Gross Substitutes: Representation and Approximation

Renato Paes Leme
(Google Research)

Joint work with Eric Balkanski, Shahar Dobzinski, Michal Feldman, Uri Feige and Mike Ostrovsky.

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

Discrete derivatives:

$$\partial_i v(S) = v(S \cup \{i\}) - v(S)$$

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

Discrete derivatives:

$$\partial_i v(S) = v(S \cup \{i\}) - v(S)$$

$$\partial_i \partial_j v(S) = \partial_j v(S \cup \{i\}) - \partial_j v(S)$$

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

Discrete derivatives:

$$\partial_i v(S) = v(S \cup \{i\}) - v(S)$$

$$\begin{aligned}\partial_i \partial_j v(S) &= \partial_j v(S \cup \{i\}) - \partial_j v(S) \\ &= v(S \cup \{i, j\}) - v(S \cup \{i\}) - v(S \cup \{j\}) + v(S)\end{aligned}$$

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes if for any prices $p \in \mathbb{R}^n$, $S \in D(v; p)$ and $p' \geq p$ it exists $S' \in D(v, p')$ s.t. $S' \supseteq S \cap \{i; p_i = p'_i\}$

Demand : $D(v; p) = \underset{S}{\operatorname{argmax}} \left[v(S) - \sum_{i \in S} p_i \right]$

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes if for any prices $p \in \mathbb{R}^n$, $S \in D(v; p)$ and $p' \geq p$ it exists $S' \in D(v, p')$ s.t. $S' \supseteq S \cap \{i; p_i = p'_i\}$

Necessary and sufficient conditions for Walrasian tatonnement to converge.

[Kelso-Crawford] [Gul-Stachetti]

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes if for any prices $p \in \mathbb{R}^n$, $S \in D(v; p)$ and $p' \geq p$ it exists $S' \in D(v, p')$ s.t. $S' \supseteq S \cap \{i; p_i = p'_i\}$

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes if for any price vector $p \in \mathbb{R}^n$ the greedy algorithm computes $S \in D(v; p)$.

Start with $S = \emptyset$ and keep adding $i \in \operatorname{argmax}_i v(S \cup \{i\}) - v(S) - p_i$ while it is positive.

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes if for any prices $p \in \mathbb{R}^n$, $S \in D(v; p)$ and $p' \geq p$ it exists $S' \in D(v, p')$ s.t. $S' \supseteq S \cap \{i; p_i = p'_i\}$

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes if for any price vector $p \in \mathbb{R}^n$ the greedy algorithm computes $S \in D(v; p)$.

What are gross substitutes?

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes iff it satisfies the following discrete inequality: $\partial_i \partial_j v(S) \leq \max\{\partial_i \partial_k v(S), \partial_k \partial_j v(S)\} \leq 0, \forall i, j, k \notin S$

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes if for any prices $p \in \mathbb{R}^n$, $S \in D(v; p)$ and $p' \geq p$ it exists $S' \in D(v, p')$ s.t. $S' \supseteq S \cap \{i; p_i = p'_i\}$

A function $v : 2^{[n]} \to \mathbb{R}$ is gross substitutes if for any price vector $p \in \mathbb{R}^n$ the greedy algorithm computes $S \in D(v; p)$.

Other definitions through Discrete Convex Analysis, Tropical Algebraic Geometry, ...

How complex are gross substitutes?

Approach #1: Constructive characterization

Approach #2: Approximation by other classes

How complex are gross substitutes?

Approach #1: Constructive characterization

GS function

Approach #2: Approximation by other classes

How complex are gross substitutes?

Approach #1: Constructive characterization

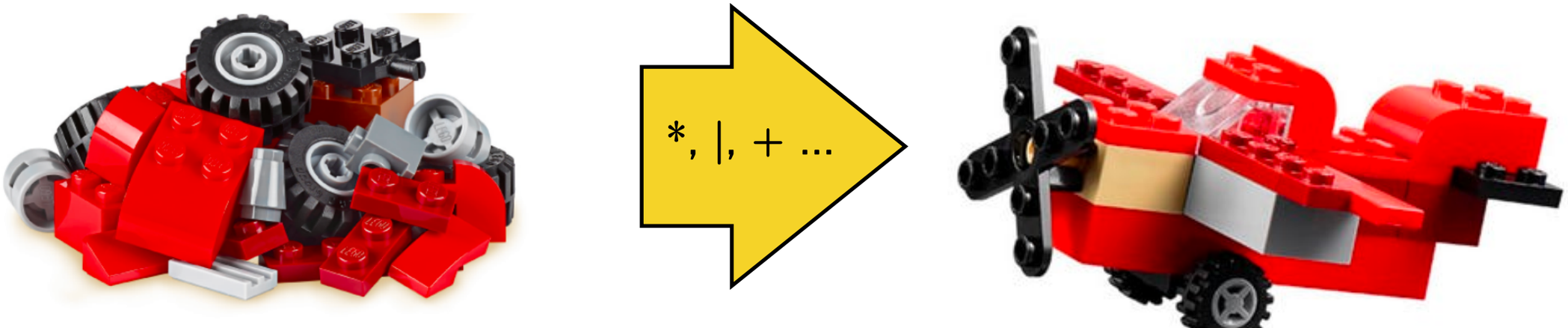
simpler functions

GS function

Approach #2: Approximation by other classes

How complex are gross substitutes?

Approach #1: Constructive characterization



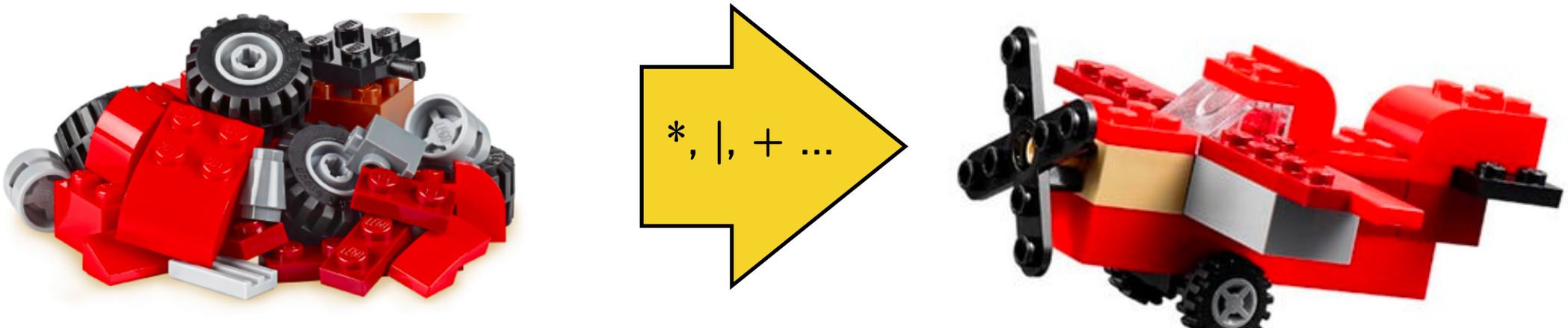
simpler functions

GS function

Approach #2: Approximation by other classes

How complex are gross substitutes?

Approach #1: Constructive characterization



simpler functions

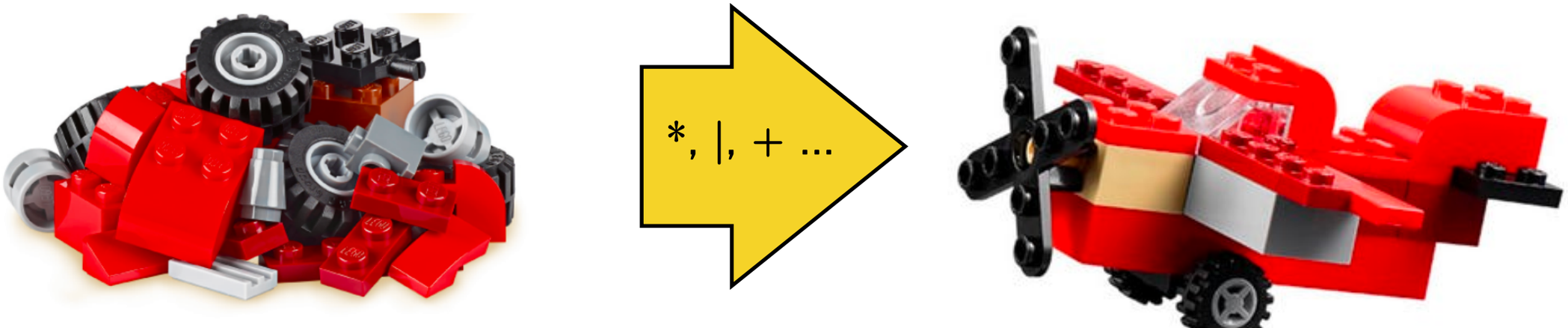
GS function

Approach #2: Approximation by other classes

GS

How complex are gross substitutes?

Approach #1: Constructive characterization



simpler functions

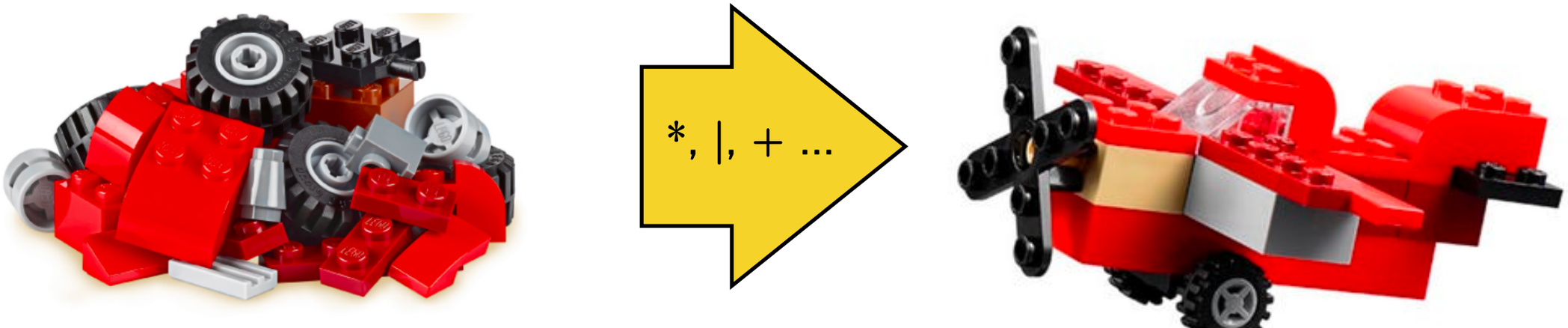
GS function

Approach #2: Approximation by other classes

GS $\subseteq$ Submodular

How complex are gross substitutes?

Approach #1: Constructive characterization



simpler functions

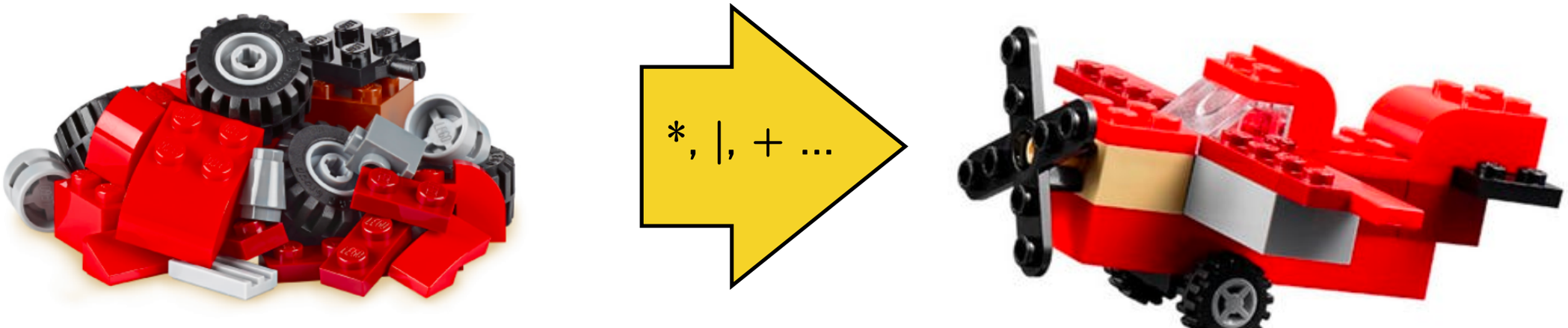
GS function

Approach #2: Approximation by other classes

GS $\subseteq$ Submodular $\subseteq$ XOS

How complex are gross substitutes?

Approach #1: Constructive characterization



simpler functions

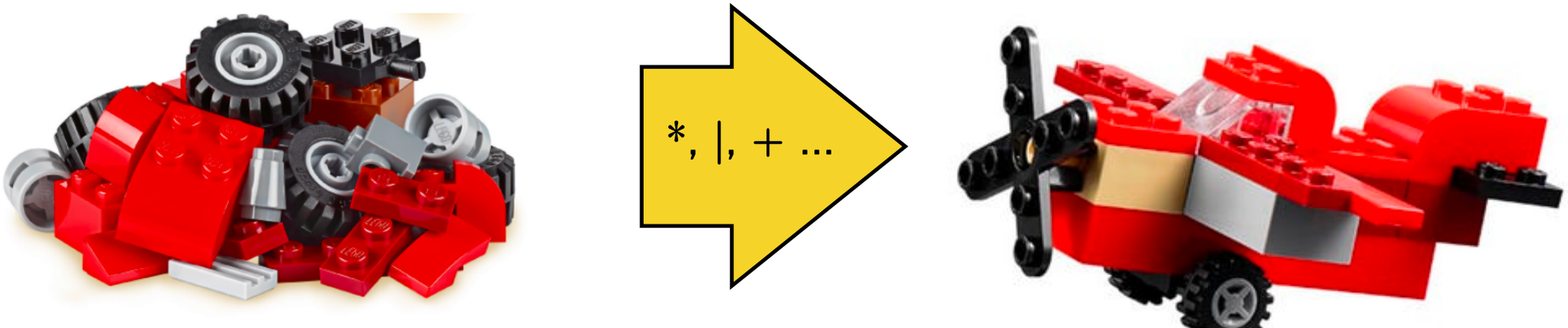
GS function

Approach #2: Approximation by other classes

GS $\subseteq$ Submodular $\subseteq$ XOS $\subseteq$ Subadditive

How complex are gross substitutes?

Approach #1: Constructive characterization



simpler functions

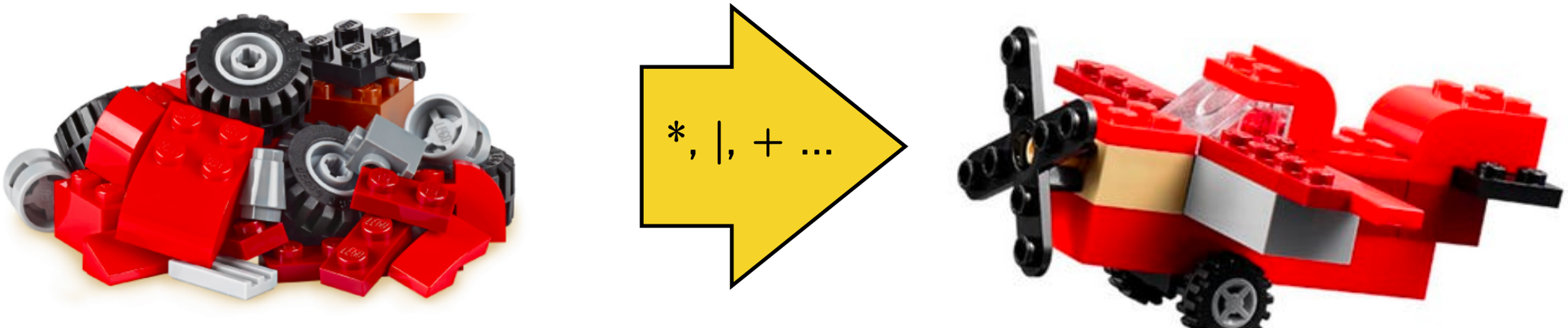
GS function

Approach #2: Approximation by other classes

Rado
Valuations $\subsetneq$ **GS** $\subsetneq$ Submodular $\subsetneq$ XOS $\subsetneq$ Subadditive

How complex are gross substitutes?

