# The Curse of Simultaneity

Renato Paes Leme[*]
Dept of Computer Science
Cornell University
renatoppl@cs.cornell.edu

Vasilis Syrgkanis[†]
Dept of Computer Science
Cornell University
vasilis@cs.cornell.edu

Éva Tardos[‡]
Dept of Computer Science
Cornell University
eva@cs.cornell.edu

## ABSTRACT

Typical models of strategic interactions in computer science use simultaneous move games. However, in applications simultaneity is often hard or impossible to achieve. In this paper, we study the robustness of the Nash Equilibrium when the assumption of simultaneity is dropped. In particular we propose studying the sequential price of anarchy: the quality of outcomes of sequential versions of games whose simultaneous counterparts are prototypical in algorithmic game theory. We study different classes of games with high price of anarchy, and show that the subgame perfect equilibrium of their sequential version is a much more natural prediction, ruling out unreasonable equilibria, and leading to much better quality solutions.

We consider three examples of such games: Cost Sharing Games, Unrelated Machine Scheduling Games and Consensus Games. For *Machine Cost Sharing Games*, the sequential price of anarchy is at most $O(\log(n))$, an exponential improvement of the $O(n)$ price of anarchy of their simultaneous counterparts. Further, the subgame perfect equilibrium can be computed by a polynomial time greedy algorithm, and is independent of the order the players arrive. For *Unrelated Machine Scheduling Games* we show that the sequential price of anarchy is bounded as a function of the number of jobs $n$ and machines $m$ (by at most $O(m2^n)$), while in the simultaneous version the price of anarchy is unbounded even for two players and two machines. For *Consensus Games* we observe that the optimal outcome for generic weights is the unique equilibrium that arises in the sequential game. We also study the related *Cut Games*, where we show that the sequential price of anarchy is at most 4. In addition we study the complexity of finding the subgame perfect equilibrium outcome in these games.

[*]Supported in part by NSF grants CCF-0729006, and by internships at Google and Microsoft, and a Microsoft Fellowship..

[†]Supported in part by ONR grant N00014-98-1-0589 and NSF grants CCF-0729006, and by a Microsoft Internship.

[‡]Supported in part by NSF grants CCF-0910940 and CCF-0729006, ONR grant N00014-08-1-0031, a Yahoo! Research Alliance Grant, and a Google Research Grant.


*ITCS* '12 Cambridge, Massachussets USA


## Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

## General Terms

games, economics, algorithms

## Keywords

price of anarchy, subgame perfect equilibrium, extensive form games

## 1. INTRODUCTION

A powerful line of algorithmic research over the past decade has developed techniques for analyzing systems composed of self-interested agents. Typical models of the strategic interactions of agents use simultaneous move games: each player or participant, simultaneously chooses an action, such as submitting a bid in an auction, or simultaneously selecting strategies in a routing game. However, simultaneity is often hard or impossible to achieve in implementations. In this paper, we propose studying the sequential price of anarchy: the quality of outcomes of sequential versions of games whose simultaneous counterparts are prototypical in algorithmic game theory.

We consider games with high price of anarchy. In many such games the equilibria resulting in the high price of anarchy require "unnatural" coordination from the players. A typical example arises in cost-sharing: when players control a job to be scheduled, it might be unreasonable to simultaneously ask them to decide which machine they will use, and unnatural to expect that they will all select the same expensive machine, even if this is an equilibrium of the game. It is more natural to allow players to select machines sequentially. We will show that the sequential decision making helps avoid bad equilibria in this game and a number of other games, and results in exponential (or better) improvement in the price of anarchy.

We consider full information sequential games, and measure quality of outcomes using *subgame perfect equilibrium* (SPE), capturing sequential rationality of the players. There is a large body of work on online games (see [22] for a survey), where players have to make strategic decisions without having any information about the future. Sequential games model the strategic behavior of agents who anticipate future strategic opportunities. While the full information assumption we make may be too strong in many applications, participants often do have knowledge about their future options and strategically anticipate opportunities, in a way not possible in online games. Studying sequential games offers insight in the effect of the sequential rationality of the players. Full information sequential games model scenarios in distributed systems where the game is close to simultaneous: players choose strategies in fast succession, but sequential choices allow the players to avoid unfortunate equilibria.

Subgame perfect equilibrium in games with a single player acting in each step has the nice feature of generally producing an unique outcome, and we show here that in many games this outcome has good quality. Unfortunately, this outcome can be hard to compute in some games. However, despite the high worst case complexity, calculating subgame perfect equilibrium is a long studied problem in the AI literature. Many methods were developed to play board games like chess and checkers which are sequential in nature. An example is the celebrated *alpha-beta search*, first documented in Hart and Edwards [15], but whose origins date back to undocumented sources in the 50's, as described in a survey by Knuth and Moore [17].

*Our Results.* In section 3 we consider **Machine Cost Sharing Games.** Players correspond to jobs, and decide sequentially on a machine to assign their job to . Each machine has a cost (possibly a cost increasing with congestion), and the players selecting a machine share the cost evenly. This congestion game has been extensively studied [2] for the social welfare, the total cost of all machines. It is known to have price of anarchy $n$, the number of players; while the price of stability is $H_n = O(\log n)$. We show that there is a unique Spe under generic costs, and show that the Price of Anarchy of subgame perfect equilibrium is bounded by $H_n = O(\log n)$. The sequential reasoning guarantees that the agents avoid the bad equilibria, resulting in an exponential improvement over the simultaneous version. We also show that the subgame perfect equilibrium can be computed in polynomial time, and the equilibrium doesn't depend on the order the players arrive.

