Dynamic Resource Allocation Games*

Guy Avni[1], Thomas A. Henzinger[1], and Orna Kupferman[2]

[1] IST Austria
[2] The Hebrew University

Abstract. In *resource allocation games*, selfish players share resources that are needed in order to fulfill their objectives. The cost of using a resource depends on the load on it. In the traditional setting, the players make their choices concurrently and in one-shot. That is, a strategy for a player is a subset of the resources. We introduce and study *dynamic* resource allocation games. In this setting, the game proceeds in phases. In each phase each player chooses one resource. A scheduler dictates the order in which the players proceed in a phase, possibly scheduling several players to proceed concurrently. The game ends when each player has collected a set of resources that fulfills his objective. The cost for each player then depends on this set as well as on the load on the resources in it – we consider both congestion and cost-sharing games. We argue that the dynamic setting is the suitable setting for many applications in practice. We study the stability of dynamic resource allocation games, where the appropriate notion of stability is that of subgame perfect equilibrium, study the inefficiency incurred due to selfish behavior, and also study problems that are particular to the dynamic setting, like constraints on the order in which resources can be chosen or the problem of finding a scheduler that achieves stability.

1 Introduction

Resource allocation games (RAGs, for short) [21] model settings in which selfish agents share resources that are needed in order to fulfill their objectives. The cost of using a resource depends on the load on it. Formally, a k-player RAG G is given by a set E of resources and a set of possible strategies for each player. Each strategy is a subset of resources, fulfilling some objective of the player. Each resource $e \in E$ is associated with a latency function $\ell_e : \mathbb{N} \rightarrow \mathbb{R}$, where $\ell_e(\gamma)$ is the cost of a single use of e when it has load γ. For example, in *network formation games* (NFGs, for short) [2], a network is modeled by a directed graph, and each player has a source and a target vertex. In the corresponding RAG, the resources are the edges of the graph and the objective of each player is to connect his source and target. Thus, a strategy for a player is a set of edges that form a simple path from the source to the target. When an edge e is used by m players, each of them pays $\ell_e(m)$ for his use.

A key feature of RAGs is that the players choose how to fulfill their objectives *in one shot* and *concurrently*. Indeed, a strategy for a player is a subset of the resources – chosen as a whole, and the players choose their strategies simultaneously. In many settings, however, resource sharing proceeds in a different way. First, in many settings, the

* This research was supported in part by the European Research Council (ERC) under grants 267989 (QUAREM) and 278410 (QUALITY), and by the Austrian Science Fund (FWF) under grants S11402-N23 (RiSE) and Z211-N23 (Wittgenstein Award).

choices of the players are made resource by resource as the game evolves. For example, when the network in an NFG models a map of roads and players are drivers choosing routes, it makes sense to allow each driver not to commit to a full route in the beginning of the game but rather to choose one road (edge) at each junction (vertex), gradually composing the full route according to the congestion observed. Second, players may not reach the junctions together. Rather, in each "turn" of the game, only a subset of the players (say, these that have a green light) proceed and chose their next road.

As another example to a rich composition and scheduling of strategies, consider the setting of *synthesis from component libraries* [16], where a designer synthesizes a system from existing components rather than from scratch as in the traditional problem [20]. It is shown in [4,6] that when multiple designers use the same library, a RAG arises. Here too, the choice of components may be made during the design process and may evolve according to choices of other designers.

In this work we introduce and study *dynamic resource allocation games*, which allow the players to choose resources in an iterative and non-concurrent manner. A dynamic RAG is given by a pair $\mathcal{G} = \langle G, \nu \rangle$, where G is a k-player RAG and $\nu : \{1, \ldots, k\} \to \{1, \ldots, k\}$ is a *scheduler*. A dynamic RAG proceeds in *phases*. In each phase, each player chooses one resource. A phase is partitioned into at most k *turns*, and the scheduler dictates which players proceed in each turn: Player i moves at turn $\nu(i)$. Note that the scheduler may assign the same turn to several players, in which case they choose a resource concurrently in a phase. Once all turns have been taken, a phase is concluded and a new phase begins. A *strategy* for a player in a dynamic RAG is a function that takes the history of choices made by the players so far (in the current phase as well as previous ones), and returns the next choice the player makes. A player finishes playing once the resources he has chosen forms a strategy in the underlying RAG. In an outcome of the game, each player selects a set of resources. His cost depends on their load and latency functions as in usual RAGs.

Example 1. Consider the 4-player network formation game that is depicted in Figure 1. The interesting edges have names, e.g., $a, b, c \ldots$, and their latency function is depicted below the edge. For example, we have $\ell_a(x) = x$ and $\ell_{c_1}(x) = 10x$. The other edges have latency function 0. The source and target of a node of Player i are depicted with a node called s and t, respectively, and with a subscript i. For example, Player 2's source is $s_{1,2}$ and he has two targets t_2^L and t_2^R. The players' strategies are paths from one of their sources to one of their targets.

