Computing Equilibria in Multi-Player Games

Christos H. Papadimitriou   Tim Roughgarden

UC Berkeley

ACM-SIAM Symposium on Discrete Algorithms 2005, Pages 82-91

Presented by Yiu-Cho Leung

on Apr 26, 2006 for course COMP670O

Overview

- A systematic study of algorithmic issues involved in finding Nash and correlated equilibria
- A polynomial-time algorithms for computing the correlated equilibria for symmetric game.
- A general framework for obtaining polynomial-time algorithms for optimizing over correlated equilibria
- Proving that such algorithms are not possible for some kind of games

Multi-player Games

Complexity of Computing Equilibria

- This paper studies the complexity of computing equilibria in games with many players.
- Immediate obstacle: massive input complexity
- $n2^n$ numbers are required to specify a general game for n players with binary decisions.
- Three games with compact representation will be studied:
  - Symmetric Games
  - Graphical Games
  - Congestion Games
- What properties of a compact game permit polynomial-time algorithms for computing equilibria?

Games with Compact Representation

Symmetric Games

- All players are identical and indistinguishable.
- They have same strategy sets and utility functions which depend only on the *number* of players choosing each strategy and the player's own strategy.
- For n players with k strategies, there are $\binom{n+k-1}{k-1}$ distinct distributions of n players among k strategies.
- The game can be summarized with only $k\binom{n+k-1}{k-1}$ numbers.

Games with Compact Representation

Graphical Games

- Players are vertices of a graph.
- The payoff of each player only depands on its strategy and those of its neighbors.
- There are polynomial-time algorithms for computing Nash and correlated equilibria for graphical games defined on tree.

Games with Compact Representation

Congestion Games

- There is a ground set of elements.
- Players choose a strategy from a prescribed collection of subset of the ground set.
- The cost of an element is a function of the number of players that select a strategy that contains it, but is independent of the identity of the player.
- The cost (negative payoff) to a player is then the sum of costs of elements in its strategy.

Preliminaries
Normal Form Game

- A normal form game is a collection of $S_1, \ldots, S_n$ of finite strategy sets and a collection of $u_1, \ldots, u_n$ of utility functions, each defined on $S_1 \times \cdots \times S_n$.
- A strategy set S_i and utility function u_i is identified by player i.
- An element s of $S_1 \times \cdots \times S_n$ is called strategy profile.
- The set of all strategy profiles is the *state space* of the game.
- For a strategy profile s, s_i is the strategy of player i and s_{-i} is the $(n-1)-$vector of strategies of players other han i.

Correlated Equilibria

Definition 1

Let $G = (\{S_i\}, \{u_i\})$ be an n-player game. Let q be a probability distribution on $S_1 \times \cdots \times S_n$ Distribution q is a correlated equilibria if for each player i and each pair l, l' of strategies is S_i,

$$\sum_{s:s_i=l} q(s)u_i(s) \geq \sum_{s:s_i=l} q(s)u_i(s'),$$

where s' is obtained from s by reassigning i's strategy to be l'.

Interpretation: A trusted authority picks a strategy profile s at random according to q, and recommends strategy s_i to each player i. Each player i is assumed to know only its recommended strategy and the other players will follow their recommendations. Then this conditional expectation should be maximized by the recommended strategy.

Symmetric Games with Two Strategies

More Definitions

- Let $G = (S = \{1, 2\}, u_1, \ldots, u_n)$ be an n-player, 2-strategy symmetric game.
- $S_i(j) \subseteq S^n$ denotes the subset of strategy profile which exactly j players, includeing player i, choose strategy 1.
- $S(j) \subseteq S^n$ denotes the subset of strategy profile which exactly j players choose strategy 1.
- $p_i(j)$ denotes the aggregated probability of the startegy profiles in $S_i(j)$ for $i \in \{1, \ldots, n\}$ and $j \in \{0, \ldots, n\}$.
- $p(j)$ denotes the aggregated probability of the startegy profiles in $S(j)$ for $j \in \{0, \ldots, n\}$.
- $u_i(j, l)$ denotes the payoff to player i when player i chooses strategy l and a total of j players choose strategy 1.