In section 4 we consider **Unrelated Machine Scheduling:** Each player controls a job that has a (potentially) different processing time on each machine. Players schedule jobs in one of $m$ machines sequentially and experience the total processing time of their chosen machine. We evaluate the system using the classical measure of the makespan, the total processing time of the machine with maximum load. The classical simultaneous version of this game has unbounded Price of Anarchy (even for $n = m = 2$). Sequential reasoning helps agents to evade from bad equilibria. For subgame perfect equilibrium, we prove that the price of anarchy is bounded as a function of the number of jobs $n$ and number of machines $m$, giving an upper bound of $O(m \cdot 2^n)$, and a lower bound of $n$ on the sequential price of anarchy.

In section 5 we consider **Consensus and Cut Games.** There are some parties (say red or blue). Players affiliate to one of the parties one by one. In the *consensus game* players incur a cost from the players in the different party. The simultaneous version of consensus can have arbitrarily bad equilibria [5]. Generically, the only equilibrium of the sequential version is the optimal solution. We also study the version of the game when players derive utilities, rather than costs, from the players in the other party, which we call *cut game*. This class of games was introduced in [11] as party-affiliation games and revisited many times later, as for example in [5]. We show that the price of anarchy of the sequential version is bounded by 4.

At last, in section 6 we study the complexity of computing a subgame perfect equilibrium. Recall that for the machine cost-sharing problem studied in section 3 the unique subgame perfect equilibrium can be computed in polynomial time. Unfortunately, for other problems finding the subgame perfect equilibrium can be hard. We show that for the unrelated machine scheduling and general congestion games computing a subgame perfect equilibrium is PSPACE-complete.

*Related Work.* We consider classes of games with bad price of anarchy, where we believe that the bad equilibria require unnatural coordination from the players. Machine cost sharing games are a special case of the network cost sharing introduced by Anshelevich et al. [2], and are sometimes also called set-cover games, for example, in [4], and have a price of anarchy equal to the number of players. Machine Scheduling Games, also called Load Balancing Games, are among traditional applications of Price of Anarchy analysis - they are studied in the seminal paper of Koutsoupias and Papadimitriou [18], and in the general form have an unbounded price of anarchy even for two jobs and two machines. Consensus games were introduced in [5], who show an arbitrarily bad price of anarchy.

In all of the above games, the examples with bad price of anarchy appear to require rather unnatural coordination from the players. Giving a solution concept for these games with better price of anarchy is an important open problem in the area. There have been several attempts in the literature to introduce solution concepts that rule out the bad examples, and results in a small price of anarchy.

**Solution Concepts with Improved Price of Anarchy.** Andelman et al. [1] proposed the study of Strong Nash Equilibria; outcomes that are stable under group deviations. They showed that for the unrelated machine scheduling games the Strong Price of Anarchy is at most $2m - 1$, where $m$ is the number of machines and that strong equilibria always exist. Later, Epstein et al. [10] showed that the Strong Price of Anarchy of cost sharing games is $H_n = O(\log n)$, same as our bound for Subgame Perfect Equilibria. Strong Equilibria assume collective rationality, and requires players to collaborate, while Spe assumes only individual rationality. Moreover, unlike Subgame Perfect Equilibria, Strong Equilibria are not guaranteed to exist, which limits its applicability.

Chekuri et al. [8] study the case where players arrive sequentially playing myopically and then perform best response until they reach a Pure Nash Equilibrium. They prove that for the case of Multicast Cost Sharing on an undirected network the price of anarchy of any Pure Nash Equilibrium reached by the above process is $O(\sqrt{n}\log^2(n))$ times the optimal. This was later improved to $O(\log^3(n))$ by Charikar et al [7]. While their model is similar to ours, as it also incorporates the fact that players act sequentially, they assume that players are myopic. In contrast, we assume players are strategic and choose their current actions taking into account future implications. Moreover, due to myopic playing their analysis doesn't carry over to the machine cost sharing games that we analyze. In fact, it is easy to find examples where the price of anarchy remains $O(n)$ even when players arrive sequentially and play myopically: when the optimal solution involves large number of players sharing a machine, myopic players do not find this solution.

In some cases noisy best response is also known to lead to improved price of anarchy. Chung et al. [9] study the price of anarchy of stochastically stable states of noisy imitation dynamics. They show that the price of anarchy of such states is bounded in the case of unrelated machine scheduling. Montanari et al. [20] study the speed of convergence of logit dynamics (noisy best response) in network coordination games, a model similar to consensus games, to the optimal outcome, which is the unique stochastically stable state. However, most of these dynamics have a very slow speed of convergence in the models that we study. Balcan et al [6] shows that there are instances of machine cost sharing games where no type of noisy dynamics can achieve a price of anarchy smaller than $n/\log(n)$ in a polynomial number of steps. The prediction that a stochastically stable state will arise is not applicable to cases where convergence is slow, as is the case in some of the games that we study.

To avoid the slow convergence Balcan et al [6] consider the case when a central authority advertises strategies and players either adopt them with some constant probability or play best response. They showed that for cost sharing games the dynamics will reach states with price of anarchy at most $\log(n)\log(nm)$ in a polynomial number of steps and for consensus games the optimal outcome will arise if players adopt the advertised strategy with probability at least $1/2$. However, Balcan [5] showed that this approach fails for the case of unrelated machines. Moreover, this technique assumes the existence of a central authority that computes optimal strategy profiles and that players trust at some level.

**Extensive Form Games.** In this paper we study the price of anarchy for sequential version of the above games. The study of extensive form games dates back to the first formal studies of Game Theory. The extensive form predates even the normal form. Starting from the first formal work on chess by Zemelo [27] and then by von Neumann [26] who first introduced the extensive form for games with perfect information. A detailed exposition of classic literature on extensive form games and multi-stage games can be found in [12].

