One-parameter mechanisms, with an application to the SPT problem

The private-edge Shortest-Paths Tree (SPT) problem

- **Given**: an undirected graph $G=(V,E)$ such that each edge is owned by a distinct player, and a source node s; we assume that player's private type $t(e)$ is the positive cost (length) of the edge, and her valuation function is equal to her negated type if edge is selected in the solution, and 0 otherwise.
- **Question**: design an efficient (in terms of time complexity) truthful mechanism in order to find a shortest-path tree rooted at s in $G_t=(V,E,t)$.

More formally…

- F: set of all spanning trees of G rooted at s
- For any T∈F, we want to **maximize**

$$f(t,T) = -\sum_{v\in V} d_T(s,v) = -\sum_{e\in E(T)} t_e \, \|e\|$$

where $\|e\|$ is the set of source-node paths in T containing edge e

On the other hand, $v_e(t_e,T) = -t_e$ if $e \in E(T)$, 0 otherwise (this models the multicast protocol)

$$\Rightarrow f(t,T) \neq \sum_{e\in E(T)} v_e(t_e,T) = -\sum_{e\in E(T)} t_e$$

⇒ non-utilitarian problem!

One-parameter MD problems

This is a mechanism design problem in which:

1. The type owned by each player i is a single parameter $t_i \in \Re$
2. The valuation function of player i w.r.t. to an output $o \in X$ is

$$v_i(t_i, o) = t_i\, w_i(o)$$

where $w_i(o) \in \Re_{\geq 0}$ is the workload function for i. Notice that for the sake of simplifying the notation, we are now assuming that the valuation function is positive, and so in the following we will invert – and +, and max with min, and so the utility will be now equal to the payment minus the valuation

The SPT problem is one-parameter

- First of all, the type owned by each player is a single real-value number
- Second, the valuation function of a player w.r.t. to a tree T is:
  - $v_e(t_e,T)= \begin{cases} t_e & \text{if } e \in E(T) \\ 0 & \text{otherwise} \end{cases}$

Multicast protocol

i.e., $v_e(t_e,T)= t_e\, w_e(T)$, where $w_e(T)= \begin{cases} 1 & \text{if } e \in E(T) \\ 0 & \text{otherwise} \end{cases}$

$\Rightarrow$ The SPT problem is a one-parameter (OP) problem!

A necessary condition for designing OP truthful mechanisms

Theorem (R.B. Myerson, 1981)

A mechanism $M=\langle g,p \rangle$ for a minimization one-parameter problem is truthful **only if** g is monotone, i.e., $\forall$ player i, $w_i(g(r_{-i},r_i))$ is non-increasing w.r.t. r_i, for every $r_{-i}=(r_1,\dots,r_{i-1}, r_{i+1},\dots,r_N)$.

Proof

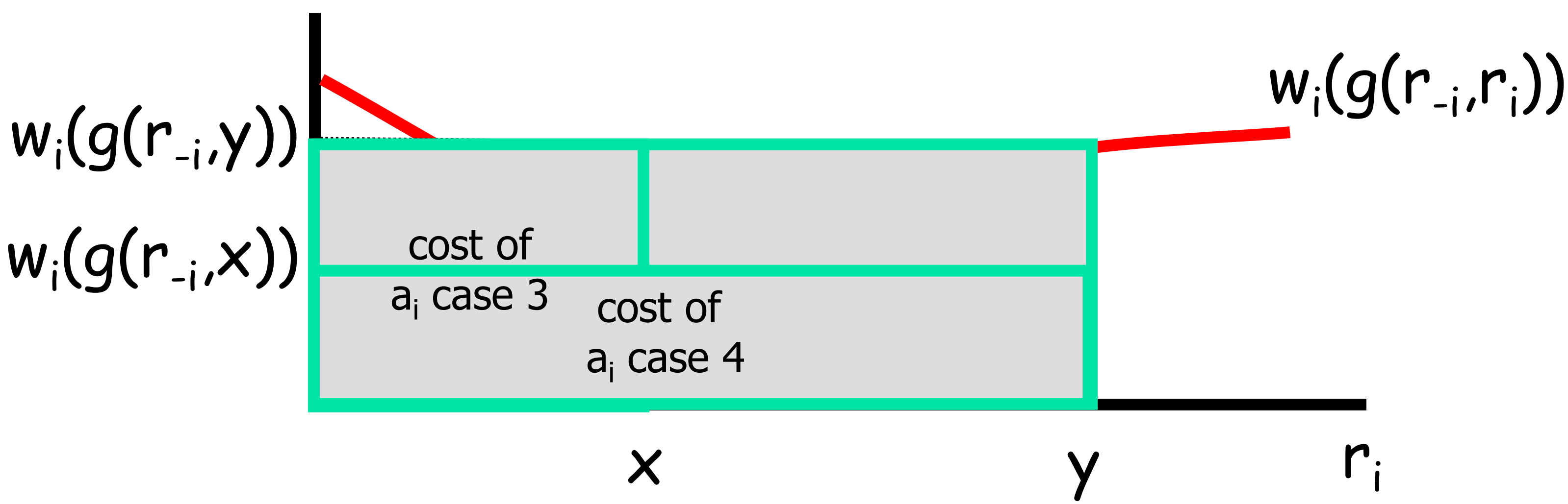
(by contradiction)

