Action-Graph Games, and an Algorithm for Computing their Equilibria

Navin A. R. Bhat
University of Toronto

Kevin Leyton-Brown
University of British Columbia

Compact Game Representations

- Extensive form: sequential structure
- Congestion games [Rosenthal, 1973]
  - anonymity: agents' payoffs depend on numbers of other agents choosing same resources, not on individual identities;
  - additivity over resources
- Graphical games [Kearns *et al.*, 2001]
  - strict utility independence holds between some pairs of agents
  - leveraged for rapid computation of equilibria (e.g.) [Blum *et al.* , 2003]
- Local-effect games [L-B & Tennenholtz, 2003]
  - context-specific independence
  - also symmetry, anonymity, monotonicity, additivity of local effects

Action-Graph Games

$AGG = \langle N, \mathbf{S}, S, \nu, u \rangle$

N = set of n **agents**

$\mathbf{S}$ = set of **pure action profiles**

$S_i \equiv$ action set of agent i

$\mathbf{S} \equiv \prod_{i \in N} S_i$

S = set of **distinct action choices**

$S \equiv \bigcup_{i \in N} S_i$

ν = **neighbor** relation

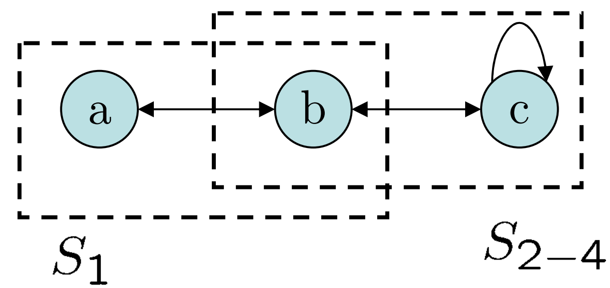
$\nu : S \mapsto 2^S$

u = **utility** function

$u : S \times \Delta \mapsto \mathbb{R}$

Δ = set of distributions of numbers of agents over distinct actions

key property: u depends only on numbers of agents who take *neighboring* actions



$N = \{1, 2, 3, 4\}; S = \{a, b, c\}$

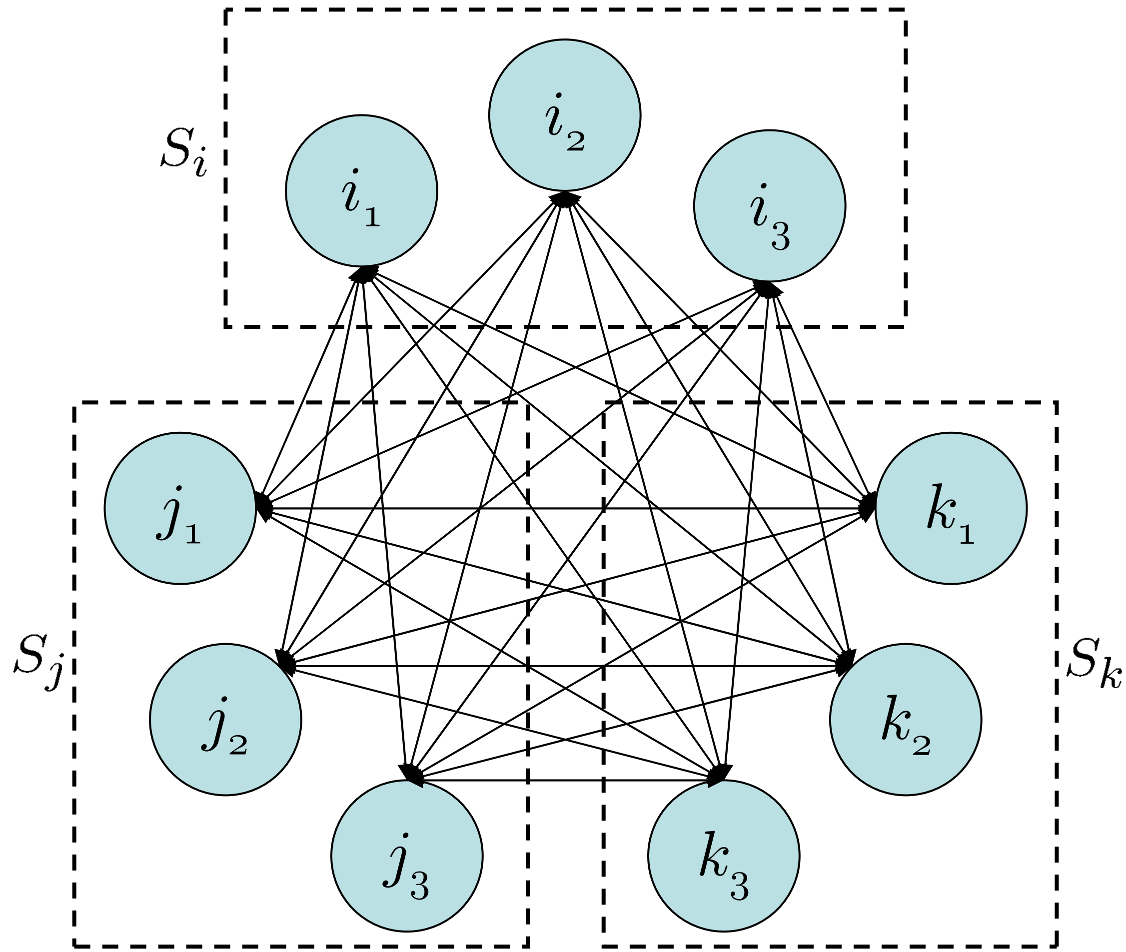
$S_1 = \{a, b\}; S_{2-4} = \{b, c\}$

$\nu(c) = \{b, c\}$

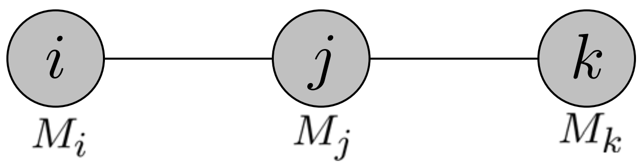
$u(c, D) = D(c) - D(b)^2$

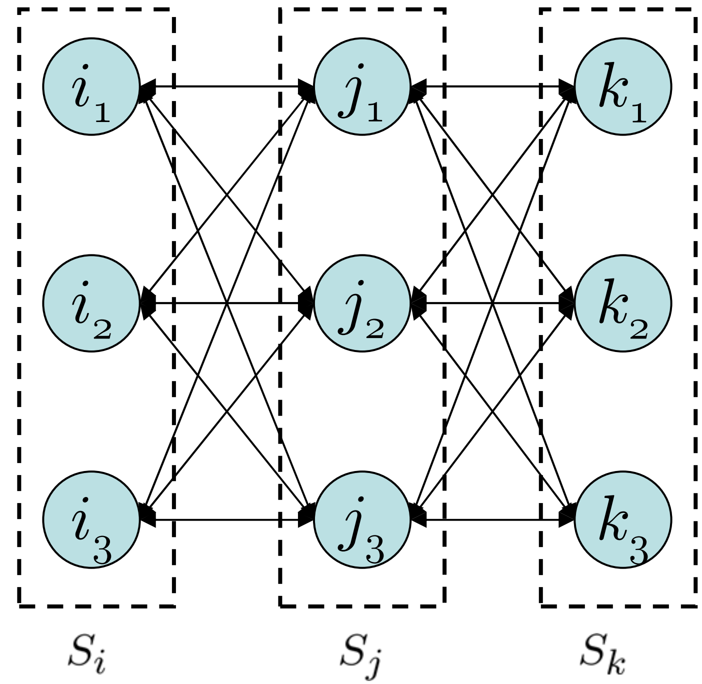
e.g., $D = (1, 1, 2)$

AGGs are Fully Expressive



Graphical Games as AGGs



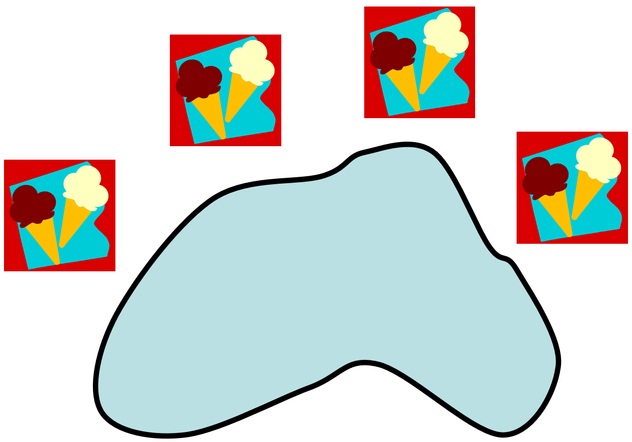


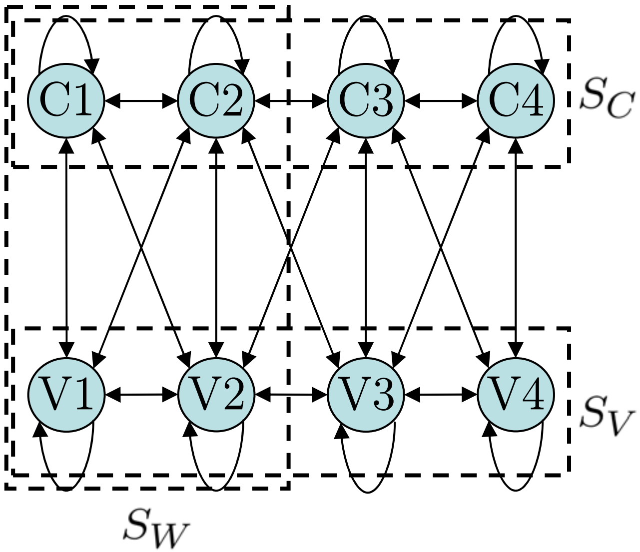
GG	AGG
Agent node	Action set box
Edge	Bipartite graphs between action sets
Local game matrix	Node utility function

Constrained Location Problem

n vendors sell either chocolate or vanilla ice cream at one of four stations along a beach

- n_C chocolate (C) vendors;
- n_V vanilla (V) vendors;
