

The Quality of Equilibria for Set Packing and Throughput Scheduling Games

Jasper de Jong · Marc Uetz

Received: date / Accepted: date

Abstract We introduce set packing games as an abstraction of situations in which n selfish players select disjoint subsets of a finite set of indivisible items, and analyze the quality of several equilibria for this basic class of games. Special attention is given to a subclass of set packing games, namely throughput scheduling games, where the items represent jobs, and the subsets that a player can select are those jobs that this player can schedule feasibly. We show that the quality of three types of equilibrium solutions is only moderately suboptimal. Specifically, the paper gives tight bounds on the price of anarchy for Nash equilibria, subgame perfect equilibria of games with sequential play, and k -collusion Nash equilibria. Under the assumption that players are allowed to play suboptimally and achieve an α -approximate equilibrium, our tight price of anarchy bounds are $\alpha + 1$ for Nash and subgame perfect equilibria, but less than $\alpha + 1/(e - 1)$ for subgame perfect equilibria when games are symmetric. For k -collusion Nash equilibria, the price of anarchy equals $\alpha + (n - k)/(n - 1)$, where $1 \leq k \leq n$.

Keywords Set Packing · Throughput Scheduling · Price of Anarchy

1 Introduction & Main Contribution

This paper addresses a problem where n players compete for a set of indivisible items $j \in J$. Each item $j \in J$ has a weight w_j which represents the item's utility. The utility of an item is the same for all n players. In competing for these items, a player i is bound in her choice by a downward-closed set system $\mathcal{S}_i \subseteq 2^J$, meaning that only the subsets $S_i \in \mathcal{S}_i$ are feasible for her. Players aim to maximize their utility, which means maximizing the total weight of the chosen items $w(S_i) = \sum_{j \in S_i} w_j$. Items are indivisible, hence a feasible solution consists of a profile of

Jasper de Jong
Dept. Applied Mathematics, University of Twente
E-mail: j.dejong-3@utwente.nl

Marc Uetz
Dept. Applied Mathematics, University of Twente
E-mail: m.uetz@utwente.nl

subsets $S = (S_1, \dots, S_n)$ such that $S_i \in \mathcal{S}_i$ for all i and $S_i \cap S_k = \emptyset$ for all $i \neq k$. Our interest goes into the efficiency loss that occurs when the distribution of items is not controlled centrally, and hence we study the price of anarchy for three types of equilibria for this class of games. The motivation to address this problem is discussed in detail in Section 2.

Nash equilibria and Sequential Play. A distinguishing feature of the game that we propose is that each item $j \in J$ can only be selected by *at most one of the players*. In that situation, a pure strategy Nash equilibrium is a selection of subsets S_i , one for every player $i = 1, \dots, n$, so that $S_i \cap S_k = \emptyset$ for any two players $i \neq k$, and for each player i we have that $w(S_i) \geq w(T_i)$ for all $T_i \in \mathcal{S}_i$ with $T_i \subseteq J \setminus (\cup_{k \neq i} S_k)$. In words, given the items S_k selected by players other than i , among the remaining items $J \setminus \cup_{k \neq i} S_k$, player i selects a feasible subset of maximum weight. When interpreted as a single-shot strategic form game, to realize this disjointness restriction we can define the payoff of a strategy S_i equal $-\infty$ whenever another player's strategy is S_k with $S_i \cap S_k \neq \emptyset$. While this seems a bit unnatural, our primary interest is to analyze the more natural variant of the game where players select their subsets sequentially. Then, the items selected by previous players are no longer available for subsequent players. We address both problem variants in this paper.

Equilibrium Concepts and Price of Anarchy. For the entire paper, we measure the quality of an (equilibrium) solution $S = (S_1, \dots, S_n)$ by the total value of all selected items, or equivalently, the sum of values of all selected items $w(S) := \sum_{i=1}^n w(S_i)$. The question is by how much an equilibrium solution falls behind an optimal solution that could be computed by some central authority, in the worst case. For a maximization problem as the one considered here, the *price of anarchy* [41, 31] denotes the ratio of the value of an optimal solution over the value of an equilibrium solution. We analyze the price of anarchy for three different equilibrium concepts, namely pure strategy Nash equilibria, subgame perfect equilibria [44] of a sequential version of the game, and a third equilibrium concept that we refer to as k -collusion Nash equilibrium. The idea of the latter is that up to k players may collude and are allowed to use any profit sharing protocol among themselves, hence can be thought of as acting like a single player. This concept also appears in a paper by Hayrapetyan et al. [25]. In fact, k -collusion equilibria are a generalization of k -strong Nash equilibria as defined by Aumann [4]; see also [1].

An illustrating example.

Example 1 Consider $n = 2$ players and two items $J = \{1, 2\}$, with weights $w_1 = w_2 = 1$, and the feasible subsets are $\mathcal{S}_1 = \{\emptyset, \{1\}, \{2\}\}$ and $\mathcal{S}_2 = \{\emptyset, \{2\}\}$. \triangleleft

In an optimal solution, player 1 chooses the first and player 2 chooses the second item, that is, the strategy profile is $S = (S_1, S_2) = (\{1\}, \{2\})$ with payoffs $(1, 1)$ and $w(S) = 2$. This allocation is a Nash equilibrium. There is another Nash equilibrium, namely $S = (\{2\}, \emptyset)$, because $\{1\} \notin \mathcal{S}_2$. As the payoffs are $(1, 0)$ and $w(S) = 1$, this instance has a price of anarchy of 2. The strategic form of this game is depicted in Figure 1.

Considering the sequential game where player 1 precedes player 2, this yields a game tree that is depicted in Figure 2. A strategy for player 2 now specifies

		player 2	
		\emptyset	$\{2\}$
player 1	\emptyset	0,0	0,1
	$\{1\}$	1,0	1,1
	$\{2\}$	1,0	$-\infty, -\infty$

Fig. 1 Strategic form for Example 1 with Nash equilibria in bold.

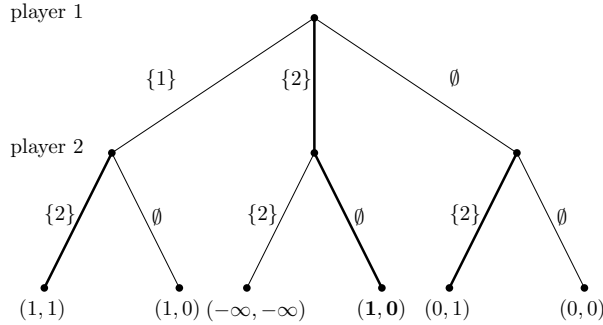


Fig. 2 Game tree for sequential version (player 1 \rightarrow player 2) of Example 1. The worst-case subgame perfect equilibrium is marked bold.

three actions, one for every possible choice of player 1. When the order is $1 \rightarrow 2$, all Nash equilibria of the strategic form game are also obtained as subgame perfect equilibria, namely $(\{1\}, (\{2\}, \emptyset, \{2\}))$ and $(\{2\}, (\{2\}, \emptyset, \{2\}))$, with outcomes $(S_1, S_2) = (\{1\}, \{2\})$ and $(\{2\}, \emptyset)$, and payoffs $(1, 1)$ and $(1, 0)$, respectively. The worst case subgame perfect equilibrium is indicated in bold in Figure 2.

For the reverse order of the sequential game (player 2 \rightarrow player 1), the only subgame perfect equilibria are $(\{2\}, (\{1\}, \{1\}))$ and $(\{2\}, (\{2\}, \{1\}))$, with as unique outcome $(S_1, S_2) = (\{1\}, \{2\})$ and corresponding payoff $(1, 1)$. As the sequential price of anarchy takes the worst case over all possible sequential games, Example 1 has sequential price of anarchy of 2.

Finally, assume that both players collude, then obviously, the only allocation that maximizes their joint payoff is $(S_1, S_2) = (\{1\}, \{2\})$ and corresponding payoff $(1, 1)$. Therefore, the 2-collusion price of anarchy equals 1.

Relation to Set Packing and Maximum Coverage Problems. The set packing problem is one of Karp's 21 problems first proved to be NP-complete [30]; it is listed as problem [SP3] in Garey and Johnson's classic on computers and intractability [19]. In set packing, the task is to select from a given collection \mathcal{S} of subsets of a set of items J , a collection of k disjoint subsets, for a given number k . In the weighted optimization version of the problem each subset $S \in \mathcal{S}$ has a weight $w(S)$, and the goal is to find disjoint subsets of maximum total weight. The general set packing problem with or without weights is NP-complete [19], and in general it is not even approximable to within any constant. We refer to [14] and [26] for more details. The problem that is considered in this paper corresponds to the much simpler variant of set packing where items $j \in J$ have weights w_j , and $w(S) := \sum_{j \in S} w_j$. Due to our assumption that the set system is downward-closed, as an instance of maximum weight set packing this problem is trivial when there is no limit on the number of chosen subsets (e.g., take all singletons in $\cup_{i=1}^n \mathcal{S}_i$). However if only a

given number of n subsets may be chosen -which is the problem that we consider- the downward closedness yields that the problem is equivalent with the the NP-hard maximum coverage problem. That problem can be approximated to within a factor $1 + 1/(e - 1)$ [27], and the same approximation guarantee is recovered in this paper as the sequential price of anarchy for symmetric set packing games.

