

# Compact Game Representations: Reasoning and Computation

Reza Lotun  
Department of Computer Science  
University of British Columbia  
rlotun@cs.ubc.ca

## ABSTRACT

A proliferation of compact representation schemes have been proposed for various classes of games, along with efficient computations of solution concepts on such representations. The two main schemes have been Graphical games [7] and Multi-agent Influence Diagrams [14]. However, various other representations exist such as Game Networks [13] and Action-Graph Games [2]. This work will present a general overview of these different representation schemes and concentrate on two ideas associated with them; what kind of reasoning and visualization can be performed with each scheme, and what sort of computations can be made on them. We focus on correlated equilibria [1] calculations on such representations. The results of [12, 11], which provided a general framework for correlated equilibria computations on various suitable compact representations, will be extended to show a polynomial time algorithm to compute correlated equilibria on Action-Graph Games.

## Categories and Subject Descriptors

F.20 [Analysis of Algorithms and Problem Complexity]: General

## General Terms

Algorithms, Theory, Economics

## Keywords

Game Theory, Compact Representation, Graphical Models, Graphical Games, Computation of Nash Equilibria, Correlated Equilibria

## 1. INTRODUCTION

Much interest has recently been taking place in the computer science community, particular in the theory and artificial intelligence communities, on the intersection between mathematical economics and computer science. Game Theory, a mathematical model of strategic interactions between

a group of self-interested agents, has been touted by some [10] as the key idea in modelling computational problems relating to the Internet, and situations in AI.

The predominant solution concept of a game is a Nash equilibrium, a “stable point” of strategic choices such that all the agents or players in the game have no incentive to deviate from it. Interestingly, the complexity of computing Nash equilibria is largely not understood, and is regarded by Papadimitriou that “together with factoring, the complexity of finding a Nash equilibrium is the most important concrete open question on the boundary of  $P$  today” [10]. Indeed, even for the case of a two-player game, no polynomial time algorithm is known to compute a Nash equilibrium, though much work has recently taken place on trying to characterize its complexity [4, 12, 3] (see [9] for a general overview of early algorithms).

There are, however, other notions of equilibrium in games. Chief among these are correlated equilibria [1], a more general form of Nash equilibrium that has a simpler form amenable to linear programming. At a high level, a correlated equilibrium models the simplest form of cooperation that can arise in a game via “shared randomness”, and has been argued by many (see next section) that it is a more natural form of equilibrium than that of Nash. In any case, the understanding of its complexity is perhaps the first step in understanding that of the Nash equilibrium problem.

Consider an  $n$ -player game, where each player has to make a binary decision. Since we have to specify a utility for each of the  $2^n$  possible outcomes of the game for each player, we thus have  $n2^n$  numbers already. Thus, without even starting to compute or reason about this game, the “input” already is exponential in the number of players, even for a game with such a small number of actions per player. However, these are very artificial constructs - in *practice* what sort of games are there in “nature”, are they amenable to compact representations?

There are various schemes for representing games, and the situation is analogous to that of *probabilistic graphical models (PGMs)*. A PGM is a graphical representation of a joint probability distribution  $P(X_1, X_2, \dots, X_n)$ , where the nodes represent random variables, and edges between nodes represent whether two variables are independent of each other. If each of the  $X_i$ 's are considered to be discrete random vari-

ables, then as in the game theoretic case,  $2^n$  numbers need be provided and manipulated when performing calculations and storage. In the directed graph case we have *bayesian networks (bayes nets)*, where a directed link from  $X$  to  $Y$  means that  $Y$  is dependent on  $X$ , whereas in the undirected case we have a *markov network*. Both representations have revolutionized probability, statistics and machine learning in that they brought seeming intractable representation problems within computational reach. There are several trends from the PGM case that can be applied to the game theory case. Firstly, there are independency structures which can invariably be exploited in the real world. Although in the worst case, PGM representations are computationally intractable, in practice there is much structure in statistical modelling. For example, for some r.v.  $X$  in the bayes net case, we need only consider its probability conditioned on its parents and their children, since it is statistically independent from all other variables in the distribution. Likewise, there exists a graphical game representation [7] which models the nodes of the graphs as players, and the edges as relationships between players. Thus the neighborhood of a node (the player and everyone he is connected to) will have a local game matrix specifying the utilities of their subgame. That is, instead of representing a large matrix interaction of all players, essentially modelling the case where everyone plays against everyone else, we instead have local games where players play only with their surrounding environment. This representation method seems suitable for games with an extremely large amount of players - such as situations arising on the internet.

