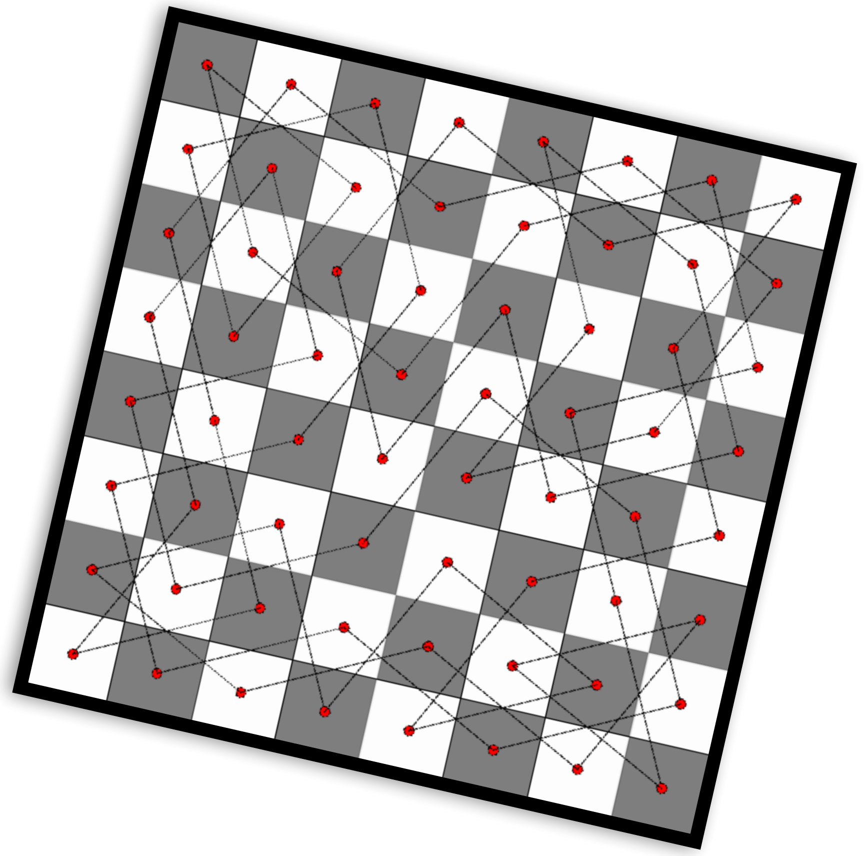
Lagrangian Relaxation

Pierre Schaus



Outline

- Lagrangian Relaxation: A quite generic technique to compute lower bounds

- Application to
  - Resource Constrained Shortest Path Problems (RCSPP)
  - The TSP (your favorite problem)

The Lagrangian relax intuition first

- Hard Problem:
  - Maximize obj
  - Subject to:
    - Constraint 1 + Constraint 2

- Is transformed into an easier problem and solving this problem gives a lower bound to initial problem
  - Maximize obj + λ_1 * violation(constraint 1)
  - Subject to:
    - Constraint 2

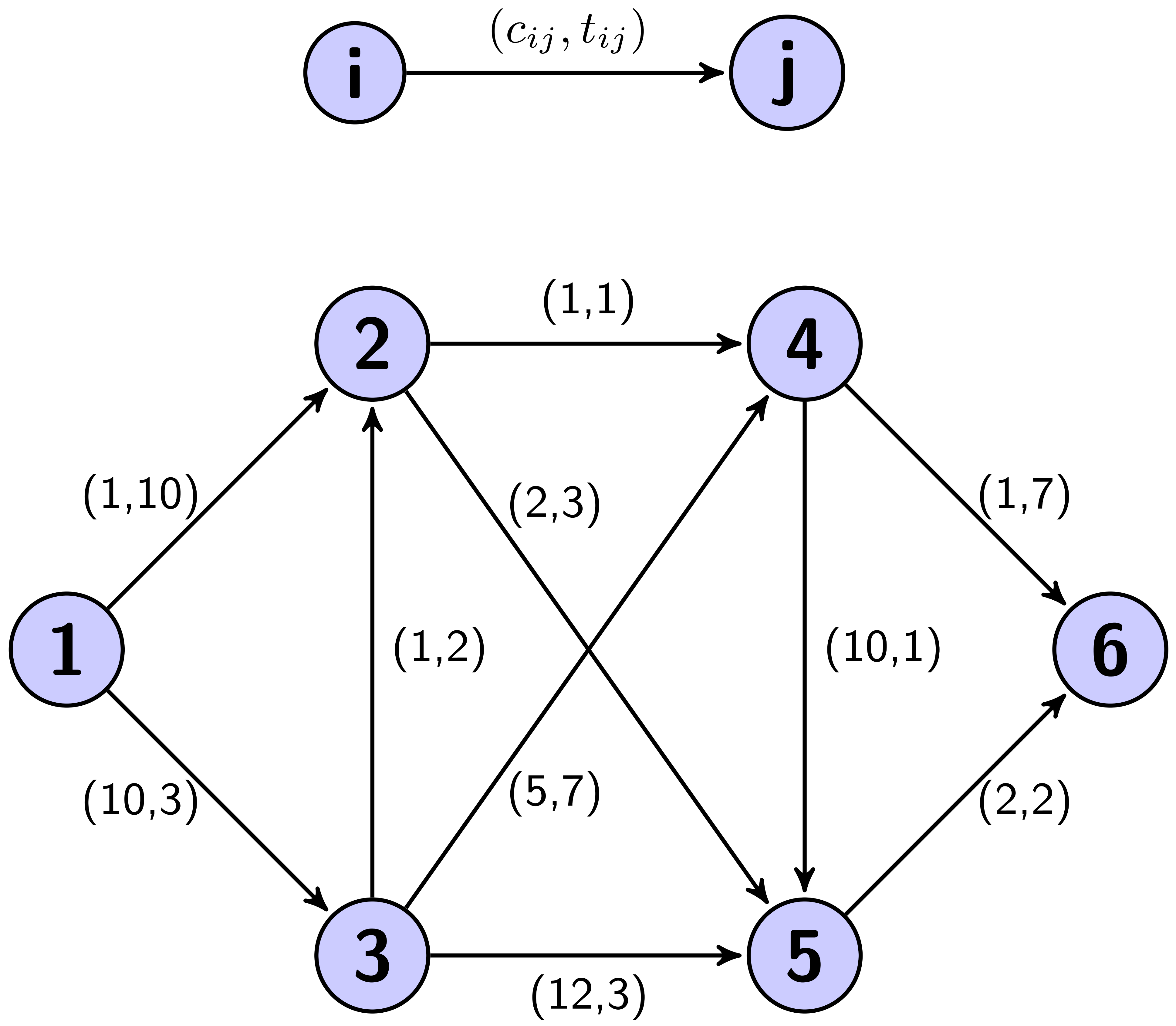
Constrained Shortest Path (our hard problem)

$$\min \sum_{(i,j)\in A} c_{ij} \cdot x_{ij}$$

subject to: $\quad$ flow conservation

$$\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} \leq T$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A.$$

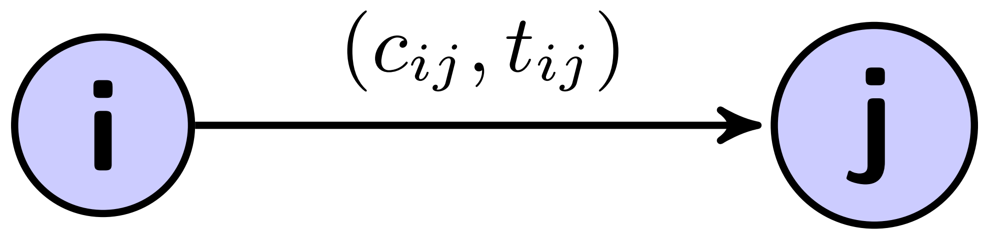


- Example: Minimize distance with time constraint
- NP-Hard Problem!

Constrained Shortest Path

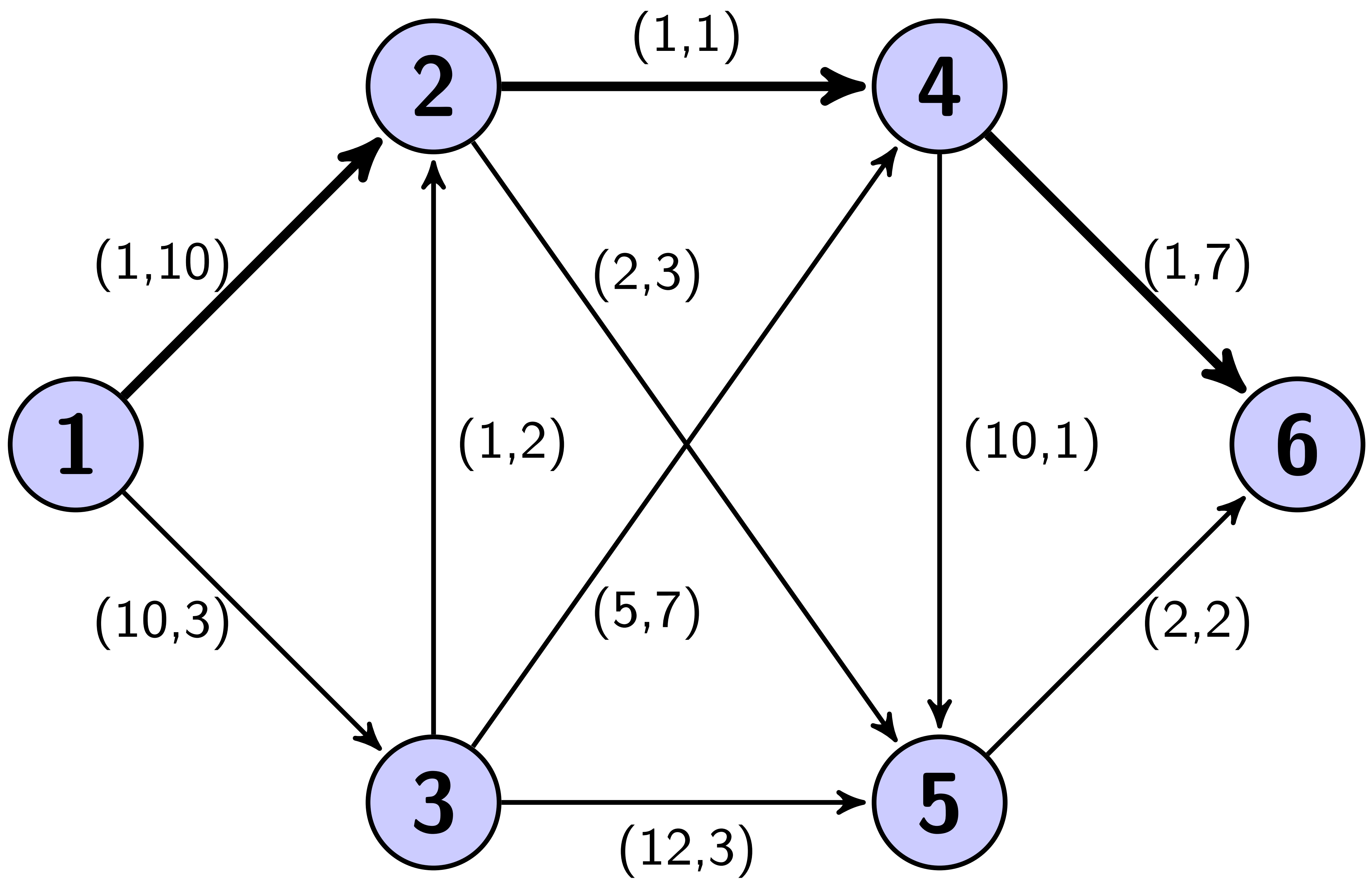
For a given path P, let

- c_p denote its path cost,
- t_p denote its path time



Example

- P = 1-2-4-6
- $c_p = 3$
- $t_p = 18$



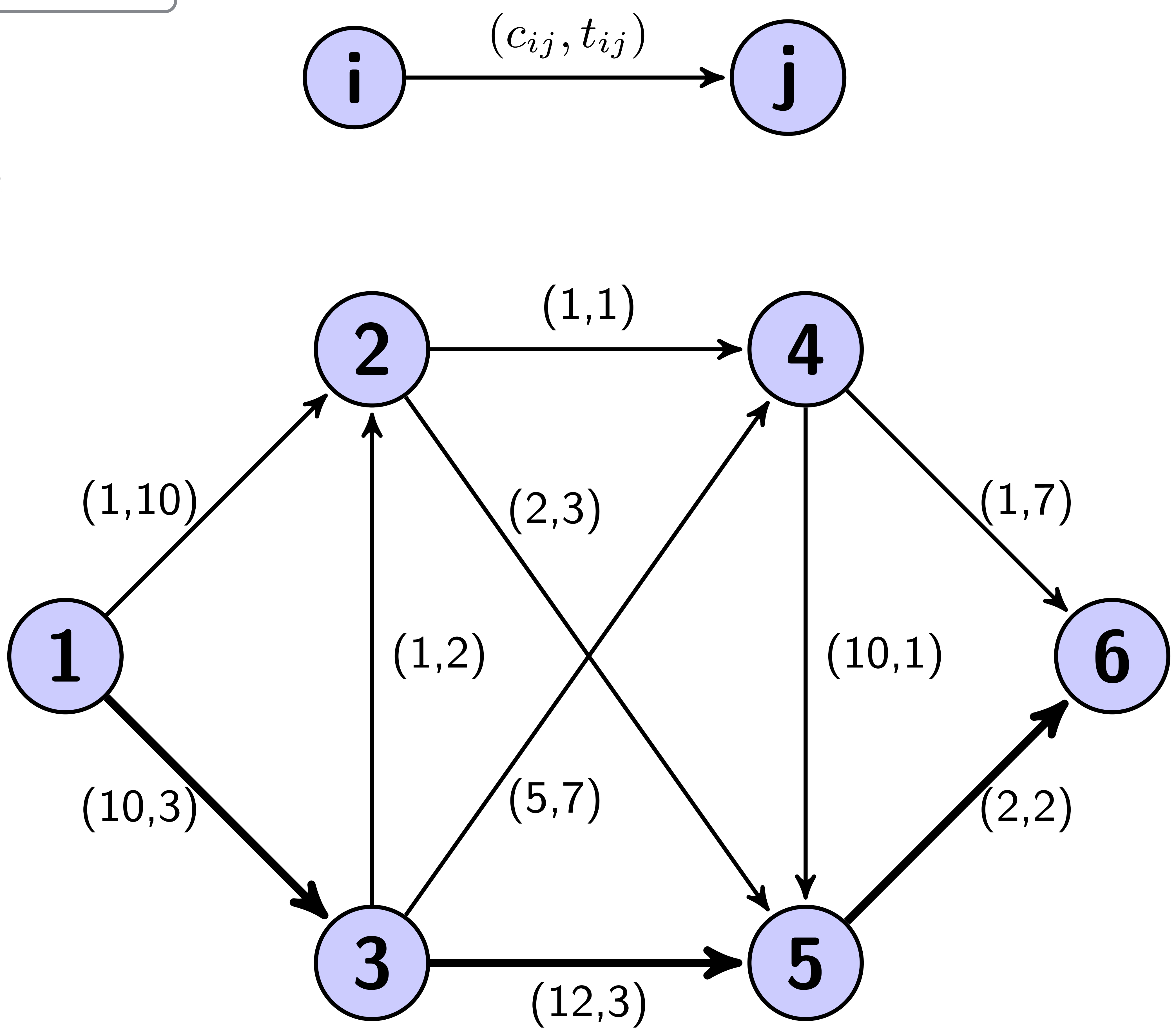
Example: Feasible Solution

objective

$$\sum_{(i,j)\in A} c_{ij} \cdot x_{ij} = 24$$

$$\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} = 8 \leq T = 10$$

path is feasible



Observation 1

- Without the resource constraint, is the problem is easy?