Approach #1: Constructive characterization



simpler functions

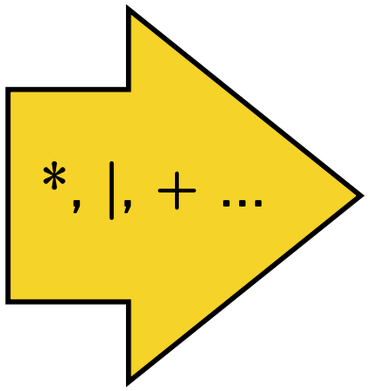
GS function

Approach #2: Approximation by other classes

Weighted Matroids
Assignment Valuations $\subseteq$ Rado Valuations $\subseteq$ **GS** $\subseteq$ Submodular $\subseteq$ XOS $\subseteq$ Subadditive

Part I: Toward a Constructive View

simpler functions



GS function

Operations Preserving GS

- Convolution: given $u, v : 2^{[n]} \to \mathbb{R}$, then we define $u * v : 2^{[n]} \to \mathbb{R}$ as: $u * v(S) = \max_{T \subseteq S} u(T) + v(S \setminus T)$

- Endowment: given $v : 2^{[n]} \to \mathbb{R}$ and a set S we define $u : [n] \setminus S \to \mathbb{R}$ as: $u(T) = v(T|S)$

- Affine Transformations: given $v : 2^{[n]} \to \mathbb{R}$, $p \in \mathbb{R}^n$ and $v_0 \in \mathbb{R}$ we can define $u(S) = v_0 + p(S) + v(S)$

Operations Preserving GS

- Convolution: given $u, v : 2^{[n]} \to \mathbb{R}$, then we define $u * v : 2^{[n]} \to \mathbb{R}$ as: $u * v(S) = \max_{T \subseteq S} u(T) + v(S \setminus T)$

- Endowment: given $v : 2^{[n]} \to \mathbb{R}$ and a set S we define $u : [n] \setminus S \to \mathbb{R}$ as: $u(T) = v(T|S)$

- Affine Transformations: given $v : 2^{[n]} \to \mathbb{R}$, $p \in \mathbb{R}^n$ and $v_0 \in \mathbb{R}$ we can define $u(S) = v_0 + p(S) + v(S)$

but in general $u + v$ may not be in GS

Operations Preserving GS

- Convolution: given $u, v : 2^{[n]} \to \mathbb{R}$, then we define $u * v : 2^{[n]} \to \mathbb{R}$ as: $u * v(S) = \max_{T \subseteq S} u(T) + v(S \setminus T)$

- Endowment: given $v : 2^{[n]} \to \mathbb{R}$ and a set S we define $u : [n] \setminus S \to \mathbb{R}$ as: $u(T) = v(T|S)$

- Affine Transformations: given $v : 2^{[n]} \to \mathbb{R}$, $p \in \mathbb{R}^n$ and $v_0 \in \mathbb{R}$ we can define $u(S) = v_0 + p(S) + v(S)$

- Strong Quotient Sum: given v in GS and a matroid rank function r s.t. $v(S|T) \leq r(S|T), \forall S, T$ then $\alpha v + \beta w$ in GS for $\alpha, \beta \geq 0$

Operations Preserving GS

- Convolution: given $u, v : 2^{[n]} \to \mathbb{R}$, then we define $u * v : 2^{[n]} \to \mathbb{R}$ as: $u * v(S) = \max_{T \subseteq S} u(T) + v(S \setminus T)$

- Endowment: given $v : 2^{[n]} \to \mathbb{R}$ and a set S we define $u : [n] \setminus S \to \mathbb{R}$ as: $u(T) = v(T|S)$

- Affine Transformations: given $v : 2^{[n]} \to \mathbb{R}$, $p \in \mathbb{R}^n$ and $v_0 \in \mathbb{R}$ we can define $u(S) = v_0 + p(S) + v(S)$

- Strong Quotient Sum: given v in GS and a matroid rank function r s.t. $v(S|T) \leq r(S|T), \forall S, T$ then $\alpha v + \beta w$ in GS for $\alpha, \beta \geq 0$ [Shioura]

Operations Preserving GS

- Convolution: given $u, v : 2^{[n]} \to \mathbb{R}$, then we define $u * v : 2^{[n]} \to \mathbb{R}$ as: $u * v(S) = \max_{T \subseteq S} u(T) + v(S \setminus T)$

- Endowment: given $v : 2^{[n]} \to \mathbb{R}$ and a set S we define $u : [n] \setminus S \to \mathbb{R}$ as: $u(T) = v(T|S)$

- Affine Transformations: given $v : 2^{[n]} \to \mathbb{R}$, $p \in \mathbb{R}^n$ and $v_0 \in \mathbb{R}$ we can define $u(S) = v_0 + p(S) + v(S)$

- Tree-Concordant Sum: given u, v in GS then $u + v$ is in GS iff the Hessians of u, v share the same tree structure.

Operations Preserving GS

- Convolution: given $u, v : 2^{[n]} \to \mathbb{R}$, then we define $u * v : 2^{[n]} \to \mathbb{R}$ as: $u * v(S) = \max_{T \subseteq S} u(T) + v(S \setminus T)$

- Endowment: given $v : 2^{[n]} \to \mathbb{R}$ and a set S we define $u : [n] \setminus S \to \mathbb{R}$ as: $u(T) = v(T|S)$

- Affine Transformations: given $v : 2^{[n]} \to \mathbb{R}$, $p \in \mathbb{R}^n$ and $v_0 \in \mathbb{R}$ we can define $u(S) = v_0 + p(S) + v(S)$

- Tree-Concordant Sum: given u, v in GS then $u + v$ is in GS iff the Hessians of u, v share the same tree structure. [Balkanski-PL]

Operations Preserving GS

- Induction by networks: given $u : 2^{[m]} \to \mathbb{R}$ in GS and a bipartite graph $([n], [m], E)$ with weights w_e on the edges, define $v : 2^{[n]} \to \mathbb{R}$ such that:

$$v(S) = \max_{\text{matching} M \subseteq E, \partial_V(M) \subseteq S} [u(\partial_U(M)) + \sum_{e \in M} w_e]$$

Operations Preserving GS

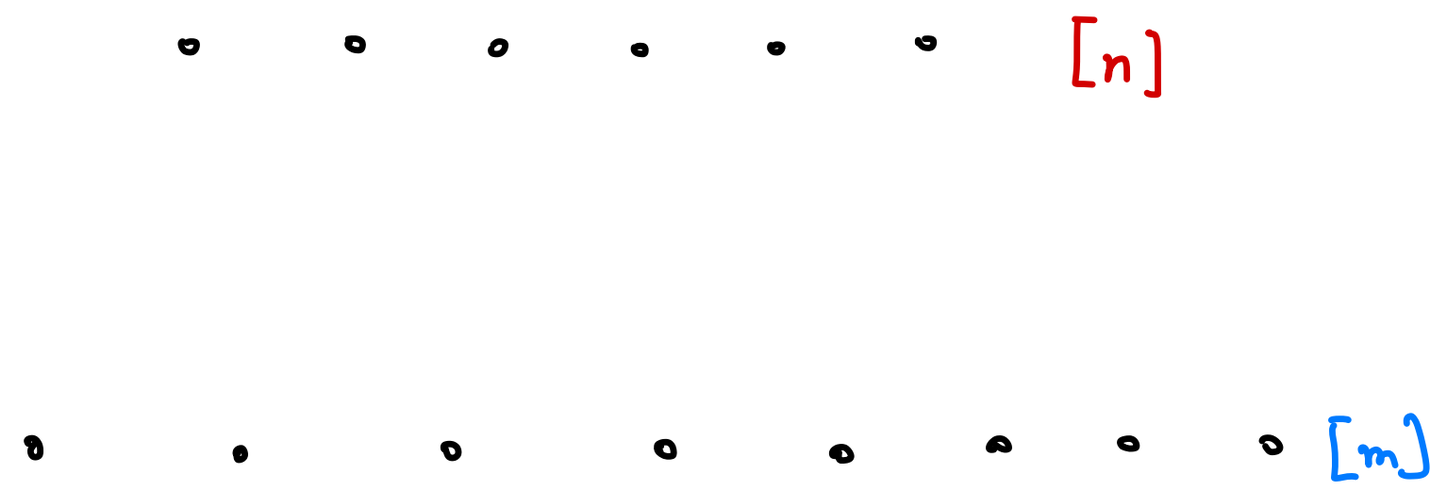
- Induction by networks: given $u : 2^{[m]} \to \mathbb{R}$ in GS and a bipartite graph $([n], [m], E)$ with weights w_e on the edges, define $v : 2^{[n]} \to \mathbb{R}$ such that:

$$v(S) = \max_{\text{matching} M \subseteq E, \partial_V(M) \subseteq S} [u(\partial_U(M)) + \sum_{e \in M} w_e]$$

Operations Preserving GS

- Induction by networks: given $u : 2^{[m]} \to \mathbb{R}$ in GS and a bipartite graph $([n], [m], E)$ with weights w_e on the edges, define $v : 2^{[n]} \to \mathbb{R}$ such that:

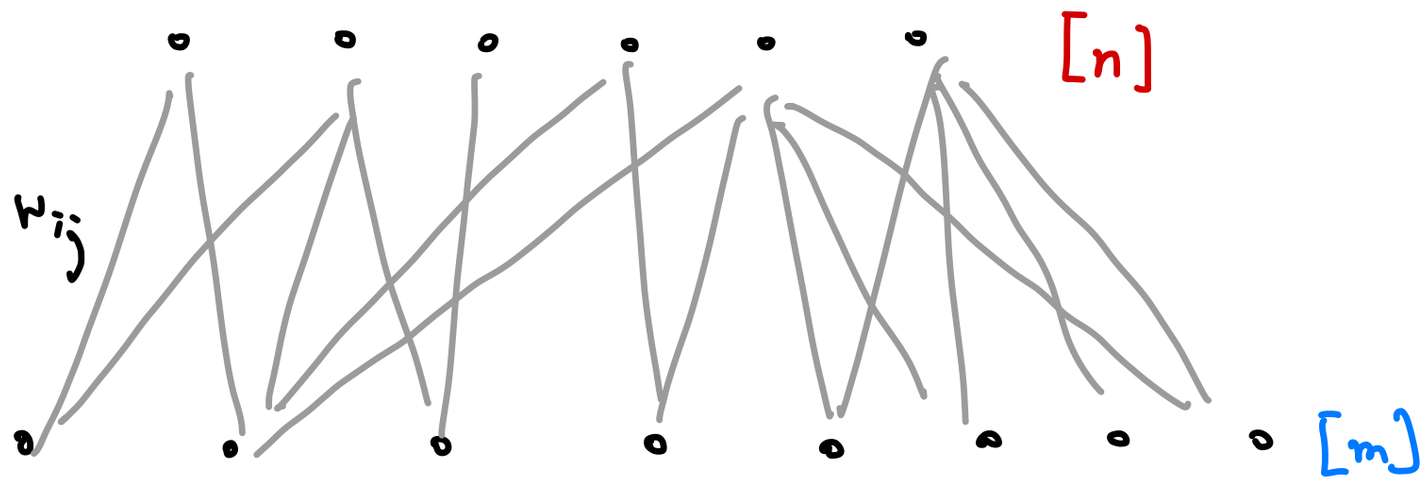
$$v(S) = \max_{\text{matching} M \subseteq E, \partial_V(M) \subseteq S} [u(\partial_U(M)) + \sum_{e \in M} w_e]$$



Operations Preserving GS

- Induction by networks: given $u : 2^{[m]} \to \mathbb{R}$ in GS and a bipartite graph $([n], [m], E)$ with weights w_e on the edges, define $v : 2^{[n]} \to \mathbb{R}$ such that:

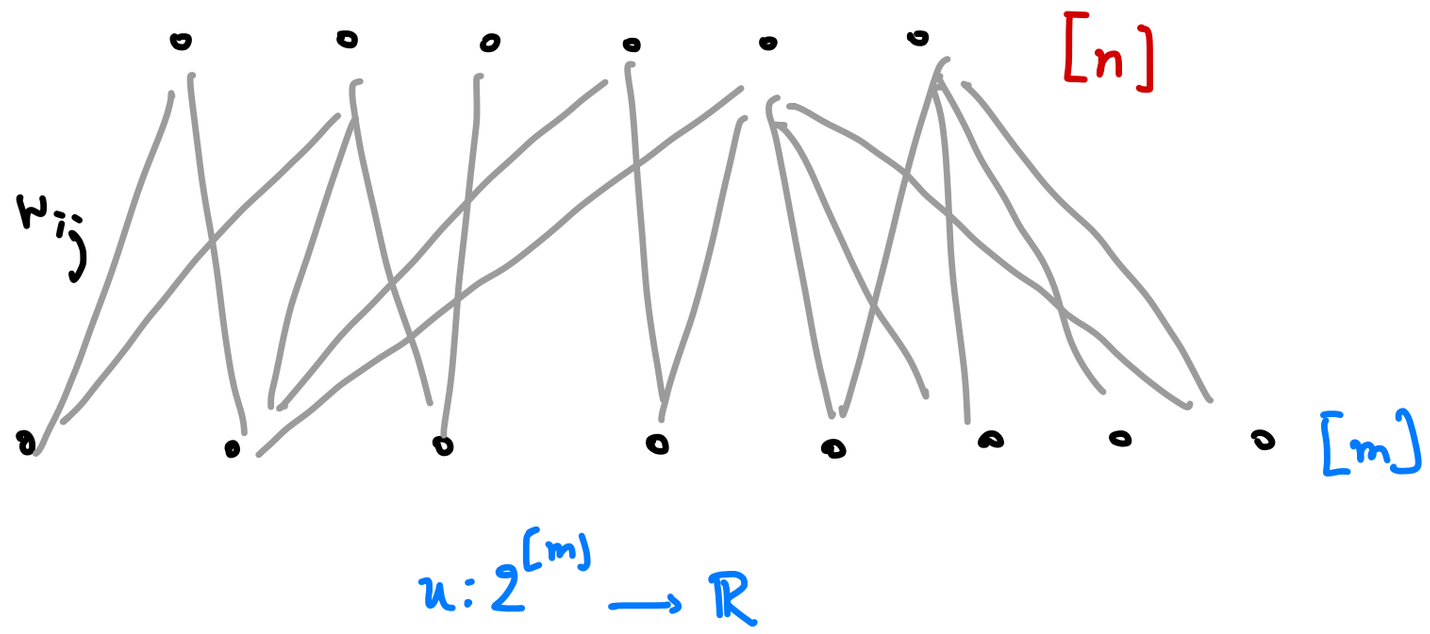
$$v(S) = \max_{\text{matching} M \subseteq E, \partial_V(M) \subseteq S} [u(\partial_U(M)) + \sum_{e \in M} w_e]$$



Operations Preserving GS

- Induction by networks: given $u : 2^{[m]} \to \mathbb{R}$ in GS and a bipartite graph $([n], [m], E)$ with weights w_e on the edges, define $v : 2^{[n]} \to \mathbb{R}$ such that:

