Selfish, local and online scheduling via vector fitting

Danish Kashaev

CWI Amsterdam, Networks & Optimization

September 28, 2025

Strategic games

Definition

An instance of a *finite strategic game* consists of:

- A set $N = \{1, \dots n\}$ of players
- A strategy set $\mathcal{S}_j$ for every player $j \in N$
- A cost function $C_j : \mathcal{S}_1 \times \cdots \times \mathcal{S}_n \to \mathbb{R}$ for every player $j \in N$.

Each player $j \in N$ needs to pick one strategy $i \in \mathcal{S}_j$. We denote by

$$x_{ij} \in \{0, 1\}$$

the indicator value whether j chooses $i \in \mathcal{S}_j$.

Example: Load Balancing

Example: Load balancing

Given is a set of *resources* E. The strategy set of every player $j \in N$ is $\mathcal{S}_j \subseteq E$ with weights $w_{ij} \geq 0$ for every $i \in \mathcal{S}_j$.

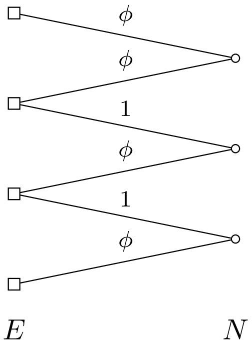
The *load* of $i \in E$ is:

$$\ell_i(x) = \sum_{j \in N} w_{ij} x_{ij}$$

The cost of $j \in N$ is:

$$C_j(x) = \sum_{i \in E} \ell_i(x)\, w_{ij}\, x_{ij}$$

The cost is the load of the picked resource multiplied by the weight.



Example: Load Balancing

Example: Load balancing

Given is a set of *resources* E. The strategy set of every player $j \in N$ is $\mathcal{S}_j \subseteq E$ with weights $w_{ij} \geq 0$ for every $i \in \mathcal{S}_j$.

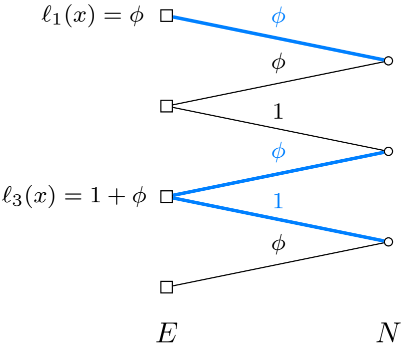
The *load* of $i \in E$ is:

$$\ell_i(x) = \sum_{j \in N} w_{ij} x_{ij}$$

The cost of $j \in N$ is:

$$C_j(x) = \sum_{i \in E} \ell_i(x)\, w_{ij}\, x_{ij}$$

The cost is the load of the picked resource multiplied by the weight.



Example: Load Balancing

Example: Load balancing

Given is a set of *resources* E. The strategy set of every player $j \in N$ is $\mathcal{S}_j \subseteq E$ with weights $w_{ij} \geq 0$ for every $i \in \mathcal{S}_j$.

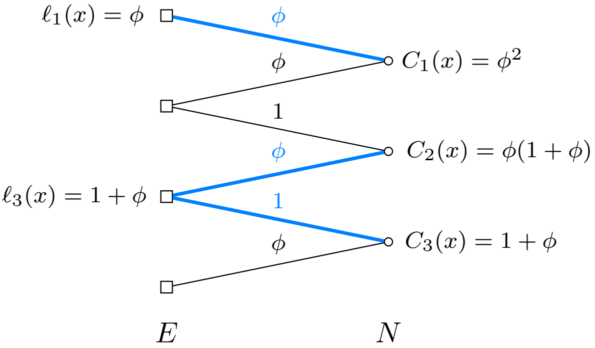
The *load* of $i \in E$ is:

$$\ell_i(x) = \sum_{j \in N} w_{ij} x_{ij}$$

The cost of $j \in N$ is:

$$C_j(x) = \sum_{i \in E} \ell_i(x)\, w_{ij}\, x_{ij}$$

The cost is the load of the picked resource multiplied by the weight.