$$\min \sum_{(i,j)\in A} c_{ij} \cdot x_{ij}$$

$$\text{flow conservation}$$

$$\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} \leq T$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A.$$

Observation 2

- This is thus a lower-bound on the initial problem

Is this term is positive or negative ?

$$\min \sum_{(i,j)\in A} c_{ij} \cdot x_{ij} + \lambda\left(\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} - T\right)$$

$$\text{flow conservation}$$

$$\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} \leq T$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A$$

$$\lambda \geq 0$$

Observation 3

Is the optimum value to this problem also a lower bound?

$$\min \sum_{(i,j)\in A} c_{ij} \cdot x_{ij} + \lambda\left(\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} - T\right)$$

$$\text{flow conservation}$$

$$\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} \leq T$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A$$

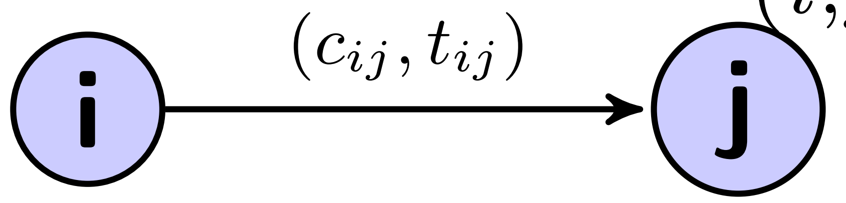
$$\lambda \geq 0$$

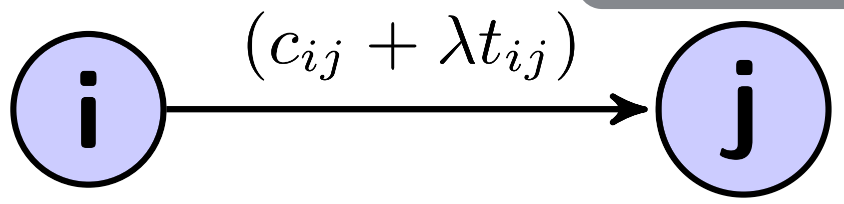
Example: Lower Bound (LB) Computation

$$\mathcal{L}(\lambda) = \min \sum_{(i,j)\in A} c_{ij} \cdot x_{ij} + \lambda\Big(\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} - T\Big)$$

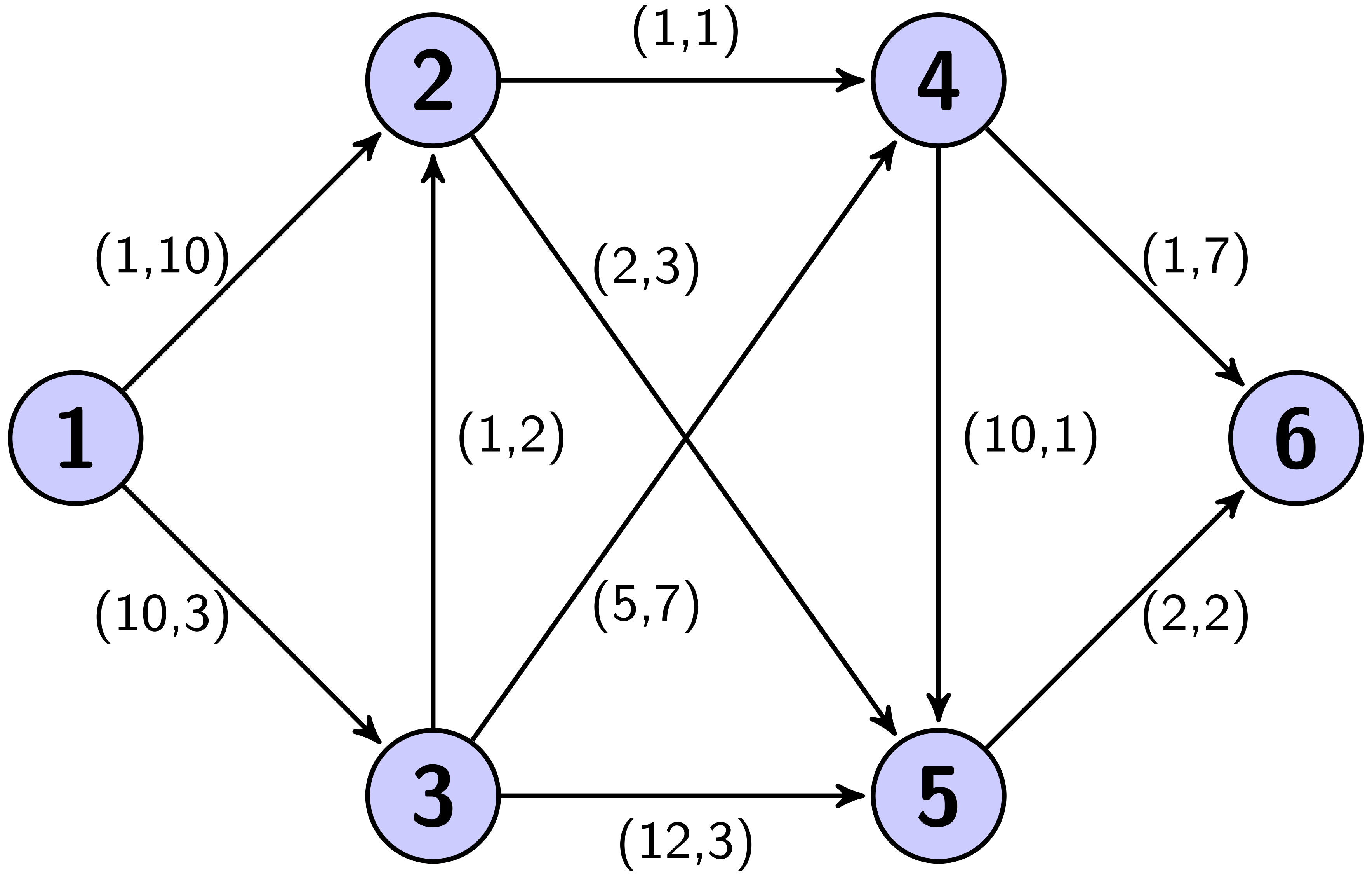
$$= \min \sum_{(i,j)\in A} (c_{ij} + \lambda t_{ij}) \cdot x_{ij} - \lambda T$$

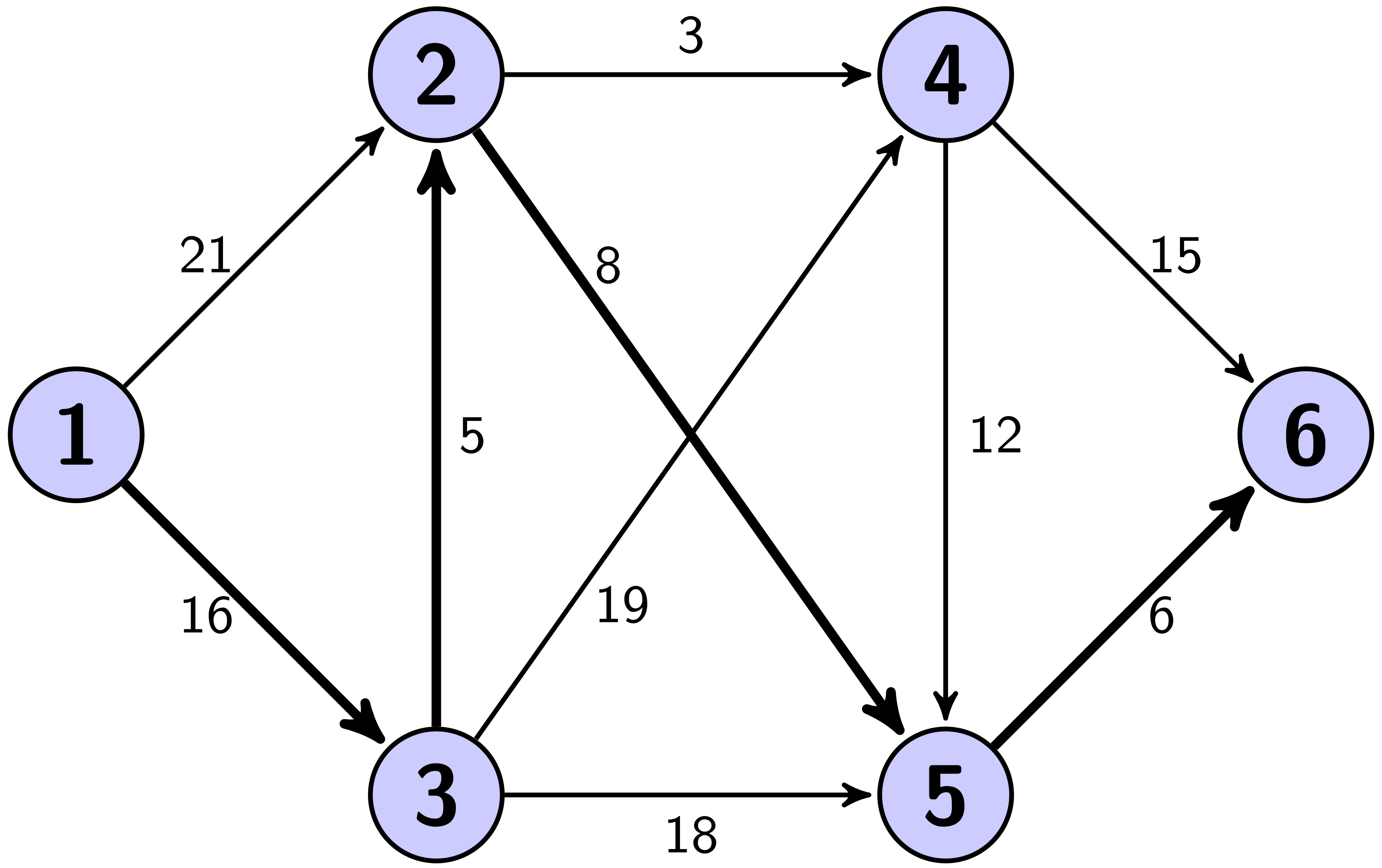
$T = 10$





$\mathcal{L}(\lambda = 2) = 35 - 2 \cdot 10 = 15$

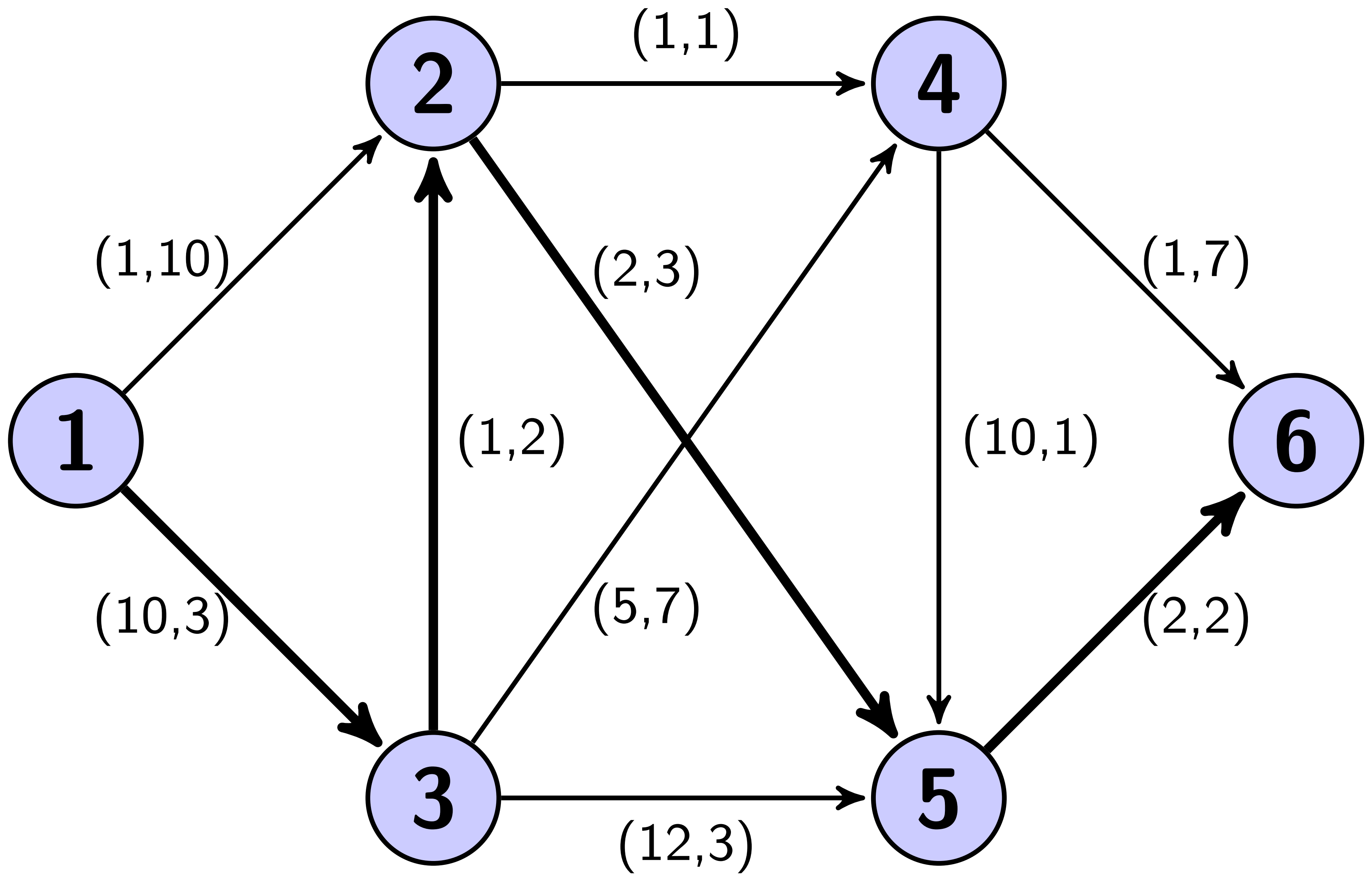




For a given value of λ, the lower bound is easily computed as a simple shortest path problem (Dijkstra algo).

