Integer Exact Network Synthesis Problem

S.N. Kabadi, J. Yan, D. Du, K.P.K. Nair

Faculty of Business Administration,

University of New Brunswick, Fredericton, N.B., Canada E3B5A3

J. Yan also in School of Mathematics and Systems Sciences,

Shandong University, Jinan 250100, P. R. China

Introduction

- 1. Introduction

Introduction

- 1. Introduction
- 2. Main algorithm

  Subroutine 1–Algorithm Tree Finding

  Subroutine 2–Algorithm Cut-tree Realization

Introduction

- 1. Introduction
- 2. Main algorithm

  Subroutine 1–Algorithm Tree Finding

  Subroutine 2–Algorithm Cut-tree Realization
- 3. An example

Introduction

- Let $R = (r_{ij})_{n\times n}$ be a symmetric, non-negative matrix of minimum flow requirements between all pairs of distinct nodes in $V = \{1, 2, \ldots, n\}$, where $r_{ii} = 0\ i \in \{1, \ldots, n\}$.

Introduction

- Let $R = (r_{ij})_{n \times n}$ be a symmetric, non-negative matrix of minimum flow requirements between all pairs of distinct nodes in $V = \{1, 2, \ldots, n\}$, where $r_{ii} = 0 \; i \in \{1, \ldots, n\}$.
- A realization of R: a network $G = [V, E, u]$ on node set V with edge set E, and non-negative edge capacities $\{u_e : e \in E\}$ if and only if for every pair $\{i, j\}$ of distinct nodes in V, the value of maximum flow between i and j in G is at least r_{ij}.

Introduction

- Let $R = (r_{ij})_{n \times n}$ be a symmetric, non-negative matrix of minimum flow requirements between all pairs of distinct nodes in $V = \{1, 2, \dots, n\}$, where $r_{ii} = 0 \; i \in \{1, \dots, n\}$.
- A realization of R: a network $G = [V, E, u]$ on node set V with edge set E, and non-negative edge capacities $\{u_e : e \in E\}$ if and only if for every pair $\{i, j\}$ of distinct nodes in V, the value of maximum flow between i and j in G is at least r_{ij}.
- G is an exact realization of R if and only if for every pair $\{i, j\}$ of distinct nodes, the value of maximum flow between i and j in G equals exactly r_{ij}.

Introduction

- Let $R = (r_{ij})_{n \times n}$ be a symmetric, non-negative matrix of minimum flow requirements between all pairs of distinct nodes in $V = \{1, 2, \dots, n\}$, where $r_{ii} = 0 \; i \in \{1, \dots, n\}$.
- A realization of R: a network $G = [V, E, u]$ on node set V with edge set E, and non-negative edge capacities $\{u_e : e \in E\}$ if and only if for every pair $\{i, j\}$ of distinct nodes in V, the value of maximum flow between i and j in G is at least r_{ij}.
- G is an exact realization of R if and only if for every pair $\{i, j\}$ of distinct nodes, the value of maximum flow between i and j in G equals exactly r_{ij}.
- If R has an exact realization, then it is exactly realizable.

Introduction

- **Four Problems.**

Introduction

- **Four Problems.**
- **NS** Problem–Network Synthesis Problem

  ENS Problem–Exact Network Synthesis Problem

  INS Problem–Integer Network Synthesis Problem

  IENS Problem–Integer Exact Network Synthesis Problem

  Each of the four problems is to construct a required realization of R with minimum sum of edge capacities.

Introduction

- Let f_{ij} denote the maximum flow value or minimum cut value for each pair nodes i, j.

NS Problem

$$\min \sum_{e \in E} u_e$$

s. t. $$f_{ij} \geq r_{ij}$$

$u_e \geq 0$ is a real number

ENS Problem

$$\min \sum_{e \in E} u_e$$

s. t. $$f_{ij} = r_{ij}$$

$u_e \geq 0$ is a real numbe

Introduction

- Let f_{ij} denote the maximum flow value or minimum cut value for each pair nodes i, j.

NS Problem

min $\sum_{e \in E} u_e$

s. t. $f_{ij} \geq r_{ij}$

$u_e \geq 0$ is a real number

ENS Problem

min $\sum_{e \in E} u_e$

s. t. $f_{ij} = r_{ij}$

$u_e \geq 0$ is a real numbe

- **NS** problem : Gomory-Hu(1961), Gusfield(1983), Tardos (1986), Talluri(1996), solved the problem in strongly polynomial time, respectively.

  ENS problem. A modification of the Gomory-Hu algorithm produces an optimal solution for **ENS** problem whenever it has a feasible solution.

Introduction

- **INS** Problem

  min $\sum_{e\in E} u_e$

  s. t. $f_{ij} \geq r_{ij}$

  $u_e \geq 0$ is an integer

  IENS Problem

  min $\sum_{e\in E} u_e$

  s. t. $f_{ij} = r_{ij}$

  $u_e \geq 0$ is an integer

Introduction

	INS Problem		**IENS** Problem
	min $\sum_{e\in E} u_e$		min $\sum_{e\in E} u_e$
s. t.	$f_{ij} \geq r_{ij}$	s. t.	$f_{ij} = r_{ij}$
	$u_e \geq 0$ is an integer		$u_e \geq 0$ is an integer

- **INS** problem. Chou and Frank(1970), Sridhar and Chandrasekaran(1992), presented combinatorial algorithms of computational complexity $O(n^2)$.

Introduction

	INS Problem		**IENS** Problem
	min $\sum_{e\in E} u_e$		min $\sum_{e\in E} u_e$
s. t.	$f_{ij} \geq r_{ij}$	s. t.	$f_{ij} = r_{ij}$
	$u_e \geq 0$ is an integer		$u_e \geq 0$ is an integer

- **INS** problem. Chou and Frank(1970), Sridhar and Chandrasekaran(1992), presented combinatorial algorithms of computational complexity $O(n^2)$.
- **IENS** Problem. Chou, Frank claim to give an algorithm for the problem. However, Schrijver gives a counter-example to their claim(Schrijver, A. 2003. *Combinatorial optimization: Polyhedra and efficiency*, Algorithms and Combinatorics 24, Springer-Verlag, New York.).

