

# The Complexity of Stackelberg Network Pricing

Sébastien Roch

École Polytechnique, Montréal

(joint work with Gilles Savard and Patrice Marcotte)

# Network Pricing

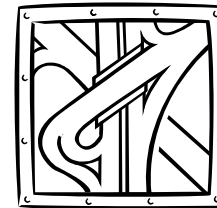
- Goal: 

- Minimizing total latency

- Selfish users: « Price of Anarchy »
- Solution: mechanism design, e.g. pricing
- Homogeneous case: marginal cost pricing
- General case [Cole, Dodis, Roughgarden '03]

- Maximizing profit

- Firm collects tolls on subset of arcs
- Users travel on shortest paths
- Applications: highway tolls, airline yield management, ...



*We focus on the second goal...*



# Outline

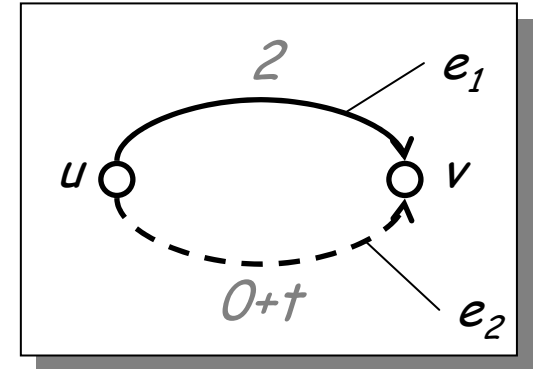
I. Problem Formulation  
and Complexity

II. Algorithms

(Most results are taken from [R, Marcotte, Savard '03]. E-print available at <http://www.gerad.ca/~sroch>.)

# Stackelberg Network Pricing

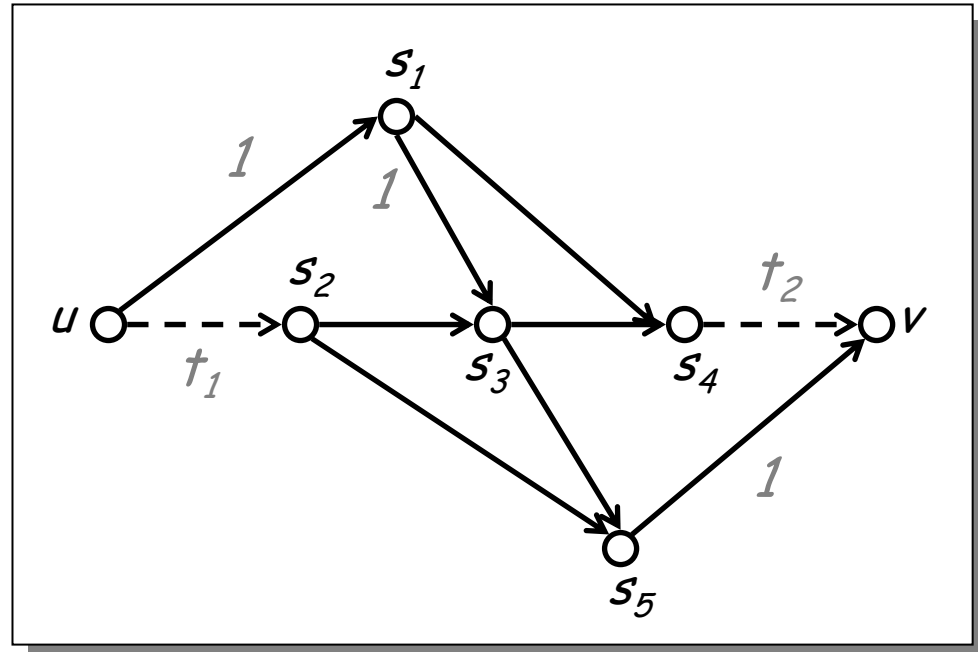
- Directed graph  $G=(V, A_t \cup A_f)$ 
  - $A_t$ : toll arcs  $\rightarrow$  cost  $c_e + t_e$   $\dashrightarrow$
  - $A_f$ : toll-free arcs  $\rightarrow$  cost  $d_e$   $\longrightarrow$
  - source  $u$  - destination  $v$
- Manager chooses tolls  $t_e$
- User travels on a shortest path  $p(t)$
- GOAL: maximize sum of tolls on  $p(t)$
- Example:
  - $t = 3 \rightarrow p(t) = e_1$ , profit = 0
  - $t = 2 \rightarrow p(t) = e_2$ , profit = 2