Consider a dynamic version of the game in which Player i chooses an edge at turn i. At first look, it seems that edge g will never be chosen. However, we show that Player 1's optimal strategy uses it. Player 1 has three options in the first turn, either choose g, a, or b[3]. Assume he chooses a (and dually b). Then, we claim that Player 2 will choose b. Note that Players 3 and 4 move oposite of Player 2 no matter how Player 1 moves, as they prefer avoiding a load of 2 on c_1 and c_2, which costs 20 each, even at the cost of a load of 3 on f, which costs only 3. Knowing this, Player 2 prefers using b alone over sharing a with Player 1. Since the loads on a and e are 1 and 3, respectively, Player 1's cost is $1 + 3 = 4$.

[3] In this example we require the players to choose their paths incrementally, which is not the general definition we use in the paper.

On the other hand, if Player 1 chooses g in the first phase, he postpones revealing his choice between left and right. If Player 2 proceeds left, then Players 3 and 4 proceed right, and Player 1 proceeds left in the second phase. Now, the load on a and e is 2 and 1, respectively, thus Player 1's cost is $\frac{1}{2} + 2 + 1 = 3\frac{1}{2}$. □

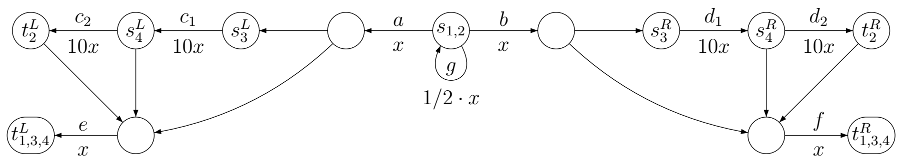

Fig. 1. A network formation game in which it is beneficial to select a path that is not simple.

The concept of what we refer to as a dynamic game is old and dates back to Von Neumann's work on *extensive form games* [18]. Most work on RAGs considers the simultaneous setting. However, there have been different takes on adding dynamicity to RAGs. In [17], the authors refine the notion of NE by considering *lookahead* equilibria; a player predicts the reactions of the other players to his deviations, and he deviates only if the outcome is beneficial. The depth of lookahead is bounded and is a parameter to the equilibria. A similar setting was applied to RAGs in [7], where the players are restricted to choose a *best-response* move rather than a deviation that might not be immediately beneficial. Concurrent ongoing games are commonly used in formal methods to model the interaction between different components of a system (c.f., [1]). In such a game, multiple players move a token on a graph. At each node, each player selects a move, and the transition function determines the next position of token, given the vector of moves the players selected. The objectives of the players refer to the generated path and no costs are involved. Closest to our model is the model of [15], and its subsequent works [8,10]. They study RAGs in which players arrive and select strategies one by one, yet in one shot.

Our dynamic games differ from all of these games in two aspects. We allow the players to reveal their choices of resources in parts, thus we allow "breaking" the strategies. Moreover, the choices the players make in all the games in earlier work are either concurrent or sequential, and we allow a mix between the two. These new aspects we introduce are natural and general, and can be applied to other games and settings.

The first question that arises in the context of games, and on which we focus in this work, is the existence of a *stable outcome* of the game. In the context of RAGs, the most prominent stability concept is that of a *Nash equilibrium* (NE, for short) – a profile such that no player can decrease his cost by unilaterally deviating from his current strategy. It is well known that every RAG has an NE [21]. The definition of an NE applies to all games, and can also be applied to our dynamic RAGs. As we demonstrate in Example 2, the dynamic setting calls for a different stability concept, and the prominent one is *subgame perfect equilibrium* (SPE, for short) [24], which we define formally in Section 2.

Classifying RAGs, we refer to the type of their latency functions as well as the type of the objectives of the players. *Congestion games* [22] are RAGs in which the latency functions are increasing, whereas in *cost-sharing games* [2], each resource has a cost that is split between the players that use it (in particular, the latency functions are decreasing). In terms of objectives, we consider *singleton* RAGs, in which the objectives of the players are singletons of resources, and *symmetric* RAGs, in which all players have the same objective.

Our most interesting results are in terms of equilibrium existence. It is easy to show, and similar results are well known, that every dynamic RAG with a *sequential scheduler* has an SPE. The proof uses backwards induction on the tree of all possible outcomes of the game. One could hope to achieve a similar proof also for schedulers that are not sequential, especially given the fact that every RAG has an NE. Quite surprisingly, however, we show that this is not the case. For congestion games, we show examples of a singleton congestion game and a symmetric congestion game with no SPE. Moreover, the latency function in both cases is linear. On the positive side, we show that singleton and symmetric congestion games are guaranteed to have an SPE for every scheduler. For cost-sharing games, we also show an example with no SPE. In the cost-sharing setting, however, we show that singleton objectives are sufficient to guarantee the existence of an SPE in all schedules. It follows that singleton dynamic congestion games are less stable than singleton dynamic cost-sharing games. This is interesting, as in the one-shot concurrent setting, congestion games are known to be more stable than cost-sharing games in various parameters. One would expect that this "order of stability" would carry over to the dynamic setting, as is the case in other extensions of the traditional setting. For example, an NE is not guaranteed for *weighted* cost-sharing games [9] as well as very restrictive classes of *multiset* cost-sharing games [5], whereas every linear weighted congestion game [12] and even linear multiset congestion game is guaranteed to have an NE [6].