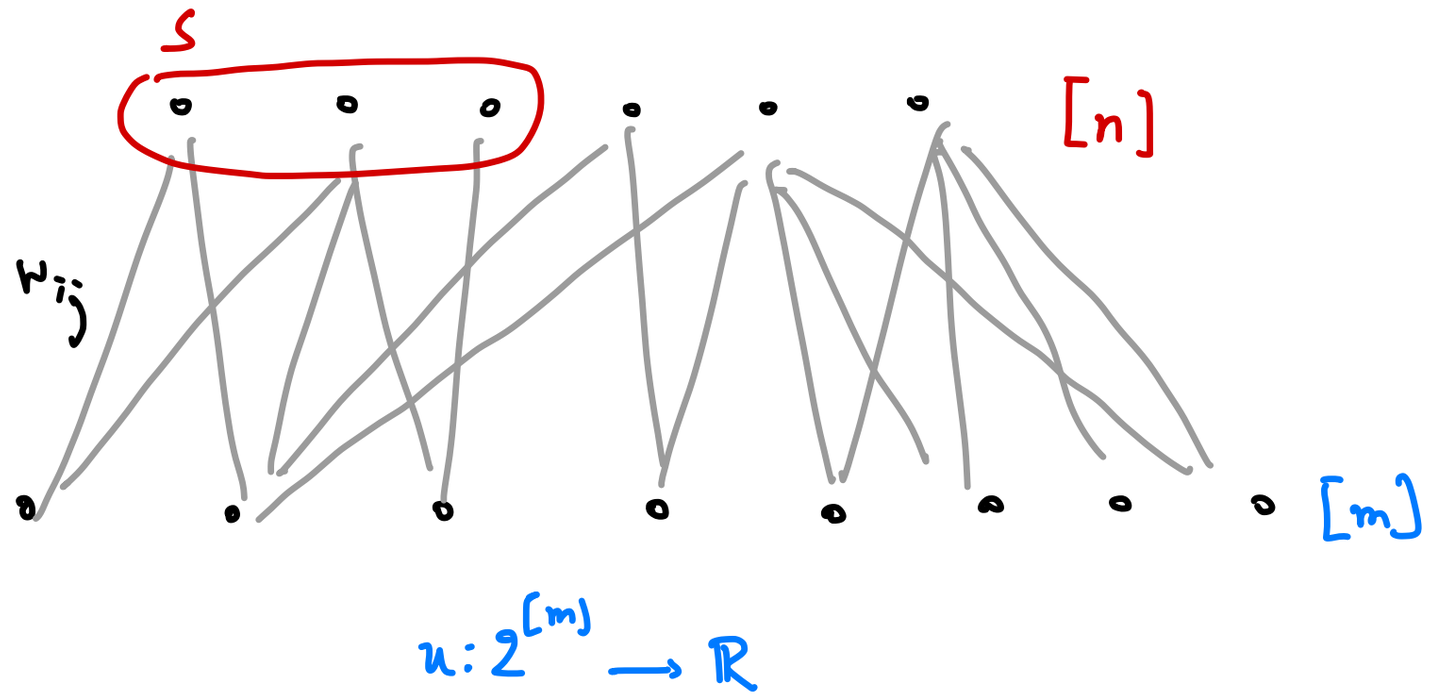
$$v(S) = \max_{\text{matching} M \subseteq E, \partial_V(M) \subseteq S} [u(\partial_U(M)) + \sum_{e \in M} w_e]$$



Operations Preserving GS

- Induction by networks: given $u : 2^{[m]} \to \mathbb{R}$ in GS and a bipartite graph $([n], [m], E)$ with weights w_e on the edges, define $v : 2^{[n]} \to \mathbb{R}$ such that:

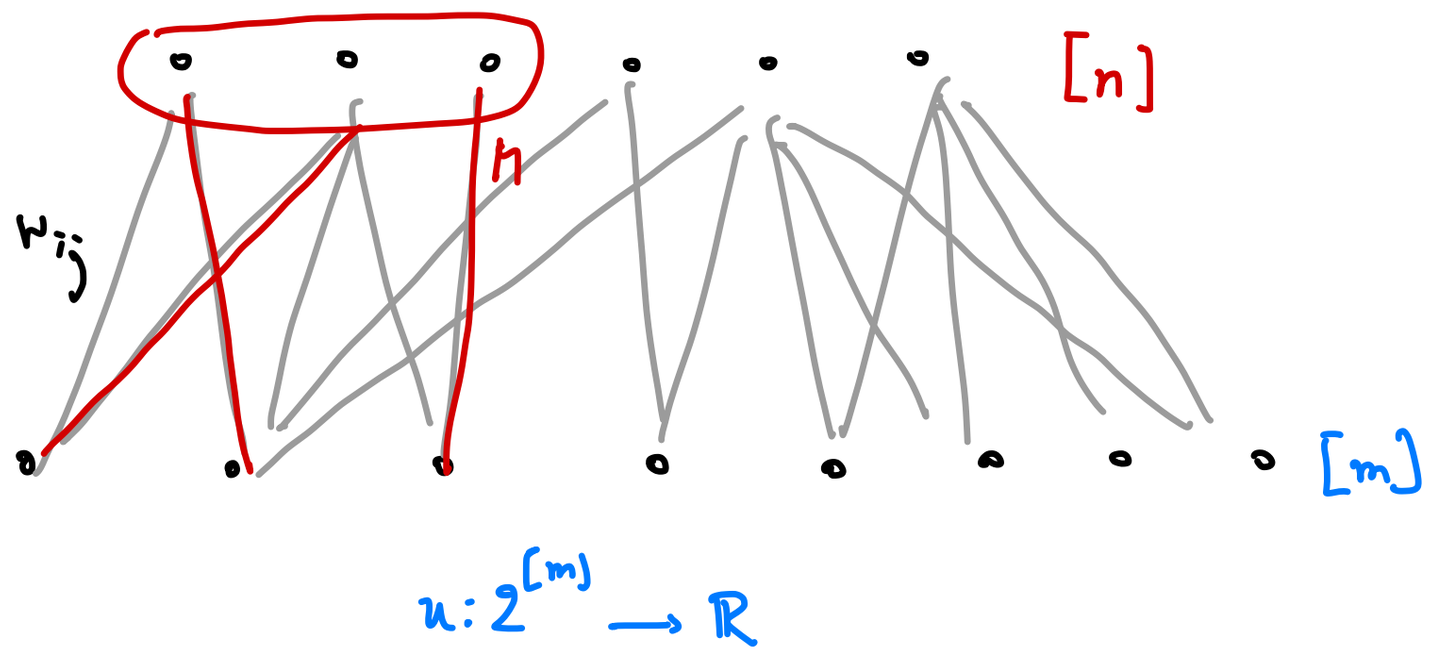
$$v(S) = \max_{\text{matching} M \subseteq E, \partial_V(M) \subseteq S} [u(\partial_U(M)) + \sum_{e \in M} w_e]$$



Operations Preserving GS

- Induction by networks: given $u : 2^{[m]} \to \mathbb{R}$ in GS and a bipartite graph $([n], [m], E)$ with weights w_e on the edges, define $v : 2^{[n]} \to \mathbb{R}$ such that:

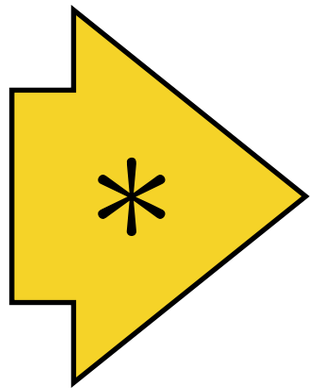
$$v(S) = \max_{\text{matching} M \subseteq E, \partial_V(M) \subseteq S} [u(\partial_U(M)) + \sum_{e \in M} w_e]$$



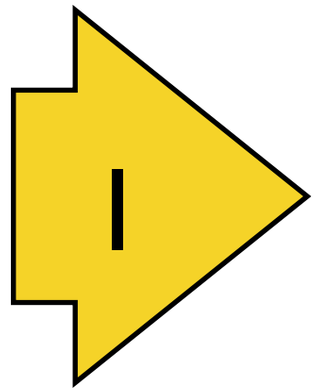
What should be the building block?

Hatfield-Milgrom observed that many interesting examples can be built starting from unit demand:

Unit Demand Valuations



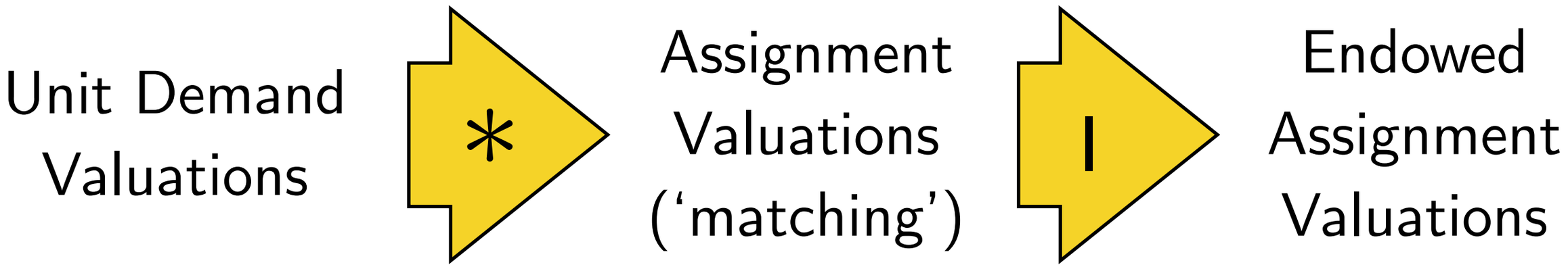
Assignment Valuations ('matching')



Endowed Assignment Valuations

What should be the building block?

Hatfield-Milgrom observed that many interesting examples can be built starting from unit demand:



[Ostrovsky, PL]: some matroid rank functions are not endowed assignment valuations.

The natural place to start is with matroids.

Matroids are the natural building blocks

Consider functions $v : 2^{[n]} \to \mathbb{R} \cap \{-\infty\}$ and consider two special cases:

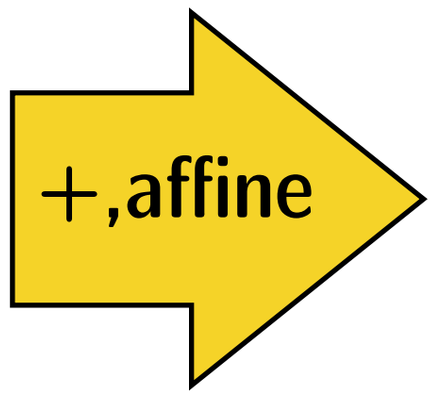
- If v is GS and $v(S) \in \{0, -\infty\}$ then v_v is the indicator function of a matroid: $v(S) = 0, S \in \mathcal{M}$ and $-\infty$ o.w.
- If v is GS with $v(S) = 0$ and $\partial_i v(S) \in \{0, 1\}$ then v is a matroid rank function: $v(S) = \max_{U \in \mathcal{M}} |S \cap U|$

Many connections to matroids established in Gul-Stachetti, Murota-Shioura, Baldwin-Klemperer,...

Potential Constructions

Potential Constructions

Matroids
Rank
Functions

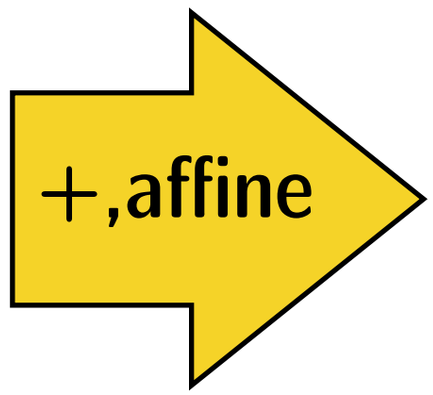


Weighted
Combinations
MRF

Potential Constructions

Matroids
Rank
Functions

Shioura

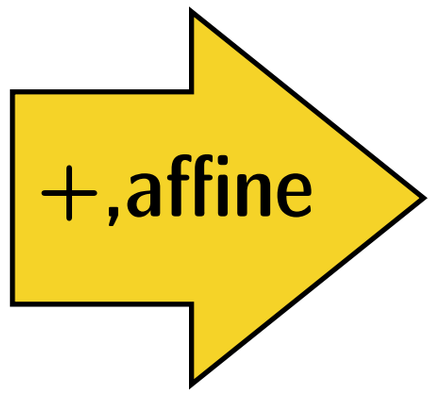


Weighted
Combinations
MRF

Potential Constructions

Matroids
Rank
Functions

Shioura

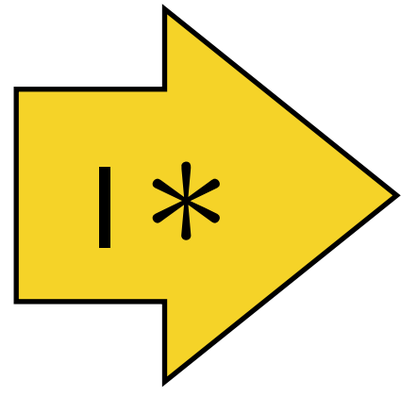


Weighted
Combinations
MRF

Yes, for n=3,4
No for n>4
[Balkanski, PL]

Potential Constructions

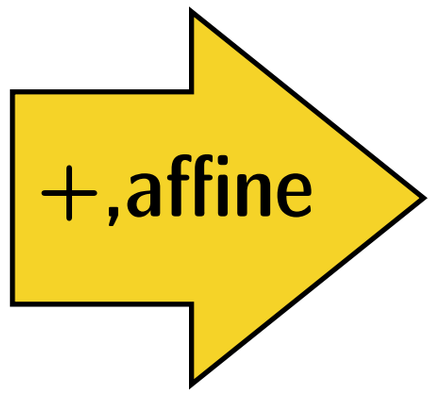
Matroid Rank Functions



Matroid Based Valuations

Matroids Rank Functions

Shioura



Weighted Combinations MRF

Yes, for n=3,4

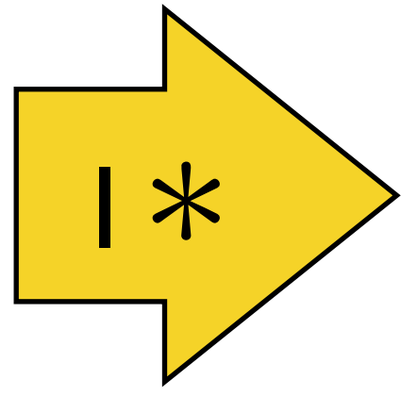
No for n>4

[Balkanski, PL]

Potential Constructions

Matroid Rank Functions

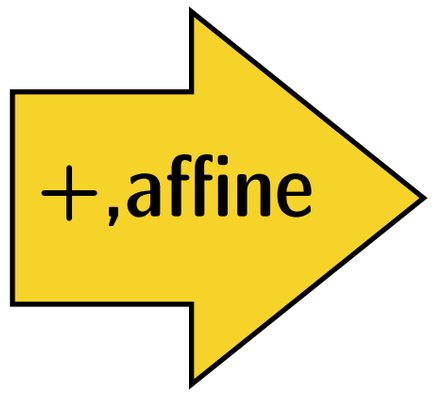
PL-Ostrovsky



Matroid Based Valuations

Matroids Rank Functions

Shioura



Weighted Combinations MRF

Yes, for n=3,4

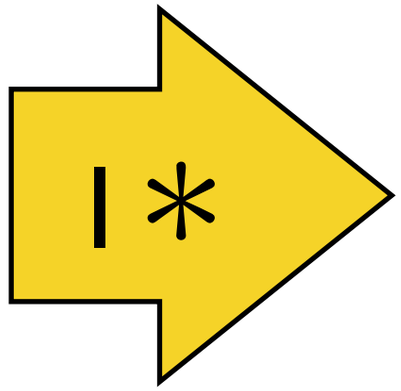
No for n>4

[Balkanski, PL]