*How hard is SNP?*

# Upper bound on profit

- Shortest path when
  - toll arcs are removed:
    - $(u, s_1, s_3, s_5, v)$
    - cost:  $L_{inf} = 3$
  - tolls are 0:
    - $(u, s_2, s_3, s_4, v)$
    - cost:  $L_0 = 0$
- Upper bound:
  - $UB = L_{inf} - L_0 = 3$



*Is UB reachable?*

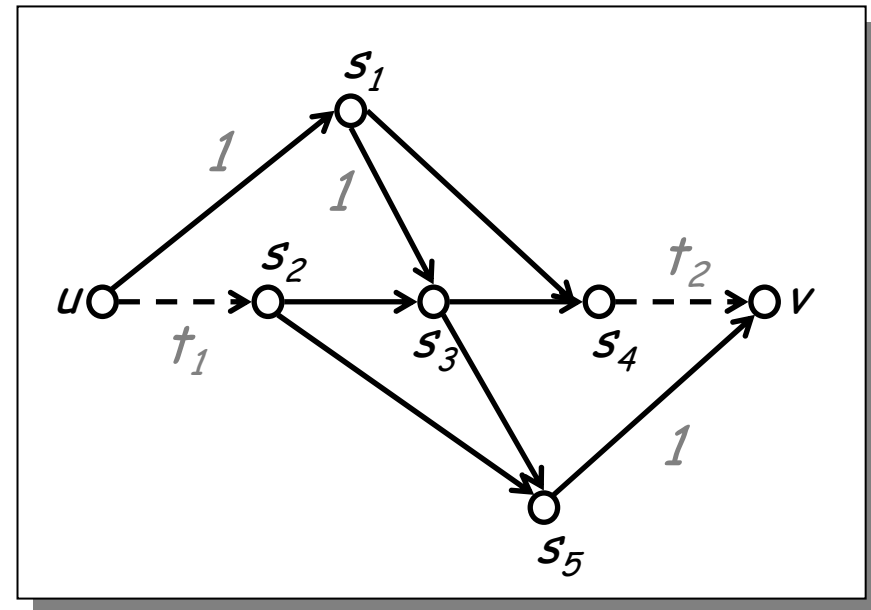
# When the path is known

- Manager wants to force user to travel on

- $(u, s_2, s_3, s_4, v)$ :
  - $t_1 = 1, t_2 = 1, \text{profit} = 2$
- $(u, s_1, s_4, v)$ :
  - $t_1 = \text{inf}, t_2 = 2, \text{profit} = 2$
- $(u, s_2, s_5, v)$ :
  - $t_1 = 2, t_2 = \text{inf}, \text{profit} = 2$