Nash equilibria

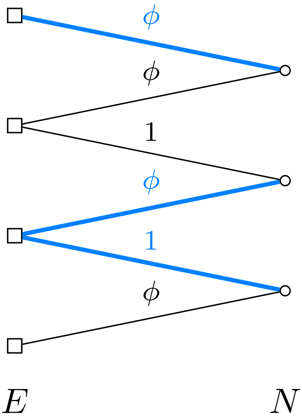
Definition

An assignment $(x_{ij})_{j \in N, i \in \mathcal{S}_j}$ is a *pure Nash equilibrium* if

$$C_j(x) \le C_j(x_{-j}, i) \qquad \forall j \in N, \forall i \in \mathcal{S}_j.$$

Let $\phi \approx 1.618$

s.t. $\phi^2 = 1 + \phi$



Nash equilibria

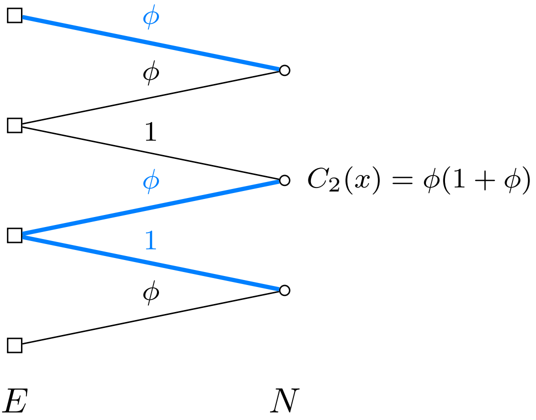
Definition

An assignment $(x_{ij})_{j \in N, i \in \mathcal{S}_j}$ is a *pure Nash equilibrium* if

$$C_j(x) \leq C_j(x_{-j}, i) \qquad \forall j \in N, \forall i \in \mathcal{S}_j.$$

Let $\phi \approx 1.618$

s.t. $\phi^2 = 1 + \phi$



Nash equilibria

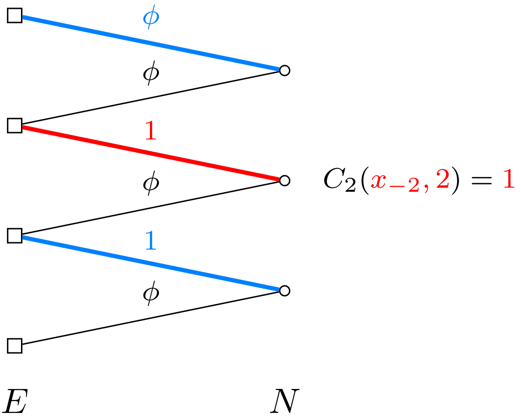
Definition

An assignment $(x_{ij})_{j \in N, i \in \mathcal{S}_j}$ is a *pure Nash equilibrium* if

$$C_j(x) \leq C_j(x_{-j}, i) \qquad \forall j \in N, \forall i \in \mathcal{S}_j.$$

Let $\phi \approx 1.618$

s.t. $\phi^2 = 1 + \phi$



Social cost and social optimum

Social cost

The *social cost* of an assignment x is defined as

$$C(x) = \sum_{j \in N} C_j(x)$$

The *social optimum* is the optimal solution x^* to:

$$\begin{aligned} & \min C(x) \\ & \sum_{i \in \mathcal{S}_j} x_{ij} = 1 \qquad \forall j \in N \\ & x_{ij} \in \{0, 1\} \qquad \forall j \in N, \forall i \in \mathcal{S}_j. \end{aligned}$$

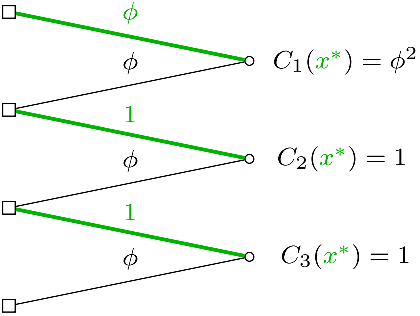
Price of Anarchy

Price of Anarchy

The *price of anarchy* of a game is the worst-case, over all instances, of

$$\frac{C(x)}{C(x^*)} \in [1, \infty]$$

where x is any Nash equilibrium and x^* is the social optimum.