It is well known that decentralized decision-making may lead to solutions that are sub-optimal from the point of view of society as a whole. In simultaneous games, the standard measures to quantify the inefficiency incurred due to selfish behavior is the *price of anarchy* (PoA) [14] and *price of stability* (PoS) [2]. In both measures we compare against the *social optimum* (SO, for short), namely the cheapest profile. The PoA is the worst-case inefficiency of an NE (that is, the ratio between the cost of a worst NE and the SO). The PoS is the best-case inefficiency of an Nash equilibrium (that is, the ratio between the cost of a best NE and the social optimum). For the dynamic setting, we adjust these two measures to consider SPEs rather than NEs, and we refer to them as DPoA and DPoS. We study the equilibrium inefficiency in the classes of games that have SPEs. We show that the DPoA and DPoS in dynamic singleton cost-sharing games as well as dynamic singleton congestion games coincide with the PoA and PoS in the corresponding simultaneous class. As mentioned above, [15,8,10] study games in which players arrive one after the other. Since their games are sequential, they always have an SPE. They study the *sequential PoA*, and show that it can either be equal, below, or above the PoA of the corresponding class of RAGs.

We then turn to study the computational complexity of deciding whether a given dynamic RAG has an SPE. We show that the problem is PSPACE-complete for both congestion and cost-sharing games. Our lower bound for cost-sharing games implies that finding an SPE in sequential games is PSPACE-hard. To the best of our knowledge, while this problem was solved in [15] for congestion games, we are the first to solve it for cost-sharing games.

Due to lack of space, some proofs and examples are given in the full version, which can be found in the authors' homepages.

2 Preliminaries

Resource allocation games For $k \geqslant 1$, let $[k] = \{1, \ldots, k\}$. A *resource-allocation game* (RAG, for short) is a tuple $G = \{[k], E, \{\Sigma_i\}_{i \in [k]}, \{\ell_e\}_{e \in E}\}$, where $[k]$ is a set of k players; E is a set of resources; for $i \in [k]$, the set $\Sigma_i \subseteq 2^E$ is a set of objectives[4] for Player i; and, for $e \in E$, we have that $\ell_e : \mathbb{N} \rightarrow \mathbb{R}$ is a latency function. The game proceeds in one-round in which the players select simultaneously one of their objectives. A *profile* $P = \langle \sigma_1, \ldots, \sigma_k \rangle \in \Sigma_1 \times \ldots \times \Sigma_k$ is a choice of an objective for each player. For $e \in E$, we denote by $nused(P, e)$ the number of times e is used in P, thus $nused(P, e) = |\{i \in [k] : e \in \sigma_i\}|$. For $i \in [k]$, the *cost* of Player i in P, denoted $cost_i(P)$, is $\sum_{e \in \sigma_i} \ell_e(nused(P, e))$.

Classes of RAGs are characterized by the type of latency functions and objectives. In *congestion* games (CGs, for short), the latency functions are increasing. An exceptionally stable class of CGs are ones in which the latency functions are affine (c.f., [12,6]); every resource $e \in E$ has two constants a_e and b_e, and the latency function is $\ell_e(x) = a_e \cdot x + b_e$. In *cost-sharing* games (SG, for short), each resource $e \in E$ has a *cost* c_e and the players that use the resource share its cost, thus the latency function for e is $\ell_e(x) = \frac{c_e}{x}$, and in particular is decreasing. We use DCGs and DSGs to refer to dynamic CGs and dynamic SGs, respectively. In terms of objectives, we study *symmetric* games, where the players' sets of objectives are equal, thus $\Sigma_i = \Sigma_j$ for all $i, j \in [k]$, and *singleton* games, where each $\sigma \in \Sigma_i$ is a singleton, for every $i \in [k]$.

Dynamic resource allocation games A *dynamic* RAG is pair $\mathcal{G} = \langle G, \nu \rangle$, where G is a RAG and $\nu : [k] \rightarrow [k]$ is a *scheduler*. Intuitively, in a dynamic game, rather than revealing their objectives at once, the game proceeds in *phases*: in each phase, each player reveals one resource in his objective. Each phase is partitioned into at most k *turns*. The scheduler dictates the order in which the players proceed in a phase by assigning to each player his turn in the phases. If the scheduler assigns the same turn to several players, they select a resource concurrently. Once all players take their turn, a phase is concluded and a new phase begins. There are two "extreme" schedulers: (1) players get different turns, i.e., ν is a permutation, (2) all players move in one turn, i.e., $\nu \equiv 1$. We refer to games with these schedulers as *sequential* and *concurrent*, respectively. Note that ν might not be an onto function. For simplicity, we assume that, for $j > 1$, if turn j is assigned a player, then so is turn $j - 1$. We use t_ν to denote the last turn according to ν, thus $t_\nu = \max_i \nu(i)$.