- *UB* reachable? NO

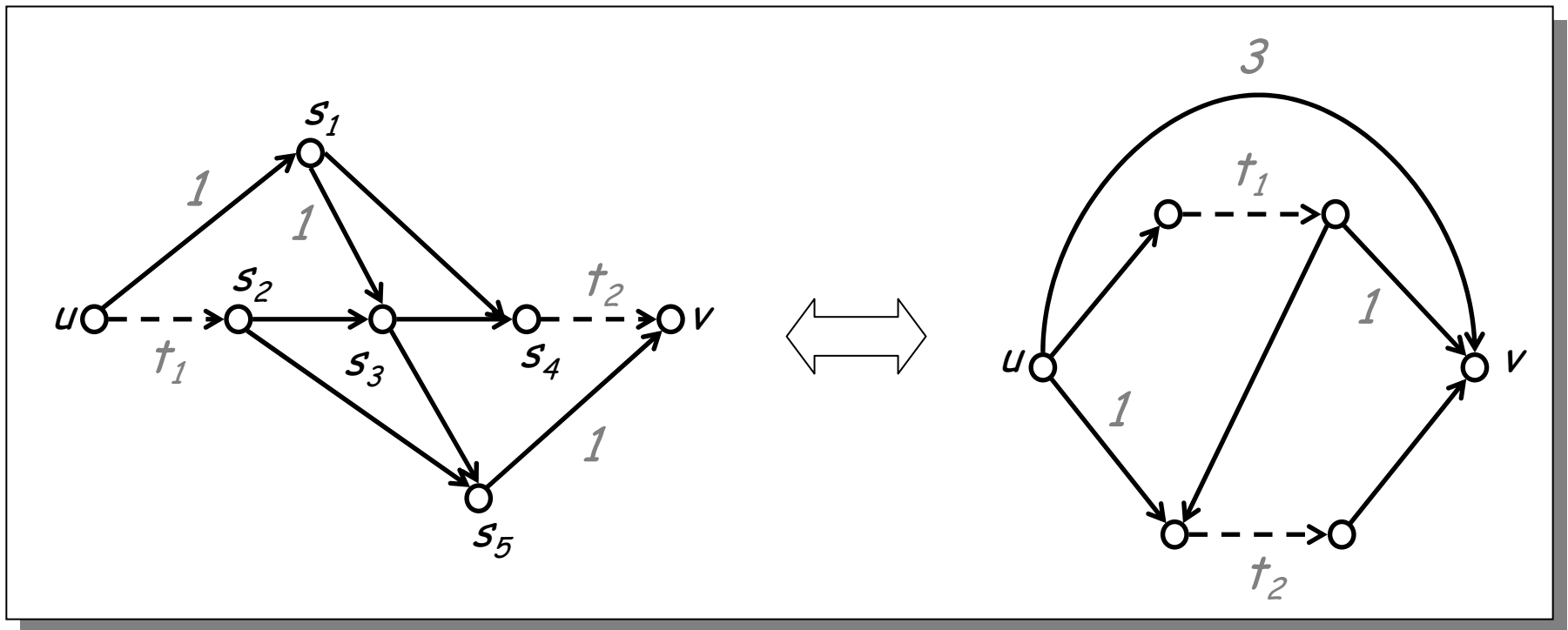
- $\text{OPT} = 2 < 3 = \text{UB}$



*How tight is UB?*

# Equivalent representation

- Need only shortest toll-free paths between toll arcs (and source-sink)



# The path is known (cont'd)

## Algorithm **MaxProfit**

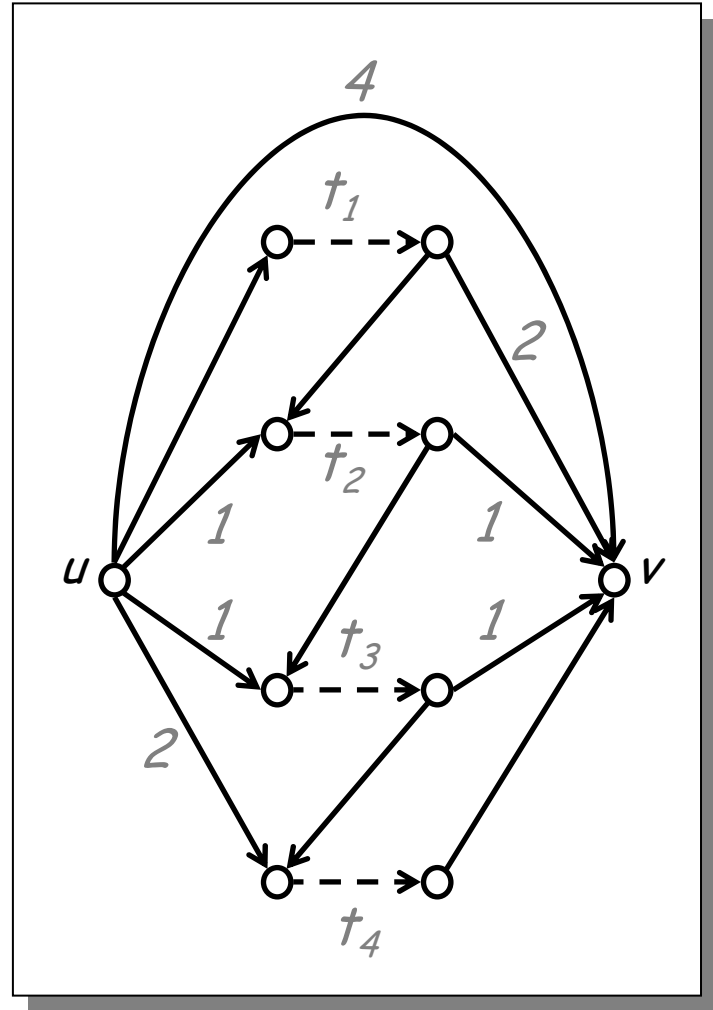
Input: path  $p$

Output: largest profit on  $p$

- For each toll arc (in traversal order)
  - Compute largest toll (taking into account tolls set previously)

