EXACT ALGORITHMS FOR THE ASYMMETRIC TRAVELLING SALESMAN PROBLEM

Paolo Toth

DEIS, University of Bologna

CONTENTS

- Given a COMPLETE DIRECTED GRAPH $G = (V,A)$ with
  - $V = \{1, \ldots, n\}$ vertex set
  - $A = \{(i, j) : i \in V, j \in V\}$ arc set
  - c_{ij} = cost associated with arc $(i, j) \in A$ ($c_{ii} = \infty$, $i \in V$)
- Find a HAMILTONIAN CIRCUIT (Tour) whose global cost is minimum (Asymmetric Travelling Salesman Problem: ATSP).

不对称

Hamiltonian Circuit: circuit passing through each vertex of V exactly once.

A Hamiltonian circuit has n arcs.

- ATSP is $\mathcal{NP}$-Hard in the strong sense.

对称的

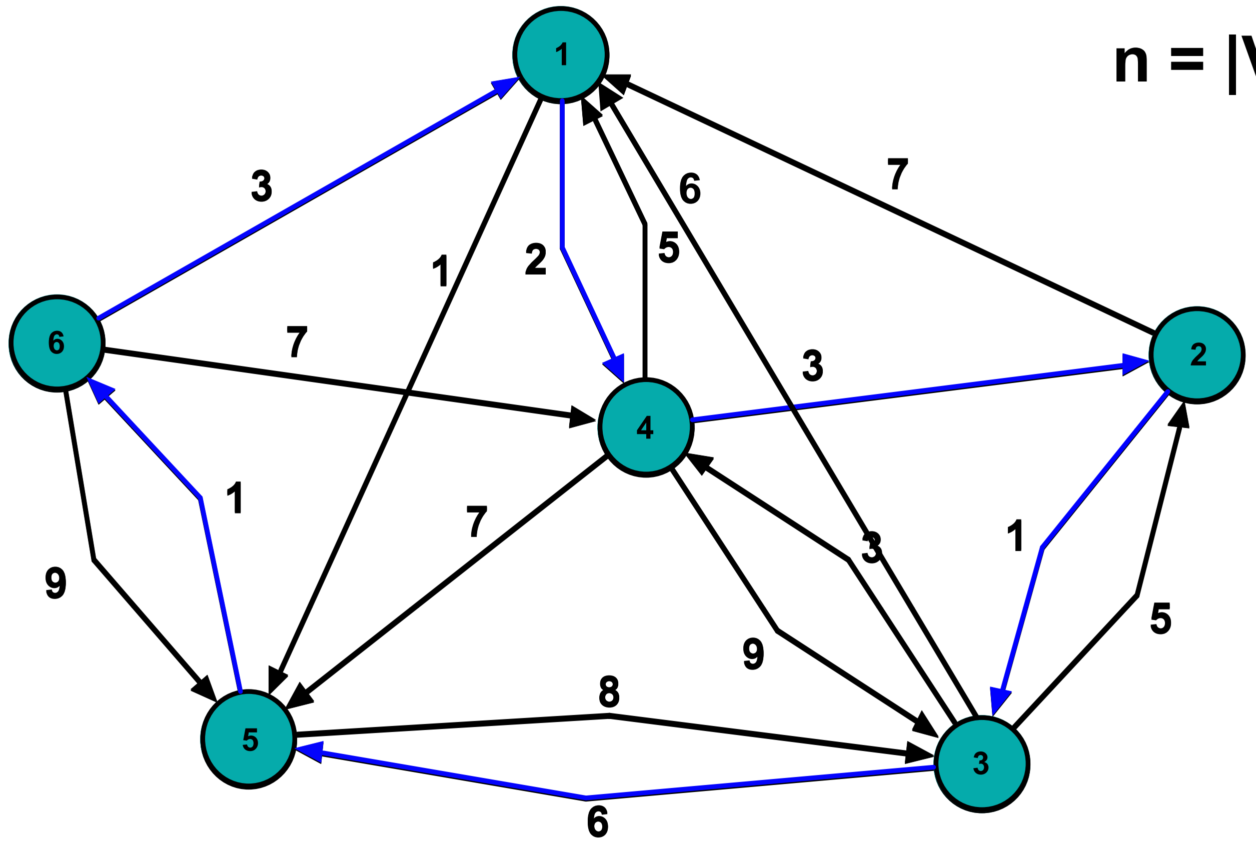
- If G is an undirected graph: **Symmetric TSP (STSP)**

  (special case of ATSP arising when $c_{ij} = c_{ji}$ for each $(i, j) \in A$)

- Any ATSP instance with n vertices can be transformed into an equivalent STSP instance with $2n$ nodes (Jonker-Volgenant, 1983; Junger-Reinelt-Rinaldi, 1995; Kumar-Li, 2007).

- If G = (V, A) is a sparse graph: $c_{ij} = \infty$ for each $(i, j) \notin A$.

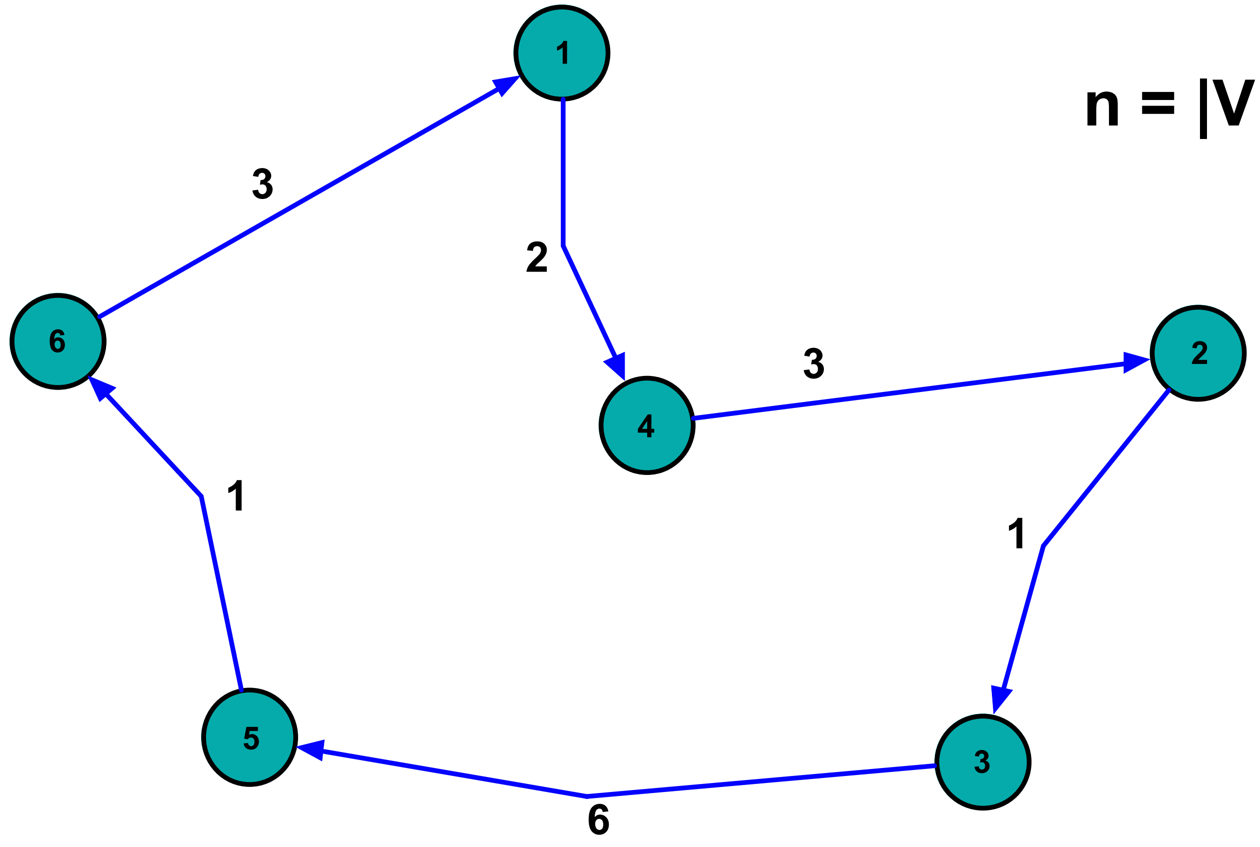
Example



n = |V| = 6

Optimal solution

Example



n = |V| = 6

Optimal solution

Optimal solution Cost = 2 + 3 + 1 + 6 +1 + 3 = 16

APPLICATIONS

* **Vehicle Routing** (sequencing the customers in each route in an urban area calls for the optimal solution of the ATSP corresponding to the depot and the customers in the route).

* **Scheduling** (optimal sequencing of jobs on a machine when the set-up costs depend on the sequence in which the jobs are processed).

* **Picking in an Inventory System** (sequence of movements of a crane to pick-up a set of items stored on shelves).

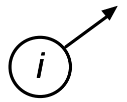
INTEGER LINEAR PROGRAMMING (ILP) FORMULATION

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is in the optimal tour} \\ 0 & \text{otherwise} \end{cases} \qquad i \in V,\ j \in V$$

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij}\, x_{ij}$$

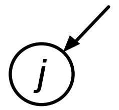
s.t.

out-degree constraints



$$\sum_{j \in V} x_{ij} = 1 \qquad i \in V$$

in-degree constraints

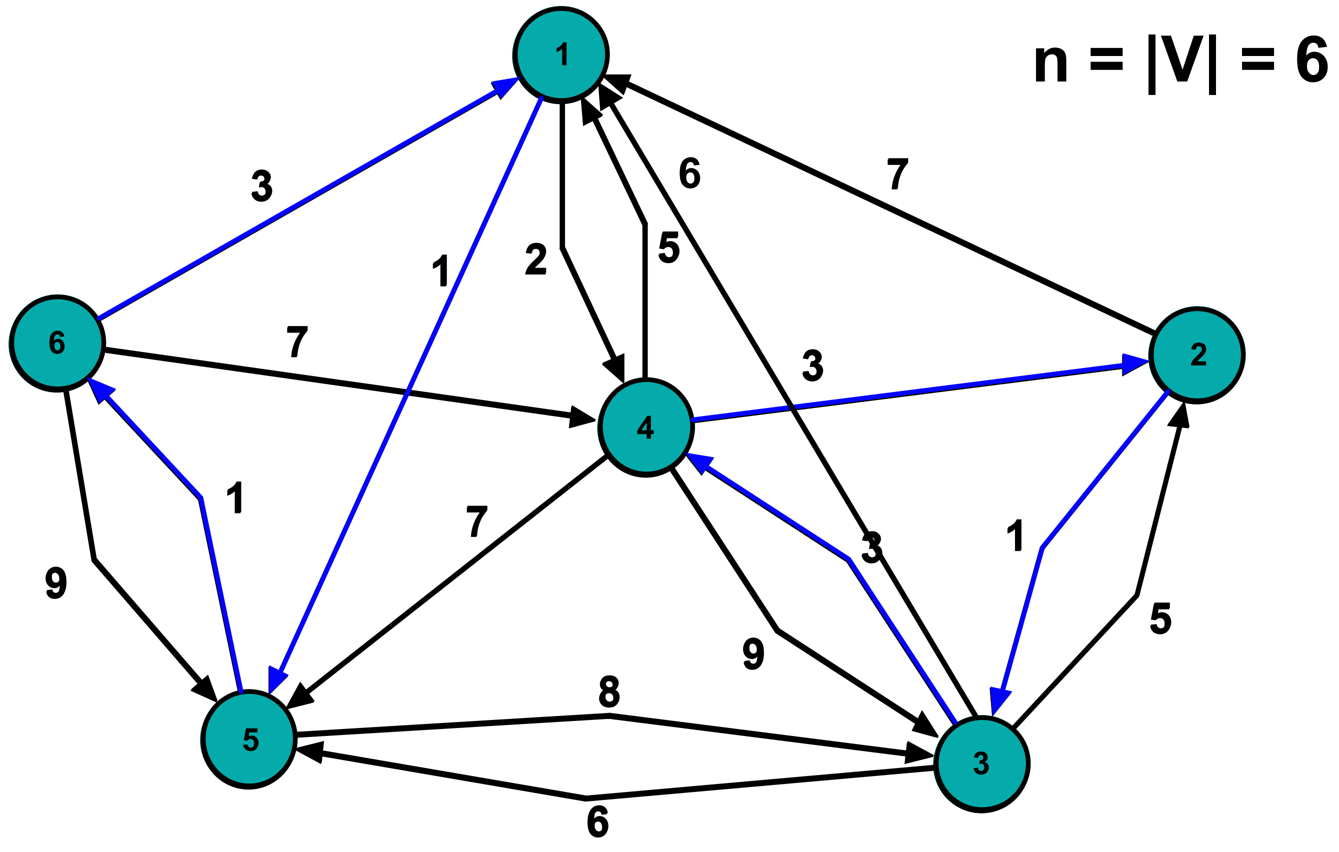


$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V$$

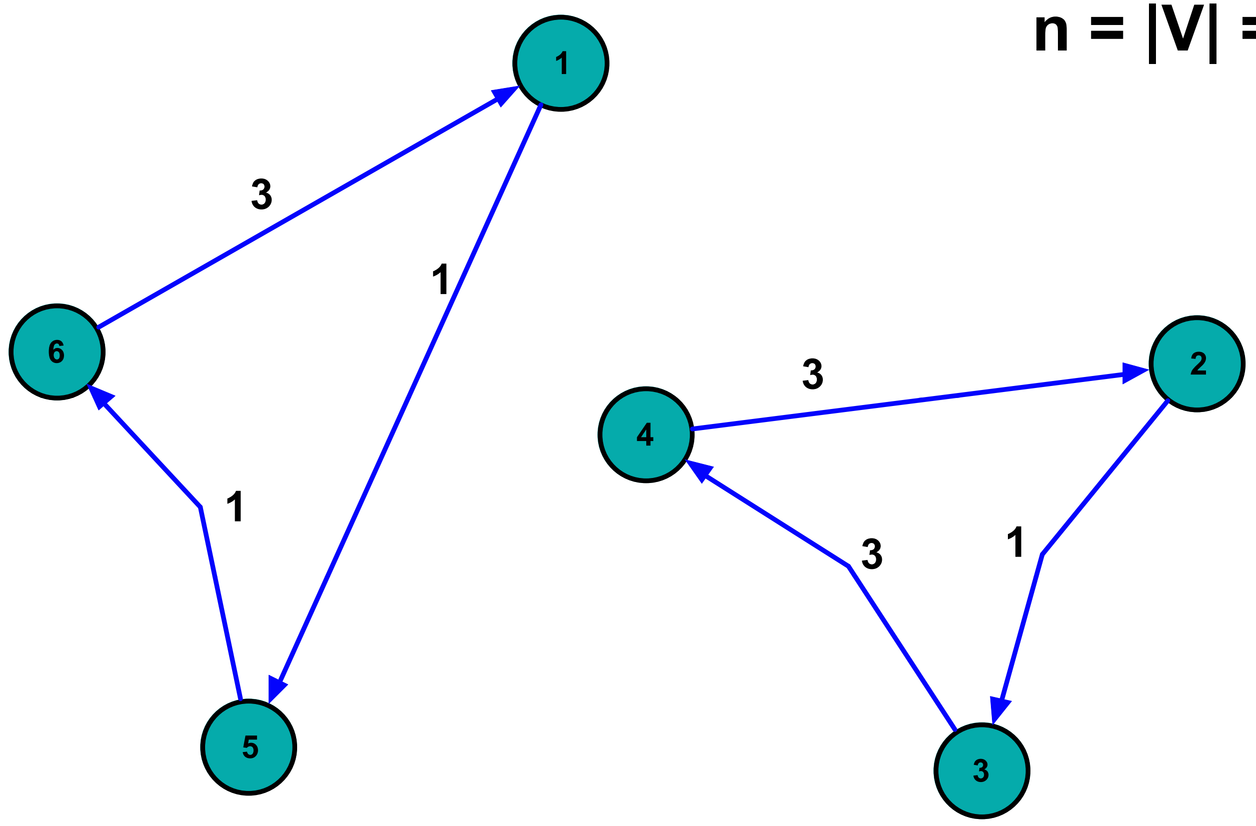
$$x_{ij} \in \{0, 1\} \qquad i \in V,\ j \in V$$

?