Using LB to proof optimality of candidate sol.

- Is this (particular) path optimal knowing that:
  - 15 = $\mathscr{L}(\lambda = 2)$



- Why do we do all this?
  - Only to get a good lower-bound. We are actually looking after the best possible one $\max_{\lambda} \mathscr{L}(\lambda)$

Objective: Compute best LB

The problem is now to find λ leading to the optimal lower bound

$$\mathcal{L}^* = \max_{\lambda} \left(\min \sum_{(i,j)\in A} (c_{ij} \cdot x_{ij}) - \lambda(\sum_{(i,j)\in A} (t_{ij} \cdot x_{ij}) - T) \right)$$

$$\text{flow conservation}$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A$$

$$\lambda \geq 0$$

Called Lagrangian Dual

For a given value of λ, the lower bound is easily computed as a simple shortest path problem (Dijkstra algo).

The Brute force approach

$$\mathcal{L}^* = \max_{\lambda} \left(\min \sum_{(i,j)\in A} (c_{ij} \cdot x_{ij}) - \lambda(\sum_{(i,j)\in A} (t_{ij} \cdot x_{ij}) - T) \right)$$

feasible paths

flow conservation

$x_{ij} \in \{0,1\}, \forall (i,j) \in A$

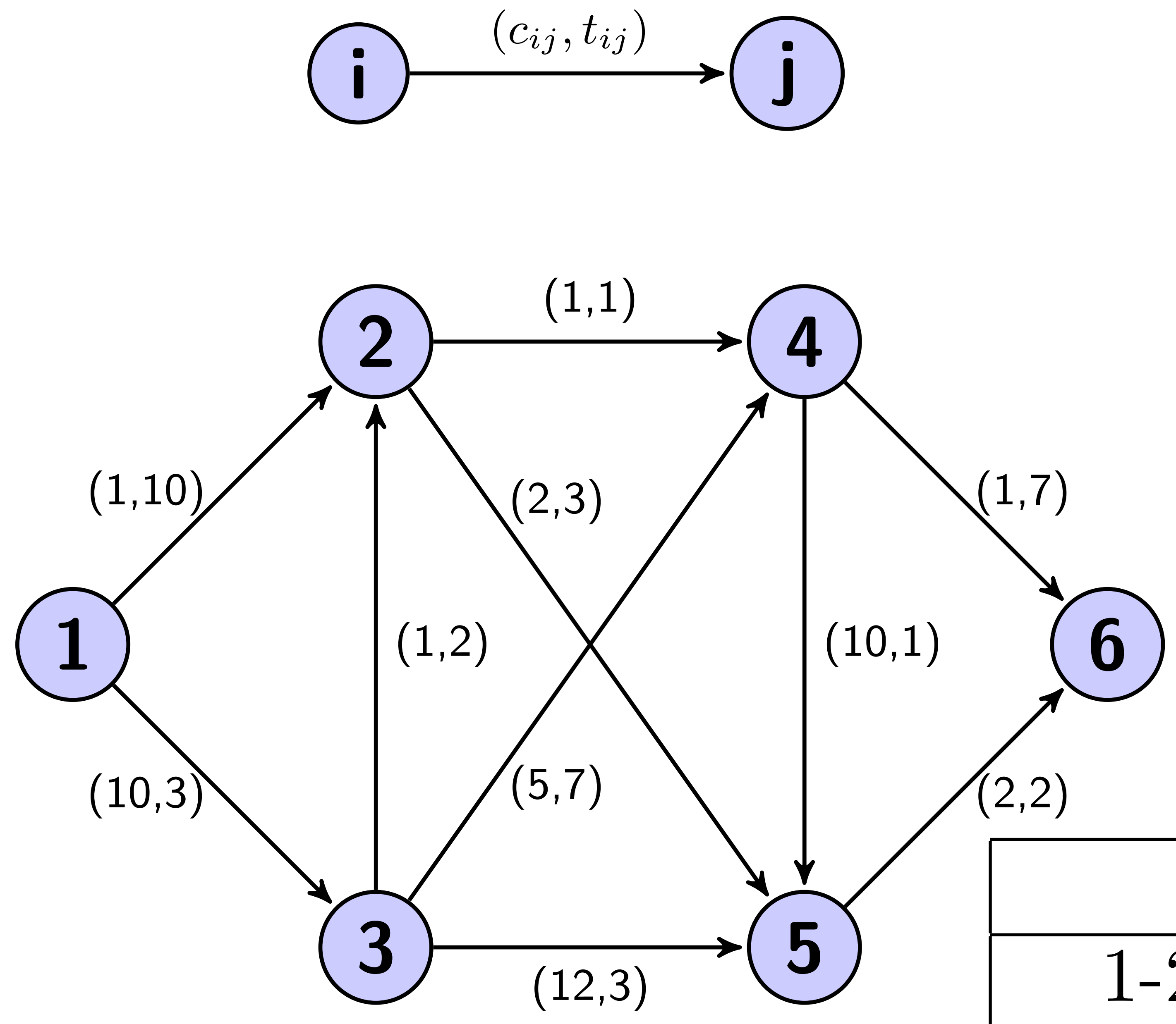
$\lambda \geq 0$

- formulate the minimization problem as a minimization over the set of all the feasible paths $\mathcal{P}$:

$$\mathcal{L}^* = \max_{\lambda} (\min\{c_P + \lambda(t_P - T) : P \in \mathcal{P}\})$$

Is this solution practical?

Brute force example (for a fixed λ)



$T = 14$

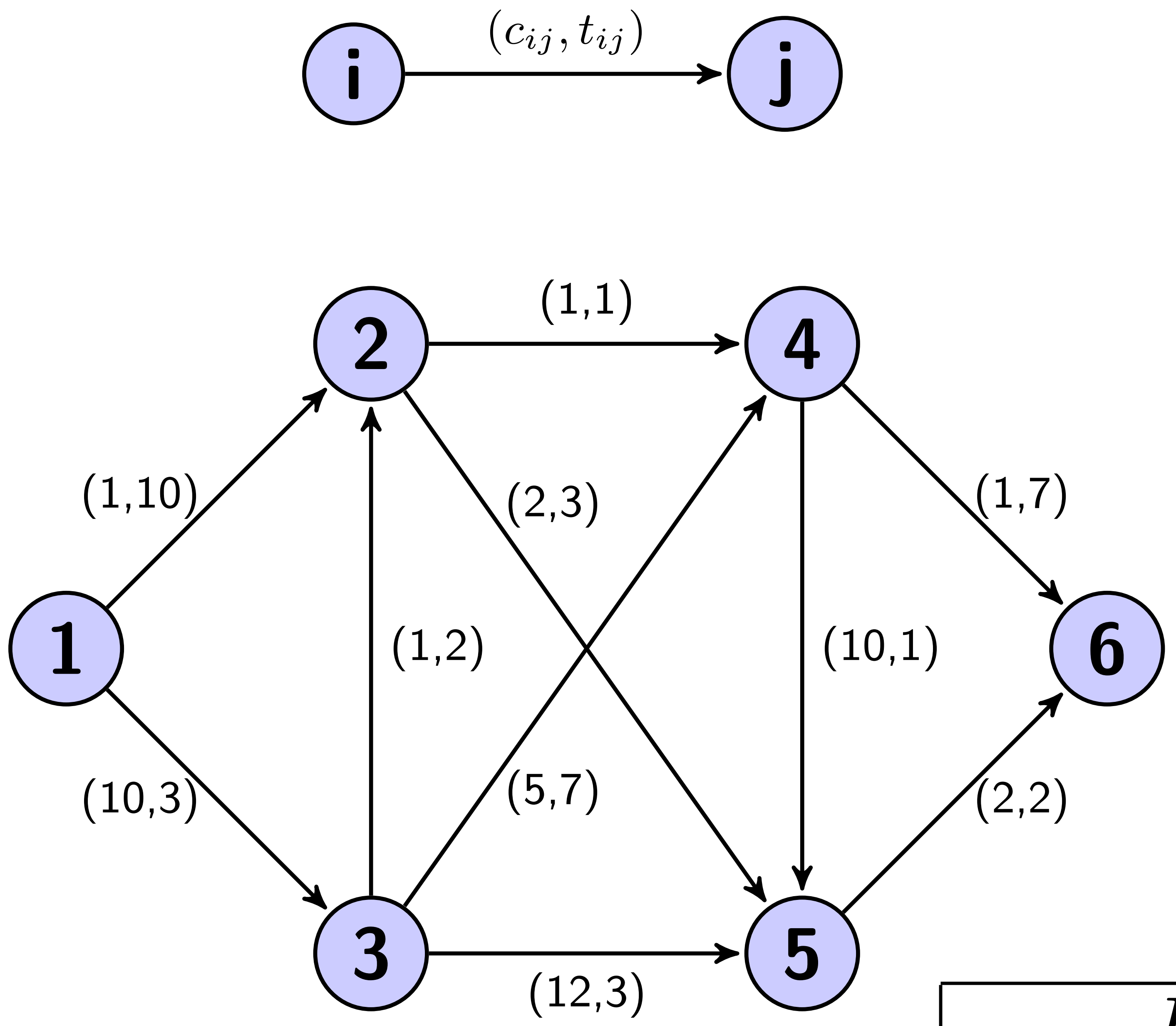
$$\mathcal{L}^* = \max_{\lambda} \left(\min\{ c_P + \lambda(t_P - T) : P \in \mathcal{P} \} \right)$$

all possible feasible paths

P	c_P	t_P	$c_P + \lambda(t_P - T)$	$c_P + 2(t_P - T)$
1-2-4-6	3	18	$3 + 4\lambda$	11
1-2-5-6	5	15	$5 + \lambda$	7
1-2-4-5-6	14	14	14	14
1-3-2-4-6	13	13	$13 - \lambda$	11
1-3-2-5-6	15	10	$15 - 4\lambda$	7
1-3-2-4-5-6	24	9	$24 - 5\lambda$	14
1-3-4-6	16	17	$16 + 3\lambda$	22
1-3-4-5-6	27	13	$27 - \lambda$	25
1-3-5-6	24	8	$24 - 6\lambda$	12

What is the Lagrangian LB for lambda = 2?

Brute force example (for a fixed λ)



$T = 14$

$$\mathcal{L}^* = \max_{\lambda} \left(\min\{c_P + \lambda(t_P - T) : P \in \mathcal{P}\} \right)$$

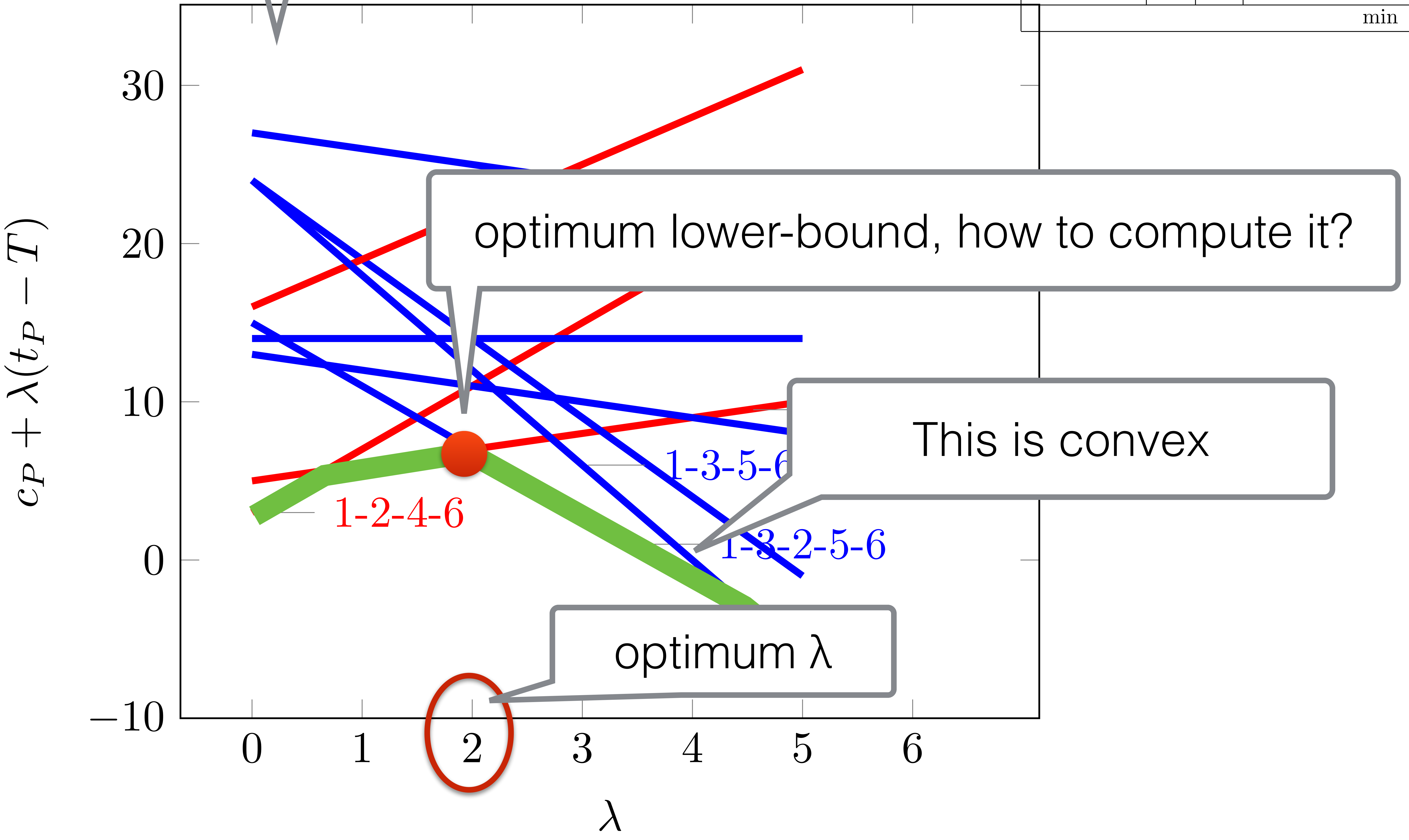
all possible feasible paths

P	c_P	t_P	$c_P + \lambda(t_P - T)$	$c_P + 2(t_P - T)$
1-2-4-6	3	18	$3 + 4\lambda$	11
1-2-5-6	5	15	$5 + \lambda$	7
1-2-4-5-6	14	14	14	14
1-3-2-4-6	13	13	$13 - \lambda$	11
1-3-2-5-6	15	10	$15 - 4\lambda$	7
1-3-2-4-5-6	24	9	$24 - 5\lambda$	14
1-3-4-6	16	17	$16 + 3\lambda$	22
1-3-4-5-6	27	13	$27 - \lambda$	25
1-3-5-6	24	8	$24 - 6\lambda$	12
			min	7

