Network-Formation Games with Regular Objectives[☆]

Guy Avni[a], Orna Kupferman[a], Tami Tamir[b]

[a] *School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel*
[b] *School of Computer Science, The Interdisciplinary Center, Herzliya, Israel*

Abstract

Classical network-formation games are played on a directed graph. Players have reachability objectives, and each player has to select a path satisfying his objective. Edges are associated with costs, and when several players use the same edge, they evenly share its cost. The theoretical and practical aspects of network-formation games have been extensively studied and are well understood. We introduce and study *network-formation games with regular objectives*. In our setting, the edges are labeled by alphabet letters and the objective of each player is a regular language over the alphabet of labels, given by means of an automaton or a temporal-logic formula. Thus, beyond reachability properties, a player may restrict attention to paths that satisfy certain properties, referring, for example, to the providers of the traversed edges, the actions associated with them, their quality of service, security, etc.

Unlike the case of network-formation games with reachability objectives, here the paths selected by the players need not be simple, thus a player may traverse some transitions several times. Edge costs are shared by the players with the share being proportional to the number of times the transition is traversed. We study the existence of a pure Nash equilibrium (NE), convergence of best-response-dynamics, the complexity of finding the social optimum, and the inefficiency of a NE compared to a social-optimum solution. We examine several classes of networks (for example, networks with uniform edge costs, or alphabet of size 1) and several classes of regular objectives. We show that many properties of classical network-formation games are

[☆] The article is based on the conference publications [7] and [8].

Email addresses: guy.avni@mail.huji.ac.il (Guy Avni), orna@cs.huji.ac.il (Orna Kupferman), tami@idc.ac.il (Tami Tamir)

no longer valid in our game. In particular, a pure NE might not exist and the Price of Stability equals the number of players (as opposed to logarithmic in the number of players in the classic setting, where a pure NE always exists). In light of these results, we also present special cases for which the resulting game is more stable.

Keywords: Network-formation games, On-going behaviors, Weighted Automata,
Nash equilibria, Inefficiency,

1. Introduction

Network design and formation is a fundamental well-studied challenge that involves many interesting combinatorial optimization problems. In practice, network design is often conducted by multiple strategic users whose individual costs are affected by the decisions made by others. Early works on network design focus on analyzing the efficiency and fairness properties associated with different sharing rules (e.g., [28, 36]). Following the emergence of the Internet, there has been an explosion of studies employing game-theoretic analysis to explore Internet applications, such as routing in computer networks and network formation [21, 1, 16, 2]. In network-formation games (for a survey, see [43]), the network is modeled by a weighted graph. The weight of an edge indicates the cost of activating the transition it models, which is independent of the number of times the edge is used. Players have reachability objectives, each given by sets of possible source and target nodes. Players share the cost of edges used in order to fulfill their objectives. Since the costs are positive, the runs traversed by the players are simple. Under the common Shapley cost-sharing mechanism, the cost of an edge is shared evenly by the players that use it.

The players are selfish agents who attempt to minimize their own costs, rather than to optimize some global objective. In network-design settings, this would mean that the players selfishly select a path instead of being assigned one by a central authority. The focus in game theory is on the *stable* outcomes of a given setting, or the *equilibrium* points. A Nash equilibrium (NE) is a profile of the players' strategies such that no player can decrease his cost by an unilateral deviation from his current strategy, that is, assuming that the strategies of the other players do not change.[1]

Reachability objectives enable the players to specify possible sources and targets. Often, however, it is desirable to refer also to other properties of the selected paths. For example, in a *communication* setting, edges may belong to different providers, and a user may like to specify requirements like “all edges are operated by the same provider” or “no edge operated by AT&T is followed by an edge operated by Verizon”. Edges may also have different quality or security levels (e.g., “noisy channel”, “high-bandwidth channel”, or “encrypted channel”), and again, users may like to specify their preferences with respect to these properties. In *planning* or in *production systems*, nodes of the network correspond to configurations, and edges correspond to the application of actions. The objectives of the players are sequences of actions that fulfill a certain plan, which is often more involved than just reachability [25]; for example “once the arm is up, do not put it down until the block is placed”.

The challenge of reasoning about behaviors has been extensively studied in the context of formal verification. While early research concerned the input-output relations of terminating programs, current research focuses on on-going behaviors of reactive systems [26]. The interaction between the components of a reactive system correspond to a multi-agent game, and indeed in recent years we see an exciting transfer of concepts and ideas between the areas of game theory and formal verification: logics for specifying multi-agent systems [3, 13], studies of equilibria in games that correspond to the synthesis problem [12, 11, 20], an extension of mechanism design to on-going behaviors [31], studies of non-zero-sum games in formal methods [14, 10], and more.

In this paper we extend network-formation games to a setting in which the players can specify regular objectives. This involves two changes of the underlying setting: First, the edges in the network are labeled by letters from a designated alphabet. Second, the objective of each player is specified by a *language* over this alphabet. Each player should select a path labeled by a word in his objective language. Thus, if we view the network as a *nondeterministic weighted finite automaton* (WFA) $\mathcal{A}$, then the

[1] Throughout this paper, we concentrate on pure strategies and pure deviations, as is the case for the vast literature on cost-sharing games.

set of strategies for a player with objective L is the set of accepting runs of $\mathcal{A}$ on some word in L. Accordingly, we refer to our extension as *automaton-formation games*. As in classical network-formation games, players share the cost of edges they use. Unlike the classical game, the runs selected by the players need not be simple, thus a player may traverse some edges several times. Edge costs are shared by the players, with the share being proportional to the number of times the edge is traversed. This latter issue is the main technical difference between automaton-formation and network-formation games, and as we shall see, it is very significant.

Many variants of cost-sharing games have been studied. A generalization of the network-formation game of [2], in which players are weighted and a player's share in an edge cost is proportional to its weight is considered in [15], where it is shown that the weighted game does not necessarily have a pure NE. In congestion games, sharing of a resource increases its cost. Studied variants of congestion games include settings in which players' payments depend on the resource they choose to use, the set of players using this resource, or both [35, 32, 33, 23]. In some of these variants a pure NE is guaranteed to exist while in others it is not.

Since a path a player selects in an automaton-formation game need not be simple, a path corresponds to a *multiset* of edges. Thus, automaton-formation games can be viewed as a special case of *multiset resource-allocation games*, where players' strategies consist of multisets of resources. These games are general and subsume previously studied models such as weighted resource-allocation games [33], where each Player i has a weight w_i, and when he selects a subset of resources, he adds a load of w_i on the resources in the selected set. Closer to our multiset games are network routing games in which flow can be split into integral fractions [42] and its generalization to resource-allocation games [27]. There, again each player has a weight, only that he can split the weight between several strategies, assigning an integral weight to each strategy. The relation between automaton-formation games and multiset resource-allocation games is analogue to the relation between network-formation games and resource-allocation games, in the sense that each network-formation game can be viewed as a resource-allocation game, where each simple path in the network corresponds to a subset of the edges. Unlike the case of network-formation games, however, the richness of automaton-formation games make them sufficiently expressive to model every multiset resource-allocation game. Essentially (see Remark 2.1 for the detailed reduction), by associating each resource with a letter from the alphabet, we can translate each strategy in the multiset resource-allocation game to a word that should be traversed in a single-state network in which each resource induces a self-loop. Thus, our results apply also to the (seemingly) more general setting of multiset resource allocation game. Moreover, while the objectives in the resource-allocation game setting are given explicitly, in the automaton-formation game setting they are given symbolically by means of regular languages. This succinctness of the symbolic approach is very significant. In particular, there may be infinitely many strategies to fulfill an objective in an automaton-formation games.

The fact automaton-formation games capture all multiset resource allocation games extends the application of our work. In the context of formal methods, an appealing application of resource-allocation games is that of *synthesis from components*, where the resources are components from a library, and agents need to synthesize their objectives using the components, possibly by a repeated use of some components. In some settings, the components have construction costs (e.g., the money paid to the designer of the component), in which case the corresponding multiset game is a cost-sharing game [6], and our results here can be generalized to apply for this settings. In other settings, the components have congestion effects (e.g., the components are CPUs, and the more players that use them, the slower the performance is), in which case the corresponding game is a multiset congestion game [9].

We study the theoretical and practical aspects of automaton-formation games. In addition to the general game, we consider classes of instances that have to do with the network, the specifications, or to their combination. Recall that the network can be viewed as a WFA $\mathcal{A}$. We consider the following classes of WFAs: (1) *all-accepting*, in which all the states of $\mathcal{A}$ are accepting, thus its language is prefix closed (2) *uniform costs*, in which all edges have the same cost, and (3) *single letter*, in which $\mathcal{A}$ is over a single-letter alphabet. We consider the following classes of specifications: (1) *single word*, where the language of each player is a single word, (2) *symmetric*, where all players have the same objective. We also consider classes of instances that are intersections of the above classes.

