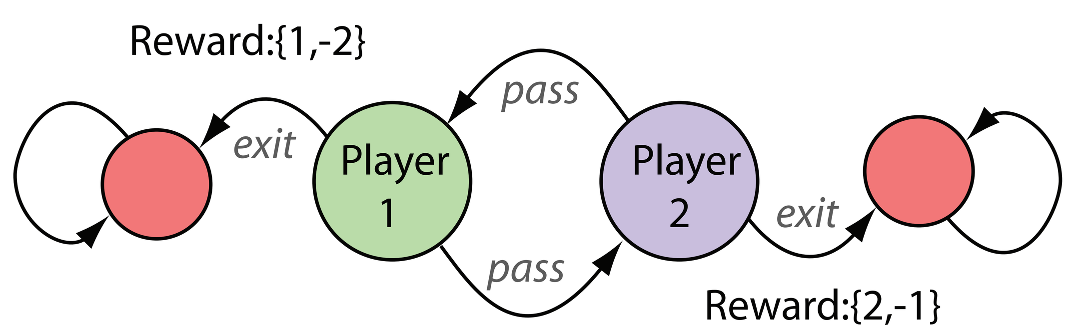
QPACE: Quick Polytope Approximation of All Correlated Equilibria in Stochastic Games

Liam MacDermed
Karthik Narayan
Charles Isbell
Lora Weiss

Key Ideas:

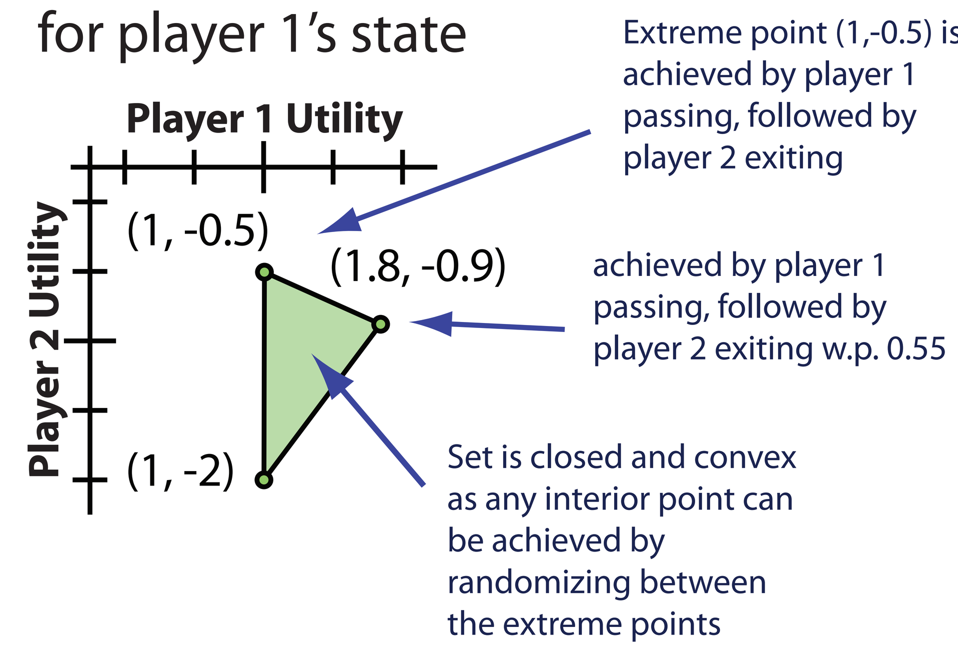
Solve multi-agent reinforcement learning by using achievable-sets instead of values (Murray & Gordon 2007).

An Example Game (The Breakup Game):

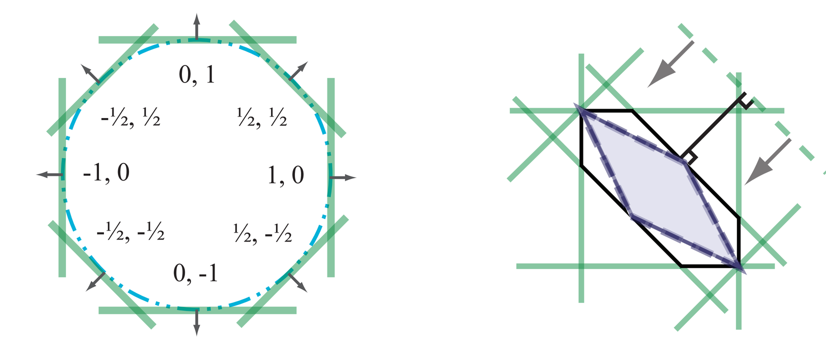


Circles represent states, outgoing arrows represent deterministic actions. Unspecified rewards are zero. Previous algorithms could not solve this game.