Finding the optimum λ (visual representation)

Every feasible path is a line, for each λ, the lower bound is the minimum value of all the paths (piecewise linear convex function). The goals is to find λ that maximises this function to find the strongest possible lower bound

P	c_P	t_P	$c_P + \lambda(t_P - T)$	$c_P + 2(t_P - T)$
1-2-4-6	3	18	$3 + 4\lambda$	11
1-2-5-6	5	15	$5 + \lambda$	7
1-2-4-5-6	14	14	14	14
1-3-2-4-6	13	13	$13 - \lambda$	11
1-3-2-5-6	15	10	$15 - 4\lambda$	7
1-3-2-4-5-6	24	9	$24 - 5\lambda$	14
1-3-4-6	16	17	$16 + 3\lambda$	22
1-3-4-5-6	27	13	$27 - \lambda$	25
1-3-5-6	24	8	$24 - 6\lambda$	12
			min	7



Solution 1: Linear Programming

Computing the optimum λ with linear programming (simplex)

P	c_P	t_P	$c_P + \lambda(t_P - T)$	$c_P + 2(t_P - T)$
1-2-4-6	3	18	$3 + 4\lambda$	11
1-2-5-6	5	15	$5 + \lambda$	7
1-2-4-5-6	14	14	14	14
1-3-2-4-6	13	13	$13 - \lambda$	11
1-3-2-5-6	15	10	$15 - 4\lambda$	7
1-3-2-4-5-6	24	9	$24 - 5\lambda$	14
1-3-4-6	16	17	$16 + 3\lambda$	22
1-3-4-5-6	27	13	$27 - \lambda$	25
1-3-5-6	24	8	$24 - 6\lambda$	12
			min	7

$$\mathcal{L}^* = \max_{\lambda} \left(\min\{c_P + \lambda(t_P - T) : P \in \mathcal{P}\}\right)$$

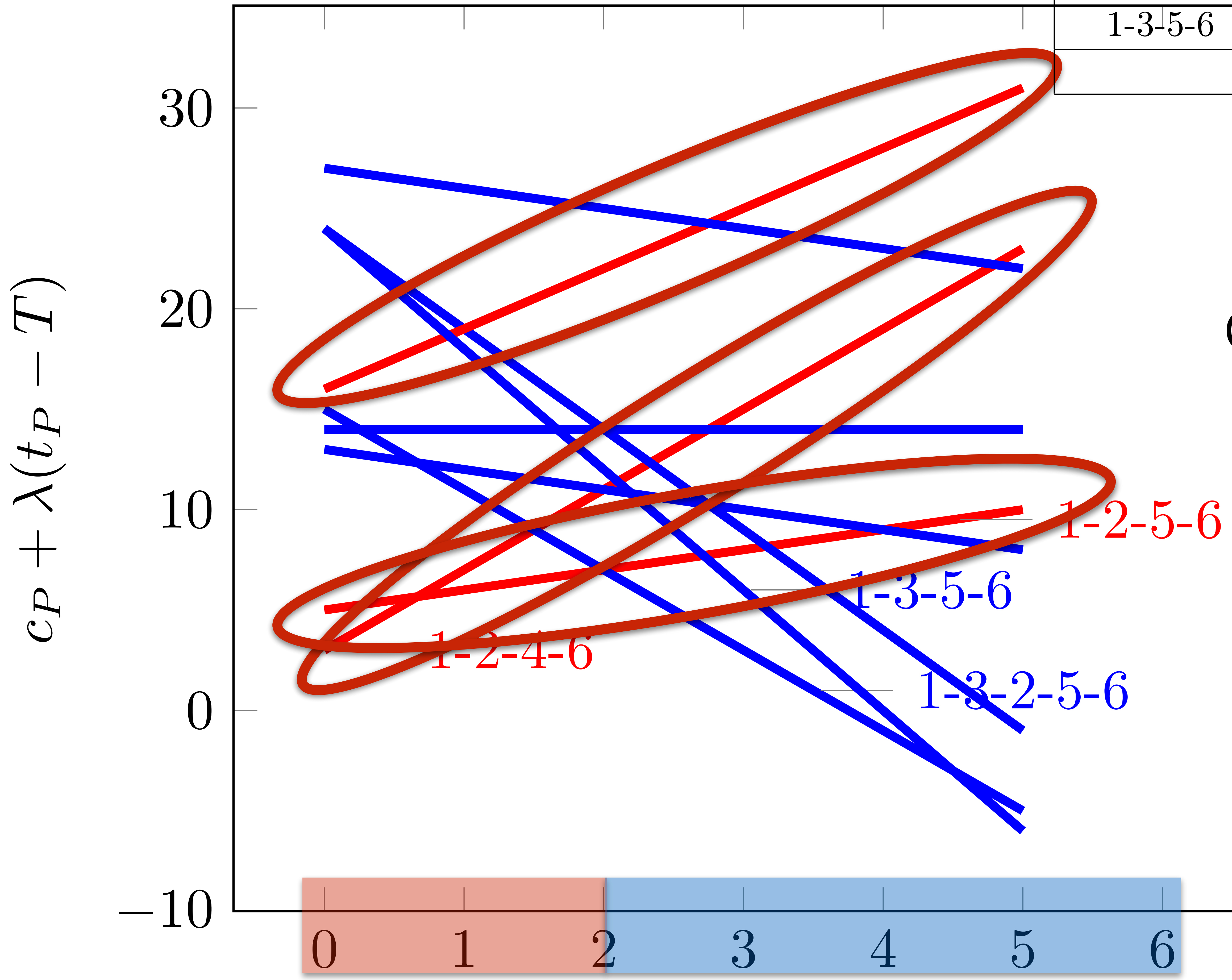
$$= \max z$$

$$\text{subject to :} \quad z \leq c_P + \lambda(t_P - T) \ , \forall P \in \mathcal{P}$$

It is a linear program but with an exponential number of constraints (one for each path) thus impracticable.

Solution2: Subgradient Algorithm

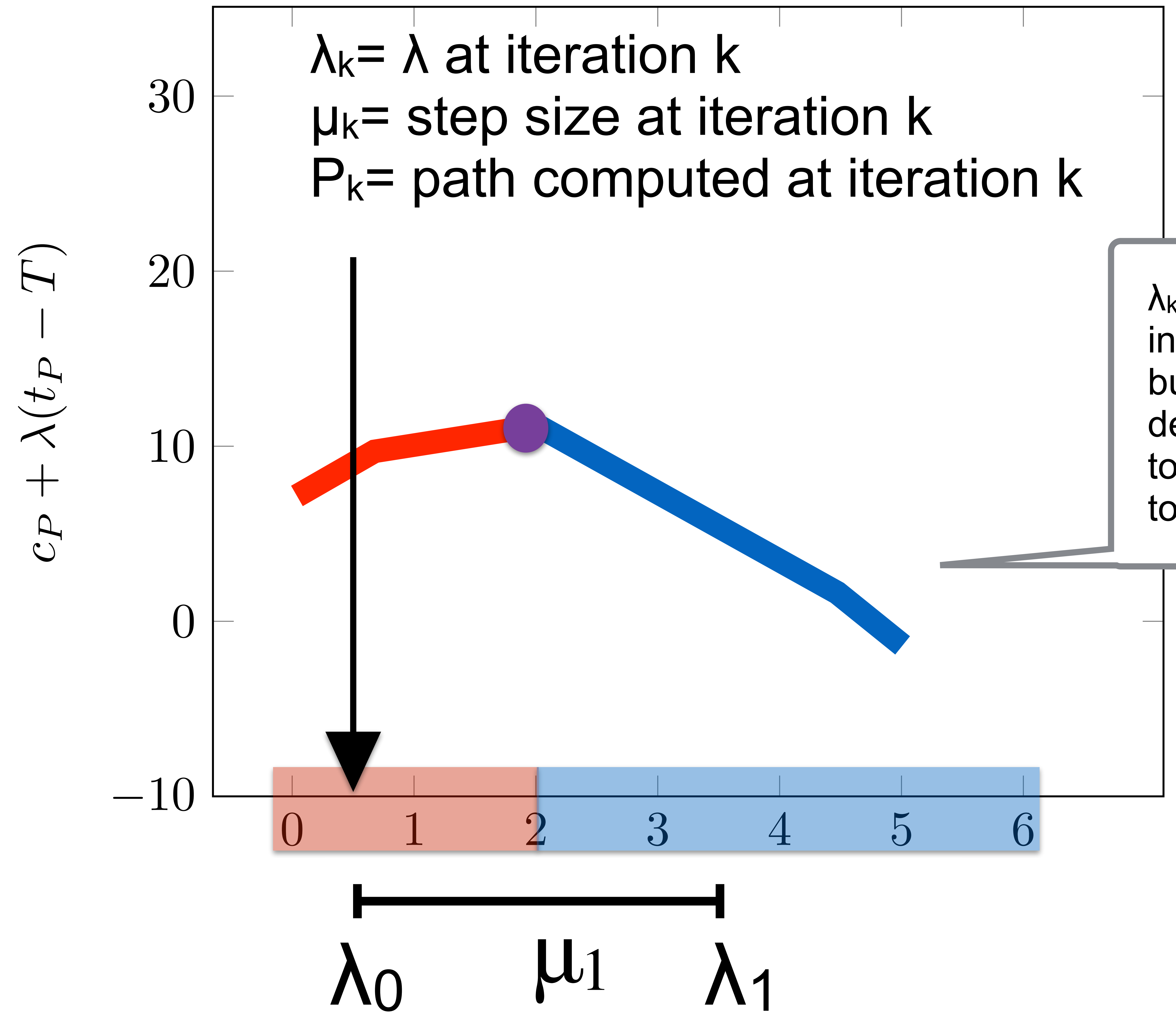
P	c_P	t_P	$c_P + \lambda(t_P - T)$	$c_P + 2(t_P - T)$
1-2-4-6	3	18	$3 + 4\lambda$	11
1-2-5-6	5	15	$5 + \lambda$	7
1-2-4-5-6	14	14	14	14
1-3-2-4-6	13	13	$13 - \lambda$	11
1-3-2-5-6	15	10	$15 - 4\lambda$	7
1-3-2-4-5-6	24	9	$24 - 5\lambda$	14
1-3-4-6	16	17	$16 + 3\lambda$	22
1-3-4-5-6	27	13	$27 - \lambda$	25
1-3-5-6	24	8	$24 - 6\lambda$	12
			min	7



violates time constraint (≤14)

Subgradient

- Sub-gradient Algorithms: Idea is to move λ to the right when on the red area, to the left when on the blue area.



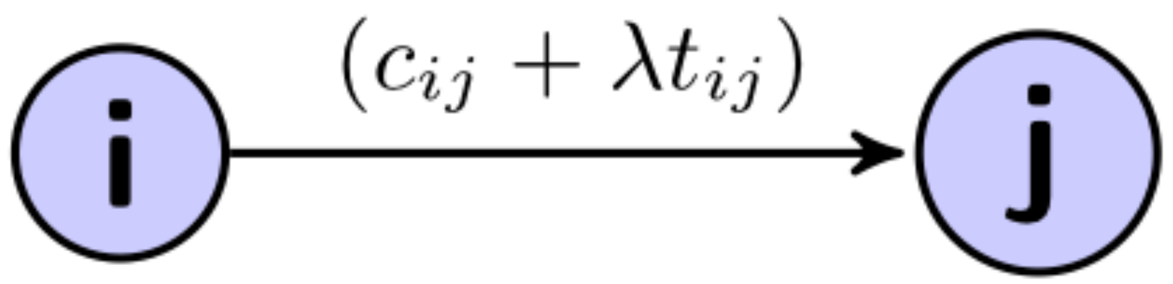
λ_k has no guarantee to increase at each iteration but if the step size is μ_k decreasing enough but not too much, we will converge toward the optimum