In the following sections we define the notions of game and correlated equilibrium, and study how we can represent games as Graphical Games [7] and Action-Graph Games [2], and show how to compute correlated equilibria on each representation.

## 2. PRELIMINARIES

A *game* consists of  $n$  players, a collection  $S_1, \dots, S_n$  finite strategy sets (actions) and a collection of  $u_1, \dots, u_n$  real-valued utility functions, where  $u_i : S_1 \times \dots \times S_n \mapsto \mathbb{R}, \forall i$ . An element  $s \in S = \prod_{i=1}^n S_i$  is called a *strategy profile*, where  $S$  is the set of all strategy profiles and is known as the *state space* for the game. Let  $n_s = \max_i |S_i|$ . Given a strategy profile  $s$  we let  $s_i$  denote the strategy for player  $i$  and  $s_{-i}$  the  $n - 1$  vector of strategies for the other  $n - 1$  players. If we let  $\phi(X)$  denote the set of all probability distributions over some set  $X$  then we let  $\Sigma_i = \phi(S_i)$  be the set of all *mixed strategies* for  $i$  and the set of all mixed strategy profiles  $\Sigma = \prod_{i=1}^n \Sigma_i$ .

A *correlated equilibrium (CE)* is a distribution  $q$  on  $S_1 \times \dots \times S_n$ , such that for each player  $i$  and each pair of strategies  $a, a'$  in  $S_i$ , so that  $s_i = a$  and  $s'_i = a'$ ,

$$\sum_{s_{-i}} q(s_i, s_{-i}) u_i(s_i, s_{-i}) \geq \sum_{s_{-i}} q(s_i, s_{-i}) u_i(s'_i, s_{-i}) \quad (1)$$

One possible interpretation of correlated equilibrium involves a third party trusted authority which knows the correlated equilibrium  $q$  for some game  $G$  and picks or samples a strategy profile  $s$  according to  $q$ , thereby "recommending" strategy  $s_i$  to each player  $i$ . Each player is assumed to know only their own recommended strategy, and not those of the other

players. A CE is thus achieved when no player has an incentive to deviate from their recommended strategy, assuming the other players will not deviate from their recommended strategies either. If the CE  $q$  is a product distribution, that is  $q = \prod_{i=1}^n p_i$ , where  $p_i \in \Sigma_i$ , then  $q$ , or rather all the  $p_i$ 's, constitute a *Nash equilibrium (NE)*, implying that every NE is also a CE. Thus we see that each player's distribution  $p_i$  over his strategy set  $S_i$  is independent of all other distributions - there is no correlation between player's decision making.

For a motivating example [11], consider the *chicken game* given in normal form,

		Player 2	
		S	G
Player 1	S	4, 4	1, 5
	G	5, 1	0, 0

The game models the classic daredevil game of drivers speeding on different streets towards some intersection, the  $S$  representing *stop* and the  $G$  representing *go*. Obviously if both choose  $G$ , the cars will crash resulting in 0 utility. If both choose  $S$  both will equally lose face but still be alive to explain away the incident. The interesting decision involves making the other driver exclusively lose face, resulting in low utility for the shamed driver, but very high utility for the victor. Any CE would be a probability distribution  $q$  over  $S_1 \times S_2 = \{SS, SG, GS, GG\}$ . Consider five such CE,

Distribution	SS	SG	GS	GG
$q_a$	0	1	0	0
$q_b$	0	0	1	0
$q_c$	1/4	1/4	1/4	1/4
$q_d$	0	1/2	1/2	0
$q_e$	1/3	1/3	1/3	0

The first two CE  $q_a$  and  $q_b$  are pure strategy NE, while  $q_c$  is a mixed strategy NE where  $p_1 = p_2 = \{1/2, 1/2\}$ . The last two  $q_d$  and  $q_e$  are not NE, and we can interpret  $q_d$  as being a *traffic light* - that is, a third party flips a fair coin and depending on the outcome suggests a strategy to each player, while  $q_e$  happens to be the CE that maximizes the expected sum of utilities for both players.