Each of the restricted classes we consider corresponds to a real-life variant of the general setting. Let us elaborate below on single-letter instances. The language of an automaton over a single letter $\{a\}$ induces a subset of $\mathbb{N}$, namely the numbers $k \in \mathbb{N}$ such that the automaton accepts a^k. Accordingly, single-letter instances correspond to settings in which a player specifies possible lengths of paths. Several communication protocols are based on the fact that a message must pass a pre-defined length before reaching its destination. This includes *onion routing*, where the message is encrypted in layers [40], or *proof-of-work* protocols that are used to deter denial of service attacks and other service abuses such as spam (e.g., [19]).

We provide a complete picture of the following questions for various instances (for formal definitions, see Section 2): (i) Existence of a *pure Nash equilibrium*. That is, whether each instance of the game has a profile of pure strategies that constitutes a NE. As we show, unlike the case of classical network design games, a pure NE might not exist in general automaton-formation games and even in very restricted instances of it. (ii) The complexity of finding the *social optimum* (SO). The SO is a profile that minimizes the total cost of the edges used by all players; thus the one obtained when the players obey some centralized authority. We show that for some restricted instances finding the SO can be done efficiently, while for other restricted instances, the complexity agrees with the NP-completeness of classical network-formation games. (iii) An analysis of *equilibrium inefficiency*. 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. We quantify the inefficiency incurred due to selfish behavior according to the *price of anarchy* (PoA) [30, 38] and *price of stability* (PoS) [2] measures. The PoA is the worst-case inefficiency of a Nash equilibrium (that is, the ratio between the worst NE and the SO). The PoS is the best-case inefficiency of a Nash equilibrium (that is, the ratio between the best NE and the SO). We show that while the PoA in automaton-formation games agrees with the one in classical network-formation games and is equal to the number of players, the PoS also equals the number of players, again already in very restricted instances. This is in contrast with classical network-formation games, where the PoS tends to log the number of players. Thus, the fact that players may choose to use edges several times significantly increases the challenge of finding a stable solution as well as the inefficiency incurred due to selfish behavior. We find this as the most technically challenging result of this work. We do manage to find structural restrictions on the network with which the social optimum is a NE.

The technical challenge of our setting is demonstrated in the seemingly easy instance in which all players have the same objective. Such *symmetric* instances are known to be the simplest to handle in all cost-sharing and congestion games studied so far. Specifically, in network-formation games, the social optimum in symmetric instances is also a NE and the PoS is 1. Moreover, in some games [22], computing a NE is PLS-complete in general, but solvable in polynomial time for symmetric instances. Indeed, once all players have the same objective, it is not conceivable that a player would want to deviate from the social-optimum solution, where each of the k players pays $\frac{1}{k}$ of the cost of the optimal solution. We show that, surprisingly, symmetric instances in automaton-formation games are not simple at all. Specifically, a NE is not guaranteed to exist in the general case, and in single-letter networks, the social optimum might not be a NE, and the PoS is at least $\frac{k}{k-1}$. In particular, for symmetric two-player automaton-formation games, we have that $PoS = PoA = 2$. We also show that the PoA equals the number of players already for very restricted instances.

Paper Organization: In Section 2 we provide a formal description of automaton-formation games, define the cost-sharing mechanism, and introduce some special classes we are going to study. We also define how inefficiency of pure Nash equilibrium is quantified in these games. In Section 3 we study the existence of pure NE and analyze the equilibrium inefficiency. Then, in Section 4, we analyze the computational complexity of three problems: finding the cost of a social optimum, finding the best-response of a player, and deciding the existence of a pure NE. In Section 5 we define the family of resistant semi-weak instances, and show that for every such instance, a pure NE is guaranteed to exist, finding the social optimum can be dome efficiently, and that the Price of Stability is 1. Finally, in Section 6 we consider the class of symmetric instances.

2. Preliminaries

2.1. Automaton-formation games

A *nondeterministic finite weighted automaton* on finite words (WFA, for short) is a tuple $\mathcal{A} = \langle \Sigma, Q, \Delta, q_0, F, c \rangle$, where Σ is a finite alphabet, Q is a finite set of states, $\Delta \subseteq Q \times \Sigma \times Q$ is a transition relation, $q_0 \in Q$ is an initial state, $F \subseteq Q$ is a set of accepting states, and $c : \Delta \to \mathbb{R}^{\geq 0}$ is a function that maps each transition to the cost of its formation [34]. A *run* of $\mathcal{A}$ on a word $w = w_1, \ldots, w_n \in \Sigma^*$ is a sequence of states $\pi = \pi^0, \pi^1, \ldots, \pi^n$ such that $\pi^0 = q_0$ and for every $0 \leq i < n$ we have $\Delta(\pi^i, w_{i+1}, \pi^{i+1})$. The run π is *accepting* iff $\pi^n \in F$. The *length* of π is n, whereas its size, denoted $|\pi|$, is the number of different transitions it traverses. Note that $|\pi| \leq n$. It is sometimes convenient to view π as a sequence of transitions rather than states, and we note when we do so when it is not clear from the context.

An *automaton-formation game* (AF game, for short) between k selfish players is a pair $\langle \mathcal{A}, O \rangle$, where $\mathcal{A}$ is a WFA over some alphabet Σ and O is a k-tuple of regular languages over Σ. Thus, the objective of Player i is a regular language L_i, and he needs to choose a word $w_i \in L_i$ and an accepting run of $\mathcal{A}$ on w_i in a way that minimizes his payments. The cost of each transition is shared by the players that use it in their selected runs, where the share of a player in the cost of a transition e is proportional to the number of times e is used by the player. Formally, the set of strategies for Player i is $\mathcal{S}_i = \{\pi : \pi \text{ is an accepting run of } \mathcal{A} \text{ on some word in } L_i\}$. We assume that $\mathcal{S}_i$ is not empty. We refer to the set $\mathcal{S} = \mathcal{S}_1 \times \ldots \times \mathcal{S}_k$ as the set of *profiles* of the game.

Consider a profile $P = \langle \pi_1, \pi_2, \ldots, \pi_k \rangle$. We refer to π_i as a sequence of transitions. Let $\pi_i = e_i^1, \ldots, e_i^{\ell_i}$, and let $\eta_P : \Delta \to \mathbb{N}$ be a function that maps each transition in Δ to the number of times it is traversed by all the strategies in P, taking into an account several traversals in a single strategy. Denote by $\eta_i(e)$ the number of times e is traversed in π_i, that is, $\eta_i(e) = |\{1 \leq j \leq \ell_i : e_i^j = e\}|$. When $\eta_i(e) > 0$, we say that e is in π_i, denoted $e \in \pi_i$. Then, $\eta_P(e) = \sum_{i=1 \ldots k} \eta_i(e)$. The *cost of Player i in the profile P* is

$$cost_i(P) = \sum_{e \in \pi_i} \frac{\eta_i(e)}{\eta_P(e)} c(e). \tag{1}$$

For example, consider the WFA $\mathcal{A}$ depicted in Figure 1. The label $e_1 : a, 1$ on the transition from q_0 to q_1 indicates that this transition, which we refer to as e_1, traverses the letter a and its cost is 1. We consider a game between two players. Player 1's objective is the language $L_1 = \{ab^i : i \geq 2\}$ and Player 2's language is $\{ab, ba\}$. Thus, $\mathcal{S}_1 = \{\{e_1, e_2, e_2\}, \{e_1, e_2, e_2, e_2\}, \ldots\}$ and $\mathcal{S}_2 = \{\{e_3, e_4\}, \{e_1, e_2\}\}$. Consider the profile $P = \langle \{e_1, e_2, e_2\}, \{e_3, e_4\}\rangle$, the strategies in P are disjoint, and we have $cost_1(P) = 2 + 2 = 4, cost_2(P) = 1 + 3 = 4$. For the profile $P' = \langle \{e_1, e_2, e_2\}, \{e_1, e_2\}\rangle$, it holds that $\eta_1(e_1) = \eta_2(e_1)$ and $\eta_1(e_2) = 2 \cdot \eta_2(e_2)$. Therefore, $cost_1(P') = \frac{1}{2} + 2 = \frac{5}{2}$ and $cost_2(P') = \frac{1}{2} + 1 = \frac{3}{2}$.

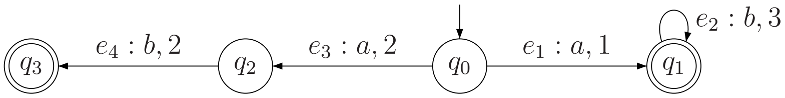

Figure 1: An example of a WFA.