Example: only degree constraints imposed



Example: only degree constraints imposed



n = |V| = 6

Infeasible solution: two partial tours (subtours)

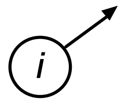
INTEGER LINEAR PROGRAMMING (ILP) FORMULATION (Dantzig, Fulkerson, Johnson, Oper. Res. 1954)

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is in the optimal tour} \\ 0 & \text{otherwise} \end{cases} \qquad i \in V,\ j \in V$$

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij}\, x_{ij}$$

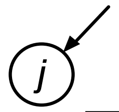
s.t.

out-degree constraints



$$\sum_{j \in V} x_{ij} = 1 \qquad i \in V$$

in-degree constraints



$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \le |S| - 1 \qquad S \subset V,\ |S| \ge 2$$

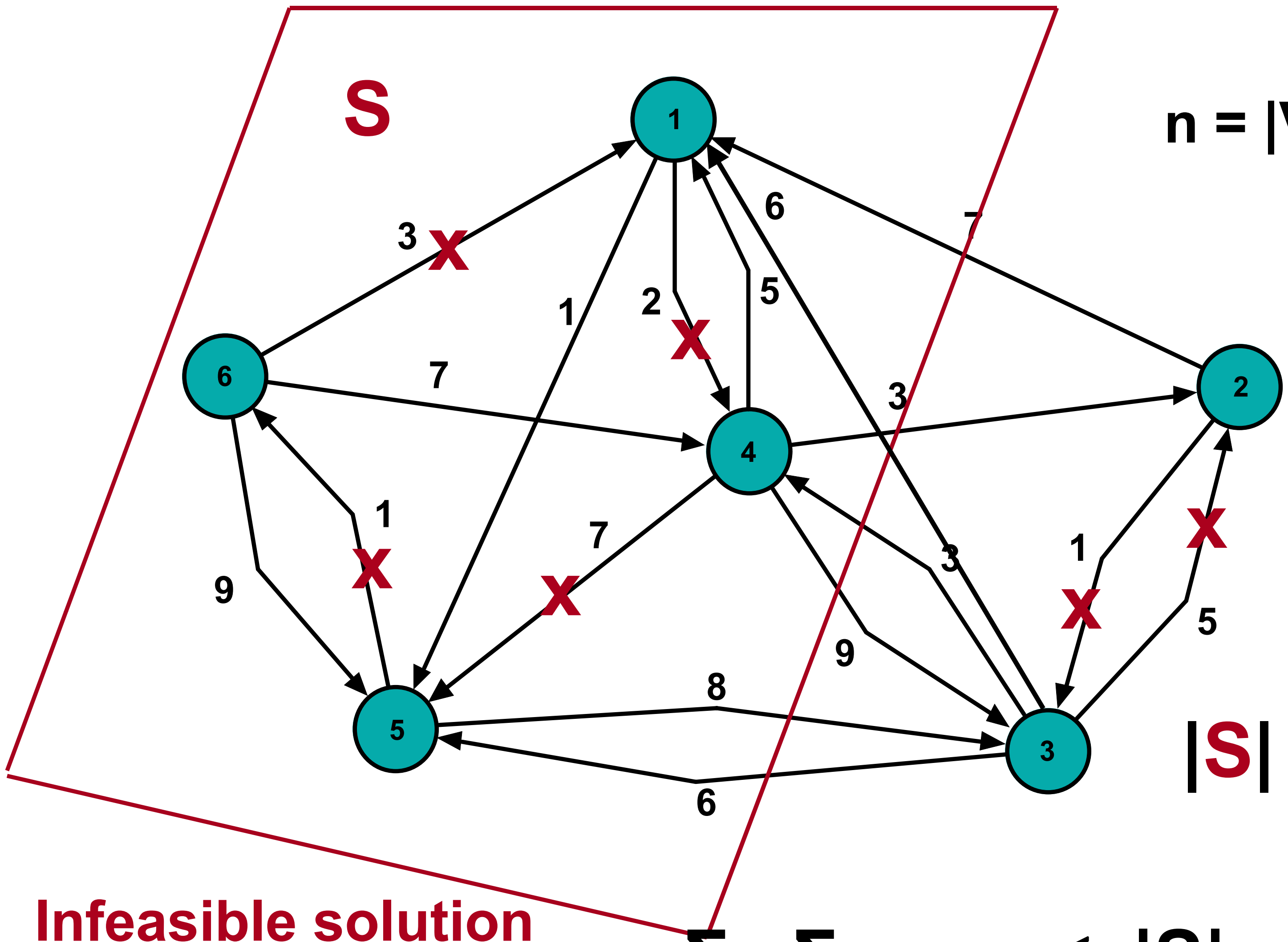
!!!

SUBTOUR ELIMINATION CONSTRAINTS

(forbid the partial tours; $O(2^n)$)

$$x_{ij} \in \{0, 1\} \qquad i \in V,\ j \in V$$

Example



n = |V| = 6

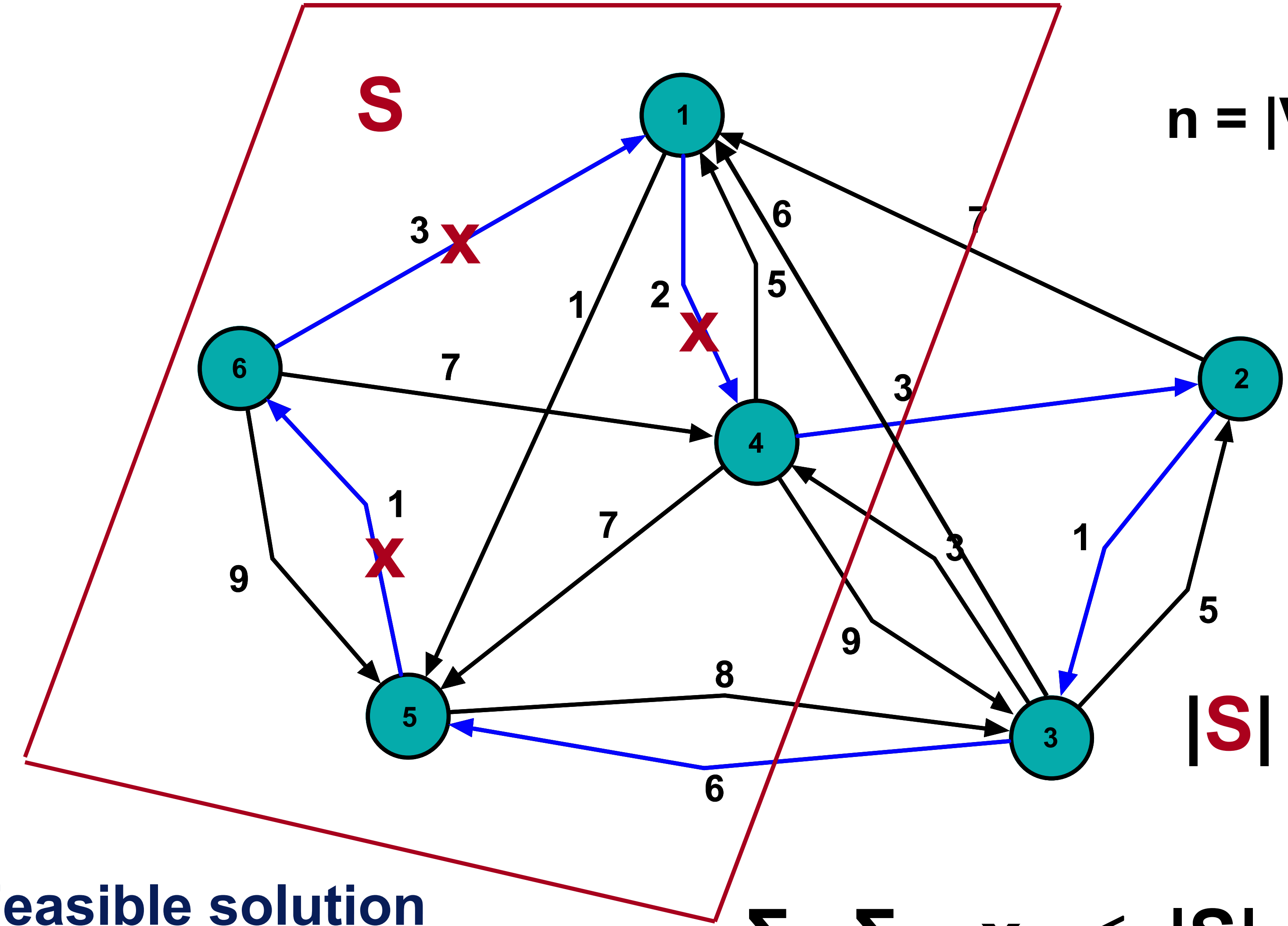
|S| = 4

X Infeasible solution

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1$$

NO!

Example



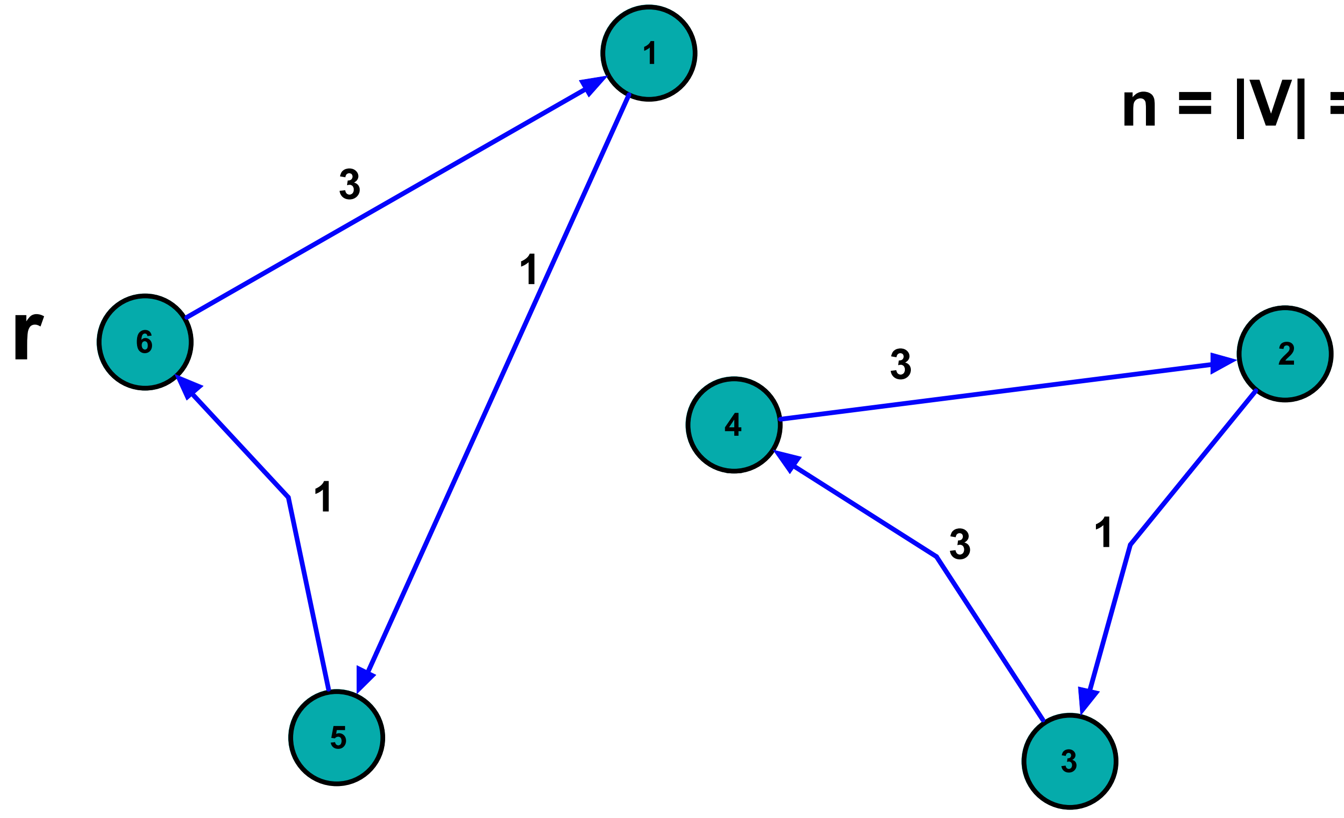
n = |V| = 6

|S| = 4

Feasible solution

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1$$ **YES!**

Example: only degree constraints imposed



n = |V| = 6

Infeasible solution: the vertices are not connected:

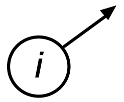
From any vertex (say r) we must reach all the other vertices (connectivity from r), and viceversa (connectivity toward r)

INTEGER LINEAR PROGRAMMING FORMULATION

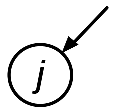
$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is in the optimal tour} \\ 0 & \text{otherwise} \end{cases} \qquad i \in V, \; j \in V$$

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} \, x_{ij}$$

s.t.



$$\sum_{j \in V} x_{ij} = 1 \qquad i \in V$$



$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V$$

$$\boxed{\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1 \qquad S \subset V, \; r \in S}$$ (for a fixed r ∈ V)