There is large body of literature advocating for a reexamination of the prime importance of Nash equilibria as solution concepts. A common question cited is why any player should assume the others are playing a Nash equilibria [5]. A possible answer could be that the assumption about how the other opponent plays could be the outcome of some learning process. Such learning would take place in repeated games, based on the history of past plays and using some sort of forecast mechanism to predict opponent strategies. There has been a large body of work on trying to find learning rules in repeated games that converge to nash equilibria, however such rules are complicated and even then convergence is not assured. For example in *fictitious play*, each player models all the other players through a histogram of some fixed mixed strategy, and plays best response to that. Fictitious play *will* converge to NE in zero sum games, but not in the general case. However, in another variant where a player

switches from the current strategy to another with probability proportional to the player's *regret* for having played the present strategy instead of the other in the past, *does* converge to a CE in general [11, 5]. Also, it has been argued that NE are not compatible with the bayesian perspective. In bayesian models, each player would have a prior to model each opponent - a NE suggests that each player should choose a *particular* prior, prompting many, for example [5], to suggest CE as an equilibrium concept compatible with the bayesian perspective. This work on learning strategies suggests that CE may be the more natural form of solution concept to games.

### 3. REPRESENTATIONS

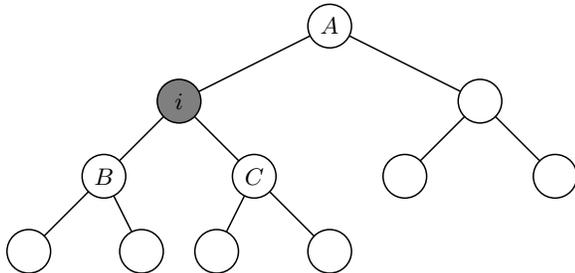
The following section will define and present examples of two forms of compact representations, graphical games, which represent nodes as *players* and edges as local interactions, and action-graph games, which represent nodes as *actions*.

#### 3.1 Graphical Games

A *graphical game* is a tuple  $(G, \{M'_i\})$ , where  $G = (V, E)$  is an undirected graph, where the vertices  $V$  represent players and the edges  $E$  represent local interactions amongst players. The  $M'_i$  are local payoff matrices defined for each node on the graph - they intuitively model the local game that a player  $i$  plays with all those in his neighborhood  $N(i)$ . Thus we see that if we let the *maximum degree of local interaction*  $k = \max_i |N(i)|$ , and assume binary actions, then our representation size becomes  $O(n2^k)$ , that is, it is exponential in the max degree of the graph. Note that when the graph is a clique (that is, it is fully connected), we recover normal form representation.

The problem of computing equilibria is still non-trivial, since if the graph is connected, the strategy of a player "affects" all other players (indirectly, via best response). If we let  $X, Y$  be subsets of vertices (players) so that  $X$  and  $Y$  are disconnected in  $G$ , then it's obvious  $X, Y$  form *independent games*. For every player  $i$ , if  $N(i) - i$  has their strategies fixed, then two independent subgames arise -  $i$  by himself, and all non-neighbors of  $i$ . In general, if we let  $SE$  be the set of players that "seperates" the remaining set of players into two non-empty subsets  $X$  and  $Y$ , then if we fix the strategies in  $SE$ , the subgames and conditional equilibrium for players in  $X$  is independent of players in  $Y$ .

For example, in the graphical game below, if we condition (fix strategy) of player  $i$  so that  $SE = \{i\}$ , we get two globally independent games, the unshaded nodes rooted at  $A$  and the set of two unshaded subtrees rooted at  $B, C$  as well as the two sub-local games each rooted at  $B$  and  $C$ .



In [7] conditioning on seperation sets and dynamic program-

ming to implement message passing are used to give polynomial approximation and exact algorithms to compute NE in tree games.

In [6] an algorithm is given to compute correlated equilibria in tree graphical games in polynomial time. We briefly survey these results, as we introduce a more general method based on linear programming in a later section. Two distributions  $p$  and  $q$  over  $S_1 \times \dots \times S_n$  are said to be *expected payoff equivalent (EPE)* if the expected payoff to all players is the same under both distributions. Also two distributions  $p$  and  $q$  with respect to a graph  $G$  are *local neighborhood equivalent (LNE)* if both have the same distribution over joint actions of the local neighborhood for each player. Note also that LNE implies EPE. For a graphical game, any CE  $q$  can be represented in  $G$  as a distribution  $p$  such that  $p$  is a local markov network (a special case of a general markov network that uses the neighborhood of each node  $i$  instead of the maximal cliques in the graph) in  $G$  and  $p, q$  are EPE. A variant of linear programming is then used on the local markov net to calculate a CE.