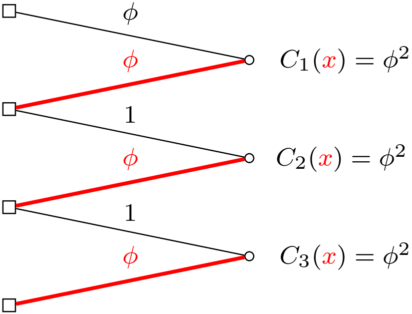
Price of Anarchy

Price of Anarchy

The *price of anarchy* of a game is the worst-case, over all instances, of

$$\frac{C(x)}{C(x^*)} \in [1, \infty]$$

where x is any Nash equilibrium and x^* is the social optimum.



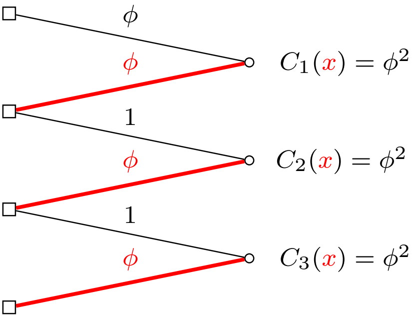
Price of Anarchy

Price of Anarchy

The *price of anarchy* of a game is the worst-case, over all instances, of

$$\frac{C(x)}{C(x^*)} \in [1, \infty]$$

where x is any Nash equilibrium and x^* is the social optimum.



$$\frac{C(x)}{C(x^*)} = \frac{n\,\phi^2}{n-1+\phi^2}$$

$$\xrightarrow{n\to\infty} \phi^2 \approx 2.618$$

Price of Anarchy

- Bounding the price of anarchy is a central question in AGT.
- One general successful approach is the *smoothness framework* [Roughgarden, 2009]
- Another idea is to use *convex programming duality* [Kulkarni, Mirrokni 2015]

Price of Anarchy

- Bounding the price of anarchy is a central question in AGT.
- One general successful approach is the *smoothness framework* [Roughgarden, 2009]
- Another idea is to use *convex programming duality* [Kulkarni, Mirrokni 2015]

This talk

Unified simple approach to tightly bound the price of anarchy for a class of games where $C(x)$ is quadratic in x.

- Dual fitting argument on a semidefinite program.
- SDP can be obtained automatically using first round of Lasserre/SoS hierarchy.

Technique also works to bound the *approximation ratio* of local search algorithms and *competitive ratio* of online algorithms.

Results obtained

Price of Anarchy

- Price of anarchy for $R||\sum_j w_j C_j$ under three different scheduling policies obtaining tight bounds of $4, 2.618$ and 2.133 (STOC 2011)
- Slight improvement to 2 for the last bound in special cases
- Price of anarchy for weighted affine congestion games
- Pure price of anarchy of $P||\sum_j w_j C_j$

Local search

- Tight analysis of best known combinatorial approximation algorithm for $R||\sum_j w_j C_j$ based on local search

Online algorithms

- Tight analysis of competitive ratio of different online algorithms for online load balancing in the L_2 norm.
- Tight analysis of (optimal) greedy online algorithm for $R||\sum_j w_j C_j$

Dual fitting: high level view

Exact integer program to compute social optimum x^*:

$$C(x^*) = \min_x \left\{ C(x) : \sum_{i \in \mathcal{S}_j} x_{ij} = 1 \quad \forall j; \quad x_{ij} \in \{0,1\} \quad \forall j, i \right\}$$