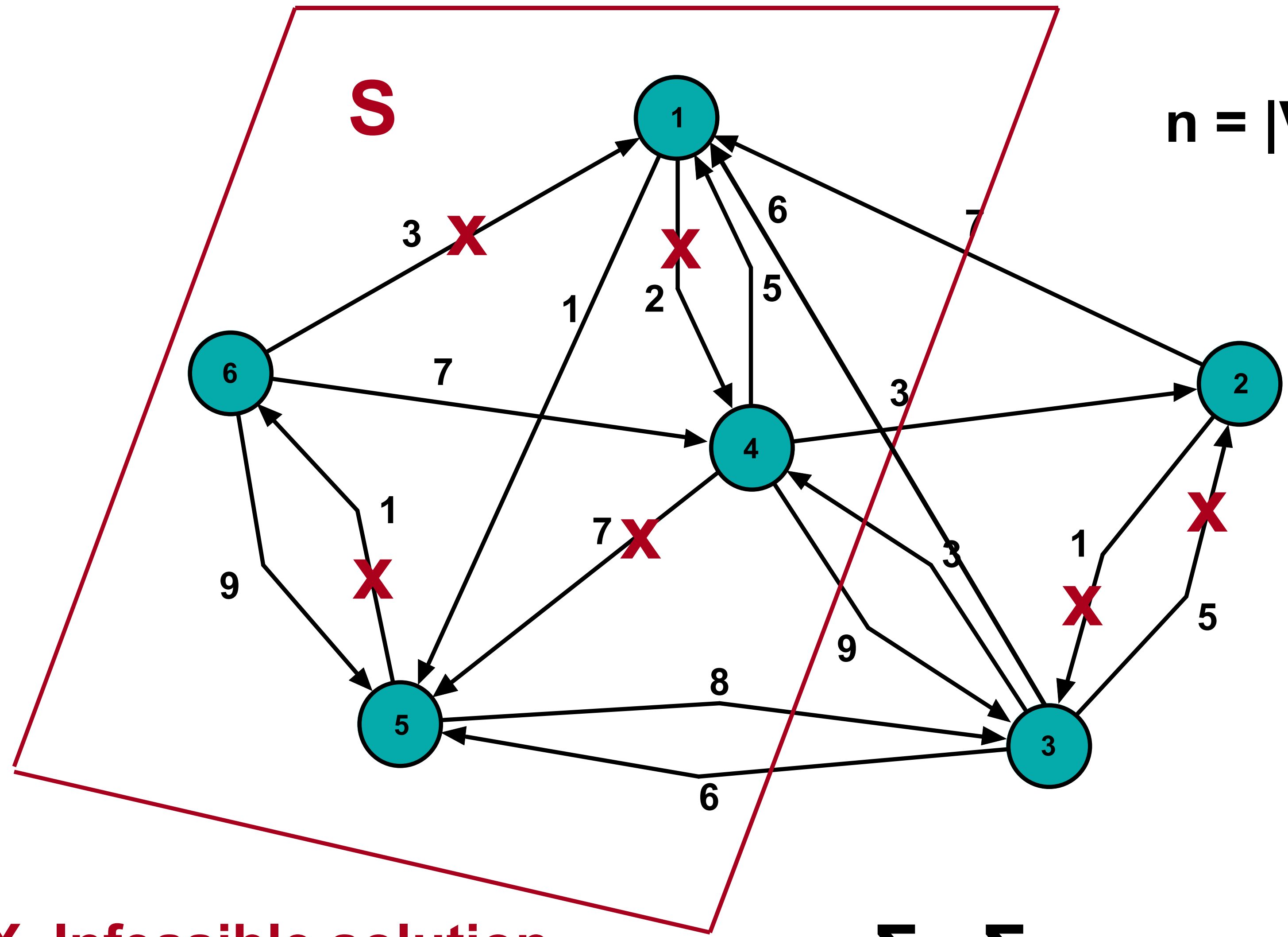
CONNECTIVITY

CONSTRAINTS

(impose the connectivity of the solution; $O(2^n)$)

$$x_{ij} \in \{0, 1\} \qquad i \in V, \; j \in V$$

Example

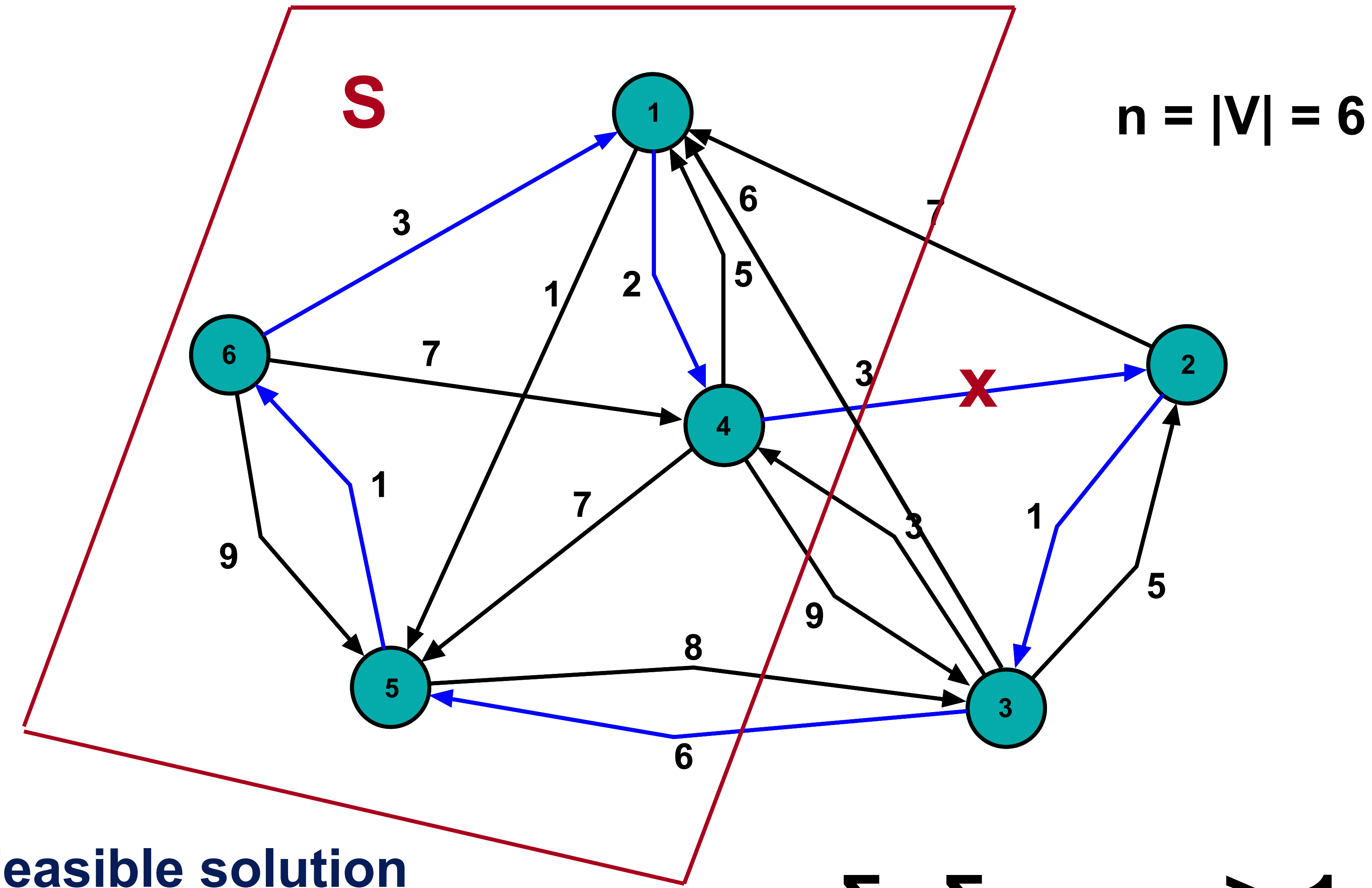


n = |V| = 6

X Infeasible solution

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1$$ **NO!**

Example



Feasible solution

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1$$ YES!

LP LOWER BOUND

- The value of the optimal solution of the Linear Programming (LP) Relaxation of the previous formulations, obtained by replacing

$x_{ij} \in \{0, 1\}$ $\qquad\qquad$ $i \in V, j \in V$

with

$0 \leq x_{ij} \leq 1$ $\qquad\qquad$ $i \in V, j \in V$

represents a valid Lower Bound on the value of the optimal solution of the ATSP.

- This LP Relaxation can be efficiently solved by using appropriate Separation Procedures.

- The LP Relaxation can be strengthened (so as to obtain better lower bounds) by adding valid inequalities, which are "redundant" for the ILP model, but can be violated by its LP Relaxation.

ALTERNATIVE ILP FORMULATIONS

- Miller-Tucker-Zemlin (J. ACM, 1960);
- Fox-Gavish-Graves (Operations Research 1980);
- Wong (IEEE Conference …, 1980);
- Claus (SIAM J. on Algebraic Discrete Methods, 1984);
- Finke-Claus-Gunn (Congressus Numerantium, 1984);
- Desrochers-Laporte (Oper. Res. Letters, 1991);
- Gouveia-Pires (EJOR, 1999);
- Gouveia-Pires (Discrete Applied Mathematics, 2001);
- Sherali-Driscoll (Operations Research 2002);
- Sarin-Sherali-Bhootra (Oper. Res. Letters, 2005);
- Sherali-Sarin-Tsai (Discrete Optimization, 2006);
- …

ALTERNATIVE ILP FORMULATIONS

- These formulations involve a polynomial number of constraints,

  but

- their Linear Programming Relaxations generally produce Lower Bounds weaker than those corresponding to the Dantzig-Fulkerson-Johnson formulation.

EXACT ALGORITHMS FOR THE ATSP

- **Branch-and-Bound Algorithms**:
- Little (Operations Research, 1964);
- Bellmore-Malone (Operations Research, 1971);
- Garfinkel (Operations Research, 1973);
- Smith-Srinivasan-Thompson (Annals Discrete Math., 1977);
- Carpaneto-T. (Management Science, 1980);
- Balas-Christofides (Mathematical Programming, 1981);
- Pekny-Miller (Oper. Res. Letters, 1989);
- Fischetti-T. (Mathematical Programming, 1992);
- Pekny-Miller (Mathematical Programming, 1992);
- Carpaneto-Dell’Amico-T. (ACM Trans. Math. Software, 1995);
- …

EXACT ALGORITHMS FOR THE ATSP

- Branch-and-Cut Algorithms:
- Fischetti-T. (Management Science, 1997);
- Fischetti-Lodi-T. (LNCS Springer, 2003);
- …

- Surveys:
- Balas-T. (The Traveling Salesman Problem, Lawler et al. eds, Wiley, 1985);
- Fischetti-Lodi-T.(The TSP and its Variations, Gutin-Punnen eds, Kluwer, 2002);
- …

CLASSICAL LOWER BOUNDS

ASSIGNMENT PROBLEM (AP) RELAXATION ($O(n^3)$ time)

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

s.t.

$$\sum_{j \in V} x_{ij} = 1 \qquad i \in V$$

$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V$$

$$\xcancel{\sum_{i \in S} \sum_{j \in S} x_{ij} \le |S| - 1 \qquad S \subset V, \ |S| \ge 2}$$ Connectivity Constraints

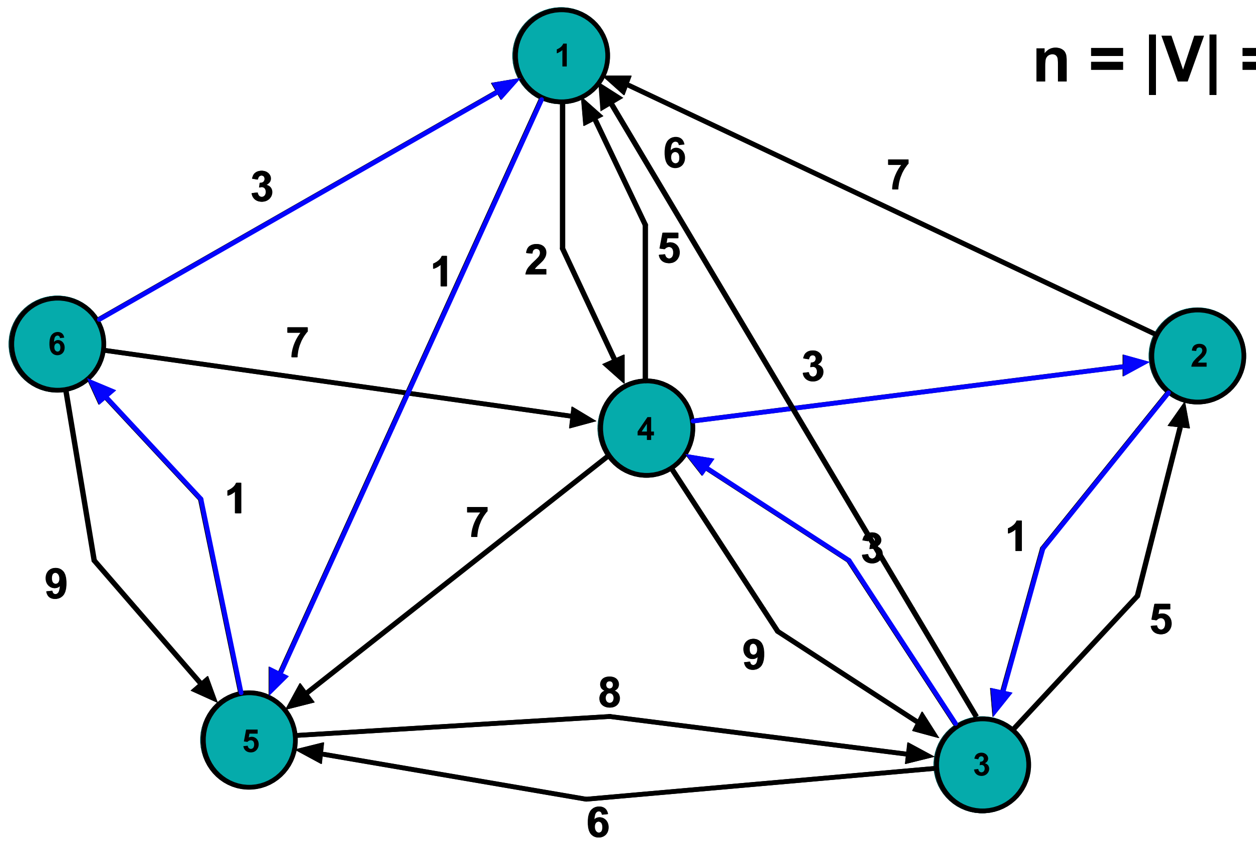
$$\xcancel{x_{ij} \in \{0, 1\} \qquad i \in V, j \in V}$$

$$\boxed{x_{ij} \ge 0 \ \ \text{(LP Relaxation)} \quad i \in V, j \in V}$$

* The AP solution is given by a family of "*subtours*" (partial circuits)