Throughput Scheduling. A subclass of set packing games are throughput scheduling games. These games were the subject of an extended abstract underlying this paper [28]. Throughput scheduling has been studied from the algorithmic perspective e.g. by Bar-Noy et al. [7] and Berman and Dasgupta [8]. Here, the set of items J corresponds to a set of non-preemptive jobs, each with a release time r_j , a due date d_j , and a weight w_j . There are $m \geq 1$ machines to process these jobs. In the most general setting, the machines can be unrelated, meaning that the processing time of any job j may depend on the machine ℓ it is processed on, and the $\ell \times m$ matrix $(p_{\ell j})$ of processing times on machines can have rank > 1 . A subset S_i of jobs is feasible for player i if there exists a schedule of the jobs in S_i on the set of machines of player i , so that each job can be processed in the time window $[r_j, d_j]$. The job sets S_i that are feasible for player i is then a downward closed set system. Except one, all lower bounds on the price of anarchy for set packing games are presented as instances of throughput scheduling games.

Approximate Equilibria. We generally consider α -approximate versions for all three equilibrium concepts, for any fixed $\alpha \geq 1$. That means that a strategy profile is considered in α -approximate equilibrium if the corresponding payoffs per player are maximal up to a multiplicative constant α , given the strategies of the other players. Indeed, the idea of considering approximate instead of exact equilibria is by now a widely accepted concept in algorithmic game theory, with different variations. Already Roughgarden and Tardos [43] consider it for network routing games, and for example Skopalik and Vöcking [45] discuss corresponding hardness results in the context of congestion games. Approximate equilibria can be generally motivated in at least two ways: The first is that players may be reluctant to changing strategies for small improvements, and the second is that players may be computationally bounded, and restricted to approximate equilibria by the computational hardness of computing a best response. We come back to this point later. It is important to note that our price of anarchy bounds are tight for all α -approximate versions of the three equilibria that we consider.

Main Contribution. Our contribution is as follows. If all players play α -approximate Nash equilibria, the price of anarchy for set packing games equals $\alpha + 1$. We also show that sequential play, that is the game where players select their sets subsequently, does not allow to improve on that bound in general. However, for the special case of *symmetric* set packing games (to be defined later), sequential play *does* yield improved results for the price of anarchy: We show that subgame perfect equilibria yield a price of anarchy of $\sqrt[e]{e}/(\sqrt[e]{e} - 1)$, which is tight, too. Note that for $\alpha = 1$, this equals $1 + 1/(e - 1)$. Finally, we analyze k -collusion Nash equilibria, and show that when players are assumed to play α -approximate k -collusion Nash equilibria, the price of anarchy interpolates between α and $\alpha + 1$, as it is equal to $\alpha + (n - k)/(n - 1)$.

2 Related Work and Motivation

Throughput scheduling games have been the subject of the extended abstract [28] underlying this full-length paper. We are not aware that set packing games as defined in this paper have been addressed in the literature. Generally spoken, much of the work in algorithmic game theory addresses auctions, network routing or congestion games, as well as scheduling and load balancing games. Refer e.g. to Section III of the textbook by Nisan et al. [36] for several of such models.

Covering Games. There is a set of publications on so-called covering games that are very closely related to our work. When allowing that one item is chosen by more than only one player, the resulting game is a covering game as studied by Gairing [18]. In this case, one needs to define a utility sharing function for the items that are chosen by several players. Gairing shows that there exist utility sharing functions such that worst case Nash equilibria are at most a factor $1+1/(e-1)$ away from the optimum. Covering games can also be interpreted as a generalization of market sharing games as studied e.g. by Goemans et al. [20]. For covering games with uniform utility sharing, the price of anarchy equals $2 - 1/n$ for n -player games [18, 11]. That result can also be obtained by a smoothness argument, so it extends to more than Nash equilibria [11]. In that context, note that the price of anarchy bound for set packing games as proved in this paper is 2, which is tight for any number of players $n \geq 2$. In contrast to covering games, a distinguishing feature of set packing games is the property that items can only be chosen by one player, and as a consequence, rational players affect each other only through the *availability* of strategies from the strategy set \mathcal{S}_i , but not each other's payoffs. As a result, there are Nash equilibria of the non-sequential game that cannot be realized as subgame perfect equilibria of a sequential game, and this feature allows us to obtain improved price of anarchy results for subgame perfect equilibria. This is indeed no longer the case when items are uniformly shared: Generalized market sharing games have a price of anarchy lower bound of $2 - 1/n$ even for subgame perfect equilibria of a sequential game [11]. Finally, we note that generalizations of covering games have also been addressed recently by Paccagnan et al. [38, 39].

Sequential price of anarchy. The analysis of subgame perfect equilibria as opposed to Nash equilibria is one of the main contributions of this paper, and it is maybe also the most interesting result from an applications point of view. At the time of writing the conference publication underlying this full-length paper [28], the idea of considering sequential versions of games, and Selten's subgame perfect equilibria as an alternative to avoid the "curse of simultaneity" of Nash equilibria had just been brought up by Paes Leme et al. [40]. In contrast to the price of anarchy which relates the outcome of the worst possible Nash equilibrium to that of an optimal solution [41, 31], the *sequential price of anarchy* [40] relates the outcome of the worst possible subgame perfect equilibrium of all sequential versions of the game where players act subsequently (and farsighted), to the outcome of an optimal solution. For set packing games, it is not hard to see that any outcome of a subgame perfect equilibrium of a sequential version of the game is also a Nash equilibrium in the non-sequential, strategic form of the game, but not necessarily vice versa; see also Theorem 6. It should be noted that subgame perfect equilibria

of a sequential game are not generally Nash equilibria of the corresponding non-sequential game: Correa et al. [16] give an instance of a network routing problem where the sequential price of anarchy is unbounded, while the price of anarchy is known to be $5/2$ [5, 15]. Indeed, subsequent to the work by Paes Leme et al. [40], for a handful of problems it was shown that the sequential price of anarchy is lower than the price of anarchy [40, 28, 29, 24], while for others this is exactly opposite [2, 9, 16]. To conclude, it should be noted that also earlier than [40], sequential versions of (routing) games have been considered, however with different utility sharing functions. This includes Olver [37], Harks, Heinz and Pfetsch [22], Harks and Vegh [23], as well as Farzad, Olver and Vetta [17].

Approximate equilibria and hardness of optimal play. The necessity to also consider α -approximate equilibrium concepts is best motivated by considering some concrete examples of throughput scheduling. For example, consider the special case where the feasibility system of a player i consists of all sets of jobs $j \in J$ that can be feasibly scheduled on a single machine. In the 3-field notation of [21], this problem reads $1|r_j|\sum w_j U_j$, where “1” stands for one single machine, r_j specifies that there are release dates, and the objective $\sum w_j U_j$ is to maximize the total weight of jobs scheduled before their due date d_j . In that case, the input of the problem would not be a list of all feasible sets \mathcal{S}_i , but only the set of jobs $j \in J$ with their time windows $[r_j, d_j]$, processing times p_j and values w_j . It follows from [32] that the problem to compute a subset of jobs maximizing the total weight is NP-hard, hence players cannot generally be assumed to be able to compute a strategy to optimize their payoff. More generally, if players control a set of several (unrelated) machines each, the problem to compute a best response reads $R|r_j|\sum w_j U_j$ in scheduling notation, where “R” stands for unrelated machines. This problem is equivalent to the throughput scheduling problem as it has been addressed e.g. by Bar-Noy et al. [7], and subsequently by Berman and Dasgupta [8]. For this problem, only constant factor approximation algorithms are available, and this constant is essentially 2; see [8]. Hence, assuming that players are equipped with such approximation algorithms, the outcome of such a game would be an α -approximate equilibrium with $\alpha = 2$. On the other hand, there are relevant special cases of throughput scheduling where players *can* be assumed to be able to play optimally. One such case is when feasibility sets \mathcal{S}_i are the sets of jobs that can be feasibly scheduled on a single machine, and jobs have unit weights and zero release dates. This single-machine problem is solved in polynomial time by the Moore-Hodgson algorithm [35]. Another case is when the feasibility system \mathcal{S}_i is the set of jobs with unit processing times that can be scheduled on a set of identical, parallel machines. This problem can be solved as an assignment problem [12]. In light of this discussion, recall that all our results are parametric in the parameter α .