Assume that there exists a truthful mechanism $M=\langle g,p\rangle$ such that g is non-monotone.

If g is non-monotone there exists a player i and a vector r_{-i} s.t. $w_i(r_{-i},r_i)$ is not monotonically non-increasing w.r.t. r_i

Proof (cont'd)

1. If t_i=x and r_i=t_i → $v_i(t_i,o) = t_i\ w_i(o) = x\ w_i(g(r_{-i},x))$
2. If t_i=y and r_i=t_i → $v_i(t_i,o) = y\ w_i(g(r_{-i},y))$
3. If t_i=x and r_i=y → $v_i(t_i,o) = x\ w_i(g(r_{-i},y))$, i.e., i increases her valuation (i.e., her cost) by *A*
4. If t_i=y and r_i=x → $v_i(t_i,o) = y\ w_i(g(r_{-i},x))$, i.e., i decreases her cost by *A*+k

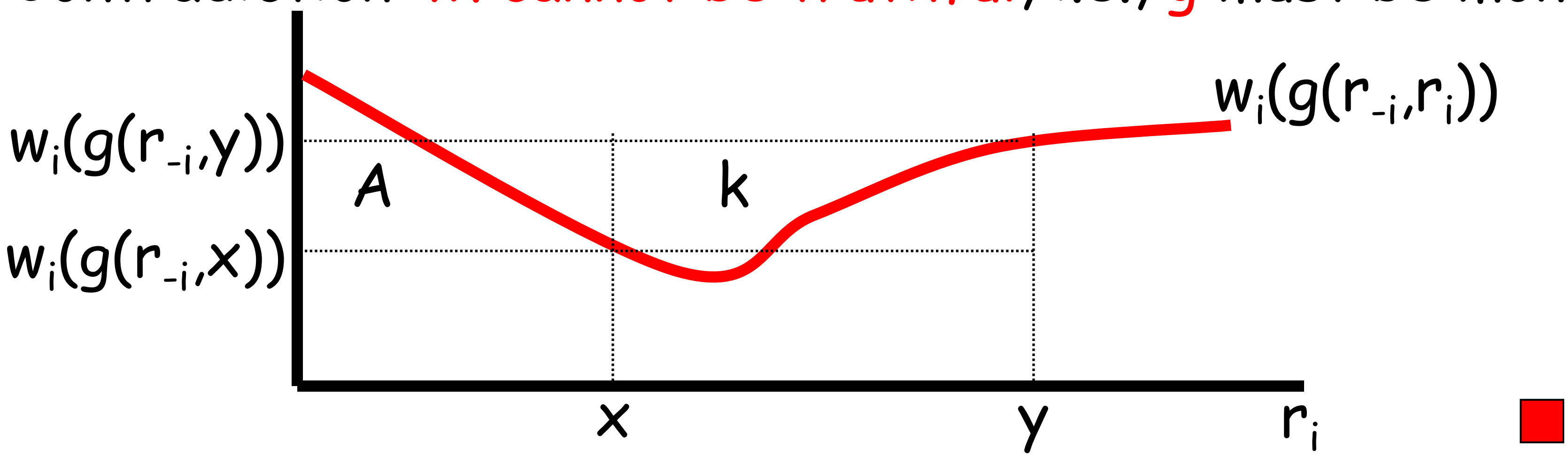


Proof (cont'd)

- Let $\Delta p = p_i(g(r_{-i},y)) - p_i(g(r_{-i},x))$
- If M is truthful it must be:
  - $\Delta p \leq A$ (since otherwise when $t_i=x$, player i will report y, since in this case her valuation will increase by A, and since $\Delta p > A$, her utility (i.e., the payment minus the valuation) will increase (i.e., $\Delta u = \Delta p - \Delta v > 0$), against the assumption that M is truthful!)
  - $\Delta p \geq A+k$ (since otherwise when $t_i=y$, player i will report x, since in this case her cost will decrease by A+k, and since $\Delta p < A+k$, this means that the payment's decrease is less than the cost's decrease, and so her utility will increase , against the assumption that M is truthful!)

