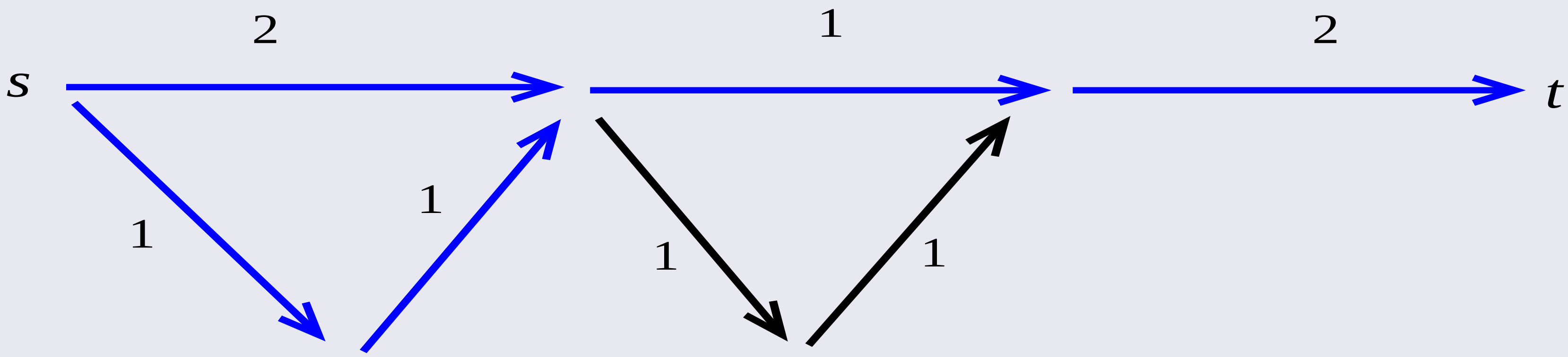
Nucleolus of shortest path games

Mourad Baïou, CNRS and Université Clermont Auvergne

Francisco Barahona, IBM T.J. Watson Research Center

Shortest Path Games



revenue: 7

cost of a shortest path: 5

profit: 7-5

how much extra to pay in each arc?

In other words we want to define the relative importance of each arc to ensure the connectivity between s and t.

Shortest Path Games

Shortest Path Games

Directed graph $G = (V, A)$, distinguished nodes s, t, $c : A \to \mathbb{R}_+$

Shortest Path Games

Directed graph $G = (V, A)$, distinguished nodes s, t, $c : A \to \mathbb{R}_+$

Cooperative game, coalition $S \subset A$

$$\mathbf{v}(S) = \begin{cases} r - m(S) & \text{if } m(S) < \infty, \\ 0 & \text{otherwise,} \end{cases}$$

$r > 0$ is a reward

$m(S) = \min\{c(P) \mid P \subseteq S,\ P \text{ is an } st\text{-path and } c(P) \leq r\}$.

Shortest Path Games

Directed graph $G = (V, A)$, distinguished nodes s, t, $c : A \to \mathbb{R}_+$

Cooperative game, coalition $S \subset A$

$$\mathbf{v}(S) = \begin{cases} r - m(S) & \text{if } m(S) < \infty, \\ 0 & \text{otherwise,} \end{cases}$$

$r > 0$ is a reward

$m(S) = \min\{c(P) \mid P \subseteq S,\ P \text{ is an } st\text{-path and } c(P) \leq r\}$.

Introduced by Fragnelli et al. (2000)

Core

$$\{x \in \mathbb{R}^A \mid x(A) = \mathbf{v}(A),\ x(S) \geq \mathbf{v}(S),\ \text{for } S \subseteq A\}.$$

Core

$$\{x \in \mathbb{R}^A \mid x(A) = \mathbf{v}(A),\ x(S) \geq \mathbf{v}(S),\ \text{for } S \subseteq A\}.$$

Let λ be the value of a shortest path

Core:

$$x(A) = r - \lambda$$
$$x(P) \geq r - c(P) \quad \text{for each } st\text{-path } P$$
$$x \geq 0$$

Core

$$\{x \in \mathbb{R}^A \,|\, x(A) = \mathbf{v}(A),\; x(S) \geq \mathbf{v}(S),\; \text{for } S \subseteq A\}.$$

Let λ be the value of a shortest path

Core:

$$\begin{aligned} x(A) &= r - \lambda \\ x(P) &\geq r - c(P) \quad \text{for each } st\text{-path } P \\ x &\geq 0 \end{aligned}$$