Motivation. On the one hand, we believe that the game theoretic version of a classical combinatorial problem such as set packing is interesting in its own right. On the other hand, the original driver of this research was to understand the potential loss of efficiency when the procurement of indivisible services is not controlled centrally. Here, think of the following service procurement setting: Each item corresponds to a service request –such as a computational task or a physical service– with a monetary value that pays off once the service is executed. Given a collection of such service requests, e.g. on an internet portal or within a larger

corporation consisting of several profit centers, players are service providers who can choose a subset of services in order to generate revenue. Service providers, however, are bound by limited resources and can therefore only choose certain subsets of services. Specifically, when these services are specified by a release time, processing time, and a due date, this gives rise to a throughput scheduling problem.

Sketch of Applications. We give three instances of application domains that lead to instances of set packing games. (1) When operating microgrids for decentralized energy production, the goal is to consume locally produced energy as much as possible. Here, the items are the operation of appliances in households (e.g. loading a car battery) which come with a time window and a certain monetary value. Players, on the other side, are intermediaries or local energy producers that want to maximize the total value of items than can be accommodated within a given a profile of available energy; see, e.g. [6] or [34] for more context. (2) In cloud computing, service providers such as Google or Amazon provide an infrastructure service. Here, the items are computational tasks that need to be distributed over data- and computing centers. Indeed, the aim of a federated cloud computing environment, as discussed e.g. in [13], is to “coordinate load distribution among different cloud-based data centers in order to determine optimal location for hosting application services”. Understanding the cost of uncoordinated load distribution is the question that we ask. (3) A final application domain is distribution logistics in a B-to-B context, like reverse auctions. Specifically, consider private car sharing portals like e.g. [47], where items correspond to car rental requests for a certain time period and a predefined price. Owners of cars or car farms are players that select subsets of such requests from the portal to rent out their car(s) in order to maximize the total revenue.

While it is true that each of these applications has additional features or practical constraints that are not taken care of in the set packing games that we address here, the paper aims to identify the combinatorial flavour that lies at the heart of such problems. The overall conclusion is that the loss of efficiency caused by the lack of central coordination is moderate.

3 Preliminaries

We here fix some notation and the basic definitions. There are n players, and a finite ground set J of items. Each item $j \in J$ has a value w_j . For $S \subseteq J$, we let $w(S) := \sum_{j \in S} w_j$. Each player i has a strategy set $\mathcal{S}_i \subseteq 2^J$ which is downward closed, i.e., if $S_i \in \mathcal{S}_i$, then $T_i \in \mathcal{S}_i$ for all $T_i \subseteq S_i$. Given a strategy profile $S = (S_1, \dots, S_n)$, as usual define $S_{-i} := (S_1, \dots, S_{i-1}, S_{i+1}, \dots, S_n)$ as the strategies of all players except i , and for any set of players $K \subseteq \{1, \dots, n\}$, define S_{-K} accordingly.

When (S_1, \dots, S_n) is a strategy profile with $S_i \in \mathcal{S}_i$ for all $i = 1, \dots, n$, the payoffs for player i are defined as

$$w_i(S_i, S_{-i}) = \begin{cases} w(S_i) & \text{if } S_i \cap S_k = \emptyset \text{ for all } k \neq i, \\ -\infty & \text{otherwise.} \end{cases}$$

A strategy profile (S_1, \dots, S_n) is an α -approximate Nash equilibrium (for $\alpha \geq 1$) if it is true that for all players $i = 1, \dots, n$

$$w_i(S_i, S_{-i}) \geq \frac{1}{\alpha} w_i(T_i, S_{-i}) \quad \text{for all } T_i \in \mathcal{S}_i. \quad (1)$$

Note that the existence of Nash equilibria with $w_i(S_i, S_{-i}) \geq 0$ for all players i is guaranteed by the fact that the feasibility systems \mathcal{S}_i are downward closed.

For a solution $S = (S_1, \dots, S_n)$, in a slight but convenient abuse of notation¹ let us write $w(S) := \sum_{i=1}^n w(S_i)$ for the total value that it achieves. The *price of anarchy* PoA [41,31] for a class of games \mathcal{I} is then the ratio

$$(\alpha\text{-approximate}) \text{ PoA} = \sup_{I \in \mathcal{I}} \sup_{S \in NE(I)} \frac{w(OPT(I))}{w(S)}, \quad (2)$$

where $NE(I)$ denotes the set of all α -approximate Nash equilibria of instance $I \in \mathcal{I}$, and $OPT(I)$ is an optimal allocation for I . Note that for set packing games, $OPT(I)$ is a Nash equilibrium too, hence the price of stability as proposed by Anshelevich et al. [3] equals 1.

Next, consider the extensive form game that is obtained when imposing some order, say $1, \dots, n$ on the players. A strategy for player i is then more complex, as it must specify one action S_i for all possible combinations of actions of preceding players $1, \dots, i-1$, that is, one action S_i for each node of the game tree on level i . An α -approximate subgame perfect equilibrium is then a strategy that guarantees at least a $1/\alpha$ -fraction of the optimal action for each of the nodes of the game tree on level i . As we deal with a full information game, (α -approximate) subgame perfect equilibria can be computed via backward induction². A nice feature of set packing games is that the computation of (α -approximate) subgame perfect equilibria is not suffering from the typical hardness results for sequential games that is due to farsighted behaviour of players: Indeed, computing outcomes of subgame perfect equilibria may be PSPACE-hard with n players [40], and NP-hard even with two players only [16]. For set packing games, an optimal action for the i -th player, upon observing the actions S_1, \dots, S_{i-1} of the preceding players, is computed by solving the optimization problem

$$\max_{T \subseteq J} w(T) \text{ s.t. } T \subseteq J \setminus \cup_{k=1}^{i-1} S_k \text{ and } T \in \mathcal{S}_i,$$

This suffices, as by the specific payoff structure of set packing games, the value attained by player i is no longer affected by players $i+1, \dots, n$, as long as they are all rational. This problem is computationally hard only if the combinatorial structure encoded by \mathcal{S}_i is hard; see the discussion in Section 2.

The price of anarchy for α -approximate subgame perfect equilibria, also called *sequential* PoA [40], is then defined analogously to the price of anarchy in (2),

$$(\alpha\text{-approximate}) \text{ sequential PoA} = \sup_{I \in \mathcal{I}} \sup_{S \in SPE(I)} \frac{w(OPT(I))}{w(S)}, \quad (3)$$

¹ We use S to denote both, a strategy vector $S = (S_1, \dots, S_n)$ as well as the total set of items that it induces, i.e., $S = \cup_{i=1}^n S_i$. That will not yield any confusion, however.

² See, e.g., [42]. That is conceptually simple but generally not polynomial time.

where the first supremum $\sup_{I \in \mathcal{I}}$ is also taken over all possible orders of players, and $SPE(I)$ denotes all outcomes that can be obtained as α -approximate subgame perfect equilibria of instance I .

Finally, assume that up to k of the given n players may collude, and are allowed to use any profit-sharing rule among them. In other words, we can think of a group K of up to k players as maximizing their joint value $w(\cup_{i \in K} S_i)$. Then, $w_K(S_K, S_{-K}) := \sum_{i \in K} w_i(S_i, S_{-i})$ is the joint value achieved by players in K , and an α -approximate k -collusion Nash equilibrium is a strategy profile (S_1, \dots, S_n) such that the following is true for all sets K of at most k players,

$$w_K(S_K, S_{-K}) \geq \frac{1}{\alpha} w_K(T, S_{-K}) \quad \text{for all } T = \cup_{i \in K} T_i \text{ and } T_i \in \mathcal{S}_i. \quad (4)$$

Obviously, the price of anarchy for α -approximate k -collusion Nash equilibria is then again defined analogously to the price of anarchy in (2) by

$$(\alpha\text{-approximate}) \ k\text{-collusion PoA} = \sup_{I \in \mathcal{I}} \sup_{S \in CE_k(I)} \frac{w(OPT(I))}{w(S)}, \quad (5)$$

where $CE_k(I)$ denotes the set of α -approximate k -collusion Nash equilibria of instance I .

Symmetric Set Packing. We call a set packing game *symmetric* whenever there is only *one* feasibility system \mathcal{S} that is the same for all players i , and a player i can select $x_i \geq 1$ feasible sets from \mathcal{S} , for some integer $x_i \geq 1$. Note that when all $x_i = 1$, this exactly means that the strategic form game is symmetric in the sense that all players have exactly the same strategy set. However we here choose a slightly more general definition of symmetry, in that we allow players to select multiple feasible sets. In the throughput scheduling context, that also captures a setting where each player controls a subset of a set of identical machines. We define $x := \sum_{i=1}^n x_i$ to be the total number of feasible sets from \mathcal{S} that can be chosen by all players together, and note that $x \geq n$.

4 The Price of Anarchy for Nash Equilibria

We begin by giving the simple proof for the upper bound on the price of anarchy for arbitrary set packing games, assuming that the outcome is an α -approximate Nash equilibrium.

Theorem 1 *The α -approximate price of anarchy equals $\alpha + 1$ for set packing games.*

Proof We first give the proof for the upper bound. Take any instance with optimal solution OPT and Nash equilibrium S , and let S_i and OPT_i , $i = 1, \dots, n$, be the items selected by player i in S and OPT , respectively. For $W \subseteq J$, let $\bar{W} = J \setminus W$ be the complement of W in J .