We consider the following instances of AF games. Let $G = \langle \mathcal{A}, O\rangle$. We start with instances obtained by imposing restrictions on the WFA $\mathcal{A}$. In *one-letter* instances, $\mathcal{A}$ is over a singleton alphabet, i.e., $|\Sigma| = 1$. When depicting such WFAs, we omit the letters on the transitions. In *all-accepting* instances, all the states in $\mathcal{A}$ are accepting; i.e., $F = Q$. In *uniform-costs* instances, all the transitions in the WFA have the same cost, which we normalize to 1. Formally, for every $e \in \Delta$, we have $c(e) = 1$. We continue to restrictions on the objectives in O. In *single-word* instances, each of the languages in O consists of a single word. In *symmetric* instances, the languages in O coincide, thus the players all have the same objective. We also consider combinations on the restrictions. In particular, we say that $\langle \mathcal{A}, O\rangle$ is *weak* if it is one-letter, all states are accepting, costs are uniform, and objectives are single words. Weak instances are simple indeed – each player only specifies a length of a path he should patrol, ending anywhere in the WFA, where the cost of all transitions is the same. As we shall see, many of our hardness results and lower bounds hold already for the class of weak instances.

Remark 2.1. *Recall that in resource allocation games (RAGs, for short) there is a set* *of resources, and the strategies of each player consist of subsets of resources. RAGs generalize NFGs, and there are RAGs with no equivalent NFG. Multiset RAGs generalize* AF *games, and we show that unlike NFGs, every multiset RAG has an equivalent* AF *game. Consider a multiset RAG with resources R and let S_i be the set of strategies for Player i, thus each $s \in S_i$ is a multiset over R. We construct a simple* AF *game with alphabet R. There is a single state in the WFA with $|R|$ self loops. Each loop is labeled by a resource (letter) in R and its cost coincides with the cost of the resource in the RAG. Finally, we view each strategy in S_i as a word over R, and set Player i's language to be S_i. Clearly, the strategies in both games coincide.*

2.2. Nash equilibrium, social optimum, and equilibrium inefficiency

For a profile P, a strategy π_i for Player i, and a strategy π, let $P[\pi_i \leftarrow \pi]$ denote the profile obtained from P by replacing the strategy for Player i by π. A profile $P \in \mathcal{S}$ is a *pure Nash equilibrium* (NE) if no player i can benefit from unilaterally deviating from his run in P to another run; i.e., for every player i and every run $\pi \in \mathcal{S}_i$ it holds that $cost_i(P[\pi_i \leftarrow \pi]) \geq cost_i(P)$. In our example, the profile P is not a NE, since Player 2 can reduce his payments by deviating to profile P'.

Consider a profile P. A *best response* strategy for Player i is the most beneficial deviation Player i can perform (if one exists), thus it is the strategy that minimizes $\min_{\pi_i \in S_i} cost_i(P[i \leftarrow \pi_i])$.

The (social) cost of a profile P, denoted $cost(P)$, is the sum of costs of the players in P. Thus, $cost(P) = \sum_{1 \leq i \leq k} cost_i(P)$. Equivalently, if we view P as a set of transitions, with $e \in P$ iff there is $\pi \in P$ for which $e \in \pi$, then $cost(P) = \sum_{e \in P} c(e)$. Note that the latter definition implies that $cost(P)$ can take only finitely many values. We refer to the *social optimum* (SO, for short) as the cheapest profile, and denote its cost by OPT, thus $OPT = \min_{P \in \mathcal{S}} cost(P)$. 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 quantify the inefficiency incurred due to self-interested behavior according to the *price of anarchy* (PoA) [30, 38] and *price of stability* (PoS) [2] measures. The PoA is the worst-case inefficiency of a Nash equilibrium, while the PoS measures the best-case inefficiency of a Nash equilibrium. Formally,

Definition 2.1. *Let $\mathcal{G}$ be a family of games, and let $G \in \mathcal{G}$ be a game in $\mathcal{G}$. Let $\Upsilon(G)$ be the set of Nash equilibria of the game G. Assume that $\Upsilon(G) \neq \emptyset$.*

- *The* price of anarchy *of G is the ratio between the* maximal *cost of a NE and the social optimum of G. That is, $PoA(G) = \max_{P \in \Upsilon(G)} cost(P)/OPT(G)$. The* price of anarchy *of the family of games $\mathcal{G}$ is $PoA(\mathcal{G}) = sup_{G \in \mathcal{G}} PoA(G)$.*
- *The* price of stability *of G is the ratio between the* minimal *cost of a NE and the social optimum of G. That is, $PoS(G) = \min_{P \in \Upsilon(G)} cost(P)/OPT(G)$. The* price of stability *of the family of games $\mathcal{G}$ is $PoS(\mathcal{G}) = sup_{G \in \mathcal{G}} PoS(G)$.*

We use PoA and PoS without specifying the the game or the family of games, when they are clear from the context.

Uniform sharing rule: A different cost-sharing rule that could be adopted for automaton-formation games is the uniform sharing rule, according to which the cost of a transition e is equally shared by the players that traverse e, independent of the number of times e is traversed by each player. Formally, let $\kappa_P(e)$ be the number of runs that use the transition e at least once in a profile P. Then, the cost of including a transition e at least once in a run is $c(e)/\kappa_P(e)$. This sharing rule induces a potential game, where the potential function is identical to the one used in the analysis of the classical network design game [2]. Specifically, let $\Phi(P) = \sum_{e \in E} c(e) \cdot H(\kappa_P(e))$, where $H(0) = 0$, and $H(k) = 1 + 1/2 + \ldots + 1/k$. Then, $\Phi(P)$ is a potential function whose value reduces with every improving step of a player, thus a pure NE exists and BRD is guaranteed to converge. The similarity with classical network-formation games makes the study of this setting straightforward. Thus, throughout this paper we only consider the proportional sharing rule as defined in (1) above.

3. Properties of Automaton-Formation Games

In this section we study the theoretical properties of AF games: existence of pure NE and equilibrium inefficiency. We show that AF games need not have a pure Nash equilibrium. This holds already in the very restricted class of weak instances, and is in contrast with network-formation games. There, BRD converges and a pure NE always

	π_1	π_1'
π_2	$4-\epsilon, \epsilon$	$3+\epsilon, 3-\epsilon$
π_2'	$4-\epsilon, 3+\epsilon$	$4.5-\epsilon, 1.5+\epsilon$

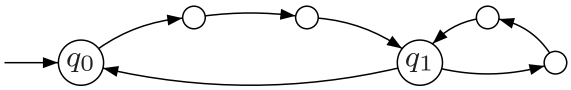

Figure 2: A weak instance of AF games with no NE. The table lists the costs in some of the profiles, where Player 1's cost is listed before Player 2's cost.

exists [2]. We then analyze the PoS in AF games and show that there too, the situation is significantly less stable than in network-formation games.

Theorem 3.1. *Automaton-formation games need not have a pure NE. This holds already for the class of weak instances.*

Proof: Consider the WFA $\mathcal{A}$ depicted in Figure 2 and consider a game with $k = 2$ players. The language of each player consists of a single word. Recall that in one-letter instances we care only about the lengths of the objective words. Let these be ℓ_1 and ℓ_2, with $\ell_1 \gg \ell_2 \gg 0$ that are multiples of 12. For example, $\ell_1 = 30000, \ell_2 = 300$. Let C_3 and C_4 denote the cycles of length 3 and 4 in $\mathcal{A}$, respectively. Let D_3 denote the path of length 3 from q_0 to q_1. Every run of $\mathcal{A}$ consists of some repetitions of these cycles possibly with one pass on D_3.

We claim that no pure NE exists in this instance. We start by considering profiles in which the players either select a run that only traverses C_4, or select a run that traverses D_3 once and then stays in C_3. Since we consider long runs, the fact that the last cycle might be partial is ignored in the calculations below. Let $\pi_1 = (C_4)^{\frac{\ell_1}{4}}$ and $\pi_1' = D_3 \cdot (C_3)^{\frac{\ell_1}{3}-3}$ be the two runs for Player 1, and $\pi_2 = (C_4)^{\frac{\ell_2}{4}}$ and $\pi_2' = D_3 \cdot (C_3)^{\frac{\ell_2}{3}-3}$ be the two runs for Player 2. The costs of the players in the four profiles using these runs is listed in the table in Figure 2. We write ϵ to indicate a small cost, but these small costs are not the same in the different profiles.

