The nucleolus and related notions in cooperative games

Kinyó Kincső

Supervisor: Király Tamás

June, 2026

Cooperative games

- player set: $N = \{1, 2, \dots n\}$
- coalition: $S \subseteq N$
- value function: $v : 2^N \to \mathbb{R}$ or cost function $c = -v$
- allocation: $x \in \mathbb{R}^n$, $I^*(N, v)$, $I(N, v)$, $C(N, v)$
- profit: $p(S, x) := x(S) - v(S)$, profit vector: $\Theta(x) \in \mathbb{R}^{2^n}$ which contains all the profit values of different coalitions in non-decreasing order
- **nucleolus**: $N(v) = \{x \in I(N, v) | \quad \Theta(y) \preccurlyeq \Theta(x) \quad \forall y \in I(N, v)\}$
- examples: bankruptcy problem, glove market

Standard tree game

Tree network:

- Γ tree network is given by a finite (V, E) tree, $|E| = q$, root: $o \in V$
- cost function on the edges: $a : E \to \mathbb{R}$, cost function on the nodes: $b : E \to \mathbb{R}$
- $\forall p \in V$ residents of p: N_p, player set $\bigcup N_p = N$
- unique road from $p \in V \setminus \{o\}$ to o; For every node q contained by this unique path from o to p: $p \geq q$ holds
- trunks: $T = (V(T), E(T))$ is a subgraph of (V, E), spanned by the nodes $V(T)$ that is closed under the precedence relation

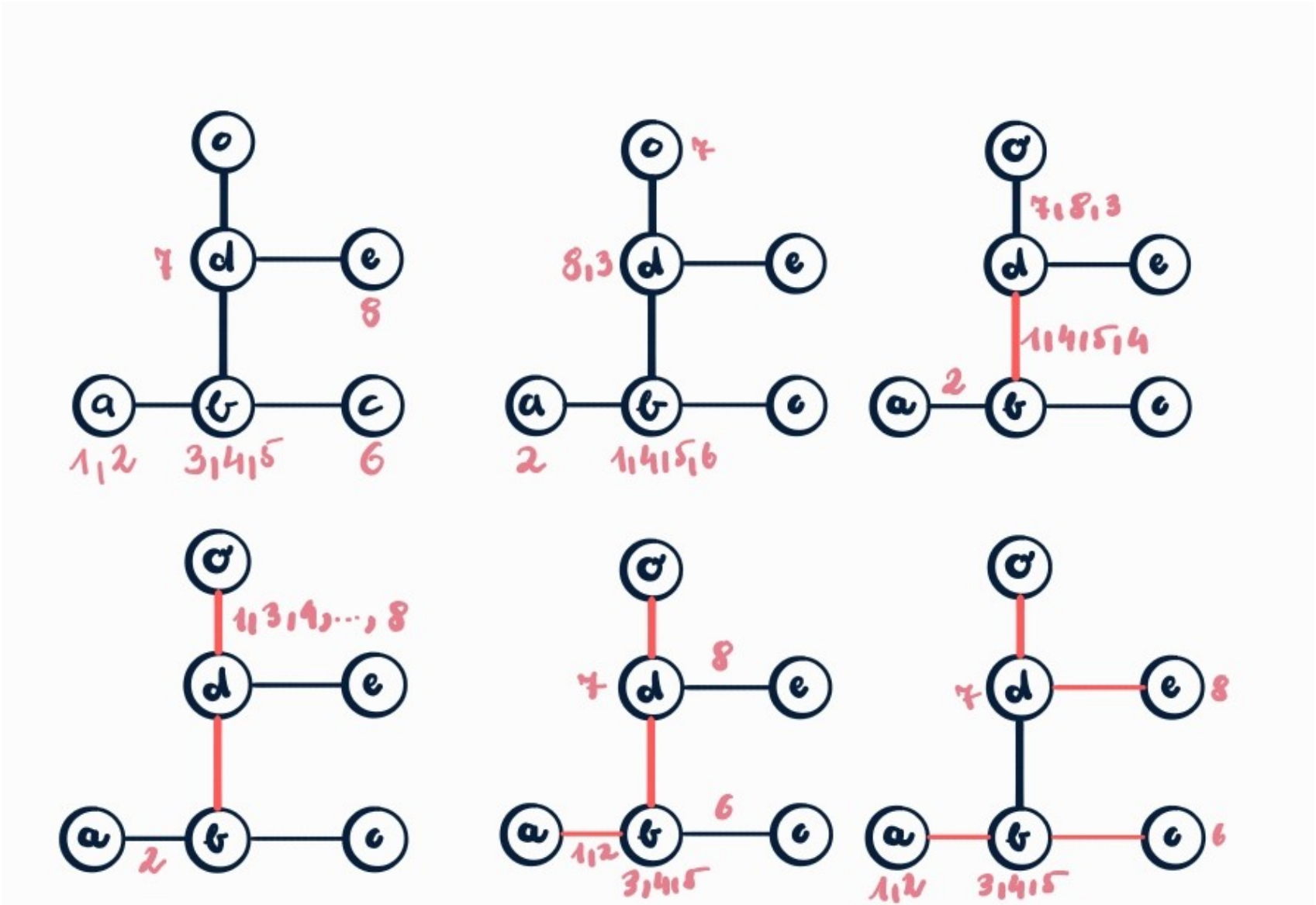
Standard tree network:

- $a \geq 0$, $b = 0$
- $N_p \neq \emptyset$ for every leaf p
- **Lemma**: *Let Γ be a tree network such that all trunks have non-negative costs. Then there exists a standard tree network Γ' that generates the same cost.*

Finding the nucleolus

- Let $\mathcal{E}$ be a certain subset of the coalitions: $N \setminus \{i\}, (i \in N)$ and $N(T_p), (p \in V \setminus \{o\})$.
- **Lemma**: *Let* Γ *be a standard tree network with player set* N *and let* $x \in \mathbb{R}^n$. *Then* $x \in core(N, c)$ *if and only if* $x(N) = c(N)$ *and* $x(S) \leq c(S)$ *for all* $S \in \mathcal{E}$.
- **Lemma**: *Let* Γ *be a standard tree network then the nucleolus of* (N, c) *is the imputation* x *maximizes lexicographically* $\{exc(S, x) : S \in \mathcal{E}\}$.

Painting story



Algorithm for the standard tree game

variables for the algorithm:

- $d_p := |\{q \in V : \pi(q) = p\}|$
- $i_p := \begin{cases} 1 & \text{if } \pi(p) = o \\ 0 & \text{if } \pi(p) \neq o \end{cases}$
- $r(p)$: to which verticies the original nodes have been contracted

main step of the iterations: $y := \max\{x \geq 0 : a_p - xg_p \geq 0\}$
running time: $O(q^2)$

AI usage

- grammar correction
- Latex codes, editing the document