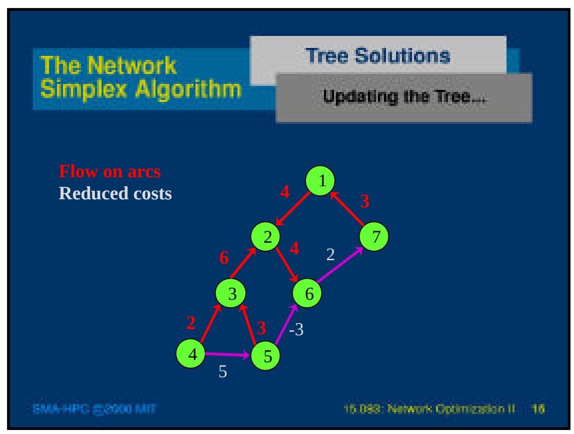

The Network Simplex Algorithm
Tree Solutions
Computing Node Potentials...
What is the potential for node 6?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 14

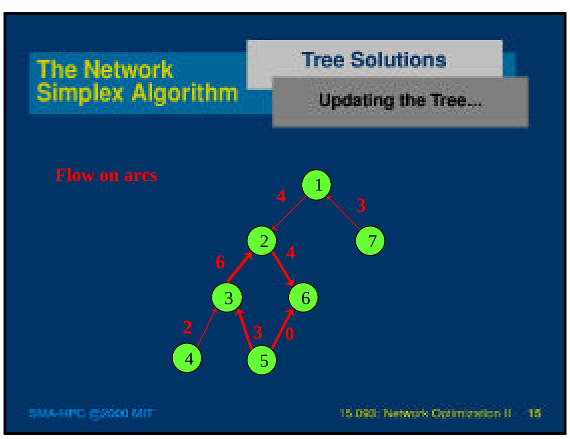

The Network Simplex Algorithm
Tree Solutions
Computing Node Potentials...
What is the potential for node 4?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 14

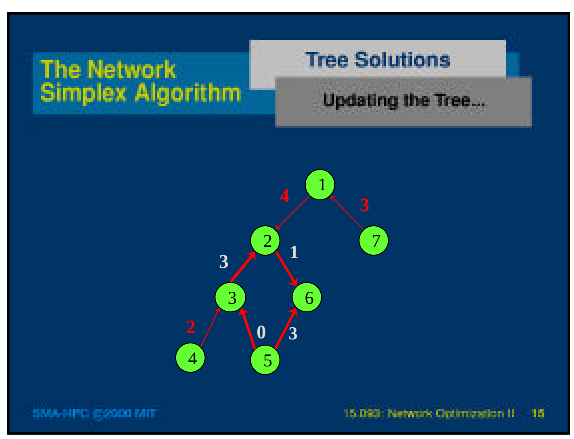

The Network Simplex Algorithm
Tree Solutions
Computing Node Potentials...
What is the potential for node 5?
SMA-HPC ©2000 MIT
15.093: Network Optimization II 14


The Network Simplex Algorithm
Tree Solutions
Computing Node Potentials...
These are the node potentials associated with this tree. They do not depend on arc flows, nor on costs of non-tree arcs.
SMA-HPC ©2000 MIT
15.093: Network Optimization II 14


The Network Simplex Algorithm
Tree Solutions
Updating the Tree...
Node potentials
Original costs
SMA-HPC ©2000 MIT
15.093: Network Optimization II 15


The Network Simplex Algorithm
Tree Solutions
Updating the Tree...
Flow on arcs
Reduced costs
SMA-HPC ©2000 MIT
15.093: Network Optimization II 15

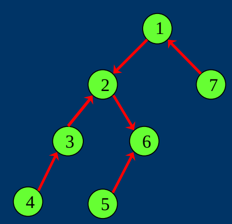

The Network Simplex Algorithm
Tree Solutions
Updating the Tree...
Flow on arcs
SMA-HPC ©2000 MIT
15.093: Network Optimization II 16


The Network Simplex Algorithm
Tree Solutions
Updating the Tree...
SMA-HPC ©2000 MIT
15.093: Network Optimization II 16

The Network Simplex Algorithm

Tree Solutions

Updating the Tree...



The Network Simplex Algorithm

Overview of the Algorithm

1. Determine an initial feasible tree T. Compute flow x and node potentials p associated with T.
2. Calculate $\bar{c}_{ij} = c_{ij} - p_i + p_j$ for $(i, j) \notin T$.
   - If $\bar{c} \geq \mathbf{0}$, x optimal; stop.
   - Select (i, j) with $\bar{c}_{ij} < 0$.
3. Add (i, j) to T creating a unique cycle C. Send a maximum flow around C while maintaining feasibility. Suppose the exiting arc is (k, ℓ).
4. $T := (T \setminus (k, \ell)) \cup (i, j)$.

Min-Cost Flow

Integrality

Our reasoning has two important and far-reaching implications:

- There always exists an integer optimal flow (if node balances b_i are integer).
- There always exist optimal integer node potentials (if arc costs c_{ij} are integer).

The Network Simplex Algorithm

An Animation

The Network Simplex Algorithm

The Algebraic View

- Bases and trees.
- Dual variables and node potentials.
- Changing bases and updating trees.
- Optimality testing.

The Network Simplex Algorithm

The Algebraic View

Bases vs. Trees...

The constraint matrix A of the min-cost flow problem is the node-arc incidence matrix of the underlying network.