… but k is strictly positive!

Contradiction: M cannot be truthful, i.e., g must be monotone!



One-parameter mechanisms

For the sake of simplifying the notation:

We will write $w_i(r)$ in place of $w_i(g(r))$

We will write $p_i(r)$ in place of $p_i(g(r))$

Definition: A one-parameter (OP) mechanism is a pair M=<g,p> such that:

- g: is any monotone algorithm for the underlying one-parameter problem
- $p_i(r) = h_i(r_{-i}) + r_i\, w_i(r) - \int_0^{r_i} w_i(r_{-i},z)\, dz$

$h_i(r_{-i})$: arbitrary function independent of r_i

Truthfulness of OP-mechanisms

Theorem 2: An OP-mechanism for an OP-problem is truthful.

Proof: We show that the utility of a player i can only decrease when i lies

Payment returned to i (when she reports r_i) is:

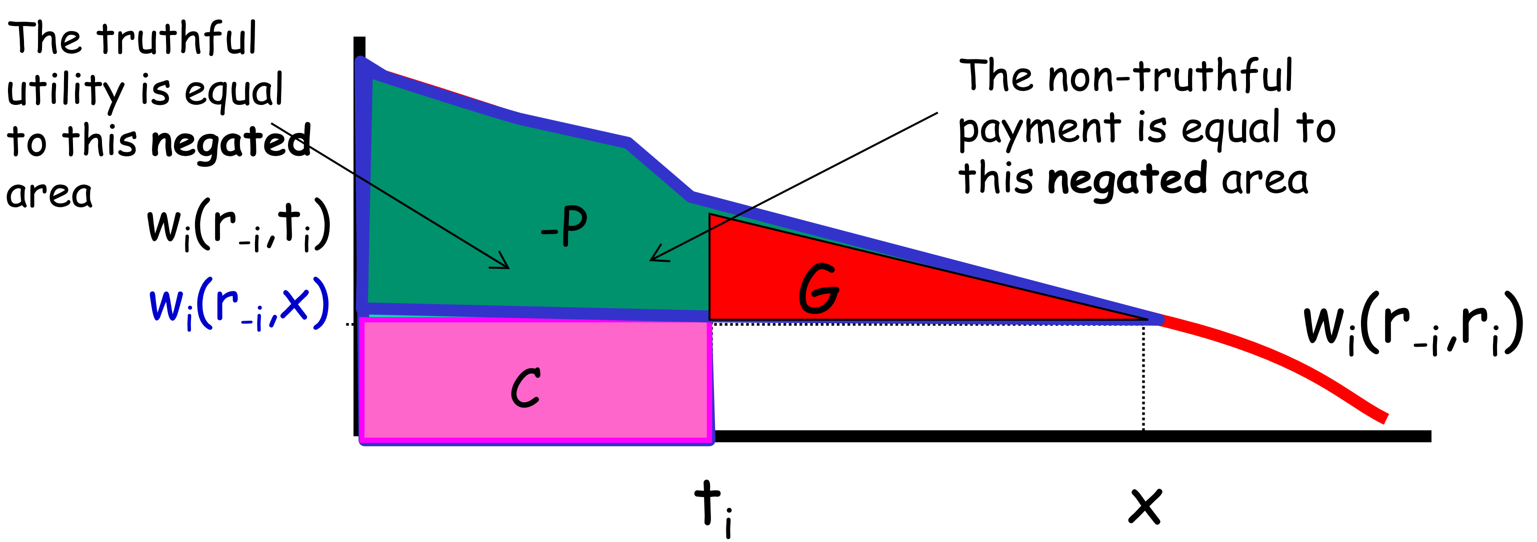
$$p_i(r) = \underbrace{h_i(r_{-i})}_{\text{Indipendent of } r_i} + r_i\, w_i(r) - \int_0^{r_i} w_i(r_{-i},z)\, dz$$

For the purpose of our proof, we set $h_i(r_{-i})=0$
(notice this will produce negative utilities)

Proof (cont'd)

- $u_i(t_i,g(r_{-i},t_i)) = p_i(r_{-i},t_i) - v_i(t_i, g(r_{-i},t_i)) =$
$t_i\, w_i(r_{-i},t_i) - \int_0^{t_i} w_i(r_{-i},z)\, dz - t_i\, w_i(r_{-i},t_i) = -\int_0^{t_i} w_i(r_{-i},z)\, dz$
- If i reports $x>t_i$:
  - Her valuation becomes: $C = t_i\, w_i(r_{-i},x)$
  - Her payment becomes: $P = x\, w_i(r_{-i},x) - \int_0^x w_i(r_{-i},z)\, dz$