## Example:

- $u \rightarrow t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4 \rightarrow v$
- $t_1 = 1, t_2 = 0, t_3 = 1, t_4 = 0$
- profit = 2



*Can this be used to solve SNP in poly-time?*

# SNP is NP-hard

## 3-SAT $\rightarrow$ SNP

- $n$  Boolean variables

$$x_1, x_2, \dots, x_n$$

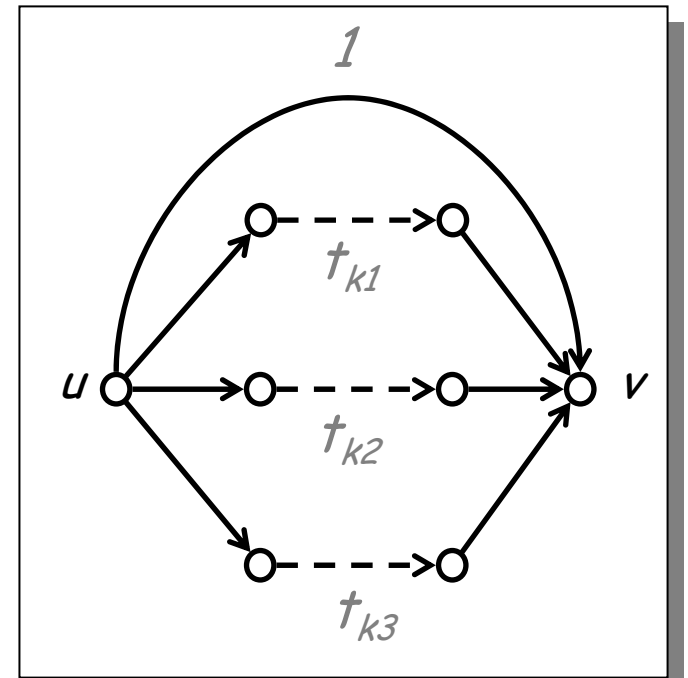
- 3-CNF formula

$$F = (l_{11} \vee l_{12} \vee l_{13}) \wedge \dots \wedge (l_{m1} \vee l_{m2} \vee l_{m3})$$