Potential Constructions

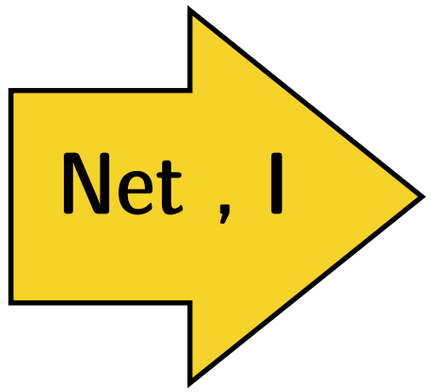
Matroid Rank Functions

PL-Ostrovsky



Matroid Based Valuations

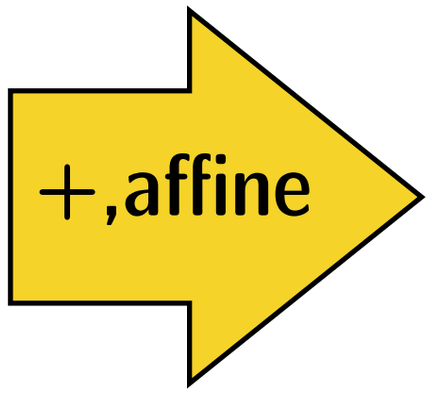
Matroid Rank Functions



Rado Minor

Matroids Rank Functions

Shioura



Weighted Combinations MRF

Yes, for n=3,4

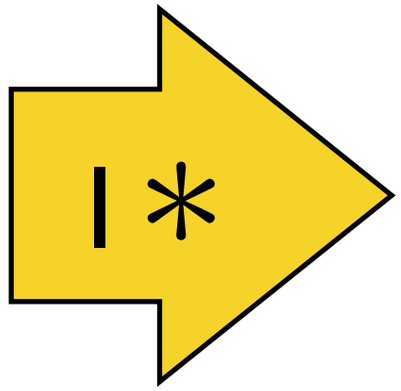
No for n>4

[Balkanski, PL]

Potential Constructions

Matroid Rank Functions

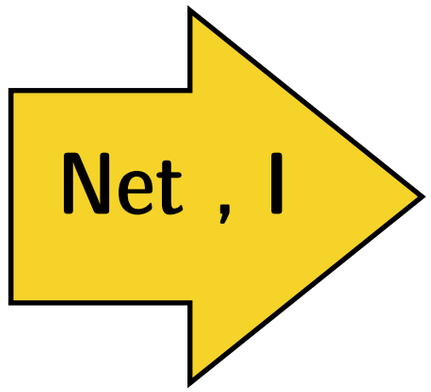
PL-Ostrovsky



Matroid Based Valuations

Matroid Rank Functions

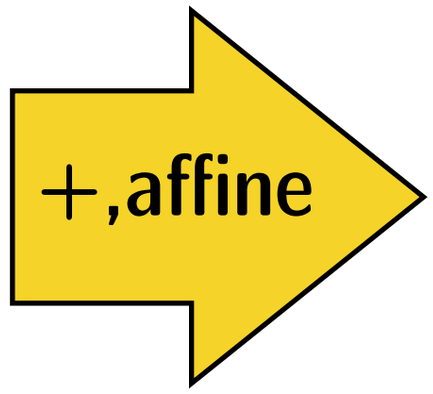
Murota



Rado Minor

Matroids Rank Functions

Shioura



Weighted Combinations MRF

Yes, for n=3,4

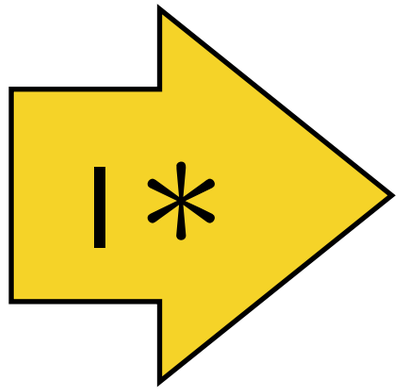
No for n>4

[Balkanski, PL]

Potential Constructions

Matroid Rank Functions

PL-Ostrovsky

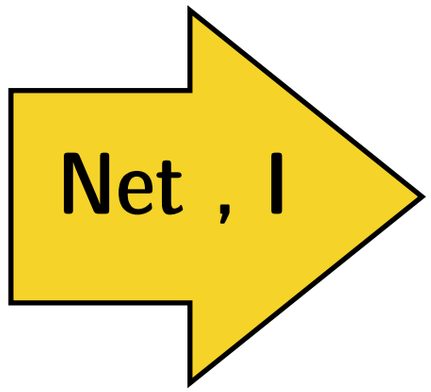


Matroid Based Valuations

No

Matroid Rank Functions

Murota

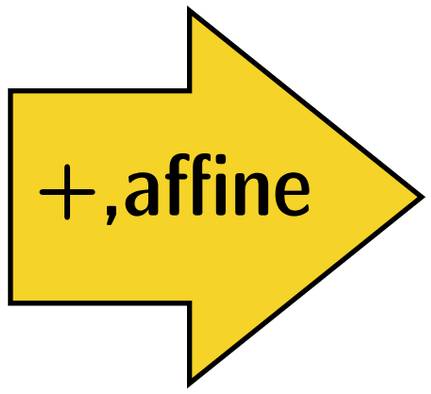


Rado Minor

No

Matroids Rank Functions

Shioura



Weighted Combinations MRF

Yes, for n=3,4

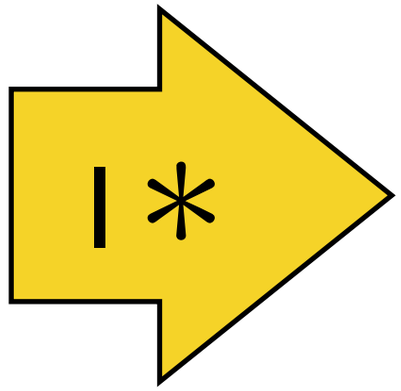
No for n>4

[Balkanski, PL]

Potential Constructions

Matroid Rank Functions

PL-Ostrovsky



Matroid Based Valuations

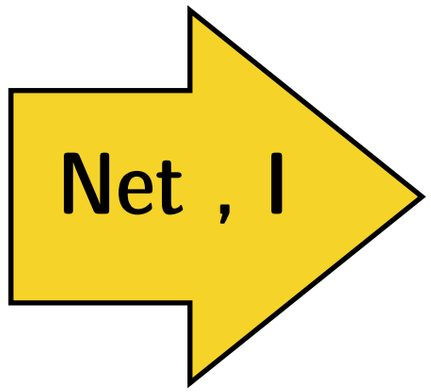
No

On complete classes of valuated matroids

[Husic, Loho, Smith, Vegh, 2021]

Matroid Rank Functions

Murota

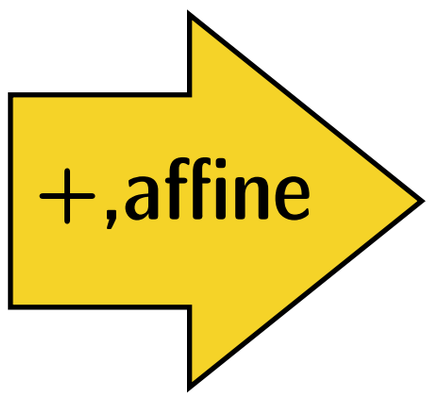


Rado Minor

No

Matroids Rank Functions

Shioura



Weighted Combinations MRF

Yes, for n=3,4

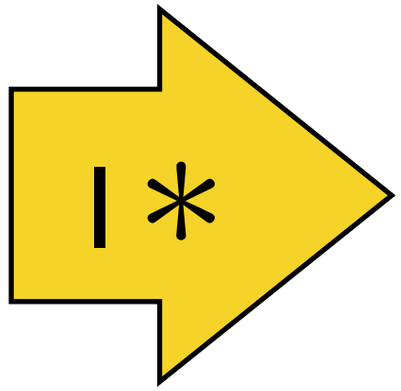
No for n>4

[Balkanski, PL]

Potential Constructions

Matroid Rank Functions

PL-Ostrovsky



Matroid Based Valuations

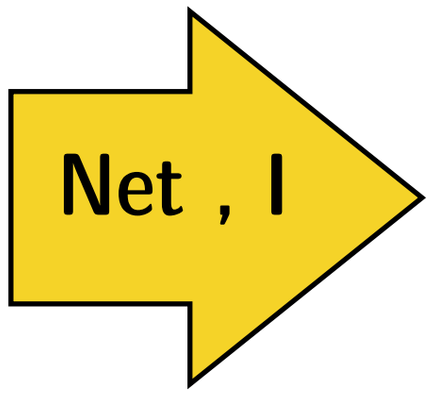
No

On complete classes of valuated matroids

[Husic, Loho, Smith, Vegh, 2021]

Matroid Rank Functions

Murota



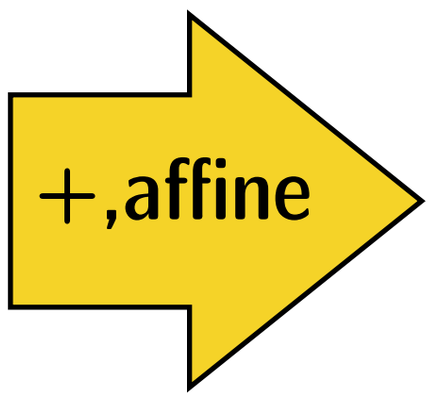
$\neq$?

Rado Minor

No

Matroids Rank Functions

Shioura



Weighted Combinations MRF

Yes, for n=3,4

No for n>4

[Balkanski, PL]

Sparse Paving Matroids

Let $\binom{[n]}{k}$ be all subset of [n] with k elements. Let $I \subseteq \binom{[n]}{k}$ such that $X, Y \in I, |X \cap Y| \leq k-2$. Now, define a matroid $\mathcal{M}$ where a set S is independent iff $S \subseteq B \in \binom{[n]}{k} \setminus I$.

Construction in Husic et al: take nested SP-matroids

$$\mathcal{M}_1 \subseteq \mathcal{M}_2 \subseteq \ldots \subseteq \mathcal{M}_m$$

and define:

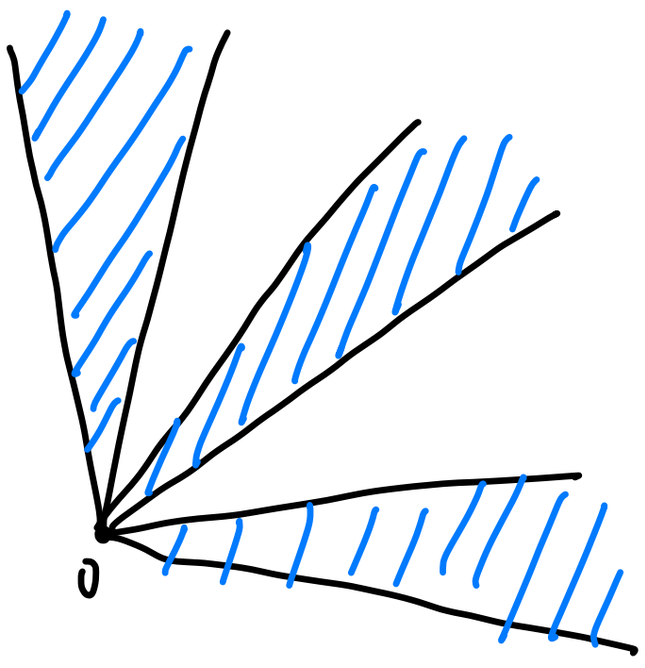
$$v(S) = \sum_{i=1}^{m} w_i \cdot \mathrm{rank}_{\mathcal{M}_i}(S)$$

Show it can't be constructed from N, |, *, ...

However, they are still a weighted combination of matroids.

Operations Preserving GS

We can view **GS** as a (non-convex) cone in $\mathbb{R}^{2^n}$:



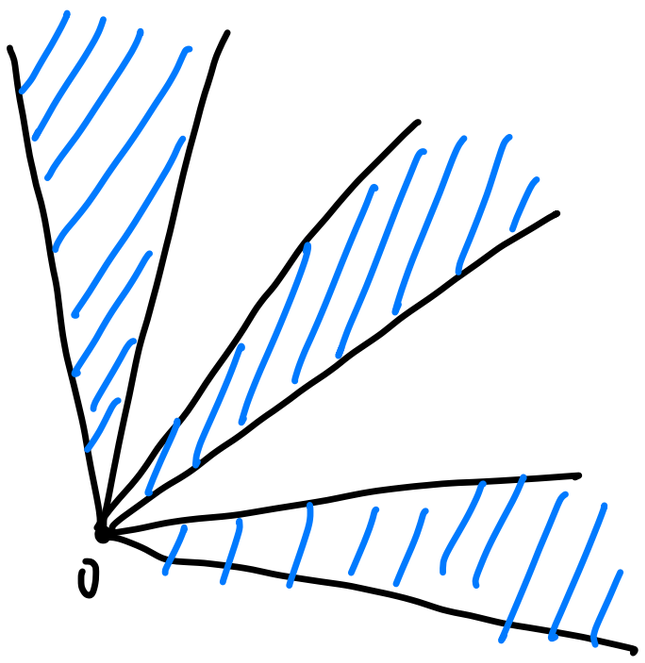
New version: can extremal GS be constructed from Matroid Rank Functions by Convolutions, endowments and induction by networks?

are extremal GS in Rado-minor?

Operations Preserving GS

We can view **GS** as a (non-convex) cone in $\mathbb{R}^{2^n}$:

union of finitely-many convex cones



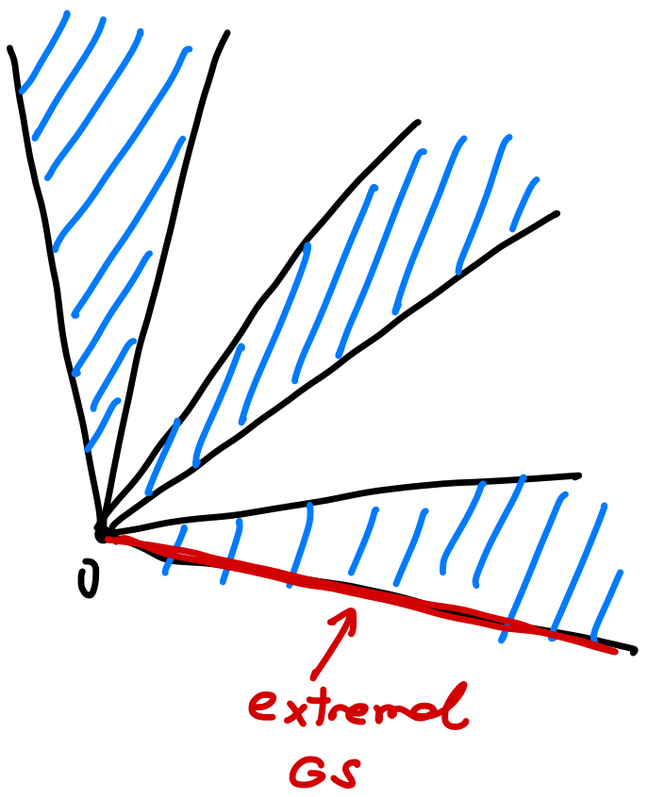
New version: can extremal GS be constructed from Matroid Rank Functions by Convolutions, endowments and induction by networks?

are extremal GS in Rado-minor?

Operations Preserving GS

We can view **GS** as a (non-convex) cone in $\mathbb{R}^{2^n}$:

union of finitely-many convex cones



New version: can extremal GS be constructed from Matroid Rank Functions by Convolutions, endowments and induction by networks?

are extremal GS in Rado-minor?

Part II: Approximating GS

Valuation Function Approximation

From now one, we will assume that valuations $v : 2^{[n]} \to \mathbb{R}$ are normalized $v(\emptyset) = 0$ and monotone.