Some previous works in the literature have studied efficiency in extensive form games, especially in the context of auctions (e.g. [3]). Recently in [21], we analyze the quality of outcomes in sequential auctions. In the setting considered there, players have valuation over bundles of items and the auctioneer holds a first-price auction for one item at a time. We analyze existence of equilibrium and quality of outcomes for different classes of valuation functions. The techniques in [21] are, however, very different from those in the current paper.

Sequential games studied here have unique equilibria (assuming generic costs), and in this sense are analogous to equilibrium refinement. Like our subgame perfect equilibrium, many equilibrium refinements, such as those of Harsanyi-Selten [14], or Homotopy methods [16], were recently shown by Goldberg, Papadimitriou and Savani [13] to be PSPACE-complete. Interestingly though, for machine cost-sharing games the SPE is easy to find.

## 2. SEQUENTIAL GAMES

We consider games that happen in a sequence of rounds, where a single player acts in each round. Given $n$ players with action sets $A_1, \ldots, A_n$, utility functions $u_i : \times_i A_i \to \mathbb{R}$ for each player and an ordering of the players, say player $1, 2, \ldots, n$.

In each round $i$, player $i$ observes the actions chosen by players $1, 2, \ldots, i-1$ and chooses an action $a_i \in A_i$. Therefore, the strategy of player $i$ is a mapping $s_i : A_1 \times \ldots \times A_{i-1} \to A_i$.

Given the strategies, the outcome $a = (a_1, \ldots, a_n)$ is defined recursively: $a_1 = s_1(\emptyset), a_2 = s_2(a_1), a_3 = s_3(a_{1..2}), \ldots, a_i = s_i(a_{1..i-1})$. Player $i$ then experiences utility $u_i(a_{1..n})$, where $a_{i..j}$ is the vector $(a_i, a_{i+1}, \ldots, a_j)$.

Given a prefix $(\alpha_{1..k}) \in A_1 \times \ldots \times A_k$ for some $k < n$, it defines an induced subgame for players $k+1, \ldots, n$ in the natural way: we define the outcome to be $a_i = s_i(\alpha_{1..k}, a_{k+1..i-1})$ and players experience utilities $u_i(\alpha_{1..k}, a_{k+1..n})$.

A set of strategies $(s_1, \ldots, s_n)$ is a **subgame perfect equilibrium** (SPE) if it is simultaneously an equilibrium of all subgames defined by its prefixes. Clearly a subgame perfect equilibrium is a Nash equilibrium of the original sequential game since it corresponds to the prefix game with empty prefix.

Subgame perfect equilibria always exist, and can be easily found by backwards, induction: Let $h_i(a_{1..i})$ be the outcome in the subgame defined by the prefix $a_{1..i}$. Now we have that $h_n(a_{1..n}) = \emptyset$, and for $i = n-1, n-2, \ldots, 1$, we define

$$h_i(a_{1..i}) = (s_{i+1}(a_{1..i}), h_{i+1}(a_{1..i}, s_{i+1}(a_{1..i})))$$

$$s_i(a_{1..i-1}) \in \operatorname{argmax}_{x \in A_i} u_i(a_{1..i-1}, x, h_i(a_{1..i-1}, x))$$

Note that if the utility functions are such that the argmax is a single element (say for example if the entries of the utility matrix are all different), then the SPE is unique. The concepts presented above are a special case of **extensive form games** (see [12] for a comprehensive treatment and a more general definition).

Given a welfare function $W : \times_i A_i \to \mathbb{R}^+$, we quantify the **sequential price of anarchy** (SPoA) of the game as the ratio between the optimal solution (measured in terms of $W$) and the quality of the worse subgame perfect equilibrium. If $\text{SPE} \subseteq \times_i A_i$ are the action profiles that can happen in a subgame perfect equilibrium and $W^* = \max_{a \in \times_i A_i} W(a)$, then we define:

$$\text{SPoA} = \max_{a \in \text{SPE}} \frac{W^*}{W(a)}$$

If the game is defined in terms of a cost function (where the optimum is the solution of minimum cost) then we simply invert the numerator and denominator.

## 3. MACHINE COST SHARING GAMES

Consider the following cost sharing game: there is a set $N$ of $n$ jobs and a set $R$ of $m$ machines. Each job $i$ has a set of machines $R_i$ from which he can choose and each machine $r$ is associated with a decreasing cost function $\gamma_r(x)$. The game played is the following: each job is a player and its strategy is to choose a machine $s_i \in R_i$. The cost of a player in a strategy profile $s$ is then given by:

$$c_i(s) = \gamma_{s_i}(n_{s_i}) \text{ where } n_r = |\{j \in N; s_j = r\}|$$

A very well studied case is that of fair cost allocation, where the cost function of a machine $r$ has the form: $\gamma_r(x) = c_r/x$, capturing the case where each machine has a fixed cost that has to be covered by the people using it and this cost is equally split among the players. In general, we can think of the cost of running a machine $r$ with congestion $x$ as $c_r(x) = x\gamma_r(x)$, and then the cost of a player is the fair share $\gamma_r(x) = c_r(x)/x$. We will assume that the cost $\gamma_r(x)$ satisfies a natural economy of scale and is decreasing in $x$.

This class of games was introduced by Anshelevich et al. [2], who study a simultaneous move game and show that the Price of Anarchy is $n$ while the Price of Stability of this game is $O(\log n)$ under the social cost function $C(s) = \sum_i c_i(s)$, when the machine cost functions have the form $\gamma_r(x) = c_r(x)/x$ with a concave function $c_r(x)$. above.