#### 3.2 Action-Graph Games

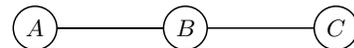
An *Action-Graph Game (AGG)* [2] is a tuple  $(N, S, A, \nu, u)$ , where  $N$  is the set of  $n$  agents,  $S = S_1 \times \dots \times S_n$ ,  $A = \bigcup_{i=1}^n S_i$  is the set of distinct action choices,  $\nu : S \mapsto 2^S$  is the neighbor relation, and  $u : S \times \Delta \mapsto \mathbb{R}$ , is the utility function, where  $\Delta$  is the set of distributions of *numbers* of agents over distinct actions - thus  $u$  depends only on the number of agents who take neighbouring actions. All agents have the same utility function. Explicitly, given any action  $a \in A$  and any pair of distributions  $D, D' \in \Delta$ ,  $\forall a' \in \nu(a)$ ,

$$[D(a') = D'(a')] \implies [u(a, D) = u(a, D')] \quad (2)$$

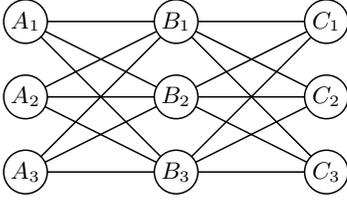
That is, for any pair of players  $i, j$  (notice how the players aren't explicitly represented) the utilities for  $i, j$  are independent, as long as the actions chosen by them aren't in the same neighborhood. This is an example of *context specific independence* and can be contrasted with the graphical model approach where independencies are implicit in the graph structure. In AGG's, depending on the action choices of the players, independencies can emerge.

Graphical games can be converted to AGGs by replacing each node  $i$  in the graphical game with a distinct cluster of nodes representing all the actions in  $S_i$ . For every edge  $(i, j)$  in the graphical game, create edges for all  $s_i \in S_i$  to all  $s_j \in S_j$ .

Consider the following graphical game consisting of three players each with strategy set of cardinality three, such that if we condition on player  $B$ , then player  $A$ 's utility is independent of player  $C$ 's actions.



The corresponding AGG is then



where we've replaced each player node with a set of actions. Notice the graph is 3-partite. The above example shows a case where the action sets for the three players are *distinct*. When players have actions in common, even more compact representations may arise. There are a number of games that can be represented compactly in AGG's but not in graphical games, see [2].

### 3.3 Other Representations

Besides graphical games and action-graph games, much early work has been on representations descended from *influence diagrams* from decision theory. *Multi-agent influence diagrams (MAIDs)* [14] are types of graphs where nodes come in three varieties: *chance*, *decision*, and *utility* variables. Depending on the parents of each node, utilities and information sets can be represented compactly. *Game networks* [13] are another scheme where nodes represent actions and two type of edges are allowed, representing causal and preferential dependencies.

## 4. COMPUTING CE

Consider again the expression for correlated equilibria (1). We have an expression for every player and every pair of strategies, giving  $O(n \cdot n_s^2)$  such expressions. Rearranging we get the following constraints,

$$\sum_{s_{-i}} [u_i(s_i, s_{-i}) - u_i(s'_i, s_{-i})] q(s_i, s_{-i}) \geq 0 \quad (3)$$

Since  $q$  is a probability distribution, represented here as a vector of length  $|S|$ , and it is in a linear combination with coefficients being the difference of two real valued utility functions, the above can be interpreted as part of a *linear program*,

$$\max \sum_{j=1}^{|S|} q_j \quad (4)$$

$$Uq \geq 0 \quad (5)$$

$$q \geq 0 \quad (6)$$

where (5) is written in standard LP form, corresponding to the fact that  $\sum_{j=1}^{|S|} q_j = 1$  since  $q$  is a probability distribution, and  $U$  corresponds to (3).

