On the Complexity of Nash Equilibria of Action-Graph Games

Constantinos Daskalakis[*] Grant Schoenebeck[†] Gregory Valiant[‡] Paul Valiant [§]

Abstract

In light of much recent interest in finding a model of multi-player multi-action games that allows for efficient computation of Nash equilibria yet remains as expressive as possible, we investigate the computational complexity of Nash equilibria in the recently proposed model of *action-graph games* (AGGs). AGGs, introduced by Bhat and Leyton-Brown, are succinct representations of games that encapsulate both local dependencies as in graphical games, and partial indifference to other agents' identities as in anonymous games, which occur in many natural settings such as financial markets. This is achieved by specifying a graph on the set of actions, so that the payoff of an agent for selecting a strategy depends only on the number of agents playing each of the neighboring strategies in the action graph. We present a simple Fully Polynomial Time Approximation Scheme for computing mixed Nash equilibria of AGGs with constant degree, constant treewidth and a constant number of agent types (but an arbitrary number of strategies), and extend this algorithm to a broader set of instances. However, the main results of this paper are negative, showing that when either of the latter conditions are relaxed the problem becomes intractable. In particular, we show that even if the action graph is a tree but the number of agent-types is unconstrained, it is NP–complete to decide the existence of a pure-strategy Nash equilibrium and PPAD–complete to compute a mixed Nash equilibrium (even an approximate one). Similarly for AGGs with a constant number of agent types but unconstrained treewidth. These hardness results suggest that, in some sense, our FPTAS is as strong a positive result as one can expect. In the broader context of trying to pin down the boundary where the equilibria of multi-player games can be computed efficiently, these results complement recent hardness results for graphical games and algorithmic results for anonymous games.

1 Introduction

What is the likely behavior of autonomous agents in a variety of competitive environments? This question has been the motivation for much of economic theory. Partly due to the increasing prevalence of vast online networks over which millions of individuals exchange information, goods, and services, and the corresponding increasing importance of understanding the dynamics of such interactions, the Computer Science community has joined in the effort of studying game-theoretic questions.

Computing equilibria in games and markets has been extensively studied in the Economics and Operations Research literatures since the 1960's, see e.g. [23, 27, 32, 28, 30]. Computational tractability has been recently recognized as an important prerequisite for modeling competitive environments and measuring the plausibility of solution concepts in Economics: if finding an equilibrium is computationally intractable, should we believe that it naturally arises? And, is it plausible that markets converge to solutions of computationally intractable problems? Probably not — but if so, the Computer Science community should certainly know about it.

Computing Nash equilibria in games, even in the case of two players, has been recently shown to be an intractable problem; in particular, it was shown to be complete for the class of fixed point computation problems termed PPAD [10, 4]. This result on the intractability of computing Nash equilibria has sparked considerable effort to find efficient algorithms for approximating such equilibria, and has increased the importance of considering special classes of games for which Nash equilibria might be efficiently computable.

For two-player games the hardness of computing approximate equilibria persists even if the required approximation is inverse polynomial in the number of strategies of the game [5]; similarly, hardness persists in graphical games if the required approximation is inverse polynomial in the number of players [10, 5]. The same hardness results apply to special cases of the problem, e.g. win-lose games, where the payoff values of the game are restricted to $\{0, 1\}$ [1, 8, 6], sparse two-player games, where the number of non-zero entries of each row an column of the payoff matrices is a constant, and two-

[*] Microsoft Research. This research was done while the author was a student at U.C. Berkeley. Supported by a Microsoft Research Fellowship. Email: costis@cs.berkeley.edu.

[†] U.C. Berkeley. Work supported under an NSF Graduate Research Fellowship. Email: grant@eecs.berkeley.edu.

[‡] U.C. Berkeley. Work supported under an NSF Graduate Research Fellowship. Email: gvaliant@cs.berkeley.edu.

[§] MIT. This research is supported in part by NSF grant 6916481. Email: pvaliant@mit.edu.

player symmetric games [17]. The emerging question of the research in this field is: Is there a Polynomial Time Approximation Scheme (PTAS) for Computing Nash Equilibria? And, which special cases of the problem are computationally tractable?

The zero-sum two-player case is well-known to be tractable by reduction to linear programming [31]. Tractability persists in the case of low-rank two-player games, in which the sum $A + B$ of the payoff matrices of the players, instead of being 0, has fixed rank; in this case, a PTAS exists for finding mixed Nash equilibria [21]. In n-player graphical games, a PTAS has been provided if the tree-width is $O(\log n)$ and the maximum degree is bounded [11]; in the case of dense graphical games, a quasi-polynomial time approximation scheme exists [13].

An important line of research on tractable special cases explores games with symmetries. Multi-player symmetric games with (almost) logarithmic number of strategies per player can be solved exactly in polynomial time by reduction to the existential theory of reals [26]. For congestion games, a pure Nash equilibrium can be computed in polynomial time if the game is a symmetric network congestion game [16], and an approximate pure Nash equilibrium can be found if the game is symmetric but not necessarily a network game, and the utilities satisfy a “bounded-jump condition” [7]. Another important class of games for which computing an approximate equilibrium is tractable is the class of anonymous games, in which each player is different, but does not care about the identities of the other players, as happens —for example— in congestion games, certain auction settings, and social phenomena [3]; a PTAS for anonymous games with a fixed number of strategies has been provided in [12, 13]. For a thorough study of the problem of computing pure Nash equilibria in symmetric and anonymous games see [15].

In this paper we continue the effort to pin down the tractability of computing equilibria for meaningful classes of games, considering the class of *Action Graph Games* (AGGs). Introduced by Bhat and Leyton-Brown [2] (see Definition 1.1), AGGs are a fully general game representation that succinctly captures both ‘local’ dependencies as in graphical games, as well as partial indifference to other agents’ identities as in anonymous games. The set of strategies each of the players may play is represented as a set of nodes in a graph, called the *strategy graph*, where the strategy sets may be disjoint, or overlapping. The game is *anonymous* to the extent that the utility of a player depends only on his action and the number of players playing each strategy, regardless of their identities. More specifically, each player’s utility only depends on the number of players playing strategies *adjacent* to his action, in the strategy graph. The only attribute that distinguishes players is the set of strategies that each player is allowed to play. In particular, all agents who play a given strategy get the same payoff. Note that AGGs are fundamentally different than Graphical Games in that the nodes in graphical games represent the agents rather than the strategies as in AGGs (see [22] for an introduction to graphical games). A variety of natural games can be concisely represented as AGGs including models of financial markets, and we refer the reader to [2, 19] for further discussion.