- For each clause

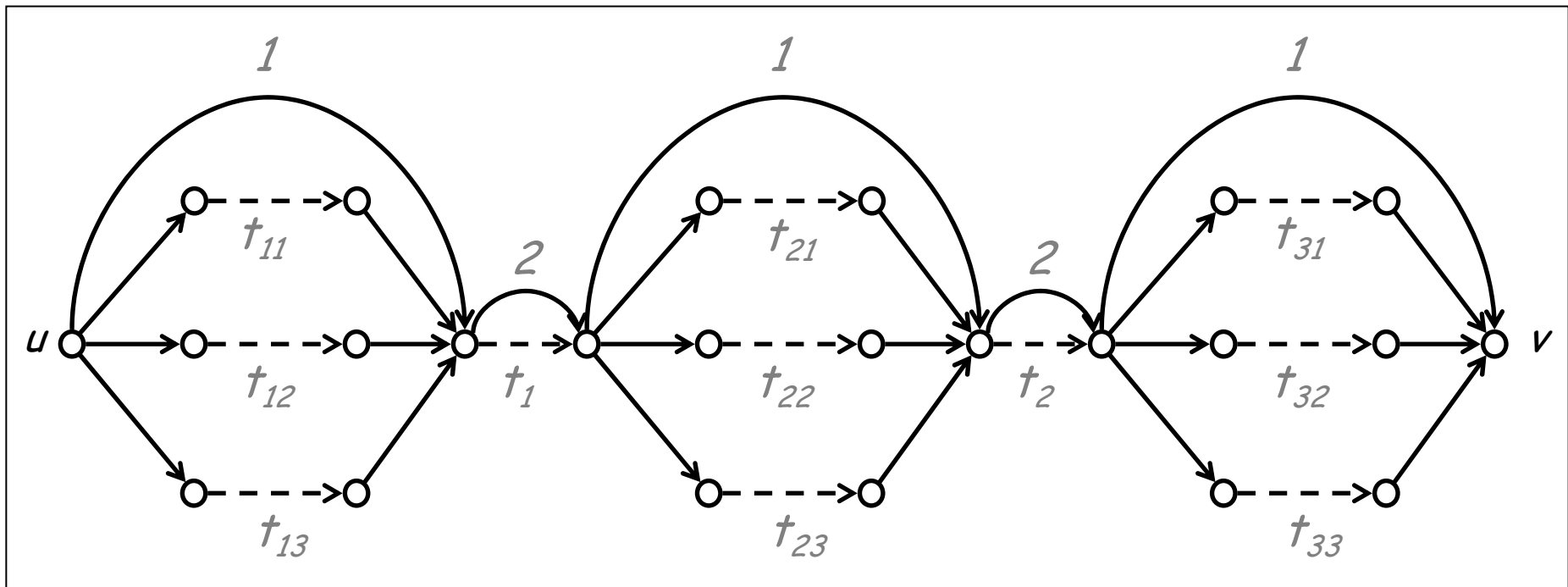
$$l_{k1} \vee l_{k2} \vee l_{k3}$$

construct subnetwork shown



# SNP is NP-hard (cont'd)

- Join clauses as shown (for  $m = 3$ )
- $UB = 3m - 2 - 0 = 7$
- Each path of length  $0$  includes  $t_1$ ,  $t_2$ , and 3 chosen literals
- These 3 chosen literals are considered *TRUE*
- For *F* unsatisfiable, we want:  $PROFIT < UB$

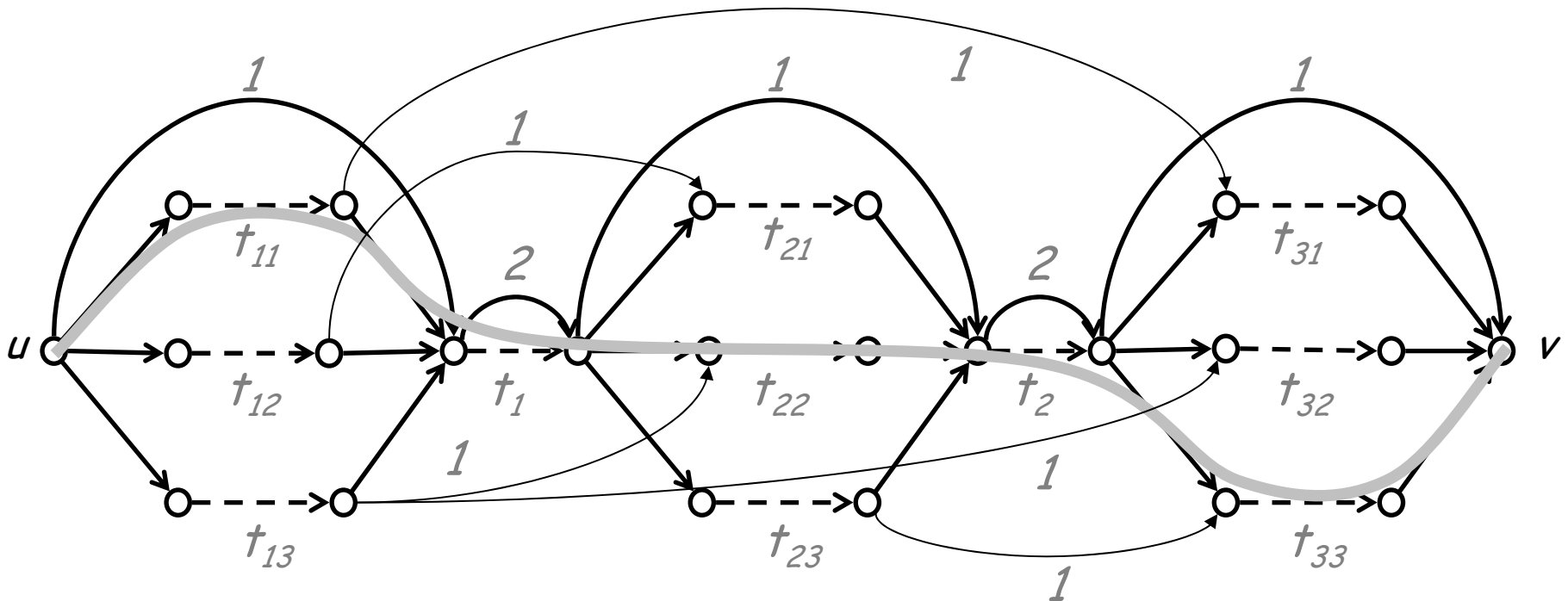


# SNP is NP-hard (cont'd)

- Take formula

$$F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee x_3 \vee x_4)$$

- Satisfying assignment:  $x_1 = x_3 = x_4 = \text{TRUE}$
- Thick path  $\rightarrow$   $PROFIT = 7 = UB$