Symmetric Games with Two Strategies
Basic Linear System for Correlated Equilibria

$$\sum_{j=0}^{n} p_i(j) u_i(j,1) \geq \sum_{j=0}^{n} p_i(j) u_i(j-1,2) \quad (1)$$

$$\sum_{j=0}^{n} [p(j) - p_i(j)] u_i(j,2) \geq \sum_{j=0}^{n} [p(j) - p_i(j)] u_i(j+1,1) \quad (2)$$

$$\sum_{j=0}^{n} p(j) = 1 \quad (3)$$

$$\sum_{i=0}^{n} p_i(j) = j \cdot p(j) \quad (4)$$

$$0 \leq p_i(j) \leq p(j) \leq 1 \quad (5)$$

Correlated Equilibria of Symmetric Games

- Every correlated equilibrium of an n-player, 2-strategy symmetric game G induces a solution to G's basic linear system.
- p is said to extend to S^n if there is a function $q : S^n \rightarrow \mathcal{R}^+$ with $\sum_{s \in S_i(j)} q(s) = p_i(j)$ and $\sum_{s \in S(j)} q(s) = p(j)$.
- If p extends to S^n, the extension is a correlated equilibrium of G.
- However, it not obvious at all that such extension must exist.

Theorem 2

Let G be an n-player, 2-strategy symmetric game. Then every solution to G's basic linear system can be extended to a correlated equilibrium of G.

Correlated Equilibria of Symmetric Games

Definition 3

- A j-basic cover is a function $x : \{S_1(j), \ldots, S_n(j)\} \to \mathcal{R}^+$ with $\sum_{i:s \in S_i(j)} x_i(j) \geq j$ for all $s \in S(j)$, where $x_i(j)$ denotes $x(S_i(j))$.
- A solution p to G's basic linear system is uniform if for all $j \in \{0, \ldots, n\}$, $\sum_{i=1}^{n} p_i(j) x_i(j) \geq \sum_{i=1}^{n} p_i(j) = j \cdot p(j)$ for every j-basic cover x.
- For every $j \in \{0, \ldots, n\}$, a uniform j-cover is obtained by setting $x_i(j) = 1$ for all $i \in \{1, \ldots, n\}$.

Correlated Equilibria of Symmetric Games

Lemma 4

Let G be an n-player, 2-strategy symmetric game. Then every uniform solution to G's basic linear system can be extended to a correlated equilibrium of G.

Lemma 5

Let G be an n-player, 2-strategy symmetric game. Then every solution to G's basic linear system is uniform.

Proof of Lemma 5

- Let p be a solution to G and u be the uniform j-cover. Proving p is uniform is equivalent to showing that, for each $j \in \{0, \ldots, n\}$, u minimizes $\sum_{i=1}^{n} p_i(j)x_i(j)$ over all j-basic cover x.
- Let x be a non-uniform j-basic cover for some $j \in \{0, \ldots, n\}$. Let U be the set of indices underused by x (i.e. $x_i(j) < 1$) and O be the set of indices overused by x (i.e. $x_i(j) > 1$).
- W.L.O.G., let $x_1(j) \geq x_2(j) \geq \cdots \geq x_n(j)$, $O = \{1, \ldots, m\}$ and $U = \{t, \ldots, n\}$ for $1 \leq m < t \leq n$.