$x(a) > 0 \Longrightarrow$ a is in the intersection of all shortest paths

Core

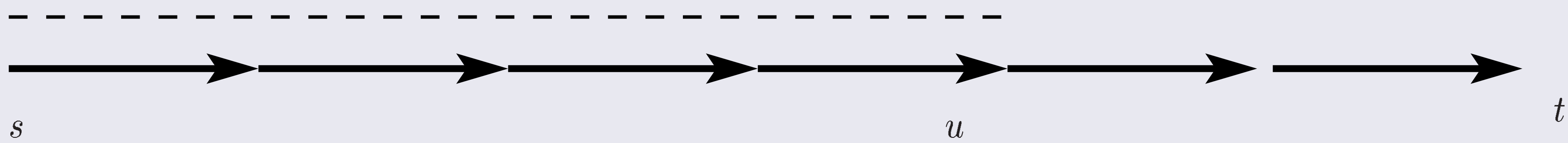
$$x(A) = r - \lambda$$
$$x(P) \geq r - c(P) \quad \text{for each } st\text{-path } P$$
$$x \geq 0$$

Let Π be one (fixed) shortest path
Let $z = x + c$
Core

$$z(\Pi) = r$$
$$z(P) \geq r \quad \text{for each } st\text{-path } P$$
$$z \geq c$$
$$z(a) = c(a) \quad \text{if } a \notin \Pi$$

Core

One more change of variables: $p_z(u) = z(\Pi_{su})$ (sum of the variables z in the sub-path from s to u)

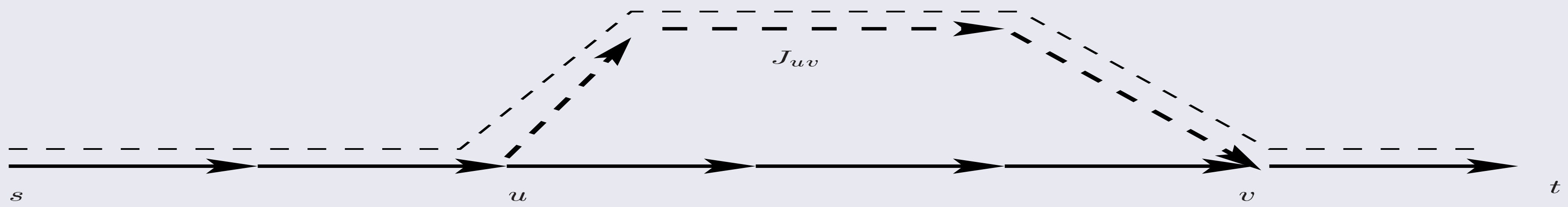


$$p_z(s) = 0, \quad p_z(t) = r$$

$$p_z(v) - p_z(u) \geq c(u,v) \text{ for } (u,v) \in \Pi$$

Core

A jump from u to v is a path that intersects Π only in $\{u, v\}$

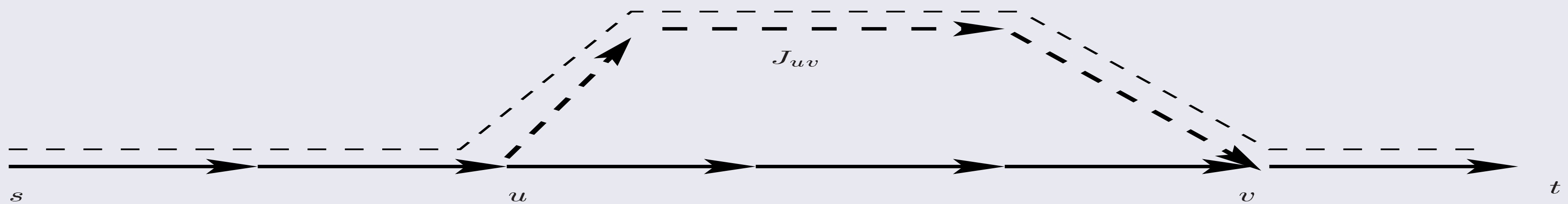


If P contains one jump, $z(P) \geq r$ is equivalent to

$$
\begin{aligned}
z(P_{su}) + z(J_{uv}) + z(P_{vt}) &\geq r, \\
p_z(u) + c(J_{uv}) + (z(\Pi) - z(P_{sv})) &\geq r, \\
p_z(u) + c(J_{uv}) + r - p_z(v) &\geq r, \\
p_z(u) - p_z(v) &\geq -c(J_{uv})
\end{aligned}
$$