# Outline

I. Problem Formulation  
and Complexity

II. Approximation Algorithms

*What can be done in poly-time?*

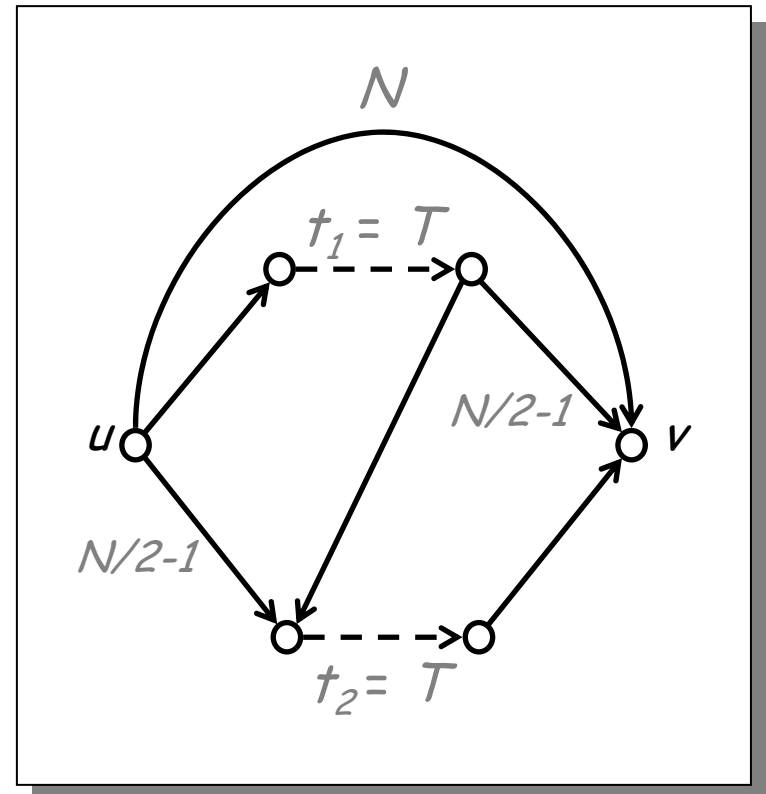
# A $2\log(m)$ -approximation

## Algorithm **UniformTolls** [Anonymous '03]

- All tolls are set to  $T$
  - Increase  $T$  from  $0$
  - Compute profit when length of shortest path is
    - $L_0 + UB/\log(m_T), L_0 + 2UB/\log(m_T), L_0 + 3UB/\log(m_T), \dots$
- 
- $p(T)$ : shortest path when tolls are  $T$
  - $T = 0$ :  $\text{length}[p(0)] = L_0$
  - $T = T^{(1)}$ :  $\text{length}[p(T^{(1)})] = L_0 + UB/\log(m_T)$ 
    - $T^{(1)} \geq UB/(m_T \log(m_T))$
    - If number of toll arcs on  $p(T^{(1)})$  is  $\geq m_T/2$  : DONE
  - $T = T^{(2)}$ :  $\text{length}[p(T^{(2)})] = L_0 + 2UB/\log(m_T)$ 
    - $T^{(2)} \geq T^{(2)} - T^{(1)} \geq 2UB/(m_T \log(m_T))$
    - If number of toll arcs on  $p(T^{(2)})$  is  $\geq m_T/4$  : DONE
  - etc.