Proof of Lemma 5

- If s is an element in $S_t(j) \cap \cdots \cap S_n(j)$, the the contribution of U to the sum $\sum_{i:s\in S_i(j)} x_i(j)$ for s is $c = \sum_{i=t}^{n}(1 - x_i(j))$ less than in the uniform solution. This implies

$$\sum_{i:s\in S_i(j), i\leq m} [x_i(j) - 1] \geq c.$$

- If $s = \{n-j+1, n-j+2, \ldots, n\}$, then,

$$\sum_{i\leq r:s\in S_i(j)} 1 = [r + j - n]^+$$

where $[\alpha]^+ = \max(\alpha, 0)$

Proof of Lemma 5

- Let $z_r(j) = x_r(j) - x_{r+1}(j)$ for $r \in \{1, 2, \ldots, m-1\}$ and $z_m(j) = x_m(j) - 1$, then

$$\begin{aligned} c &\leq \sum_{i:s\in S_i(j), i\leq m} [x_i(j) - 1] \\ &= \sum_{i:s\in S_i(j), i\leq m} \sum_{r=i}^{m} z_r(j) \\ &= \sum_{r=1}^{m} \sum_{i\leq r:s\in S_i(j)} z_r(j) \\ &= \sum_{r=1}^{m} [r + j - n]^+ z_r(j) \end{aligned}$$

Proof of Lemma 5

- $$\sum_{i=1}^{n} p_i(j)x_i(j) = \sum_{i=1}^{n} p_i(j)u_i(j) + \sum_{i=1}^{m} p_i(j)[x_i(j)-1] - \sum_{i=t}^{n} p_i(j)[1-x_i(j)]$$

- $$\sum_{i=t}^{n} p_i(j)[1-x_i(j)] \leq p(j)\sum_{i=t}^{n}[1-x_i(j)] = c \cdot p(j)$$

Proof of Lemma 5

- $$\begin{aligned}\sum_{i=1}^{m} p_i(j)[x_i(j)-1] &= \sum_{i=1}^{m}\left(z_r(j)\sum_{i=1}^{r} p_i(j)\right)\\ &\geq \sum_{i=1}^{m} z_r(j)[r+j-n]^{+} p_i(j)\\ &\geq c\cdot p(j)\end{aligned}$$

A General Framework

Let $G = (S_1, \ldots, S_n, u_1, \ldots, u_n)$ be a game in normal form. For $i = 1, 2, \ldots, n$, Let $P_i = \{P_i^1, \ldots, P_i^{m_i}\}$ be a partition of S_{-i} into m_i classes.

- For a player i, two strategy profiles s and s' are i-equivalent if $s_i = s'_i$, and both s_{-i} and s'_{-i} belong to the same class of the partition P_i.
- The set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of partition is a compact representation of G if $u_i(s) = u_i(s')$ whenever s and s' are i-equivalent.

Separation Problem and Correlated Equilibria

Definition 6

Let $\mathcal{P}$ be a compact representation of a game G. The separation problem for $\mathcal{P}$ is that: Given rational numbers $y_i(j,l)$ for all i, j and $l \in S_i$, is there a strategy profile s with $\sum_{(i,j,l):s_i=l,s_{-i}\in P_i^j} y_i(j,l) < 0$?

Theorem 7

Let $\mathcal{P}$ be a compact representationof a game G. If the separation problem for $\mathcal{P}$ can be solved in polynomial time, then a correlated equilibrium of G can be computed in time polynomial in the size of $\mathcal{P}$.

Complexity of Computing Correlated Equilibria

for Compactly represented Game

Corollary 8

A correlated equilibrium of a symmetric game can be found in time polynomial in its natural compact representation.

Corollary 9

A correlated equilibrium of a graphical game with a tree topology can be found in time polynomial in its natural compact representation.

Complexity of Computing Correlated Equilibria
for Compactly represented Game

Corollary 10

There is no polynomial-time algorithm for computing a correlated equilibrium of a compactly represented graphical game.

Corollary 11

There is no polynomial-time algorithm for computing a correlated equilibrium of a compactly represented congestion game.

Symmetric Nash Equilibrium

Theorem 12

A symmetric Nash equilibrium in a symmetric game with n players and k strategies can be computed to arbitary precision in time polynomial in n^k, the number of bits required to described the utility functions, and the number of bits of precision desired.

Corollary 13

A Nash equilibrium of a compactly represented n-player k-strategy symmetric game with $k = O(\log n / \log \log n)$ can be computed in polynomial time.