Computing the optimum λ: subgradient optim

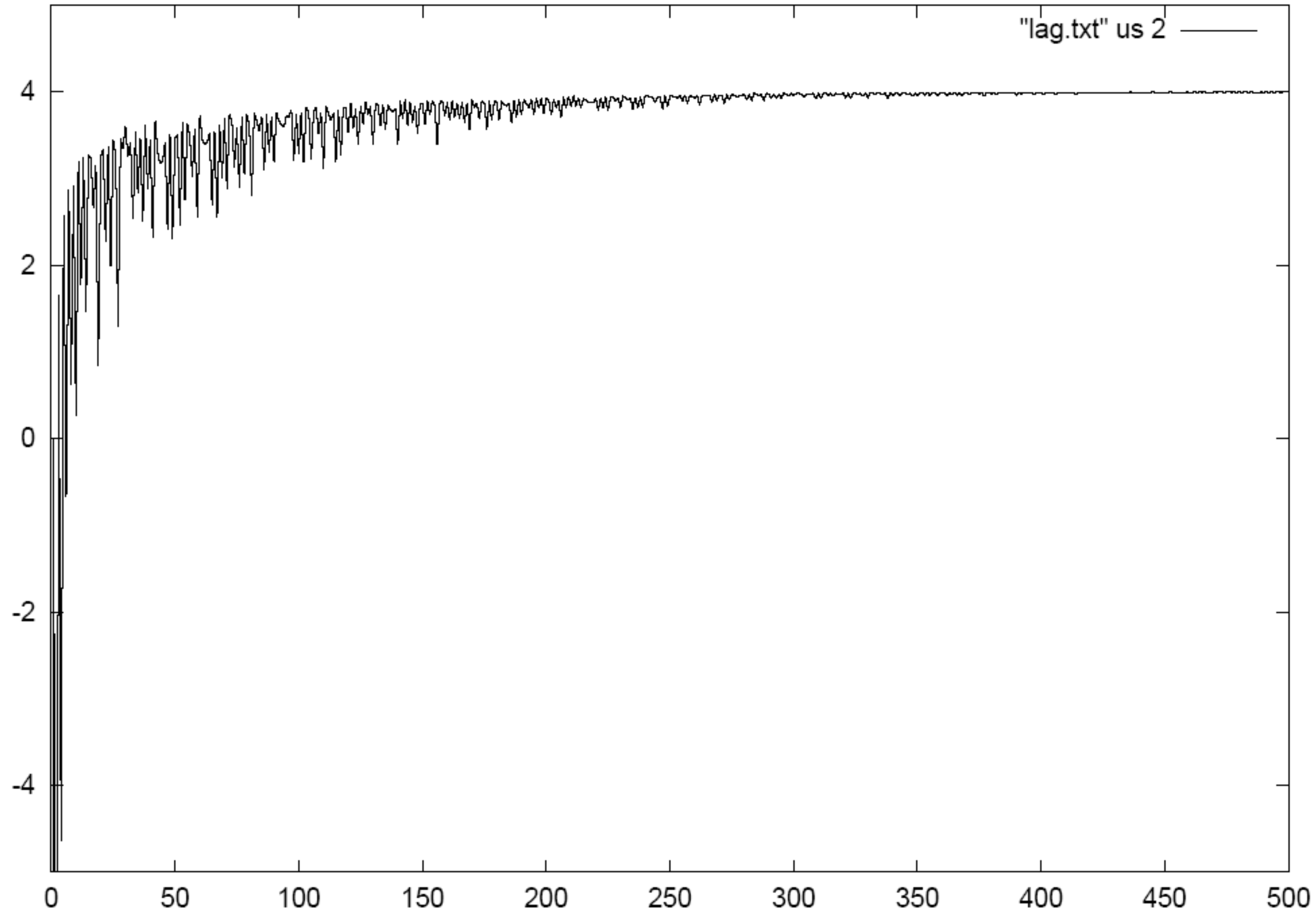
- Convergence guarantee if $\mu_k \rightarrow 0$ and $\sum_{k=1}^{\infty} \mu_k \rightarrow \infty$

- Note that $\mathscr{L}_k$ (Lagrangian LB) has no guarantee to increase at each step

- At iteration k if P_k violates time constraint, increase λ, otherwise decrease it.
- $\lambda_{k+1} = \max(0, \lambda_k + \mu_k(t_{P_k} - T))$
- $\mu_{k+1} = 1/k$
- $\lambda_0 = 0$



typical lower bound evolution along iterations, no guarantee to monotonically increase at each step



Constrained Shortest Path Algorithm

Result: A lower bound $\mathcal{L}*$ and a potentially good (not proven optimal) feasible candidate path $P*$

$\mathcal{L}* \leftarrow -\infty, k \leftarrow 0,\ \mu_0 = 1,\ \lambda_0 = 0$
$P* \leftarrow$ shortest path using weights t_{ij}
if $(t_{P*} > T)$ **then**
    return the problem is unfeasible
end
while $\mu \geq \epsilon$ **do**
    Compute shortest path P_k using weights $c_{ij} + \lambda_k t_{ij}$
    $\mathcal{L}_k \leftarrow c_{P_k} + \lambda_k (t_{P_k} - T)$
    if $\mathcal{L}_k \geq \mathcal{L}*$ **then**
        $\mathcal{L}* \leftarrow \mathcal{L}_k$
        if P_k is feasible **then**
            $P* \leftarrow P_k$
        end
    end
    Update λ_k and μ_k
    $k \leftarrow k + 1$
end

It has not guarantee to find the best one. But we have a lower-bound at the end thus we can compute the « gap »: $(c_{P_*} - \mathcal{L}*)/\mathcal{L}*$

The gap should be non decreasing

For our problem

$$\mathcal{L}^* = \max_{\lambda} \left(\min\{c_P + \lambda(t_P - T) : P \in \mathcal{P}\}\right)$$

$$= \max z$$

$$\text{subject to}: \; z \leq c_P + \lambda(t_P - T) \;, \forall P \in \mathcal{P}$$

- The sub gradient method is over-complex in this case because we only have one multiplier (but it is very useful because you generally have many lambda's)
- You can use a binary search instead to discover the optimum lambda.

How good is the Lagrangian relaxation LB?

As good as the linear relaxation:

$$\mathcal{L}* = \min \sum_{(i,j)\in A} c_{ij} \cdot x_{ij}$$

$$\text{flow conservation}$$

$$\sum_{(i,j)\in A} t_{ij} \cdot x_{ij} \leq T$$

$$x_{ij} \in [0,1], \forall (i,j) \in A$$

~~$x_{ij} \in \{0,1\}, \forall (i,j) \in A.$~~

But the linear relaxation will not give you feasible solutions during the process …

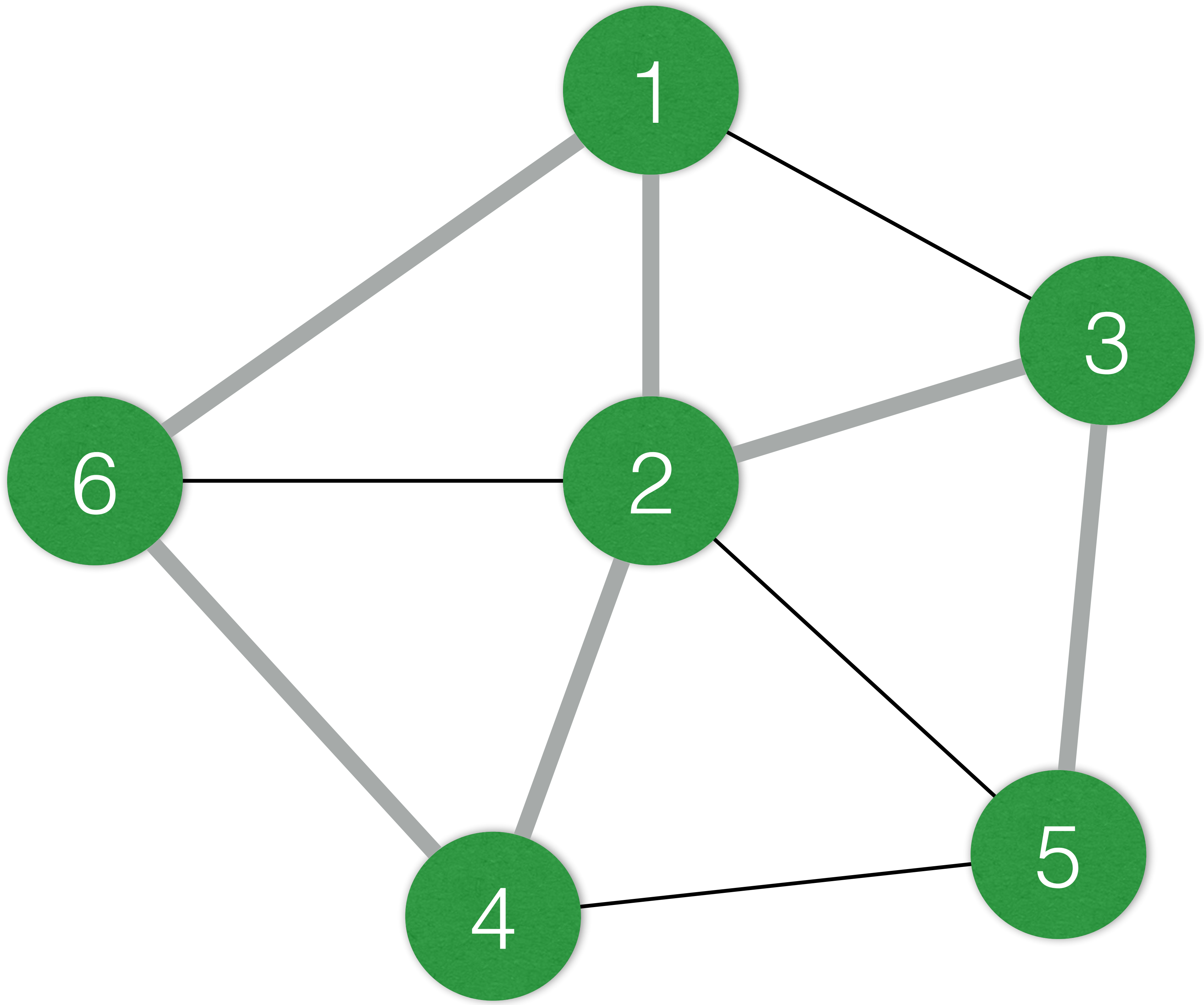
Lagrangian Relaxation for the TSP

- A TSP is a combination of two constraints
  - The degree of each node is exactly 2
  - The selected edges form a single connected component (otherwise sub tours are still possible)

- The two constraints can be relaxed
  - Minimum 1-Tree relaxation
  - Minimum Assignment Problem in a bipartite graph

One-Tree

- One-tree = spanning tree of subgraph {2,..n} + two edges connected to node 1



- In a weighted graph, we can find the minimum one-tree

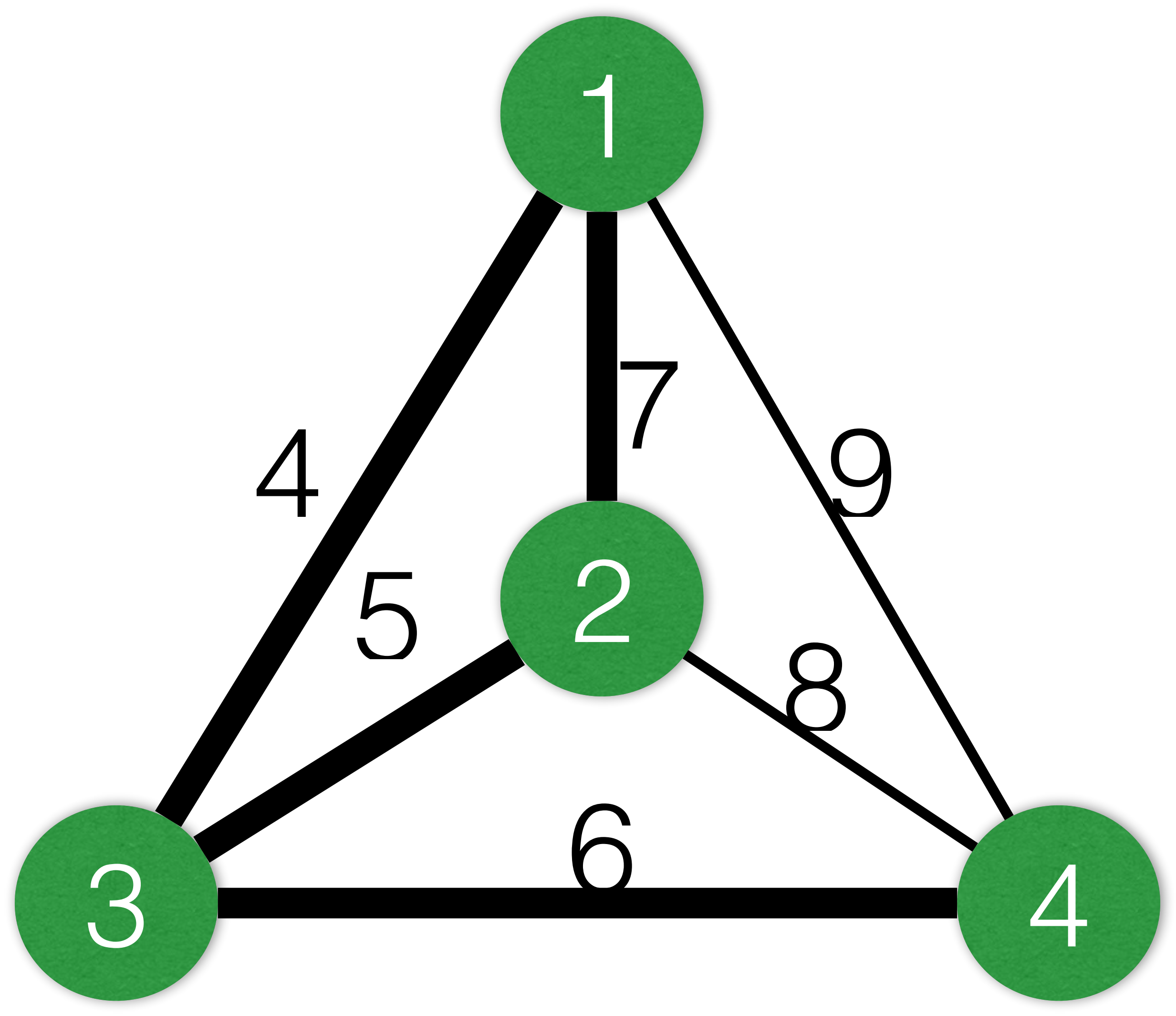
Minimum 1-Tree Relaxation

- On edges 1,…,n,

  1. Find the minimum spanning tree (MST) on {2,…,n}

  2. Reconnect node 1 with the two lightest edges

The result is a graph with exactly n edges and exactly one cycle, node 1 has a degree of 2 *but the degree of the other nodes is not necessarily 2*.