局部回路

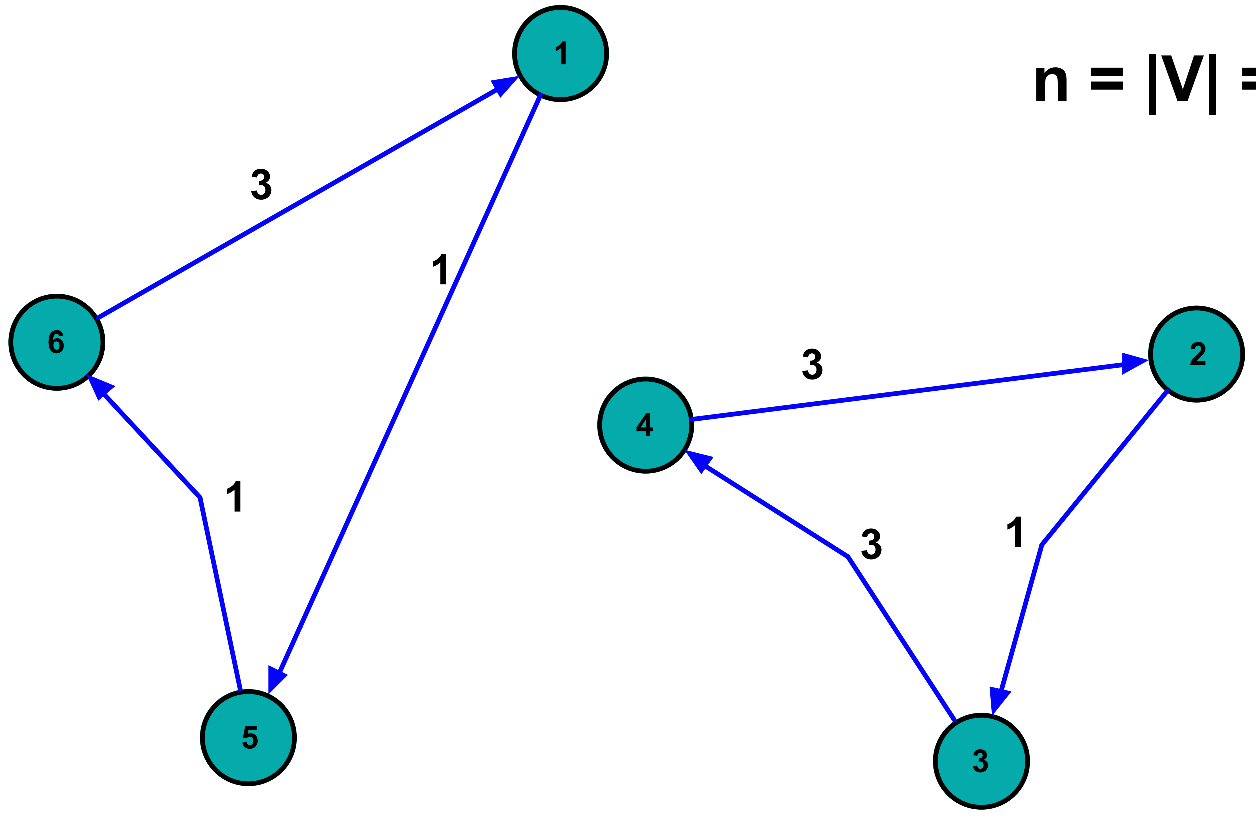
Example: AP relaxation of ATSP



n = |V| = 6

Optimal assignment

Example: AP relaxation of ATSP



n = |V| = 6

Optimal assignment

Lower bound = v(AP) = (1 + 1 + 3) + (3 + 1 + 3) = 12

Optimal solution Cost = 16

r - SHORTEST SPANNING ARBORESCENCE RELAXATION

$$\min \sum_{i\in V} \sum_{j\in V} c_{ij}\, x_{ij}$$

s.t.

$$\xcancel{\sum_{j\in V} x_{ij} = 1 \qquad i \in V}$$ out-degree constraints

$$\sum_{i\in V} x_{ij} = 1 \qquad j \in V$$ in-degree constraints

$$\sum_{i\in S} \sum_{j\in V\setminus S} x_{ij} \geq 1 \qquad S \subset V,\ r \in S$$ (for a fixed $r \in V$)

$$x_{ij} \in \{0, 1\} \qquad i \in V,\ j \in V$$

最小生成树

r - SHORTEST SPANNING ARBORESCENCE RELAXATION

Partition the Arc Set A into two subsets: (i,j) $\quad i \in V, \ j \in V \setminus r$

(i,r) $\quad i \in V$

$$\min \sum_{i \in V} \sum_{j \in V \setminus r} c_{ij} x_{ij} + \sum_{i \in V} c_{ir} x_{ir}$$

s.t.

$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V \setminus r$$

in-degree constraints

$$\sum_{i \in V} x_{ir} = 1$$

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1 \qquad S \subset V, \ r \in S \qquad (j \in V \setminus r)$$

$$x_{ij} \in \{0, 1\} \qquad i \in V, \ j \in V \setminus r$$

$$x_{ir} \in \{0, 1\} \qquad i \in V$$

The Relaxed Problem can be split into two independent problems

r - SHORTEST SPANNING ARBORESCENCE RELAXATION

Problem concerning the arc set: $(i,j) \quad i \in V, \; j \in V \setminus r$

$$\min \sum_{i \in V} \sum_{j \in V \setminus r} c_{ij} x_{ij}$$

s.t.

$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V \setminus r \qquad \text{in-degree constraints}$$

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1 \qquad S \subset V, \; r \in S \qquad \text{connectivity constraints from } r$$

$$\boxed{\xcancel{x_{ij} \in \{0, 1\} \qquad i \in V, \; j \in V \setminus r}}$$

$$x_{ij} \geq 0 \quad \text{(LP Relaxation)} \qquad i \in V, \; j \in V \setminus r$$

The problem calls for an r-SSA: $(O(n^2)$ time$)$

Shortest (Minimum Cost) Spanning Arborescence rooted at vertex r (involving n - 1 arcs)

r - SHORTEST SPANNING ARBORESCENCE RELAXATION

Problem concerning the arc set: (i,r) $\quad i \in V$

$$\min \sum_{i \in V} c_{ir} \, x_{ir}$$

s.t.

$$\sum_{i \in V} x_{ir} = 1 \qquad \text{in-degree constraint for vertex r}$$

$$\cancel{\boxed{x_{ir} \in \{0, 1\} \qquad\qquad i \in V}}$$

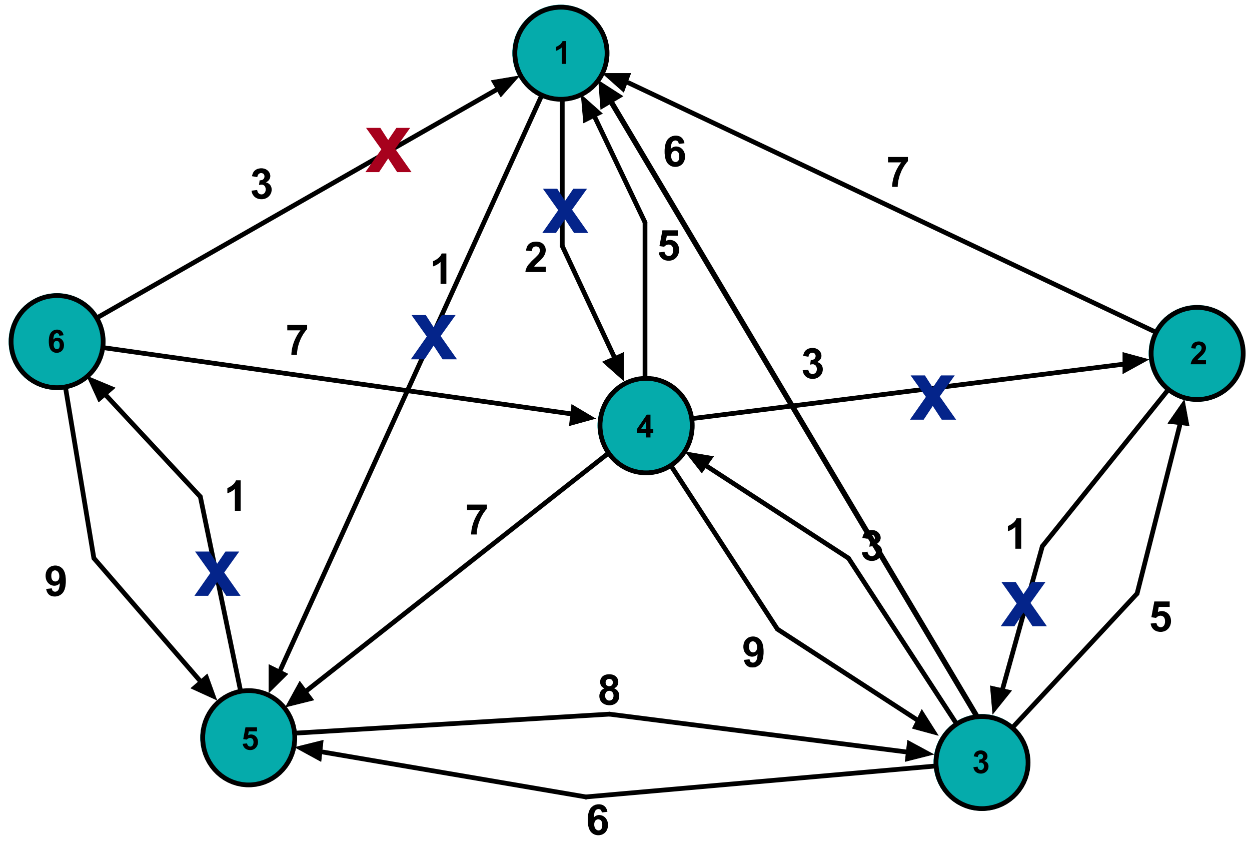
$$x_{ir} \geq 0 \quad \text{(LP Relaxation)} \qquad\qquad i \in V$$

The problem calls for an r-MCA:

Minimum Cost Arc entering vertex r (O(n) time)

Lower Bound = v(r-SSAR) = v(r-SSA) + v(r-MCA)

Example: r-SSA relaxation of ATSP (r = 1)



Optimal 1-SSA, v(1-SSA) = 1 + 1 + 2 + 3 + 1 = 8

Optimal 1-MCA, v(1-MCA) = 3,

Lower bound = 8 + 3 = 11,   Optimal solution Cost = 16

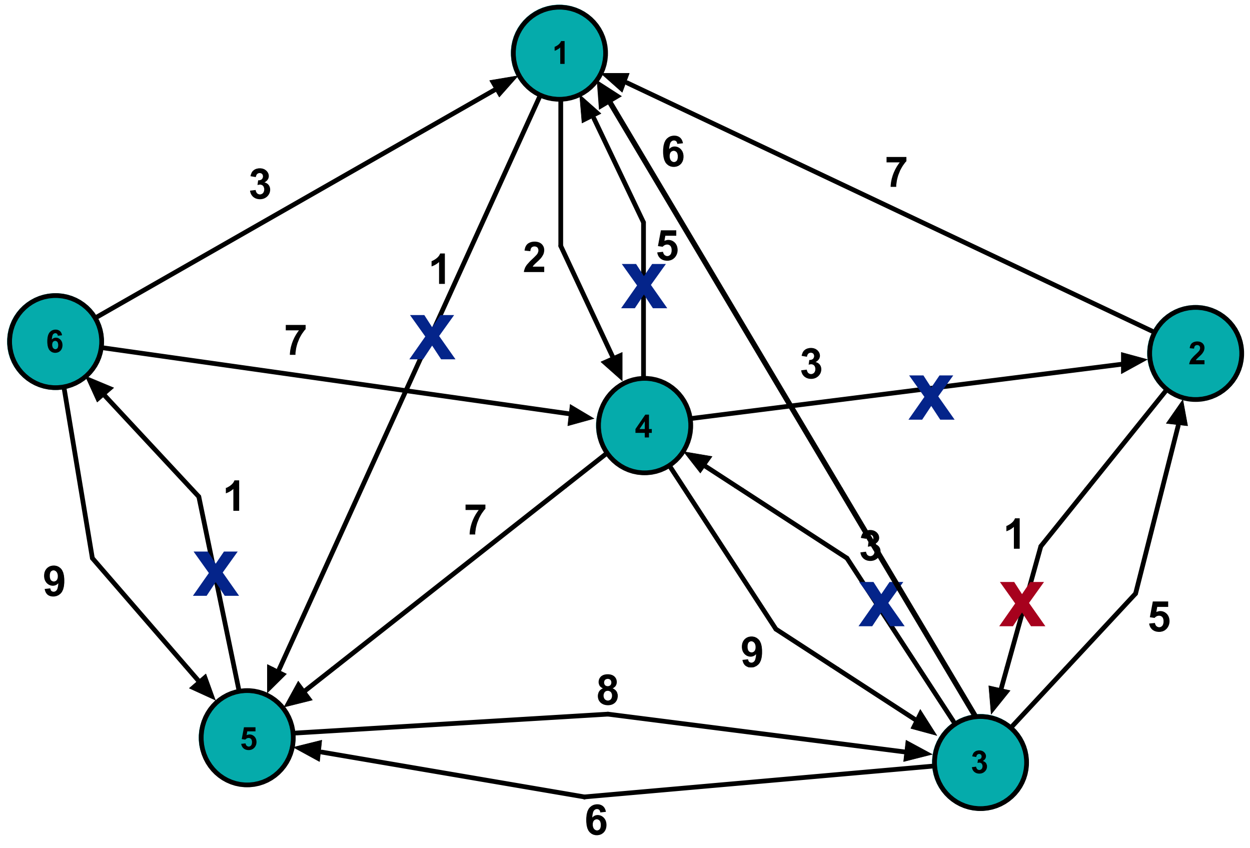
r - SHORTEST SPANNING ARBORESCENCE RELAXATION

- Different choices of the root vertex r generally produce different values of the corresponding lower bounds.

- The r - Shortest Spanning Anti - Arborescence Relaxation can be used as well (connectivity toward r).

- The lower bounds can be strengthened through Lagrangian Relaxation of the out-degree constraints, and Subgradient Optimization Procedures (to determine “good” Lagrangian multipliers).