Since all items in \bar{S} are available, and all items in OPT_i are feasible for player i , and all \mathcal{S}_i are downward closed, by the definition of α -approximate Nash equilibrium we have for all players i that $\alpha w(S_i) \geq w(OPT_i \cap \bar{S})$. Now we get, by the

fact that $S_i \cap S_k = \emptyset$ and $OPT_i \cap OPT_k = \emptyset$ for any $i \neq k$,

$$\begin{aligned} (\alpha + 1)w(S) &\geq \alpha w(S) + w(OPT \cap S) \\ &= \sum_{i=1}^n \alpha w(S_i) + w(OPT \cap S) \\ &\geq \sum_{i=1}^n w(OPT_i \cap \bar{S}) + w(OPT \cap S) \\ &= w(OPT). \end{aligned}$$

For a matching lower bound, see the proof of the next theorem. \square

We next give a matching lower bound example, which is an instance for throughput scheduling (and an asymmetric set packing game).

Example 2 Assume that $\alpha = p/q$, where $p \geq q$. When α is not rational, note that we can approximate α arbitrarily well by appropriate p and q . Consider a game with $q + 1$ players. For each player i , there is one machine, which we also denote by i . The set J of items are jobs that are partitioned into two sets P and Q , with $|P| = p, |Q| = q$. Each job $j \in J$ has deadline $d_j = 1$, unit weight $w_j = 1$, and its processing time on machine 1 is $p_{j1} = 1/p$. Moreover, jobs $j \in Q$ have processing time $p_{ji} = 1$ on any other machine $i \neq 1$, while jobs $j \in P$ have processing time $p_{ji} = 2$ on any other machine $i \neq 1$. Note that any subset of jobs of size p can be feasibly allocated to player 1. Players $2 \dots n$ can be allocated only one job each, and only jobs from Q . See Figure 3 for an illustration in the case where $\alpha = 3/2$. \triangleleft

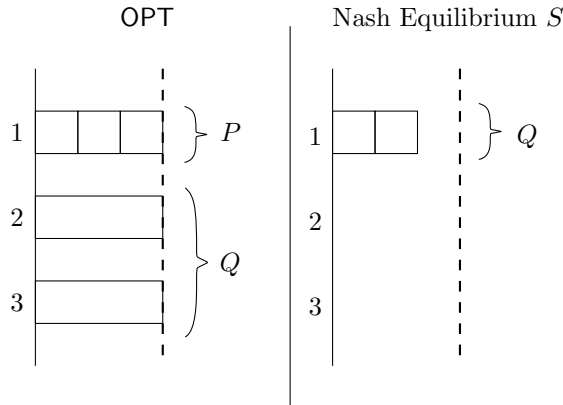


Fig. 3 Example 2 for $p = 3$ and $q = 2$. Numbers represent machines. Rectangles represent jobs. The left side of each job is its starting time, its width is its processing time on the machine on which it is allocated. The dashed line is the deadline, which is the same for all jobs in this example.

Theorem 2 *The α -approximate price of anarchy equals $\alpha + 1$ for throughput scheduling games.*

Proof $\text{PoA} \leq \alpha + 1$ follows from Theorem 1, since throughput scheduling games give rise to downward closed feasibility sets per player. To see that $\text{PoA} \geq \alpha + 1$, consider the instance of Example 2. In the optimum solution OPT , all $p + q$ jobs are feasibly allocated: All jobs in P are allocated to player 1, and each of the jobs in Q is allocated to one of the q other players $2, \dots, q + 1$. Now consider the α -approximate Nash equilibrium S where only q jobs are allocated: All jobs from Q are allocated to player 1, and no jobs are allocated to players $2, \dots, q + 1$. This is indeed an α -approximate Nash equilibrium, as player 1 achieves a total value of q , while maximally that player can be allocated $p = \alpha q$ jobs. In other words, the α -approximate Nash condition (1) holds for player 1. Moreover, given that all jobs from Q are allocated to player 1, players $2, \dots, q + 1$ cannot do better than a value 0, as none of the jobs from P are feasible for these players. We conclude that $\text{PoA} \geq w(OPT)/w(S) = (p + q)/q = \alpha + 1$. \square

Note that the upper bound is universal in the sense that it is independent of how the (α -approximate) Nash equilibrium is obtained. It is conceivable that specific algorithms can yield a better bound for the price of anarchy. However, the existence of more complicated counter-examples for specific algorithms is not unlikely either; see the next Section 5 for an example.

4.1 Symmetric Set Packing Games.

The price of anarchy of set packing games does not improve if the game is symmetric.

Theorem 3 *The α -approximate price of anarchy equals $\alpha + 1$ for symmetric set packing games.*

The upper bound $\alpha + 1$ is a consequence of Theorem 1. The lower bound follows from the following theorem for throughput scheduling games, based on Example 3 below, which is an instance of a symmetric set packing game.

Example 3 Let $\alpha = p/q$. There are n players i , each corresponding to one machine. The set J of $p + (q + 1)(n - 1)$ jobs is again partitioned into two sets P, Q , $|Q| = q(n - 1) + p$, $|P| = (n - 1)$. All jobs $j \in J$ have deadline $d_r = 1$. Jobs $j \in Q$ have processing times $p_j = 1/(q(n - 1) + p)$ and weight $w_j = 1$, while jobs $j \in P$ have processing times $p_j = 1$ and weight $w_j = p$. See Figure 4 for an illustration for the case where $p = 3, q = 2$ and $n = 3$. \triangleleft

Theorem 4 *The α -approximate price of anarchy equals $\alpha + 1$ for symmetric throughput scheduling games.*

Proof We are only left to show $\text{PoA} \geq \alpha + 1$. Consider Example 3. In the optimum solution OPT , player 1 is allocated all jobs $j \in Q$, and each remaining player is allocated exactly one job in P . Consider Nash equilibrium S where each player is allocated q jobs in Q . Note that S is indeed an α -approximate Nash equilibrium: Any player i could choose at most one job from P or at most p jobs from Q , since other players are allocated $q(n - 1)$ jobs from Q in total. Neither of the feasible deviations increases player i 's utility by more than a factor α . For this example, $w(OPT)/w(S) = \frac{pn + q(n-1)}{qn} = \frac{p+q}{q} - \frac{1}{n} \rightarrow 1 + \alpha$ for $n \rightarrow \infty$. \square

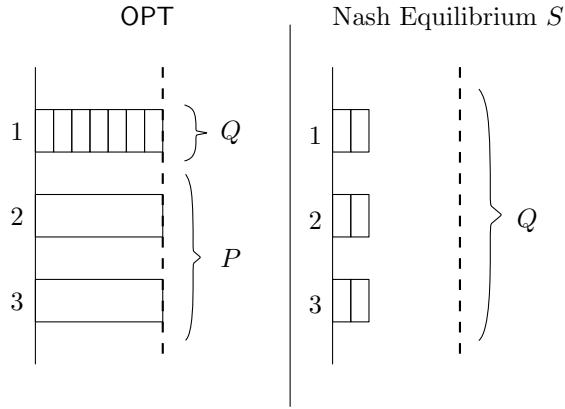


Fig. 4 Example 3 for $p = 3, q = 2, n = 3$. Numbers represent machines. Rectangles represent jobs. The left side of each job is its starting time, its width is its processing time. The dashed line is the deadline, which is the same for all jobs in this example.

However this Nash equilibrium is generally not subgame perfect when considering the corresponding sequential game. This observation is exploited in the following.

5 The Sequential Price of Anarchy

In this section we aim to show that substantial improvements for the price of anarchy are possible when considering sequential games and subgame perfect equilibria. This improvement is interesting, we believe, because in almost all applications that we can think of, it is the sequential version of the game which is practically relevant. To start with, however, we observe that in general, Example 2 also provides a lower bound for the quality of subgame perfect equilibria, and we get the following.