Core

A jump from u to v is a path that intersects Π only in $\{u, v\}$



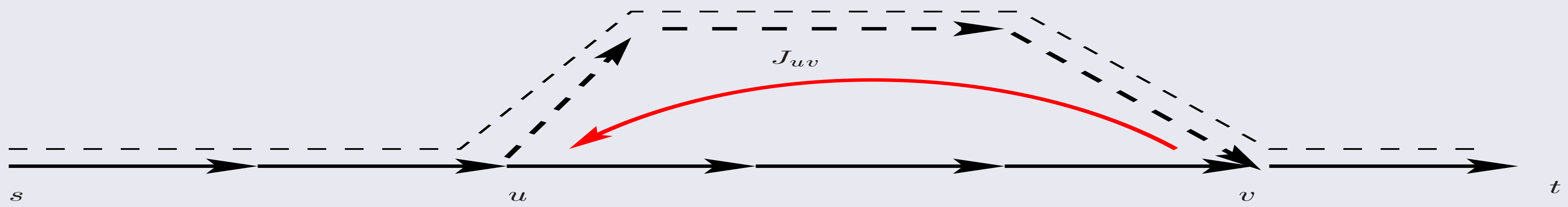
If P contains one jump, $z(P) \geq r$ is equivalent to

$$
\begin{aligned}
z(P_{su}) + z(J_{uv}) + z(P_{vt}) &\geq r, \\
p_z(u) + c(J_{uv}) + (z(\Pi) - z(P_{sv})) &\geq r, \\
p_z(u) + c(J_{uv}) + r - p_z(v) &\geq r, \\
p_z(u) - p_z(v) &\geq -c(J_{uv})
\end{aligned}
$$

Paths with more than one jump are not needed

Core

A jump from u to v is a path that intersects Π only in $\{u, v\}$



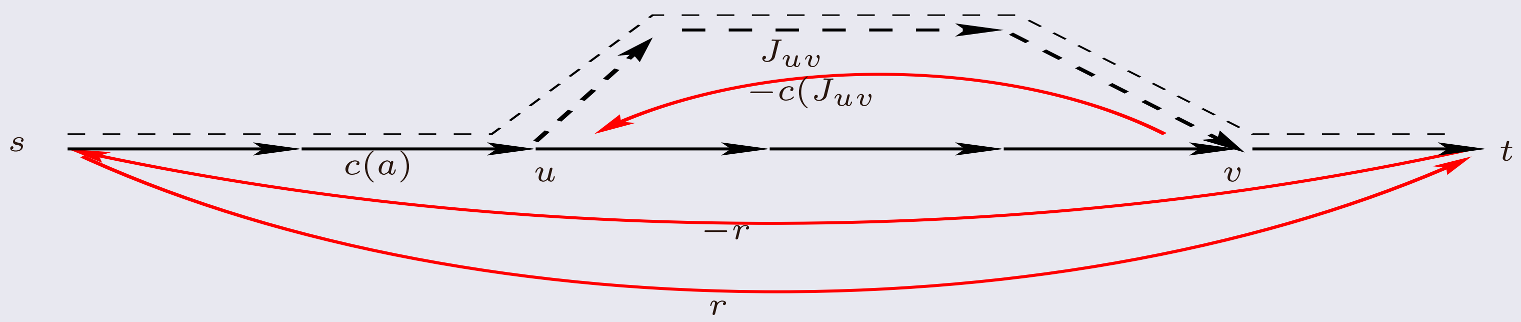
If P contains one jump, $z(P) \geq r$ is equivalent to

$$z(P_{su}) + z(J_{uv}) + z(P_{vt}) \geq r,$$
$$p_z(u) + c(J_{uv}) + (z(\Pi) - z(P_{sv})) \geq r,$$
$$p_z(u) + c(J_{uv}) + r - p_z(v) \geq r,$$
$$p_z(u) - p_z(v) \geq -c(J_{uv})$$