Example: r-SSA relaxation of ATSP (r = 3)

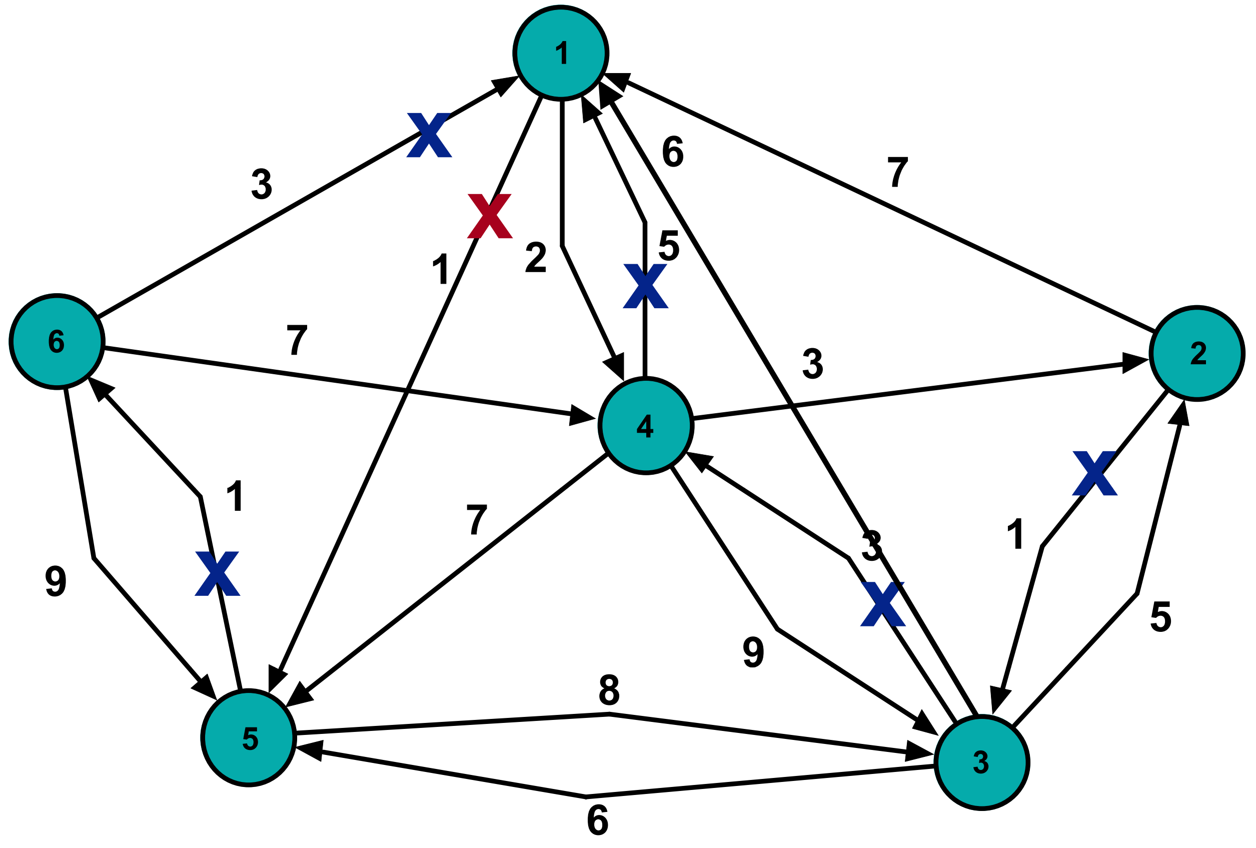


Optimal 3-SSA, v(3-SSA) = 3 + 3 + 5 + 1 + 1 = 13

Optimal 3-MCA, v(3-MCA) = 1,

Lower bound = 13 + 1 = 14, Optimal solution Cost = 16

Example: r-Anti-SSA relaxation of ATSP (r = 1)



Optimal 1-Anti-SSA, v(1-Anti-SSA) = 3 + 1 + 5 + 3 + 1 = 13

Optimal 1-Anti-MCA, v(1-Anti-MCA) = 1,

Lower bound = 13 + 1 = 14,   Optimal solution Cost = 16

LAGRANGIAN RELAXATION OF THE r - SHORTEST SPANNING ARBORESCENCE RELAXATION

(u$_i$: "Lagrangian multiplier" associated with the i-th out-degree constraint)

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

s.t.

u_i $\quad \boxed{\sum_{j \in V} x_{ij} = 1 \qquad i \in V}$ out-degree constraints

$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V$$ in-degree constraints

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1 \qquad S \subset V,\ r \in S \text{ (for a fixed } r \in V)$$

$$x_{ij} \in \{0, 1\} \qquad i \in V, j \in V$$

LAGRANGIAN RELAXATION OF THE r - SHORTEST SPANNING ARBORESCENCE RELAXATION

(u_i: “Lagrangian multiplier” associated with the i-th out-degree constraint)

$$z(u) = \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + \sum_{i \in V} u_i \left(\sum_{j \in V} x_{ij} - 1 \right)$$

s.t.

u_i $\quad \boxed{\xcancel{\sum_{j \in V} x_{ij} = 1 \qquad i \in V}}$ out-degree constraints

$\sum_{i \in V} x_{ij} = 1 \qquad j \in V$ in-degree constraints

$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1 \qquad S \subset V,\ r \in S$ (for a fixed $r \in V$)

$x_{ij} \in \{0, 1\} \qquad i \in V, j \in V$

LAGRANGIAN RELAXATION OF THE r - SHORTEST SPANNING ARBORESCENCE RELAXATION

$$z(u) = \min \sum_{i\in V}\sum_{j\in V} c_{ij}\, x_{ij} + \sum_{i\in V} u_i \Big(\sum_{j\in V} x_{ij} - 1 \Big)$$

$$= -\sum_{i\in V} u_i + \min \sum_{i\in V} \sum_{j\in V} \big(c_{ij} + u_i\big)\, x_{ij}$$

$$= -\sum_{i\in V} u_i + \min \sum_{i\in V} \sum_{j\in V} c'_{ij}\; x_{ij} \quad (\text{where } c'_{ij} = c_{ij} + u_i\,)$$

c'_{ij} is the Lagrangian cost of arc (i,j) (with $(i,j) \in A$)

LAGRANGIAN DUAL PROBLEM

Find the optimal Lagrangian multiplier vector **u*** such that:

$$z(u^*) = \max_{u} \{z(u)\} \qquad (\text{with } z(u) \text{ “nondifferentiable function”})$$

Difficult problem (heuristic solution through Subgradient Optimization Procedures)

SUBGRADIENT OPTIMIZATION PROCEDURE

(Held-Karp, Operations Research 1971, for STSP)

Lagrangian cost: $c'_{ij} = c_{ij} + u_i \quad (i,j) \in A$

* Initially: $u_i := 0 \quad i \in V$; $LB := 0$

At each iteration:

* Solve the r-SSA Relaxation with respect to the current Lagrangian costs c'_{ij} : (x_{ij}) is the corresponding optimal solution

* $LB := \max (LB.\ v(x_{ij}))$

* Subgradient vector (s_i): $s_i := \sum_{j \in V} x_{ij} - 1 \qquad i \in V$

* If $s_i = 0$ for all $i \in V$ then STOP ((x_{ij}) is the opt. sol. of ATSP)
* $u_i := u_i + a\ s_i \quad i \in V$

   where $a := b\ (UB - LB) / \sum_{i \in V} s_i^2$ (with $0 < b \leq 2$)

Stopping criteria: max number of iterations, slow increase of LB,…

SUBGRADIENT OPTIMIZATION PROCEDURE: UPDATING OF THE LAGRANGIAN MULTIPLIERS

* $c'_{ij} = c_{ij} + u_i$ (with $(i,j) \in A$)
* $u_i := u_i + a\ s_i \quad i \in V$ (with a positive)
* $s_i := \sum_{j \in V} x_{ij} - 1 \quad i \in V$
* d_i = outdegree of vertex i ($i \in V$)
  ($d_i = 0$ if $s_i = -1$; $d_i = 1$ if $s_i = 0$; $d_i \geq 2$ if $s_i \geq 1$)
* the Lagrangian cost c'_{ij} of arc (i,j) is

  decreased if $d_i = 0$.
  unchanged if $d_i = 1$,
  increased if $d_i \geq 2$.

Note that: $a \sum_{j \in V} s_i = a \left(\sum_{j \in V} \sum_{j \in V} x_{ij} - n \right) = a\,(n - n) = 0$

$\sum_{i \in V} u_i = 0$

Example: r-SSA relaxation of ATSP (r = 1)

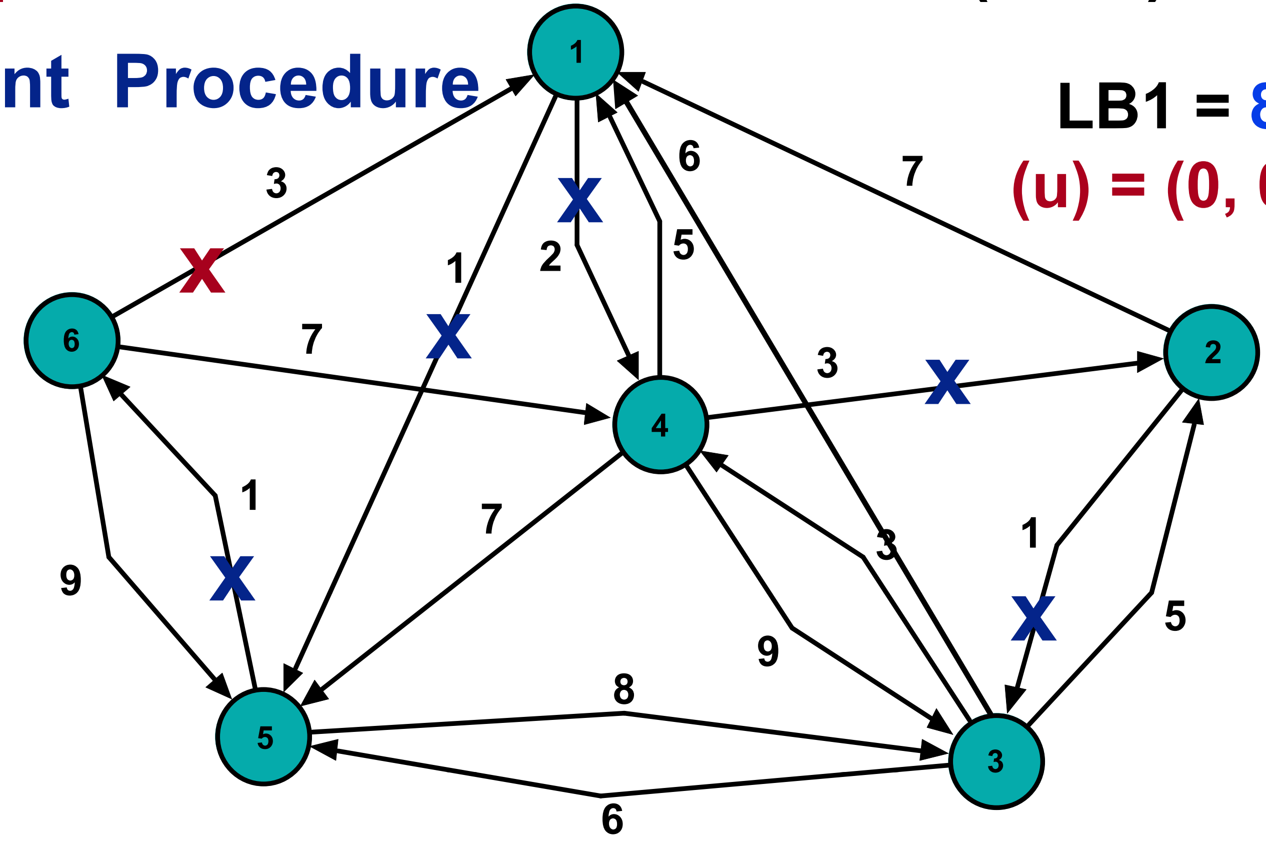
Subgradient Procedure

(b = 0.25,

UB = 19)

LB1 = 8 + 3 = 11

(u) = (0, 0, 0, 0, 0, 0)



(d) = (2, 1, 0, 1, 1, 1), (s) = (1, 0, -1, 0, 0, 0)

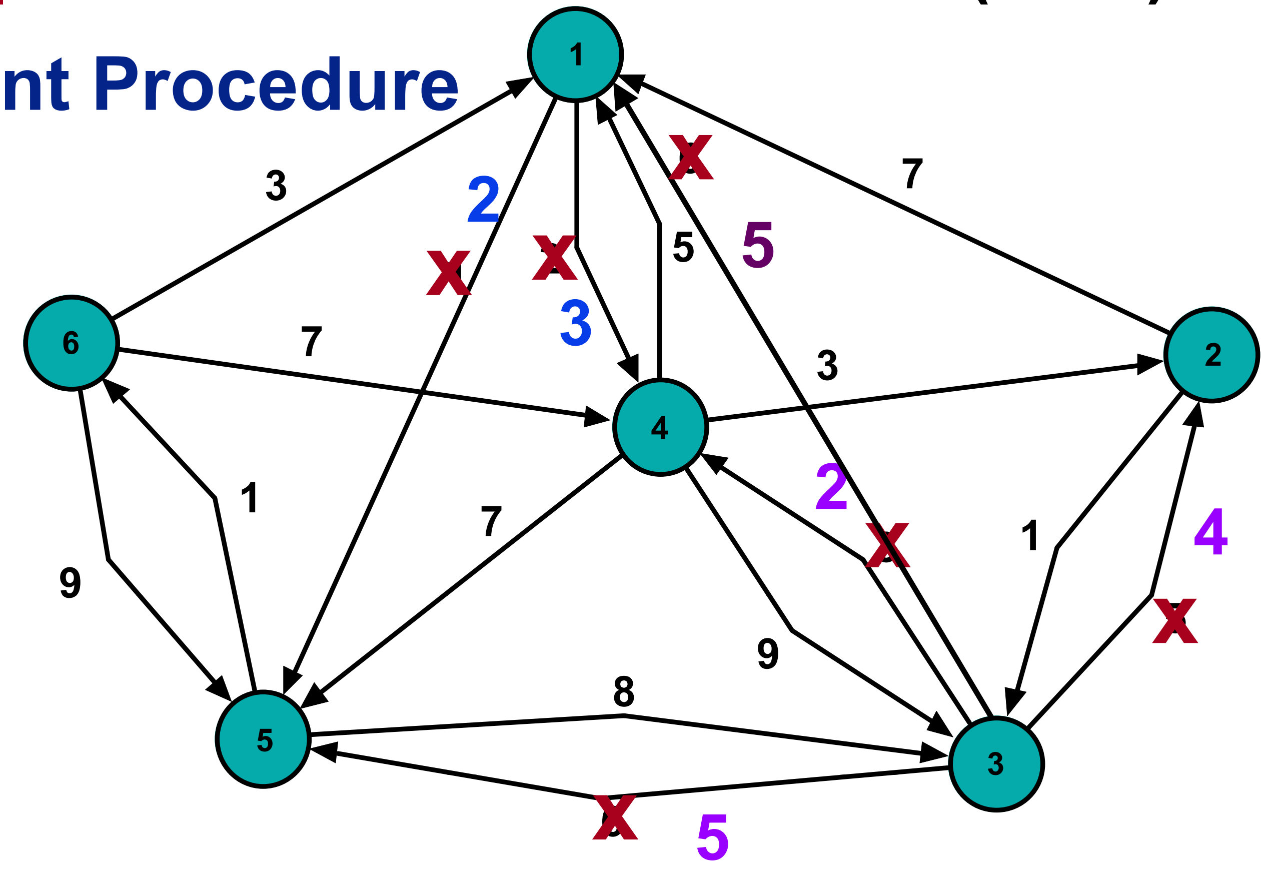
$$a := b\,(UB - LB1) / \sum_{i \in V} s_i^2 = 0.25\,(19 - 11) / 2 = 1$$

(u) = (u) + a (s) (u) = (1, 0, -1, 0, 0, 0)

Example: r-SSA relaxation of ATSP (r = 1)

Subgradient Procedure

(a = 1)



(c') = (c) + (u)