None of these profiles is a NE as a sequence of best-response moves results in a clockwise cycle. We describe the intuition of this cycle as its idea will be helpful later

[2] Best-response-dynamics (BRD) is a local-search method where in each step some player is chosen and plays his best-response strategy, given that the strategies of the other players do not change.

on. One might expect the profile $\langle \pi_1, \pi_2 \rangle$, which is the social optimum, to be a NE. However, it is not a NE as Player 1 can benefit from deviating to π'_1. Indeed, in the social optimum, Player 1 pays almost all of the cost of the cycle C_4, while in $\langle \pi'_1, \pi_2 \rangle$, he pays the full cost of C_3 while Player 2 pays most of the cost of the path D_3, leaving a cost of ϵ for Player 1 for this path. Thus, the deviation decreases his cost from $4 - \epsilon$ to $3 + \epsilon$ (for different ϵ's). The profile $\langle \pi'_1, \pi_2 \rangle$ is not a NE as Player 2 benefits from "joining" Player 1 by using D_3 once and completing the run in C_3, where most of the cost is paid by Player 1. This deviation decreases his cost from $4 - \epsilon$ to $1.5 + \epsilon$. Now, Player 1 pays $4.5+\epsilon$, which is more than buying C_4 by himself, and he can benefit from deviating "back" to π_1. Again, Player 2 benefits from joining Player 1. This concludes the best-response cycle.

To conclude the proof, we show that none of the other profiles are a NE. Consider a profile P. We show that P is not a NE. Note that if $cost_1(P) > 4$, Player 1 can deviate to π_1 and pay at most 4. We distinguish between several cases. First, assume that one of the players plays one of his *pure* strategies, i.e., one of π_1, π'_1, π_2, or π'_2, and the other plays a mix between C_4 and C_4. Note that the case in which both players play a pure strategy is taken care of above. If Player 1 plays π_1 and Player 2 plays a mixed strategy, then Player 2 pays the full cost of C_3, and can benefit from joining Player 1 in C_4 and playing π_2. Assume Player 1 plays π'_1 and that he pays no more than 4. Since $\ell_1 \gg \ell_2$, Player 1 pays most of the cost of C_3, thus Player 2 pays most of the cost of C_4, which is $4 - \epsilon$. Thus, he can benefit from deviating to π'_2 and joining Player 1, thereby reducing the cost to $1.5 + \epsilon$. Next, assume Player 1 plays a mixed strategy. The case in which Player 2 plays π_2 is similar to the previous one. If Player 2 plays π'_2 and Player 1 pays less than 4, Player 2 pays most of the cost of C_3, and can benefit from deviating to π_2. Finally, assume both players play a mixed strategy and Player 1 pays less than 4. Since $\ell_1 \gg \ell_2$, Player 2 pays for most of one of the cycles, thus he either pays $4 - \epsilon$ or $3 - \epsilon$. In the first case he can benefit from deviating to π'_2 and in the second to π_2. We conclude that P is not a NE, and we are done. □

The fact a pure NE may not exist is a significant difference between standard cost-sharing games and AF games. The bad news do not end here and extend to equilibrium inefficiency. We first note that the cost of any NE is at most k times the social optimum (as otherwise, some player pays more than the cost of the SO and can benefit from migrating to his strategy in the SO). Thus, it holds that $PoS \leq PoA \leq k$. The following theorem shows that this is tight already for highly restricted instances.

Theorem 3.2. *The PoS in* AF *games equals the number of players. This holds already for the class of weak instances.*

Proof: We show that for every $k, \delta > 0$ there exists a simple game with k players for which the PoS is more than $k - \delta$. Given k and δ, let r be an integer such that $r > \max\{k, \frac{k-1}{\delta} - 1\}$. We assume $k \geq 2$ as the claim is trivial for $k = 1$. Consider the WFA $\mathcal{A}$ depicted in Figure 3. Let $L = \langle \ell_1, \ell_2, \ldots, \ell_k \rangle$ for $\ell_2 = \ldots = \ell_k$ and $\ell_1 \gg \ell_2 \gg 0$ denote the lengths of the objective words. Thus, Player 1 has an "extra-long word" and the other $k-1$ players have words of the same, long, length. Let C_r and C_{r+1} denote, respectively, the cycles of length r and $r+1$ to the right of q_0. Let D_r denote the path of length r from q_0 to q_1, and let D_{kr} denote the "lasso" consisting of the kr-path and the single-edge loop to the left of q_0.

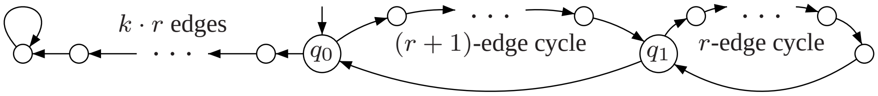

Figure 3: A weak instance of AF games for which $PoS = k$.

The social optimum of this game is to buy C_{r+1}. Its cost is $r+1$ (recall that the cost of a profile is the sum of costs of the transitions it uses). However, as we show, the profile P in which all players use D_{kr} is the only NE in this game. We first show that P is a NE. In this profile, the players split evenly the cost of the first rk edges in the path, while Player 1 pays most of the cost of the self loop. Thus, Player 1 pays $r + (1 - \varepsilon)$ and each other player pays $r + \varepsilon/(k-1)$. No player will deviate to a run that includes edges from the right side of $\mathcal{A}$. Indeed, such a deviation requires using at least $r+1$ edges in the right side of $\mathcal{A}$, and these edges are not shared, thus the cost is at least $r+1$, which is clearly not beneficial.

Next, we show that P is the only NE of this game. Thus, we show that there is no NE that uses edges in the right side of $\mathcal{A}$. Showing that a profile in which all the players select a run in the right side of $\mathcal{A}$ is done in a similar manner to Theorem 3.1. Using similar arguments, we can show that it suffices to focus on *pure* strategies, namely ones of the form C^*_{r+1} or $D_r \cdot C^*_r$. Next, a profile with pure strategies is not an NE for similar reasons to these in Theorem 3.1. The social optimum is not a NE as Player 1 would deviate to $D_r \cdot C^*_r$ and will reduce his cost to $r + \varepsilon'$. The other players, in turn, will also deviate to $D_r \cdot C^*_r$. In the profile in which they are all selecting a run of the form $D_r \cdot C^*_r$, Player 1 pays $r + r/k - \varepsilon > r + 1$ and prefers to return to C^*_{r+1}. The other players will join him sequentially, until the non-stable social optimum is reached. Finally, a profile that uses edges from both the right and left parts of $\mathcal{A}$ cannot be a NE, and in any case, its cost is higher than the profile in which all players proceed left.

The cost of the NE profile is $kr + 1$ and the PoS is therefore $\frac{kr+1}{r+1} = k - \frac{k-1}{r+1} > k - \delta$. □

4. Computational Complexity Issues in AF Games

In this section we study the computational complexity of three problems: finding the cost of the social optimum, finding the best-response of a player, and deciding the existence of a NE. Recall that the social optimum (SO) is a profile that minimizes the total cost the players pay. It is well-known that finding the social optimum in a network-formation game is NP-complete. We show that this hardness is carried over to simple instances of AF games. On the positive side, we identify non-trivial classes of instances, for which it is possible to compute the SO efficiently. The other issue we consider is the complexity of finding the best strategy of a single player, given the current profile, namely, the best-response of a player. In network-formation games, computing the best-response reduces to a shortest-path problem, which can be solved efficiently. We show that in AF games, the problem is NP-complete. Finally, recall that AF games are not guaranteed to have a NE. We study the problem of deciding, given an AF game, whether it has a NE. We term this problem $\exists$NE. We show that the $\exists$NE problem is Σ^2_P-complete.

We start with the problem of finding the value of the social optimum. Similar hardness results are known for traditional network formation games [37].

Theorem 4.1. *Given an* AF *game $\mathcal{G}$ and a value c, deciding whether the value of the social optimum in $\mathcal{G}$ is at most c, is NP-complete. Moreover, it is NP-complete already in single-worded instances that are also uniform-cost and are either single-lettered or all-accepting.*

Proof: We start with membership in NP. Consider a WFA $\mathcal{A}$ with objectives $\mathcal{D}_1, \dots, \mathcal{D}_k$, where $\mathcal{D}_i$ is a deterministic finite automaton that recognizes the regular language of Player i, for $1 \leq i \leq k$, and a value $c \in \mathbb{R}$. We claim that there is a social optimum profile in which, for $1 \leq i \leq k$, the length of the i-th word does not exceed $|\mathcal{A}| \cdot |\mathcal{D}_i|$. Thus, we guess such a witness profile P and check whether it satisfies $cost(P) \leq c$ in polynomial time. Assume towards contradiction that there is a social optimum profile P in which the words the players choose are the shortest, and in which Player i selects a word w_i of length greater than $|\mathcal{A}| \cdot |\mathcal{D}_i|$. We claim that there is a word $w' \in L(\mathcal{D}_i)$ of length shorter than w such that $cost(P[i \leftarrow w']) \leq cost(P)$, which contradicts our assumption. Indeed, let $r_1 = q_0, q_1, \dots, q_{|w|}$ and $r_2 = p_0, p_1, \dots, p_{|w|}$ be the runs of $\mathcal{A}$ on w and $\mathcal{D}_i$ on w, respectively. Then, using a pigeonhole-principle argument we can show that there are indices $0 \leq j_1 < j_2 \leq |w|$ such that there are loops in r_1 and r_2 between j_1 and j_2, thus $q_{j_1} = q_{j_2}$ and $p_{j_1} = p_{j_2}$. We construct $w' \in L(\mathcal{D}_i)$ by removing the sub-word that corresponds to this cycle. Shortening the word can only decrease the cost of P, thus we are done.