In the remainder of this section, we discuss previous work on AGGs and summarize our results. At the end of the section, we provide definitions.

1.1 Previous Work

Action graph games were first defined by Bhat and Leyton-Brown [2] who considered the problem of computing Nash equilibria of these games. In particular, they analyzed the complexity of computing the Jacobian of the payoff function—a computation that is, in practice, the bottleneck of the continuation method of computing a Nash equilibrium. They considered this computation for both general AGGs and AGGs with a single player type (symmetric AGGs), and found that this computation is efficient in the latter case. In [18], Jiang and Leyton-Brown describe a polynomial-time algorithm for computing expected utilities of an AGG. For pure Nash equilibria, Jiang and Leyton-Brown [19] show that deciding the existence of such equilibria in AGGs is **NP**-complete, even in the case of a single player type and bounded degree. On the other hand, they provide a polynomial time algorithm for finding pure-Nash equilibria in AGGs with constant number of player types when the strategy graph has bounded treewidth. In [14] Dunkel and Schultz show hardness results for computing pure-Nash equilibria for a special case of AGGs, called Local-Effect Games, in which the utility for playing an action can be decomposed into a sum of edge-wise utility functions, one for every adjacent edge. Their results are incomparable to ours.

1.2 Our Results

We examine, and largely resolve the computational complexity of computing Nash equilibria in action graph games. We give a fully polynomial algorithm for computing an ϵ-Nash equilibrium for AGGs with constant degree, constant treewidth and a constant number of agent types (and arbitrarily many strategies), together with hardness results for the cases when either the treewidth or the number of agent types is unconstrained. In particular, we show that even if the strategy graph is a tree with bounded degree but the number of agent types is unconstrained, it is NP–complete to decide the existence of a pure-strategy Nash equilibrium and PPAD–complete to compute a mixed Nash equilibrium; similarly for AGGs in which there are a constant number (10) of player types if we allow the strategy graph to have arbitrary treewidth. These hardness results suggest that, in some sense, our FPTAS is as strong of a positive result as one can expect. While Bhat and Leyton-Brown studied heuristics for computing mixed Nash equilibria [2], there are few complexity theoretic results concerning mixed Nash equilibria for AGGs—**PPAD**-hardness follows from AGGs being a generalization of normal-form games, and membership in **PPAD** which follows from the nontrivial fact that computing a Nash equilibrium of an AGG can be efficiently reduced to the problem of computing a Nash equilibrium of a graphical game, which was also noted in [20].

1.3 Definitions

In this section we give a formal definition of AGGs and introduce the terminology that will be used in the remainder of this paper. We follow the notation and terminology introduced in [19].

Definition 1.1. *An* action-graph game*, A, is a tuple $\langle P, \mathbf{S}, G, u\rangle$ where*

- *$P := \{1, \dots, n\}$ is the set of agents.*
- *$\mathbf{S} := (S_1, \dots, S_n)$, where S_i denotes the set of pure strategies that agent i may play.*
- *For convenience, let $S := \bigcup_i S_i = \{s_1, \dots, s_{|S|}\}$ denote the set of all strategies, and thus each $S_i \subseteq S$. Also, we write $S_i = \{s_{i,1}, s_{i,2}, \dots, s_{i,|S_i|}\}$ Furthermore, we'll let $s(i)$ denote the strategy played by agent i.*
- *For any $S' \subset S$, let $\Delta(S')$ denote the set of valid configurations of agents to strategies $s \in S'$; we represent a configuration $D(S') \in \Delta(S')$ as an $|S'|$-tuple $D(S') = \{n_1, \dots, n_{|S'|}\}$ where n_i is the number of agents playing the i^{th} strategy of S'.*
- *G is a directed graph with one node for each action s_i. Let $\nu : S \to 2^S$, be the neighbor relation induced by graph G, where $s' \in \nu(s)$ if the edge $(s', s) \in G$. Note that self-loops are allowed, and thus it is possible that $s \in \nu(s)$. We refer to G as the* strategy graph *of A.*
- *The utility function u assigns identical utilities to all agents playing a given strategy s, with the utility depending only on the number of agents playing neighboring strategies. Formally, $u : \Delta(S) \to \mathbb{R}^{|S|}$, via maps $u_1, \dots, u_{|S|}$ where $u_i : \Delta[\nu(s_i)] \to \mathbb{R}$ defines the common utility of all agents playing strategy s_i.*

Note that AGGs are fully expressive because any games can be written as an action graph game in which the strategy sets of different players are disjoint, and the strategy graph G is complete.

We now define a further type of possible symmetry between agents that will be important in our analysis of the complexity of computing Nash equilibria.

Definition 1.2. *We say that an AGG has k* player types *if there exists a partition of the agents into k sets $P_1, \dots, P_k$, such that if $p, p' \in P_i$, then $S_p = S_{p'}$. (The terminology of [19] refers to such games as k-symmetric AGGs.)*

Since agents who play the same strategy receive the same utility, all agents of a given type are identical—for example an AGG with a single player type is a symmetric game. While the number of player types does not significantly alter the description size, decreasing the number of player types adds structure to the space of possible Nash equilibria; this is the motivation for considering AGGs with few player types as a possible class of tractable games.

A *strategy profile*, $M := [m_1, \dots, m_n]$, with $m_i = (p_{i,1}, \dots, p_{i,|S_i|})$ assigns to each agent a probability distribution over the possible strategies that the agent may play, with $\Pr[s(i) = s_{i,k}] = p_{i,k}$ where $s_{i,k}$ is the k^{th} element of S_i. Thus a given strategy profile induces an expected utility for each player $\mathbb{E}[u|M] = \sum_{D \in \Delta} u(D) \Pr(D)$, where the probability is with respect to the strategy profile M.