The worst case PoA example is when there are two machines of costs $1+\epsilon$ and $n$ and each player can have access to both of these machines. It is a Nash equilibrium for all players to choose the machine that costs $n$ since all the players have cost 1 and they don't want to switch alone and increase their cost to $1+\epsilon$. This worst case example breaks if players arrive sequentially. Informally, they can choose the cheaper machine and rely on the rationality of the following players that they will do the same. In fact, if players arrive in some fixed order and play an SPE, we show that the worst possible efficiency deterioration is exponentially better than that in the simultaneous move version.

To simplify the presentation, we will focus on the simple case of fair cost sharing ($\gamma_r(x) = c_r/x$) and it is easy to see that they extend to the more general case.

We say that the machines have generic costs if $c_r/k \neq c_{r'}/k'$ for two different machines $r \neq r'$ and any $1 \leq k, k' \leq n$. Any cost vector $c$ can be made generic with a small random perturbation.

**Theorem 1** *For any machine cost sharing game with fair cost allocation and generic costs, there is a unique* SPE *and it is within an $O(\log n)$ factor from the optimal. Moreover, it can be computed by a natural greedy algorithm. When the costs are not generic, there may be more then one* SPE *but the Price of Anarchy bound still holds.*

PROOF. For simplicity we will consider generic costs only. Notice that the problem of finding $s$ to minimize $C(s)$ can be modeled as set cover: the players are elements and each machine is represented by the set of players it can serve. So, the objective is to find the set of machines minimum cost that covers all the players. There is a classic $O(\log n)$ greedy approximation algorithm for this problem: while there are elements that are not covered, pick the set that has the smallest ratio of cost to number of uncovered elements.

We show that the outcome of this greedy algorithm is the unique subgame perfect equilibrium of this game. To solve the game, we calculate for $t = n, n-1, \ldots, 1$ the best move player $t$ has on each node, a unique move by the generic costs assumption. Now, we show that in the backwards-induction solution, all players play according to the greedy algorithm. Let $r_1, r_2, \ldots, r_k$ be the machines in the order picked by the greedy algorithm and let $N_j$ be the players that were first allocated to machine $r_j$.

To show the greedy outcome is the backwards-induction solution it suffices to show that no player wants to deviate on its turn. First, consider the players in $N_1$. They have cost $c_{r_1}/|N_1|$. Notice that this is the smallest cost any player can incur, so the last in $N_1$ to play will definitely choose $r_1$, given that the previous players in $N_1$ did so. Now, consider the second to last player in $N_1$. Given that all previous players in $N_1$ played $r_1$, he also prefers to play $r_1$, since he knows that by doing so, the last player will play $r_1$ too, giving him cost $c_{r_1}/|N_1|$. Continuing this argument, it is easy to see that all players in $N_1$ will choose $r_1$ regardless of what the players outside $N_1$ do.

Now we look at the players in $N_2$. Since we proved that all the players in $N_1$ will choose $r_1$ regardless of what all the other players do, by the definition of the greedy algorithm the best possible cost for players in $N_2$ is $c_{r_2}/|N_2|$. Again we can employ the same argument: the last player in $N_2$ will choose $r_2$ given that the previous players did so. The second to last player will choose $r_2$ if the previous players did so, since he knows the next player will do so, etc. ■

Syrgkanis [24] has recently shown the outcome of the greedy algorithm for this problem is a high quality Nash equilibrium. Our result strengthens this by showing that the outcome of the greedy algorithm is also a subgame perfect equilibrium of the sequential game.

**Observation 2** *The subgame perfect equilibrium outcome is independent of the order in which the players move. Moreover, the players don't need to know the order in which the rest of the players act to find their optimal move. Consequently, the subgame perfect equilibrium is also a Nash equilibrium of the game.*

Notice that the greedy algorithm is still well-defined for general decreasing cost functions: at each moment pick the machine $r$ with minimum $\gamma_r(d_r)$, where $d_r$ is the number of uncovered players that can be allocated to that machine. The outcome of the greedy algorithm still captures the backwards induction solution (which is unique in case of no ties). In this more general case using the results in [24] we get that the social cost of any SPE is at most the potential of the optimal outcome. Hence, for the more general case of cost functions our result implies that the SPoA is at most the best upper bound on the PoS that could be derived by the Potential Method [2].

**Theorem 3** *For machine cost sharing games with arbitrary decreasing cost functions the social welfare of any* SPE *is at most the potential of the optimal solution.*

## 4. UNRELATED MACHINE SCHEDULING

Consider a set $M$ of $m$ unrelated machines and $n$ players each holding a job $j$. Let $t_{ji}$ be the processing time of job $j$ in machine $i$. We consider the classical optimization problem associated with this setting: assign jobs to machines so as to minimize the makespan $\max_{i \in M} \sum_{j;\phi(j)=i} t_{ji}$. Lenstra, Shmoys and Tardos [19] give a 2-approximation to this problem based on rounding the linear programming solution.

Here, we consider the game-theoretical version of this problem. Each player (job) has as action space the set of machines and as utility function the load of the machine he is in. A traditional ordinal potential function argument [25] shows that a pure Nash equilibrium always exists. However, the makespan of a Nash equilibrium of this game can be arbitrarily worse then the optimal makespan. The traditional example is two jobs and two machines where $t_{11} = t_{22} = 1$

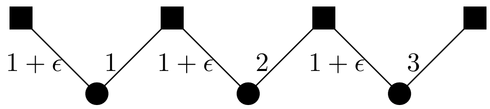


**Figure 1: Price of Anarchy $O(m)$ when jobs (circles) arrive from left to right.**

and $t_{12} = t_{21} = L \gg 1$. One Nash equilibrium is job 1 in machine 2 and job 2 in machine 1. It is easy to see this bad example is easily avoidable when players act sequentially.