Naively, we may think that by establishing an LP formulation of the CE problem, we have found a polynomial solution to it, since any LP problem is solvable in polynomial time. The subtlety here is that the *input is exponential* - that is, though there are a polynomial number of constraints  $O(n \cdot n_s^2)$ , there are an exponential number of variables  $|S| \in O(s_n^n)$ .

It seems we are at an impasse. However, in [11] the *dual* of the above LP (the original LP hereafter denoted as the *primal*),

$$U^T y \leq -1 \quad (7)$$

$$y \geq 0 \quad (8)$$

is considered and is used to provide a constructive proof for the existence of CE in every game (without relying, as is usually the case, on Nash's theorem to prove their existence). The primal is conjectured to have a CE  $q$  with unbounded feasibility region (it cannot be enclosed by a high-dimension analogue of a circle), and thus by the *fundamental duality theorem* of LP, the dual must be infeasible. A key lemma in [11] states that  $\exists q$  so that  $q$  is a product distribution such that  $qU^T y = 0$ . So the constructive proof for the existence of CE pits some polynomial LP algorithm such as the *ellipsoid algorithm* on the dual. Since the dual now has a polynomial number of variables and exponential number of constraints it is guaranteed to run in polynomial time. After the ellipsoid algorithm running on the dual terminates, we say at the  $L$ th step, we have  $L$  product distributions  $q_1, \dots, q_L$  such that, for each  $1 \leq j \leq L$ ,  $qU^T y \leq -1$  is violated by  $y_j$ . Thus  $QU^T y \leq -1$ , where  $Q$  is the matrix whose rows are the  $q_j$ 's. The dual to *this* program  $[UX^T]\alpha \geq 0, \alpha \geq 0$  is unbounded. This nonzero  $\alpha$  vector will give us our final CE, which is a convex combination of the  $q_j$  product distributions that satisfy the primal.

Thus to summarize, we run the ellipsoid algorithm on the dual program and extend the algorithm to  $M = \text{poly}(n \cdot n_s^2)$  steps (the exact expression is derived in [11]), leading to an  $M \times O(n \cdot n_s^2)$  infeasible system  $QU^T y \leq \frac{1}{2}, y \geq 0$  (the  $\frac{1}{2}$  here arising from a numerical error correction made in the original proof in [11]). Call this program  $D'$ . The dual of  $D'$  is an  $O(n \cdot n_s^2) \times M$  system  $UQ^T \alpha \geq 0, \alpha \geq 0$ . Call *this* program  $P'$ . Since  $Q^T \alpha$  is a solution of the original primal, it is a CE.

The subtlety here rests in one step - the construction of the  $UQ^T$  needed for the solution of the  $P'$  program. The issue is that  $U$  is a  $O(n \cdot n_s^2) \times O(s_n^n)$  matrix, and  $Q^T$  is a  $O(s_n^n) \times M$ . The product of these matrices will not take polynomial time in general. At last however, we arrive to the subject of this work - because for a certain class of compact games, this product *will* be in polynomial time.

### 4.1 Succinct Games

A *succinct game* [11]  $G = (I, T, U)$  is defined in terms a set of inputs  $I$  and two polynomial algorithms  $T$  and  $U$ . Conceptually  $T$  gives the *type* of the game, returning the number of players as well as a  $n$ -tuple  $(t_1, \dots, t_n)$  which give the cardinalities of the strategy set for each player, while  $U$  is a utility function for the game. The  $I$  represents a compact representation of a games, such as a graphical game or action-graph game. If  $n$  and the  $t_i$ 's are bounded in  $|z|, z \in I$ , then the game is said to be of *polynomial type*. Furthermore, we say that  $G$  has the *polynomial expectation property* if there is a polynomial algorithm  $E$  which, given an input representation, a player and a product distribution over  $S$ , returns the expectation of the utility under the product distribution,

$$E(z, i, q) = \mathbb{E}_q[u_i(s), s \in S] \quad (9)$$

In [11] it was shown that any representation scheme that can be shown to be a succinct game with polynomial type and have the polynomial expectation property, then there exists a polynomial time algorithm which computes a CE for it (this polynomial expectation property is precisely what we need to circumvent the above  $UQ^T$  matrix product, since this product is in fact a calculation of the expected utility given the product distributions in  $Q^T$ ). A number of games were shown to have these properties, such as polymatrix games, graphical games, hypergraphical games, congestion games, local effect games, scheduling games, facility location games, network design games, and symmetric games. Below we give a description of how graphical games are succinct with the polynomial expectation property, and thereafter extend the results to action-graph games.