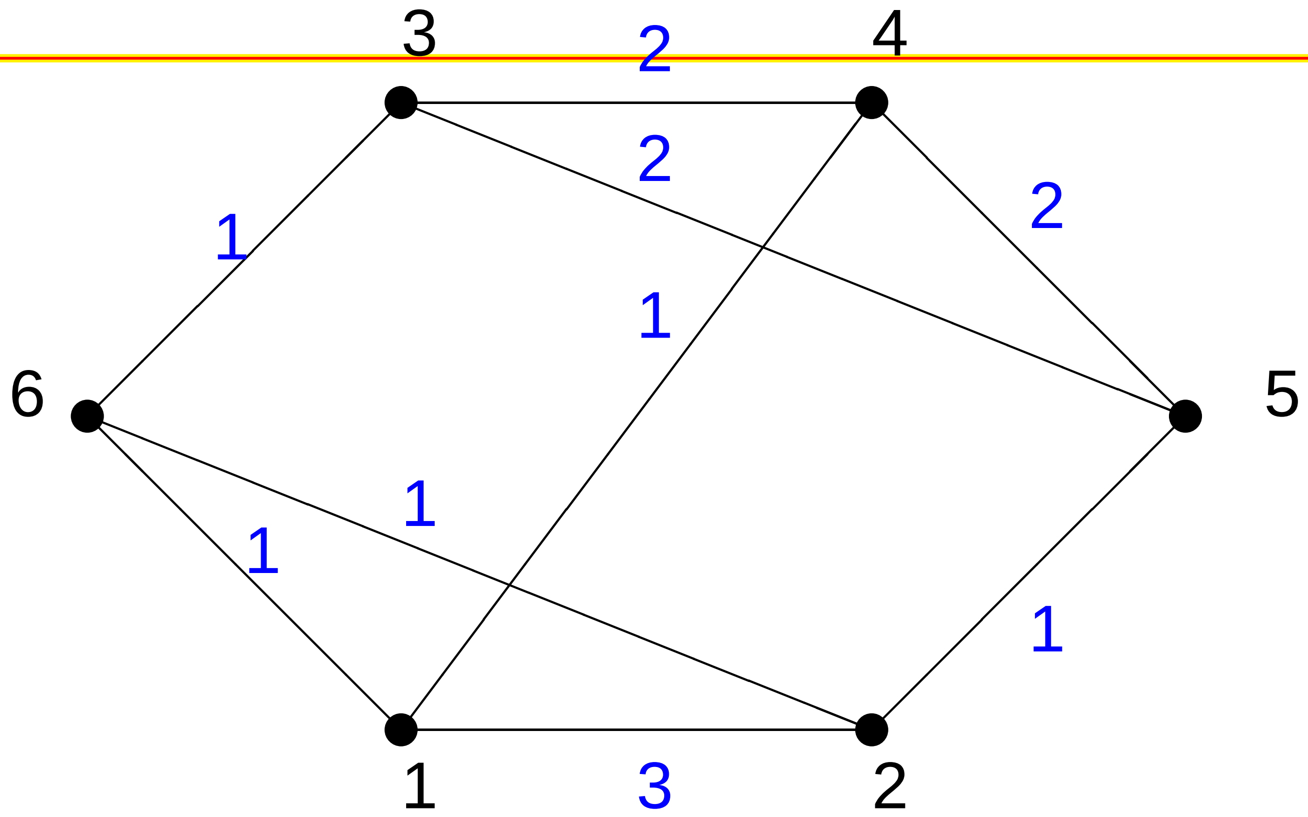
Introduction: An counter-example

- **An counter-example.** $R = \begin{pmatrix} 0 & 5 & 3 & 3 & 3 & 3 \\ 5 & 0 & 3 & 3 & 3 & 3 \\ 3 & 3 & 0 & 5 & 5 & 3 \\ 3 & 3 & 5 & 0 & 5 & 3 \\ 3 & 3 & 5 & 5 & 0 & 3 \\ 3 & 3 & 3 & 3 & 3 & 0 \end{pmatrix}$