Paths with more than one jump are not needed

Only shortest jumps are needed

Core

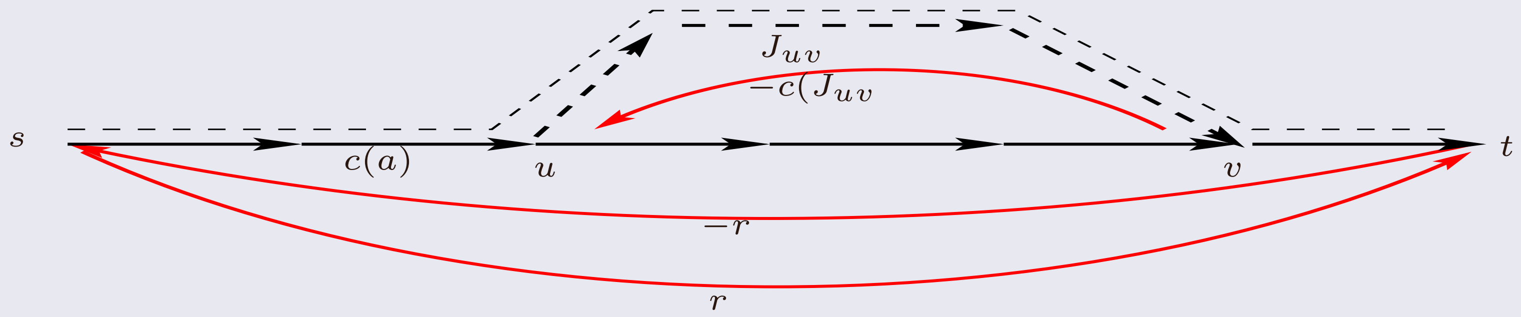


Let V' be the set of nodes of Π and A' the set of arcs composed by Π and the red arcs.
Define $d(u,v) = c(u,v)$ if $(u,v) \in \Pi$

$d(u,v) = -c(J_{u,v})$ if $(u,v) \notin \Pi$,
where $J_{u,v}$ is shortest jump from u to v

The core is defined by: $p_z(v) - p_z(u) \geq d(u,v) \quad (u,v) \in A'$

Core



Let V' be the set of nodes of Π and A' the set of arcs composed by Π and the red arcs.
Define $d(u, v) = c(u, v)$ if $(u, v) \in \Pi$

$d(u, v) = -c(J_{u,v})$ if $(u, v) \notin \Pi$,
where $J_{u,v}$ is shortest jump from u to v

The core is defined by: $p_z(v) - p_z(u) \geq d(u, v) \quad (u, v) \in A'$

Polynomial number of inequalities. Based on that Granot et al. (1998) showed that computing the nucleolus (when the core is nonempty) reduces to a sequence of combinatorial linear programs.

Core

Dual of a network flow problem

$$\max \sum_{(u,v)\in A'} d(u,v)x(u,v)$$

$$\sum_{(u,v)\in A'} x(u,v) - \sum_{(v,u)\in A'} x(v,u) = 0, \qquad \text{for } v \in V',$$

$$x(u,v) \geq 0, \quad \text{for all } (u,v) \in A'.$$

This problem is unbounded if and only if there is no cycle with positive cost. Thus the dual problem has a solution if and only if there is no cycle with positive cost.

The nucleolus

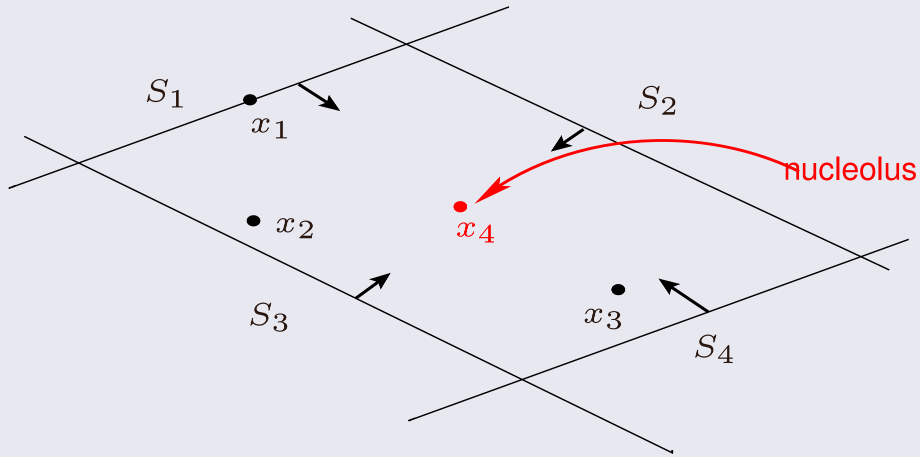
$$x(A) = \mathbf{v}(A)$$
$$x(S) \geq \mathbf{v}(S) \quad \forall S \subset A$$