Observation

- Since a Hamiltonian circuit is a degree-constrained one-tree, this problem is completely equivalent to the minimum TSP

$$\min \sum_{e} x_e \cdot w_e$$

selected edges $\{e \mid x_e = 1\}$ form a 1-tree

$$\sum_{e \in \delta(i)} x_e = 2, \forall i$$

$$x_e \in \{0,1\}, \forall e$$

- And thus equally NP hard to solve, let's relax it …

Introducing multipliers …

- Add a zero term (introduce multipliers, one for each node)

∈ ℝ

0

$$\min \sum_{e} x_e \cdot w_e + \sum_{i} \pi_i \left(2 - \sum_{e \in \delta(i)} x_e\right)$$

selected edges $\{e \mid x_e = 1\}$ form a 1-tree

$$\sum_{e \in \delta(i)} x_e = 2, \forall i$$

$$x_e \in \{0,1\}, \forall e$$

… and then relaxing …

- Add a zero term (introduce multipliers, one for each node)

Lower bound since removing a constraint can only relax the problem!

$$\min \sum_e x_e \cdot w_e + \sum_i \pi_i (2 - \sum_{e \in \delta(i)} x_e)$$

selected edges $\{e \mid x_e = 1\}$ form a 1-tree

~~$\sum_{e \in \delta(i)} x_e = 2, \forall i$~~

$x_e \in \{0,1\}, \forall e$

Lagrangian Lower Bound

- Add a zero term (introduce multipliers, one for each node)

$$\mathscr{L}(\pi) = \min \sum_{e} x_e \cdot w_e + \sum_{i} \pi_i (2 - \sum_{e \in \delta(i)} x_e)$$

selected edges $\{e \mid x_e = 1\}$ form a 1-tree

$x_e \in \{0,1\}, \forall e$

Is this problem difficult? Let's see...

- And the goal is of course to maximize this lower-bound

$$\mathscr{L}^* = \max_{\pi} \mathscr{L}(\pi)$$

Lagrangian Lower Bound

- Add a zero term (introduce multipliers, one for each node)

$$\mathscr{L}(\pi) = \min \sum_{e} x_e \cdot w_e + \sum_{i} \pi_i (2 - \sum_{e \in \delta(i)} x_e)$$

selected edges $\{e \mid x_e = 1\}$ form a 1-tree

$x_e \in \{0,1\}, \forall e$

Is this problem difficult?
Let's see...

- And the goal is of course to maximize this lower-bound

$$\mathscr{L}^* = \max_{\pi} \mathscr{L}(\pi)$$

Lagrangian Lower Bound

- Add a zero term (introduce multipliers, one for each node)

$$\mathscr{L}(\pi) = \min \sum_{e} x_e \cdot w_e + \sum_{i} \pi_i (2 - \sum_{e \in \delta(i)} x_e)$$

selected edges $\{e \mid x_e = 1\}$ form a 1-tree

$x_e \in \{0,1\}, \forall e$

Is this problem difficult? Let's see…

- Can be rewritten as

modified weight   constant term

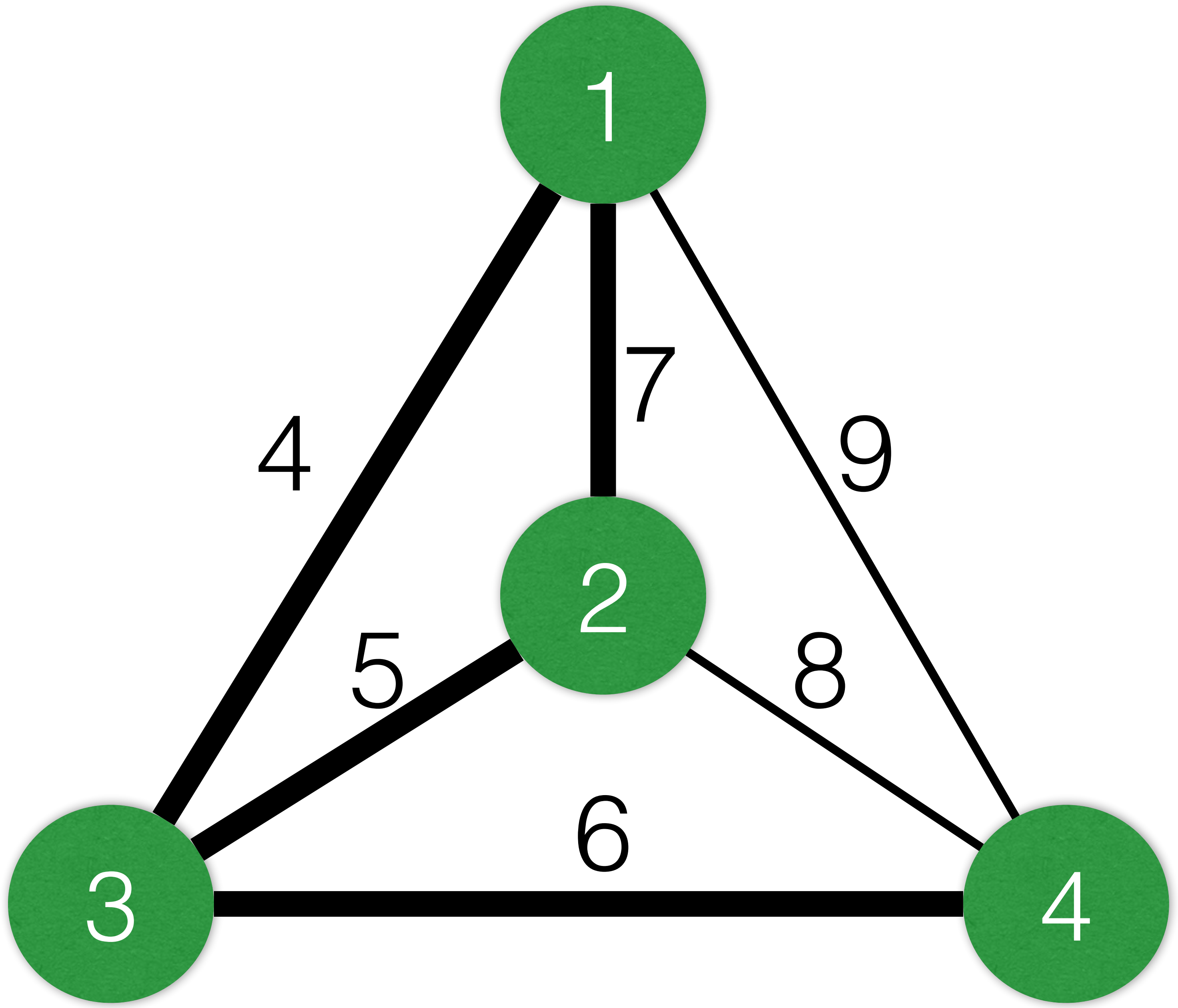
$$\mathscr{L}(\pi) = \min \sum_{e=\{i,j\}} x_e \cdot (w_e - \pi_i - \pi_j) + 2 \sum_{i} \pi_i$$

selected edges $\{e \mid x_e = 1\}$ form a 1-tree

$x_e \in \{0,1\}, \forall e$

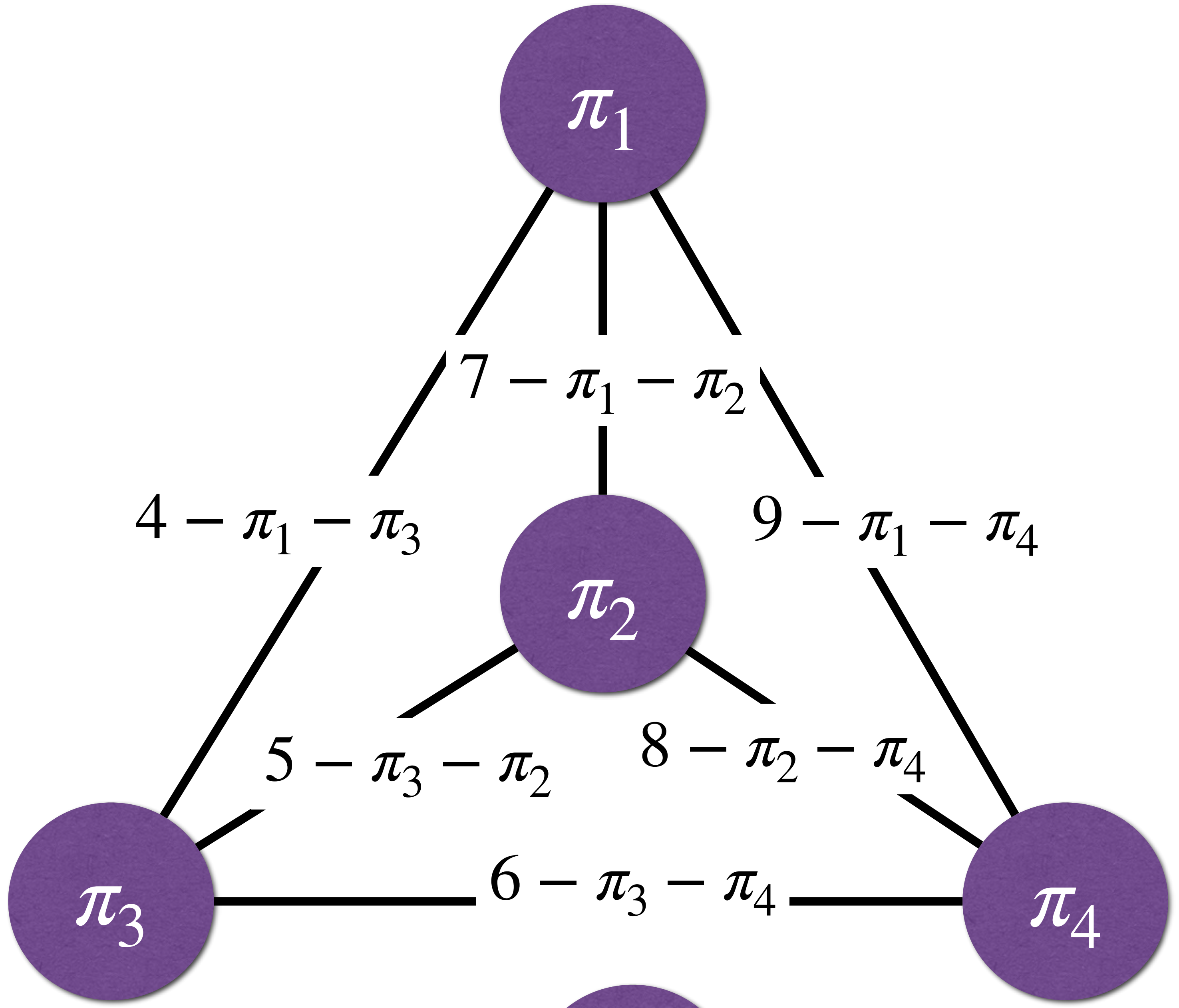
simple min one-tree

Example: Min One-Tree Lower-Bound



One tree lower-bound: 22

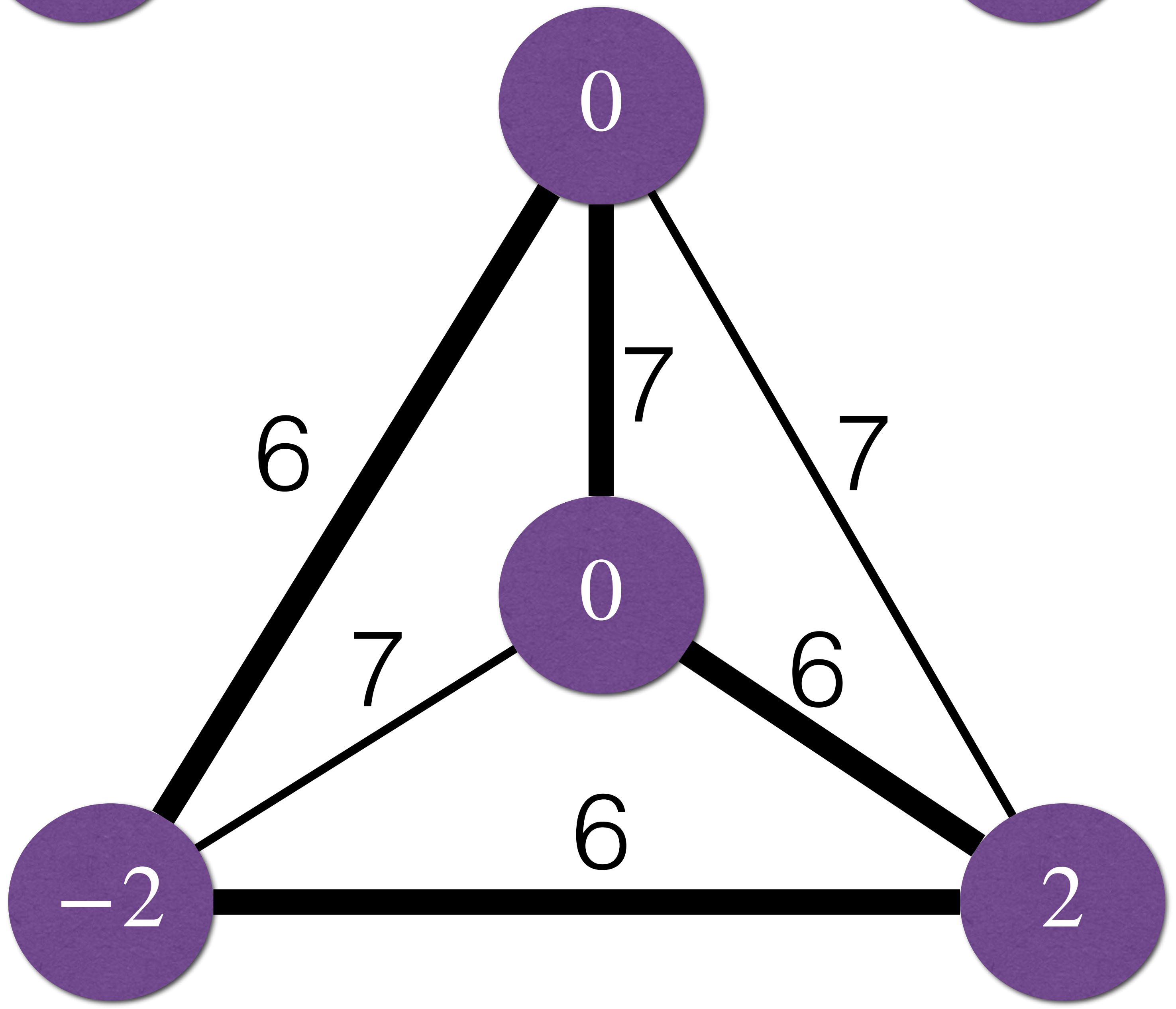
Example: Min One-Tree Lower-Bound



$$\mathscr{L}(\pi) = \min \sum_{e=\{i,j\}} x_e \cdot (w_e - \pi_i - \pi_j) + 2 \sum_i \pi_i$$