# Example

- $m_T = 2 \rightarrow 2\log(m_T) = 2$
- $UB = N - 0 = N$
- $T = 0$ :
  - $\text{length}[p(0)] = L_0 = 0$
  - profit = 0
- $T = T^{(1)} = N/2 + 1$ :
  - $\text{length}[p(T^{(1)})] = L_0 + UB$
  - profit =  $N/2 + 1 \sim UB/2$



*Can we do better?*

# Another idea

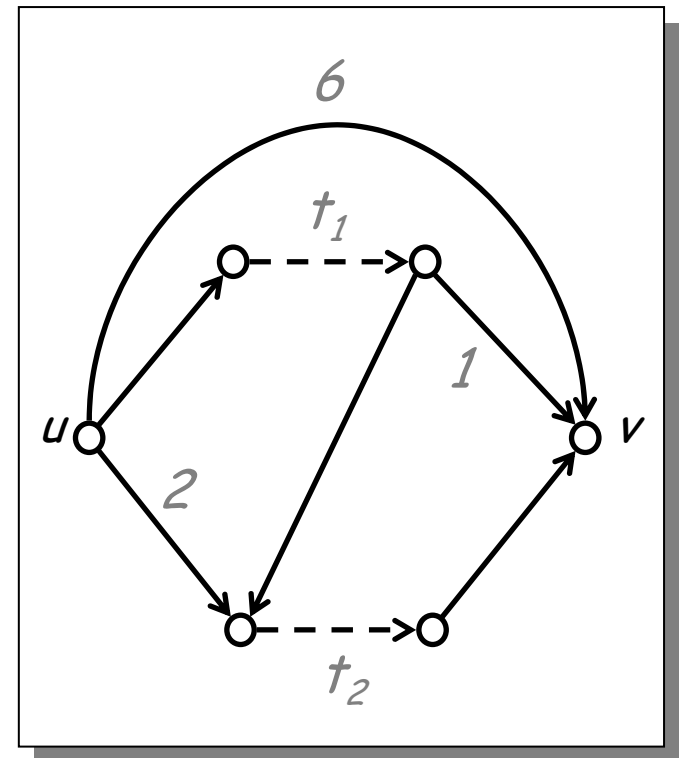
## Algorithm ExploreDescendants

Start with shortest path when tolls are  $0 \rightarrow p$

- Compute largest profit on  $p$
- If profit on  $p$  is "too small"
  - Remove some toll arcs from  $p$  by travelling on "active toll-free subpaths"
  - Return to first step

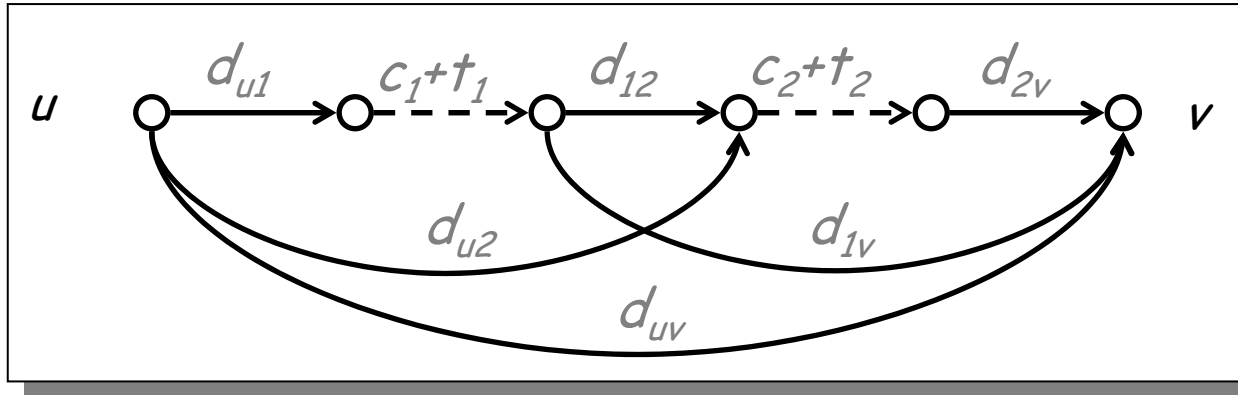
Example:

- $u \rightarrow t_1 \rightarrow t_2 \rightarrow v$ , profit = 3
- Remove  $t_2$ :  $u \rightarrow t_1 \rightarrow v$ , profit = 5



*What is the approximation factor?*

# Idea of proof



$$(1): \begin{cases} t_1^{(1)} = d_{u2} - d_{u1} - c_1 - d_{12}; t_2^{(1)} = d_{1v} - d_{12} - c_2 - d_{2v} \\ t_1^{(1)} + t_2^{(1)} < 1/\alpha UB \end{cases}$$

$$(2): \begin{cases} t_1^{(2)} = d_{uv} - d_{u1} - c_1 - d_{1v}; t_2^{(2)} = +\infty \end{cases}$$