The nucleolus

$$x(A) = \mathbf{v}(A)$$
$$x(S) \geq \mathbf{v}(S) \quad \forall S \subset A$$

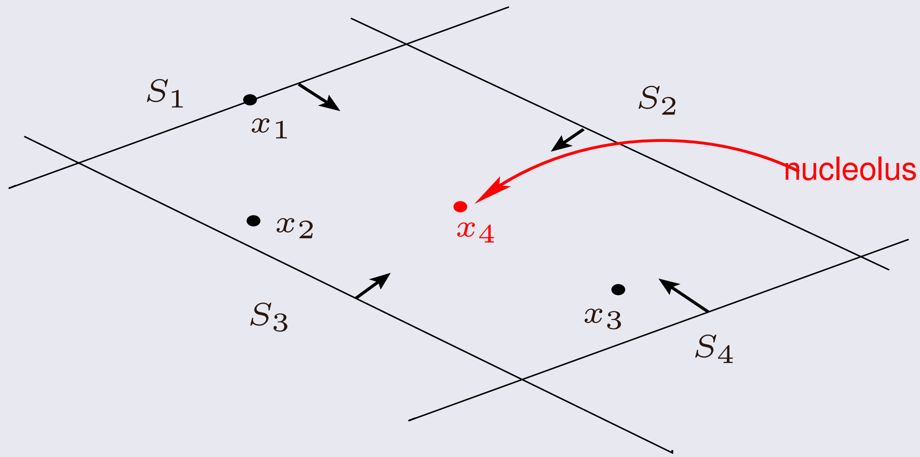
For a coalition S and a vector $x \in \mathbb{R}^A$, their excess is $e(x, S) = x(S) - v(S)$.
The nucleolus has been introduced Schmeidler (1969), trying to minimize dissatisfaction of players. Schmeidler defined it as the allocation that lexicographically maximize the vector of non-decreasing ordered excess.

The nucleolus

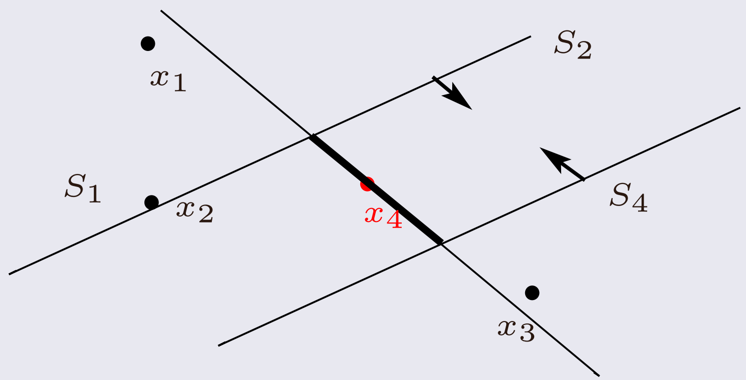


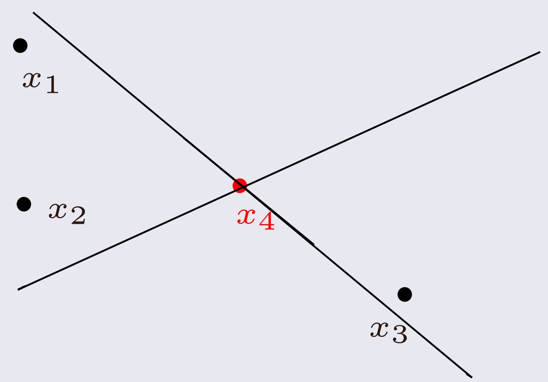
$$x_1 \preceq x_2 \preceq x_3 \preceq x_4$$

The nucleolus



$$x_1 \preceq x_2 \preceq x_3 \preceq x_4$$





Nucleolus

$$
\begin{aligned}
&\max \quad \epsilon \\
&x(A) = \mathbf{v}(A) \\
&x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \neq A
\end{aligned}
$$

Nucleolus

$$\max \quad \epsilon$$
$$x(A) = \mathbf{v}(A)$$
$$x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \neq A$$

ϵ_1 optimal value, $P_1(\epsilon_1)$ set of optimal solutions

Nucleolus

$$\max \quad \epsilon$$
$$x(A) = \mathbf{v}(A)$$
$$x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \neq A$$

ϵ_1 optimal value, $P_1(\epsilon_1)$ set of optimal solutions
F_1: coalitions S with $x(S) = \mathbf{v}(S) + \epsilon_1$ for all $x \in P_1(\epsilon_1)$.