Definition 1.3. *A strategy profile M is a* Nash-equilibrium *if no player can increase her expected utility by changing her strategy m_i given the strategy profiles m_{-i} of the other agents. That is, for all strategy profiles m_i', $\mathbb{E}[u_i|m_{-i}, m_i] \geq \mathbb{E}[u_i|m_{-i}, m_i']$.*

Definition 1.4. *A strategy $m \in M$ is an* ϵ-Nash-equilibrium *if no player can increase her expected utility by more than ϵ by changing her strategy profile.*

Note that there is the slightly stronger definition of an ϵ–Nash equilibrium in which, for all agents i, the expected utility of playing every strategy s in the support of m_i is at most ϵ less than the expected utility of playing a different $s' \in S_i$. We do not stress the distinction, as our FPTAS finds such an ϵ–Nash equilibrium, and our hardness results apply to the weaker definition given above.

2 FPTAS

Action graph games have properties of both anonymous games and graphical games. As such, one might expect that classes of AGGs that resemble tractable classes of anonymous or graphical games could have efficiently computable equilibria. For anonymous games, the symmetry imposed by the limited number of types implies the existence of a highly symmetric mixed equilibrium which seems easier to find than asymmetric equilibria. For graphical games with small treewidth, the tree structure allows for an efficient message-passing dynamic-programming approach. In line with this intuition, we give an FPTAS for computing ϵ-Nash equilibria for the class of AGGs that has both player symmetries and a tree-like graph structure. While these conditions might seem strong, we show in Section 3 that if either condition is omitted the problem of computing an ϵ-Nash equilibrium is hard.

The following theorem both motivates, and is implied by the stronger Theorem 2.2, which we state at the end of this section.

Theorem 2.1. *For any fixed constants d, k, and t, an AGG A with k player types and strategy graph G_A with bounded degree d and treewidth t, an ϵ-Nash equilibrium can be computed in time polynomial in $|A|, 1/\epsilon, n$.*

For clarity, we begin by outlining the key components of our simple dynamic-programming based FPTAS for the case that there is a single player type, and the action graph is a tree. These ideas generalize easily to the case that there are a constant number of player types and the action graph has a constant treewidth. Finally, we describe a larger class of AGGs for which a modified version of our FPTAS holds (Theorem 2.2). Due to space limitations, we defer the main proofs of this section to the full version.

We begin with a fact about games with few player types.

Fact 2.1. *[24] Any AGG with k player types has a Nash equilibrium where all players of a given type play identical mixed strategy profiles. Formally, there is a strategy profile $M = [m_1, \ldots, m_n]$ such that if $S_i = S_j$, then $m_i = m_j$. We refer to such equilibria as* type-symmetric *equilibria.*

The high-level outline of the FPTAS is as follows: we discretize the space of mixed strategy profiles such that each player may play a given strategy with probability $N\delta$ for $N \in \mathbb{N}$, and some fixed $\delta > 0$ that will depend on ϵ and n. We also discretize the space of target expected utilities into the set $V = \{0, \epsilon/2, \epsilon, \ldots, 1\}$. Then, for each $i \in \{0, \ldots, |V|\}$, starting from the leaves of the strategy-graph tree, we employ dynamic programming to efficiently search the discretized strategy space for a type-symmetric ϵ-Nash equilibrium in which each strategy in the support has an expected utility close to v_i. To accomplish this we associate to each strategy s_i a polynomially sized table expressing the set of probabilities with which s_i could be played so that there is some assignment of probabilities to the strategies below s_i in the strategy tree that can be extended to an ϵ-Nash equilibrium for the whole game. The following lemma guarantees the existence of such a type-symmetric ϵ-Nash equilibrium in our discretized search space.

Lemma 2.1. *Given an n-player AGG A with utilities in $[0, 1]$, with 1 player type and strategy graph G_A with maximum degree d, for any $\delta > 0$ there is a strategy profile $Q = (q_1, \ldots, q_{|S|})$ with each q_i a multiple of δ and the property that if all agents play profile Q, for any strategy s in the support of Q, $\mathbb{E}[u_s|Q] \geq \mathbb{E}[u_{s'}|Q] - 4\delta dn$ for all $s' \in S$.*

Proof. From Fact 2.1, there exists a mixed Nash equilibrium in which each player plays the same mixed strategy $P = (p_1, \ldots, p_{|S|})$. Consider another mixed strategy Q with the property that each q_i is a multiple of δ, $q_i = 0$ if $p_i = 0$, and otherwise $|q_i - p_i| \leq \delta$. (Note that such a profile clearly exists.) For a given strategy s with $|\nu(s)| = d$, we now show that

$$|\mathbb{E}[u_s|Q] - \mathbb{E}[u_s|P]| \leq 2\delta dn,$$

from which our lemma follows.

The utility that an agent receives for playing strategy s depends on how many of the other agents play the strategies $s_1, s_2, ..., s_d \in \nu(s)$. Define $P_s := (p_1, \ldots, p_d, 1 - \sum_1^d p_i)$, and $Q_s := (q_1, \ldots, q_d, 1 - \sum_1^d q_i)$. The number of players playing strategies $s_1, s_2, ..., s_d$ under the mixed strategies P and Q follow multinomial distributions with probability vectors P_s, Q_s (where the $d + 1^{st}$ outcome represents selecting a strategy that is not in the neighborhood of s). It is easy to couple the outcomes of a single draw from these multinomials to exhibit that their total variation distance is at most δd; indeed, we can couple the two outcomes so that each $i = 1, \ldots, d$ contributes at most δ in total variation distance, and so that the $d + 1^{st}$ outcome contributes δd. Since we have $n - 1$ agents independently selecting strategies according to P_s and Q_s the total variation distance between the distributions derived from P and Q of assignments of numbers of players to strategies in the neighborhood $s_1, \ldots, s_d$ will less than $n\delta d$. To conclude, note that since all utilities are between 0 and 1, $|\mathbb{E}[u_s|Q] - \mathbb{E}[u_s|P]| \leq 2\delta dn$ and our lemma follows.

Using standard techniques, the dynamic programming approach extends to the case of a strategy graph of constant treewidth. In this case, the strategy graph decomposes into a tree over cliques of vertices of size t and one can process all the vertices in each constant-sized clique simultaneously, resulting in at most a polynomial increase in running time. In the case that there are a constant number of player types, we need only modify the algorithm so as to maintain separate tables for each player type, and enforce that each player type uses only the allowed strategies.