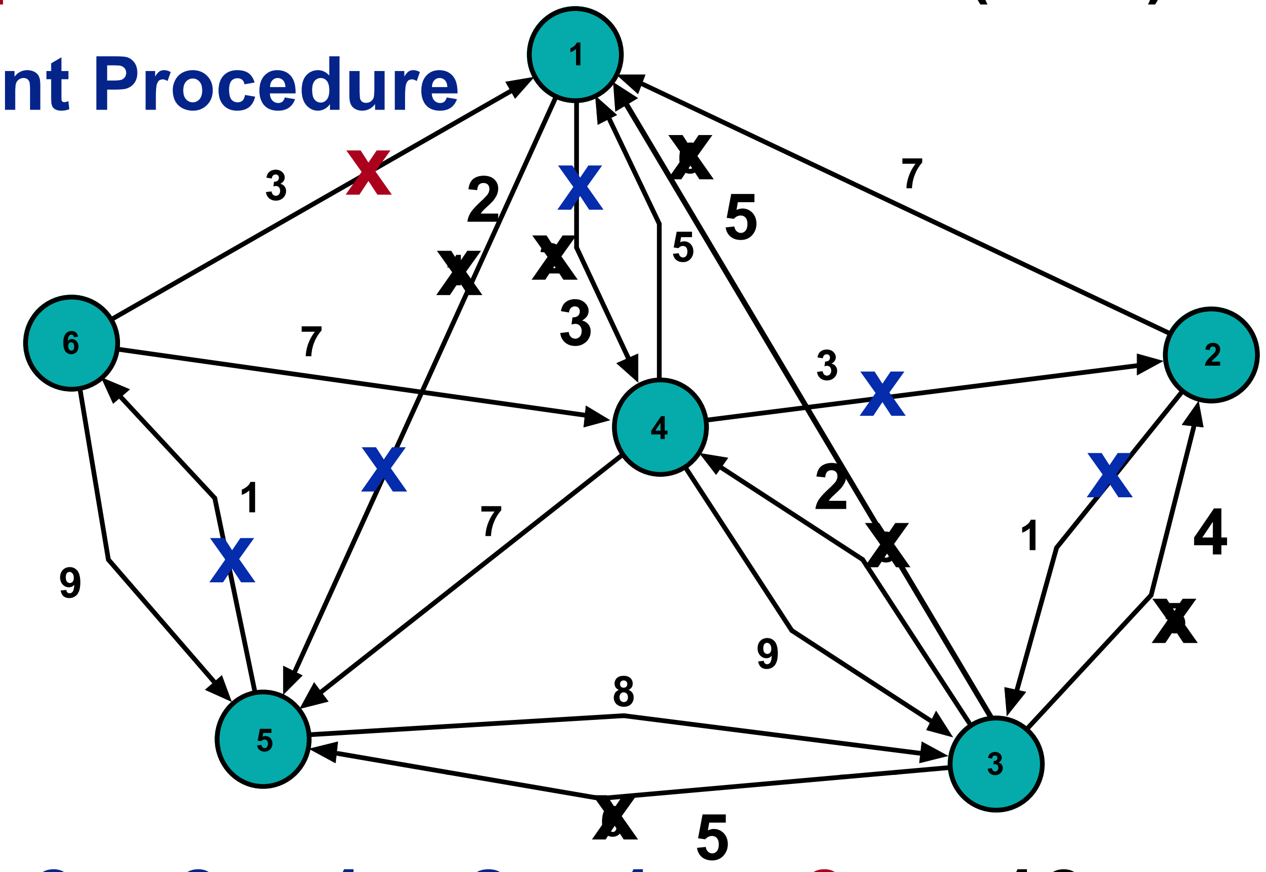
(u) = (1, 0, -1, 0, 0, 0)

Example: r-SSA relaxation of ATSP (r = 1)

Subgradient Procedure

(a = 1),

LB1 = 11



LB2 = 3 + 3 + 1 + 2 + 1 + 3 = 13

(d) = (2, 1, 0, 1, 1, 1)

…

- The AP Relaxation can be strengthened by combining the AP substructure with the SSA substructures:

- **Restricted Lagrangian Relaxation**

  (Balas-Christofides, 1981):

  1) Lagrangian Relaxation of a subset of Subtour Elimination Constraints (SECs);

  2) Solve the corresponding AP problem (w.r.t. the current Lagrangian costs);

  3) Determine good Lagrangian multipliers (through a subgradient optimization procedure) and additional SECs, to be inserted into the current Lagrangian relaxation

  and iterate on Steps 2) and 3).

• Additive Bounding Procedure (Fischetti-T., 1992):

1) Solve the AP Relaxation through a “primal-dual” algorithm ($O(n^3)$ time)
 LB := v(AP), c'_{ij} := “reduced cost” of arc (i,j);

2) Solve the r-SSA Relaxation w.r.t. costs c'_{ij} ($O(n^2)$ time):
 LB := LB + v(r-SSAR), c'_{ij} := new “reduced cost” of arc (i,j);

3) Solve the r-Anti-SSA Relaxation w.r.t. costs c'_{ij} ($O(n^2)$ time):
 LB := LB + v(r-Anti-SSAR), c'_{ij} := new “reduced cost” of arc (i,j);

Iterate Steps 2) and 3) with different root nodes r.

(globally $O(n^3)$ time)

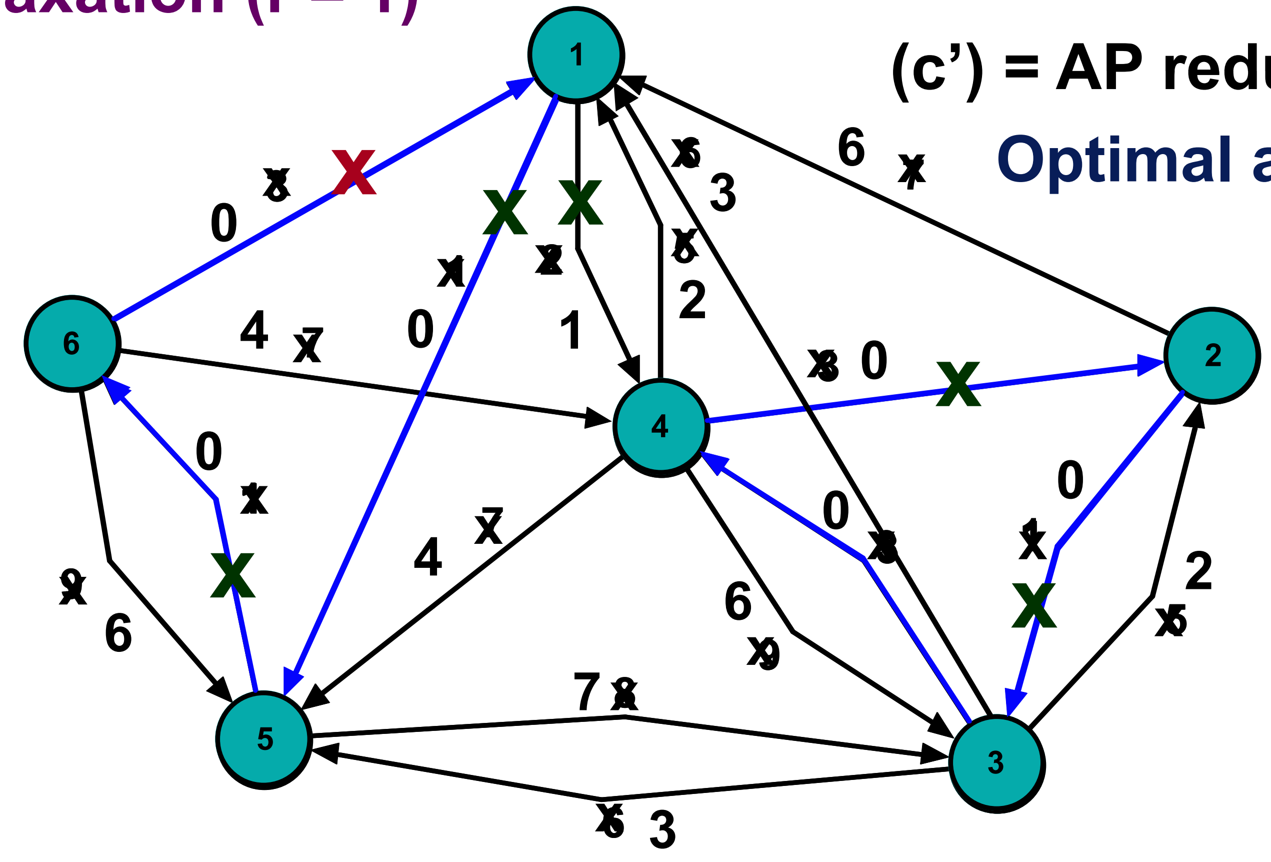
Example: Additive Bounding Procedure

r-SSA Relaxation (r = 1)

LB1 = 12

(c') = AP reduced costs

Optimal assignment



v(1-SSAR) = 0 + 0 + 1 + 0 + 0 + 0 = 1

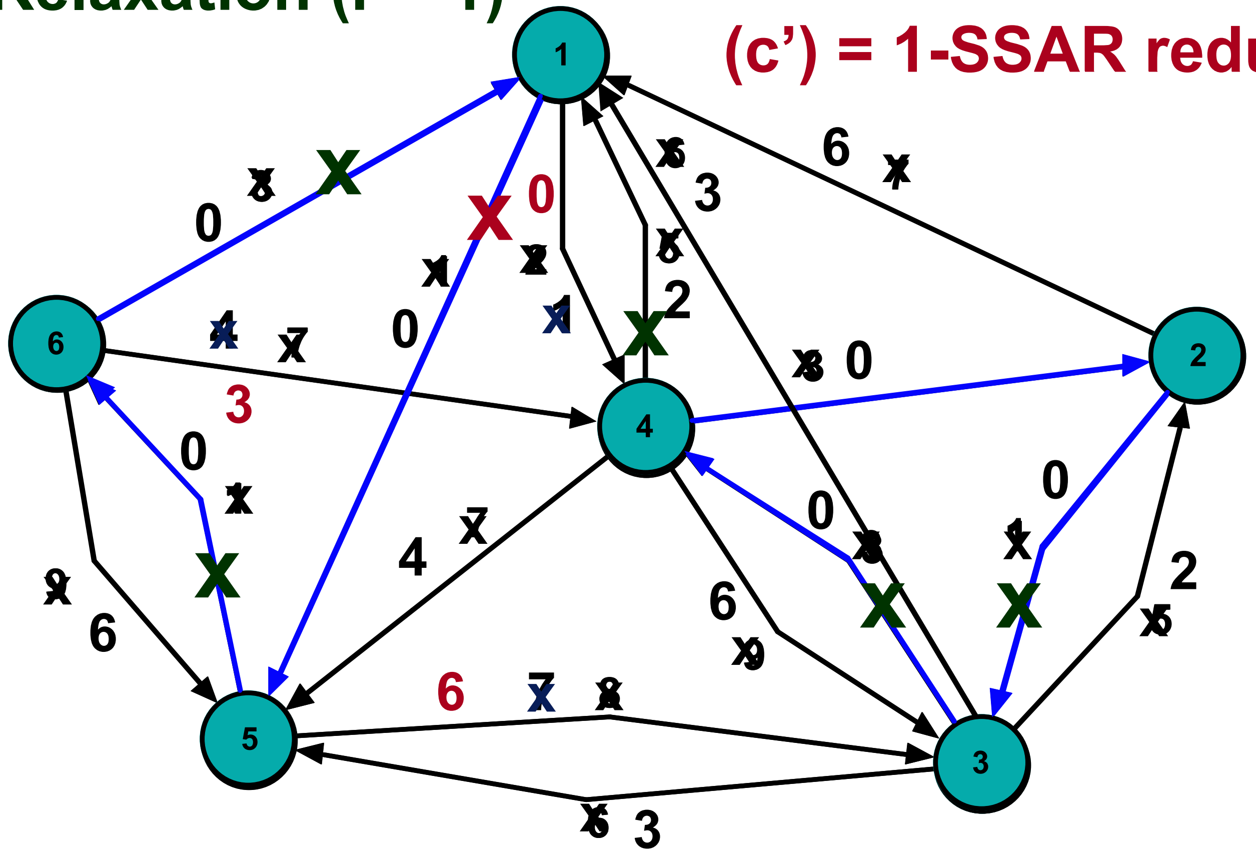
LB2 = LB1 + v(1-SSAR) = 12 + 1 = 13

Example: Additive Bounding Procedure

r-Anti-SSA Relaxation (r = 1)

LB1 = 13

(c') = 1-SSAR reduced costs



v(1-Anti-SSAR) = 0 + 0 + 2 + 0 + 0 + 0 = 2

LB2 = LB1 + v(1-Anti-SSAR) = 13 + 2 = 15

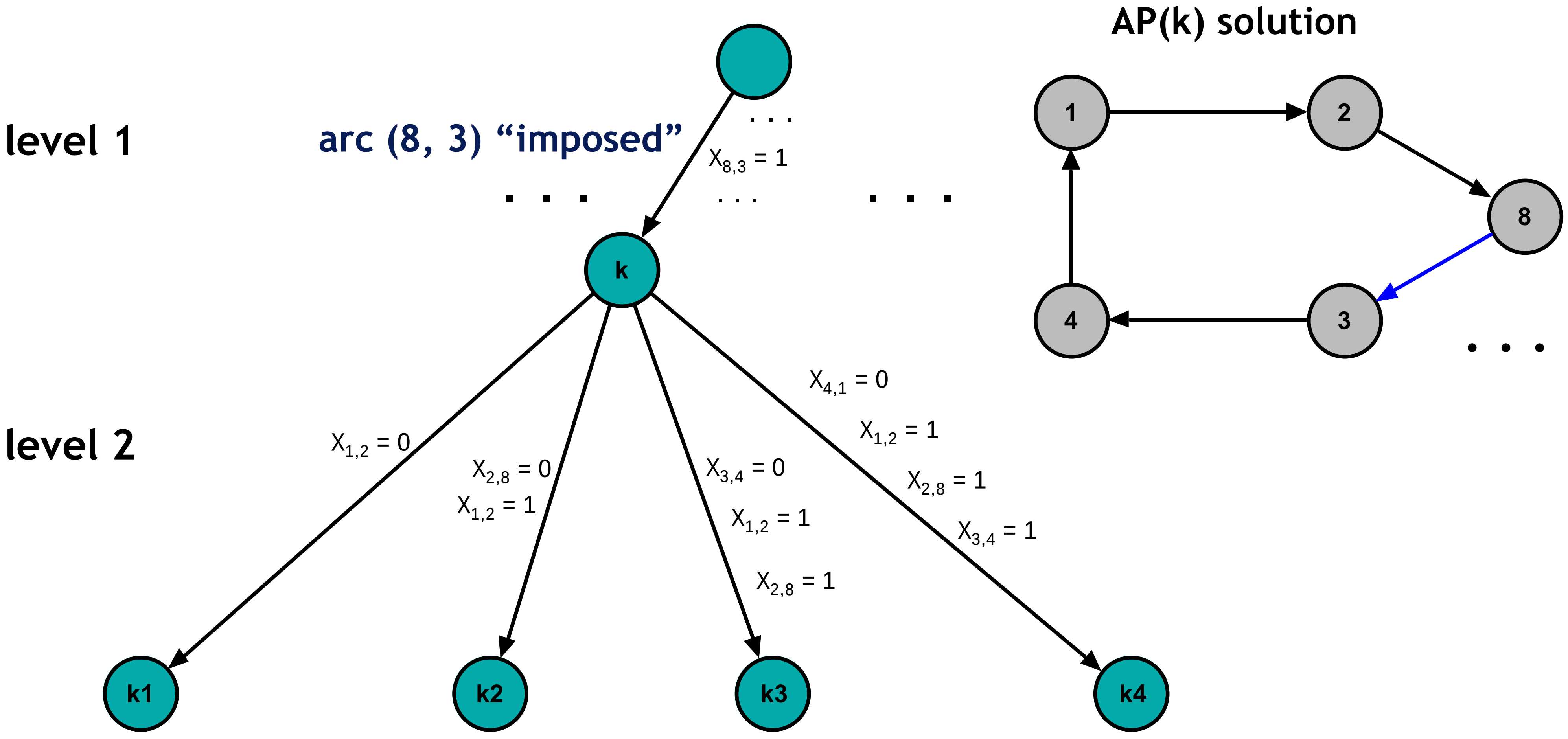
BRANCH-AND-BOUND ALGORITHM FOR ATSP BASED ON THE AP RELAXATION

(Carpaneto-T., Man. Sc. 1980)

- At each node of the decision tree solve the AP-Relaxation of the corresponding subproblem (LB= v(AP)).
- If LB $\geq$ Z* fathom the node (Z* = cost of the best solution).
- If the AP solution contains no subtour (feasible solution), update Z* (and the best solution) and "fathom" the node.
- Otherwise: SUBTOUR-ELIMINATION BRANCHING SCHEME:

  - Select the subtour S with the minimum number h of not imposed arcs.
  - Generate h descendent nodes so as to forbid subtour S for each of them (by "imposing" and "excluding" proper arc subsets).