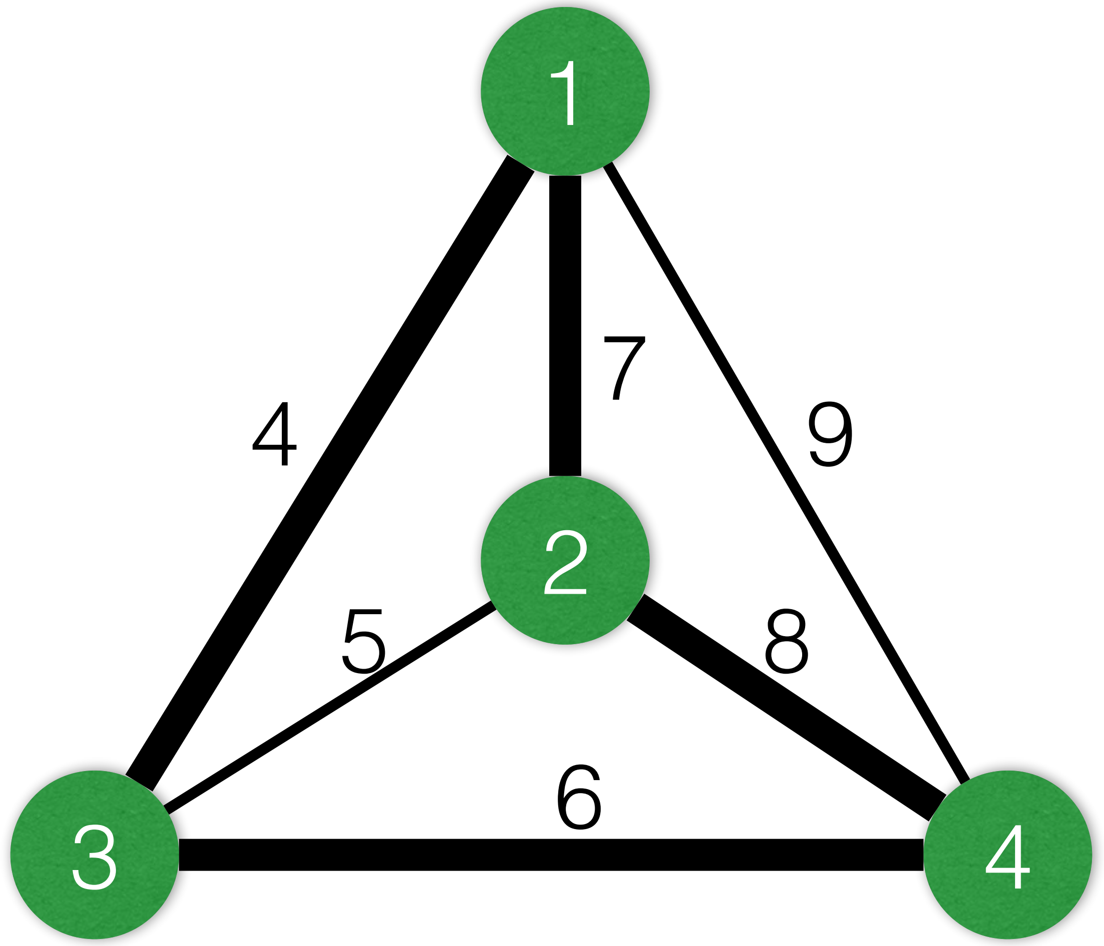
selected edges $\{e \mid x_e = 1\}$ form a 1-tree

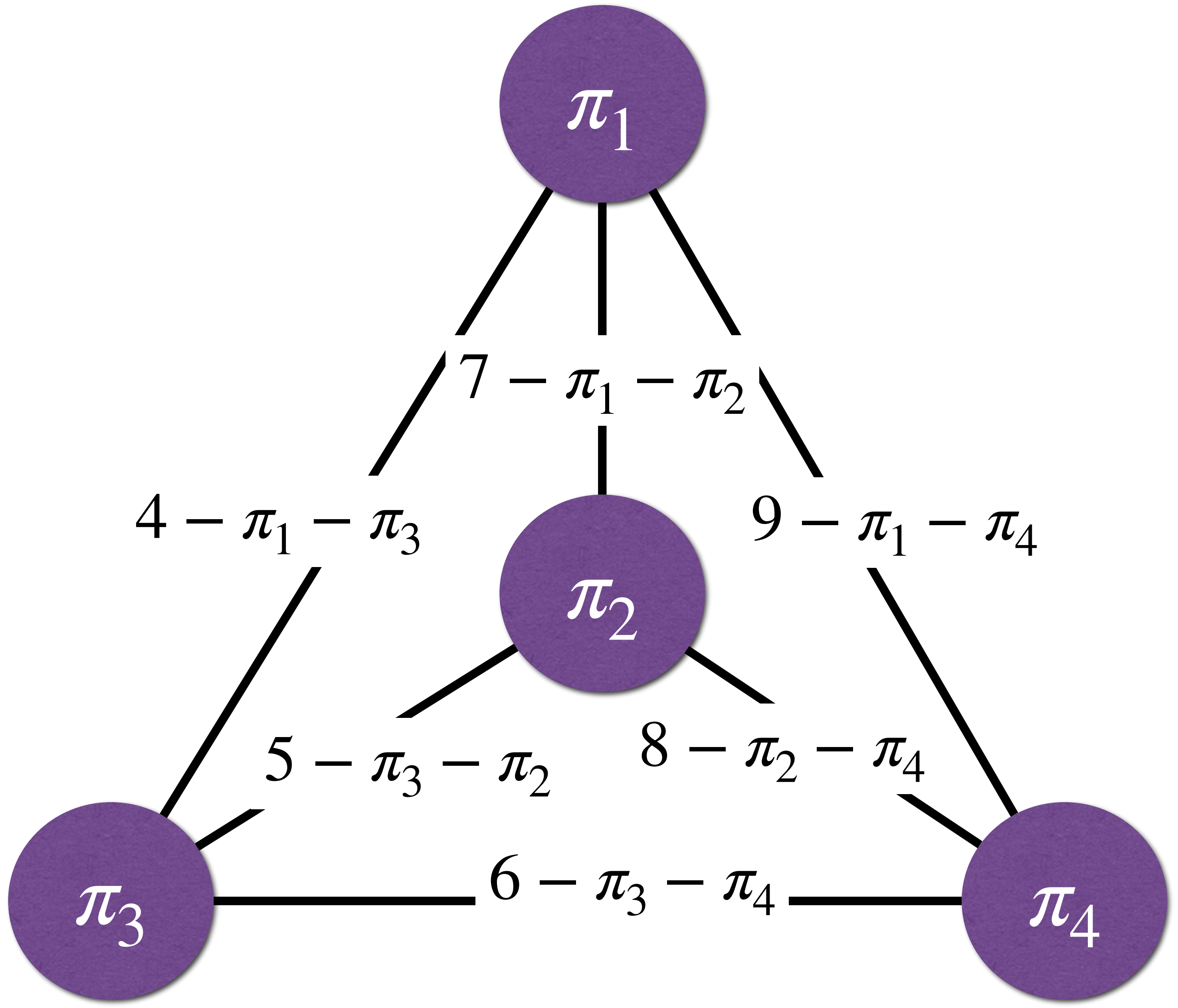
$x_e \in \{0,1\}, \forall e$



Lower-Bound = 6+7+6+6=25

- Notice that 4+7+8+6 = 25 (obtained with the same set of edges of our one-tree but with original weights) is gives the same value, is it pure chance ?



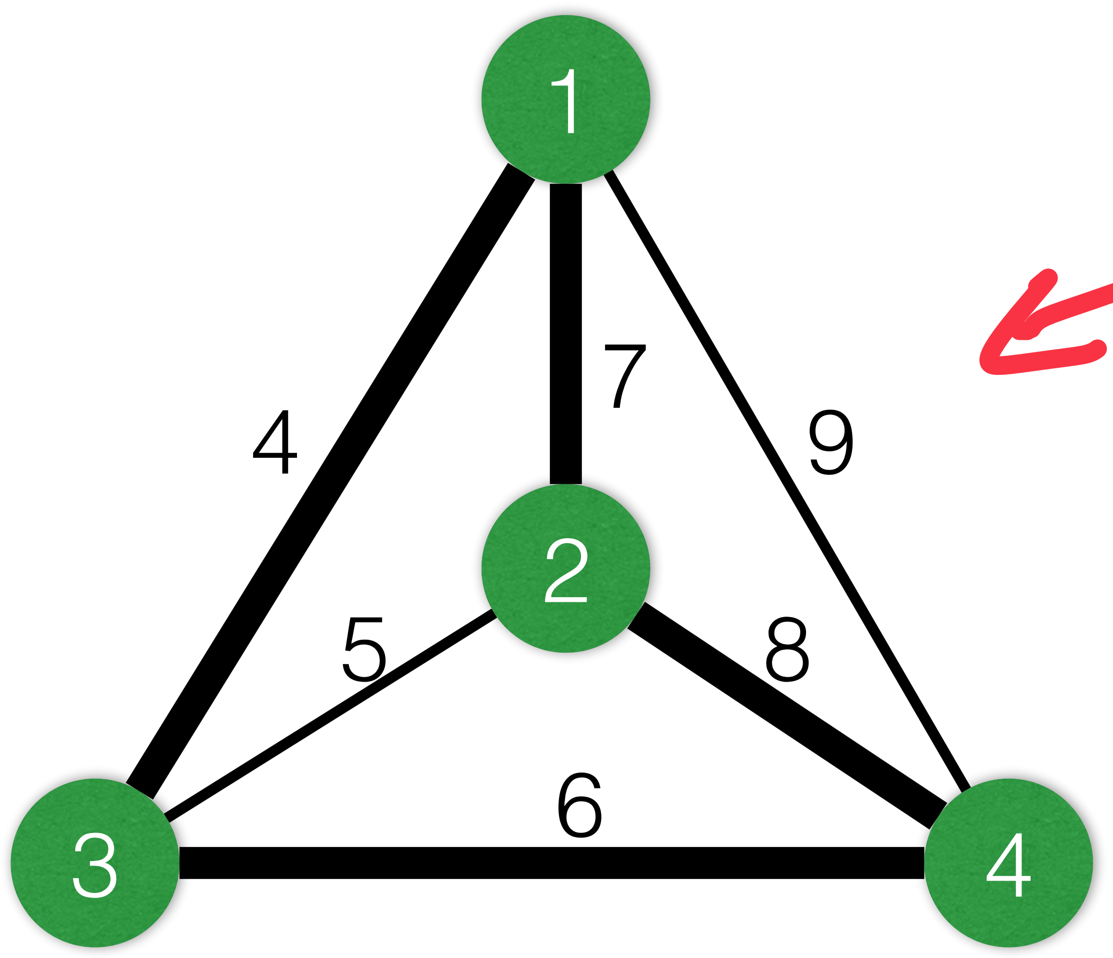


- No! It is a consequence of the fact that we are working with multipliers that sum to 0

$$\sum_i \pi_i = 0$$

Proof of optimality

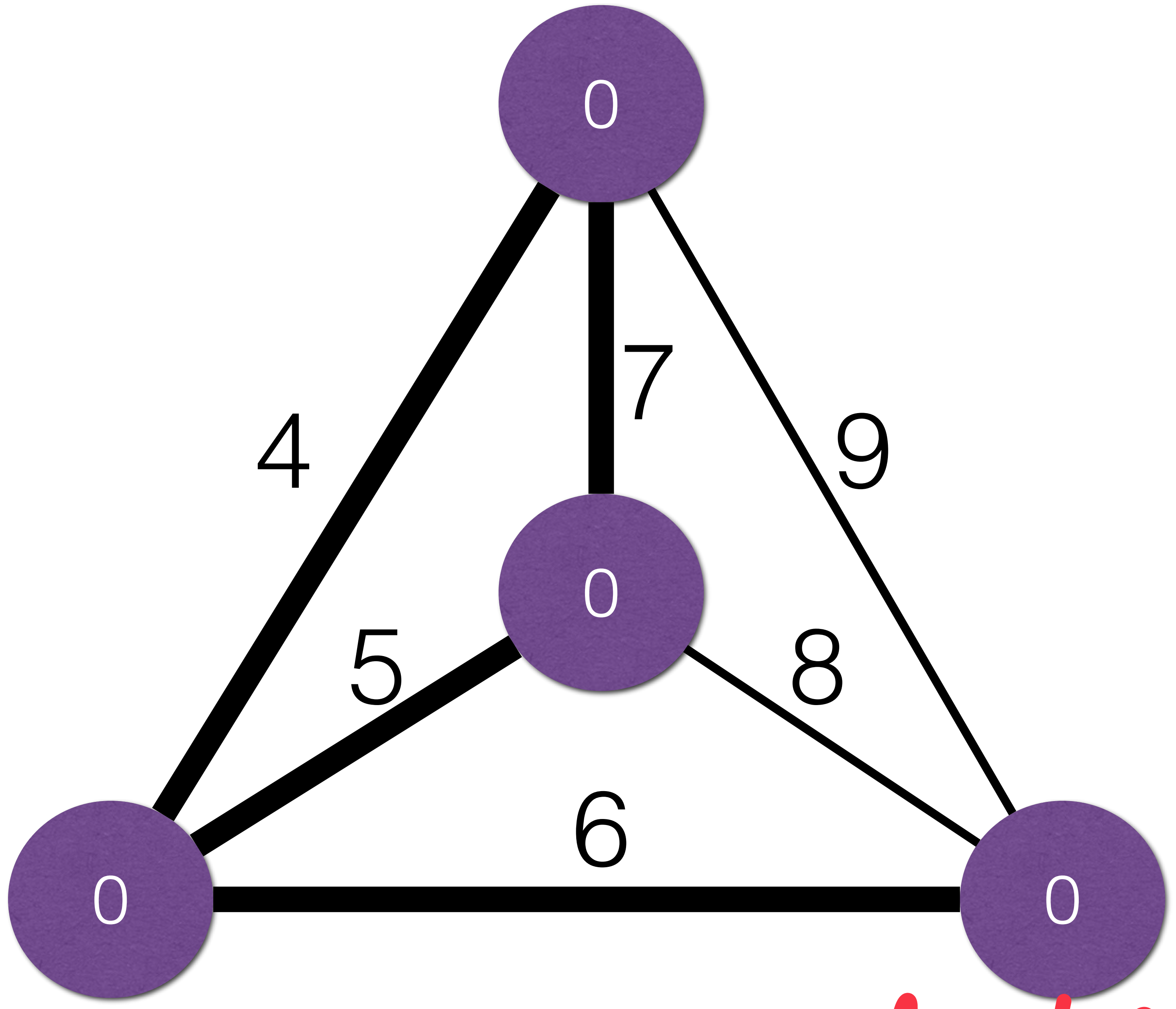
- It is thus interesting to work with multipliers summing to zero since:
  - The tour found in the Lagrangian relaxation has exactly the same weight as in the original graph.
  - Therefore if the tour of the Lagrangian relaxation is a Hamiltonian circuit, it is optimal since we have found an upper-bound (feasible solution) equal to the value of our lower-bound.