Idea: formulate a convex relaxation and consider the dual. By weak duality, for any feasible solution y to the dual:

$$obj(y) \le C(x^*)$$

Dual fitting: high level view

Exact integer program to compute social optimum x^*:

$$C(x^*) = \min_x \left\{ C(x) : \sum_{i \in \mathcal{S}_j} x_{ij} = 1 \quad \forall j; \quad x_{ij} \in \{0,1\} \quad \forall j, i \right\}$$

Idea: formulate a convex relaxation and consider the dual. By weak duality, for any feasible solution y to the dual:

$$obj(y) \leq C(x^*)$$

Goal:

For any Nash equilibrium x, find a dual solution y such that

$$obj(y) \geq \rho\, C(x) \quad \text{for some} \quad \rho \in [0,1]$$

This implies a price of anarchy upper bound:

$$\implies \frac{C(x)}{C(x^*)} \leq \frac{1}{\rho}$$

Convex SDP relaxation

Exact program to compute x^*:

$$\min_x \left\{ C(x) : \sum_{i \in \mathcal{S}_j} x_{ij} = 1 \quad \forall j; \quad x_{ij} \in \{0,1\} \quad \forall j,i \right\}$$

If $C(x)$ is quadratic, then $C(x) = \langle C, X \rangle = \mathrm{Tr}(CX)$ for some symmetric matrix C, where $X = (1,x)(1,x)^T$ encodes all the linear and quadratic terms of x.

$$\begin{aligned}
\min \langle C, X \rangle & & \\
\sum_{i \in \mathcal{S}_j} X_{\{ij,\, ij\}} &= 1 & \forall j \in N \\
X_{\{0,0\}} &= 1 & \\
X_{\{0,\, ij\}} &= X_{\{ij,\, ij\}} & \forall j \in N, i \in \mathcal{S}_j \\
X &\succeq 0 &
\end{aligned}$$

The dual SDP

$$\max \sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2$$

$$y_j \le C_{\{ij,\, ij\}} - \frac{1}{2}\|v_{ij}\|^2 + \langle v_0, v_{ij}\rangle \qquad \forall j \in N, i \in \mathcal{S}_j$$

$$\langle v_{ij}, v_{i'k}\rangle \le 2\, C_{\{ij,\, i'k\}} \qquad \forall (i,j) \neq (i',k) \text{ with } j,k \in N$$

Variables:

- Scalars $y_j \in \mathbb{R}$ for every $j \in N$
- Vectors $v_0 \in \mathbb{R}^d$ and $v_{ij} \in \mathbb{R}^d$ for every $j \in N, i \in \mathcal{S}_j$

The dual SDP

$$\max \sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2$$

$$y_j \leq C_{\{ij,\, ij\}} - \frac{1}{2}\|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \qquad \forall j \in N, i \in \mathcal{S}_j$$

$$\langle v_{ij}, v_{i'k} \rangle \leq 2\, C_{\{ij,\, i'k\}} \qquad \forall (i,j) \neq (i',k) \text{ with } j,k \in N$$

Variables:

- Scalars $y_j \in \mathbb{R}$ for every $j \in N$
- Vectors $v_0 \in \mathbb{R}^d$ and $v_{ij} \in \mathbb{R}^d$ for every $j \in N, i \in \mathcal{S}_j$

Goal:

For any Nash equilibrium x, find a feasible dual solution such that

$$\sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2 \geq \rho\, C(x) \quad \text{for some} \quad \rho \in [0,1]$$

The dual SDP

$$\max \sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2$$

$$y_j \leq C_{\{ij,\, ij\}} - \frac{1}{2}\|v_{ij}\|^2 + \langle v_0, v_{ij}\rangle \qquad \forall j \in N, i \in \mathcal{S}_j \tag{1}$$

$$\langle v_{ij}, v_{i'k}\rangle \leq 2\, C_{\{ij,\, i'k\}} \qquad \forall (i,j) \neq (i',k) \text{ with } j,k \in N$$

Definition: Nash equilibria

An assignment $(x_{ij})_{j \in N, i \in \mathcal{S}_j}$ is a *pure Nash equilibrium* if

$$C_j(x) \leq C_j(x_{-j}, i) \qquad \forall j \in N, i \in \mathcal{S}_j. \tag{2}$$

Key insight: Make sure (1) corresponds to (2) in the dual fitting.