BRANCHING TREE FOR ATSP



BRANCH-AND-BOUND ALGORITHM by Carpaneto-Dell'Amico.-T. (1995)

- At the root node of the decision tree:

  - solve the AP Relaxation;

  - apply the Patch Heuristic Algorithm (Karp-Steele, 1985) to determine an initial tour of cost Z^*;

  - apply a Reduction Procedure (based on the AP "reduced costs" c'_{ij}) to fix to zero as many variables as possible (transformation of the complete graph into a sparse one): if $v(AP) + c'_{ij} \geq Z^*$ set $x_{ij} = 0$ (i.e. remove arc (i,j) from A).

- At each node, the corresponding AP Relaxation is computed (by using effective parametric techniques) in $O(n^2)$ time.

BRANCH-AND-CUT ALGORITHMS (Padberg - Rinaldi for STSP, 1987-1991)

- At each node, a Lower Bound is obtained by solving an LP Relaxation of the corresponding subproblem, containing only a subset of the constraints (degree constraints, some connectivity constraints, …).

- The LP Relaxation is iteratively tightened by adding valid inequalities that are violated by the current optimal LP solution (x^*_{ij}).
- These inequalities are identified by solving the Separation Problem:

- given a solution (x^*_{ij}), find a member $a\,x \leq b$ of a given family F of valid inequalities for ATSP, such that

$a\,x^* > b$ holds (maximum of $d = a\,x^* - b$, with $d > 0$) .

* Exact or heuristic Separation Procedures can be used.

SEPARATION PROBLEM FOR THE CONNECTIVITY CONSTRAINTS

- Given an optimal LP solution (x^*_{ij}) such that:

$$\sum_{i \in V} x^*_{ih} = \sum_{i \in V} x^*_{hj} = 1 \qquad h \in V$$

does exist a vertex subset S $(S \subset V, \; r \in S)$ such that:

$$\sum_{i \in S} \sum_{j \in V \setminus S} x^*_{ij} < 1 \quad ?$$

EXACT SEPARATION PROCEDURE FOR THE CONNECTIVITY CONSTRAINTS

- For each vertex $t \subset V \setminus \{r\}$, define the NETWORK

  $G^* = (V,A)$, source = r, sink = t

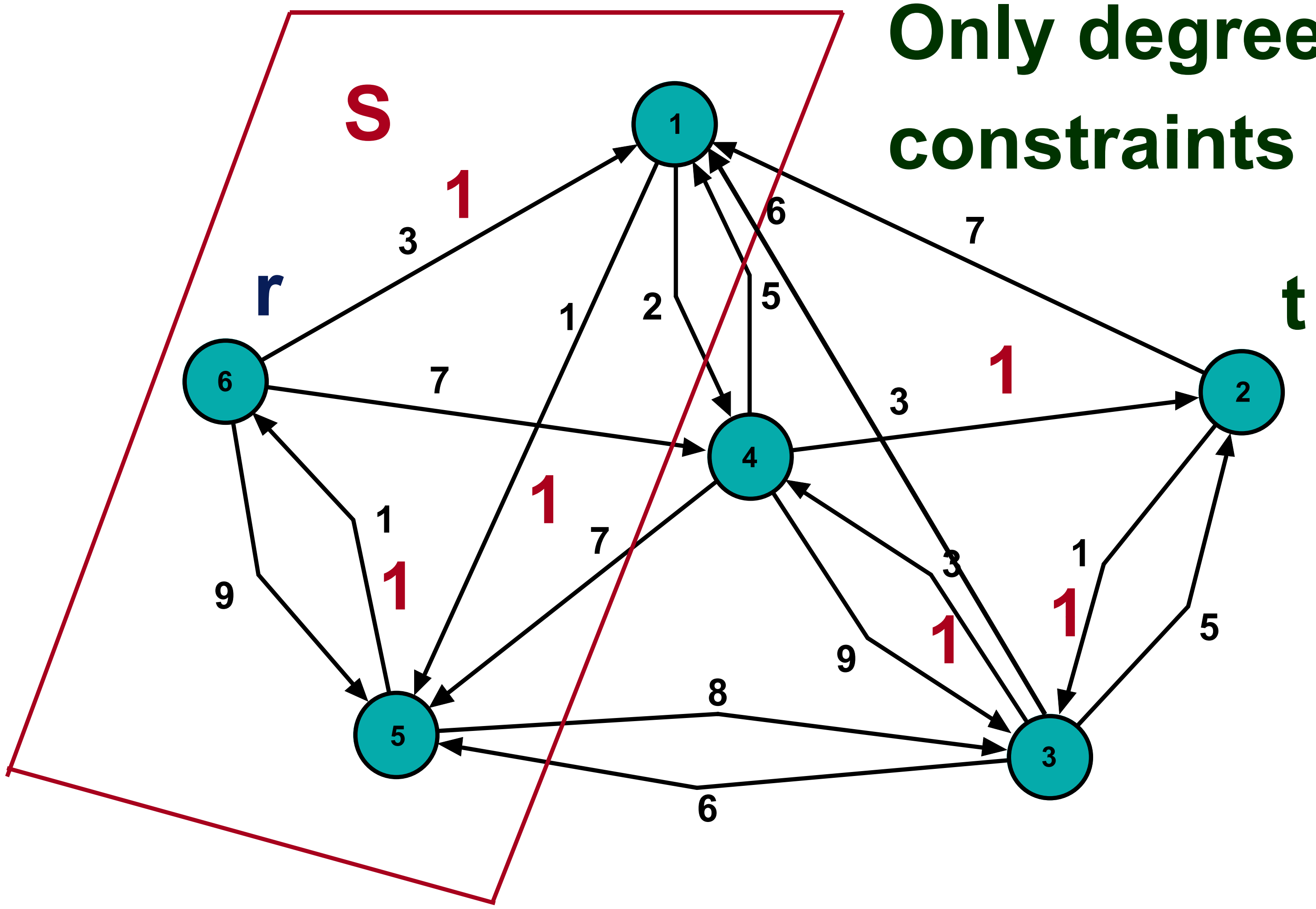
  capacity of arc (i,j) = x^*_{ij}

- CAPACITY of "cut" (S, V\S) with $r \in S$ and $t \in V\setminus S$ =

$$\sum_{i \in S} \sum_{j \in V\setminus S} x^*_{ij}$$

Example NETWORK G* = (V,A) capacity (x*)

Only degree constraints imposed



CAPACITY of “cut” (S, V\S) = 0

EXACT SEPARATION PROCEDURE FOR THE CONNECTIVITY CONSTRAINTS

- Minimum Capacity Cut from r to t = Maximum Flow from r to t
- Determine the MAXIMUM FLOW from r to t: capacity of cut (S*, V\S*)

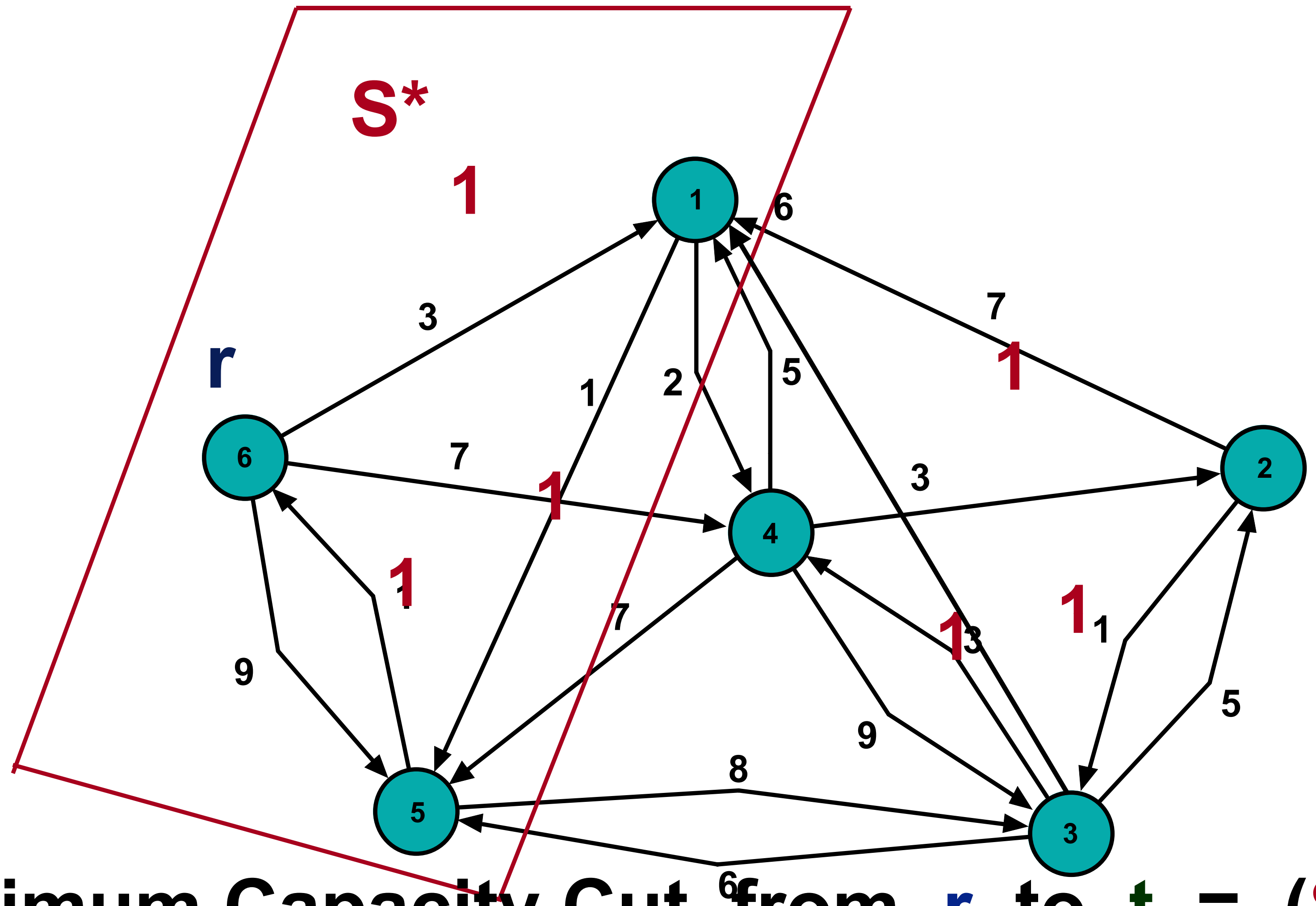
* If Maximum Flow < 1 then constraint

$$\sum_{i \in S^*} \sum_{j \in V \setminus S^*} x_{ij} \geq 1 \quad \text{is violated}$$

(most violated connectivity constraint with $r \in S$ and $t \in V \setminus S$)

$O(n^3)$ time for each vertex t (global time $O(n^4)$)

Example NETWORK G* = (V,A) capacity (x*)



Minimum Capacity Cut from r to t = (S*, V\S*)

Maximum Flow from r to t = 0: $\sum_{i \in S^*} \sum_{j \in V \setminus S^*} x_{ij} = 0$

BRANCH-AND-CUT ALGORITHM (Fischetti-T., 1997)

- Separation procedures for the identification of the following valid inequalities:

- Connectivity constraints (exact alg.);

- Comb inequalities (heur. alg.);

- D_k^+ and D_k^- inequalities (Groetschel-Padberg, 1985) (exact and heur. alg.);

- ODD CAT inequalities (Balas, 1989) (heur. alg.)

PRICING PROCEDURE

A pricing procedure is applied to decrease the number of variables considered in the LP Relaxation:

1) Select a subset T of variables (other variables set to zero);
2) Solve the LP Relaxation by considering only the variables in T;
3) Compute the reduced cost of the variables not in T;
4) Add to T (a subset of) the variables with negative reduced cost and return to Step 2);

   if no negative reduced cost is found then STOP (the current solution is the optimal solution of the original LP Relaxation).

- An effective Pricing Scheme, based on the optimal solution of an associated Assignment Problem, is used to decrease the number of variables added to T in Step 4).

BRANCHING SCHEME (Fischetti-Lodi-T., 2003)

- Select a fractional variable x^*_{ij} close to 0.5 and having been "persistently" fractional in the "last" optimal LP solutions.

- Generate 2 descendent nodes by imposing:

  $x_{ij} = 1$ and $x_{ij} = 0$, respectively.

COMPUTATIONAL RESULTS

* Algorithms:

- CDT95: Carpaneto-Dell'Amico-T. (AP Relax.) (ACM Trans. Math. Soft. 1995);
- FT92 : Fischetti-T. (Additive Bounding Procedure) (Math. Program.1992);
- FLT03: Fischetti-Lodi-T. (B. & C.) (Man. Sc. 1997, LNCS Springer 2003).

- DIGITAL ALPHA 533 MHz (times in seconds).

- LP Solver: CPLEX 6.5.3

- Test instances:

  Randomly Generated instances:

  c_{ij} integer uniformly random in [1,1000] (n = 500, 1000);

  ATSP benchmark instances from TSPLIB (Reinelt, ORSA J. Comp., 1991), additional real-world instances from Balas (2000) and F.-T.-V. (2002).

  Time limit for each instance: 3600 seconds.

	CDT95			FT92			FLT03		
Name	%Gap	Nodes	Time	%Gap	Nodes	Time	%Gap	Nodes	Time
ran500.a	0.03	15	0.2	0.03	11	2.2	0.00	3	7.3
ran500.b	0.01	53	0.1	0.01	24	1.4	0.00	17	23.5
ran500.c	0.09	23	0.1	0.08	35	1.7	0.00	61	54.4
ran1000.a	0.02	91	1.2	0.02	82	3.2	0.00	12	14.5
ran1000.b	0.08	132	2.7	0.07	99	6.8	0.01	89	73.6
ran1000.c	0.03	64	1.0	0.03	65	3.7	0.00	37	54.8
rbg323	0.00	1	0.1	0.00	1	0.3	0.00	1	0.4
rbg443	0.00	1	0.1	0.00	1	1.2	0.00	1	1.4
p43	97.37	186698	T. L.	0.37	321417	T. L.	0.16	141	9.3
ry48p	13.21	196577	T. L.	2.94	22825	20.3	0.53	7	0.8
uk66	72.91	193383	T. L.	31.71	50005	152.1	2.29	47	5.7
balas108	25.00	366420	T. L.	9.87	224537	T. L.	1.97	267	89.0
balas160	19.40	248006	T. L.	11.34	114938	T. L.	1.26	737	671.1
ft53	14.11	154920	T. L.	1.56	131	0.2	0.00	1	0.1
ft70	1.80	2624	0.4	0.57	220	0.3	0.02	5	0.3
kro124p	6.22	166664	T. L.	2.73	18909	135.7	0.04	3	1.0
ftv47	6.98	585	0.1	3.49	1026	0.4	1.01	11	0.5
ftv150	3.91	2645	3.1	3.03	2778	17.0	0.27	21	2.6

TRANSFORMATION OF ATSP INTO STSP

Any ATSP instance with n vertices can be transformed into an equivalent STSP instance with 2n nodes (Jonker-Volgenant, 1983; Junger-Reinelt-Rinaldi, 1995; Kumar-Li, 2007).