Introduction: An counter-example

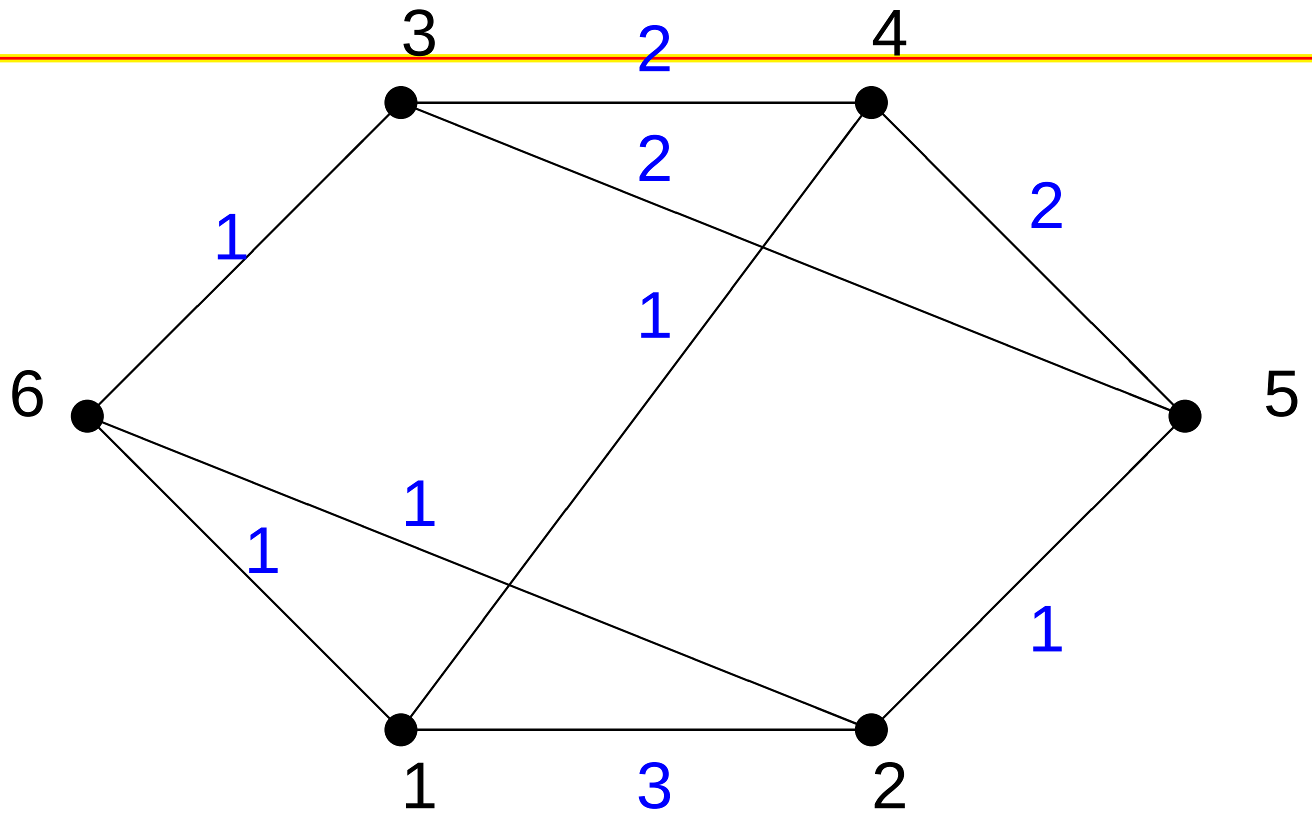
- **An counter-example.** $R = \begin{pmatrix} 0 & 5 & 3 & 3 & 3 & 3 \\ 5 & 0 & 3 & 3 & 3 & 3 \\ 3 & 3 & 0 & 5 & 5 & 3 \\ 3 & 3 & 5 & 0 & 5 & 3 \\ 3 & 3 & 5 & 5 & 0 & 3 \\ 3 & 3 & 3 & 3 & 3 & 0 \end{pmatrix}$
- Chou-Frank Algorithm gives the total capacities $\sum_{e \in E} u_e = 15$, but there is an exact realization with $\sum_{e \in E} u_e = 14$.

Introduction: known results



-

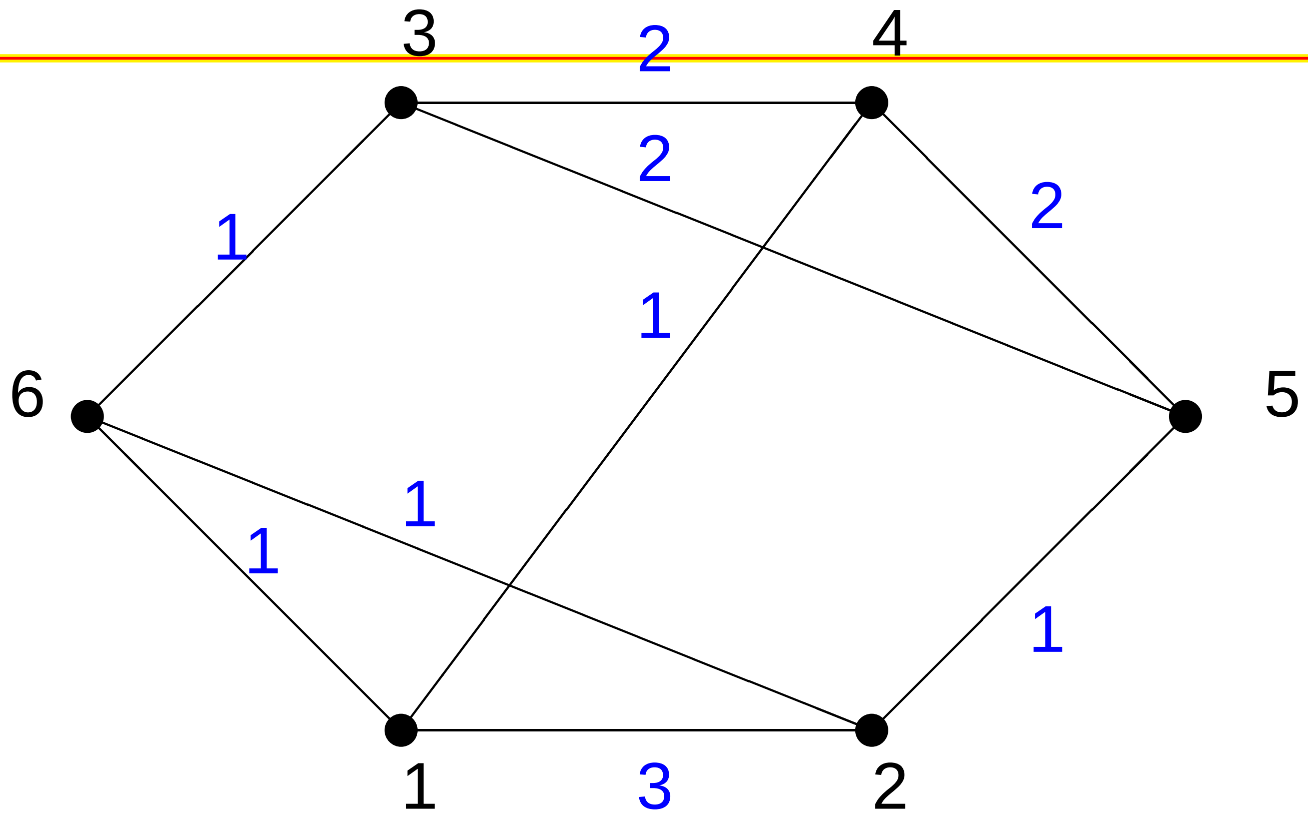
Introduction: known results



-

- A realization $G = [V, E, u]$ of R with $\sum_{e \in E} u_e = 14$.