Theorem 5 *The α -approximate sequential price of anarchy equals $\alpha + 1$ for set packing games and throughput scheduling games.*

Proof Let us first argue that the sequential PoA $\geq \alpha + 1$. Recall Example 2, and assume that player 1 is the first to make a selection. Then if player 1 makes the same selection of job set Q as in the proof of Theorem 2, the obtained solution can indeed be obtained as an α -approximate subgame perfect equilibrium, as player 1 cannot improve by more than a factor α by selecting other jobs, and given that, all other players have nothing to choose. (We can specify any reasonable actions for those parts of the game tree that are not played in this equilibrium.) By the same argument as before, the lower bound on the price of anarchy follows. To see that the upper bound on the sequential PoA also holds, the next theorem shows that any subgame perfect equilibrium of the sequential game is a Nash equilibrium of the corresponding non-sequential game, hence Theorem 1 carries over. \square

Theorem 6 *For set packing games and any $\alpha \geq 1$, the actions played in an α -approximate subgame perfect equilibrium of any sequential version of the set packing game, are an α -approximate Nash equilibrium in the corresponding non-sequential strategic form game.*

Proof Consider the actions $S = (S_1, \dots, S_n)$ played in any subgame perfect equilibrium *SPE* of any sequential version of the set packing game. Assume w.l.o.g. the order was $1, \dots, n$. Consider any player i choosing S_i . We need to show that the Nash condition holds, which is $\alpha w(S_i) \geq w(T_i)$ for all $T_i \in \mathcal{S}_i$ with $T_i \subseteq J \setminus \cup_{k \neq i} S_k$. But as the choice S_i is part of a subgame perfect strategy, we even know that $\alpha w(S_i) \geq w(T_i)$ for all $T_i \in \mathcal{S}_i$ with $T_i \subseteq J \setminus \cup_{k=1}^{i-1} S_k$. This inequality must be true because i 's choice is α -approximately optimal, and since in a subgame perfect equilibrium, i 's payoff is not affected by rational subsequent players $k > i$, for *any* T_i that player i might choose. Hence S a Nash equilibrium in the non-sequential game. \square

It is not generally true that subgame perfect equilibria of a sequential game are Nash equilibria of the corresponding non-sequential game. See, e.g., [16] for an example.

5.1 Symmetric Set Packing Games

When considering symmetric set packing games, sequential play and subgame perfection rule out worst-case Nash equilibria. Assuming the outcome of such a game is an α -approximate subgame perfect equilibrium, the main result of this section is:

Theorem 7 *The α -approximate sequential price of anarchy equals $\sqrt[\alpha]{e}/(\sqrt[\alpha]{e} - 1)$ for symmetric set packing games.*

We prove the theorem in several steps. First we derive the lower bound, which is again obtained by considering a throughput scheduling instance.

Example 4 There are n players. Each player i corresponds to one machine. The set J of n^2 jobs is partitioned into n sets J_1, \dots, J_n , $|J_k| = n$ for all $k \in [n]$. We refer to a job from J_k as a k -job. All k -jobs have deadline k . All jobs $j \in J$ have processing time $p_j = 1$ and weight $w_j = 1$. See Figure 5 for an illustration for the case where $n = 5$ and $\alpha = 1$. \triangleleft

Lemma 1 *The α -approximate sequential PoA $\geq \sqrt[\alpha]{e}/(\sqrt[\alpha]{e} - 1)$ for symmetric set packing games.*

Proof In the optimum solution *OPT*, every player is allocated exactly one k -job for all $k = 1, \dots, n$. Therefore $w(\text{OPT}) = n^2$.

We construct an α -approximate subgame perfect outcome S , as follows: For every player $i = 1, \dots, n$ in this order, we find the maximum number of jobs that can be feasibly allocated to this player, given jobs already assigned to players $1, \dots, i - 1$, and when considering jobs with the largest deadlines first (which are the most flexible jobs). Denote this number of jobs m_i . We allocate to player i exactly $\lceil m_i/\alpha \rceil$ of these jobs, so that the allocation is still an α -approximation. Let S_i be the jobs allocated to player i in this way.

We bound $w(S)$ in the following way: Let $r_k(i) = \frac{|S_i \cap J_k|}{|S_i|}$, i.e. $r_k(i)$ is the fraction of k -jobs allocated to player i , relative to the total number of jobs allocated to player i . Let $r_k = \sum_{i=1}^n r_k(i)$. Now,

$$\sum_{k=1}^n r_k = \sum_{k=1}^n \sum_{i=1}^n r_k(i) = \sum_{i=1}^n \sum_{k=1}^n \frac{|S_i \cap J_k|}{|S_i|} = \sum_{i=1}^n 1 = n. \quad (6)$$

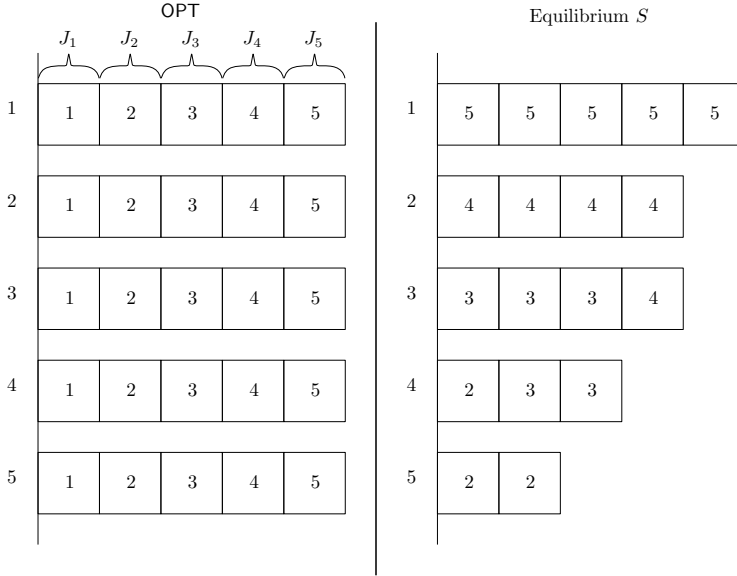


Fig. 5 Example 4 in case of 5 players and $\alpha = 1$. Numbers represent machines. Rectangles represent jobs. The left side of each job is its starting time, its width is its processing time. The number in each job is its deadline.

In S , any player i who gets allocated a k -job, is not allocated any job from J_j , $j \geq k + 2$, hence she is allocated at most $\lceil (k + 1)/\alpha \rceil \leq (k + 1 + \alpha)/\alpha$ jobs. Therefore, each k -job contributes at least $\alpha/(k + 1 + \alpha)$ to r_k . For any k for which all of the n k -jobs are allocated in S , we obtain

$$r_k \geq n\alpha/(k + 1 + \alpha). \quad (7)$$

Now, for some $k' \geq 0$, by construction of the allocation we have that for all $k \geq n - k'$, all n k -jobs are allocated, as well as a subset of the $(n - (k' + 1))$ -jobs. We obtain

$$n \geq \sum_{k=n-k'}^n r_k \geq \sum_{k=n-k'}^n \frac{n\alpha}{k + 1 + \alpha} \geq \int_{k=n-k'}^n \frac{n\alpha}{k + 1 + \alpha} dk, \quad (8)$$

where the first inequality follows from (6), the second inequality follows from (7), and the last inequality follows from basic calculus.

Because the last term is upper bounded by n , we can derive an upper bound on k' . In fact, basic calculus shows that

$$k' > \frac{(n + 1 + \alpha)(\sqrt[n]{e} - 1)}{\sqrt[n]{e}} \Rightarrow \int_{k=n-k'}^n \frac{n\alpha}{k + 1 + \alpha} dk > n,$$

which together with (8) yields that $k' \leq \frac{(n+1+\alpha)(\sqrt[n]{e}-1)}{\sqrt[n]{e}}$. Because only k -jobs with $k \geq n - (k' + 1)$ are allocated, we conclude that

$$w(S) \leq (k' + 1)n \leq \frac{(n + 1 + \alpha + \frac{\sqrt[n]{e}}{\sqrt[n]{e}-1})(\sqrt[n]{e} - 1)}{\sqrt[n]{e}} \cdot n.$$

We see that

$$\frac{w(OPT)}{w(S)} \geq \frac{n \sqrt[n]{e}}{(n+1 + \alpha + \frac{\sqrt[n]{e}}{\sqrt[n]{e}-1})(\sqrt[n]{e}-1)} \rightarrow \frac{\sqrt[n]{e}}{\sqrt[n]{e}-1} \quad \text{for } n \rightarrow \infty,$$

and the claim follows. \square

Note that the lower bound construction assumes that players choose the most flexible jobs first, which is reasonable from a practical point of view. Also note that in the lower bound example, $x_i = 1$ for all players. Therefore, the lower bound holds even in the special case when the strategic form game is a symmetric game.

To derive a matching upper bound on the sequential price of anarchy for symmetric set packing games, we generalize some of the proof ideas from [7] in their analysis of the k -GREEDY algorithm for throughput scheduling. We here generalize the proof to the case where players may control more than one machine, or in our case, more than one feasible set. Algorithmically, that means that in one iteration of the greedy algorithm not only one but several sets from the feasibility system get chosen. Stripping off the technical details, one of the core ingredients of the proof below is the fact that the next player may greedily choose her feasible subset(s), and this choice approximates the optimal solution for the set of currently available items. That optimal value is in turn at least as large as the optimal solution restricted to the set of currently available items. This is expressed in inequality (9) below. Indeed, this is a generic approach when analyzing greedy algorithms for subset selection problems, and the same idea also appears in the analysis of the greedy algorithm for maximum coverage problems; see [27, Lemma 3.13]. We want to prove:

Lemma 2 *The α -approximate sequential PoA $\leq \sqrt[n]{e}/(\sqrt[n]{e}-1)$ for symmetric set packing games.*

Denote by S_i the items selected by player i in an α -approximate subgame perfect equilibrium, and recall that S denotes both the strategy vector and $S = \cup_{i=1}^n S_i$, the total set of selected items. The following lemma lower bounds the total weight collected by player i .