Here, we show an upper bound of $O(m \cdot 2^n)$ for the Sequential Price of Anarchy and a lower bound of $\Omega(n)$. It is an improvement that the bound depends only on $(n, m)$ and not on the numerical data of the problem. The example with $\Omega(n)$ SPoA is a generalization of the example given in Figure 1 when jobs (circles) arrive from left to right. The only SPE is always to choose their right option, which ends up with makespan 3 (and makespan $n$ in general), instead of $1+\epsilon$, which is the optimal. Now, we show an upper bound on the SPoA:

**Theorem 4** *The* SPoA *for unrelated machines scheduling is bounded by $O(m \cdot 2^n)$.*

Proof. Let $\overrightarrow{L_0}$ be a vector in $\mathbb{R}_+^M$ representing an initial load on each of the machines, $\textsc{Spe}(\overrightarrow{L_0}, k)$ be the makespan of the SPE we get when players $k, k+1, \ldots, n$ play starting from load $\overrightarrow{L_0}$ and let $t_j^* = \min_{i \in M} t_{ji}$. We will do induction on $k$ from 1 to $n$ using the induction hypothesis that:

$$\forall \overrightarrow{L_0} \in \mathbb{R}_+^M : \textsc{Spe}(\overrightarrow{L_0}, k) \leq \|\overrightarrow{L_0}\|_\infty + 2^{n-k} \sum_{j=k}^{n} t_j^*$$

Then the theorem follows by taking $\overrightarrow{L_0} = \overrightarrow{0}$ and $k = 1$ and noticing that $\sum_{j=1}^n t_j^*$ is smaller than $m$ times the optimal makespan.

Now we proceed to the induction. For $k = n$, this is trivial, because if just one player plays, he definitely will choose an option of optimal makespan and we know that the machine on which he has weight $t_n^*$ will lead to makespan at most $\|\overrightarrow{L_0}\|_\infty + t_n^*$.

Suppose the hypothesis holds for $k+1, \ldots, n$. Player $k$ has the option of playing the machine in which he has $t_i^*$ load. Let $\overrightarrow{L_1}^*$ be the load on the machines after such a move. Apparently, $\|\overrightarrow{L_1}^*\|_\infty \leq t_k^* + \|\overrightarrow{L_0}\|_\infty$. Moreover, by the induction hypothesis, the makespan and thereby player $k$'s cost in the end is at most $\|\overrightarrow{L_1}^*\|_\infty + 2^{n-k-1} \sum_{j=k+1}^n t_j^*$ which is at most $t_k^* + \|\overrightarrow{L_0}\|_\infty + 2^{n-k-1} \sum_{j=k+1}^n t_j^*$.

Now, if player $k$ chooses some other machine $i$ then it must be that it yields for him a smaller or equal cost. Let $\overrightarrow{L_1}$ be the load vector after $k$ plays machine $i$. Player $i$'s cost on $i$ is at least $L_1^i$. Hence,

$$L_1^i \leq t_k^* + \|\overrightarrow{L_0}\|_\infty + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^*$$

and since $L_1^{i'} = L_0^{i'}$ for any other machine $i'$, we get that $\|\overrightarrow{L_1}\|_\infty \leq \|\overrightarrow{L_0}\|_\infty + 2^{n-k-1} \sum_{j=k}^n t_j^*$. Therefore:

$$\begin{aligned} \textsc{Spe}(\overrightarrow{L_0}, k) &= \textsc{Spe}(\overrightarrow{L_1}, k+1) \leq \\ &\leq \|\overrightarrow{L_1}\|_\infty + 2^{n-k-1} \sum_{j=k+1}^{n} t_j^* \leq \\ &\leq \|\overrightarrow{L_0}\|_\infty + 2^{n-k} \sum_{j=k}^{n} t_j^* \end{aligned}$$

■

## 5. CONSENSUS AND CUT GAMES

In consensus and cut games we consider $n$ players which are vertices of a given weighted graph $G = (V, E, w)$, where $w : E \to \mathbb{R}_+$. The action set of each player is binary: $A_i = \{R, B\}$, which corresponds to choosing a color (red and blue). **Consensus Games** are cost games where the cost of player $i$ is the sum of weights of edges from $i$ to players of different color. **Cut Games**, are utility games where the utility of player $i$ is the sum of weights of edges from $i$ to players of a different color. We say that the weight vector $w$ is *generic*, if no weight $w_i$ is 0. Any weight vector $w$ can be made generic with a small random perturbation

In consensus games, the optimal outcome corresponds to every player choosing the same color. In the simultaneous version, it is easy to see that there are instances admitting non-optimal Pure Nash Equilibria. However, we observe the following for the sequential version:

**Observation 5** *The unique* Spe *in generic consensus games is the optimal outcome.*

We also study the closely related cut games. It is well known that the simultaneous PoA for Pure Nash Equilibria is 2. Here we show that for this class of games sequential rationality does not improve the price of anarchy. We show an upper bound of 4 on the SPoA (notice that the Spe might not be a pure Nash equilibrium, so the bound of 2 doesn't necessarily carry over), and a lower bound of 2.

**Theorem 6** *The* SPoA *of sequential cut games is at most* 4.

Proof. Consider the players in the order they arrive. Let $E_k = \{(i,k) | i < k\}$ be the set of edges of player $k$ to all players that arrived previously.