- n_W can sell C/V, but only on the west side.
- competition between nearby sellers of same type; synergy between nearby different types





Notes:

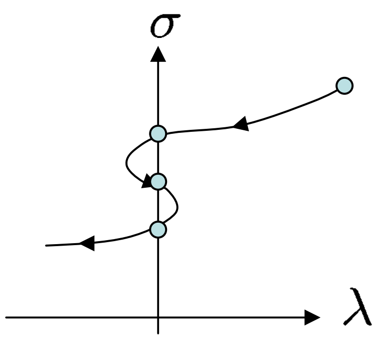
- representation independent of # agents
- overlapping action sets
- context-specific independence without strict independence

Other examples of compact AGGs:

- Role formation games
- Traffic routing games
- Product placement games
- Party affiliation games

Continuation Method for Equilibria

[Govindan & Wilson, 2003]



- $V^i_{s_i}(\sigma) \equiv$ expected payoff to agent i for playing action s_i, if other agents play according to mixed-strategy profile σ
- Deform payoff to obtain a game with known equilibrium:
  - add bonus, parameterized by λ: $V^i_{s_i}(\sigma) + \lambda b^i_{s_i}$
- Strategy improvement mapping: $\sigma \mapsto R(\sigma + V(\sigma))$
  - fixed points define equilibria
- Path following:
  - Initial (σ, λ) known
  - Compute local path direction
    - ∇V is bottleneck computation
  - Take small step along path; repeat