Finally, to motivate a generalization of Theorem 2.1, consider an AGG with an unbounded number of player types, but whose action-graph is a tree with each player type restricted to disjoint connected components of the tree. In this setting, the dynamic programming approach clearly still applies, essentially without modification. Thus it is intuitive that even with an unbounded number of player types, if there is sufficient structure to the organization of the set of strategies available to each player type, an equilibrium can still be efficiently computed. The following definition allows us to formalize this intuition.

Definition 2.1. *We define the* agent-augmented action graph (AAAG) *of an AGG $A = \langle P, \mathbf{S}, G_A, u\rangle$ to be the graph resulting from starting with G_A, adding a vertex for each player type and adding an edge between each player type and the vertex corresponding to each strategy available to that player type. Formally, given a k-symmetric action graph game $A = \langle P, \mathbf{S}, G_A = (V_A, E_A), u\rangle$ with player types $P_1, P_2, \ldots, P_k$ with strategy spaces $S_1, S_2, \ldots, S_k$ respectively, the AAAG corresponding to A is*

$$G' = (V_A \cup \{P_1, P_2, \ldots, P_k\}, E_A \cup \{(P_i, s) : \forall i, \forall s \in P_i\})$$

The dynamic programming FPTAS can be generalized to prove the following extension of Theorem 2.2.

Theorem 2.2. *For any fixed constants d and t, an AGG A with AAAG G' which has tree-width t, and a t-treewidth decomposition graph with bounded degree d, an ϵ-Nash equilibrium can be computed in time polynomial in $|A|, 1/\epsilon, n$.*

Theorem 2.2 yields a variety of corollaries (in addition to Theorem 2.1) that apply to many naturally-arising games. As an example, we provide one that captures our motivating example.

Corollary 2.1. *Given an AGG A with strategy graph G_A which is a tree of bounded degree, and player types with strategy sets $S_1, \ldots, S_k$ that are each connected components of G_A, if $\max_s |\{i : s \in S_i\}|$ is bounded, then an ϵ-Nash equilibrium can be computed in time polynomial in $|A|, 1/\epsilon, n$.*

3 Hardness Results

In this section we state and prove our four hardness results. We show that it is (1) NP–complete to decide the existence of pure-strategy Nash equilibria, and (2) PPAD complete to approximate general (mixed Nash) equilibria for the classes of action graph games that either (a) have action graphs of treewidth 1 or (b) are symmetric (all agents are of a single type). Our two hardness results for pure equilibria will come from reductions from the NP–complete problem CIRCUIT-SAT, and follow the approach of [29]. Our hardness results for approximating mixed Nash equilibria are via equilibria-preserving gadgets that let us reduce from the PPAD-complete problem of computing equilibria in the class of graphical games where the maximum degree is 3 and each player has only two possible strategies. We begin by showing that action graph games are in the class PPAD, which was independently discovered in [20].

Mapping Action Graph Games to Graphical Games

We show the following result which reduces the problem of computing a Nash equilibrium of an action graph game to the problem of computing a Nash equilibrium of a graphical game. Since the latter is in PPAD [25], it follows that the former is in PPAD as well.

Theorem 3.1. *Any action-graph game A can be mapped in polynomial time to a graphical game $\mathcal{G}$ so that there is a polynomial-time computable surjective mapping from the set of Nash equilibria of $\mathcal{G}$ to the set of Nash equilibria of A.*

We define a bounded division-free straight-line program to be an arithmetic binary circuit with nodes performing addition, subtraction, or multiplication on their inputs, or evaluating to pre-set constants, with the additional constraint that the values of all the nodes remain in [0, 1]. To prove Theorem 3.1, we show that there exists a bounded division-free staight-line program of polynomial size in the description of the action graph game which, given a mixed strategy profile $M := \{(p_{i,1}, \ldots, p_{i,|S_i|})\}_{i=1}^{n}$, computes, for every agent i, $i = 1, \ldots, n$, and for every pure strategy s_i, $s_i \in S_i$, of that agent, the expected utility that this agent gets for playing pure strategy s_i. The proof then follows from Theorems 1 and 2 of [9]. We defer the details of the proof to the full version.

A Copy Gadget

As a preliminary to the hardness results of the next two subsections, we describe a *copy gadget* which will prove useful in both NP-completeness and PPAD-

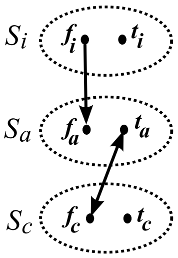

Figure 1: The *copy gadget*—in any Nash equilibrium $\Pr[s(c) = f_c] = \Pr[s(i) = f_i]$.

completeness results. Intuitively, to simulate games G of high treewidth by treewidth 1 action graph games H, we create several "copies" of each player, but only one copy of each edge relating players, thus ending up with a very "sparse" simulation, whose treewidth we can control. Explicitly, given an AGG A, and an agent i whose strategy set consists of the two strategies $S_i = \{f_i, t_i\}$ used only by player i, our copy gadget will add two additional players a, c, of which player c will be the "copy" and player a is an auxiliary player, whose inclusion will allow player i's strategies to be disconnected from player c's. We add strategies for a and c that are $\{f_a, t_a\}$ and $\{f_c, t_c\}$ respectively, and set the incentives so that in any Nash equilibrium $\Pr[s(i) = t_i] = \Pr[s(c) = t_c]$ (and $\Pr[s(i) = f_i] = \Pr[s(c) = f_c]$).

Definition 3.1. *Given an AGG $A = \langle P, \mathbf{S}, G, u\rangle$ and an agent i with two strategy choices $S_i = \{f_i, t_i\}$ such that no other player may play strategy f_i, we create AGG $A' = \langle P', \mathbf{S}', G', u'\rangle$ from A via the addition of a* copy gadget on i *as follows:*

- $P' := P \cup \{a, c\}$.
- $\mathbf{S}' := (S_1, \ldots, S_{|P|}, S_a, S_c)$*, where* $S_a = \{f_a, t_a\}$*, and* $S_c = \{f_c, t_c\}$*, where* $f_a, t_a, f_c, t_c \notin S$.
- *G' consists of the graph G with the additional vertices corresponding to f_a, t_a, f_c, t_c, and the directed edges (f_i, f_a), (t_a, f_c), (f_c, t_a).*
- *u' is identical to u for all strategies in $S' \backslash \{S_a \cup S_c\}$, and for a configuration D, $u'(f_a) = D(f_i)$, $u'(t_a) = D(f_c)$, $u'(f_c) = 1 - 2D(t_a)$, and $u'(t_c) = 0$.*

See Figure 1 for a depiction of the copy gadget.