Nucleolus

$$\max \quad \epsilon$$
$$x(A) = \mathbf{v}(A)$$
$$x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \neq A$$

ϵ_1 optimal value, $P_1(\epsilon_1)$ set of optimal solutions
F_1: coalitions S with $x(S) = \mathbf{v}(S) + \epsilon_1$ for all $x \in P_1(\epsilon_1)$.

$$\max \quad \epsilon$$
$$x(A) = \mathbf{v}(A)$$
$$x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \neq A$$
$$x(S) = \mathbf{v}(S) + \epsilon_1, \quad \forall S \in F_1$$

Nucleolus

$$\begin{aligned} &\max \quad \epsilon \\ &x(A) = \mathbf{v}(A) \\ &x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \neq A \end{aligned}$$

ϵ_1 optimal value, $P_1(\epsilon_1)$ set of optimal solutions
F_1: coalitions S with $x(S) = \mathbf{v}(S) + \epsilon_1$ for all $x \in P_1(\epsilon_1)$.

$$\begin{aligned} &\max \quad \epsilon \\ &x(A) = \mathbf{v}(A) \\ &x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \neq A \\ &x(S) = \mathbf{v}(S) + \epsilon_1, \quad \forall S \in F_1 \end{aligned}$$

This gives ϵ_2 ... continue ... (at most m times)

Nucleolus

$$\max \epsilon$$

$$p_z(s) = 0, \quad p_z(t) = r$$

$$p_z(v) - p_z(u) \geq d(u, v) + \epsilon$$

Nucleolus

$$\max \epsilon$$

$$p_z(s) = 0, \quad p_z(t) = r$$

$$p_z(v) - p_z(u) \geq d(u, v) + \epsilon$$

This system has a solution iff the graph with weights $d(u, v) + \epsilon$ has no cycle of positive weight.

Nucleolus

$$\max \epsilon$$
$$p_z(s) = 0, \quad p_z(t) = r$$
$$p_z(v) - p_z(u) \geq d(u, v) + \epsilon$$

This system has a solution iff the graph with weights $d(u, v) + \epsilon$ has no cycle of positive weight.

$$d(C) + n(C)\epsilon \leq 0$$

for every cycle C

Nucleolus

$$\max \epsilon$$
$$p_z(s) = 0, \quad p_z(t) = r$$
$$p_z(v) - p_z(u) \geq d(u, v) + \epsilon$$

This system has a solution iff the graph with weights $d(u, v) + \epsilon$ has no cycle of positive weight.

$$d(C) + n(C)\epsilon \leq 0$$

for every cycle C

$$\epsilon \leq -d(C)/n(C)$$

Nucleolus

$$\max \epsilon$$
$$p_z(s) = 0, \quad p_z(t) = r$$
$$p_z(v) - p_z(u) \geq d(u,v) + \epsilon$$

This system has a solution iff the graph with weights $d(u,v) + \epsilon$ has no cycle of positive weight.

$$d(C) + n(C)\epsilon \leq 0$$

for every cycle C

$$\epsilon \leq -d(C)/n(C)$$

$$\epsilon = \min_C \frac{-d(C)}{n(C)} \quad \text{min ratio cycle}, \quad O(n^3)$$

Algorithm for nucleolus

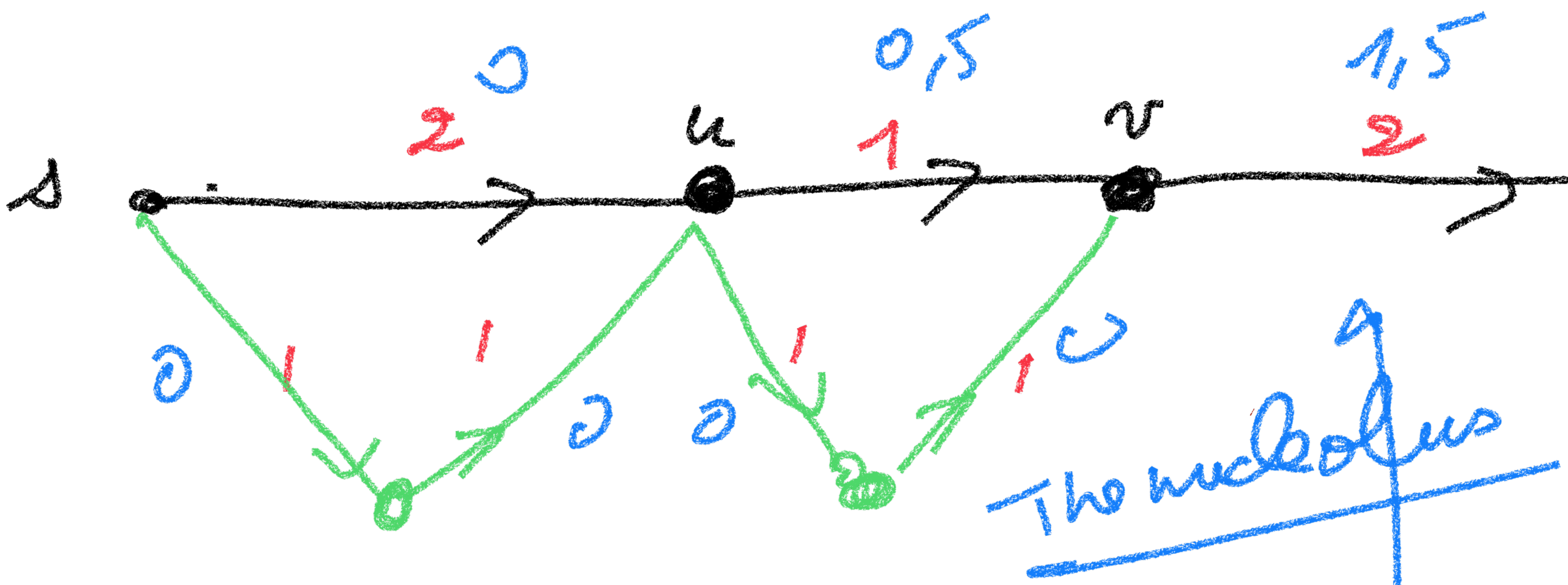
Find min ratio cycle. Fix variables on this cycle