Let $E_\perp = E \cup \{\perp\}$, where $\perp$ is a special symbol that represents the fact that a player finished playing. Consider a turn $j \in [k]$. We denote by $\text{before}(j)$ the set of players that play before turn j; thus $\text{before}(j) = \{i \in [k] : \nu(i) < j\}$. A player has full knowledge of the resources that have been chosen in previous phases and the

[4] We use "objectives" rather than "strategies" as the second will later be used for dynamic games.

resources chosen in previous turns in the current phase. A strategy for Player i in $\mathcal{G}$ is a function $f_i : (E_\perp^{[k]})^* \cdot (E_\perp^{\text{before}(\nu(i))}) \to E_\perp$. A *profile* $P = \langle f_1, \ldots, f_k \rangle$ is a choice of a strategy for each player. The *outcome* of the game given a profile P, denoted $out(P)$, is an infinite sequence of functions $\pi^1, \pi^2, \ldots$, where for $i \geqslant 1$, we have $\pi^i : [k] \to E_\perp$. We define the sequence inductively as follows. Let $m \geqslant 1$ and $j \in [k]$. Assume $m - 1$ phases have been played as well as $j - 1$ turns in the m-th phase, thus $\pi^1, \pi^2, \ldots, \pi^{m-1}$ are defined as well as $\pi^m_{j-1} : \text{before}(j) \to E_\perp$. We define π^m_j as follows. Consider a player i with $\nu(i) = j$. The resource Player i chooses in the m-th phase is $f_i(\pi^1, \ldots, \pi^{m-1}, \pi^m_{j-1})$. Finally, we define $\pi^m = \pi^m_{t_\nu}$.

We restrict attention to *legal* strategies for the players, namely ones in which the collection of resources chosen by Player i in all phases is an objective in Σ_i [5]. Also, once Player i chooses $\perp$, then he has finished playing and all his choices in future phases must also be $\perp$. Formally, for a profile $P = \langle f_1, \ldots, f_k \rangle$ with $out(P) = \pi^1, \pi^2, \ldots$ and $i \in [k]$, let $out_i(P)$ be $\pi^1(i), \pi^2(i), \ldots$. For $j \geqslant 1$, let $e_j = \pi^j(i)$ be the resource Player i selects in the j-th phase. Thus, $out_i(P)$ is an infinite sequence over $E_\perp$. We say that f_i is legal if (1) there is an index m such that $e_j \in E$ for all $j < m$ and $e_j = \perp$ for all $j \geqslant m$, and (2) the set $\{e_1, \ldots, e_{m-1}\}$ is an objective in Σ_i. (In particular, a player cannot select a resource multiple times nor a resource that is not a member in his chosen objective). We refer to an outcome in which the players use legal strategies as a *legal outcome* and a prefix of a legal outcome as a *legal history*.

In $out(P)$, every player selects a set of resources. The cost of a player is calculated similarly to RAGs. That is, his cost for a resource e, assuming the load on it is γ, is $\ell_e(\gamma)$, and his total cost is the sum of costs of the resources he uses. When the outcome of a profile P in a dynamic RAG coincides with the outcome of a profile Q in a RAG G, we say that P and Q are *matching* profiles.

Equilibrium concepts A *Nash equilibrium*[6] (NE, for short) in a game is a profile in which no player has an incentive to unilaterally deviate from his strategy. Formally, for a profile P, let $P[i \leftarrow f'_i]$ be the profile in which Player i switches to the strategy f'_i and all other players use their strategies in P. Then, a profile P is an NE if for every $i \in [k]$ and every legal strategy f'_i for Player i, we have $cost_i(P) \leqslant cost_i(P[i \leftarrow f'_i])$. It is well known that every RAG is guaranteed to have an NE [21].

The definition of NE applies to all games, in particular to dynamic ones. Every NE Q in a RAG G matches an NE in a dynamic game $\langle G, \nu \rangle$, for some scheduler ν, in which the players ignore the history of the play and follow their objectives in Q. However, such a strategy is not rational. Thus, one could argue that an NE is not necessarily achievable in a dynamic setting. We illustrate this in the following example.

Example 2. Consider a two-player DCG with resources $\{a, b\}$, latency functions $\ell_a(x) = x$ and $\ell_b(x) = 1.5x$, and objectives $\Sigma_1 = \Sigma_2 = \{\{a\}, \{b\}\}$. Consider the sequential scheduling in which Player 1 moves first followed by Player 2. Since the players' objectives are singletons, the dynamic game consists of one phase. Consider the Player 2 strategy f_2 that "promises" to select the resource a no matter what Player 1 selects, thus

[5] It is interesting to allow players to use "redundant resources"; a player's choice of resources should contain one of his objectives. While in the traditional setting, using a redundant resource cannot be beneficial, in the dynamic setting, it is, as a variant of Example 1 demonstrates.

[6] Throughout this paper, we consider *pure* strategies as is the case in the vast literature on RAGs.