Lemma 3.1. *Given an AGG A and an agent i, the addition of a copy gadget on i yields A' that satisfies the following properties:*

- *The description size of A' is at most a constant larger than A.*
- *In the strategy graph $G_{A'}$, f_c and t_c are not path connected to either f_i or t_i.*
- *In every ϵ^2–Nash equilibrium with agent i's profile $(p_{i,f}, 1 - p_{i,f})$, agent c's profile will have $|p_{c,f} - p_{i,f}| \leq \epsilon$ (and $|p_{c,t} - p_{i,t}| \leq \epsilon$).*

Proof. The first two properties follow directly from Definition 3.1. For the third property, assume otherwise and consider the case where $p_{c,f} > \epsilon + p_{i,f}$. Agent a's expected utility for playing f_a is $p_{i,f}$, and is $p_{c,f}$ for playing t_a, thus our assumption that $p_{c,f} > \epsilon + p_{i,f}$ implies that agent a must be playing t_a with probability at least $1 - \epsilon$ since the game is at ϵ^2–equilibrium. Given that a plays t_a with probability at least $1 - \epsilon$, agent c maximizes her utility by playing t_c, and thus $p_{c,f} \leq \epsilon$, which contradicts our assumption that $p_{c,f} > \epsilon + p_{i,f}$. An analogous argument applies to rule out the case $p_{c,f} < p_{i,f} - \epsilon$.

3.1 PPAD-Completeness

Our PPAD hardness results are reductions from the problem of computing equilibria in graphical games, and rely on the following fact due to [5].

Fact 3.1. *For the class of graphical games with n players, maximum degree 3 and payoffs in $\{0, 1, 2\}$, it is PPAD–complete to compute ϵ-Nash equilibria where $\epsilon \propto 1/poly(n)$.*

Theorem 3.2. *Computing a Nash equilibrium for AGGs with strategy graph G_A is PPAD-complete even if treewidth$(G_A) = 1$, and G_A has constant degree.*

Proof. From Theorem 3.1 this problem is in PPAD.

To show PPAD-hardness, we reduce from the known PPAD-hard problem of Fact 3.1. Given an instance of such a graphical game H, we construct an AGG A'_H with treewidth 1 and maximum degree 4 with similar description size to H such that there a polynomial time mapping from ϵ–Nash equilibria of A'_H to the ϵ–Nash equilibria of H. We construct A'_H via the intermediate step of constructing an AGG A_H which will be equivalent to H and might have large treewidth. From A_H, we construct A'_H using our copy gadget to reduce the treewidth of the associated strategy graph. See Figure 2 for a depiction of the reduction.

The construction of A_H is straightforward: for each player i_H in the graphical game, we have a corresponding player i_A in the AGG with strategy set $S_{i_A} = \{f_i, t_i\}$, corresponding to the two strategies that i_H may play in H. For each undirected edge between players $(i, j) \in H$, we add directed edges between the t nodes $(t_j, t_i), (t_i, t_j)$, and edges between the f nodes $(f_j, f_i), (f_i, f_j)$ to the strategy graph G_{A_H} of A_H. We define utilities u by simulating the utility functions from the original game H: from each f strategy connected to i_A in the AGG we know that if it is played then the corresponding t strategy is not played and vice versa; thus we have recovered the strategy choice of each neighbor of I_H in original graphical game; we then apply the utility function of the graphical game to compute the utility in the AGG. We do the symmetric procedure for the t nodes of the AGG. From the construction, it is clear that H and A_H represent the same game via the correspondence $i_H \to i_A$, and in particular an ϵ–Nash equilibrium of one game will correspond to an ϵ–Nash equilibrium of the other game via the natural mapping.

We obtain A'_H from A_H by making three copies of each i_A via the copy gadget. Thus for each i there are agents i_A, i^1_A, i^2_A, i^3_A with $S_{i^k_A} = \{f^k_{i_A}, t^k_{i_A}\}$. Finally, for each of the (at most three) outgoing edges of f_{i_A} that are not part of copy gadgets, i.e the edges of the form (f_{i_A}, f_{j_A}), we replace the edge by $(f^k_{i_A}, f_{j_A})$, with each $f^k_{i_A}$ having at most one outgoing edge, and modify the utility function u analogously so as to have the utility of strategy f_{j_A} depend on $f^k_{i_A}$ instead of f_{i_A}. Analogous replacements are made for the outgoing edges of t_{i_A}. Since the copied strategies $f^k_{i_A}, t^k_{i_A}$ are disconnected from the original strategies f_{i_A}, t_{i_A} the longest path in the strategy graph $G_{A'_H}$ associated with A'_H has length at most 4, with maximum degree 6, and $treewidth(G_{A_H}) = 1$. (See Figure 2.) Lemma 3.1 guarantees that the transformation from A_H to A'_H increases the representation size by at most a constant factor. Further, from an $\frac{1}{144}\epsilon^2$–Nash equilibrium of A'_H we can extract an ϵ–Nash equilibrium of A_H by simply ignoring the new players: all of the copies i^k_A of a player i_A will play strategies with probabilities within $\frac{1}{12}\epsilon$ of the probabilities of playing the original by Lemma 3.1; thus the joint distribution of any triple of these will have joint distribution within $\frac{1}{4}\epsilon$ of the "true" joint distribution; since each utility has magnitude at most 2 the computed utilities will be within $\frac{1}{2}\epsilon$ of the utilities computed in A_H; thus each of the mixed strategies of a player i_A in A'_H, interpreted as a strategy in A_H will yield utility within ϵ of optimal. From Fact 3.1 we conclude that finding an $\frac{1}{144}\epsilon^2$–Nash equilibrium of A'_H is PPAD complete for any polynomial ϵ, yielding the desired result.

We now turn our attention to AGGs that have a constrained number of player types.