For proving hardness, we show a reduction from the *Set-Cover* (SC) problem. Consider an input $\langle U, S, m\rangle$ to SC. Recall that U is a set of elements, $S = \{C_1, \dots, C_z\} \subseteq 2^U$ is a collection of subsets of elements of U, and $m \in \mathbb{N}$. Then, $\langle U, S, m\rangle$ is in SC iff there is a subset S' of S of size at most m that covers U. That is, $|S'| \leq m$ and $\bigcup_{C \in S'} C = U$.

Given an input $\langle U, S, m\rangle$ to SC, we construct a uniform-cost single-letter WFA $\mathcal{A}$ and a vector of k integers, where the i-th integer corresponds to the length of the (single) word in L_i. We construct $\mathcal{A} = \langle \{a\}, Q, q_0, \Delta, \{q_{acc}\}, c\rangle$ as follows (see an example in the left of Figure 4). The set Q includes the initial and accepting states, a state for every set in S, and intermediate states required for the disjoint runs defined below. Without loss of generality, we assume that $U = \{1, \dots, k\}$. Consider an element $i \in U$. For every $C \in S$ such that $i \in C$, there is a disjoint run of length i from C to q_{acc}. Also, for every $C \in S$, there is a transition $\langle q_0, C\rangle$ in Δ. The cost of all transitions in Δ is 1. For every $1 \leq i \leq k$, the length of the word in L_i is $i+1$. The size of $\mathcal{A}$ is clearly polynomial in $|U|$ and $|S|$.

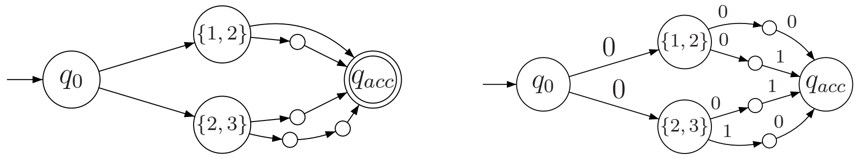

Figure 4: The WFAs produced by the reduction for $U = \{1,2,3\}$ and $S = \{\{1,2\},\{2,3\}\}$.

The construction for uniform-cost all-accepting instances is very similar (see an example in the right of Figure 4). Let $z = \lceil \log(k) \rceil$ and $\Sigma = \{0,1\}$. For $C \in S$ and $i \in C$, we have a z-length path from C to q_{acc} that is labeled with the binary representation of $i-1$ (padded with preceding zeros if needed). The label on all transitions from q_0 to the S states is 0. For $1 \leq i \leq k$, the word for Player i is a single 0 letter followed by the binary representation of $i-1$. The size of $\mathcal{A}$ is clearly polynomial in $|U|$ and $|S|$.

We claim that there exists a set-cover of size m iff $OPT \leq m+(1+2+\ldots+k)$ for the uniform-cost single-letter instance and $OPT \leq m + k \cdot z$ for the uniform-cost all-accepting instance. We prove the claim for the uniform-cost single-letter instance. The proof for the uniform-cost all-accepting instance is very similar. For the first direction, let $S' = \{s_{i_1}, \ldots, s_{i_m}\}$ be a set cover. We show a profile $P = \{\pi_1, \ldots, \pi_k\}$ such that $cost(P) = m + (1+2+\ldots+k)$. Recall that the input length for Player i is $i+1$. Since S' is a set cover, there is a set $s \in S'$ with $i \in s$. We define the run π_i to proceed from q_0 to s and from there to q_{acc} on a run of length i. Clearly, the runs $\pi_1, \ldots, \pi_k$ are all legal-accepting runs. Moreover, the runs use m transitions from $\{q_0\} \times S \subseteq E$. Thus, $cost(P) = m + (1+2+\ldots+k)$, implying $OPT \leq m + (1+2+\ldots+k)$, and we are done.

For the second direction, assume $OPT = m' + (1+2+\ldots+k) \leq m + (1+2+\ldots+k)$, we prove that there is a set cover of size m'. Let $P^* = \langle \pi_1, \ldots, \pi_k\rangle$. Thus, $OPT = cost(P^*)$. Let $S' \subset S$ be such that $s \in S'$ iff the transition $\langle q_0, s\rangle$ is used in one of the runs in P^*. Note that the run of every player consists of a transition (q_0, s) followed by a disjoint run of length i to q_{acc}. Therefore, $OPT = m' + (1+2+\ldots+k)$, and, $|S'| = m' \leq m$. We claim that S' is a set cover. For every $i \in U$, the first transition in the run is a transition $\langle q_0, s\rangle$ for some $s \in S$, as otherwise, Player i can not proceed to q_{acc} along a run of length i. By our definition of S' we have $s \in S'$, thus $i \in U$ is covered. □

The hardness results in Theorem 4.1 for single-word specification use one of two properties: either there is more than one letter, or not all states are accepting. We show that finding the SO in instances that have both properties can be done efficiently, even for specifications with arbitrary number of words.

For a language L_i over $\Sigma = \{a\}$, let $short(i) = \min_j \{a^j \in L_i\}$ denote the length of the shortest word in L_i. For a set O of languages over $\Sigma = \{a\}$, let $\ell_{max}(O) = \max_i short(i)$ denote the length of the longest shortest word in O. Clearly, any solution, in particular the social optimum, must include a run of length $\ell_{max}(O)$. Thus the cost of the social optimum is at least the cost of the cheapest run of length $\ell_{max}(O)$. Moreover, since the WFA is single-letter and all-accepting, the other players can choose runs that are prefixes of this cheapest run, and no additional transitions should be acquired. We show that finding the cheapest such run can be done efficiently.

Theorem 4.2. *The cost of the social optimum in a single-letter all-accepting instance $\langle \mathcal{A}, O\rangle$ is the cost of the cheapest run of length $\ell_{max}(O)$. Moreover, this cost can be found in polynomial time.*

Proof: Clearly, any solution, in particular the social optimum, must include a run of length $\ell_{max}(O)$. Thus the cost of the social optimum is at least the cost of the cheapest run of length $\ell_{max}(O)$. Moreover, since there are no target vertices, the other players can be assigned runs that are prefixes of the cheapest run, and no additional transitions should be acquired.

We claim that finding the cheapest such run can be done efficiently. Recall that q_0 is the initial state in $\mathcal{A}$, and let $|Q| = n$. We view $\mathcal{A}$ as a weighted-directed graph $G = \langle V, E, c\rangle$, where the vertices V are the states Q, there is an edge $e \in E$ between two vertices if there is a transition between the two corresponding states, and the cost of the edges is the same as the cost of the transition in $\mathcal{A}$. For $0 \leq i \leq n$, let $d_i : V \times V \to \mathbb{R}^{\geq 0}$ be the function that, given two vertices $u, v \in V$, returns the value of the cheapest path of length i from u to v, and ∞ if no such path exists. Note that there is no requirement that the path is simple, and indeed we may traverse cycles in order to accommodate i transitions. The function $d : V \times V \to \mathbb{R}^{\geq 0}$, returns the value of the cheapest path of any length between two given vertices, where, for $v \in V$, we require that $d(v, v)$ is the value of a non-trivial cycle. Given two vertices $u, v \in V$, computing $d(u, v)$ can be done using Dijkstra's algorithm, and, given an index $i \in \mathbb{N}$, it is possible to compute $d_i(u, v)$ by a slight variation of the Bellman-Ford algorithm.

We distinguish between two cases. If $\ell_{max} > 2n - 2$, we claim that the value of the social optimum is $min\{d(q_0, v) + d(v, v) : v \in V\}$. If $\ell_{max} \leq 2n - 2$, then we claim that the value of the social optimum is the minimum value of $d_i(q_0, v) + d_j(v, v)$, where $v \in V$, $0 \leq i \leq \ell_{max}$, $0 \leq j \leq \ell_{max} - i$, and if $j = 0$, then $i = \ell_{max}$.