Let $A, B$ be the two partitions of the nodes in the Spe. Consider the decision problem of player $k$. Wlog we can assume that the weight of the edges of player $k$ to predecessor players in partition $A$ is more than that to predecessor players in partition $B$: $w(E_k \cap A) \geq w(E_k \cap B) \geq \frac{1}{2} w(E_k)$. Hence, the utility of player $k$ when choosing $B$ is at least $\frac{1}{2} w(E_k)$. Thus, $u_k(\textsc{Spe}) \geq \frac{1}{2} w(E_k)$. Summing up over all $k$, we get:

$$2\textsc{Spe} = \sum_k u_k \geq \frac{1}{2} \sum_k w(E_k) = \frac{1}{2} w(E) \geq \frac{1}{2} \textsc{Opt}$$

■

However, we conjecture that the true Sequential Price of Anarchy for Cut Games is 2. Moreover, as the following example shows it cannot be better than 2.

**Example.** In the following example we show that the SPoA is at least 2. We consider the Sequential Cut Game that is implied by the following symmetric (almost bipartite) weighted graph:

$$W = \begin{bmatrix} 0 & \epsilon & 1 & 1 \\ \epsilon & 0 & 1+\epsilon & 1+\epsilon \\ 1 & 1+\epsilon & 0 & 0 \\ 1 & 1+\epsilon & 0 & 0 \end{bmatrix}$$

The unique Spe of the above game is for players $1, 3$ and 4 to go to one partition and player 2 go to the other. This leads to a social welfare of $2+3\epsilon$. The optimal is for players 1 and 2 to go to one partition leading to a social welfare of $4+2\epsilon$. ■

# 6. COMPLEXITY OF COMPUTING A SPE

In this section we address the complexity of computing a Spe outcome in the games that we study and generally in congestion games. Specifically we show PSPACE completeness for Unrelated Machine Scheduling and for General Congestion Games. Our proofs are based on reductions from the Quantified Boolean Formula problem and the main technical aspect of them is the idea of simulating NAND circuits with our games. Our Unrelated Machine Scheduling reduction introduces a novel simulation of NAND circuits, while the general congestion games proof uses techniques from Skopalik et al. [23].

**Theorem 7** *Computing the outcome of an* Spe *in Unrelated Machine Scheduling is PSPACE-complete.*

Proof. We will prove completeness via a reduction from the Quantified Boolean Formula (QBF) problem, which is the most basic PSPACE-complete problem. QBF asks whether a quantified form over a set of Boolean variables

$$\exists x_1 \forall x_2 \dots Q_n x_n \phi(x_1, \dots, x_n)$$

is true or false. It is easy to see that computing the outcome of a SPE of a succinctly represented sequential zero-sum game is PSPACE-complete. We can achieve this by creating a player for each variable in the QBF. Players play in the order that their associated variables appear in the quantification of the QBF instance. The strategy of each player is a boolean assignment to his variable. Given a strategy profile the utility function of each player associated with an existential quantifier is 1 if at the end of the game the resulting boolean assignment is a satisfying assignment for $\phi$ and $-1$ otherwise. The utility of players associated with universal quantifiers is the opposite. If one could compute the SPE outcome then he could derive whether QBF is true or false according to whether the resulting strategy profile is a satisfying assignment.

Now the main problem of our reduction is simulating the type of utility function described above through an Unrelated Machine Scheduling (UMS) game. In other words given a boolean formula we have to create a UMS instance such that the players controlling existentially quantified variables will have a higher cost when the outcome of the formula corresponding to the current strategy profile is true and lower otherwise. Respectively for the players controlling universally quantified variables. Wlog we can assume that the QBF instance is given in prenex normal form and that the boolean formula consists of only NAND operations (Any QBF instance can be transformed to the above form in polynomial time).

The main idea of the reduction is the following: We will create a player for each boolean variable. Each such input player will have two possible machines he can be assigned to. His 0 and his 1 machine, each representing the corresponding boolean assignment of the variable controlled by the player. We will then create circuit players and machines such that given the strategies chosen by the input players, in the only dominant strategy remaining the circuit players simulate the circuit semantics. Hence, the last circuit player will play his 0 strategy if the outcome of the boolean circuit/formula is 0 and 1 otherwise. Also he will trigger some feedback players to increase the load on the machines of the players controlling existentially quantified variables if the outcome is 0 and the load on the machines of those controlling universally quantified variables otherwise.

We now move to the details of the reduction. We first describe how to simulate the NAND semantics with UMS game. In Figure 2 we depict the simulation of a NAND gate whose output can become the input of $k$ other NAND gates at a next level of the circuit tree. Players $X$ and $Y$ are the input players. We assume that the input players are fixed (e.g. because they are the outcomes of a NAND gate of a previous level or they are global input players). Moreover, we assume that the leftmost machine of every player is his 1 strategy and the rightmost machine is his 0 strategy.

- If both players $X, Y$ are playing 1, then player $A$ will also play 1. Now player $B$ has cost $2\alpha - \epsilon$ on his 0 strategy and at most $2\alpha - 2\epsilon$ on his 1 strategy. Thus $B$ will also play 1. Now $C$ has cost $2\alpha - 2\epsilon$ on his 1 strategy and $2\alpha - 3\epsilon + \epsilon/2$ on his 0. Thus $C$ will play his 0 strategy. The output players will thus have $\epsilon$ cost on their 0 strategy and at least $\alpha - 2\epsilon$ on their 1 strategy. Thus all output players $O_i$ will player their 0 strategies.

- If any of $X, Y$ is player his 0 strategy then $A$ has cost at most $\alpha$ on his 0 strategy and cost $2\alpha - \epsilon$ on his 1. Thus $A$ will play 0. Now $B$ has cost 0 on his 0 strategy and $\epsilon$ on his 1. Thus $A$ will play 0. $C$ has cost $2\alpha - 3\epsilon$ on his 1 strategy and $2\alpha - 3\epsilon + \epsilon/2$ on his 0. Thus $C$ will play 1. Now all the outcome players have cost $2\alpha - 3\epsilon$ on their 0 strategies and cost at most $2\alpha - 4\epsilon$ on their 1 strategy. The latter is because by our overall construction it is easy to see that at most 2 players occupy any machine and also any player connected to a machine of the next level has weight on that machine at most $\alpha - 2\epsilon$ (we can achieve this by transforming the circuit such that all players of level $k$ gates are output players of level $k-1$ gates). Thus the output players $O_i$ will player their 1 strategy.