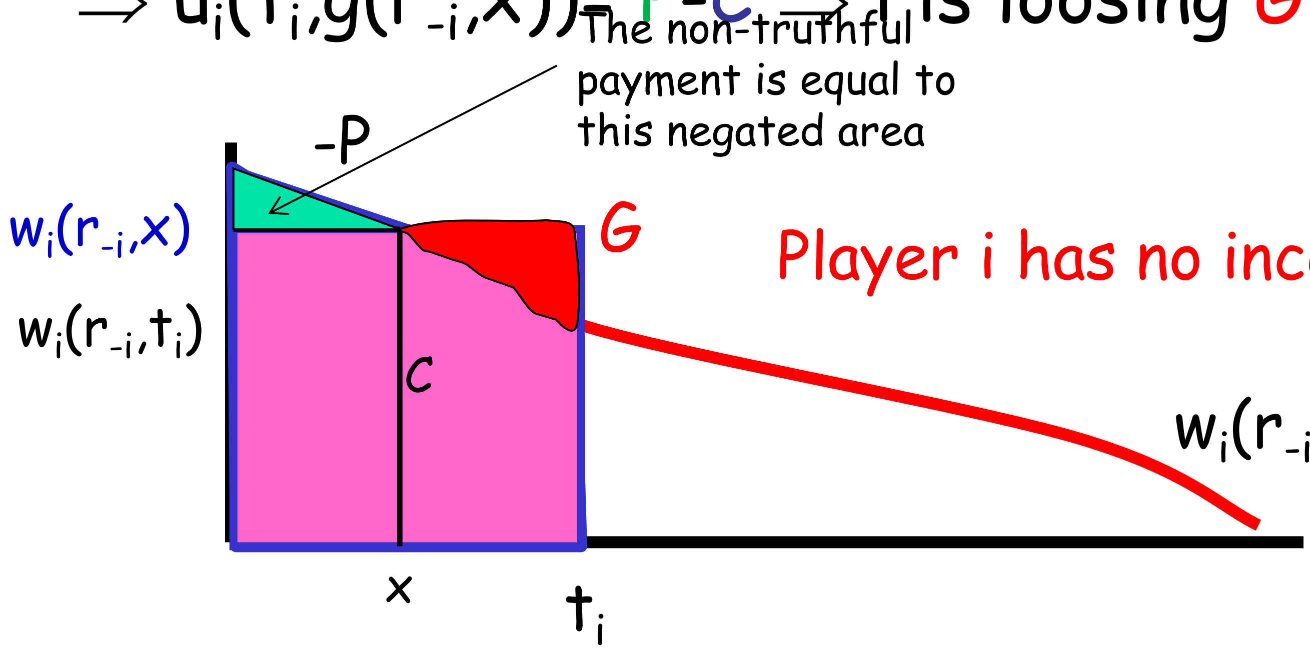
  $\Rightarrow u_i(t_i,g(r_{-i},x)) = P-C \Rightarrow$ the non-truthful utility is given by the negated green-pink-red region $\Rightarrow$ i is loosing G



Proof (cont'd)

- $u_i(t_i, g(r_{-i}, t_i)) = -\int_0^{t_i} w_i(r_{-i}, z)\, dz$
- If i reports $x < t_i$
  - Her valuation becomes $C = t_i\, w_i(r_{-i}, x)$
  - Her payment becomes $P = x\, w_i(r_{-i}, x) - \int_0^{x} w_i(r_{-i}, z)\, dz$

  $\Rightarrow u_i(t_i, g(r_{-i}, x)) = P - C \Rightarrow$ i is loosing G



Player i has no incentive to lie!

On the $h_i(r_{-i})$ function

Once again, we want to guarantee voluntary participation (VP)

But when player i reports r_i, her payment is:

$$p_i(r) = h_i(r_{-i}) + r_i\, w_i(r) - \int_0^{r_i} w_i(r_{-i},z)\, dz$$

If we set $h_i(r_{-i})=\int_0^{\infty} w_i(r_{-i},z)\, dz$, the payment becomes:

$$p_i(r) = r_i\, w_i(r) + \int_{r_i}^{\infty} w_i(r_{-i},z)\, dz$$

$\Rightarrow$ The utility of player i when reporting the true becomes:

$$u_i(t_i,g(r)) = \int_{t_i}^{\infty} w_i(r_{-i},z)\, dz \geq 0.$$

Summary: VCG vs OP

- VCG-mechanisms: arbitrary valuation functions and types, but only utilitarian problems
- OP-mechanisms: arbitrary social-choice function, but only one-parameter types and workloaded valuation functions
- If a problem is both utilitarian and one-parameter → VCG and OP coincide!