$f_2(a) = f_2(b) = a$. Let f_1^a and f_1^b be the Player 1 strategies in which he selects a and b, respectively, thus $f_1^a(\epsilon) = a$ and $f_1^b(\epsilon) = b$, where ϵ denotes the empty history. Note that these are all of Player 1's possible strategies. The profile $P = \langle f_1^b, f_2 \rangle$ is an NE. Indeed, Player 2 pays 1, which is the least possible payment, so he has no incentive to deviate. Also, by deviating to f_1^a, Player 1's payoff increases from 1.5 to 2, so he has no incentive to deviate either. Note, however, that this strategy of Player 2 is not rational. Indeed, when it is Player 2's turn, he is aware of Player 1's choice. If Player 1 plays f_1^a, then a rational Player 2 is not going to choose a, as this results in a cost of 2, whereas by b, his cost will be 1.5. Thus, an NE profile with f_2 may not be achievable. □

To overcome this issue, the notion of *subgame perfect equilibrium* (SPE, for short) was introduced. In order to define SPE, we need to define a subgame of a dynamic game. Let $\mathcal{G} = \langle G, \nu \rangle$. It is helpful to consider the *outcome tree* $\mathcal{T}_\mathcal{G}$ of $\mathcal{G}$, which is a finite rooted tree that contains all the legal histories of $\mathcal{G}$. Each internal node in $\mathcal{T}_\mathcal{G}$ corresponds to a legal history, its successors correspond to possible extensions of the history, and each leaf corresponds to a legal outcome. Consider a legal history h. We define a dynamic RAG $\mathcal{G}_h$, which, intuitively, is the same as $\mathcal{G}$ after the history h has been played. More formally, the outcome tree of $\mathcal{G}_h$ is the subtree $\mathcal{T}_\mathcal{G}^h$ whose root is the node h. We define the costs in $\mathcal{G}_h$ so that the costs of the players in the leaves of $\mathcal{T}_\mathcal{G}^h$ are the same as the corresponding leaves in $\mathcal{T}_\mathcal{G}$. Assume that h ends at the m-th turn. A profile P in $\mathcal{G}$ corresponds to a trimming of $\mathcal{T}_\mathcal{G}$ in which the internal node h has exactly one child $h \cdot \overline{\sigma}$, where $\overline{\sigma}$ is the set of choices of the players in $\nu^{-1}(m)$ when they play according to their strategies in P. The profile P induces a profile P^h in $\mathcal{G}_h$, where the trimming of $\mathcal{T}_\mathcal{G}^h$ according to P^h coincides with the trimming of $\mathcal{G}$ according to P. We formally define the outcome tree and a subgame in the full version.

Definition 1. *A profile P is an SPE if for every legal history h, the profile P^h is an NE in $\mathcal{G}_h$.*

Note that the profile $P = \langle f_1^b, f_2 \rangle$ in the example above is an NE but not an SPE. Indeed, for the history $h = a$, the profile P^h is not an NE in $\mathcal{G}_h$ as Player 2 can benefit from unilaterally deviating as described above.

3 Existence of SPE in Dynamic Congestion Games

It is easy to show that every sequential dynamic game has an SPE by unwinding the outcome tree, and similar results have been shown before (c.f., [15]). The proof can be found in the full version.

Theorem 1. *Every sequential dynamic game has an SPE.*

One could hope to prove that a general dynamic game $\mathcal{G}$ also has an SPE using a similar unwinding of $\mathcal{T}_\mathcal{G}$, possibly using the well-known fact that every CG is guaranteed to have an NE [21]. Unfortunately, and somewhat surprisingly, we show that this is not possible. We show that (very restrictive) DCGs might not have an SPE. For the good news, we identify a maximal fragment that is guaranteed to have an SPE.

Recall that a CG is singleton when the players' objectives consist of singletons of resources, and a CG is symmetric if all the players agree on their objectives. We start with the bad news and show that symmetric DCGs and singleton DCGs need not have an SPE, even with linear latency functions. We then show that the combination of these two restrictions is sufficient for existence of an SPE in a DCG.

Theorem 2. *There are symmetric and singleton linear DCGs with no SPE.*

Proof. We first describe a linear DCG with no SPE, and then alter it to make it symmetric. The proof for singleton linear DCG is given in the full version.Consider the following three-player linear CG G with resources $E = \{a, a', b, b', c\}$ and linear latency functions $\ell_a(x) = \ell_b(x) = x$, $\ell_{a'}(x) = \frac{3}{4}x$, $\ell_{b'}(x) = 1\frac{1}{4}$, and $\ell_c(x) = x + \frac{2}{3}$. Let $\Sigma_1 = \Sigma_2 = \{\{a, a'\}, \{b, b'\}, \{c\}\}$ and $\Sigma_3 = \{\{c\}, \{a', b\}\}$. Consider the dynamic game $\mathcal{G}$ in which Players 1 and 2 move concurrently followed by Player 3. Formally, $\mathcal{G} = \langle G, \nu \rangle$, where $\nu(1) = \nu(2) = 1$ and $\nu(3) = 2$.