Introduction: known results



- A realization $G = [V, E, u]$ of R with $\sum_{e \in E} u_e = 14$.
- We give an $O(n^2)$ combinatorial algorithm for the **IENS** problem.

2. Main algorithm-definition and theorems

- Definition 2.1 Two networks $G^1 = [V, E^1, u^1]$ and $G^2 = [V, E^2, u^2]$ on the same node set V and with edge capacities $\{u_e^1 : e \in E^1\}$ and $\{u_e^2 : e \in E^2\}$ are said to be flow-equivalent if for any pair $\{i, j\}$ of distinct nodes in V, the maximum flow values between i and j in the two networks are the same.

2. Main algorithm-definition and theorems

- Definition 2.1 Two networks $G^1 = [V, E^1, u^1]$ and $G^2 = [V, E^2, u^2]$ on the same node set V and with edge capacities $\{u_e^1 : e \in E^1\}$ and $\{u_e^2 : e \in E^2\}$ are said to be flow-equivalent if for any pair $\{i, j\}$ of distinct nodes in V, the maximum flow values between i and j in the two networks are the same.
- **Theorem 2.1**[9, 21]There exists an $O(n^2)$ algorithm to test if a given matrix is exactly realizable.

2. Main algorithm-definition and theorems

- **Theorem 2.2**[9] For a symmetric, $n \times n$, non-negative matrix R, let G^R be a complete network on node set $V = \{1, 2, \ldots, n\}$ with capacity of each edge (i, j) equal to r_{ij}. Then
  (I) Every maximum weight spanning tree of G^R is a realization of R.
  (II) The following three statements are equivalent.
  (IIa) R is exactly realizable.
  (IIb) $r_{ij} \geq \min\{r_{ik}, r_{kj}\}$ for all distinct $i, j, k \in V$.
  (IIc) Every maximum weight spanning tree $T = [V, E]$ of G^R with edge capacities
  $u_e = r_{ij} \; \forall e = (i, j) \in E$ is an exact realization of R.

2. Main algorithm

- An Algorithm for the IENS Problem

2. Main algorithm

- An Algorithm for the IENS Problem
- Input. A symmetric, integer, non-negative matrix $R = (r_{ij})_{n\times n}$.

2. Main algorithm

- An Algorithm for the IENS Problem
- Input. A symmetric, integer, non-negative matrix $R = (r_{ij})_{n\times n}$.
- Output. A network $G^* = [V, E^*, u^*]$, that is an optimal solution to the instance of the **IENS** problem.

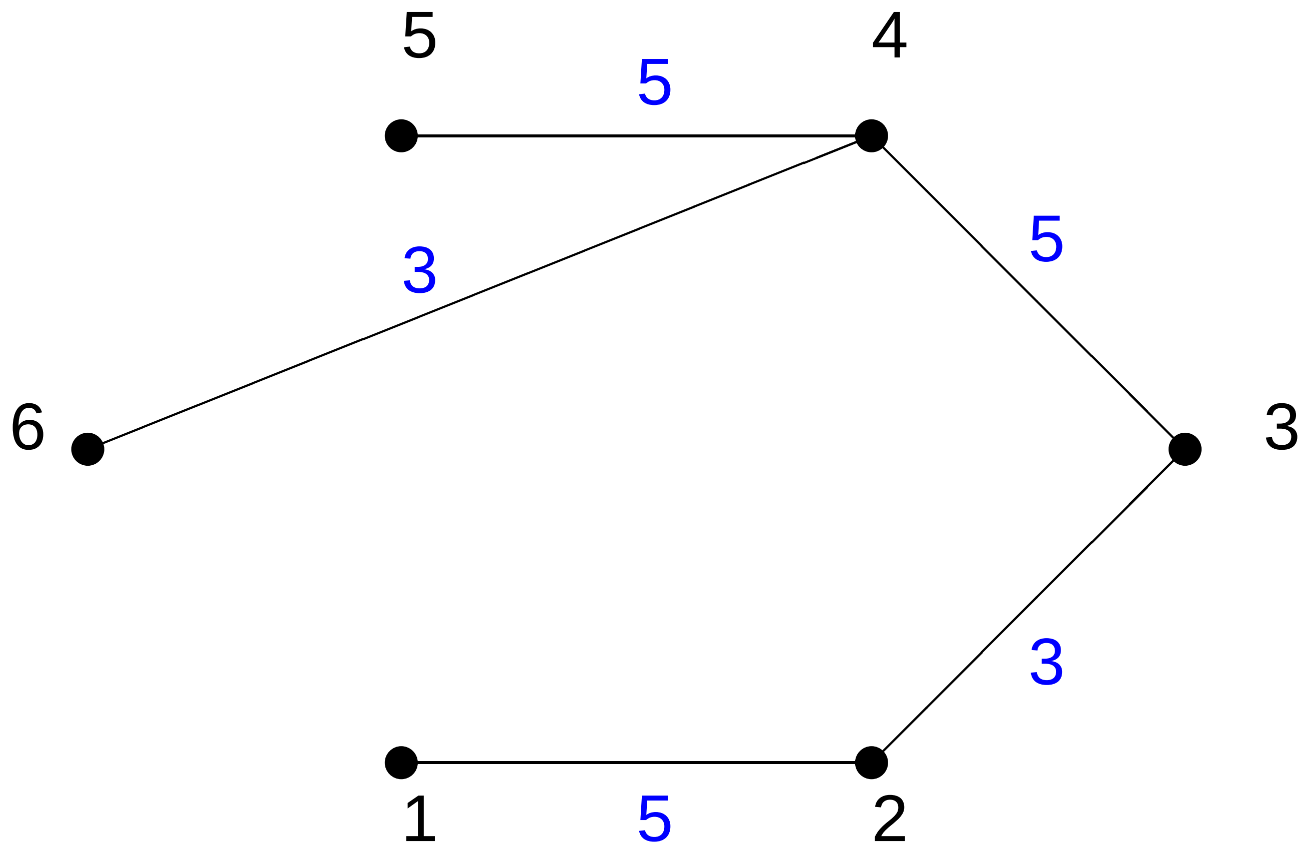
2. Main algorithm

- An Algorithm for the IENS Problem
- Input. A symmetric, integer, non-negative matrix $R = (r_{ij})_{n \times n}$.
- Output. A network $G^* = [V, E^*, u^*]$, that is an optimal solution to the instance of the **IENS** problem.
- Step 0. Find a maximum weight spanning tree $T = [V, E, u]$ in G^R, (the complete graph on node set V with edge weight $u_e = r_{ij} \; \forall \;\; e = (i, j), i \neq j$.