Theorem 3.3. *Computing a Nash equilibrium for AGGs with 10 player types is PPAD-complete even if the strategy graph G_A has bounded degree.*

To show PPAD-hardness, as above we reduce from the known PPAD-hard problem of computing Nash equilibria in graphical games of degree at most 3 where each player chooses between 2 strategies f, t and has utility 0, 1, or 2. Given such a graphical game H, we will reduce it to an AGG A_H that has strategies f_i, t_i corresponding to the two strategies that agent i may choose in H. Intuitively, if our reduction is to be successful there are several properties of H that seem necessary. First, in every Nash equilibrium of H, there must be at least one agent playing either f_i or t_i for every i. This is accomplished by giving agents a bonus payment if they choose either of the two strategies of a sparsely-played f_i, t_i pair. Second, there must be some unambiguous mapping between the number of agents playing f_i and t_i in A_H to a choice of actions of agent i in H. This is accomplished via the MAJORITY function: if more agents play f_i than t_i in A_H, we say that i plays f. This motivating intuition is formalized in the proof below.

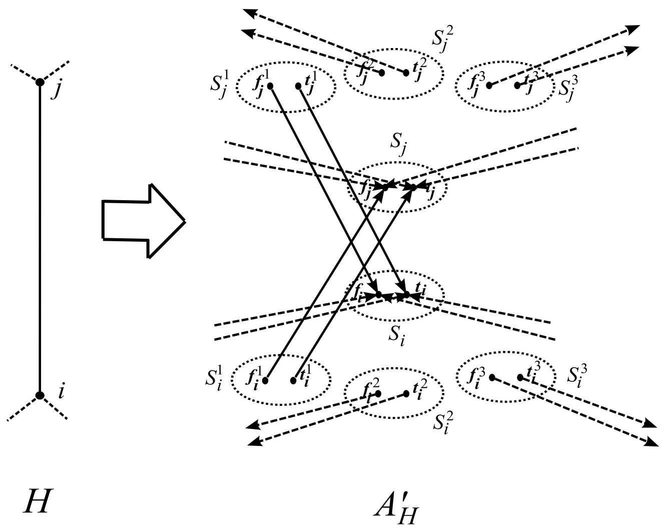

Figure 2: The transformation from the graphical game H to the AGG A'_H. For simplicity, the internal strategies and edges associated with the copy gadgets are omitted.

Proof of Theorem 3.3: From Theorem 3.1 this problem is in PPAD.

To show PPAD-hardness, we reduce from the PPAD-hard problem of Fact 3.1, namely, computing ϵ-Nash equilibria in graphical games of degree at most 3 where each player chooses between 2 strategies and has utility 0, 1, or 2, where ϵ is chosen to be inverse polynomial in n but less than $\frac{1}{18n}$. Given an instance of such a graphical game H with n agents, we construct the AGG A_H so that an ϵ^2–Nash equilibrium of A_H can be efficiently mapped to an ϵ–Nash equilibrium of H. We construct $A_H = \langle P, \mathbf{S}, G_A, u\rangle$ as follows:

- $P := \{1, \ldots, 3cn\}$ with $c > \frac{64}{\epsilon^2}$.
- Let the strategy set $S := \{f_1, t_1, \ldots, f_n, t_n\}$ where strategies f_i and t_i correspond to the two strategies of the i^{th} agent of H; the strategy sets for each player are defined as follows: since the graph of H has degree ≤ 3, we may trivially 10-color it so that no nodes within distance 2 are of the same color; for each of the ten colors, chose $\frac{1}{10}$ of the players P (in the action graph game) and let their strategy set consist of those f_i, t_i with i of that color.
- For every undirected edge (i, j) in the graph of H, the strategy graph G_A has the eight directed edges (f_i, f_j), (f_j, f_i), (f_i, t_j), (t_j, f_i), (t_i, f_j), (f_j, t_i), (t_i, t_j), (t_j, t_i). Furthermore, for all $i \in \{1, \ldots, n\}$, G_A contains the edges $(f_i, t_i), (t_i, f_i)$ and the self loops (f_i, f_i) and (t_i, t_i).
- To simplify the description of the utility function u, it will be useful to define the indicator functions $I_1[D(f_1, t_1)], \ldots, I_n[D(f_n, t_n)]$ where

  $$I_i[D(f_i, t_i)] := \begin{cases} f & \text{if } D(f_i) \geq D(t_i) \\ t & \text{if } D(f_i) < D(t_i) \end{cases}$$

  Let u assign utility to f_i as a function of $D(\nu(i))$, where $\nu(i)$ denotes i's neighbors, by applying the utility function for agent i from H on the simulated actions of her neighbors j_1, j_2, j_3 evaluated as $I_{j_1}[D(f_{j_1}, t_{j_1})], I_{j_2}[D(f_{j_2}, t_{j_2})]$, and $I_{j_3}[D(f_{j_3}, t_{j_3})]$, respectively. Finally, if $D(f_i) + D(t_i) \leq c$, u assigns an extra 100 utility to strategies f_i and t_i.

Observe that the description size of A_H is polynomial in cn, and thus is polynomial in the description size of H (since $\frac{1}{\epsilon}$ and hence c are polynomial in n). From Fact 3.1, our theorem will follow if we show that any ϵ^2–Nash equilibrium of A_H can be efficiently mapped to an ϵ–Nash equilibrium of H.

Consider the map from mixed strategy profiles of A_H to mixed strategy profiles of H given by $\phi : \mathbf{M_A} \to \mathbf{M_H}$ that assigns $M_H = [(p_{1,f}, 1 - p_{1,f}), \ldots, (p_{n,f}, 1 - p_{n,f})]$ by setting $p_{i,f} := \Pr_{M_A}(I_i = f)$ where the probability is taken over the distribution over Δ defined by M_A. It is clear that the map ϕ can be computed efficiently, as it simply involves computing the probabilities that certain numbers of independent random variables take a certain value.