#### 4.1.1 Graphical Games

It is easy to see that graphical games are succinct. An input  $z \in I$  is an undirected graph  $G$  and for each player  $i$  an game matrix  $M_i$  explicitly represented with players in  $N(i)$ . Assuming low maximum degree,  $T$  can be calculated in polynomial time by going through each node in the graph and keeping appropriate statistics, as can  $U$  in a similar fashion.

All such  $z$  thus have the polynomial expectation property because given  $z, i, q_1, \dots, q_n, E$  will look at  $N(i)$  and go over every strategy profile in the local matrix, ignoring the mixed strategies not in  $N(i)$ , thus giving a polynomial procedure.

#### 4.1.2 Action-Graph Games

In [11] a *subset* of AGGs were shown to be succinct and have the polynomial expectation property, namely *local effect games* [8]. Again, the problem is that of computing

$$\sum_s u_i(s, D(s))q(s) \quad (10)$$

where  $q$  is distributed according to some product distribution  $q_1, \dots, q_n$ . For each  $a \in A$  and each player  $i$ , as in [11] we let  $w_i(a) = \sum_{a \in s \in S_i} q_i^s$ , where  $q_i^s$  is the  $i$ 's probability for choosing strategy  $s$ . Thus  $w_i(a)$  is the probability that player  $i$  will choose a strategy containing  $a$ . This can be done in polynomial time. Next, we would like to compute the expected number of players who would choose strategy  $a$  according to the  $q_i$ 's. We can do this by dynamic programming using the  $w_i$ 's as explained in [11]. By linearity of expectation we can combine everything to calculate the expected utility above.

## 5. CONCLUSION

In this paper we have demonstrated the importance of representation in reasoning and computing problems in game theory. After first introducing correlated equilibria and justifying why it is important, we introduced two compact representations - graphical games and action-graph games. We then showed how to calculate correlated equilibria on both in polynomial time, given certain instances of each representation. In the future more more work could be spent on trying to elucidate the relationship between graphical games and action graph games, for example algorithms to convert an AGG to a graphical game. More work on studying what sort of games are suitable for each representation can be made to make it easier to determine which is more appropriate.

## 6. REFERENCES

- [1] R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1, 1973.
- [2] N. Bhat and K. Leyton-Brown. Computing nash equilibria of action-graph games. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.
- [3] B. Blum, C. R. Shelton, and D. Koller. A continuation method for nash equilibria in structured games. In *IJCAI03: International Joint Conference on Artificial Intelligence*, 2003.
- [4] V. Conitzer and T. Sandholm. Complexity results about nash equilibria. In *IJCAI03: International Joint Conference on Artificial Intelligence*, 2003.
- [5] D. P. Foster and R. V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behaviour*, 21, 1997.
- [6] S. Kakade, M. Kearns, J. Langford, and L. Ortiz. Correlated equilibria in graphical games. In *EC '03: Proceedings of the 4th ACM conference on Electronic commerce*, pages 42–47, New York, NY, USA, 2003. ACM Press.
- [7] M. J. Kearns, M. L. Littman, and S. P. Singh. Graphical models for game theory. In *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 253–260, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [8] K. Leyton-Brown and M. Tenenholz. Local-effect games. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2003.
- [9] R. D. McKelvey and A. McLennan. Computation of equilibria in finite games. In H. Amman, D. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 1, pages 87–142. Elsevier, 1996.
- [10] C. H. Papadimitriou. Algorithms, games, and the internet. In *STOC01: 33rd ACM Symposium on the Theory of Computing*, 2001.
- [11] C. H. Papadimitriou. Computing correlated equilibria in multi-player games. In *STOC05: 37th ACM Symposium on the Theory of Computing*, 2005.
- [12] C. H. Papadimitriou and T. Roughgarden. Computing equilibria in multi-player games. In *SODA'05: ACM-SIAM Symposium on Discrete Algorithms*, 2005.
- [13] L. M. Pierfrancesco. Game networks. In *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 335–342. Morgan Kaufmann Publishers, 2000.
- [14] D. Vickrey and D. Koller. Multi-agent algorithms for solving graphical games. In *Eighteenth national conference on Artificial intelligence*, pages 345–351, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.