A one-parameter mechanism for the private-edge SPT problem

The one-parameter SPT problem

- F: set of spanning tree rooted at s
- For any T ∈ F, we aim to **minimize** (remember indeed that we have changed sign to the valuations)
  - $f(t,T) = \sum_{v \in V} d_T(s,v) = \sum_{e \in E(T)} t_e \, \|e\|$
  - $v_e(t_e,T) = \begin{cases} t_e & \text{if } e \in E(T) \\ 0 & \text{otherwise} \end{cases}$

i.e., $v_e(t_e,T) = t_e \, w_e(T)$, with $w_e(T) = \begin{cases} 1 & \text{if } e \in E(T) \\ 0 & \text{otherwise} \end{cases}$

A corresponding one-parameter mechanism

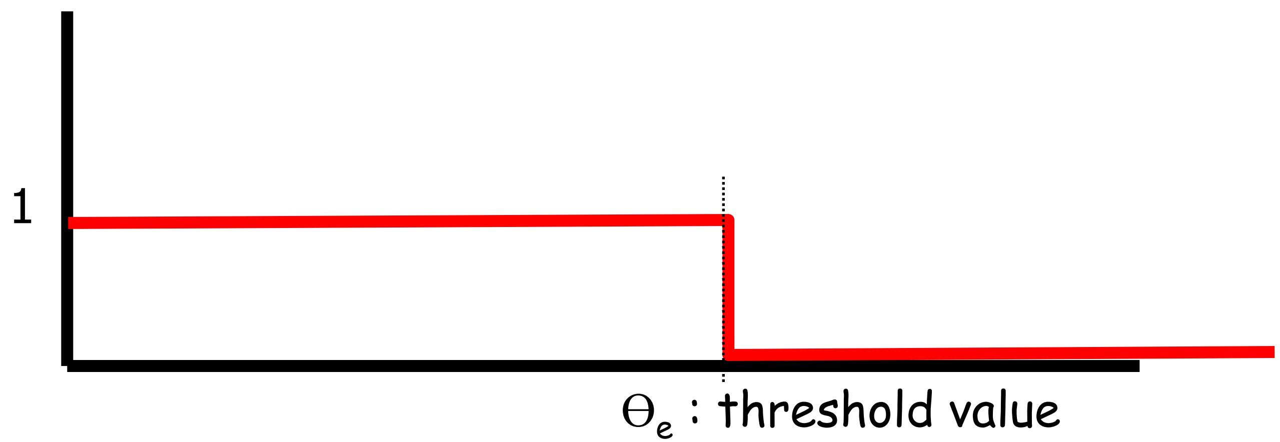
- $M_{SPT} = \langle g,p \rangle$
  - g: given the input graph G, the source node s, and the reported types r, compute an SPT $S_G(s)$ of $G=(V,E,r)$ by using Dijkstra's algorithm;
  - p: for every $e \in E$, let a_e denote the agent owning edge e, and let r_e be her reported type. Then, the payment for a_e is:

$$p_e(r)=r_e w_e(r) + \int_{r_e}^{\infty} w_e(r_{-e},z)\,dz$$

so that VP is guaranteed ($u_e=p_e-v_e \geq 0$).

M_{SPT} is truthful

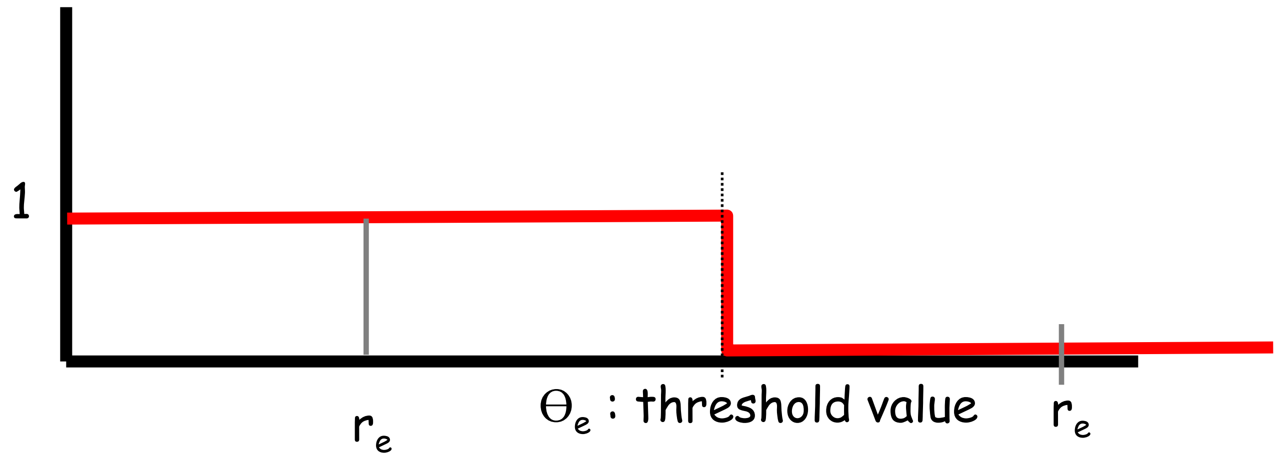
M_{SPT} is truthful, since it is an OP-mechanism. Indeed, Dijkstra's algorithm is **monotone**, since the workload for a_e has always the following shape:



where Θ_e is the value such that, once fixed r_{-e}:

- if a_e reports at most Θ_e, then e is selected in the SPT
- if a_e reports more than Θ_e, then e is not selected in the SPT

On the payments



- p_e=0, if e is not selected

$$p_e = r_e\, w_e(r) + \int_{r_e}^{\infty} w_e(r_{-e},z)\, dz = 0+0 = 0$$

- p_e=Θ_e, if e is selected

$$p_e = r_e\, w_e(r) + \int_{r_e}^{\infty} w_e(r_{-e},z)\, dz = r_e + \Theta_e - r_e = \Theta_e$$

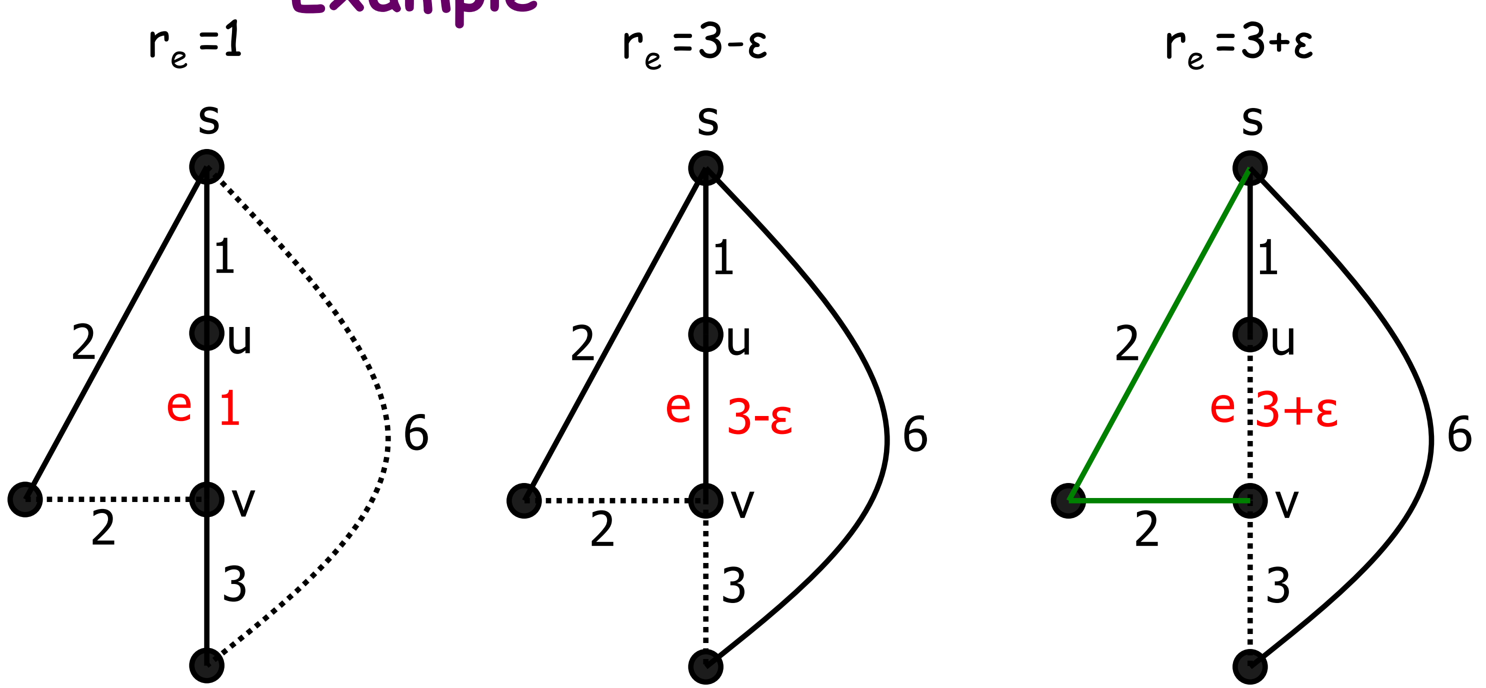
On the threshold values for the SPT problem