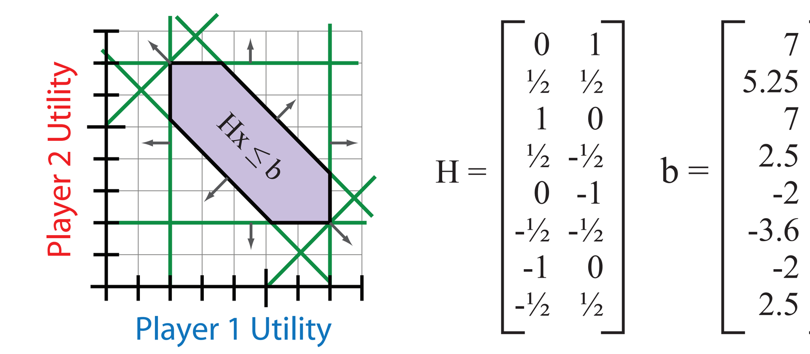
The final achievable-set for player 1's state



Approximate achievable-sets using a fixed collection of half-spaces (MacDermed & Isbell 2009).



Achievable sets are approximated using regular half-spaces with a fixed set of normals.



Offsets are contracted to construct an overestimate. Normals remain constant.

This polytope representation can be used to dramatically improve performance by (this paper):

- Eliminating vertex computation
- Permitting very fast Minkowski sums
- Preventing sets from changing dramatically, allowing linear program solution caching
- Transforming multi-objective LPs into a series of single-objective LPs

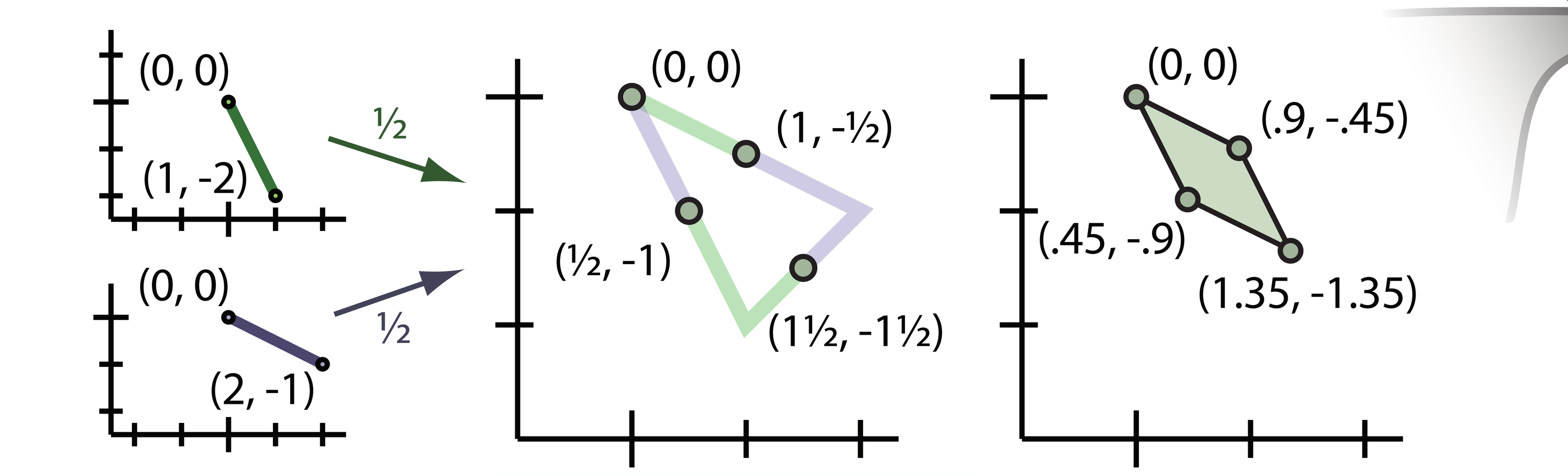
The tractable backup of achievable sets: (an iteration consists of a single backup for each state)

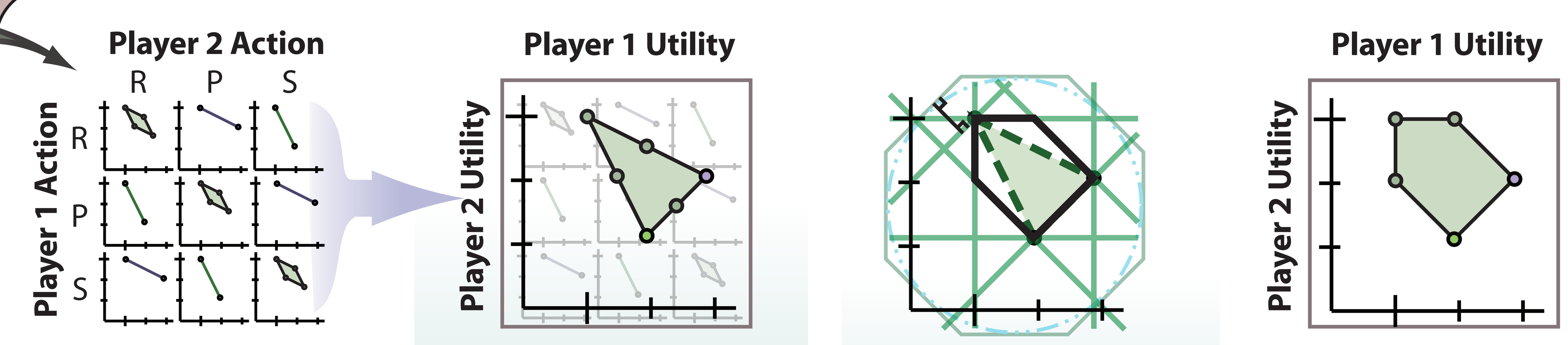
$$Q(s,\vec{a})_j = R(s,\vec{a})\cdot H_j + \gamma \sum_{s'} P(s'|s,\vec{a})V(s')_j$$