A class of functions $\mathbf{C}$ can be α-approximated by class $\mathbf{B}$ if for every function $f \in \mathbf{C}$ there exists $g \in \mathbf{B}$ such that:

$$g(S) \leq f(S) \leq \alpha \cdot g(S), \forall S \subseteq [n]$$

Valuation Function Approximation

From now one, we will assume that valuations $v : 2^{[n]} \to \mathbb{R}$ are normalized $v(\emptyset) = 0$ and monotone.

A class of functions $\mathbf{C}$ can be α-approximated by class $\mathbf{B}$ if for every function $f \in \mathbf{C}$ there exists $g \in \mathbf{B}$ such that:

$$g(S) \leq f(S) \leq \alpha \cdot g(S), \forall S \subseteq [n]$$

Example:

Submodular $\subsetneq$ XOS $\subsetneq$ Subadditive

Valuation Function Approximation

From now one, we will assume that valuations $v : 2^{[n]} \to \mathbb{R}$ are normalized $v(\emptyset) = 0$ and monotone.

A class of functions $\mathbf{C}$ can be α-approximated by class $\mathbf{B}$ if for every function $f \in \mathbf{C}$ there exists $g \in \mathbf{B}$ such that:

$$g(S) \leq f(S) \leq \alpha \cdot g(S), \forall S \subseteq [n]$$

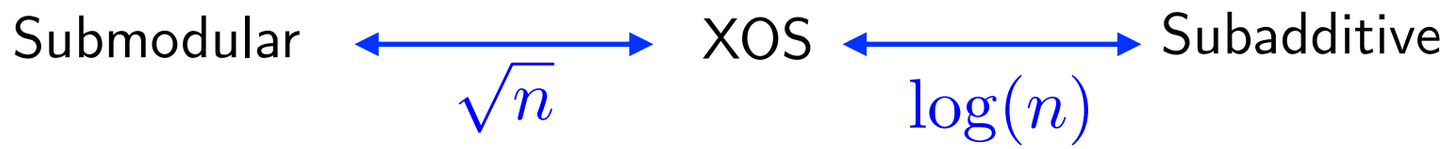
Example:

Why? Another way to measure relative complexity of valuation classes.

Also: small approx means algorithms for one class may work for the other.

Valuation Function Approximation

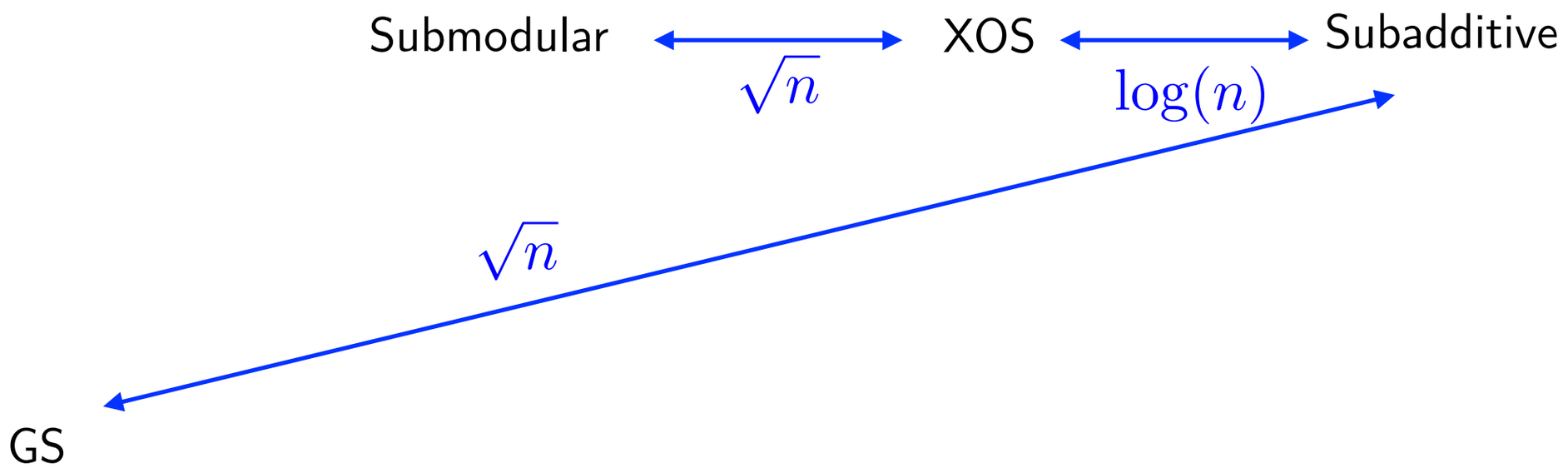
How does GS compare with other classes?



GS

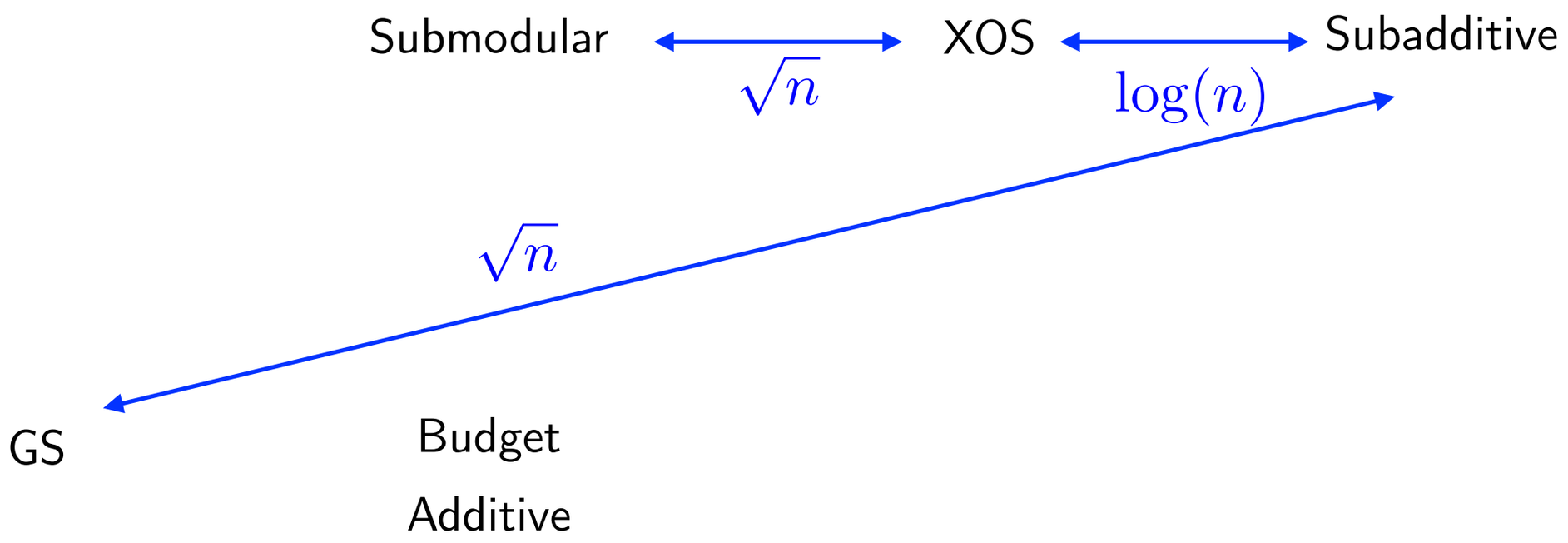
Valuation Function Approximation

How does GS compare with other classes?



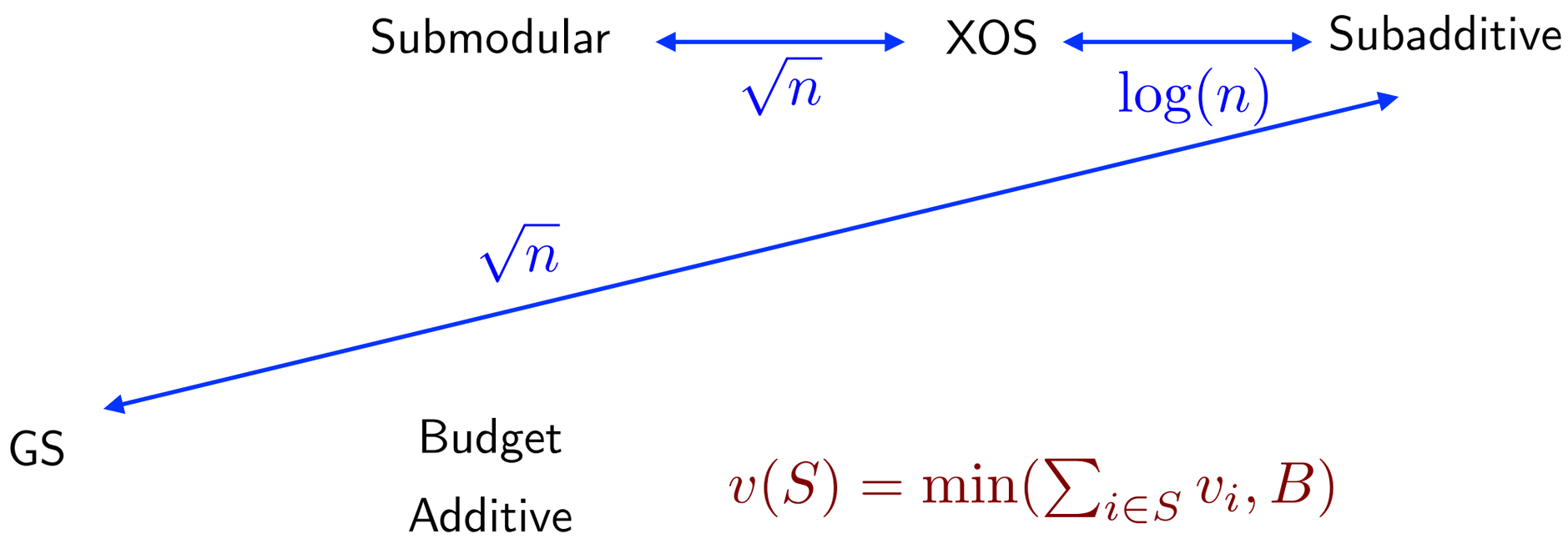
Valuation Function Approximation

How does GS compare with other classes?



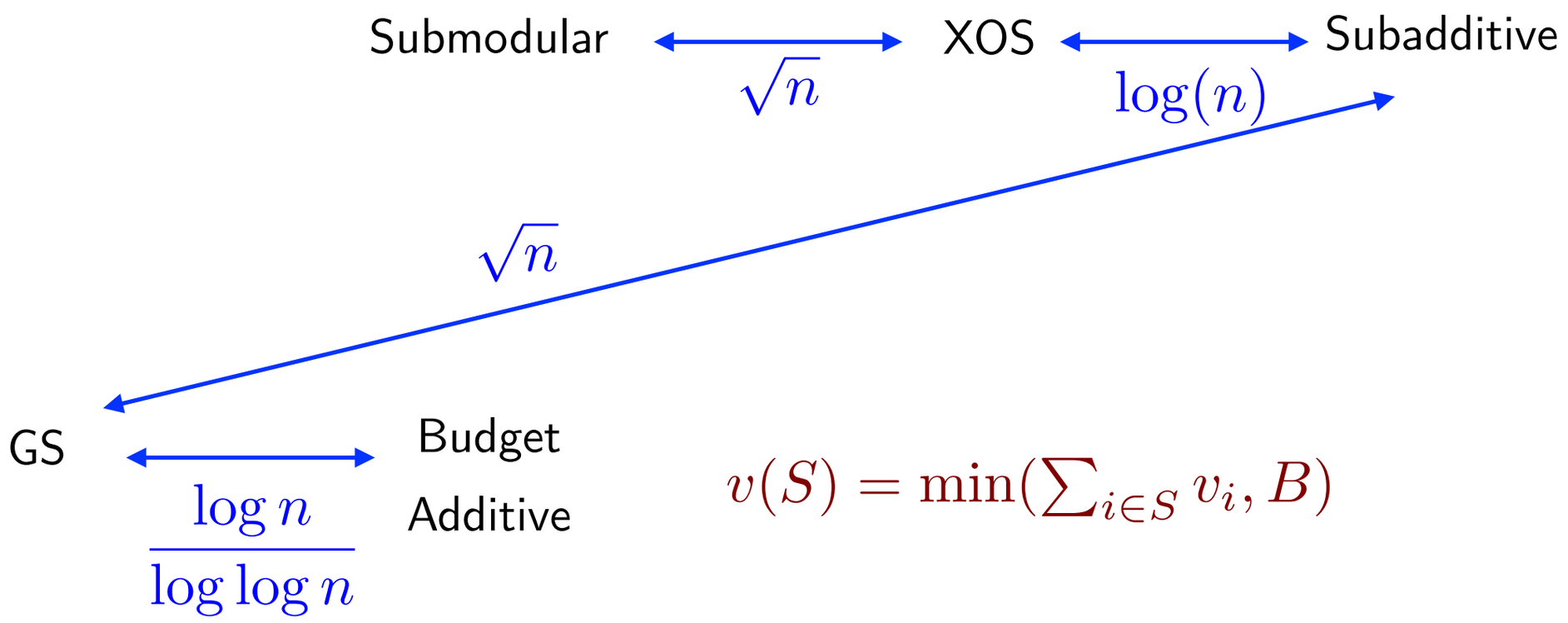
Valuation Function Approximation

How does GS compare with other classes?



Valuation Function Approximation

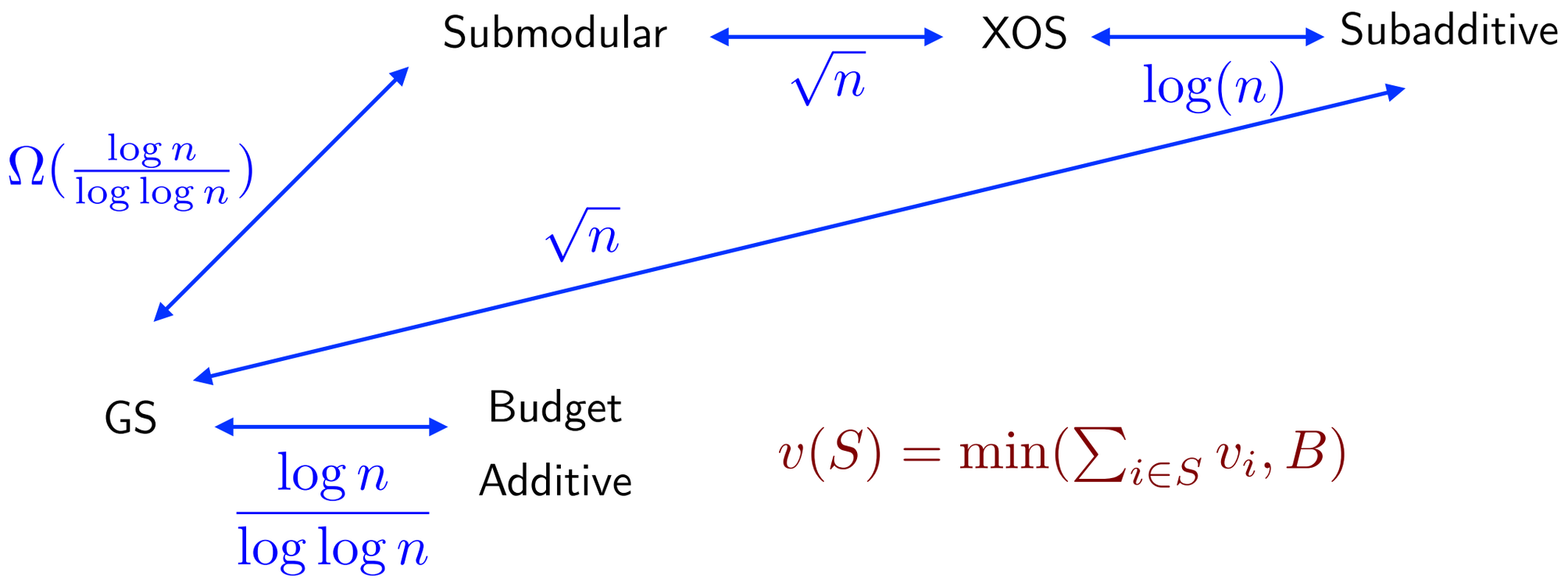
How does GS compare with other classes?