Lemma 3 *We have for all players i*

$$w(S_i) \geq \frac{x_i}{x\alpha} w(OPT(J \setminus \cup_{j<i} S_j)).$$

where $OPT(W)$ denotes an optimal solution for a fixed subset of items $W \subseteq J$.

Proof Let $W := J \setminus \cup_{j<i} S_j$. Let OPT^i denote the maximum weight set of items that player i can achieve from W . Observe that $w(OPT^i) \geq (x_i/x)w(OPT(W))$. This follows because player i could potentially select the x_i most valuable feasible sets from $OPT(W)$. Now, because we assume an α -approximate subgame perfect equilibrium, $w(S_i) \geq w(OPT^i)/\alpha \geq x_i w(OPT(W))/(x\alpha)$. \square

Proof (of Lemma 2.) Let $\gamma := x\alpha$, and recall that $w(OPT) = w(OPT(J))$ denotes the value of an optimal solution. We use Lemma 3, to get

$$w(S_i) \geq \frac{x_i}{\gamma} w(OPT(J \setminus \cup_{j<i} S_j)) \geq \frac{x_i}{\gamma} \left(w(OPT) - \sum_{j<i} w(S_j) \right). \quad (9)$$

Here the last inequality is crucial and it holds because $w(OPT) - \sum_{j < i} w(S_j)$ represents the value of some feasible solution for the items $J \setminus \cup_{j < i} S_j$. Now add $\sum_{j=1}^{i-1} w(S_j)$ to both sides to get

$$\sum_{j=1}^i w(S_j) \geq \frac{x_i w(OPT)}{\gamma} + \frac{\gamma - x_i}{\gamma} \sum_{j=1}^{i-1} w(S_j). \quad (10)$$

We now follow the proof of Theorem 3.3 in [7], and aim to prove that

$$\sum_{j=1}^i w(S_j) \geq \frac{\gamma^{x'_i} - (\gamma - 1)^{x'_i}}{\gamma^{x'_i}} w(OPT), \quad (11)$$

where $x'_i = \sum_{j=1}^i x_j$. This part of the proof is done by mathematical induction on the player index i . In contrast to [7], this is technically more involved here. The proof of inequality (11) is by double induction and given in Appendix A. Now we get for $i = n$ (cp. to the proof of Thm. 3.3 in [7])

$$w(S) = \sum_{j=1}^n w(S_j) \geq \frac{\gamma^x - (\gamma - 1)^x}{\gamma^x} w(OPT).$$

We conclude that the

$$\text{sequential PoA} \leq \frac{\gamma^x}{\gamma^x - (\gamma - 1)^x} = \frac{(x\alpha)^x}{(x\alpha)^x - (x\alpha - 1)^x} \leq \frac{\sqrt[x]{e}}{\sqrt[x]{e} - 1},$$

where the last inequality follows because the right hand side is exactly the limit for $x \rightarrow \infty$, and the series $b_x = (x\alpha)^x / ((x\alpha)^x - (x\alpha - 1)^x)$ is monotone in x , with $b_1 = \alpha \leq \sqrt[x]{e} / (\sqrt[x]{e} - 1)$. This ends the proof of Lemma 2. \square

Basic calculus shows that

$$\alpha + \frac{1}{2} \leq \frac{\sqrt[x]{e}}{\sqrt[x]{e} - 1} \leq \alpha + \frac{1}{e - 1}$$

for $\alpha \geq 1$. Hence the improvement over the (pure Nash equilibrium) price of anarchy of $\alpha + 1$ is substantial, but only for small values of α . We explicitly mention the special case where $\alpha = 1$.

Corollary 1 *The sequential price of anarchy for symmetric set packing games equals $1 + 1/(e - 1) \approx 1.58$.*

6 The k -Collusion Price of Anarchy

While sequential play is a way to reduce the price of anarchy for symmetric set packing games, we now show that collusion of players helps to reduce the price of anarchy, too. This is true also for general, asymmetric set packing games. Recall that an α -approximate k -collusion Nash equilibrium means that no coalition K of up to k players can improve their total value $w(S_K)$ by more than a factor α . The following theorem generalizes Theorem 1 to the case where collusion is possible.

Theorem 8 *The α -approximate k -collusion price of anarchy equals $\alpha + \frac{n-k}{n-1}$ for set packing games.*

Note that for $k = n$, we consider an α -approximate (centralized) solution, and for $\alpha = 1$, this is an optimal solution. First we give a proof for the upper bound.

Lemma 4 *The α -approximate k -collusion PoA $\leq \alpha + \frac{n-k}{n-1}$ for set packing games.*

Proof The proof mimics our earlier proof of Theorem 1, only here we have to keep track of the values of more subsets of J . We fix an optimal solution OPT and a k -collusion Nash equilibrium S , write $N = \{1, \dots, n\}$, and use the following shorthand notation:

$$x_{ij} = \begin{cases} \text{the total weight of items in } OPT_i \cap S_j & \text{for } i, j \in N, \\ \text{the total weight of items in } S_j \setminus OPT & \text{for } i = 0, j \in N, \\ \text{the total weight of items in } OPT_i \setminus S & \text{for } i \in N, j = 0. \end{cases}$$

Our proof is based on the following observation: Players from any coalition K collude and collectively deviate if and only if the total weight of items allocated to them increases by more than a factor $\alpha \geq 1$, by choosing any set of items in $(\cup_{i \in K} S_i) \cup (J \setminus \cup_{i \notin K} S_i)$. Therefore, in particular for all coalitions K of size k in any α -approximate k -collusion Nash equilibrium, we have by (4) that

$$\alpha \left(\sum_{j \in K} \left(\sum_{i \in N} x_{ij} + x_{0j} \right) \right) \geq \sum_{i \in K} \left(\sum_{j \in K} x_{ij} + x_{i0} \right).$$

Note that all items that contribute to the left-hand side are allocated to players in K in the equilibrium S . Also note that all items that contribute to the right-hand side can be feasibly allocated to players in K , since these items are allocated to players from K in OPT . Also, these items are available for coalition K , since they are either allocated to players in K in S , or not allocated. We rewrite this as

$$\alpha \left(\sum_{j \in K} \left(\sum_{i \in N} x_{ij} + x_{0j} \right) \right) \geq \sum_{\substack{i \in K \\ i \neq j}} \sum_{j \in K} x_{ij} + \sum_{i \in K} (x_{ii} + x_{i0}). \quad (12)$$

Now, any player i is in $\binom{n-1}{k-1}$ coalitions of size k , and any combination of two players i, j is in $\binom{n-2}{k-2}$ coalitions of size k . Therefore, summing (12) over all coalitions K of size k yields

$$\begin{aligned} & \alpha \binom{n-1}{k-1} \left(\sum_{j \in N} \left(\sum_{i \in N} x_{ij} + x_{0j} \right) \right) \\ & \geq \binom{n-2}{k-2} \sum_{\substack{i \in N \\ i \neq j}} \sum_{j \in N} x_{ij} + \binom{n-1}{k-1} \sum_{i \in N} (x_{ii} + x_{i0}). \end{aligned}$$

Adding

$$\left(\binom{n-1}{k-1} - \binom{n-2}{k-2} \right) \sum_{\substack{i \in N \\ i \neq j}} \sum_{j \in N} x_{ij} = \binom{n-2}{k-1} \sum_{i \in N} \sum_{\substack{j \in N \\ i \neq j}} x_{ij}$$

to both sides yields

$$\begin{aligned}
& \alpha \left(\binom{n-1}{k-1} + \binom{n-2}{k-1} \right) \sum_{i \in N} \sum_{\substack{j \in N \\ i \neq j}} x_{ij} + \binom{n-1}{k-1} \left(\sum_{j \in N} (x_{jj} + x_{0j}) \right) \\
& \geq \binom{n-1}{k-1} \sum_{i \in N} \sum_{\substack{j \in N \\ i \neq j}} x_{ij} + \binom{n-1}{k-1} \sum_{i \in N} (x_{ii} + x_{i0}). \tag{13}
\end{aligned}$$

Therefore,

$$\begin{aligned}
& \alpha \left(\binom{n-1}{k-1} + \binom{n-2}{k-1} \right) w(S) \\
& = \alpha \left(\left(\binom{n-1}{k-1} + \binom{n-2}{k-1} \right) \sum_{i \in N} \sum_{j \in N} x_{ij} + \left(\binom{n-1}{k-1} + \binom{n-2}{k-1} \right) \sum_{j \in N} x_{0j} \right) \\
& \geq \alpha \left(\left(\binom{n-1}{k-1} + \binom{n-2}{k-1} \right) \sum_{i \in N} \sum_{\substack{j \in N \\ i \neq j}} x_{ij} + \binom{n-1}{k-1} \left(\sum_{j \in N} (x_{jj} + x_{0j}) \right) \right) \\
& \geq \binom{n-1}{k-1} \sum_{i \in N} \sum_{\substack{j \in N \\ i \neq j}} x_{ij} + \binom{n-1}{k-1} \sum_{i \in N} (x_{ii} + x_{i0}) \\
& = \binom{n-1}{k-1} w(OPT),
\end{aligned}$$

where the last inequality follows from (13). This yields

$$k\text{-collusion PoA} \leq \alpha \frac{\binom{n-1}{k-1} + \binom{n-2}{k-1}}{\binom{n-1}{k-1}} = \alpha + \frac{n-k}{n-1}.$$

□

In fact, this proof of the upper bound provides us with an easy way to create a tight lower bound example for any n .