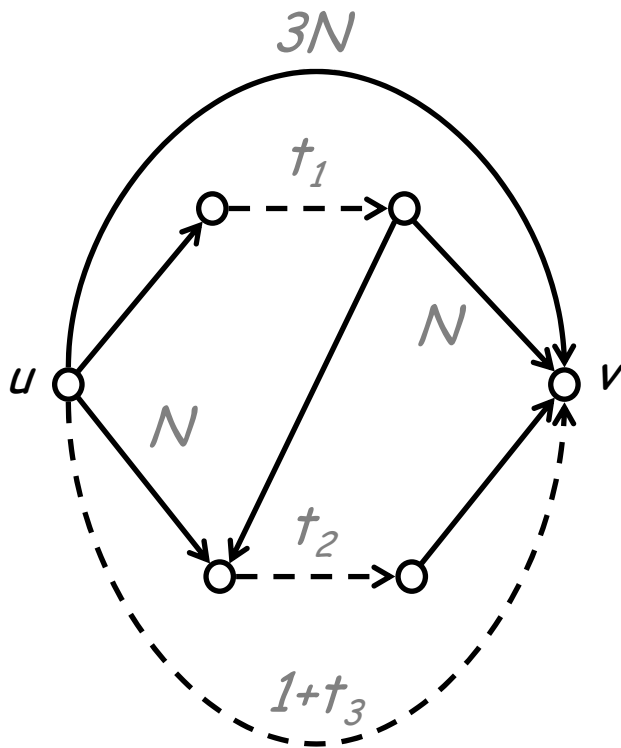
$$(3): \begin{cases} t_1^{(3)} = +\infty; t_2^{(3)} = d_{uv} - d_{u2} - c_2 - d_{2v} \end{cases}$$

$$t_1^{(2)} + t_2^{(3)} = 2UB - t_1^{(1)} - t_2^{(1)} > (2 - 1/\alpha)UB > 2/\alpha UB$$

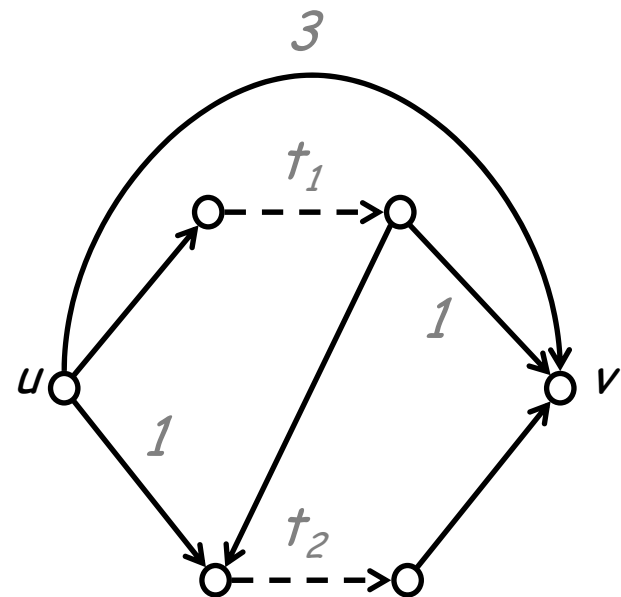
$$\therefore \alpha = \frac{3}{2} \Rightarrow t_1^{(2)} \vee t_2^{(3)} > 1/\alpha UB$$

# Tight examples

- Tightness of analysis
  - $OPT = 3N - 1$
  - Algo. returns  $2N$



- Tightness of approximation (w.r.t. UB)
  - $OPT = 3, UB = 2$



# General results

Theorem 1 (Approximation factor) Define

- $APP$ : value returned by the algorithm
- $I(m_T)$ : set of all instances with  $m_T$  toll arcs

Then

$$\alpha(m_T) \equiv \max_{\gamma \in I(m_T)} \left\{ \frac{OPT[\gamma]}{APP[\gamma]} \right\} \leq \frac{1}{2} \log(m_T) + 1$$

Theorem 2 (Worst-case relaxation gap)

$$\max_{\gamma \in I(m_T)} \left\{ \frac{UB[\gamma]}{OPT[\gamma]} \right\} = \alpha(m_T)$$

# Open problems

- Can we improve on the approximation factor?
- Other models
  - several sources and sinks
  - non-constant latencies

*Questions?*