Payoff Jacobian

$$\frac{\partial V^i_{s_i}(\sigma_{-i})}{\partial \sigma_{i'}(s_{i'})} \equiv \nabla V^{i,i'}_{s_i,s_{i'}}(\overline{\sigma})$$
$$= \sum_{\overline{\mathbf{s}} \in \overline{\mathbf{S}}} u_i\left(s_i, s_{i'}, \overline{\mathbf{s}}\right) Pr(\overline{\mathbf{s}}|\overline{\sigma})$$

$$(\ \overline{*} \equiv -\{i, i'\}\ )$$

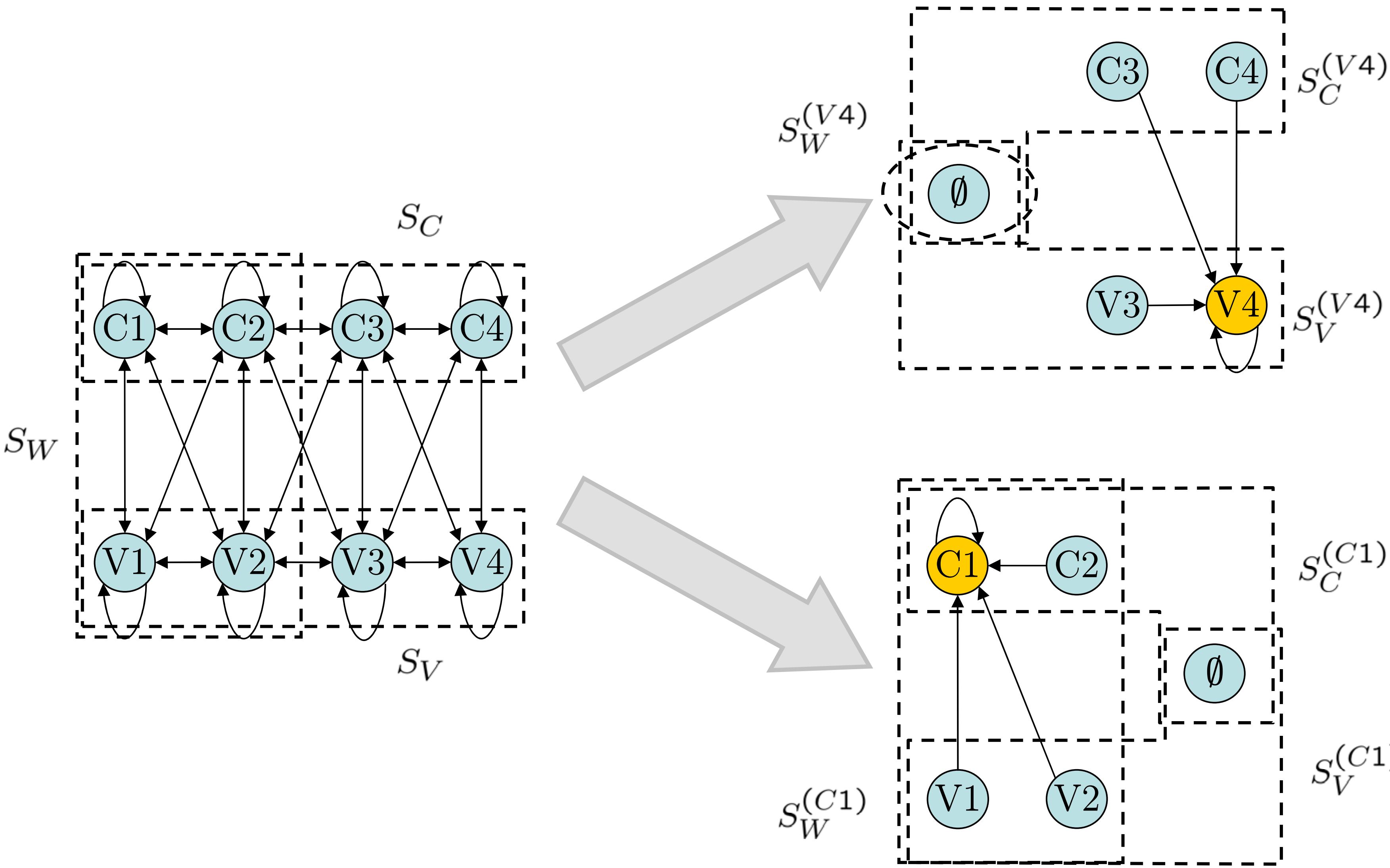
Computational complexity:

- $O\left(poly(\overline{n})poly(|S|) \qquad \right)$

Other applications of this Jacobian:

- Iterated Polymatrix Approximation (IPA)
  - a quick start for the continuation method
- Gradient for policy search multiagent RL algorithms

Projection



AGG Jacobian for Arbitrary Equilibria

- Projection captures context-specific independence and strict independence
- Writing in terms of the distribution captures anonymity

$$\nabla V^{i,i'}_{s_i,s_{i'}}(\sigma) = \sum_{\overline{D}^{(s_i)} \in \overline{\Delta}^{(s_i)}} u\left(s_i, \mathcal{D}\left(s_i, s_{i'}, \overline{D}^{(s_i)}\right)\right) Pr\left(\overline{D}^{(s_i)} | \overline{\sigma}^{(s_i)}\right);$$

$$Pr\left(\overline{D}^{(s_i)} | \overline{\sigma}^{(s_i)}\right) = \sum_{\overline{\mathbf{s}}^{(s_i)} \in \mathcal{S}\left(\overline{D}^{(s_i)}\right)} Pr\left(\overline{\mathbf{s}}^{(s_i)} | \overline{\sigma}^{(s_i)}\right)$$

$*^{(s)} \equiv$ projection with respect to action s

$\overline{*} \equiv -\{i, i'\}$

$\mathcal{S}(D) \equiv$ class of D, i.e. set of pure action profiles corresponding to D

AGG Jacobian for Arbitrary Equilibria

Theorem 1 *Computation of the Jacobian for an arbitrary action-graph game with maximum indegree* $\mathcal{I}$ *takes time that is* $O\left((\mathcal{I}+1)^{\overline{n}} poly(\overline{n}) poly\left(|S|\right)\right)$.

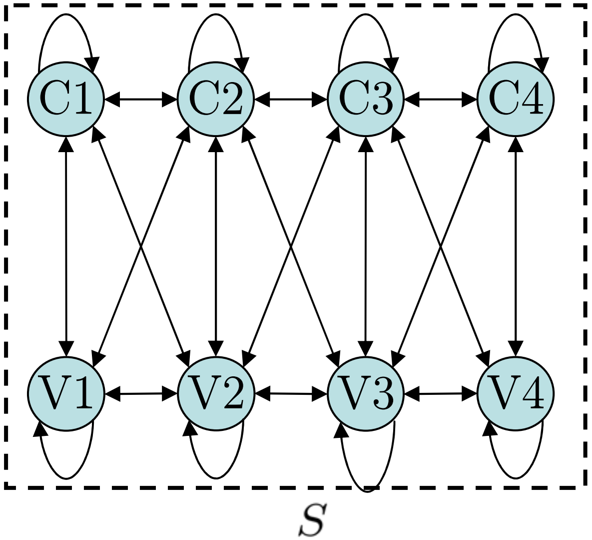
- Exponential speedup vs. GW: $O\left(|S|^{\overline{n}} poly(\overline{n}) poly(|S|)\right)$

Corollary 1 *For a* ***graphical game*** *encoded as an AGG, if* f *is the maximum family size and* α *is the maximum number of actions available to each agent, the Jacobian can be computed in time that is* $O\left(poly(\alpha^{f}) poly(\overline{n}) poly\left(|S|\right)\right)$.

- Same exponential speedup as Blum *et. al.* for computing the Jacobian for a graphical game using an explicit graphical game representation

Symmetric Equilibria

- Symmetric games are important
  - AGGs are symmetric when $\forall i,\ S_i = S$
- Nash [1951] proved all symmetric games have symmetric mixed-strategy equilibria: $\forall i,\ \sigma_i \equiv \sigma^*$
  - Jacobian simplifies because elements are agent-independent



- Continuation method:
  - seed with a symmetric equilibrium of the perturbed game
  - Jacobian is agent-independent
  - path traces to symmetric equilibrium of game of interest