We start with the first case. Assume $\ell_{max} > 2n - 2$. Let $ALG = \min\{d(q_0, v) + d(v, v) : v \in V\}$. Let P^* be the social optimum profile, thus $OPT = cost(P^*)$. For the first direction, we claim that $ALG \leq OPT$. Let π be a run in P^* of length ℓ_{max}, where we assume π is a sequence of transitions. We use $cost(\pi)$ to denote the sum of the costs of the transitions that π traverses. Clearly, $OPT \geq cost(\pi)$. Since ALG takes the minimum over all vertices, it suffices to prove that $cost(\pi) \geq d(q_0, v) + d(v, v)$ for some $v \in V$. Since $|\pi| > 2n - 2$, there is a vertex v that reoccurs in π. Let v be the first reoccuring vertex, and let $\tau_1 \cdot \tau_2$ be a subpath of π such that τ_1 starts in q_0 and ends in the first occurrence of v, and τ_2 is ends at the second occurrence of v. Since the cost of the edges are non-negative, we have $cost(\tau_1 \cdot \tau_2) \leq cost(\pi)$. On the other hand, we have $d(q_0, v) \leq cost(\tau_1)$ and $d(v, v) \leq cost(\tau_2)$, thus $ALG \leq cost(\tau_1 \cdot \tau_2)$, and we are done.

We continue to prove that $ALG \geq OPT$. Let $v \in V$ be the vertex that attains the minimum in $\min\{d(q_0, v) + d(v, v) : v \in V\}$. Let $\tau = \tau_1 \cdot \tau_2$ be a run such that τ_1 is a simple path from q_0 to v with $cost(\tau_1) = d(q_0, v)$ and τ_2 is a simple cycle from v to itself with $cost(\tau_2) = d(v, v)$. We claim that $cost(\tau) \geq OPT$. Since τ_1 and τ_2 are simple, we have $|\tau_1| \leq n - 1$ and $|\tau_2| \leq n - 1$. Thus, $|\tau| \leq 2n - 2$. We extend τ to a path π of length ℓ_{max} by traversing the cycle τ_2 sufficiently many times. Clearly, π

is a legal run of the automaton $\mathcal{A}$ on a word of length ℓ_{max} as all states are accepting. Consider the profile P in which the players choose runs that are prefixes of π. Since the only transitions used in P are those in τ, we have $cost(P) = cost(\tau)$. Clearly $cost(P) \geq OPT$, thus $ALG \geq OPT$, and we are done.

The case in which $\ell_{max} \leq 2n - 2$ is proven in a similar manner. □

We turn to prove the hardness of finding the best-response of a player. Our proof is valid already for a single player that needs to select a strategy on a WFA that is not used by other players (one-player game).

Theorem 4.3. *Finding the best-response of a player in* AF *games is NP-complete. Formally, given a* AF *game $\mathcal{G}$, a profile P in $\mathcal{G}$, an index i, and a value c, deciding whether Player i has a strategy π_i such that $cost_i(P[i \leftarrow \pi_i]) \leq c$, is NP-complete.*

Proof: We start with membership in NP. Given a WFA $\mathcal{A}$, runs $r_1, \ldots, r_{k-1}$ of $\mathcal{A}$, an objective DFA $\mathcal{D}$ for Player k, and value $c \in \mathbb{R}$, we check whether there is a word $w \in L(\mathcal{D})$ and a run r_k of $\mathcal{A}$ on w such that $cost_k(\langle r_1, \ldots, r_k \rangle) \leq c$. The same argument as in Theorem 4.1 shows that we can bound the length of w by $|\mathcal{D}_i| \cdot |\mathcal{A}|$. Thus, we can guess such a word w and a run of $\mathcal{A}$ on w, and check whether the corresponding profile P has $cost_k(P) \leq c$ in polynomial time.

For proving hardness, we show a reduction from the *Set-Cover* (SC) problem. Consider an input $\langle U, S, m \rangle$ to SC. Recall that $U = \{1, \ldots, n\}$ is a set of elements, $S = \{C_1, \ldots, C_z\} \subseteq 2^U$ is a collection of subsets of elements of U, and $m \in \mathbb{N}$. Then, $\langle U, S, m \rangle$ is in SC iff there is a subset S' of S of size at most m that covers U. That is, $|S'| \leq m$ and $\bigcup_{C \in S'} C = U$.

Given an input $\langle U, S, m \rangle$ to SC, we construct a one-player game $\langle \mathcal{A}, O \rangle$ such that $\langle U, S, m \rangle$ is in SC iff the cost of the SO in the game is at most m, where the SO coincides with the best response in such a game. We start by describing the specification L of the player. The alphabet of L is $S \cup U$ and it is given by the regular expression $(C_1 + \ldots + C_m) \cdot 1 \cdot (C_1 + \ldots + C_m) \cdot 2 \cdot \ldots \cdot (C_1 + \ldots + C_m) \cdot n$. The WFA $\mathcal{A}$ is over the alphabet $S \cup U$. There is a single initial state q_{in} and a state for every set in S. For $1 \leq i \leq z$, there is a C_i-labeled transition from q_{in} to the state C_i, and for every $j \in C_i$, there is a j-labeled transition from the state C_i back to q_{in}. The first type of transitions cost 1 and the second cost 0 (for an example see Figure 5).

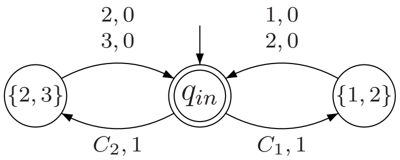

Figure 5: The WFA produced by the reduction for $U = \{1, 2, 3\}$ and $S = \{\{1, 2\}, \{2, 3\}\}$.

We prove the correctness of the reduction: For the first direction, assume there is a set cover of at most m. Consider the word w in which, for every $1 \leq j \leq n$, the letter that precedes j is $C_i \in S$ such that $j \in C_i$ and C_i is in the set cover. Clearly, $w \in L$ and since it uses at most m letters from S, the profile in which the player chooses it, costs at most m. Thus, the cost of the SO is also at most m. For the other direction, assume the SO is attained in a profile with the word $w \in L$. It is not hard to see that the letters from S that appear in w form a set cover of size at most m. □

We turn to study the problem of deciding whether a NE exists. We show that ∃NE is complete for Σ_2^P; the second level of the polynomial hierarchy. Namely, decision problems solvable in polynomial time by a nondeterministic Turing machine augmented by an *oracle* for an NP-complete problem. An oracle for a computational problem is a black box that is able to produce a solution for any instance of the problem in a single operation. Thus, for every problem $P \in \Sigma_2^P$ there is a machine such that for every $x \in P$ there is a polynomial accepting computation (with polynomial many queries to the oracle). As co-NP is the dual complexity class of NP, the dual complexity class of Σ_2^P is Π_2^P. Thus, a problem P is Σ_2^P-complete iff its complement $\overline{P}$ is Π_2^P-complete.

Theorem 4.4. *The ∃NE problem is Σ_2^P-complete.*

Proof: The upper bound is similar to the two previous theorems. We guess a profile in which the length of the i-th word is at most $|\mathcal{D}_i| \cdot |\mathcal{A}|$, where for $1 \leq i \leq k$, the DFA $\mathcal{D}_i$ recognizes the language of Player i. We use k calls to an oracle for the best-response problem to verify that no player can benefit from deviating.

For the lower bound, we alter the reduction in [6], for the ∃NE problem in a similar game. The reduction is from the complement of the *min-max vertex cover* problem (MMVC, for short), which is known to Σ_2^P-complete [29]. The input to the MMVC problem is $\langle G, I, J, N, c\rangle$, where $G = \langle V, E\rangle$ is an undirected graph, I and J are sets of indices, $N : V \to \{V_{i,j} \subseteq V : i \in I \text{ and } j \in J\}$ partitions the vertices, and $c \in \mathbb{N}$ is a value. Note that since G is undirected, its edges are sets with two vertices. We refer to the sets in the partition $\{V_{i,j}\}_{i\in I, j\in J}$ as *neighborhoods* and for $v \in V$ we refer to $N(v)$ as the neighborhood of v. Note that there is a coarser partition of V, namely $\{V_i\}_{i\in I}$, where $V_i = \bigcup_{j\in J} V_{i,j}$. We refer to the elements in this partition as *districts*. For a function $t : I \to J$ we define $V_t = \bigcup_{i\in I} V_{i,t(i)}$. Intuitively, t is a choice of neighborhood in each district. Let $G_t = \langle V_t, E_t\rangle$ be the induced subgraph of G on the vertex set V_t. Formally, for $e \in E$, we have $e \in E_t$ iff $e \subseteq V_t$. For a graph G, we say that $S \subseteq V$ is a vertex cover of G if for every $e = \{u, v\} \in E$ we have $S \cap \{u, v\} \neq \emptyset$. An input $\langle G, I, J, N, c\rangle$ is in MMVC iff for any choice of neighborhoods in the districts given by a function t, the smallest vertex cover of the resulting graph is at most c. Formally, $\max_{t\in J^I} \min\{|S| : S \subseteq V_t \text{ is a vertex cover of } G_t\} \leq c$. We assume without loss of generality that $c \leq |V|$.