$$v(S) = \min(\textstyle\sum_{i \in S} v_i, B)$$

Valuation Function Approximation

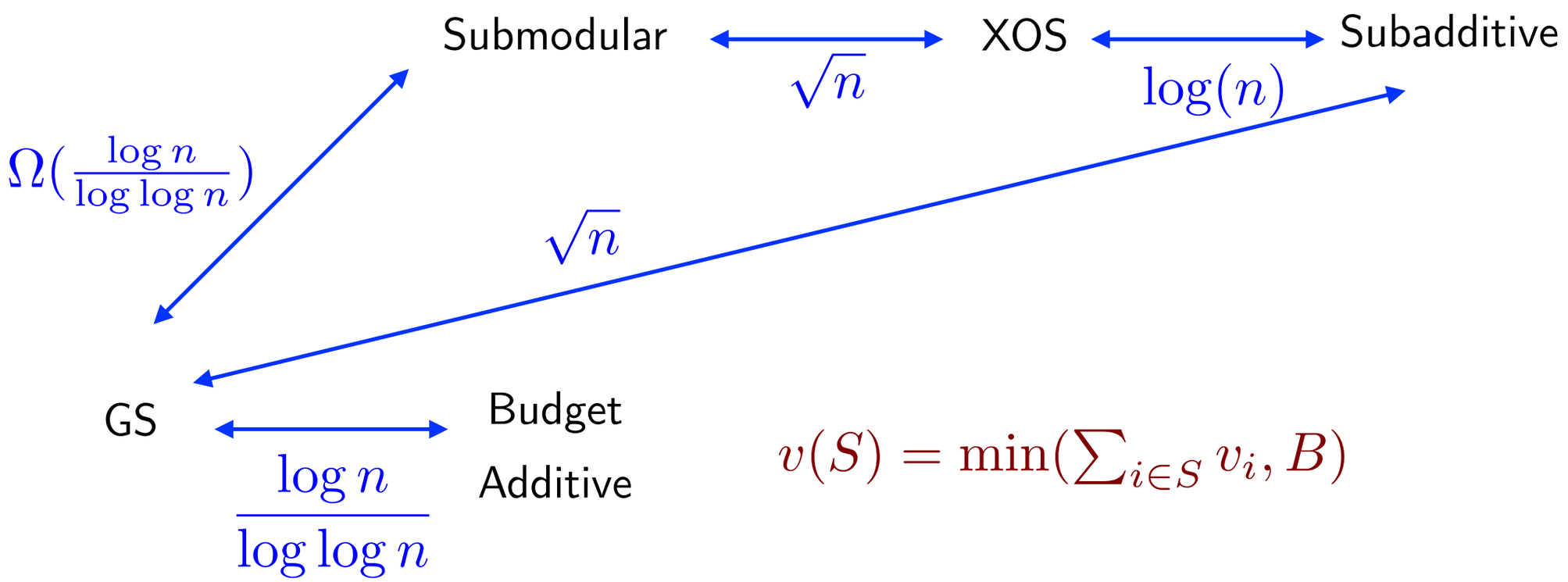
How does GS compare with other classes?



$v(S) = \min(\sum_{i \in S} v_i, B)$

Valuation Function Approximation

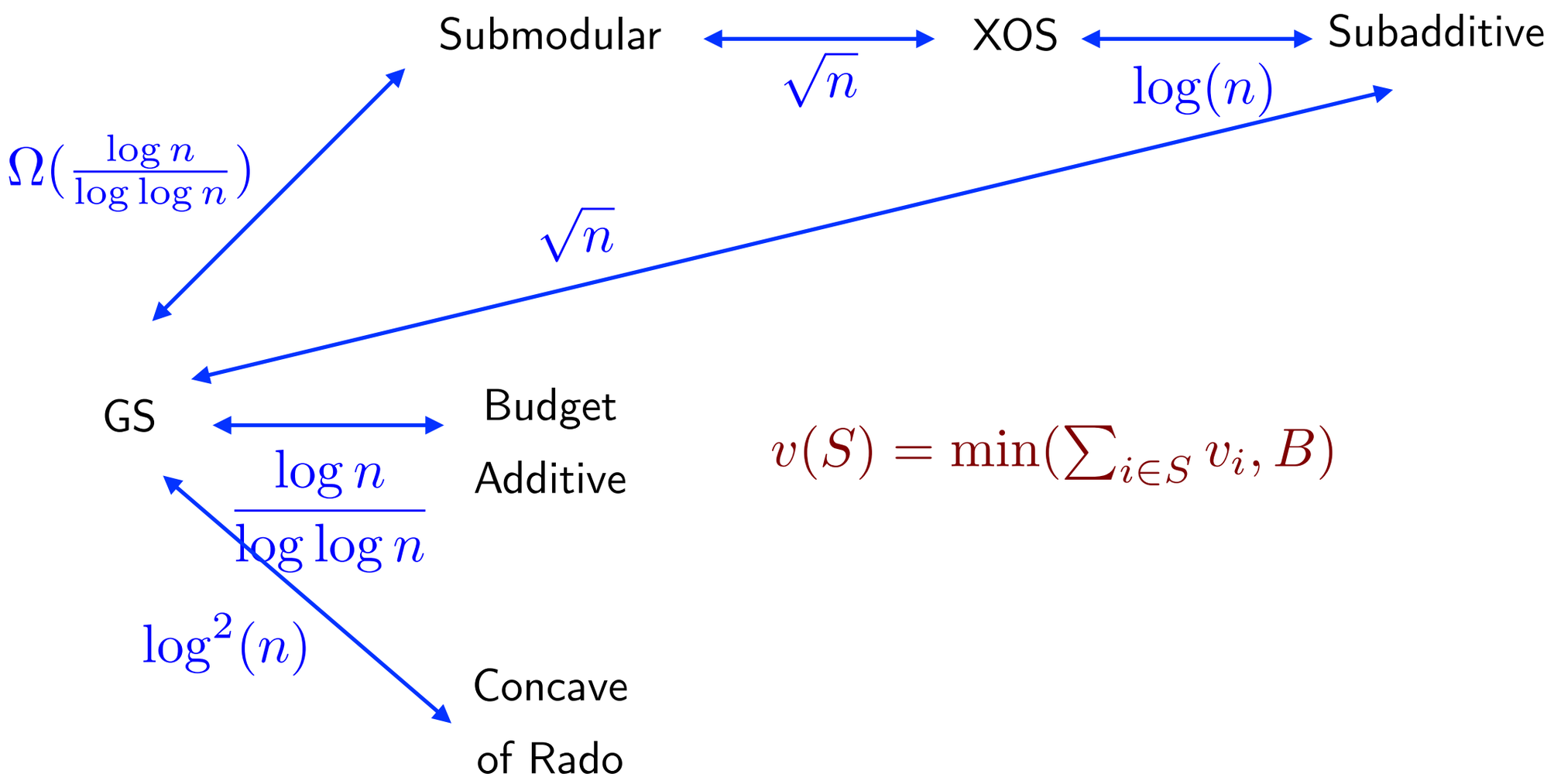
How does GS compare with other classes?



$v(S) = \min(\sum_{i \in S} v_i, B)$

Valuation Function Approximation

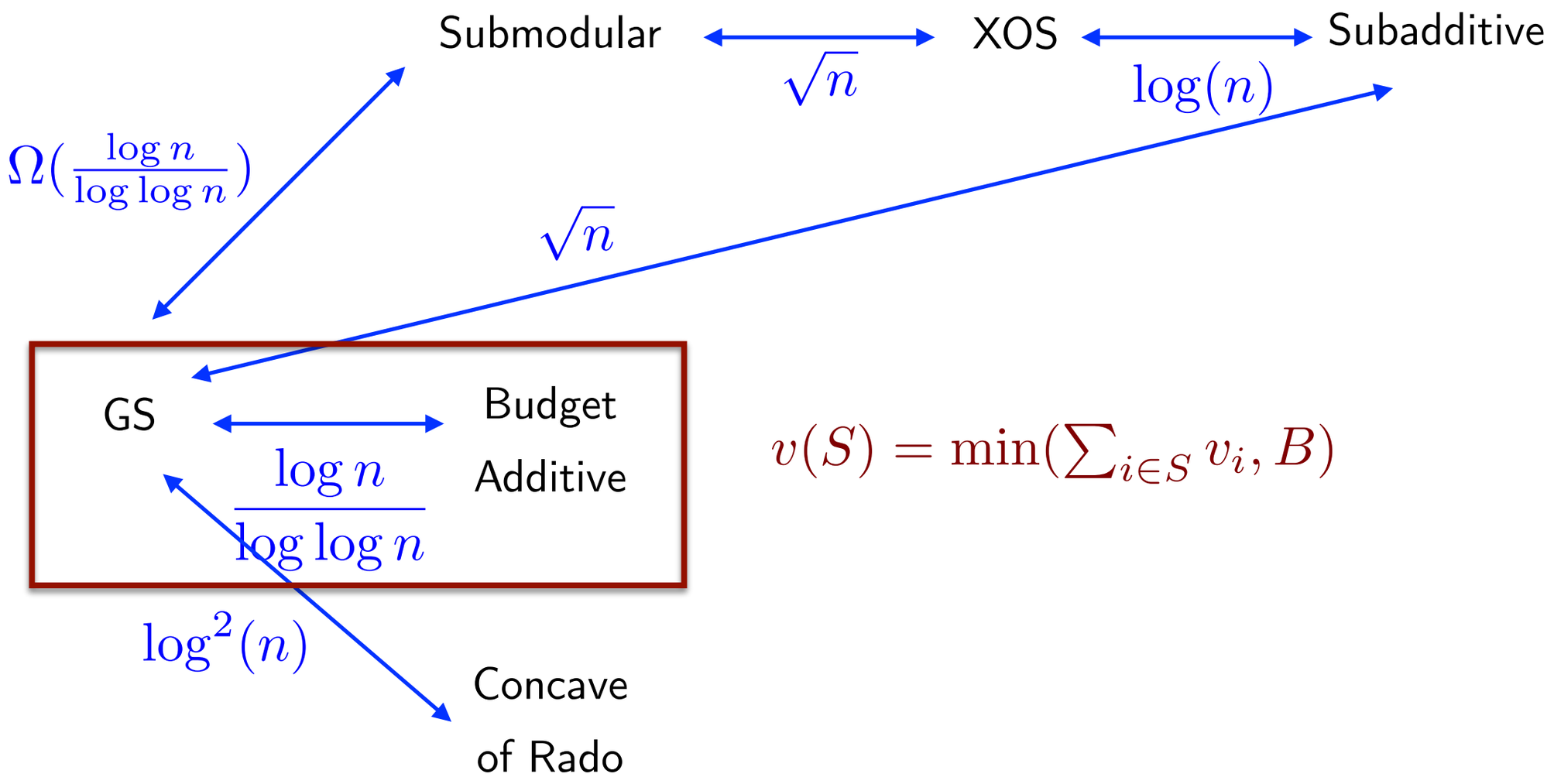
How does GS compare with other classes?



$v(S) = \min(\sum_{i \in S} v_i, B)$

Valuation Function Approximation

How does GS compare with other classes?



$$v(S) = \min(\textstyle\sum_{i \in S} v_i, B)$$

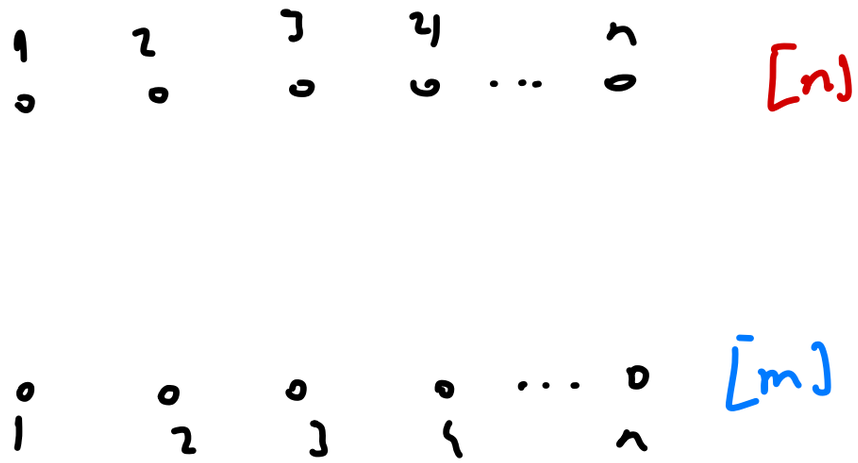
Valuation Function Approximation

For every budget additive function $v(S) = \min(\sum_{i \in S} v_i, B)$ there is a GS function f such that: $f(S) \leq v(S) \leq f(S) \cdot c\frac{\log n}{\log \log n}$. Moreover, this is the best possible approximation ratio.

Valuation Function Approximation

For every budget additive function $v(S) = \min(\sum_{i \in S} v_i, B)$ there is a GS function f such that: $f(S) \leq v(S) \leq f(S) \cdot c\frac{\log n}{\log \log n}$. Moreover, this is the best possible approximation ratio.

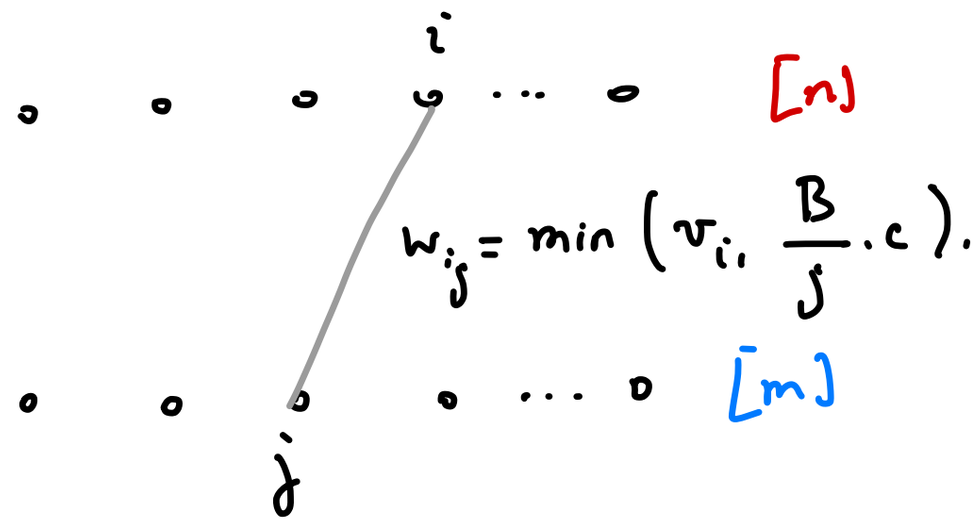
Upper bound: approximate v by an assignment valuation (OXS).



Valuation Function Approximation

For every budget additive function $v(S) = \min(\sum_{i \in S} v_i, B)$ there is a GS function f such that: $f(S) \leq v(S) \leq f(S) \cdot c\frac{\log n}{\log \log n}$. Moreover, this is the best possible approximation ratio.

Upper bound: approximate v by an assignment valuation (OXS).



Valuation Function Approximation

For every budget additive function $v(S) = \min(\sum_{i \in S} v_i, B)$ there is a GS function f such that: $f(S) \leq v(S) \leq f(S) \cdot c\frac{\log n}{\log \log n}$. Moreover, this is the best possible approximation ratio.

Lower bound via symmetrization: given an approximation $f(S) \leq v(S) \leq \alpha f(S)$ if v has a symmetry and f is in GS, we can symmetrize f such that the approximation continue to hold.