2. Main algorithm-an example

- **An example.** $R = \begin{pmatrix} 0 & 5 & 3 & 3 & 3 & 3 \\ 5 & 0 & 3 & 3 & 3 & 3 \\ 3 & 3 & 0 & 5 & 5 & 3 \\ 3 & 3 & 5 & 0 & 5 & 3 \\ 3 & 3 & 5 & 5 & 0 & 3 \\ 3 & 3 & 3 & 3 & 3 & 0 \end{pmatrix}$

3. Main algorithm-an example



- A maximum weight spanning tree $T = [V, E, u]$ in G^R.

2. Main algorithm

- If T does not exactly realizes R, then conclude that "R is not exactly realizable" and stop.
  Otherwise, let $\bar{E} = \{e : e \in E, u_e \leq 1\}$, $F = [V, E - \bar{E}, u]$.
  Let $T^1, T^2, \cdots, T^k$ be the connected components of F with node sets $V^1, \ldots, V^k$, respectively. Set $i = 1$.

2. Main algorithm

- If T does not exactly realizes R, then conclude that "R is not exactly realizable" and stop.
  Otherwise, let $\bar{E} = \{e : e \in E, u_e \leq 1\}$, $F = [V, E - \bar{E}, u]$. Let $T^1, T^2, \cdots, T^k$ be the connected components of F with node sets $V^1, \ldots, V^k$, respectively. Set $i = 1$.
- Step 1. Find an optimal solution $T^{*i} = [V^i, E^i, u^i]$ with input T^i, using **Algorithm Tree Finding**.

2. Main algorithm

- If T does not exactly realizes R, then conclude that "R is not exactly realizable" and stop.
  Otherwise, let $\bar{E} = \{e : e \in E, u_e \leq 1\}$, $F = [V, E - \bar{E}, u]$.
  Let $T^1, T^2, \cdots, T^k$ be the connected components of F with node sets $V^1, \ldots, V^k$, respectively. Set $i = 1$.
- Step 1. Find an optimal solution $T^{*i} = [V^i, E^i, u^i]$ with input T^i, using **Algorithm Tree Finding**.
- Step 2. Find optimal solution $G^{*i} = [V^i, E^{*i}, u^{*i}]$ with input T^{*i}, using **Algorithm Cut-tree Realization**.

2. Main algorithm

- **Step 3.** If $i < k$, then increment $i = i + 1$ and go to Step 1. Else, construct network $G^* = [V, E^*, u^*]$ with $E^* = (\cup_{i=1}^{k} E^{*i}) \cup \bar{E}$ and

$$u_e^* = \begin{cases} u_e^{*i} & \text{if } e \in E^{*i} \text{ for some } i \in \{1, 2, \cdots, k\} \\ u_e & \text{if } e \in \bar{E} \end{cases}$$

Output the network G^* and stop.

2. Main algorithm

- Step 3. If $i < k$, then increment $i = i + 1$ and go to Step 1. Else, construct network $G^* = [V, E^*, u^*]$ with $E^* = (\cup_{i=1}^{k} E^{*i}) \cup \bar{E}$ and

$$u_e^* = \begin{cases} u_e^{*i} & \text{if } e \in E^{*i} \text{ for some } i \in \{1, 2, \cdots, k\} \\ u_e & \text{if } e \in \bar{E} \end{cases}$$

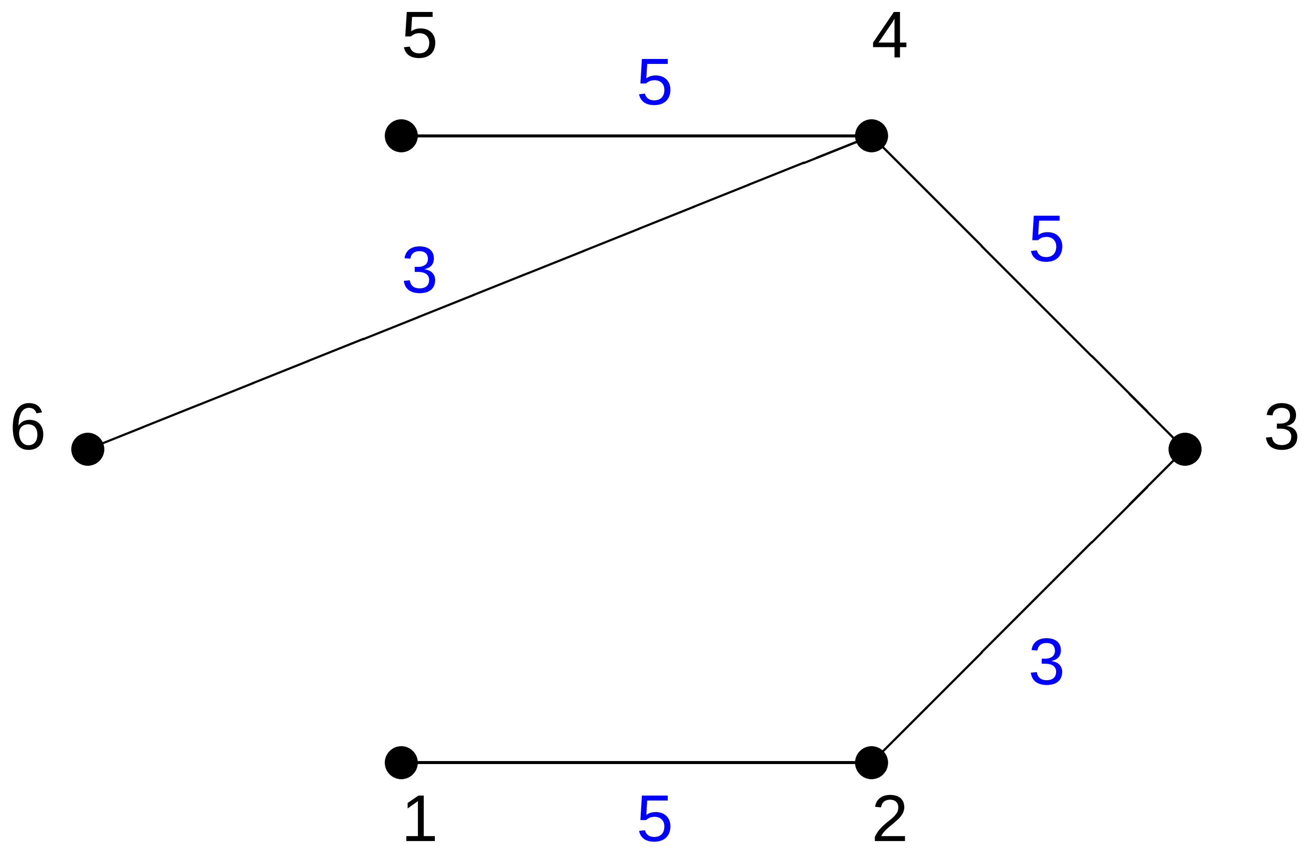
  Output the network G^* and stop.