Example 5 We make the upper bound analysis tight by setting $x_{ii} = 0$ and $x_{0i} = 0$ for all players $i \in N$. We normalize $x_{ij} = 1$ for all players $i, j \in N$ for which $i \neq j$, and finally we set $x_{i0} = n - k + (n - 1)(\alpha - 1)$ for all players $i \in N$. We construct the strategy spaces such that any player i can only choose subsets of either OPT_i or S_i , where S_i is the set chosen in the k -collusion Nash equilibrium. The resulting game for $n = 3, k = 2$ is shown in Figure 6. ◁

Proof (of the matching lower bound for α -approximate k -collusion PoA.) To see that the above construction actually yields an α -approximate k -collusion Nash equilibrium, consider any coalition K of k players. If players play strategy profile S , any player in K has utility $n - 1$. By switching to the strategy chosen in OPT ,

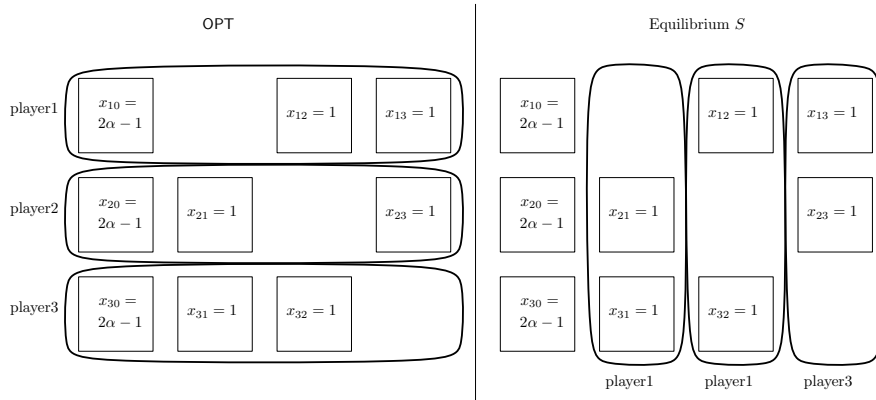


Fig. 6 The k -collision Nash equilibrium from Example 5 for $k = 2$ and $n = 3$. Circled items are allocated to the same player. Each item is named after the value used in the upper bound proof.

each player in K obtains utility $(k - 1)1 + 1((n - 1)(\alpha - 1) + n - k) = \alpha(n - 1)$, which is fine. If some players in K choose a subset of the items chosen in OPT , and other players in K choose a subset of the items chosen in S , then this yields a total value at most $\alpha(n - 1)$ for each player. We see that no coalition of k players can improve by deviating, from which the result follows. \square

For $\alpha = 1$, we obtain the following as a special case.

Corollary 2 *The k -collision price of anarchy equals $1 + \frac{n-k}{n-1}$ for set packing games.*

Although the k -collision price of anarchy is strictly lower than the price of anarchy for all $k \geq 2$, note that this improvement becomes negligible for large n . Interestingly, as opposed to all other lower bound examples in this paper, we did not find a matching lower bound example for throughput scheduling games.

7 Concluding Remarks

As discussed earlier, when more than one player may select one and the same item, one needs a utility sharing protocol, and the game becomes a covering game for which several results have been obtained, e.g. by Gairing [18], and for more general settings also by Paccagnan et al. [38,39]. The price of anarchy for Nash equilibria is then $2 - 1/n$, assuming that items are shared uniformly. We have also studied the sequential version of that class of games [11], and observed that sequential play does *not* allow to improve this price of anarchy bound. To pin down the exact value of the sequential price of anarchy for that class of games is open.

Acknowledgements The first author was supported by funding from the Dutch applied mathematics institute 3TU.AMI, and the Center for Telematics and Information Technology (CTIT) of the University of Twente. An extended abstract with parts of the results appeared in the conference proceedings [28]. Thanks to Rudolf Müller, Frits Spijksma, and Johann Hurink for some helpful discussions. Also thanks to the anonymous reviewers of an earlier draft of this paper, and to the reviewers of this paper for constructive comments.

References

1. N. Andelman, M. Feldman, and Y. Mansour (2009). Strong price of anarchy. *Games and Economic Behavior*, vol 65, 289-317.
2. A. Angelucci, V. Bilò, M. Flammini, and L. Moscardelli (2013). On the sequential price of anarchy of isolation games. In: Z. Cai, A. Zelikovsky, A.G. Bourgeois (eds.), *Computing and Combinatorics (COCOON 2013)*, Lecture Notes in Computer Science 8591, 17-28.
3. E. Anshelevich, A. DasGupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden (2004). The price of stability for network design with fair cost allocation. In *Proc. 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*, 295-304.
4. R.J. Aumann (1959). Acceptable points in general cooperative n -person games. In: *Contributions to the Theory of Games IV*, Annals of Mathematics Studies 40 (R.D. Luce and A. W. Tucker, eds.) 287-324, Princeton University Press.
5. B. Awerbuch, Y. Azar, and A. Epstein (2005). The price of routing unsplittable flow. In *Proceedings of the 37th STOC*, 57-66.
6. V. Bakker, M.G.C. Bosman, A. Molderink, J.L. Hurink and G.J.M. Smit (2010). Demand side load management using a three step optimization methodology. In *Proceedings of the 1st SmartGridComm*, 431-436.
7. A. Bar-Noy, S. Guha, J. Naor, and B. Schieber (2001). Approximating the throughput of multiple machines in real-time scheduling. *SIAM Journal on Computing*, vol 31, 331-52.
8. P. Berman and B. DasGupta (2000). Multi-phase Algorithms for Throughput Maximization for Real-Time Scheduling. *Journal of Combinatorial Optimization*, vol 4, 307-323.
9. V. Bilò, M. Flammini, G. Monaco, and L. Moscardelli (2013). Some anomalies of farsighted strategic behavior. In: T. Erlebach and G. Persiano (eds.), *Approximation and Online Algorithms (WAOA 2012)*, Lecture Notes in Computer Science 7846, 229-241.
10. V. Bilò, M. Flammini, G. Monaco, and L. Moscardelli (2011). On the performances of Nash equilibria in isolation games. *Journal of Combinatorial Optimization*, vol 22, 378-391.
11. J.-T. Brethouwer, The quality of equilibria in generalized market sharing games. MSc thesis, University of Twente, 2018. <https://essay.utwente.nl/76986/>
12. P. Brucker (2004). *Scheduling Algorithms* (4th ed.), Springer Verlag, Berlin.
13. R. Buyya, R. Ranjan, and R.N. Calheiros (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In: C.H. Hsu, L.T. Yang, J.H. Park and S.S. Yeo (eds.), *Algorithms and Architectures for Parallel Processing*, Lecture Notes in Computer Science 6081, 13-31.
14. B. Chandra, and M.M. Halldórsson (2001). Greedy local improvement and weighted set packing approximation. *Journal of Algorithms*, vol 39, 223-240.
15. G. Christodoulou and E. Koutsoupias (2005). The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on the Theory of Computing (STOC 2005)*, 67-73.
16. J.R. Correa, J. de Jong, B. de Keijzer, and M.J. Uetz (2015). The Curse of Sequentiality in Routing Games. In: E. Markakis and G. Schäfer (eds.), *Web and Internet Economics (WINE 2015)*, Lecture Notes in Computer Science 9470, 258-271.
17. B. Farzad, N. Olver, and A. Vetta (2008). A priority-based model of routing Chicago J. Theoret. Comput. Sci., 2008(1).
18. M. Garing (2009). Covering Games: Approximation through Non-cooperation. In: S. Leonardi (ed.), *Internet and Network Economics (WINE 2009)*, Lecture Notes in Computer Science 5929, 184-195.
19. M.R. Garey and D.S. Johnson (1979), *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, New York.
20. M.X. Goemans, L. Li, V.S. Mirrokni, and M. Thottan (2004). Market Sharing Games Applied to Content Distribution in Ad-Hoc Networks. In: *Proc. 5th ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 2004)*, 55-66.
21. R. Graham, E. Lawler, J. Lenstra, and A. Rinnooy Kan (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5(2):287-326.
22. T. Harks, S. Heinz, and M. Pfetsch (2009). Competitive online multicommodity routing *Theory Comput. Syst.*, 45: 533-545.
23. T. Harks and L. Vegh (2007). Selfish routing with online demands In: Jansen J., Pralat P. (eds.), *Combinatorial and Algorithmic Aspects of Networking (CAAN 2007)*, Lecture Notes in Computer Science, Springer, pp. 27-45.