Thus interconnecting the above NAND gadgets such that they simulate the circuit we can have a UMS game such that, given what the input players have played, the unique subgame perfect equilibrium simulates the circuit semantics.

Next we describe how the variable players are connected to their NAND gates. Each input player has two strategies $0, 1$ each of them having equal weight of 1. If a variable player is connected to $k$ NAND gates then we connect $k$ output players to the 1 machine of the variable player with

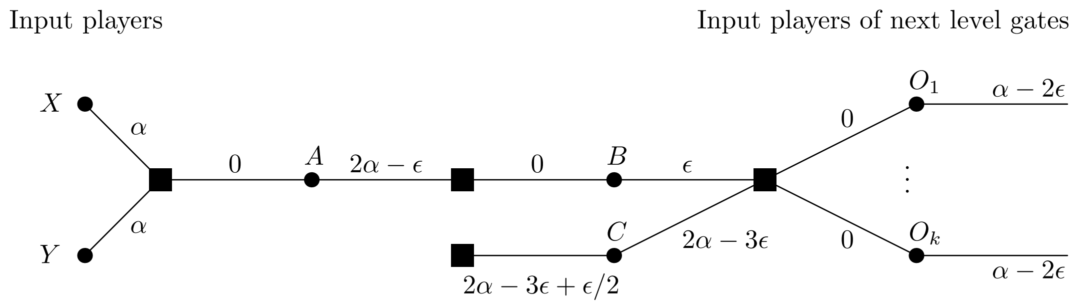


**Figure 2: Simulating a NAND gate with selfish unrelated machine scheduling.**

a weight of 0. Those output players are then connected to their NAND machines with a weight of $1/2 - \epsilon$. Thus the NAND gates of the first level of the circuit have $\alpha = 1/2 - \epsilon$. Hence, the output players of the last level NAND gate of the circuit will have weight on their 1 strategy of $1/2-(2k+1)\epsilon$. Moreover, their 0 machine will have a cost of $1-(4k+1)\epsilon+\epsilon$.

Now we move on to the gadget that gives the essential feedback that creates the incentive for input players controlling existential quantified variables to make the output of the circuit 1. This is depicted in Figure 3. In this figure we show how the output players of the last gate are connected with the machines of the variable players. We may assume that we have both the output of the circuit and its negation, since we can produce both these values using an extra level of gates. For each variable player that controls an existentially quantified variable we have two output players that come from the negation of the circuit output. For each variable player that controls a universally quantified variable we have two output players that come from the circuit output. In the picture we depict the interconnection of each such variable player with his corresponding two output players. We assume that each $F_i$ player is playing before each corresponding $I_i$.

- If the output circuit (corr. its negation) is 1 then the $O_1, O_2$ are triggered to play their 1 strategy because their zero strategy has cost $1 - (4k + 1)\epsilon + \epsilon$, where $k$ is the depth of the circuit, while if they play their one strategy then each $I_i$ will choose his 1 strategy later on, leading to a cost of $1 - (4k + 1)\epsilon)$ for them. This will cause the input player to have cost $1+\epsilon$ no matter which strategy he plays.
- If the output is 0 then $O_1, O_2$ player their 0 strategy. If the input player is on his 1 strategy then $I_1$ will play his 0 strategy and $I_2$ will play his 1. If the input player is on his 0 strategy then $I_0$ will play his 0 strategy and $I_1$ his 1. In any case the input player incurs a cost of 1 no matter which strategy he plays.

■

**Theorem 8** *Computing a* Spe *in Congestion Games is PSPACE-complete.*

**Proof Sketch.** The proof follows similar lines as the previous theorem. The construction of Skopalik and Vocking [23] allows us to simulate a forward NAND circuit with a congestion game. Unlike the simultaneous case in the sequential version we can create a feedback. Although the output players of the Skopalik et al. [23] construction occupy resources with exponentially smaller congestion levels than the input players, we just need to make them occupy an extra small congestion shared resource with the input players. This way we can again cause the $X$ players to incur an $\epsilon$ extra cost when the output of the circuit is 1 and the $Y$ players when the output is 0. ■

# 7. CONCLUSIONS AND OPEN PROBLEMS

In this work we showed how sequentiality can have a very positive impact on the quality of outcomes for several natural and well-studied classes of games. The main open problem is to extend our analysis for even bigger classes of games. It is easy to see that for general potential games one can create pathological examples that will make the sequential version behave arbitrarily worse. However, we believe the merits of sequentiality will carry over to natural subclasses of potential games. We consider as a very interesting direction the case of general cost sharing games, where we believe that the sequential price of anarchy is still exponentially better than it's simultaneous counterpart.

Another interesting direction is to show classes of games where the subgame perfect equilibrium of the sequential version is a pure Nash equilibrium of the simultaneous version. In those classes of games our technique would be a very natural equilibrium refinement solution. We showed that such a property holds for machine cost-sharing games. In general it is easy to see that if the subgame perfect equilibrium outcome doesn't depend on the ordering of the players' arrival then it is always a Nash Equilibrium. It is interesting to see whether other properties lead to such a conclusion.

**Acknowledgements.** We would like to thank Alexander Skopalik for useful comments.

# 8. REFERENCES

[1] N. Andelman, M. Feldman, and Y. Mansour. Strong price of anarchy. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 189–198, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[2] E. Anshelevich, A. Dasgupta, J. M. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The Price of Stability for Network Design with Fair Cost Allocation. In *45th Symposium on Foundations of Computer Science (FOCS 2004)*, pages 295–304. IEEE Computer Society, 2004.