This is thus a proven optimal solution!

Update of the multipliers (sub-gradient)

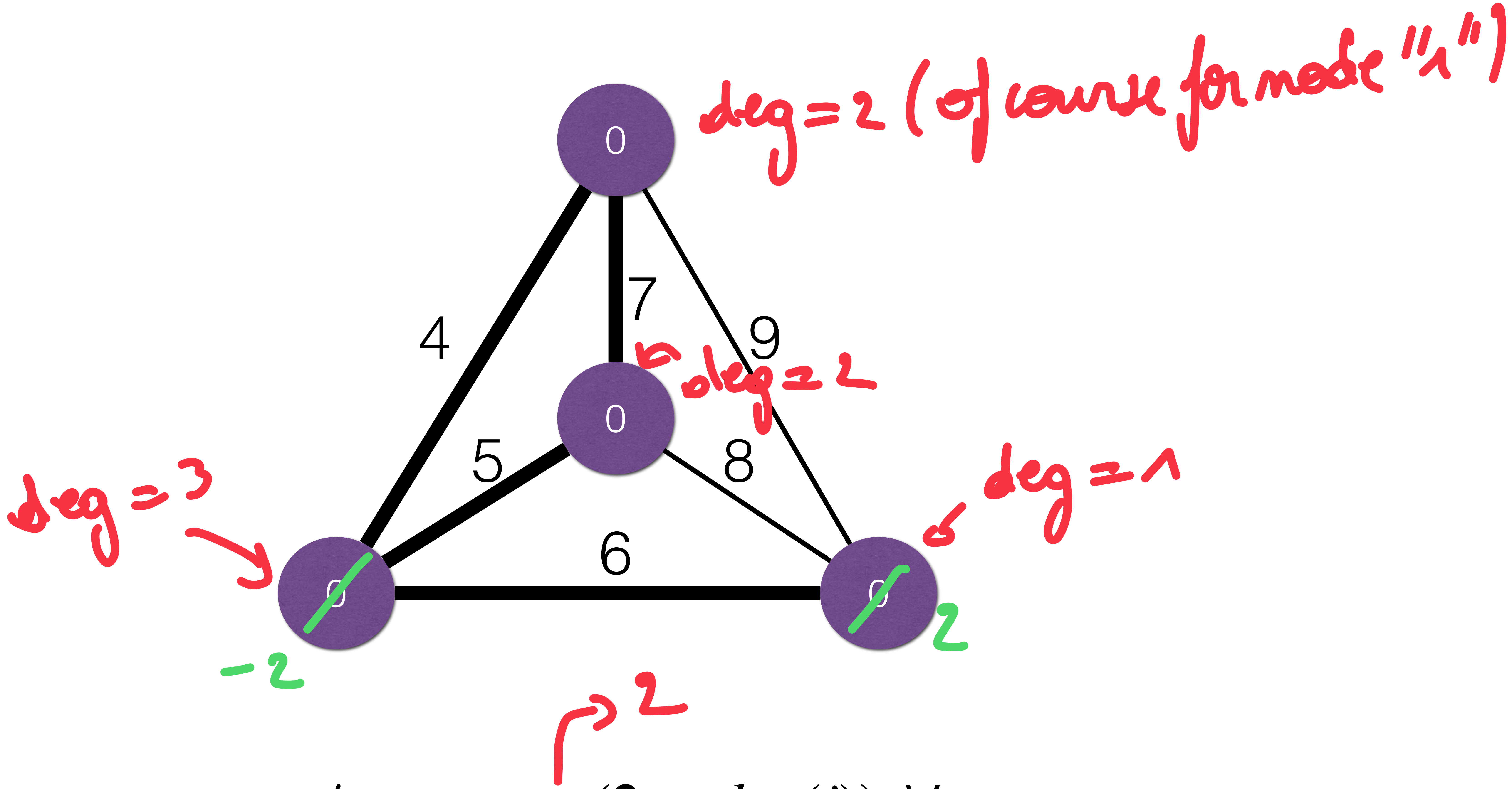
- Intuition: nodes having a too high degree (>2) should become less attractive and nodes with a too low degree (=1) should become more attractive.



step factor at iteration k

$$\pi_i' \leftarrow \pi_i + \mu_k(2 - deg(i)), \forall_i$$

Update of the multipliers (sub-gradient)



$$\pi_i' \leftarrow \pi_i + \mu_k(2 - deg(i)), \forall_i$$

Does the update rule guarantee that ?

- $\sum_i \pi_i = 0$ should remain true after the update

$$\pi_i' \leftarrow \pi_i + \mu_k(2 - deg(i)), \forall_i$$

- Let's verify this

- $$\sum_i \pi_i' = \sum_i (\pi_i + 2\mu_k - \mu_k \cdot deg(i)) = (\underbrace{\sum_i \pi_i}_{=0\ (\text{hypothesis})}) + \overbrace{2 \cdot V \cdot \mu_k - \mu_k \sum_i deg(i)}^{=0 \text{ since } |V| \text{ edges}}$$

Lagrangian Relaxation

- $\mu_k = \dfrac{\lambda_k \cdot \mathscr{L}^k}{\sum_i (deg(i) - 2)^2}$

- $\lambda_{k+1} \leftarrow \lambda_k$ if improvement , $0.9 \cdot \lambda_k$ otherwise

Valenzuela, C. L., & Jones, A. J. (1997). Estimating the Held-Karp lower bound for the geometric TSP. *European journal of operational research*, *102*(1), 157-175.

Final Algo

Result: A lower bound for the TSP

$\pi_i \leftarrow 0, \ \forall i$

$\lambda \leftarrow 0.1$

$lb \leftarrow \infty$

$best \leftarrow \infty$

while $\lambda \geq \epsilon$ **do**

  $(lb', 1 - tree) \leftarrow \mathcal{L}(\pi)$

  if $isHamiltonian(1 - tree)$ **then**

    optimal TSP found

    break

  end

  if $lb' > lb$ **then**

    $\lambda \leftarrow \lambda \cdot 0.9$

  end

  $\mu \leftarrow \frac{\lambda \cdot lb}{\sum_i (deg(i) - 2)^2}$

  $\pi_i \leftarrow \pi_i + \mu(2 - deg(i)), \forall_i$

  $lb \leftarrow lb'$

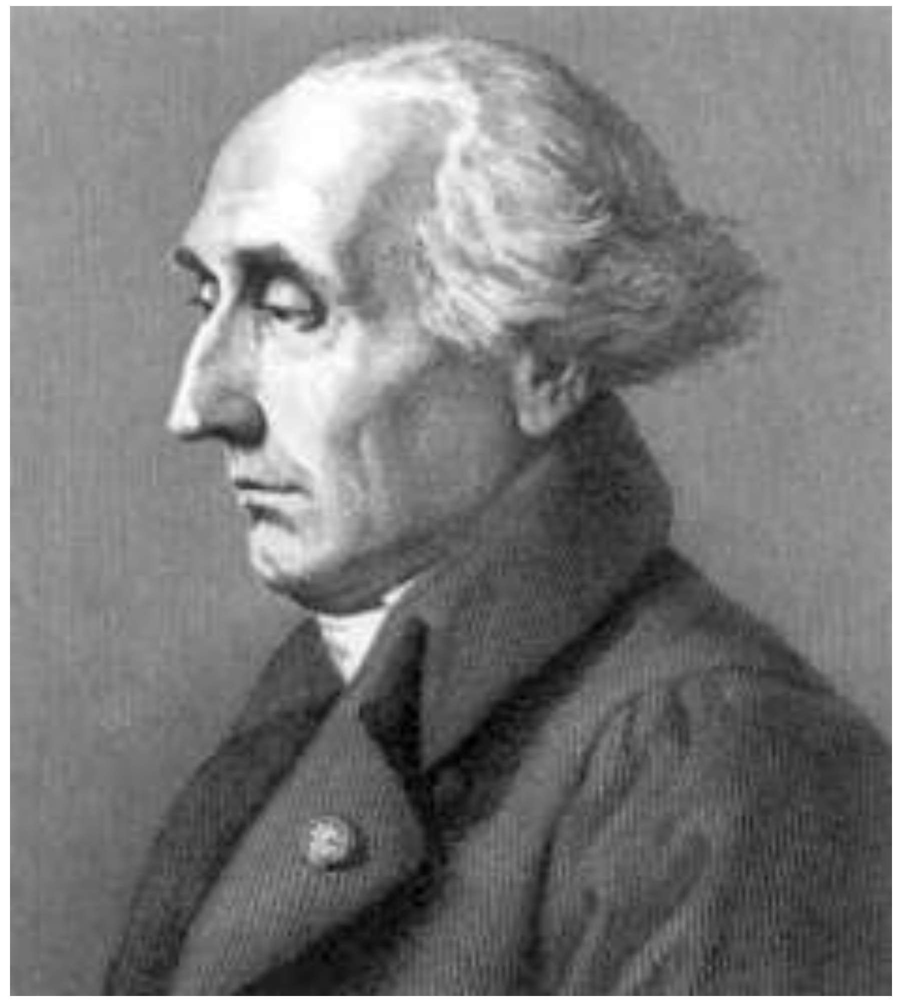
  $best \leftarrow \max(lb, best)$

end

return $best$

History

Joseph-Louis Lagrange



1736-1813

method of Lagrange multipliers (named after Joseph Louis Lagrange[1]) is a strategy for finding the local maxima and minima of a function subject to equality constraints.

Hugh Everett III

1930-1982

he developed the use of generalized Lagrange multipliers for operations research

Naum Zuselevich Shor

1937-2006

subgradient methods

Michael Held & Richard M. Karp (IBM)

Held

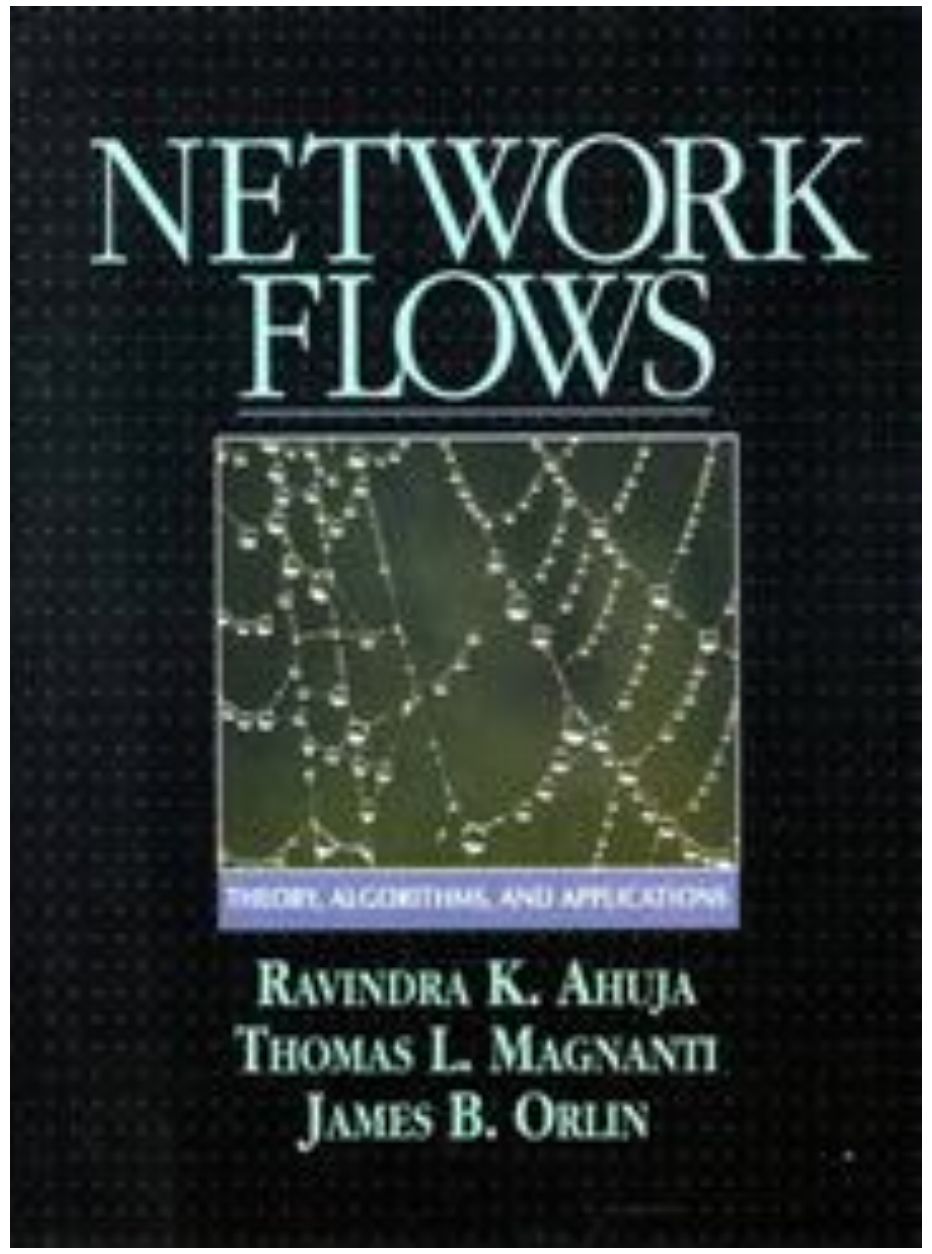
Karp

Turing Award 1985

January 3, 1935 (age 87)

Bibliograpy

- R. K. Ahuja, T. L. Magnanti and J. B. Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.



THE TRAVELING-SALESMAN PROBLEM AND MINIMUM SPANNING TREES

Michael Held

IBM Systems Research Institute, New York, New York

and

Richard M. Karp

University of California, Berkeley, California

(Received September 2, 1969)