Back to our Example: Load Balancing

Specialize the SDP:

$$\max \sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2$$

$$y_j \leq w_{ij}^2 - \frac{1}{2}\|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \qquad \forall j \in N, i \in \mathcal{S}_j$$

$$\langle v_{ij}, v_{i'k} \rangle \leq 2\, w_{ij}\, w_{ik}\, \mathbb{1}_{\{i=i'\}} \qquad \forall (i,j) \neq (i',k) \text{ with } j,k \in N$$

Nash equilibria inequalities:

$$C_j(x) \leq w_{ij}^2 + w_{ij}\, \ell_i(x) \qquad \forall j \in N, \forall i \in \mathcal{S}_j.$$

Fitting idea:

$$y_j \sim C_j(x) \quad , \quad w_{ij}^2 - \frac{1}{2}\|v_{ij}\|^2 \sim w_{ij}^2 \quad , \quad \langle v_0, v_{ij} \rangle \sim w_{ij}\, \ell_i(x)$$

Back to our Example: Load Balancing

SDP constraints:

$$y_j \leq w_{ij}^2 - \frac{1}{2}\|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \qquad \forall j \in N, i \in \mathcal{S}_j$$

$$\langle v_{ij}, v_{i'k} \rangle \leq 2\, w_{ij}\, w_{ik}\, \mathbb{1}_{\{i=i'\}} \qquad \forall (i,j) \neq (i',k)$$

We want:

$$y_j \sim C_j(x) \quad , \quad w_{ij}^2 - \frac{1}{2}\|v_{ij}\|^2 \sim w_{ij}^2 \quad , \quad \langle v_0, v_{ij} \rangle \sim w_{ij}\, \ell_i(x)$$

Fitting ensuring the above:

- $v_{ij} \in \mathbb{R}^E$ such that $v_{ij}(e) = \alpha\, w_{ij}\, \mathbb{1}_{\{i=e\}}$ for some $0 \leq \alpha \leq \sqrt{2}$
- $v_0 \in \mathbb{R}^E$ such that $v_0(e) = \beta\, \ell_e(x)$ for some $\beta \geq 0$
- $y_j = \alpha\beta\, C_j(x)$ where $\alpha\beta = 1 - \alpha^2/2$

Example: Load Balancing

- $v_{ij}(e) = \alpha\, w_{ij}\, \mathbb{1}_{\{i=e\}}$ for some $0 \leq \alpha \leq \sqrt{2}$
- $v_0(e) = \beta\, \ell_e(x)$ for some $\beta \geq 0$
- $y_j = \alpha\beta\, C_j(x)$ where $\alpha\beta = 1 - \alpha^2/2$

SDP objective: $\sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2 = \left(\alpha\beta - \frac{\beta^2}{2}\right) C(x)$

Example: Load Balancing

- $v_{ij}(e) = \alpha\, w_{ij}\, \mathbb{1}_{\{i=e\}}$ for some $0 \leq \alpha \leq \sqrt{2}$
- $v_0(e) = \beta\, \ell_e(x)$ for some $\beta \geq 0$
- $y_j = \alpha\beta\, C_j(x)$ where $\alpha\beta = 1 - \alpha^2/2$

SDP objective: $\sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2 = \left(\alpha\beta - \frac{\beta^2}{2}\right) C(x)$

Goal:

For any Nash equilibrium x, find a feasible dual solution such that

$$\sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2 \geq \rho\, C(x) \quad \text{for some} \quad \rho \in [0,1]$$

Solve:

$$\max\left\{\alpha\beta - \frac{\beta^2}{2} : \alpha\beta = 1 - \alpha^2/2, \alpha \in [0, \sqrt{2}], \beta \geq 0\right\} = \frac{2}{3 + \sqrt{5}}$$

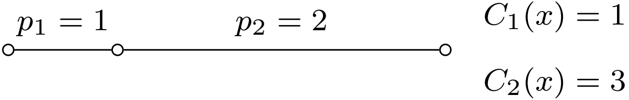
The scheduling problem $R||\sum_j w_j C_j$

Given is a set of machines M and a set of jobs N. Each job $j \in N$ has a weight $w_j > 0$ and processing time $p_{ij} > 0$ for every machine $i \in M$.