Algorithm for nucleolus

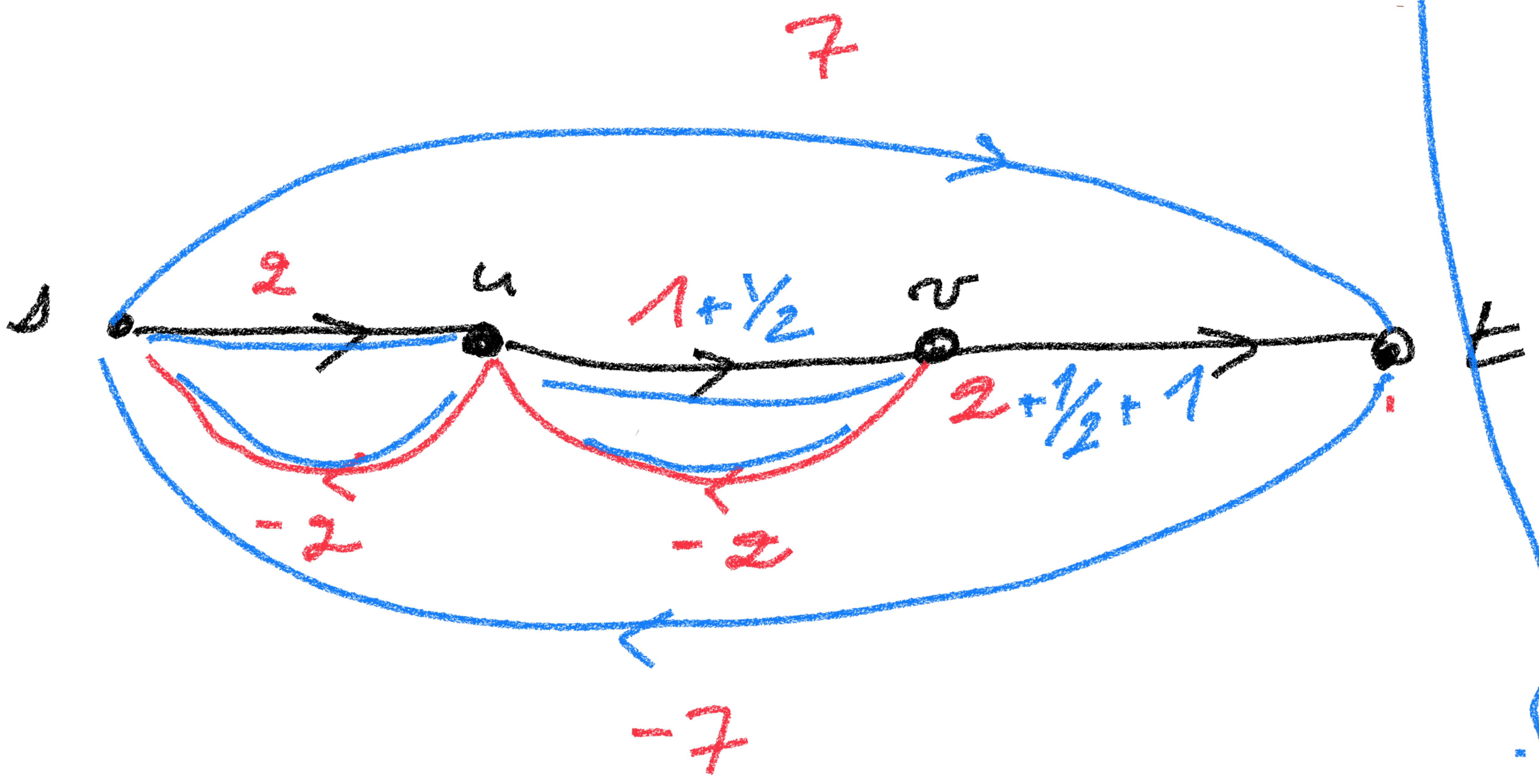
Find min ratio cycle. Fix variables on this cycle