replaces 'max' in single agent RL

$$\mathbf{V}^*(s) = \underset{\vec{a}}{\text{equilibrium}}(\mathbf{Q}^*(s,\vec{a}))$$

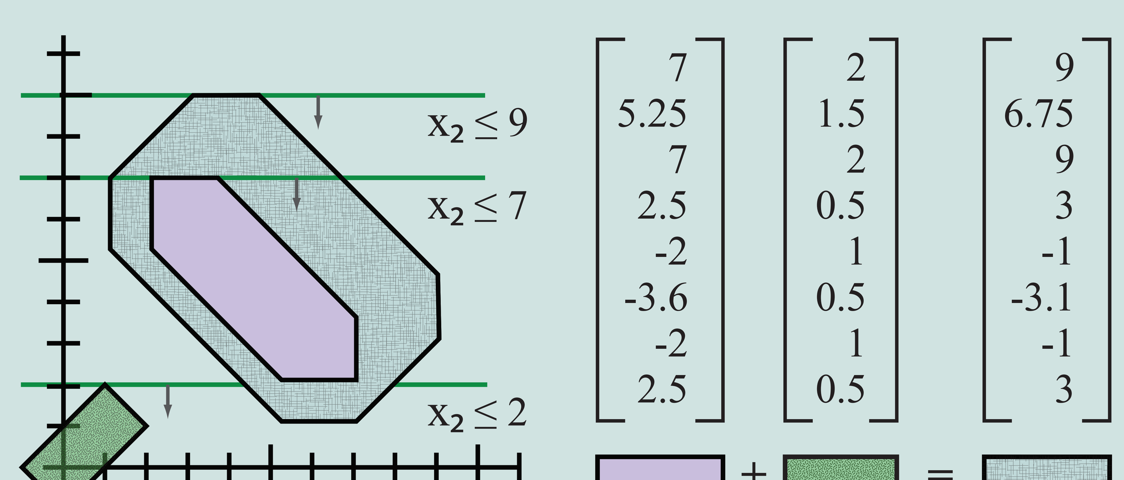
The state shown being calculated is an initial rock-paper-scissors game played to decide who goes first in the breakup game





Value Computed:	Achievable sets of successor states	Expected values of all policies	Achievable set of expected values	Achievable sets of all joint actions	Define the set of correlated equilibrium	Approximation Using Set of Hyperplanes	Final Achievable Set of initial state
	V(s')	(Weighted Minkowski Sum)	Q(s,a)	{Q(s,a)}	***equilibria***({Q*(s,a)})	$V(s)_j$	$V(s)$

Minkowski sums can be very efficiently computed using our regular polytope scheme



For each player i, distinct actions $\alpha, \beta \in A_i$,

$$\sum_{\vec{a}\in A^n} \overrightarrow{cu_{\vec{a}(\alpha)}}_i \geq \sum_{\vec{a}\in A^n} x_{\vec{a}(\alpha)}[\overrightarrow{gt_{\vec{a}(\beta)}}_i + R(s,\vec{a}^{(\beta)})_i]$$

$$\sum_{\vec{a}\in A^n} x_{\vec{a}} = 1 \text{ and } \forall \vec{a}\in A^n,\ x_{\vec{a}} \geq 0$$

For each joint-action $\vec{a}\in A^n$,

$$\overrightarrow{cu_{\vec{a}}} \in x_{\vec{a}}\ Q(s,\vec{a}) \quad (i.e.\quad H_j\ \overrightarrow{cu_{\vec{a}}} \leq x_{\vec{a}}\ Q(s,\vec{a})_j\ )$$

$$\max \sum_{\vec{a}} \sum_{i\in I} H_{j,i}\cdot \overrightarrow{cu_{\vec{a}i}}$$

subject to: these inequalities

Each LP changes slowly from iteration to iteration, granting very fast results when we start the LP from the previous solution

Results:

We provide the first approximation algorithm which solves stochastic games to within ε absolute error of the optimal game-theoretic solution, with running time polynomial in the error bound and the number of states and joint-action.