Goal: Assign each job to a machine and order the jobs on each machine to minimize the *sum of weighted completion times* of the jobs.

Known: Once an assignment is fixed, each machine should order the jobs assigned to it optimally by increasing *Smith* ratio

$$\delta_{ij} := \frac{p_{ij}}{w_j}$$



Game theoretic setting

- Each job $j \in N$ picks a machine $i \in M$. Denote by $x_{ij} \in \{0,1\}$ if $j \in N$ chooses machine $i \in M$. For two jobs $j \neq k$ assigned to i:

$$k \prec_i j \iff \delta_{ik} < \delta_{ij} \iff p_{ik}/w_k < p_{ij}/w_j$$

The **completion time (cost)** is:

$$C_j(x) = \sum_{i \in M} x_{ij}\Big(p_{ij} + \sum_{k \prec_i j} p_{ik} x_{ik}\Big).$$

The **social cost** is:

$$C(x) = \sum_{j \in N} w_j C_j(x)$$

Theorem

The price of anarchy of this game is 4 (STOC 2011) [Cole et. al.]

Proof maps strategy vectors into a cleverly chosen *inner product space*.

Bounding the PoA

Specialize the SDP:

$$\max \sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2$$

$$y_j \leq w_j\, p_{ij} - \frac{1}{2}\|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \qquad \forall j \in N, i \in \mathcal{S}_j$$

$$\langle v_{ij}, v_{i'k} \rangle \leq w_j w_k \min\{\delta_{ij}, \delta_{ik}\}\, \mathbb{1}_{\{i=i'\}} \qquad \forall (i,j) \neq (i',k)$$

Nash equilibria inequalities imply:

$$w_j C_j(x) \leq w_j\, p_{ij} + \sum_{k \in N} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} \qquad \forall j \in N, \forall i \in \mathcal{S}_j.$$

Fitting idea:

$$y_j \sim w_j C_j(x) \quad , \quad w_j p_{ij} - \frac{1}{2}\|v_{ij}\|^2 \sim w_j p_{ij} \quad , \quad \langle v_0, v_{ij} \rangle \sim \ldots$$

Using inner product space of [Cole et. al.]

We want:

$$y_j \sim w_j C_j(x) \quad , \quad \|v_{ij}\|^2 \sim w_j\, p_{ij} \quad , \quad \langle v_{ij}, v_{ik}\rangle \sim w_j w_k \min\{\delta_{ij}, \delta_{ik}\}$$

Inner product space: Interpret SDP vectors as functions $f : [0, \infty) \to \mathbb{R}_+$ with inner product

$$\langle f, g\rangle = \int_0^\infty f(t) g(t) dt$$

Fitting: $v_{ij}(t) \sim w_j\, \mathbb{1}_{\{t \le \delta_{ij}\}}$

$$\implies \|v_{ij}\|^2 \sim w_j^2 \int_0^\infty \mathbb{1}_{\{t \le \delta_{ij}\}} dt = w_j^2 \delta_{ij} = w_j\, p_{ij}$$

$$\implies \langle v_{ij}, v_{ik}\rangle \sim w_j\, w_k \int_0^\infty \mathbb{1}_{\{t \le \delta_{ij}\}} \mathbb{1}_{\{t \le \delta_{ik}\}} dt = w_j w_k \min\{\delta_{ij}, \delta_{ik}\}$$

Different coordination mechanisms

Changing the ordering policy on each machine can improve the price of anarchy.