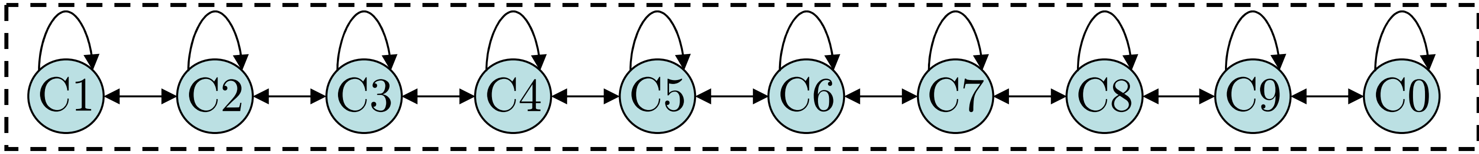
Symmetric AGG Jacobian

- All pure action profiles giving rise to the same distribution of agents are equally likely, so $Pr\left(\overline{D}^{(s_i)}|\overline{\sigma}^{(s_i)}\right)$ is just $Pr\left(\overline{\mathbf{s}}^{(s_i)}|\overline{\sigma}^{(s_i)}\right)$ times the number profiles that achieve $\overline{D}^{(s_i)}$
  - number of profiles: multinomial coefficients on projected graph

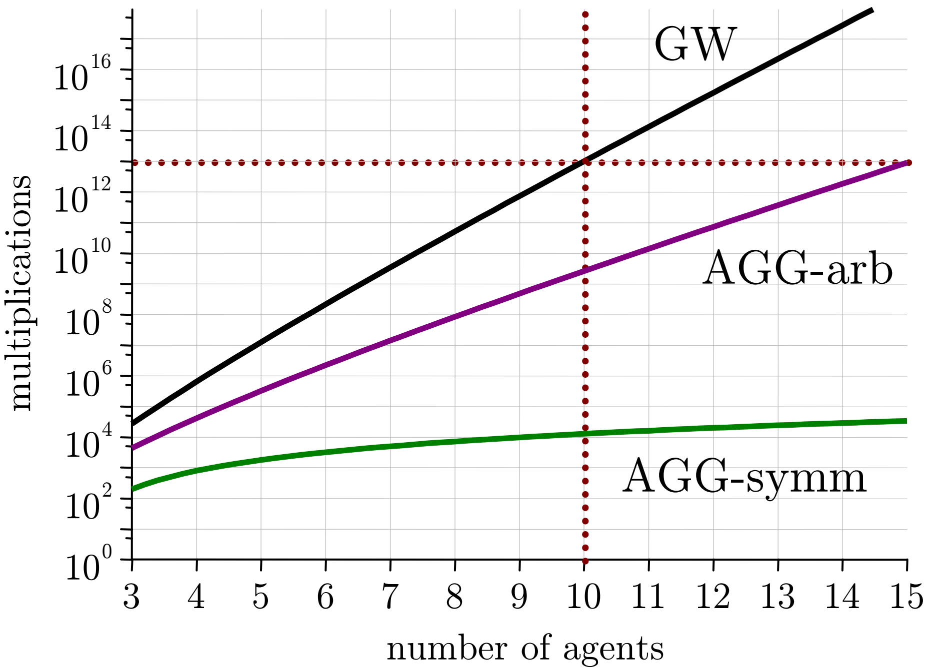
- Jacobian: sum over space $\overline{\Delta}^{(s_i)}$
  - space of projected distributions is polynomial-sized (number of combinatorial compositions)

Theorem 2 *Computation of the Jacobian for symmetric action-graph games takes time that is* $O\left(poly(\overline{n}^{\mathcal{I}})poly(|S|)\right)$.

Speedup Results



One flavor ice cream vendor game; 10 stations



Given a 1 GFLOP computer, solve Jacobian for:

10 agents: GW ~1 hr;

1 hr: GW 10 agents;

Conclusions

- AGGs are a fully expressive compact representation for games
- They compactly express:
  - context-specific and/or strict utility independencies
  - anonymity in utility functions
- We leverage the AGG representation to compute Nash equilibria using a continuation method; guarantee
  - arbitrary equilibria: exponential speedup of continuation method
  - symmetric equilibria: bounded indegree implies polytime computation of Jacobian

google://"Navin Bhat"   google://"Kevin Leyton-Brown"