Valuation Function Approximation

For every budget additive function $v(S) = \min(\sum_{i \in S} v_i, B)$ there is a GS function f such that: $f(S) \leq v(S) \leq f(S) \cdot c\frac{\log n}{\log \log n}$. Moreover, this is the best possible approximation ratio.

Lower bound via symmetrization: given an approximation $f(S) \leq v(S) \leq \alpha f(S)$ if v has a symmetry and f is in GS, we can symmetrize f such that the approximation continue to hold.

e.g. $v(S \cup \{i\}) = v(S \cup \{j\}), \forall S \setminus \{i, j\}$

Valuation Function Approximation

For every budget additive function $v(S) = \min(\sum_{i \in S} v_i, B)$ there is a GS function f such that: $f(S) \leq v(S) \leq f(S) \cdot c\frac{\log n}{\log \log n}$. Moreover, this is the best possible approximation ratio.

Lower bound via symmetrization: given an approximation $f(S) \leq v(S) \leq \alpha f(S)$ if v has a symmetry and f is in GS, we can symmetrize f such that the approximation continue to hold and the function is still GS.

Max-symmetrization: given f in GS and two items i and j, define the symmetrized version f_{sym} such that for all $S \subseteq [n] \setminus \{i, j\}$:

$$f_{\text{sym}}(S) = f(S) \qquad f_{\text{sym}}(S \cup \{i, j\}) = f(S \cup \{i, j\})$$

$$f_{\text{sym}}(S \cup \{i\}) = f_{\text{sym}}(S \cup \{j\}) = \max[f(S \cup \{i\}), f(S \cup \{j\})]$$

Valuation Function Approximation

Open question: can we approximate any submodular function by a GS function with ratio polylog(n) ?

Conclusion

Two different ways to estimate the complexity of GS functions with respect to other classes:

* Can GS be constructed from simpler functions?
* Can GS approximate richer classes of valuations?

Many open questions in both areas.

Appendix

Discrete Differential Equations

- Given a function $v : 2^{[n]} \to \mathbb{R}$ we define the discrete derivative with respect to $i \in [n]$ as the function $\partial_i v : 2^{[n]\setminus i} \to \mathbb{R}$ which is given by:

$$\partial_i v(S) = v(S \cup i) - v(S)$$

(another name for the marginal)

Discrete Differential Equations

- Given a function $v : 2^{[n]} \to \mathbb{R}$ we define the discrete derivative with respect to $i \in [n]$ as the function $\partial_i v : 2^{[n]\setminus i} \to \mathbb{R}$ which is given by:

$$\partial_i v(S) = v(S \cup i) - v(S)$$

(another name for the marginal)

- If we apply it twice we get:

$$\partial_{ij} v(S) := \partial_j \partial_i v(S) = v(S \cup ij) - v(S \cup i) - v(S \cup j) + v(S)$$

- Submodularity: $\partial_{ij} v(S) \leq 0$

Discrete Differential Equations

- [Reijnierse, Gellekom, Potters] A function $v : 2^{[n]} \to \mathbb{R}$ is in gross substitutes iff it satisfies:

$$\partial_{ij} v(S) \leq \max(\partial_{ik} v(S), \partial_{kj} v(S)) \leq 0$$

condition on the discrete Hessian.

$$H(S) = [\partial_{ij}(S)]_{i,j \notin S}$$

- Easier notation to work with positive numbers:

$$\Delta_{ij}^S = -\partial_{ij} v(S)$$

Then the condition becomes:

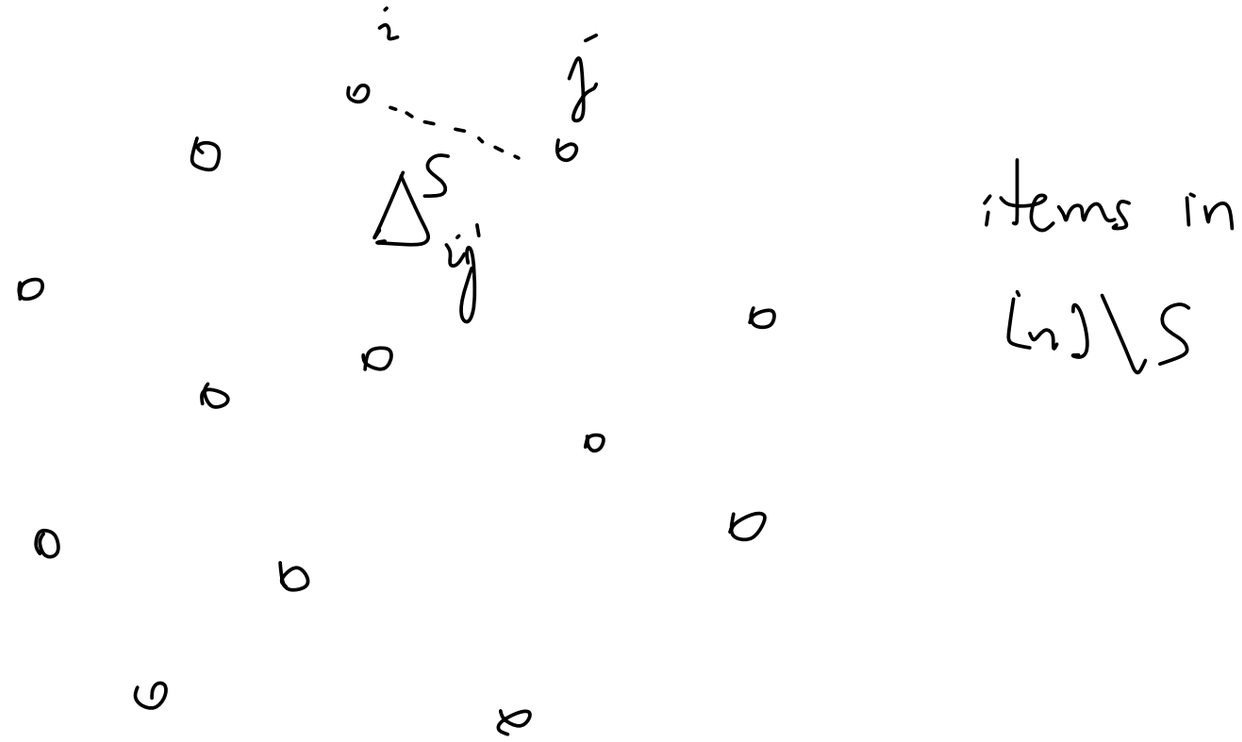
$$\Delta_{ij}^S \geq \min(\Delta_{ik}^S, \Delta_{kj}^S) \geq 0$$

Tree Structure

- [Hirai-Murota] Tree structure on the Hessian. Since:

$$\Delta^S_{ij} \geq \min(\Delta^S_{ik}, \Delta^S_{kj}) \geq 0$$

implies that in $\Delta^S_{ij}, \Delta^S_{ik}, \Delta^S_{kj}$ the minimum must appear twice. So, let $m = \min_{i,j \notin S} \Delta^S_{ij}$

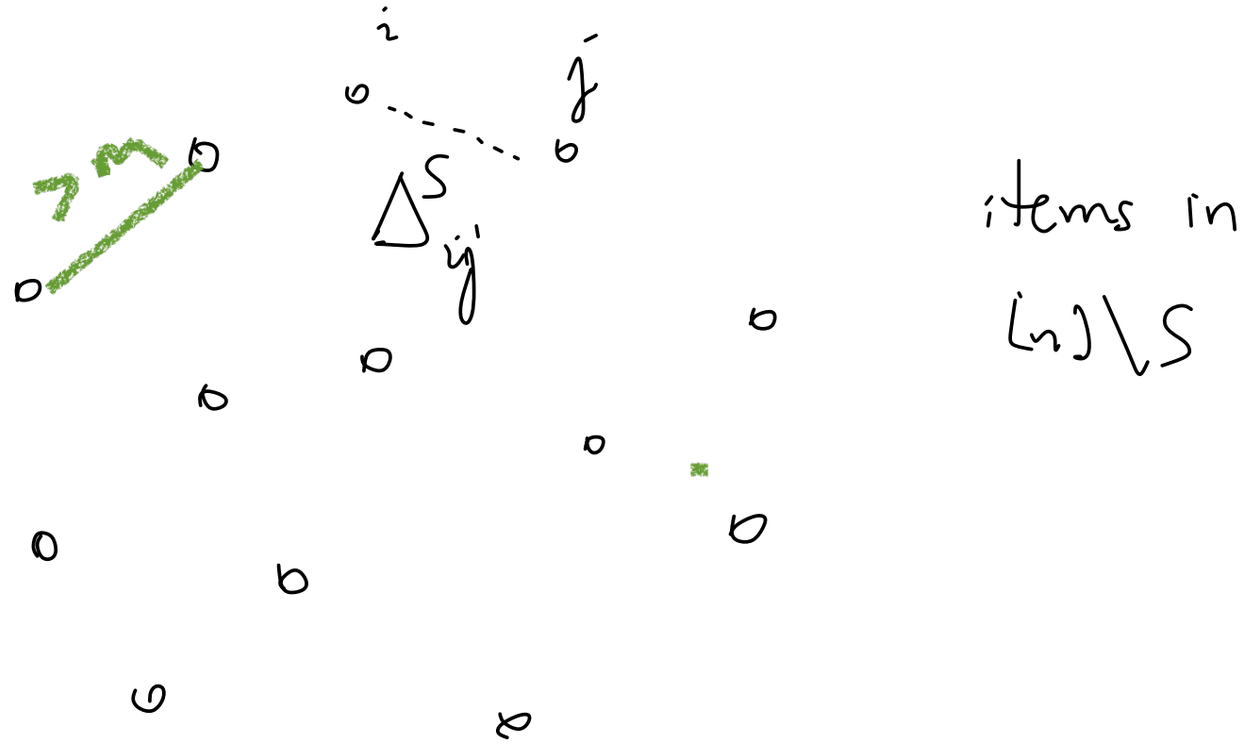


Tree Structure

- [Hirai-Murota] Tree structure on the Hessian. Since:

$$\Delta^S_{ij} \geq \min(\Delta^S_{ik}, \Delta^S_{kj}) \geq 0$$

implies that in $\Delta^S_{ij}, \Delta^S_{ik}, \Delta^S_{kj}$ the minimum must appear twice. So, let $m = \min_{i,j \notin S} \Delta^S_{ij}$

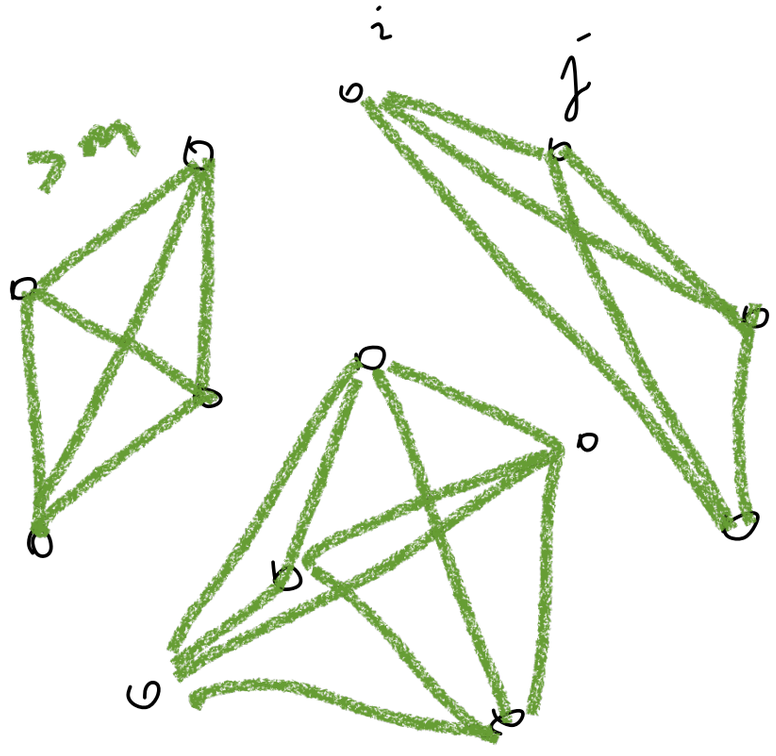


Tree Structure

- [Hirai-Murota] Tree structure on the Hessian. Since:

$$\Delta^S_{ij} \geq \min(\Delta^S_{ik}, \Delta^S_{kj}) \geq 0$$

implies that in $\Delta^S_{ij}, \Delta^S_{ik}, \Delta^S_{kj}$ the minimum must appear twice. So, let $m = \min_{i,j \notin S} \Delta^S_{ij}$

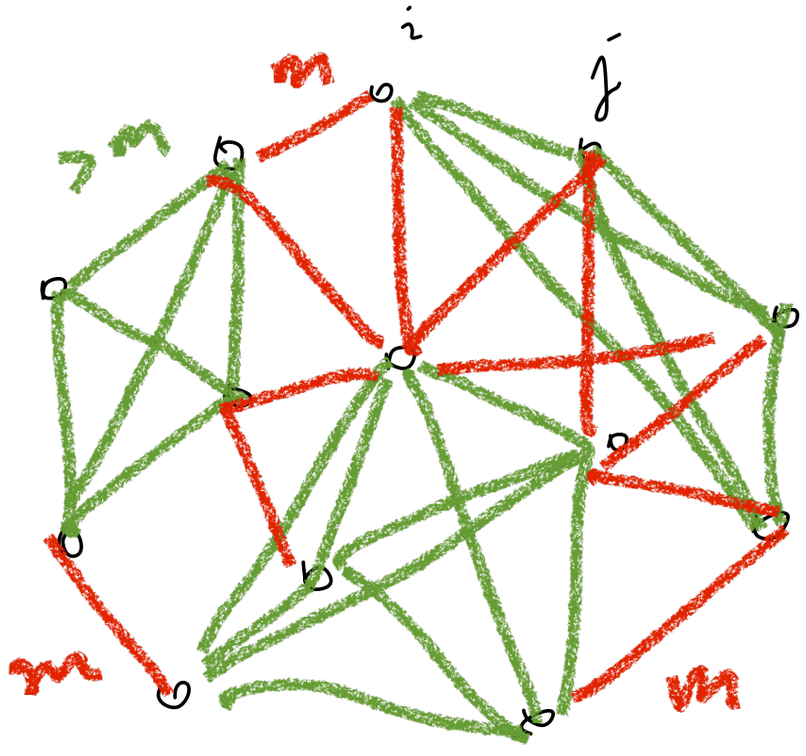


Tree Structure

- [Hirai-Murota] Tree structure on the Hessian. Since:

$$\Delta^S_{ij} \geq \min(\Delta^S_{ik}, \Delta^S_{kj}) \geq 0$$

implies that in $\Delta^S_{ij}, \Delta^S_{ik}, \Delta^S_{kj}$ the minimum must appear twice. So, let $m = \min_{i,j \notin S} \Delta^S_{ij}$