Inner product spaces for MinSum coordination mechanisms (STOC 2011) [Cole et. al.]

Results

- Smith's Rule leads to a PoA of 4
- A preemptive mechanism called *Proportional Sharing* leads to a PoA of $(3+\sqrt{5})/2 \approx 2.618$
- A randomized mechanism called *Rand* leads to a PoA of 2.133

All these results can be recovered using the vector fitting approach by exploiting the inner product space developed in [Cole et. al.]

Congestion games and selfish routing

Selfish routing

Given a graph $G = (V, E)$ and a set of N players. Each player $j \in N$ has a *weight* $w_j > 0$, a *source* node $s_j \in V$, *sink* node $t_j \in V$ and needs to pick a path in G between s_j and t_j

$$\mathcal{S}_j := \{P \subseteq E : P \text{ is a path between } s_j \text{ and } t_j\}$$

Congestion games and selfish routing

Selfish routing

Given a graph $G = (V, E)$ and a set of N players. Each player $j \in N$ has a *weight* $w_j > 0$, a *source* node $s_j \in V$, *sink* node $t_j \in V$ and needs to pick a path in G between s_j and t_j

$$\mathcal{S}_j := \{P \subseteq E : P \text{ is a path between } s_j \text{ and } t_j\}$$

If player j picks a path P, the **cost** is

$$C_j(x) = w_j \sum_{e \in P} \ell_e(x)$$

where $\ell_e(x)$ is the total weight of players using edge $e \in E$ under an assignment x. The **social cost** is

$$C(x) = \sum_{j \in N} C_j(x)$$

Congestion games and selfish routing

Theorem

The price of anarchy of this game is $(3+\sqrt{5})/2 \approx 2.618$ (STOC 2005) .
It can be recovered using the vector fitting approach.

Key idea: We now have a variable $v_{Pj} \in \mathbb{R}^E$ for each player $j \in N$ and each path $P \in \mathcal{S}_j$. Fit

$$v_{Pj}(e) = w_j \, \mathbb{1}_{\{e \in P\}}$$

The *support* of the vector v_{Pj} are the edges on the path. Previously (in the scheduling setting), the support had size one.

Analyzing local search and online algorithms

$$\max \sum_{j \in N} y_j - \frac{1}{2}\|v_0\|^2$$

$$y_j \leq C_{\{ij,\, ij\}} - \frac{1}{2}\|v_{ij}\|^2 + \langle v_0, v_{ij}\rangle \qquad \forall j \in N, i \in \mathcal{S}_j \tag{3}$$

$$\langle v_{ij}, v_{i'k}\rangle \leq 2\, C_{\{ij,\, i'k\}} \qquad \forall (i,j) \neq (i',k) \text{ with } j,k \in N$$

- *Price of anarchy*: make (3) correspond to Nash equilibria inequalities
- *Local search:* make (3) correspond to local optima inequalities
- *Online algorithms:* make (3) correspond to inequalities satisfied by an online algorithm at every time step

All three are applicable to the scheduling problem $R||\sum_j w_j C_j$ in these different settings.

Conclusion

Conclusion

- Unified proof technique for problems whose optimal solution can be cast as a binary quadratic program
- SDP relaxation can be obtained by the first round of Lasserre/SoS hierarchy
- Recovers and unifies numerous results
- Extension from scheduling to congestion
- Works in game theoretic, local search and online settings

Future work ideas

- Apply this technique to new games or problems with a quadratic objective
- Possible to extend this technique to higher degree polynomial objective by considering later rounds of the hierarchy?

Conclusion

Conclusion

- Unified proof technique for problems whose optimal solution can be cast as a binary quadratic program
- SDP relaxation can be obtained by the first round of Lasserre/SoS hierarchy
- Recovers and unifies numerous results
- Extension from scheduling to congestion
- Works in game theoretic, local search and online settings

Future work ideas

- Apply this technique to new games or problems with a quadratic objective
- Possible to extend this technique to higher degree polynomial objective by considering later rounds of the hierarchy?

Thanks!