Very effective Branch-and-Cut algorithms for STSP have been proposed:

- Padberg-Rinaldi (Oper. Res. Letters 1987, SIAM Review 1991)
- Groetschel-Holland (Math. Progr. 1991)
- Applegate-Bixby-Chvatal-Cook (Doc. Math., J. DMV 1998, Concorde Code 1999, Princeton Univ. Press 2006).

* Concorde Code

Most effective code for STSP

TRANSFORMATION OF ATSP INTO STSP

Given a directed graph G = (V,A) with n vertices and m arcs,

build an undirected graph G` = (V`,E) with
2n nodes and (m + n) edges:

- for each vertex $i \in V$ consider two nodes : i and $(n + i)$;
- for each vertex $i \in V$ consider an edge $(i, n + i)$ with cost 0;
- for each arc $(i, j) \in A$ consider an edge $(n + i, j)$ with cost

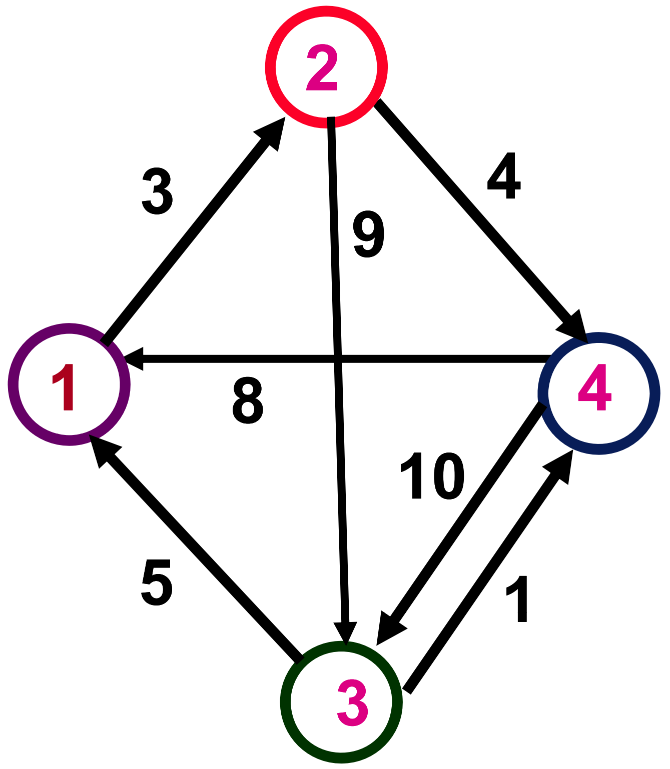
$$c_{ij} + M$$

where M is a “sufficiently” large positive value

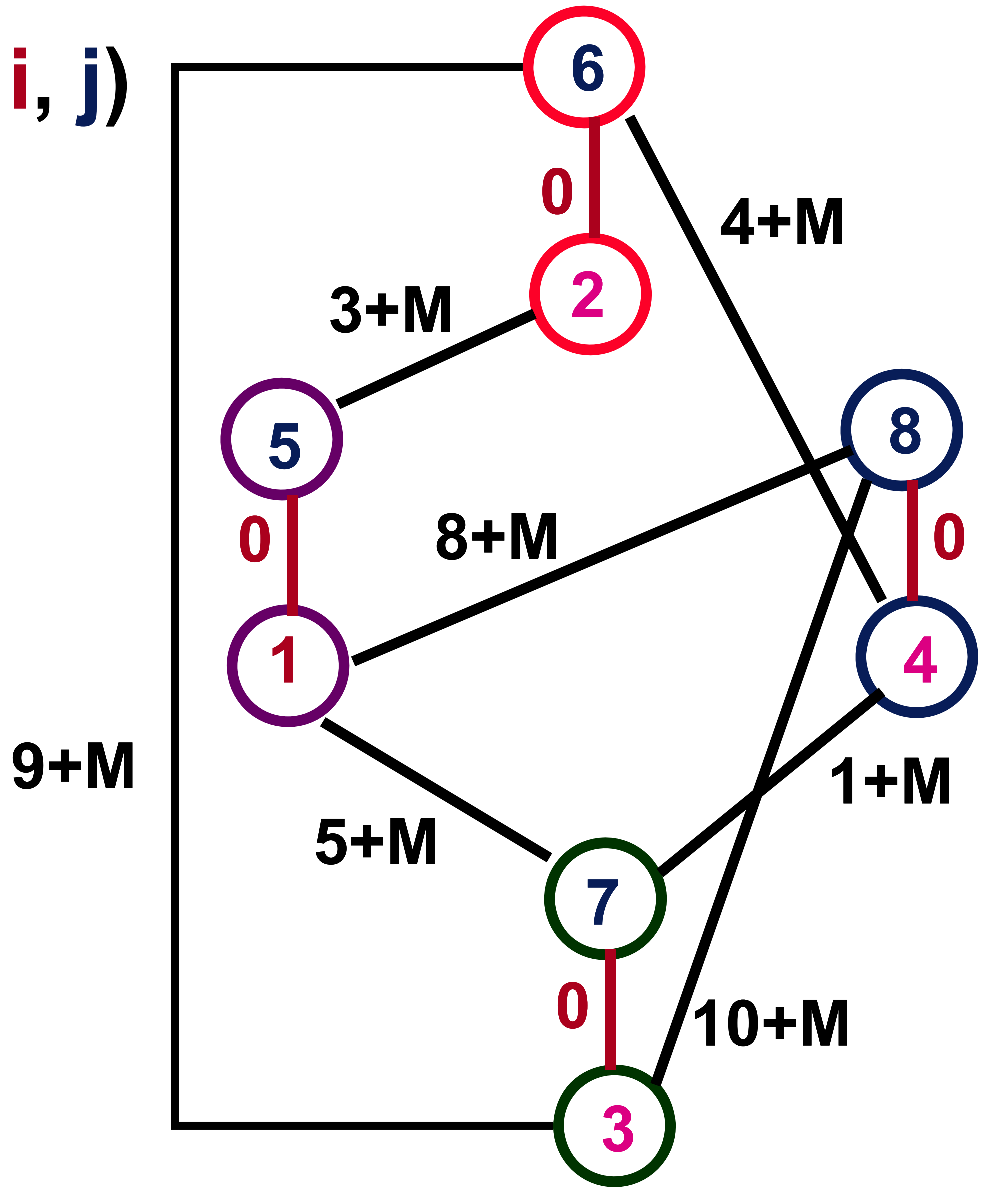
Cost of ATSP = Cost of STSP - n M

TRANSFORMATION OF ATSP INTO STSP

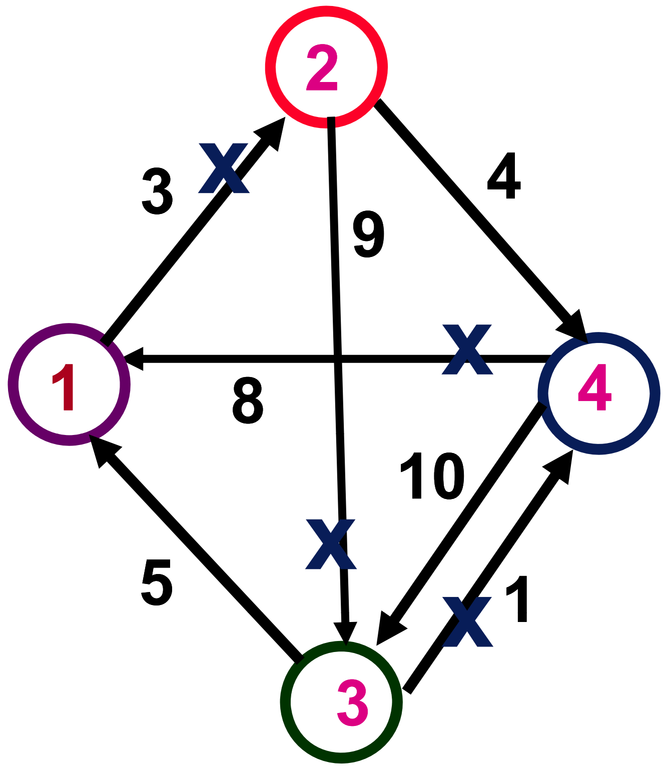
arc (*i*, *j*): edge (n + i, j)



n = 4

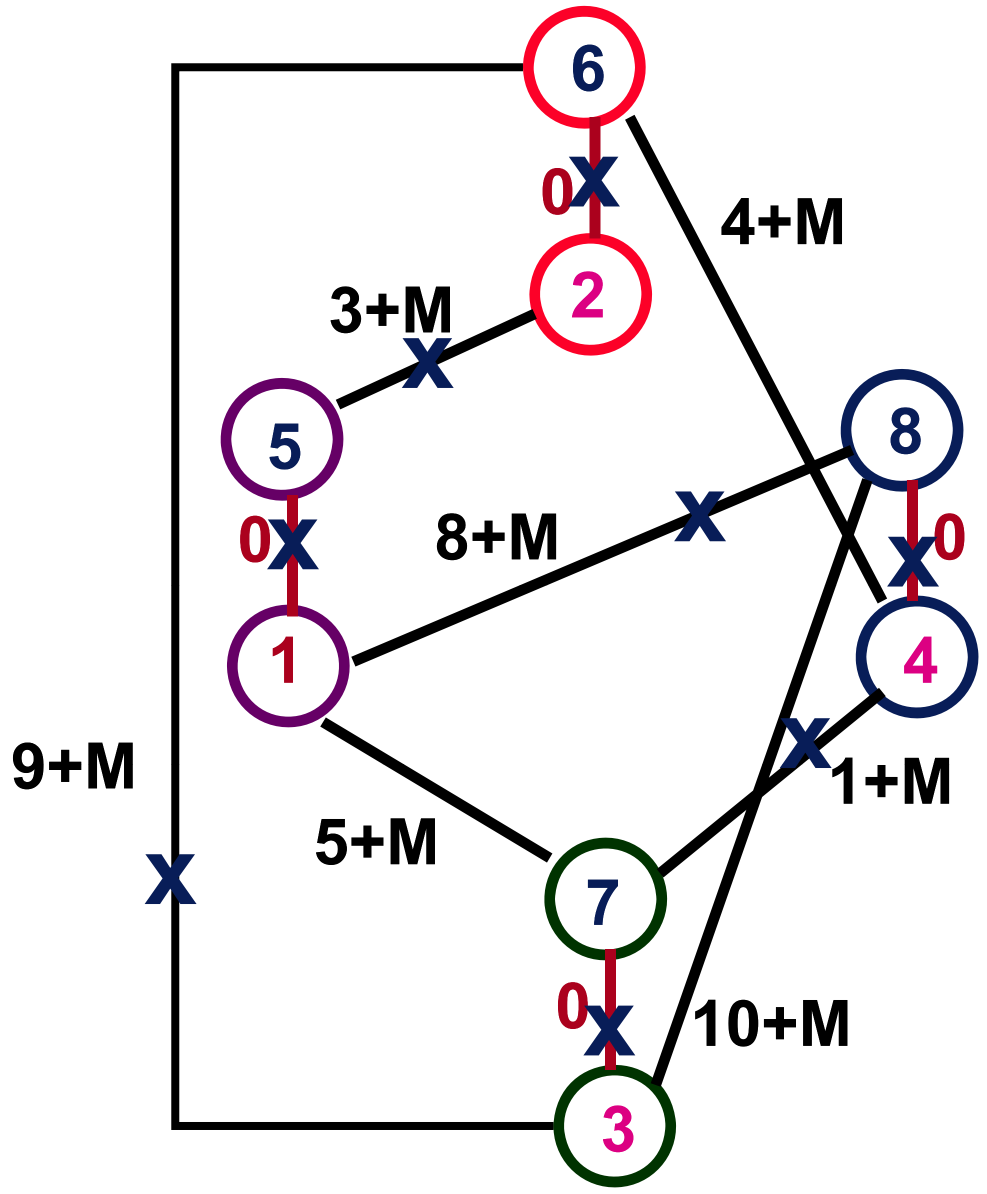


TRANSFORMATION OF ATSP INTO STSP



Optimal solution

Cost = 21



Cost = 21 + 4M

ADDITIONAL COMPUTATIONAL RESULTS

* Algorithms:

- FT92 : Fischetti-T. (Additive Bounding Procedure) (Math. Program.1992);
- FLT03: Fischetti-Lodi-T. (B. & C.) (Man. Sc. 1997, LNCS Springer 2003).
- CONCORDE (transformation from ATSP to STSP): Applegate, Bixby, Chvatal, Cook 1999.

- DIGITAL ALPHA 533 MHz for FT92 and FLT03 (times in seconds).
- Pentium IV 3 GHz for CONCORDE (times in seconds, scaled w.r.t. DIGITAL ALPHA).

- LP Solver: CPLEX 6.5.3 for FLT03.
- LP Solver: CPLEX 10.0 for CONCORDE.

	CONCORDE			FT92			FLT03		
Name	%Gap	Nodes	Time	%Gap	Nodes	Time	%Gap	Nodes	Time
ran500.a	0.00	1	18.1	0.03	11	2.2	0.00	3	7.3
ran500.b	0.00	15	30.8	0.01	24	1.4	0.00	17	23.5
ran500.c	0.04	93	71.5	0.08	35	1.7	0.04	61	54.4
ran1000.a	0.00	13	133.4	0.02	82	3.2	0.00	12	14.5
ran1000.b	0.01	43	403.5	0.07	99	6.8	0.01	89	73.6
ran1000.c	0.01	411	3417.2	0.03	65	3.7	0.01	37	54.8
rbg323	0.00	3	21.7	0.00	1	0.3	0.00	1	0.4
rbg443	0.00	3	81.2	0.00	1	1.2	0.00	1	1.4
p43	0.16	11	26.3	0.37	321417	T. L.	0.16	141	9.3
ry48p	0.53	5	32.0	2.94	22825	20.3	0.53	7	0.8
uk66	1.47	13	53.9	31.71	50005	152.1	2.29	47	5.7
balas108	2.63	521	1931.1	9.87	224537	T. L.	1.97	267	89.0
balas160	1.26	479	6250.4	11.34	114938	T. L.	1.26	737	671.1
ft53	0.00	1	1.8	1.56	131	0.2	0.00	1	0.1
ft70	0.01	3	4.3	0.57	220	0.3	0.02	5	0.3
kro124p	0.00	1	10.6	2.73	18909	135.7	0.04	3	1.0
ftv47	0.62	5	16.0	3.49	1026	0.4	1.01	11	0.5
ftv150	0.00	3	22.6	3.03	2778	17.0	0.27	21	2.6