24. R. Hassin and U. Yovel (2014). Sequential scheduling on identical machines. *Operations Research Letters*, vol 43, 530–533.
25. A. Hayrapetyan, É. Tardos, and T. Wexler (2006). The effect of collusion in congestion games. In *Proceedings of the 38th Annual ACM Symposium on the Theory of Computing (STOC 2006)*, 89–98.
26. E. Hazan, S. Safra, and O. Schwartz (2006). On the complexity of approximating k -Set Packing. *Computational Complexity*, vol 15, 20–39.
27. D. Hochbaum (1997). Approximating Covering and Packing Problems. Chap. 3 in: *Approximation Algorithms for NP-hard Problems*. D. Hochbaum (ed.), PWS Publishing Company, Boston.
28. J. de Jong, M. Uetz, and A. Wombacher (2013). Decentralized throughput scheduling. In: P.G. Spirakis and M. Serna (eds.), *Algorithms and Complexity (CIAC 2013)*, Lecture Notes in Computer Science 7878, 134–145.
29. J. de Jong and M. Uetz (2014). The sequential price of anarchy for atomic congestion games. In: T.-Y. Liu, Q. Qi and Y. Ye (eds.), *Web and Internet Economics (WINE 2014)*, Lecture Notes in Computer Science 8877, 429–434.
30. R.M. Karp (1972). Reducibility among combinatorial problems. In: *Complexity of Computer Computations* (R.E. Miller, J.W. Thatcher, and J.D. Bohlinger, eds.), The IBM Research Symposia Series, 85–103, Springer.
31. E. Koutsoupias and C.H. Papadimitriou (2009). Worst-case equilibria. *Computer Science Review*, vol 3, 65–69. (Preliminary version in *Proc. 16th STACS*, 1999)
32. J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker (1977). Complexity of Machine Scheduling Problems. *Annals of Discrete Mathematics*, vol 1, 343–362.
33. I. Milchtaich (1998). Crowding Games are Sequentially Solvable. *International Journal of Game Theory*, vol 27, 501–509.
34. A. Molderink, V. Bakker, M.G.C. Bosman, J.L. Hurink, and G.J.M. Smit (2010). Management and control of domestic smart grid technology. *IEEE transactions on Smart Grid*, vol 1, 109–119.
35. J.M. Moore (1968). An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, 15:102–109.
36. N. Nisan, T. Roughgarden, É. Tardos, and V.V. Vazirani (2007). *Algorithmic Game Theory*. Cambridge University Press.
37. N. Olver (2006). The price of anarchy and a priority-based model of routing (MSc thesis), McGill University, Montreal, Canada.
38. D. Paccagnan and J.R. Marden (2018). Distributed resource allocation through utility design Part I: Optimizing the performance certificates via the price of anarchy. ArXiv paper 1807.01333v1.
39. D. Paccagnan, R. Chandan and J.R. Marden (2018). Distributed resource allocation through utility design Part II: Applications to submodular, supermodular and set covering problems. ArXiv paper 1807.01343v1.
40. R. Paes Leme, V. Syrgkanis, and É. Tardos (2012). The curse of simultaneity. In *Proc. 3rd Conference on Innovations in Theoretical Computer Science (ITCS 2012)*, ACM, 60–67.
41. C.H. Papadimitriou (2001). Algorithms, Games, and the Internet. In *Proc. 33rd Annual ACM Symposium on the Theory of Computing (STOC 2001)*, 749–753.
42. H. Peters (2015), *Game Theory: A Multi-Leveled Approach*. Springer, 2nd ed.
43. T. Roughgarden and É. Tardos (2002). How bad is selfish routing? *Journal of the ACM*, vol 49, 236–259.
44. R. Selten (1965). Spieltheoretische Behandlung eines Oligopolmodells mit Nachfragerträglichkeit: Teil 1: Bestimmung des dynamischen Preisgleichgewichts. *Zeitschrift für die gesamte Staatswissenschaft*, vol 121, 301–324.
45. A. Skopalik and B. Vöcking (2008). Inapproximability of pure Nash equilibria. In *Proc. 40th Annual ACM Symposium on the Theory of Computing (STOC 2008)*, 355–364.
46. A. Vetta (2002). Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002)*, 416–425.
47. <http://www.snappcar.de>, retrieved 2017.

A Appendix: Proof of inequality (11) by induction.

Recall that we want to prove the following by induction on player index i .

$$\sum_{j=1}^i w(S_j) \geq \frac{\gamma^{x'_i} - (\gamma - 1)^{x'_i}}{\gamma^{x'_i}} w(OPT),$$

The base case $i = 1$ is the following lemma, proved by yet another inductive argument on x_1 .

Lemma 5

$$w(S_1) \geq \frac{\gamma^{x_1} - (\gamma - 1)^{x_1}}{\gamma^{x_1}} w(OPT).$$

Proof We know by definition of γ , and by plugging $i = 1$ into Lemma 3 that

$$w(S_1) \geq \frac{x_1}{\gamma} w(OPT).$$

Hence we are done when we can prove by induction on x_1 that

$$\frac{x_1}{\gamma} \geq \frac{\gamma^{x_1} - (\gamma - 1)^{x_1}}{\gamma^{x_1}}.$$

When $x_1 = 1$, we get

$$\frac{1}{\gamma} \geq \frac{\gamma - (\gamma - 1)}{\gamma} = \frac{1}{\gamma},$$

which clearly holds. Assume the claim holds for $x_1 = k - 1$. We get

$$\begin{aligned} \frac{k}{\gamma} &= \frac{k-1}{\gamma} + \frac{1}{\gamma} \\ &\geq \frac{\gamma^{k-1} - (\gamma-1)^{k-1}}{\gamma^{k-1}} + \frac{1}{\gamma} \\ &= \frac{\gamma^k - (\gamma-1)^k - (\gamma-1)^{k-1} + \gamma^{k-1}}{\gamma^k} \\ &\geq \frac{\gamma^k - (\gamma-1)^k}{\gamma^k}, \end{aligned}$$

proving Lemma 5. □

Assume now that (11) holds for $i - 1$. Applying the induction hypothesis to (10) we get

$$\sum_{j=1}^i w(S_j) \geq \frac{x_i w(OPT)}{\gamma} + \frac{\gamma - x_i}{\gamma} \cdot \frac{\gamma^{x'_{i-1}} - (\gamma - 1)^{x'_{i-1}}}{\gamma^{x'_{i-1}}} w(OPT).$$

This can be used to prove the inductive claim, using the following.

Lemma 6

$$\frac{x_k}{\gamma} + \frac{\gamma - x_k}{\gamma} \cdot \frac{\gamma^{x'_{k-1}} - (\gamma - 1)^{x'_{k-1}}}{\gamma^{x'_{k-1}}} \geq \frac{\gamma^{x'_k} - (\gamma - 1)^{x'_k}}{\gamma^{x'_k}}.$$

Proof We have

$$\begin{aligned}
& \frac{x_k}{\gamma} + \frac{\gamma - x_k}{\gamma} \cdot \frac{\gamma^{x'_k-1} - (\gamma - 1)^{x'_k-1}}{\gamma^{x'_k-1}} \\
&= \frac{x_k}{\gamma} \cdot \frac{(\gamma - 1)^{x'_k-1}}{\gamma^{x'_k-1}} + \frac{\gamma^{x'_k-1} - (\gamma - 1)^{x'_k-1}}{\gamma^{x'_k-1}} \\
&\geq \frac{\gamma^{x_k} - (\gamma - 1)^{x_k}}{\gamma^{x_k}} \cdot \frac{(\gamma - 1)^{x'_k-1}}{\gamma^{x'_k-1}} + \frac{\gamma^{x'_k-1} - (\gamma - 1)^{x'_k-1}}{\gamma^{x'_k-1}} \\
&= \left(1 - \frac{(\gamma - 1)^{x_k}}{\gamma^{x_k}}\right) \cdot \frac{(\gamma - 1)^{x'_k-1}}{\gamma^{x'_k-1}} + 1 - \frac{(\gamma - 1)^{x'_k-1}}{\gamma^{x'_k-1}} \\
&= 1 - \frac{(\gamma - 1)^{x_k}}{\gamma^{x_k}} \cdot \frac{(\gamma - 1)^{x'_k-1}}{\gamma^{x'_k-1}} \\
&= \frac{\gamma^{x'_k} - (\gamma - 1)^{x'_k}}{\gamma^{x'_k}},
\end{aligned}$$

where the first inequality follows from $\frac{x_k}{\gamma} \geq \frac{\gamma^{x_k} - (\gamma - 1)^{x_k}}{\gamma^{x_k}}$, as shown in the proof of Lemma 5, and the last equality follows from $x'_k = x'_{k-1} + x_k$. \square