- **Theorem 3.1** Main Algorithm produces an optimal network $G^* = [V, E^*, u^*]$ with optimal value $u^*[V] = \frac{\sum_{i \in V} \pi_i + |V_1^e| + |V_2^o|}{2}$ to the **IENS** problem in $O(n^2)$ time.

3. An example

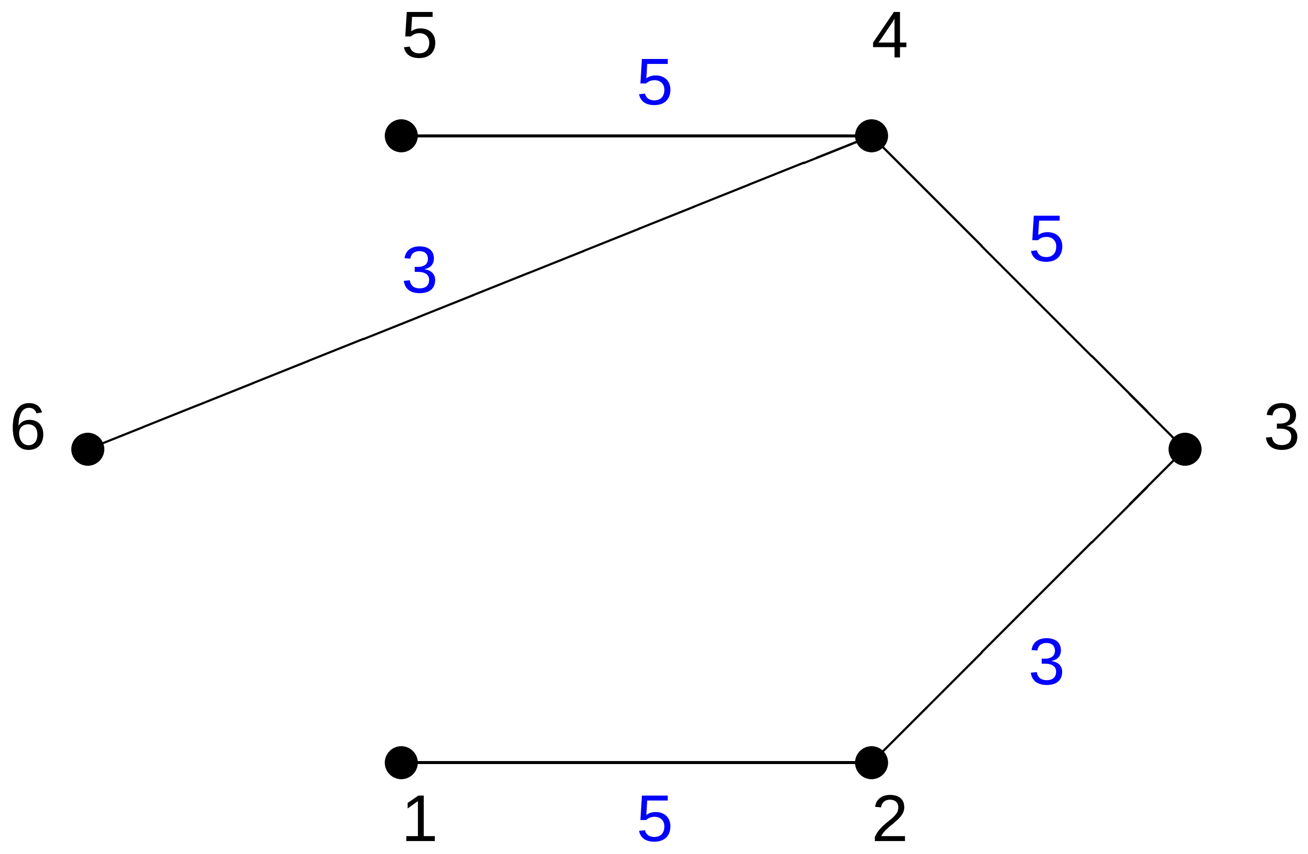
- **An axample.** $R = \begin{pmatrix} 0 & 5 & 3 & 3 & 3 & 3 \\ 5 & 0 & 3 & 3 & 3 & 3 \\ 3 & 3 & 0 & 5 & 5 & 3 \\ 3 & 3 & 5 & 0 & 5 & 3 \\ 3 & 3 & 5 & 5 & 0 & 3 \\ 3 & 3 & 3 & 3 & 3 & 0 \end{pmatrix}$