Before showing that ϕ maps ϵ^2–equilibria to ϵ–equilibria we first show that the "extra utility" of 100 correctly incentivizes a large number of players to play on each strategy pair. We observe that in any mixed strategy profile there will be at least one agent, j, who has probability at most 1/3 of receiving a payoff of at least 100 (since in any outcome, at most cn of the 3cn players receive this extra payoff of 100, there must be some player who receives this payoff with probability no more than 1/3). Since his payoff from the simulation of H is at most 2, such an agent's expected utility is at most $100/3 + 2 < 36 - \epsilon$, and thus any Nash equilibrium mixed strategy profile must satisfy $100 \Pr(D(f_i) + D(t_i) < c) < 36, \forall i \in \{1, \ldots, n\}$, for if this were not the case, then agent j could improve her expected utility to at least 36 by always choosing strategy f_i, contradicting the fact that the players are in equilibrium. Thus with probability at least 64% we have $D(f_i) + D(t_i) \geq c$, in which case we also have $\max(D(f_i), D(t_i)) \geq \frac{c}{2}$. Since $64\% > \frac{1}{2}$ we have:

$$\mathbb{E}\left[\max\left(D(f_i), D(t_i)\right)\right] > \frac{c}{4}.$$

We now proceed with the proof of correctness of the map ϕ. Let M_A be an ϵ^2–Nash equilibrium of A_H, and $M_H = \phi(M_A)$. Consider a player i in the graphical game, and a strategy of his that he plays with probability at least ϵ. Without loss of generality let this strategy be f_i. We show that his utility for playing f_i is at least his utility for playing his other choice, t_i, minus ϵ; taken together, these statements imply that M_H is an ϵ–Nash equilibrium of the graphical game, as desired.

Since $\mathbb{E}\left[\max\left(D(f_i), D(t_i)\right)\right] > \frac{c}{4}$, we have that if f_i is played with probability at least ϵ, namely if $\Pr(D(f_i) \geq D(t_i)) \geq \epsilon$ then (by Chernoff bounds) we must have $\mathbb{E}[D(f_i)] \geq \frac{c}{6}$. This implies that for at least one of the $3cn$ players j in A_H, his probability of playing f_i is at least $\frac{1}{18n}$, which is at least ϵ. Thus, since M_A is, by assumption, an ϵ^2–Nash equilibrium, we have that player j's utility for playing f_i is at most ϵ below his utility for playing t_i, when the other players play from M_A. Since, by construction, the distribution of the actions of i's neighbors j_1, j_2, j_3 (in the graphical game) is identical to the distribution of $I_{j_1}[D(f_{j_1}, t_{j_1})], I_{j_2}[D(f_{j_2}, t_{j_2})], I_{j_3}[D(f_{j_3}, t_{j_3})]$ (since the marginals are equal by definition, and the marginals are independent since j_1, j_2, j_3 and i are played by different player types due to our 10-coloring) we conclude that the expected payoffs for t_i, f_i in the action graph game are identical to those in the graphical game, plus the constant coming from the probability of the extra 100 payoff. Thus since the expected utility in the AGG for f_i is at most epsilon less than the expected utility of t_i, this also holds in the original graphical game H, which was exactly the conclusion we needed to prove that the ϵ^2-Nash equilibria M_A in A_H maps to an ϵ-Nash equilibria in H. Thus approximating Nash equilbiria of AGGs of constant degree and number of player types is PPAD hard.

3.2 NP–Completeness

Both of our NP–completeness results are reductions from the NP-Complete problem CIRCUITSAT and follow an approach employed in [29].

Fact 3.2. *It is NP-complete to decide satisfiability for the class of circuits consisting of AND, OR, and NOT gates, with maximum degree* 3 *(in-degree plus out-degree).*

In our reductions from CIRCUITSAT, given a circuit C, we construct an AGG A_C that computes C in the sense that pure strategy Nash equilibria of A_C map to valid circuit evaluations. To this game we add two agents that have a simple pure-strategy equilibrium if C evaluates to $true$, but when C evaluates to $false$ play *pennies*—a simple game that has no pure strategy Nash equilibria. Thus the existence of a pure strategy Nash equilibrium is equivalent to the satisfiability of C.

Theorem 3.4. *Deciding the existence of a pure strategy Nash equilibrium for AGGs with strategy graph G_A is NP-complete even if $treewidth(G_A) = 1$, and G_A has constant degree.*

Proof. Membership in NP is clear. To show hardness, given a circuit C, we construct the associated AGG $A_C := \langle P, \mathbf{S}, G_A, u\rangle$ as follows:

- $P := \{1, \dots, n, p_1, p_2\}$, where n is the number of gates in C, and the gate corresponding to player n is the output gate.
- $\mathbf{S} := ((f_1, t_1), \dots, (f_n, t_n), (f_{p_1}, t_{p_1}), (f_{p_2}, t_{p_2}))$.
- For every pair of gates i, j for which the output of gate i is an input to gate j, G_A has the edges (f_i, f_j), and (f_i, t_j). Furthermore, we add edges $(f_n, f_{p_1}), (f_n, t_{p_1}), (f_n, f_{p_2}), (f_n, t_{p_2})$, and the edges (f_{p_1}, f_{p_2}), (f_{p_1}, t_{p_2}), (f_{p_2}, f_{p_1}), (f_{p_2}, t_{p_1}).
- The utility function u is defined as follows: if agent i corresponds to an input gate, then strategies f_i, t_i both have utility 0. For any other agent i corresponding to a gate of C, the payoff of strategy f_i is 1 or 0 according to whether f_i is the correct output value of gate i given the values corresponding to the strategies played by neighboring agents/strategies. Similarly for the payoff for strategy t_i. If $D(f_n) = 0$, then f_{p_1} and t_{p_1} have utility 0, otherwise the utility of p_1 is 1 if $D(f_{p_1}) = D(f_{p_2})$, and is 0 otherwise. The utility of p_2 is 1 if $D(f_{p_1}) \neq D(f_{p_2})$, and is 0 otherwise.

From the construction it is clear that if C is satisfiable then there is a pure strategy profile for agents $1, \dots, n$ with agent n playing t_n, such that agents $1, \dots, n$ can not improve their utility by deviating from their strategies. Furthermore, p_1 will be indifferent between her strategies, and p_2 will play the opposite of p_1; in particular, there will be a pure strategy Nash equilibrium. If C is not satisfiable, then any pure strategy profile that is an equilibrium for agents $1, \dots, n$ will have $D(f_n) = 1$, and thus p_1 will be incentivized to agree with p_2, and p_2 will be incentivized to disagree, and thus A_C will admit no pure strategy Nash equilibrium.

To complete the proof, note that we can apply the copy gadget to each agent of A_C, as was done in the proof of Theorem 3.2 to yield the game A'_C that has strategy graph of treewidth 1, and a mapping from equilibria of A'_C to equilibria of A_C.