Given an instance $\langle G, I, J, N, c\rangle$, where $G = \langle V, E\rangle$, we construct an AF game that has a NE iff $\langle G, I, J, N, c\rangle \notin$ MMVC. For ease of presentation, we assume that the costs of the automaton are given on the vertices rather than on the transitions. It is easy to translate an instance of such a model to a standard AF game by splitting vertices and adding letters to the words. We start by describing the automaton $\mathcal{A}$ (see Figure 6 for an illustration). Its alphabet is $\Sigma = V \cup E \cup I \cup \{\#\}$. The states of $\mathcal{A}$ consist of an initial state q_0, *neighborhood states* $V_{i,j}$, for $i \in I$ and $j \in J$, *vertex states* $v \in V$, and *index states* $i \in I$, as well as other states that are depicted in the illustration, and we elaborate on them below. For $i \in I$ and $j \in J$, the neighborhood state $V_{i,j}$ has an i-labeled self loop, and an i-labeled transition leading to the state i. For every $e \in E$ such that $e \cap (V_i \setminus V_{i,j}) \neq \emptyset$, there is a cycle labeled with the word $e\#$ around $V_{i,j}$. Finally, for every $e \in E$ that has $v \in e$ with $v \in V_{i,j}$, there is an e-labeled transition leading to the vertex state v. For each vertex state as well as for the initial state, there are $|I| \cdot |J|$ outgoing transitions labeled with the letter $\#$ to all neighborhood states. The cost of every vertex state is 1, the cost of every neighborhood state is $2(c + 1) + 1$, the cost of an index state is strictly less than $(c+2)/2$, using the left path in $\mathcal{A}$ costs $c+1$, and the other states cost 0.

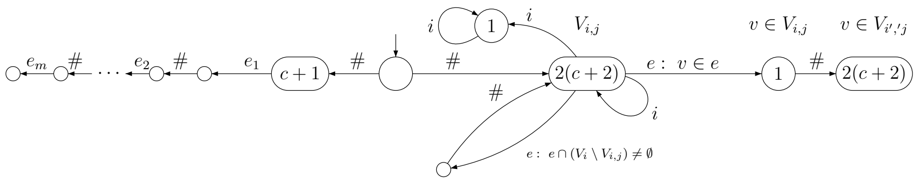

Figure 6: An illustration of the automaton that is constructed in the reduction from MMVC. The cost of a non-free state is stated in it, and the type of the state is given above it.

There are $|I|+1$ players in the AF game. The players' languages consist of a single word. The first type of players are *index players* and there are $|I|$ such players. For $i \in I$, the word of Player i is $\# \cdot i^\ell$, for a sufficiently large ℓ, which we choose later on. So, a run for Player i starts from the initial state, continues to a neighborhood state $V_{i,j}$, and either stays there, or proceeds at some point to the index state i. Note that the index players do not share vertices between themselves. We assume there is an arbitrary order on the edges in E. The word of Player 0 is $\#e_1\#e_2\#\ldots\#e_m$. The left part of $\mathcal{A}$ recognizes this word. When choosing the run that proceeds left, he does not share any of the states, and pays $c+1$. By proceeding right, he can hope to share the neighborhood states with the other players but buys the vertex states by himself. Clearly the reduction is polynomial.

We proceed to prove the correctness of the reduction. Assume first that $\langle G, N, I, J, c\rangle \notin$ MMVC, and we show a NE in the AF game. Let $t : I \to J$ be the function for which G_t has no VC of size at most c. Consider the following profile P. For $i \in I$, Player i chooses the run that proceeds from the initial state to the neighborhood state $V_{i,t(i)}$, and stays there. Player 0 proceeds left. We claim that P is a NE. Since the index players do not share states, a deviation for Player i, for each $i \in I$, means that rather than paying only for $V_{i,t(i)}$ in P, he also buys the index state i, which is clearly not beneficial. We claim that Player 0 cannot benefit from unilaterally deviating. Assume towards contradiction that he can, and let r be the beneficial deviation. Let $S \subseteq V$ be the vertex states that r traverses. We claim that $S \subseteq V_t$, and that it is a VC of G_t of size at most c, contradicting our assumption that G_t has no VC of size at most c.

Since $cost_0(P) = c + 1$, and a neighborhood states costs more than $c + 1$, the run r traverses only neighborhood states that are used by an index player. Since $\ell \gg 0$, the index player pays most of the cost for the neighborhood state, leaving an $\epsilon > 0$ to be paid by Player 0. We choose ℓ such that the total cost Player 0 pays for neighborhood states, assuming the index players play the strategies described above, is less than 1. Since $cost_0(P[0 \leftarrow r]) < c + 1$, index players do not use S, and r traverses at least one neighborhood state at a positive cost, we have $|S| \leq c$. Consider an edge $e \in E_t$ and let $V_{i,t(i)}$ be the neighborhood state that r visits before e is read. There must be an e-labeled outgoing transition from $V_{i,t(i)}$ otherwise the run is stuck. Since $e \in E_t$ it cannot be part of the two-letter cycle, thus it leads to a vertex state v having $v \in e$. By our definition of the transitions, we have $v \in V_{i,t(i)}$, thus $S \subseteq V_t$. By our choice of S, we have $v \in S$. Thus, S is a VC of G_t, and we are done.

Assume that $\langle G, N, I, J, c\rangle \in$ MMVC, and we show that the AF has no NE. Assume towards contradiction that there is a NE P. Since proceeding left cost $c + 1$ for Player 0, we have $cost_0(P) \leq c+1$. Assume first that Player 0 proceeds right. Thus, he uses some neighborhood state $V_{i,j}$. Player i's best response is the run r_i that traverses $V_{i,j}$ once and ends in i. So, he splits evenly the cost of $V_{i,j}$ with Player 0, and the cost of the state i is less than half the cost of $V_{i,j}$. Every other strategy clearly costs more. Thus, Player i chooses r in P, but then, Player 0 pays more than $c + 1$ for $V_{i,j}$ and can benefit from deviating to the run that proceeds left, and we reach a contradiction.

Next, we assume Player 0 proceeds left. Thus, each Player i chooses a run that stays in some neighborhood state $V_{i,j}$. A function $t : I \to J$ clearly corresponds to the choices of the index players. Let $S \subseteq V_t$ be a VC of size at most c for G_t. We describe a run r that is a beneficial deviation for Player 0. Consider $e \in E$. If $e \in E_t$, then there is $v \in (S \cap e)$. Before reading e, the run r proceeds to the neighborhood state $N(v)$, which has an e-labeled outgoing transition to the state v. Note that $N(v) = V_{i,t(i)}$, for some $i \in I$, thus the cost Player 0 pays for $N(v)$ is ϵ as Player i pays most of its cost. If $e \notin E_t$, there is a vertex $v \in e$ such that $v \in V_i$, Player i uses $V_{i,t(i)}$, and $v \notin V_{i,t(i)}$. Thus, there is a $e\#$-labeled cycle originating from $V_{i,t(i)}$. Again, using $V_{i,t(i)}$ costs ϵ. The run r traverses at most c vertex states and neighborhood states that are used by index players. Thus, we have $cost_0(P[0 \leftarrow r]) < c + 1$, and we are done. □

5. Tractable Instances of AF Games

In the example in Theorem 3.1, Player 1 deviates from a run on the shortest (and cheapest) possible path to a run that uses a longer path. By doing so, most of the cost of the original path, which is a prefix of the new path and accounts to most of its cost, goes to Player 2. We consider *semi-weak* games in which the WFA is uniform-cost, all-accepting, and single-letter, but the objectives need not be a single word. We identify a property of such games that prevents this type of deviation and which guarantees that the social optimum a NE. Thus, we identify a family of AF games in which a NE exists, finding the SO is easy, and the PoS is 1.

Definition 5.1. *Consider a semi-weak game $\langle \mathcal{A}, O \rangle$. A* lasso *is a path $u \cdot v$, where u is a simple path that starts from the initial state and v is a simple cycle. A lasso ν is minimal in $\mathcal{A}$ if $\mathcal{A}$ does not have shorter lassos. Note that for minimal lassos $u \cdot v$, we have that $u \cap v = \emptyset$. We say that $\mathcal{A}$ is* resistant *if it has no cycles or there is a minimal lasso $\nu = u \cdot v$ such that for every other lasso ν' we have $|u \setminus \nu'| + |v| \leq |\nu' \setminus \nu|$.*

Consider a resistant semi-weak game $\langle \mathcal{A}, O \rangle$. In order to prove that the social optimum is a NE, we proceed as follows. Let ν be the lasso that is the witness for the resistance of $\mathcal{A}$. We show that the profile P^* in which all players choose runs that use only the lasso ν or a prefix of it, is a NE. The proof is technical and we go over all the possible types of deviations for a player and use the weak properties of the network along with its resistance. By Theorem 4.2, the cost of the profile is the cost of the SO. Hence the following.