3. An example



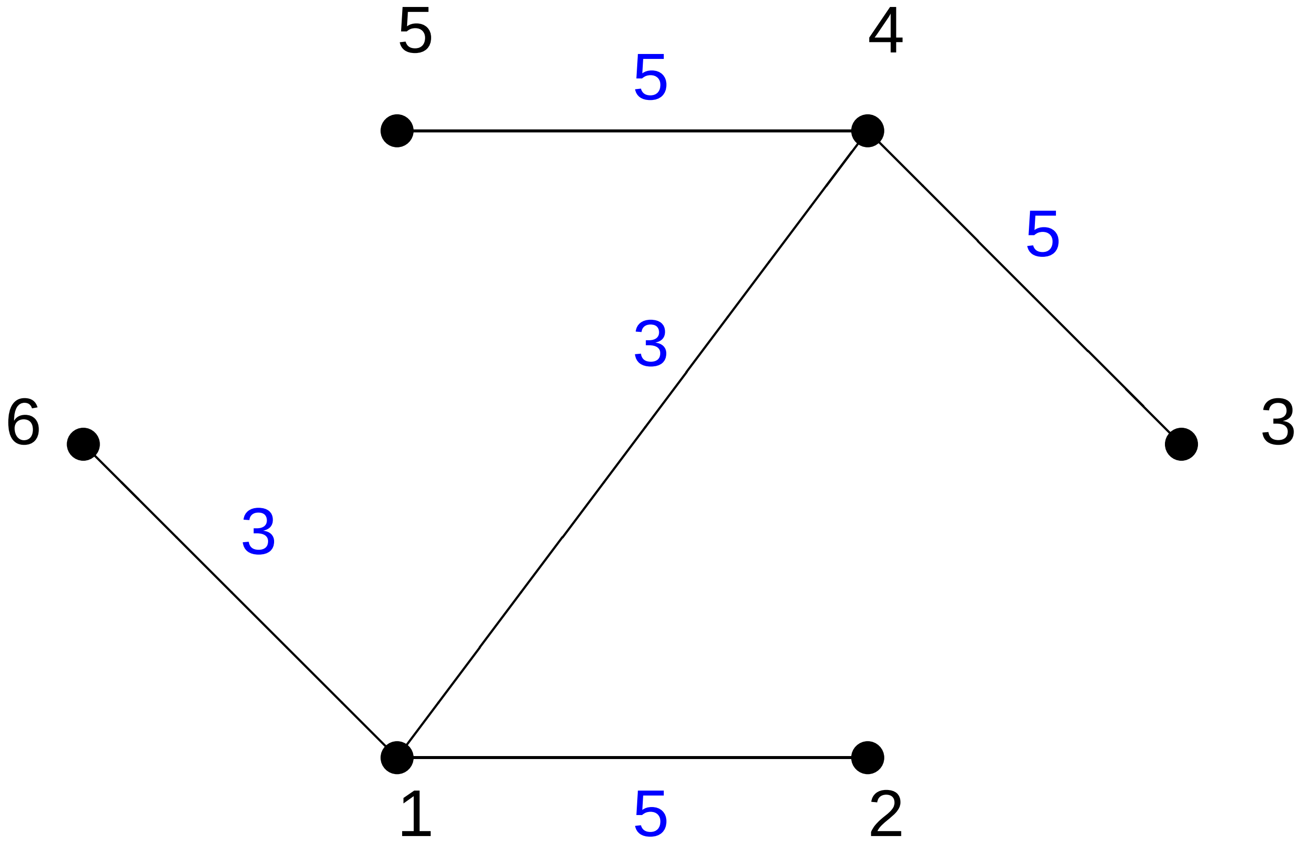
- A maximum weight spanning tree T

3. An example



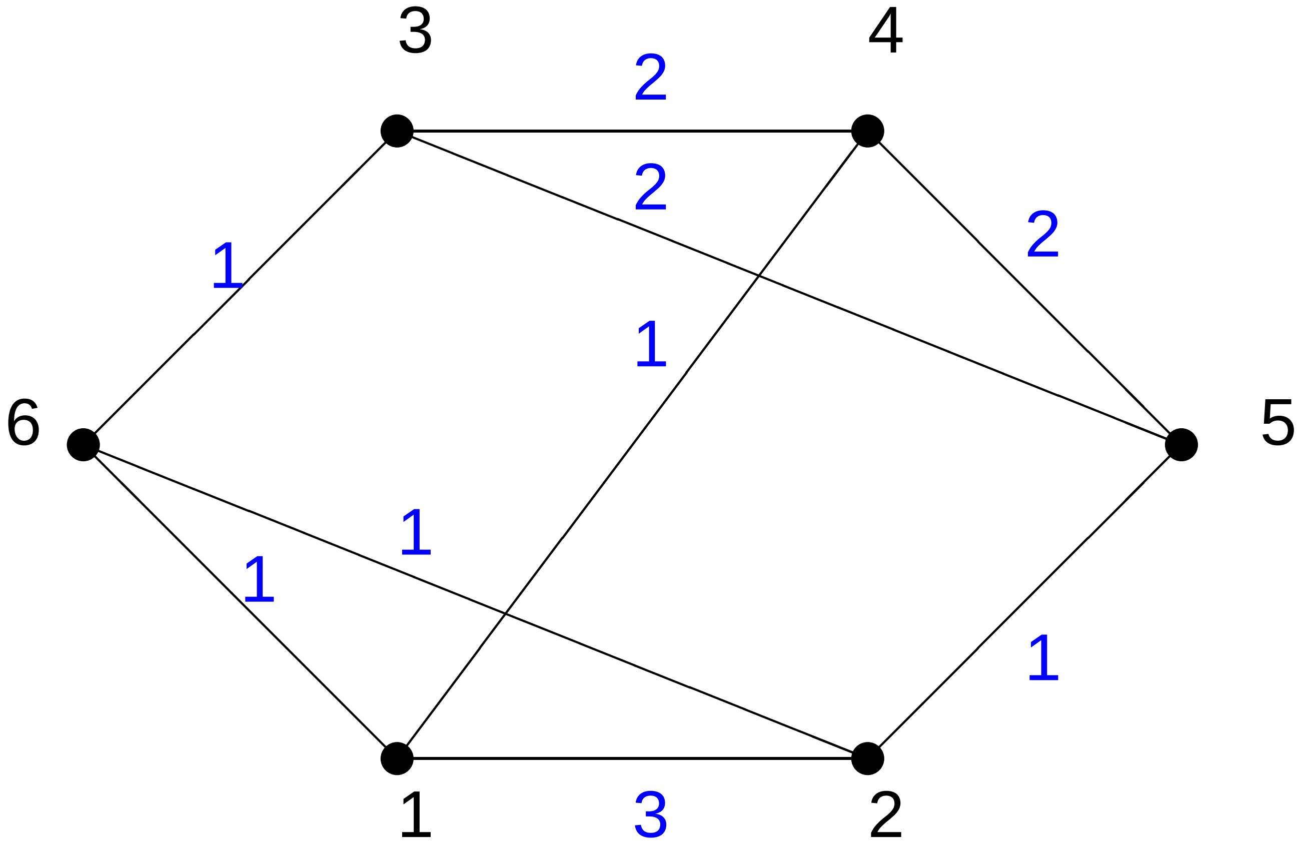
- A maximum weight spanning tree T
- R is exactly realizable.