We claim that there is no SPE in $\mathcal{G}$. Note that since the players' objectives are disjoint, then once a player reveals the first choice of resource, he reveals the whole objective he chooses, thus we analyze the game as if it takes place in one phase in which the players' reveal their whole objective. The profiles in which Players 1 and 2 choose the same objective are clearly not a SPE as they are not an NE in the game $\mathcal{G}_\epsilon$. As for the other profiles, in Figure 2, we go over half of them, and show that none of them is an SPE. The other half is analogous. The root of each tree is labeled by the objectives of Players 1 and 2, and its branches according to Player 3's objectives. In the leaves we state Player 3's payoff. In an SPE, Player 3 performs a best-response according to the objectives he observes as otherwise the subgame is not in an NE. We depict his choice with a bold edge. Beneath each tree we note the payoffs of all the players in the profile, and the directed edges represent the player that can benefit from unilaterally deviating. In the full version, we construct a symmetric DCG $\mathcal{G}'$ by altering the game $\mathcal{G}$ above. We do this by adding a fourth player and three new resources so that $\mathcal{G}'$ simulates $\mathcal{G}$. □

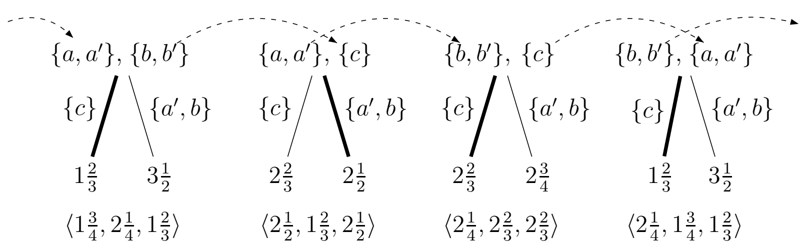

Fig. 2. Profiles in the game $\mathcal{G}$ with no SPE.

We now prove that combining the two restrictions does guarantee the existence of SPE. We note that while our negative results hold for linear DCGs, which tend to be stabler than other DCGs, our positive result holds for every increasing latency function.

Theorem 3. *Every symmetric singleton DCG has an SPE.*

Proof. Consider a symmetric singleton DCG $\mathcal{G} = \langle G, \nu \rangle$. Recall that since G is a singleton game, every outcome of $\mathcal{G}$ consists of one phase. Let P be an NE in G (recall that according to [21] an NE exists in every CG). Since G is symmetric, we can assume that, for $1 \leqslant j < k$, the players that move in the j-th turn do not pay more than the players that move after them. Formally, for $i, i' \in [k]$, if $\nu(i) < \nu(i')$, then $cost_i(P) \leqslant cost_{i'}(P)$. In particular, the players who move in the first turn pay the least, and the players that move in the last turn pay the most. We construct a profile Q in $\mathcal{G}$ and show that it is an SPE. Intuitively, in Q, the players follow their objectives in P assuming the previous players also follow it. Since the costs are increasing with turns, if Player i deviates, a following Player j will prefer switching resources with Player i and also switching the costs. Thus, the deviation is not beneficial for Player i. In the full version, we construct Q formally and prove that it is an SPE. □

4 Existence of SPE in Dynamic Cost-sharing Games

Cost sharing games tend to be less stable than congestion games in the concurrent setting; for example, very simple fragments of multiset cost-sharing games do not have an NE [5] while linear multiset congestion games are guaranteed to have an NE [6]. In this section we are going to show that, surprisingly, there are classes of games in which an SPE exists only in the cost-sharing setting. Still, SPE is not guaranteed to exist in general DSGs. We start with the bad news.

Theorem 4. *There is a DSG with no SPE.*

Proof. Consider the following four-player SG G with resources $E = \{a, a', a'', b, b', b'', c, c', c''\}$ and costs $c_a = c_b = c_c = 6$, $c_{a'} = c_{b'} = c_{c'} = 4$, and $c_{a''} = c_{b''} = c_{c''} = 3$. Let $\Sigma_1 = \{\{a, a'\}, \{b, b''\}\}$, $\Sigma_2 = \{\{b, b'\}, \{c, c''\}\}$, $\Sigma_3 = \{\{c, c'\}, \{a, a''\}\}$, and $\Sigma_4 = \{\{a, a'\}, \{b, b'\}, \{c, c'\}\}$. Consider the dynamic game $\mathcal{G}$ in which players 1,2, and 3 move concurrently followed by Player 4. Formally, $\mathcal{G} = \langle G, \nu \rangle$, where $\nu(1) = \nu(2) = \nu(3) = 1$ and $\nu(4) = 2$.

We claim that there is no SPE in $\mathcal{G}$. Similar to Theorem 2, since the players' objectives are disjoint, we analyze the game as if it takes place in one phase. In Figure 3, we depict some of the profiles and show that none of them are an SPE. As in Theorem 2, the root of each tree is labeled by the objectives of Players $1, 2$, and 3, its branches according to Player 4's choices, and in the leaves we state the cost of Player 4 assuming he chooses his best choice given the other players' choices. Finally, it is not hard to show that every profile not on the cycle of profiles cannot be an SPE. □

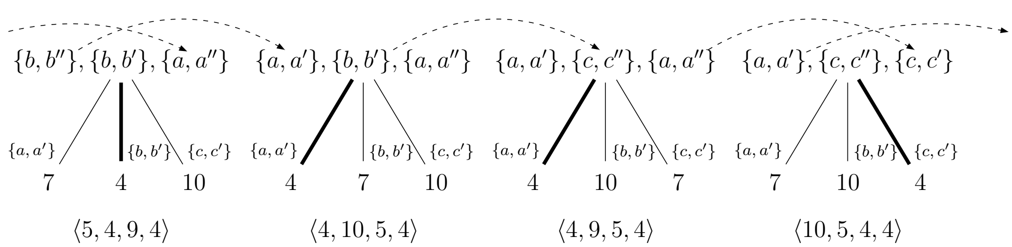

Fig. 3. Profiles in the game with no SPE. Bold edges depict Player 4's best choice given the other players choices. Directed edges represent the player that can benefit from unilaterally deviating.