Let $e=(u,v)$ be an edge in $S_G(s)$ (with u closer to s than v)

- e remains in $S_G(s)$ until e is used to reach v from s

$\Rightarrow \Theta_e = d_{G-e}(s,v) - d_G(s,u)$

Example



$\Rightarrow \Theta_e = 3$

A trivial solution to find Θ_e

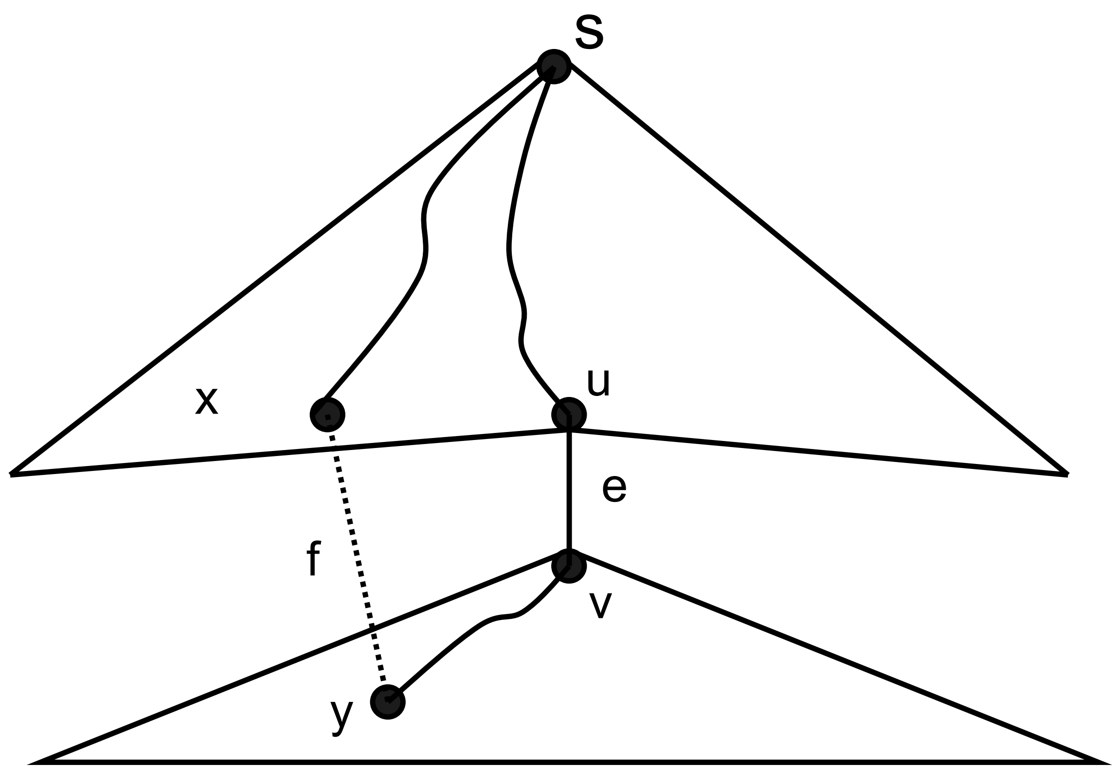
$\forall e=(u,v) \in S_G(s)$ we run the Dijkstra's algorithm on $G-e=(V,E\setminus\{e\},r)$ to find $d_{G-e}(s,v)$

Time complexity: $k=n-1$ edges multiplied by $O(m + n \log n)$ time (Dijkstra with Fibonacci Heaps): $O(mn + n^2 \log n)$ time

The improved solution will cost as much as:

$O(m + n \log n)$ time

Definition of Θ_e



$$d_{G-e}(s,v)= \min_{f=(x,y)\in C(e)} \{d_G(s,x)+ r_f +d_G(y,v)\}$$

Observation: the quality of a crossing edge depends on the considered edge e, since of the quantity $d_G(y,v)$

Definition of Θ_e (cont'd)