Theorem 3.5. *Deciding the existence of a pure strategy Nash equilibrium for symmetric AGGs (*1 *player type) is NP-complete even if the strategy graph G_A has bounded degree.*

Proof. Membership in NP is clear; to show hardness we proceed as was done in the proof of Theorem 3.4, and obtain AGG A_C from circuit C. Now, we make A_C symmetric by retaining the same number of agents, but allowing each of them to pick any of the strategies. We modify the strategy graph G by adding edges $(f_x, t_x), (t_x, f_x), (f_x, f_x), (t_x, t_x)$ for each player x from A_C, and extend the utility function u so that if $D(f_x) + D(t_x) > 1$ then strategies f_x and t_x have utility -1. Thus in any pure strategy Nash equilibrium $D(f_x) + D(t_x) = 1$, and the reasoning in the proof of Theorem 3.4 applies to complete our reduction.

4 Conclusions and Open Problems

The results in this paper are of a negative nature. While we exhibit a simple FPTAS for the case of a bounded number of player types and bounded treewidth, we show that neither of these conditions can be relaxed if we hope to retain a polynomial time solution. Unfortunately, this suggests that the search must continue for computationally tractable models of large-scale games. We leave this as the main open question, and note that perhaps there are other restricted classes of games that circumvent our hardness results while retaining some of the motivating features of general action graph games.

References

[1] Timothy Abbot, Daniel Kane, and Paul Valiant. On the complexity of two-player win-lose games. In *Proceedings of the 46th FOCS*, pages 113–122, 2005.

[2] Navin A. R. Bhat and Kevin Leyton-Brown. Computing nash equilibria of action-graph games. In *UAI*, pages 35–42, 2004.

[3] Matthias Blonski. Anonymous games with binary actions. *Games and Economic Behavior*, 28(2):171–180, August 1999. available at http://ideas.repec.org/a/eee/gamebe/v28y1999i2p171-180.html.

[4] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272, Washington, DC, USA, 2006. IEEE Computer Society.

[5] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Computing nash equilibria: Approximation and smoothed complexity. In *FOCS*, pages 603–612, 2006.

[6] Xi Chen, Shang-Hua Teng, and Paul Valiant. The approximation complexity of win-lose games. In *Proceedings of SODA*, 2007.

[7] Steve Chien and Alistair Sinclair. Convergence to approximate nash equilibria in congestion games. In *SODA*, pages 169–178, 2007.

[8] Bruno Codenotti and Daniel Štefankovič. On the computational complexity of nash equilibria for (0, 1) bimatrix games. *Inf. Process. Lett.*, 94(3):145–150, 2005.

[9] Constantinos Daskalakis, Alex Fabrikant, and Christos H. Papadimitriou. The game world is flat: The complexity of nash equilibria in succinct games. In *ICALP (1)*, pages 513–524, 2006.

[10] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. In *STOC*, 2006.

[11] Constantinos Daskalakis and Christos H. Papadimitriou. Computing pure nash equilibria in graphical games via markov random fields. In *ACM Conference on Electronic Commerce*, pages 91–99, 2006.

[12] Constantinos Daskalakis and Christos H. Papadimitriou. Computing equilibria in anonymous games. In *FOCS*, 2007.

[13] Constantinos Daskalakis and Christos H. Papadimitriou. Discretized multinomial distributions and nash equilibria in anonymous games. FOCS, 2008.

[14] Juliane Dunkel and Andreas S. Schulz. On the complexity of pure-strategy nash equilibria in congestion and local-effect games. In *WINE*, pages 62–73, 2006.

[15] F. Fischer F. Brandt and M. Holzer. Equilibria of graphical games with symmetries. Technical Report TR07-136, Electronic Colloquium on Computational Complexity (ECCC), December 2007.

[16] Alex Fabrikant, Christos H. Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *STOC*, pages 604–612, 2004.

[17] D Gale, HW Kuhn, and AW Tucker. On symmetric games. *Contributions to the Theory Games, Annals of Mathematics Studies*, 24, 1950.

[18] Albert Xin Jiang and Kevin Leyton-Brown. A polynomial-time algorithm for Action-Graph Games. In *American Association for Artificial Intelligence (AAAI)*, pages 679–684, 2006.

[19] Albert Xin Jiang and Kevin Leyton-Brown. Computing pure nash equilibria in symmetric action graph games. In *AAAI*, pages 79–85. AAAI Press, 2007.

[20] Albert Xin Jiang, Kevin Leyton-Brown, and Navin Bhat. Action–graph games. 2007. Working paper, available at: http://www.cs.ubc.ca/∼jiang/papers/AGG.pdf.

[21] Ravi Kannan and Thorsten Theobald. Games of fixed rank: a hierarchy of bimatrix games. In *SODA*, pages 1124–1132, 2007.

[22] Michael Kearns, Michael L. Littman, and Satinder Singh. Graphical models for game theory. In *UAI*, pages 253–260, 2001.

[23] C. E. Lemke and Jr J. T. Howson. Equilibrium points of bimatrix games. *SIAM Journal of Applied Mathematics*, 12:413–423, 1964.

[24] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[25] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994.

[26] Christos H. Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. In *SODA '05*, pages 82–91, 2005.

[27] J. Rosenmuller. On a generalization of the lemke–howson algorithm to noncooperative n-person games. *SIAM Journal of Applied Mathematics*, 21:73–79, 1971.

[28] H. E. Scarf. The approximation of fixed points of a continuous mapping. *SIAM Journal of Applied Mathematics*, 15:1328–1343, 1967.

[29] Grant Schoenebeck and Salil Vadhan. The computational complexity of nash equilibria in concisely represented games. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 270–279, New York, NY, USA, 2006. ACM.

[30] G. van der Laan and A. J. J. Talman. On the computation of fixed points in the product space of unit simplices and an application to noncooperative n person games. *Mathematics of Operations Research*, 7, 1982.

[31] J. von Neumann and O. Morgenstern. On the computation of fixed points in the product space of unit simplices and an application to noncooperative n person games. *Theory of Games and Economic Behavior*, 1944.

[32] R. Wilson. Computing equilibria of n-person games. *SIAM Journal of Applied Mathematics*, 21:80–87, 1971.