Recall that singleton DCGs are not guaranteed to have an SPE (Theorem 2). On the other hand, we show below that singleton DSGs are guaranteed to have an SPE. In order to find an SPE in such a game, we use a firmer notion of an equilibria in SGs.

A *strong equilibrium* (SE, for short) [3] is a profile that is stable against deviations of *coalitions* of players rather than deviations of a single player as in NEs (see the full version for a formal definition). We show a connection between strong equilibria and SPEs in singleton SGs. It is shown in [13] that every singleton SG has an SE.

Theorem 5. *Consider a singleton DSG $\mathcal{G} = \langle G, \nu \rangle$. Then, every strong equilibrium in G matches an SPE of $\mathcal{G}$. In particular, every singleton DSG has an SPE.*

Proof. We describe the intuition of the proof and the details can be found in the full version. Consider a singleton DSG $\mathcal{G} = \langle G, \nu \rangle$, and let Q be an SE in G. We describe a profile P in $\mathcal{G}$ that matches Q, and we claim that it is an SPE. Consider a history h that ends in the i-th turn. Assume the players that play in h follow their objective in Q. Then, the players who play next, namely these in $\nu^{-1}(i+1)$, also follow Q. Thus, P matches Q. The definition of the strategies in P for histories that do not follow Q is inductive: assume only the players in $\nu^{-1}(i)$ choose differently than in Q, then the subgame $\mathcal{G}_h$ is a singleton DSG. We find a strong equilibrium in $\mathcal{G}_h$ and let the players in $\nu^{-1}(i+1)$ choose according to it. In order to show that no Player i can unilaterally benefit from deviating to a resource e from P, we observe that it is not possible that all players that deviate into e decrease their costs (as Q is an SE). So, there must be a Player j_1 that deviates from some resource e' to e and increases his cost. This can only happen if there is a Player j_2 that also uses e' in Q and deviates to e'' while decreasing his cost. The same reasoning holds for players deviating to e''. Thus, we find a sequence of resources, which must contain a loop as there are finitely many resources. Using it we can reach a contradiction to the fact that Player i benefits. □

5 Equilibrium Inefficiency

It is well known that decentralized decision-making may lead to sub-optimal solutions from the point of view of society as a whole. We define the cost of a profile P, denoted $cost(P)$, to be $\sum_{i \in [k]} cost_i(P)$. We denote by OPT the cost of a social-optimal solution; i.e., $OPT = \min_P cost(P)$. Two standard measures that quantify the inefficiency incurred due to self-interested behavior are the *price of anarchy* (PoA) [14,19] and *price of stability* (PoS) [2,23]. The PoA is the worst-case inefficiency of an NE; The PoA of a game G is the ratio between the cost of the most expensive NE and the cost of the social optimum. The PoS measures the best-case inefficiency of an NE, and is defined similarly with the cheapest NE. The PoA of a family of games $\mathcal{F}$ is $\sup_{G \in \mathcal{F}} \text{PoA}(G)$, and the definition is similar for PoS.

In dynamic games we consider SPE rather than NE. We adapt the definitions above accordingly, and we refer to the new measures as *dynamic PoA* and *dynamic PoS* (DPoA and DPoS, for short). We study the equilibrium inefficiency in the classes of games that are guaranteed to have an SPE, namely singleton DSGs and symmetric singleton DCGs.

The lower bounds for the PoA and PoS for singleton SG and singleton symmetric CGs follow to the dynamic setting as we can consider the scheduler in which all players choose simultaneously in the first turn. For the upper bound we start with the DPoS.

In the congestion setting, we show that every NE in the underlying RAG matches an SPE. In the cost-sharing setting, recall that an SE in the traditional game matches an SPE in the dynamic game, and by [26], a singleton SG has an SE whose cost is at most $\log(k) \cdot OPT$. This matches the $\log(k)$ lower bound. We continue to study DPoA. In the cost-sharing setting, the upper bound follows from the same argument as traditional games. For congestion games, it follows by applying a recent result by [10] to our setting. The details can be found in the full version.

Theorem 6. *The DPoA and DPoS in singleton DSGs and singleton symmetric DCGs coincide with the PoA and PoS in singleton SGs and singleton symmetric CGs, respectively.*

Thus, for singleton DSGs we have DPoA $= k$ and DPoS $= \log k$ [2], and for singleton symmetric DCGs we have DPoA $= 4/3$ [11] and we are not aware of bounds for the PoS in the corresponding CGs.

6 Deciding the Existence of SPE

In the previous sections we showed that dynamic RAGs are not guaranteed to have an SPE. A natural decision problem arises, which we refer to as ∃SPE: given a dynamic RAG, decide whether it has an SPE. We show that the problem is PSPACE-complete in DSGs and DCGs. We start with the lower bound. The crux of the proof is given in the following lemma. For DCGs, such a construction is described in [15], which uses a construction by [25] in order to simulate the logic of a NAND gate by means of a CG. For SGs we are not aware of a similar known result. We describe the construction in the full version, which is inspired by the construction in [15].