- Computing $d_{G-e}(s,v)$ (and then Θ_e) is equivalent to finding an edge f^* such that:

$$f^* = \arg\min_{f=(x,y)\in C(e)} \{d_G(s,x) + r_f + d_G(y,v)\}$$

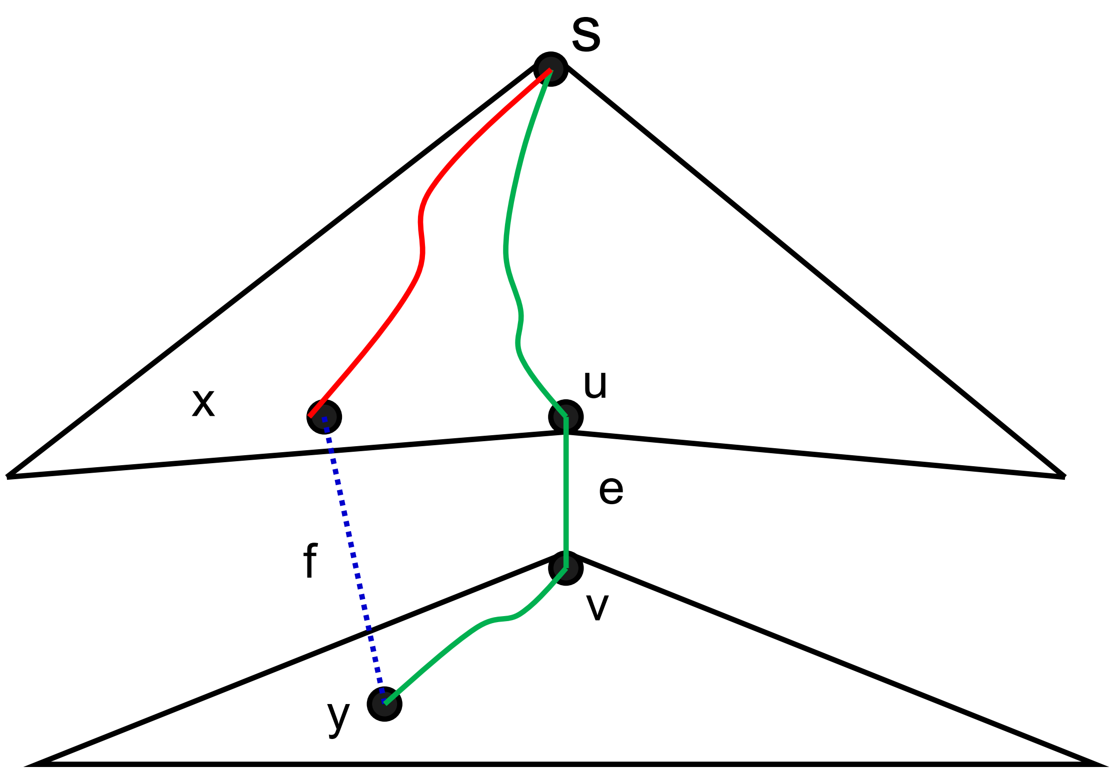
$$= \arg\min_{f=(x,y)\in C(e)} \{d_G(s,x) + r_f + d_G(y,v) + d_G(s,v)\}$$

$$= \arg\min_{f=(x,y)\in C(e)} \underbrace{\{d_G(s,x) + r_f + d_G(s,y)\}}_{\text{call it } k(f)}$$

Since $d_G(s,v)$ is independent of f

Observation: $k(f)$ is now a value univocally associated with edge f and is independent of e: it is the length of the **fundamental cycle** of f w.r.t. $S_G(s)$, and it will stay the same for all the edges of $S_G(s)$ on such a cycle (i.e., edges of $S_G(s)$ for which f is a crossing edge)

Definition of k(f)



k(f) will stay the same for all the edges on the green and on the red path: is the length of the fundamental cycle!

$$k(f) = d_G(s,x)+r_f+d_G(s,y)$$

Computing the threshold

- Once again, we build a **transmuter** (w.r.t. $S_G(s)$), where now sink nodes are labelled with k(f) (instead of w(f) as in the MST case), and we process the transmuter in reverse topological order
- At the end of the process, every edge $e \in S_G(s)$ will remain associated with its replacement edge f^*, and

$$\Theta_e = \underbrace{(k(f^*) - d_G(s,v))}_{d_{G-e}(s,v)} - d_G(s,u)$$

- Time complexity: $O(m\,\alpha(m,n))$, once that d_G are given

Analysis

Theorem

M_{SPT} can be implemented in $O(m + n \log n)$ time.

Proof:

Time complexity of g: $O(m + n \log n)$ (Dijkstra with Fibonacci Heaps)

Computing all the payments costs: $O(m\,\alpha(m,n))=O(m + n \log n)$ time

↓

Since $\alpha(m,n)$ is constant when $m=\Omega(n \log n)$: indeed, $\alpha(m,n)= \min\{i>0: A(i, \lfloor m/n \rfloor)>\log n\}$, and then $\alpha(n \log n,n)= \min\{i>0: A(i, \log n)>\log n\}=1$, since $A(1, \log n)=2^{\log n}=n>\log n$.

■

This is the end
My only friend, the end…
(The Doors, 1967)

Thanks for your attention!