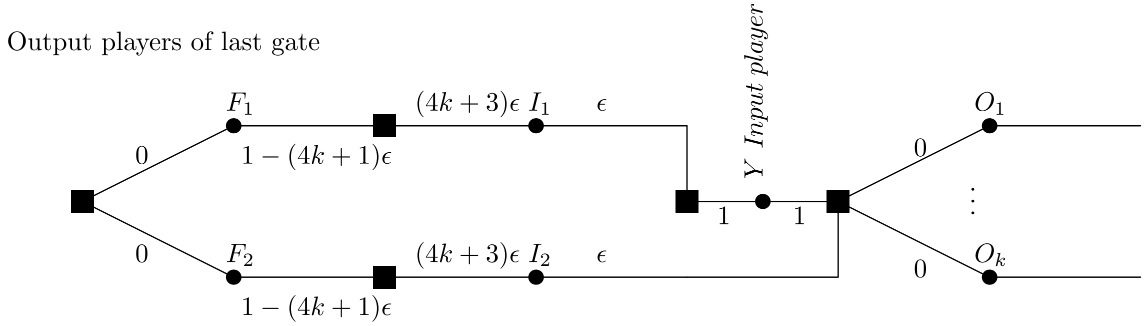


**Figure 3: The effect that the final output has on the input players. If the final output is $1$ then the $Y$ input players receive an extra $\epsilon$ job on the machine that they occupy. If it is $0$ then they don't. The inverse happens for the $X$ input players. This can be done with the exact same way by just inverting the outcome of the circuit using an extra NAND gate.**


[3] J. Bae, E. Beigman, R. Berry, M. L. Honig, and R. Vohra. On the efficiency of sequential auctions for spectrum sharing. *2009 International Conference on Game Theory for Networks*, pages 199–205, May 2009.

[4] M. Balcan, A. Blum, and Y. Mansour. The price of uncertainty. *EC'09*, 2009.

[5] M. F. Balcan. Leading dynamics to good behavior. *SIGecom Exch.*, 10:19–22, June 2011.

[6] M.-F. Balcan, A. Blum, and Y. Mansour. Circumventing the price of anarchy: Leading dynamics to good behavior. In *ICS*, pages 200–213, 2010.

[7] M. Charikar, H. Karloff, C. Mathieu, J. Naor, and M. Saks. Online multicast with egalitarian cost sharing. In *SPAA 2008: Proceedings of the 20th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 70–76. ACM, 2008.

[8] C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. Naor, and A. Orda. Non-Cooperative Multicast and Facility Location Games. *IEEE Journal on Selected Areas in Communications*, 25(6):1193–1206, 2007.

[9] C. Chung, K. Ligett, K. Pruhs, and A. Roth. The price of stochastic anarchy. In *Proceedings of the 1st International Symposium on Algorithmic Game Theory*, SAGT '08, pages 303–314, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] A. Epstein, M. Feldman, and Y. Mansour. Strong equilibrium in cost sharing connection games. In *8th ACM Conference on Electronic Commerce (EC-2007)*, pages 84–92. ACM, 2007.

[11] A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, STOC '04, pages 604–612, New York, NY, USA, 2004. ACM.

[12] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

[13] P. Goldberg, C. Papadimitriou, and R. Savani. The complexity of the homotopy method, equilibrium selection, and lemke-howson solutions. *FOCS 2011*, 2011.

[14] J. Harsanyi and R. Selten. A general theory of equilibrium selection in games. *Cambridge:MIT Press*, 1988.

[15] T. Hart and D. Edwards. The tree prune (tp) algorithm. Technical report, M.I.T. Artificial Intelligence Project Memo # 30, 1961.

[16] P.-J. Herings and R. Peeters. Homotopy methods to compute equilibria in gametheory. *Economic Theory, 42(1)*, 2010.

[17] D. Knuth and R. Moore. An analysis of alpha-beta pruning. *Artificial Intelligence*, page 293âĂŞ326, 1975.

[18] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[19] J. K. Lenstra, D. B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.*, 46:259–271, February 1990.

[20] A. Montanari and A. Saberi. Convergence to equilibrium in local interaction games. *SIGecom Exch.*, 8:11:1–11:4, July 2009.

[21] R. Paes Leme, V. Syrgkanis, and Éva Tardos. Sequential auctions and externalities. In *Proceedings of the 23rd ACM Symposium on Discrete Algorithms (SODA'12)*. 2012, 2012.

[22] D. Parkes. Online Mechanisms. *Algorithmic Game Theory (eds, N. Nisan et al)*, 2007.

[23] A. Skopalik and B. Vöcking. Inapproximability of pure nash equilibria. *Proceedings of the fourtieth annual ACM symposium on Theory of computing - STOC '08*, page 355, 2008.

[24] V. Syrgkanis. The Complexity of Equilibria in Cost Sharing Games. In *WINE 2010: Proceedings of the 6th Workshop on Internet and Network Economics*, pages 366–377, 2010.

[25] B. Vocking. Selfish load balancing. In *Algorithmic Game Theory*, chapter 20. Cambridge University Press, 2007.

[26] J. von Neumann. Zur Theories der Gesellschaftsspiele. *Mathematische Annalen 100:295-320, 1928, English translation by S. Bergmann in R. D. Luce, A. W. Tucker (eds.), Contributions to the Theory of Games IV (1959), pp. 13-42, Princeton U. Press.*

[27] E. F. F. Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In *Fifth International Congress of Mathematicians, vol. II*, pages 501–504, 1913.