Find min ratio cycle involving non-fixed variables. Fix variables on the new cycle, continue.

Example: $r = 7$



blues arcs are fixed



$p_t - p_v = 7 - 3{,}5 = 3{,}5$

$p_v - p_u = 1{,}5$

$p_u - p_s = 2$

$p_u = 2\ ,\ p_v = 3{,}5$

$\Longrightarrow$

$x(t,v) = 7 - 3{,}5 = 3{,}5$

$x(u,v) = 1{,}5$

$x(s,u) = 2$

$\Longrightarrow$

$x(v,t) = 1{,}5$

$x(u,v) = ½$

Nucleolus when the core is empty

When the core is empty the solution of the program below is $\epsilon_1 < 0$.

$$\begin{aligned} &\max \quad \epsilon \\ &x(A) = \mathbf{v}(A) \\ &x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \neq A \end{aligned}$$

We use parametric linear programming and look for the maximum value of the parameter $\epsilon < 0$, so that the value of the parametric linear program below is $r - \lambda$.

$$\begin{aligned} &\min \quad x(A) \\ &x(P) \geq r + \epsilon - c(P), \quad \forall st - \text{path } P, \\ &x \geq 0. \end{aligned}$$

Nucleolus when the core is empty

The dual of this problem is:

$$\begin{aligned}&\max \quad \sum_{P} (r + \epsilon - c(P)) y_P \\ &\sum_{a \in P} y_P \le 1, \quad \forall a \in A, \\ &y \ge 0.\end{aligned}$$

We reduce this problem to a network flow problem:

- We add an arc from t to s with cost $r + \epsilon$.
- each arc $a \in A$ receive the cost $-c(a)$.

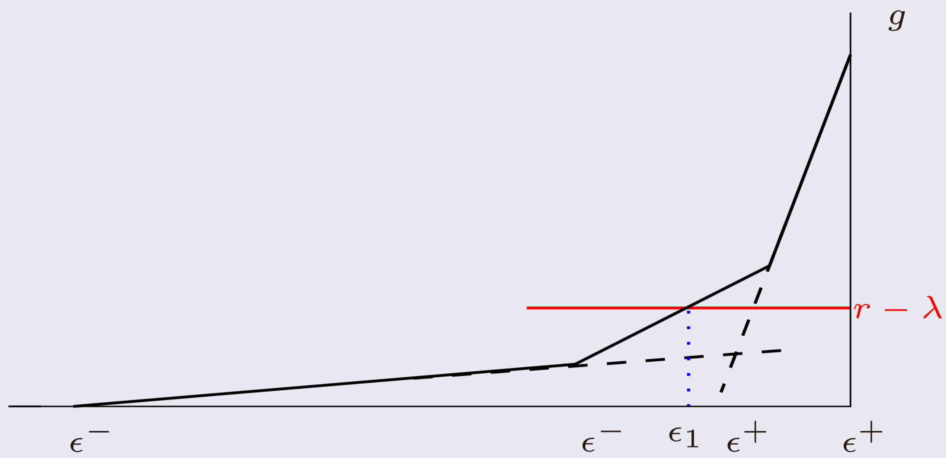
Then we look for a maximum circulation cost.

Capacities 1 implies: there is an optimal circulation that corresponds to a set of arc-disjoint st-paths of minimum cost.

Nucleolus when the core is empty

Thus the optimal value of the dual problem my be written as:

$$g(\epsilon) = \max_{k}\{k(r+\epsilon) - f(k)\}$$



Nucleolus when the core is empty

Thus the optimal value of the dual problem my be written as:

$$g(\epsilon) = \max_k \{k(r+\epsilon) - f(k)\}$$

Similar approach is used:
change variables, use the dual of a network flow problem

solve a sequence of min ratio cycles