3. An example



- An optimal tree T^* that is flow equivalent to the maximum weight spanning tree T, using **Algorithm Tree Finding.**

3. An example



- An optimal exactly realization G of R, constructed by **Algorithm Cut-tree Realization**

Thanks

Thank you for coming!

References

[1] Aneja, Y. P., R. Chandrasekaran, S. N. Kabadi, K. P. K. Nair. 2002. *Flows over edge-disjoint mixed multipaths and applications*. Working paper, Faculty of Administration, University of New Brunswick, Canada.

[2] Chandrasekaran, R., K. P. K. Nair, Y. P. Aneja and S. N. Kabadi. 2004. Multi-terminal multipath flows: Synthesis. *Discrete Applied Mathematics*, 143, 182-193.

[3] Chou, W. and H. Frank. 1970. Survivable Communication Networks and the Terminal Capacity Matrix. *IEEE Trans. on Circuit Theory*, CT-17, 2, 192-197.

[4] Cornuejols, G., D. Naddef and W. R. Pulleyblank. 1983. Halin graphs and the traveling salesman problem. *Mathematical Programming*, 26, 287-94. CT-17, 2, 192-197.

[5] Ford, L. R. and D. R. Fulkerson. 1962. *Flows in networks*, Princeton University Press, Princeton, New Jersey.

[6] Frank, A. 1994. Connectivity augmentation problems in network design. *Mathematical Programming : State of the Art, edited by J. Birge and K. G. Murty*, The University of Michigan Press, 34-63.

[7] Gibbens, R. J. and F. P. Kelly. 1990. Dynamic routing in fully connected networks. *IMA Journal of Mathematical Control and Information*, 7, 77-111.

[8] Gusfield, D. 1983. Simple constructions for the multi-terminal network flow synthesis, *SIAM J. Comput.*, 12, 1, 157-165.

[9] Gomory, R. E. and T. C. Hu. 1961. Multi-terminal network flows, *Journal of SIAM*, 9, 551-570.

[10] Gomory, R. E. and T. C. Hu. 1961. An application of generalized linear programming to network flows. *Journal of SIAM*, 10, 260-283.

[11] Hassin, R. and A. Levin. 2001. Synthesis of 2-commodity flow networks. *Proceedings of IPCO*, LNCS 2081, 226-235.

[12] Hojati, M. 1995. The network synthesis problem in a cycle. *Operations Research Letters*, 17, 231-236.

[13] Kabadi, S. N. 2003. *Strongly polynomial algorithms for the continuous and integer versions of the network augmentation problem in a cycle*. Working paper, Faculty of Administration, University of New Brunswick, Canada.

[14] Kabadi, S. N., R. Chandrasekaran and K. P. K. Nair. 2003. *2-commodity integer network synthesis problem*. Working paper, Faculty of Administration, University of New Brunswick, Canada.

[15] Kabadi, S. N., R. Chandrasekaran and K. P. K. Nair. 2005. Multiroute flows: Cut-trees and realizability. *Discrete Optimization* (to appear).

[16] Kabadi, S. N., R. Chandrasekaran, K.P.K. Nair and Y.P. Aneja. 2003. *Integer version of the multipath flow network synthesis problem*. Working paper, Faculty of Administration, University of New Brunswick.

[17] Kabadi, S. N., J. Kang, R. Chandrasekaran and K. P. K. Nair. 2003. *Hop-constrained network flows : Analysis and synthesis*. Working paper, Faculty of Administration, University of New Brunswick, Canada.

[18] Kabadi, S. N. and R. Sridhar. 1996. *Peeling algorithm for integral network synthesis*. Working paper, Faculty of Administration, University of New Brunswick, Canada.

[19] Kishimoto, W. 1996. A method for obtaining the maximum multi-route flows in a network. *Networks*, 27 , 279-291.

[20] Lawler, E. L. 1976. *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York.

[21] Mayeda, W. 1960. Terminal and Branch Capacity Matrices of a Communication Net, *IRE Transactions on Circuit Theory*, CT-7, 261-269.

[22] Punnen, A. 2005. *Private communication*.

[23] Schrijver, A. 2003. *Combinatorial optimization: Polyhedra and efficiency*, Algorithms and Combinatorics 24, Springer-Verlag, New York.

[24] Sridhar, R. and R. Chandrasekaran. 1992. Integer solution of synthesis of communication network. *Mathematics of Operations Research*, 17, 3, 581-585.

[25] Talluri, K. 1996. Network synthesis with few edges, *Networks*, 27, 109-115.

[26] Tardos, E. 1986. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34, 250-56.