	$(1,2)$	$(2,6)$	$(3,2)$	$(4,3)$	$(4,5)$	$(5,3)$	$(5,6)$	$(6,7)$	$(7,1)$
1	$+1$	0	0	0	0	0	0	0	-1
2	-1	$+1$	-1	0	0	0	0	0	0
3	0	0	$+1$	-1	0	-1	0	0	0
4	0	0	0	$+1$	$+1$	0	0	0	0
5	0	0	0	0	-1	$+1$	$+1$	0	0
6	0	-1	0	0	0	0	-1	$+1$	0
7	0	0	0	0	0	0	0	-1	$+1$

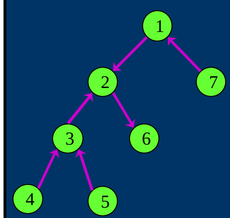
The rows of A are linearly dependent.

The Network Simplex Algorithm

The Algebraic View

...Bases vs. Trees...

Let B be the submatrix corresponding to the tree



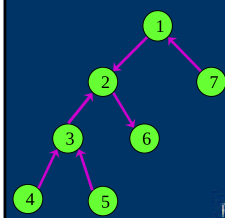
	(1,2)	(2,6)	(3,2)	(4,3)	(5,3)	(7,1)
1	+1	0	0	0	0	−1
2	−1	+1	−1	0	0	0
3	0	0	+1	−1	−1	0
4	0	0	0	+1	0	0
5	0	0	0	0	+1	0
6	0	−1	0	0	0	0
7	0	0	0	0	0	+1

The Network Simplex Algorithm

The Algebraic View

...Bases vs. Trees...

Let B be the submatrix corresponding to the tree



	(1,2)	(2,6)	(3,2)	(4,3)	(5,3)	(7,1)
4	0	0	0	+1	0	0
5	0	0	0	0	+1	0
6	0	−1	0	0	0	0
7	0	0	0	0	0	+1
3	0	0	+1	−1	−1	0
2	−1	+1	−1	0	0	0
1	+1	0	0	0	0	−1

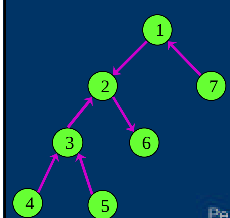
Permuting Rows

The Network Simplex Algorithm

The Algebraic View

...Bases vs. Trees...

Let B be the submatrix corresponding to the tree



	(4,3)	(5,3)	(2,6)	(7,1)	(3,2)	(1,2)
4	+1	0	0	0	0	0
5	0	+1	0	0	0	0
6	0	0	−1	0	0	0
7	0	0	0	+1	0	0
3	−1	−1	0	0	+1	0
2	0	0	+1	0	−1	−1
1	0	0	0	−1	0	+1

Permuting Columns

The Network Simplex Algorithm

The Algebraic View

...Bases vs. Trees...

Corollary 1

(a) *The matrix A has rank $n - 1$.*

(b) *Every tree solution is a basic solution.*

The Network Simplex Algorithm

The Algebraic View

...Bases vs. Trees...

Theorem 3 *Every tree defines a basis and, conversely, every basis defines a tree.*

Suppose the graph defined by a basis contains a cycle $1 - 2 - 3 - 4 - 5 - 6$:

	(1,2)	(2,3)	(4,3)	(5,4)	(5,6)	(1,6)
1	+1	0	0	0	0	+1
2	−1	+1	0	0	0	0
3	0	−1	−1	0	0	0
4	0	0	+1	−1	0	0
5	0	0	0	+1	+1	0
6	0	0	0	0	−1	−1

The Network Simplex Algorithm

The Algebraic View

Dual Variables vs. Node Potentials...

Remember, the simplex algorithm computes the dual variables p as the solution to $p' B = c_B'$

$$(p_4, p_5, p_6, p_7, p_3, p_2) \begin{pmatrix} +1 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 \\ -1 & -1 & 0 & 0 & +1 & 0 \\ 0 & 0 & +1 & 0 & -1 & -1 \end{pmatrix}$$

$$= (c_{43}, c_{53}, c_{26}, c_{71}, c_{32}, c_{12})$$

Hence, $p_2 = -c_{12}$, $p_3 = c_{32} - p_2$, $p_7 = c_{71}, \ldots$

The Network Simplex Algorithm

The Algebraic View

Optimality Testing...

Remember, the simplex algorithm computes the reduced costs $\bar{c}$ as $\bar{c}_{ij} = c_{ij} - p' A_{ij}$.

	(1,2)	(2,6)	(3,2)	(4,3)	(4,5)	(5,3)	(5,6)	(6,7)	(7,1)
1	+1	0	0	0	0	0	0	0	−1
2	−1	+1	−1	0	0	0	0	0	0
3	0	0	+1	−1	0	−1	0	0	0
4	0	0	0	+1	+1	0	0	0	0
5	0	0	0	0	−1	+1	+1	0	0
6	0	−1	0	0	0	0	−1	+1	0
7	0	0	0	0	0	0	0	−1	+1

Therefore, $\bar{c}_{ij} = c_{ij} - p_i + p_j$.

The Network Simplex Algorithm

Summary

- The network simplex algorithm is extremely fast in practice.
- Relying on network data structures, rather than matrix algebra, causes the speedups. It leads to simple rules for selecting the entering and exiting variables.
- Running time per pivot:
  - arcs scanned to identify an entering arc,
  - arcs scanned of the basic cycle,
  - nodes of the subtree.
- A good pivot rule can dramatically reduce running time in practice.