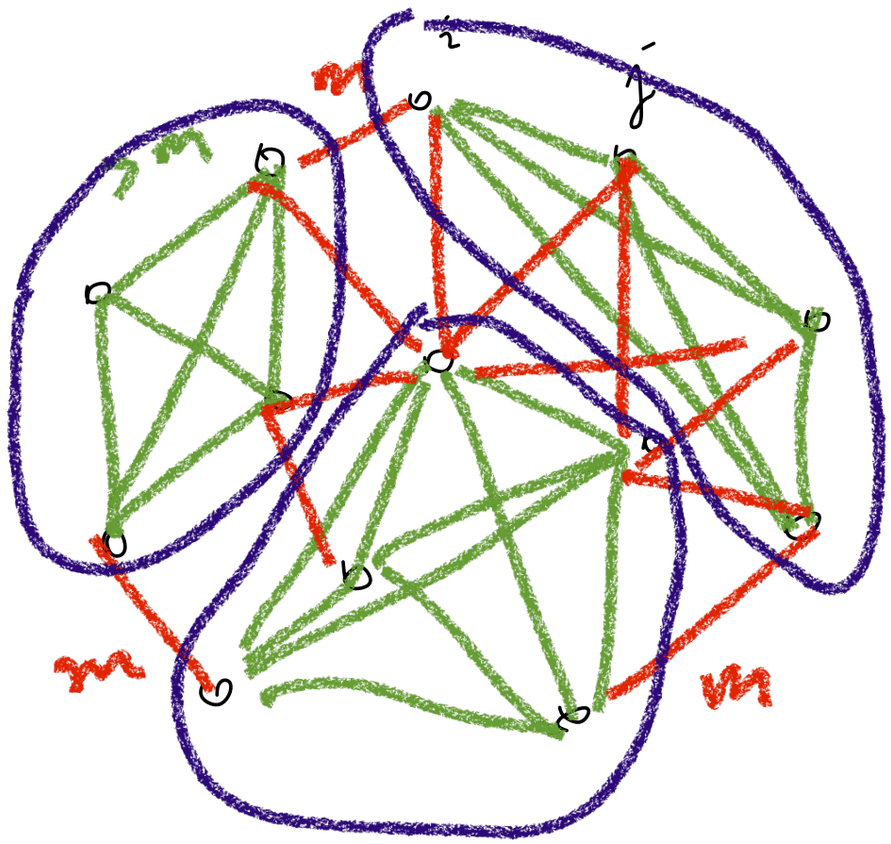


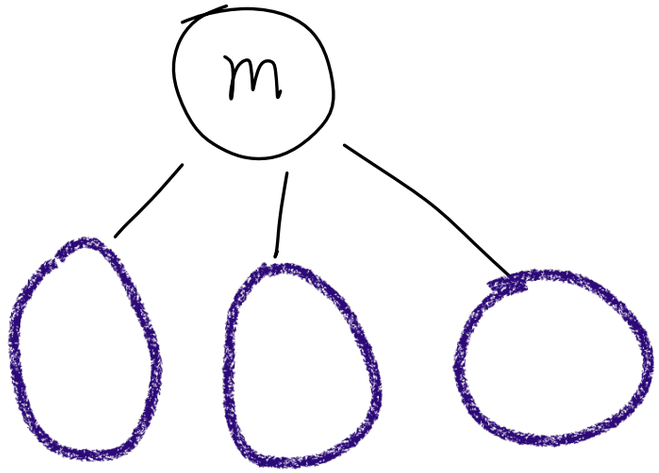
Tree Structure

- [Hirai-Murota] Tree structure on the Hessian. Since:

$$\Delta^S_{ij} \geq \min(\Delta^S_{ik}, \Delta^S_{kj}) \geq 0$$

implies that in $\Delta^S_{ij}, \Delta^S_{ik}, \Delta^S_{kj}$ the minimum must appear twice. So, let $m = \min_{i,j \notin S} \Delta^S_{ij}$





Tree Structure

- For every subset S we can represent the Hessian by a tree with $[n] \setminus S$ in the leaves and labels in the internal nodes such that:

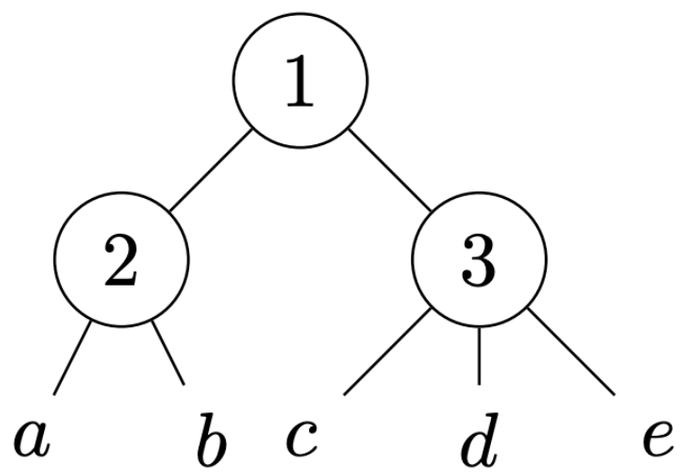
$$\Delta^S_{ij} = \mathrm{label}(\mathrm{LCA}_{ij})$$

Tree Structure

- For every subset S we can represent the Hessian by a tree with $[n] \setminus S$ in the leaves and labels in the internal nodes such that:

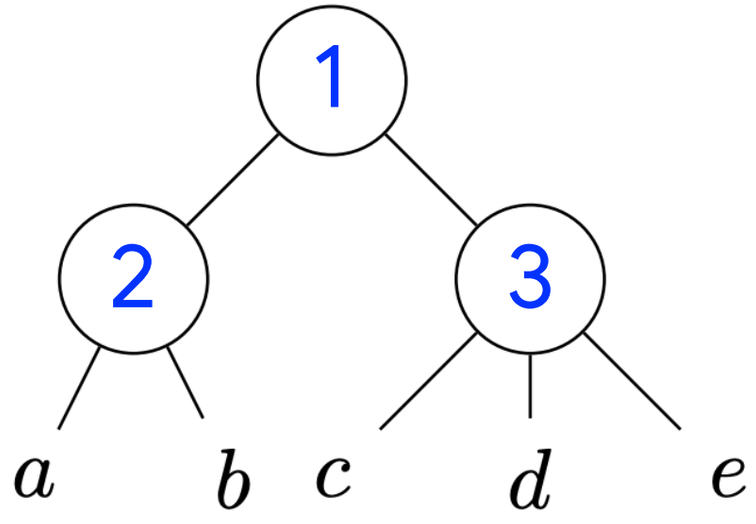
$$\Delta^S_{ij} = \mathrm{label}(\mathrm{LCA}_{ij})$$

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} * & 2 & 1 & 1 & 1 \\ 2 & * & 1 & 1 & 1 \\ 1 & 1 & * & 3 & 3 \\ 1 & 1 & 3 & * & 3 \\ 1 & 1 & 3 & 3 & * \end{bmatrix} \end{array}$$



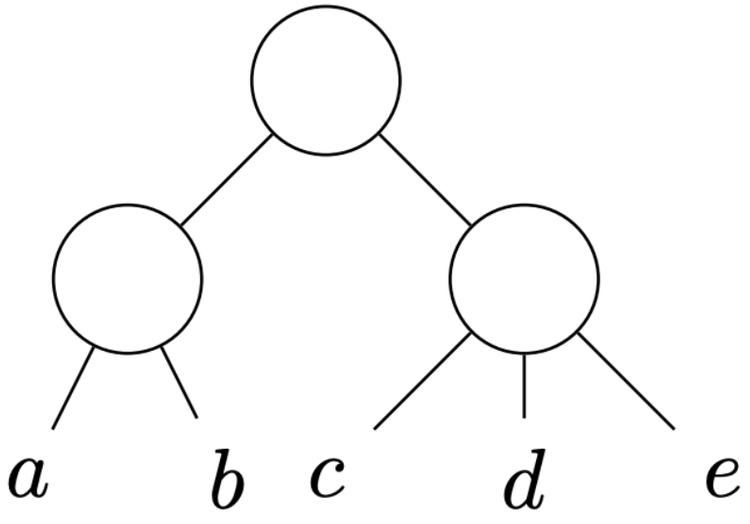
Tree Structure

- This allows us to decompose a GS function in a combinatorial structure and numerical data. For each subset S we associate:



Tree Structure

- This allows us to decompose a GS function in a combinatorial structure and numerical data. For each subset S we associate:

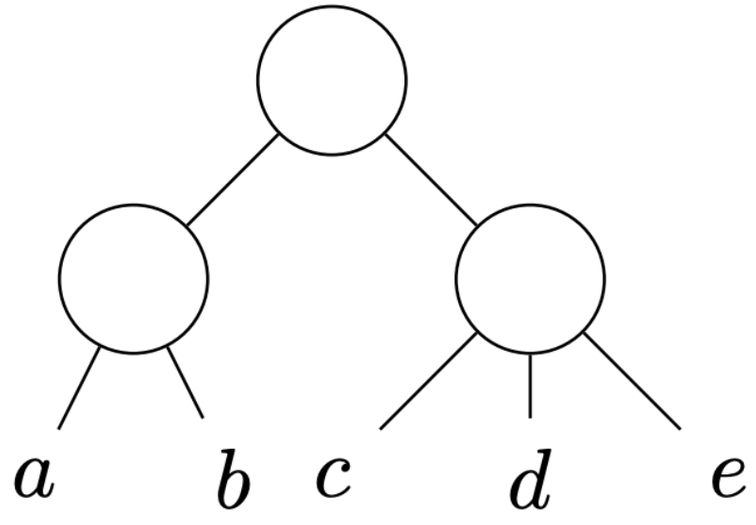


1

2 3

Tree Structure

- This allows us to decompose a GS function in a combinatorial structure and numerical data. For each subset S we associate:



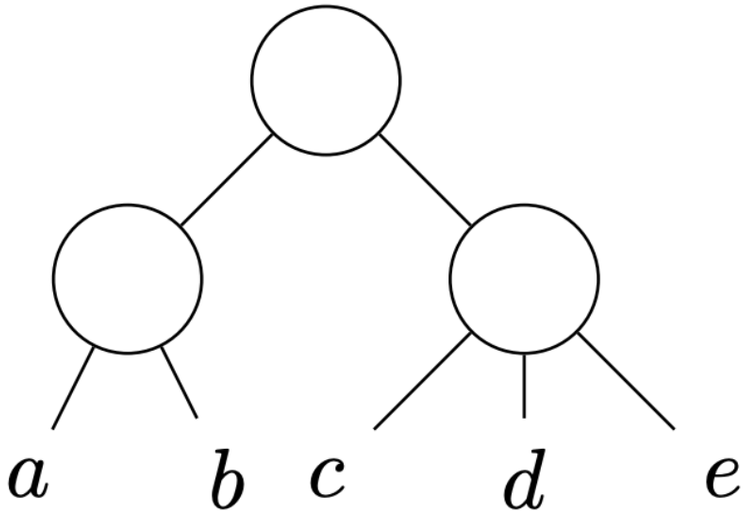
Combinatorial Structure

1

2 3

Tree Structure

- This allows us to decompose a GS function in a combinatorial structure and numerical data. For each subset S we associate:

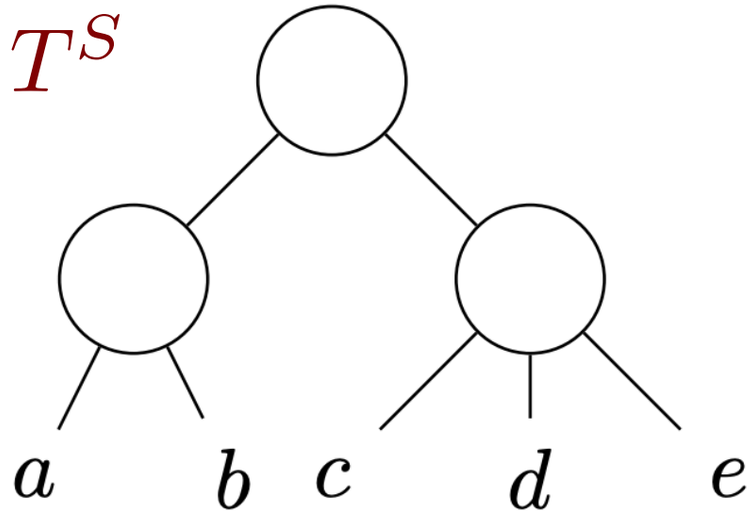


Combinatorial Structure

Numerical Data

Tree Structure

- This allows us to decompose a GS function in a combinatorial structure and numerical data. For each subset S we associate:



Combinatorial Structure

Numerical Data

Signature $= \{T^S\}_{S \subseteq [n]}$

Tree Structure

- Thm: given $u, v \in \mathrm{GS}$ then $u + v \in \mathrm{GS}$ iff they share the same signature.

- What are the building blocks of GS (under tree-concordant sum) ?

- Let's fix a signature $\{T^S\}_{S \subseteq [n]}$ and look at the GS functions with that signature. This forms a convex set.
- What are the extremal points ?

Tree Structure

- Thm: given $u, v \in \mathrm{GS}$ then $u + v \in \mathrm{GS}$ iff they share the same signature.

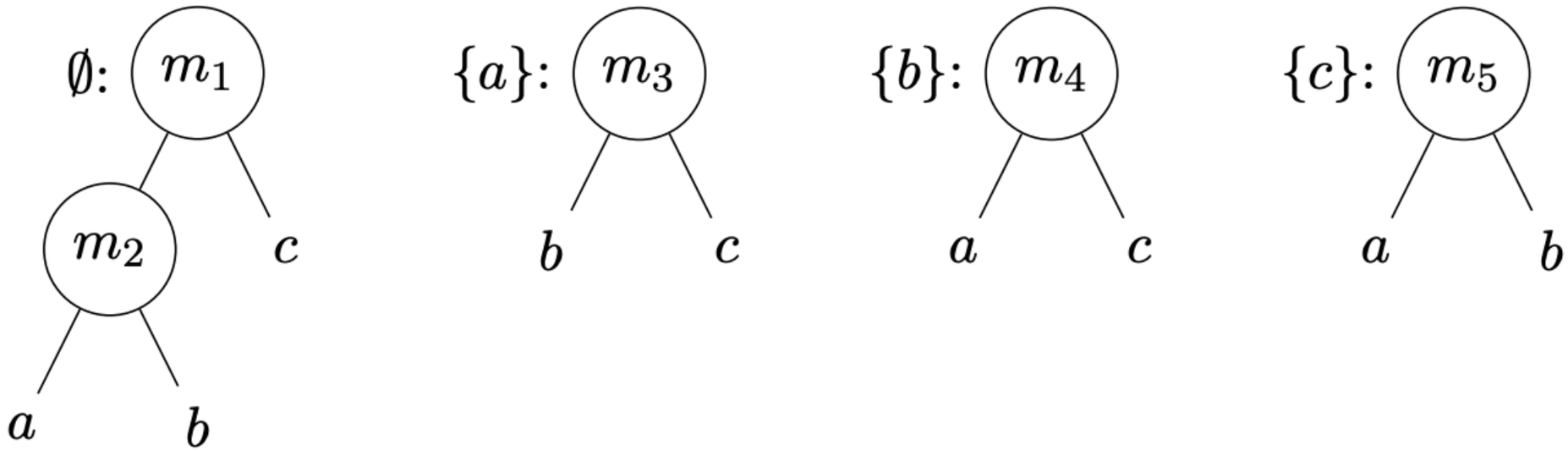
- What are the building blocks of GS (under tree-concordant sum) ?

- Let's fix a signature $\{T^S\}_{S \subseteq [n]}$ and look at the GS functions with that signature. This forms a convex set.
- What are the extremal points ?

I was hoping they would be all matroid rank functions. But it is not always the case.

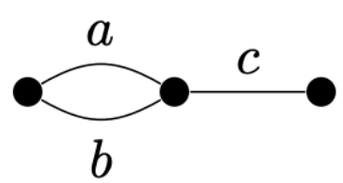
Extremal GS valuations

- Say we have 3 items. Then there is only one signature (modulo symmetries):



- Look at the polytope formed by m_1, m_2, m_3, m_4, m_5

  Monotonicity ($m_1 \leq m_2$) + Integrability

- For n = 3, the extreme points are all matroid rank functions: 

+ uniform matroids of ranks 1 and 2

Extremal GS valuations

- Matroid rank functions correspond to {0,1} labels, i.e. $v \in \mathrm{GS}$ and $\Delta_{ij}^{S} \in \{0,1\}$ iff v is matroid rank function.
- So we want to check if polytopes are integral.
- True for n $=$ 4 but fails for n $>$ 4.

- By product is a complete description of GS up to 4 items.