Lemma 1. *Given a QBF instance ψ, there is a fully sequential game $\mathcal{G}_\psi$ that is either a DCG or a DSG, and two constants $\gamma, \delta > 0$, such that in every SPE P in $\mathcal{G}_\psi$, (1) if ψ is true, then $cost_1(P) < \gamma$, and (2) if ψ is false, then $cost_1(P) > \delta$.*

To conclude the lower-bound proof, we combine the game that is constructed in Lemma 1 and a game that has no SPE as in the examples we show in the previous sections. For the upper bound, consider a dynamic RAG $\mathcal{G}$, and let $\mathcal{T}_\mathcal{G}$ be the outcome tree of $\mathcal{G}$. Recall that there is a one-to-one correspondence between leaves in $\mathcal{T}_\mathcal{G}$ and legal outcomes of $\mathcal{G}$. In order to decide in PSPACE whether $\mathcal{G}$ has an SPE, we guess a leaf l in $\mathcal{T}_\mathcal{G}$ and verify that it is an outcome of an SPE. Thus, we ask if there is an SPE P in $\mathcal{G}$ whose outcome corresponds to l.

Theorem 7. *The ∃SPE problem is PSPACE-complete for dynamic RAGs.*

7 Extensions

In the full version we study two extensions of the dynamic setting. In the first, we consider the problem of finding a schedule that admits an SPE under given constraints on the order the players move, and show that this problem is also PSPACE-complete. Then, we consider dynamic RAGs in which there is an order on the resources that the players choose. So, if for two resources e_1 and e_2, we have $e_1 \prec e_2$, then a player cannot choose e_1 in a later phase than e_2. The motivation for an order on resources is natural. For example, returning to network formation games, a driver can only extend the path he chooses as the choices are made during driving. We show that all our results carry over to the ordered case.

References

1. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.
2. E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM J. Comput.*, 38(4):1602–1623, 2008.
3. R. Aumann. Acceptable points in games of perfect information. *Contributions to the Theory of Games*, 4:287–324, 1959.
4. G. Avni and O. Kupferman. Synthesis from component libraries with costs. In *Proc. 25th CONCUR*, pp. 156–172, 2014.
5. G. Avni, O. Kupferman, and T. Tamir. Network-formation games with regular objectives. In *Proc. 17th FoSSaCS*, 8412 LNCS, pp. 119–133. Springer, 2014.
6. G. Avni, O. Kupferman, and T. Tamir. Congestion games with multisets of resources and applications in synthesis. In *Proc. 35th FSTTCS*, pp. 365–379, 2015.
7. V. Bilò, A. Fanelli, and L. Moscardelli. On lookahead equilibria in congestion games. In *Proc. 9th WINE*, pp. 54–67, 2013.
8. J. R. Correa, J. de Jong, B. de Keijzer, and M. Uetz. The curse of sequentiality in routing games. In *Proc. 11th WINE*, pp. 258–271, 2015.
9. H. Chen and T. Roughgarden. Network design with weighted players. *Theory Comput. Syst.*, 45(2):302–324, 2009.
10. J. de Jong and M. Uetz. The sequential price of anarchy for atomic congestion games. In *Proc. 10th WINE*, pp. 429–434, 2014.
11. D. Fotakis. Stackelberg strategies for atomic congestion games. In *Proc. 15th ESA*, 2007.
12. T. Harks and M. Klimm. On the existence of pure Nash equilibria in weighted congestion games. *Math. Oper. Res.*, 37(3):419–436, 2012.
13. R. Holzman and N. Law-Yone. Strong equilibrium in congestion games. *Games and Economic Behavior*, 21(1-2):85–101, 1997.
14. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *CS Review*, 3(2):65–69, 2009.
15. R. P. Leme, V. Syrgkanis, and E. Tardos. The curse of simultaneity. In *Proc. 3rd ITCS*, 2012.
16. Y. Lustig and M.Y. Vardi. Synthesis from component libraries. *STTT*, 15:603–618, 2013.
17. V. S. Mirrokni, N. Thain, and A. Vetta. A theoretical examination of practical game playing: Lookahead search. In *Proc. 5th SAGT*, pp. 251–262, 2012.
18. J. Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320.
19. C. Papadimitriou. Algorithms, games, and the internet. In *Proc. 33rd STOC*, 2001.
20. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th POPL*, pp. 179–190, 1989.
21. R.W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
22. T. Roughgarden and E. Tardos. How bad is selfish routing? *JACM*, 49(2):236–259, 2002.
23. A. S. Schulz and N. E. Stier Moses. On the performance of user equilibria in traffic networks. In *Proc. 14th SODA*, pp. 86–87, 2003.
24. R. Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft*, 121, 1965.

25. A. Skopalik and B. Vöcking. Inapproximability of pure Nash equilibria. In *Proc. 40th STOC*, pp. 355–364, 2008.
26. V. Syrgkanis. The complexity of equilibria in cost sharing games. In *Proc. 6th WINE*, pp. 366–377, 2010.