Theorem 5.1. *For resistant semi-weak games, the social optimum is a NE.*

Proof: Consider a resistant semi-weak game $\langle \mathcal{A}, O \rangle$, thus $\mathcal{A}$ has no cycles or there is a minimal lasso in $\mathcal{A}$ that satisfies the resistance requirements. Recall that by Theorem 4.2, the social optimum is the profile P^* in which all players use prefixes of the cheapest run of length $\ell_{max}(O)$. Formally, let $\ell_1 \geq \ell_2 \geq \ldots \geq \ell_k$, where for $1 \leq i \leq k$, ℓ_i be the minimal length of a word in L_i. That is, $\ell_1 = \ell_{max}(O)$. Then, $P^* = \langle \pi_1, \ldots, \pi_k \rangle$, where for $1 \leq i \leq k$, the run π_i is of length ℓ_i and π_1 uses the lasso that is the witness for resistance, or an acyclic path if the lasso's length is larger than ℓ_1.

We claim that P^* is a NE. Assume otherwise, thus there are $1 \leq i \leq k$ and π'_i such that $cost_i(P^*) > cost_i(P^*[i \leftarrow \pi'_i])$. Assume Wlog that $|\pi'_i| = \ell_i$ as otherwise Player i can deviate to a prefix of length ℓ_i of π'_i and only improve his payment. We use P' to refer to $P^*[i \leftarrow \pi'_i]$. For $1 \leq j \leq k$, let ν_j be the set of transitions that are used in π_j. Similarly, let ν'_i be the transitions used in π'_i. Note that $\nu_1, \ldots, \nu_k, \nu'_i$ are paths of transitions.

We distinguish between four cases. In the first case, both ν_i and ν'_i are simple paths. First, note that every transition in $\nu_i \cap \nu'_i$ costs the same for Player i in both profiles. Next, we claim that every transition in $\nu'_i \setminus \nu_i$ costs at least as much as any transition in $\nu_i \setminus \nu'_i$. Indeed, every transition in $\nu'_i \setminus \nu_i$ is used only by Player i, thus its cost is 1. On the other hand, every transition in $\nu_i \setminus \nu'_i$ is shared with at least one player, thus its cost is at most 1. Since the cost Player i pays for a transition in $\nu_i \cap \nu'_i$ is the same in both profiles, we have $cost_i(P^*) \leq cost_i(P')$, and we reach a contradiction to the fact that Player i deviates.

In the second case, ν_i is simple and ν'_i is lasso. Thus, $|\nu'_i| \leq |\nu_i|$. If $|\nu'_i| = |\nu_i|$, we return to the previous case. We assume $|\nu'_i| < |\nu_i|$, and show that we reach a contradiction to our assumption that $\mathcal{A}$ is resistant. Recall that $|\nu_1| \geq |\nu_i|$. If π_1 uses a lasso, then ν'_i is a shorter lasso, contradicting the minimality of the witness lasso for resistance. If π_1 does not use a lasso, then we reach a contradiction to our assumption that the witness lasso has length greater than ℓ_1.

In the third case, ν_i is a lasso and ν'_i is simple. Thus, $\nu_i = \nu_1$. Consider a transition $e \in \nu_i$. Let a_e and a'_e be the number of times Player i uses e in π_i and π'_i, respectively. Thus, $a_e > 0$ and $a'_e \leq 1$. Let b_e be the number of times the other players use e in P^* and also in P' as none of them alter their strategy. Consider a transition $e \in \nu_i$ having $a'_e = 1$. That is, Player i reduces his number of uses of transition e from a_e to 1. Since the number of times Player i uses a transition in π'_i is at most 1, there are $(a_e - 1)$ transitions that are not used by Player i in π_i and are used once in π'_i. Since $\nu_i = \nu_1$, these transitions are all in $\nu'_i \setminus \nu_i$ and Player i pays 1 for each of them. Thus, there is a mapping between every such transition $e \in \nu_i$ to a set $S_e \subseteq \nu'_i$ of a_e transitions, where for every $e \neq e' \in \nu_i$, we have $S_{e'} \cap S_e = \emptyset$. We denote by $cost_i^e(P^*)$ the cost that Player i pays for e in P^*. Note that $cost_i^e(P^*) = \frac{a_e}{b_e + a_e}$. We denote by $cost_i^{S_e}(P')$ the cost that Player i pays for S_e in P'. Note that $cost_i^{S_e}(P') = \frac{1}{b_e+1} + (a_e - 1)$. A simple calculation shows that $cost_i^e(P^*) - cost_i^{S_e}(P') \geq 0$. Similarly, when $a'_e = 0$, we have $cost_i^{S_e}(P') = a_e$, and again $cost_i^e(P^*) - cost_i^{S_e}(P') \geq 0$. We sum up for all the transitions ν_i. Note that $\sum_{e \in \nu_i} cost_i^e(P^*) = cost_i(P^*)$ and $\sum_{e \in \nu_i} cost_i^{S_e}(P') \leq cost_i(P')$. Thus, $0 \geq \sum_{e \in \nu_i} cost_i^e(P^*) - cost_i^{S_e}(P') \geq cost_i(P^*) - cost_i(P')$, and we are done.

We continue to the final case in which both ν_i and ν'_i are lassos. As in the previous case, $\nu_i = \nu_1$. Recall that the lasso ν_1 is the lasso that is the witness for the resistance of $\mathcal{A}$. We show that the lasso ν'_i violates our requirement for ν_1 and thus we reach a contradiction. Let $\nu_1 = u \cdot v$, where u is a simple path from the initial state and v is a simple cycle. Thus,

$$cost_i(P^*) = cost_i(P^*, u) + cost_i(P^*, v) \leq cost_i(P^*, u \cap \nu'_i) + |u \setminus \nu'_i| + |v|.$$

Also,

$$cost_i(P') = cost_i(P', u \cap \nu'_i) + cost_i(P', \nu'_i \cap v) + |\nu'_i \setminus \nu_i| \geq cost_i(P^*, u \cap \nu'_i) + |\nu'_i \setminus \nu_i|.$$

Subtracting both inequalities we get:

$$cost_i(P^*) - cost_i(P') \leq |u \setminus \nu'_i| + |v| - |\nu'_i \setminus \nu_i|.$$

Since $cost_i(P^*) - cost_i(P') > 0$, we get:

$$|\nu'_i \setminus \nu_i| > |u \setminus \nu'_i| + |v|,$$

which is a contradiction to the resistance of $\mathcal{A}$, and we are done. □

A corollary of Theorem 5.1 is the following:

Corollary 5.2. *For resistant semi-weak games, we have PoS*$= 1$.

We note that resistance can be defined also in WFAs with non-uniform costs, with $cost(\nu)$ replacing $|\nu|$. Resistance, however, is not sufficient in the *slightly* stronger model where the WFA is single-letter and all-accepting but not uniform-cost. Indeed, given k, we show a such a game in which the PoS is kx, for a parameter x that can be arbitrarily close to 1. Consider the WFA A in Figure 5. Note that $\mathcal{A}$ has a single lasso and is thus a resistant WFA. The parameter ℓ_1 is a function of x, and the players' objectives are single words of lengths $\ell_1 \gg \ell_2 \gg \ldots \gg \ell_k \gg 0$. Similar to the proof of Theorem 3.2, there is only one NE in the game, which is when all players choose the left chain. The social optimum is attained when all players use the self-loop, and it is not a NE for similar reasons as in Theorem 3.1: Player 1 pays the majority of the cost of the self loop and can benefit from deviating to the run that uses the self loop once and traverses the path to $q_{\ell_1 - 1}$. The other players follow one by one until Player 1 buys the self loop by himself, in which case it is beneficial to return to the run that uses only the self loop. Then, all the players return one by one until the unstable social optimum is reached. Thus, for a game in this family, $PoS = \frac{k \cdot x}{1}$. Since x tends to 1, we have $PoS = k$ for resistant all-accepting single-letter games.

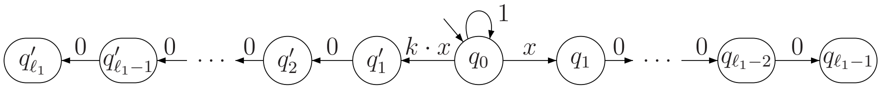

Figure 7: A resistant all-accepting single-letter game in which the PoS tend to k.

6. Surprises in Symmetric Instances

In this section we consider the class of symmetric instances, where all players share the same objective. That is, there exists a language L, such that for all $1 \leq i \leq k$, we have $L_i = L$. In such instances it is tempting to believe that the social optimum is also a NE, as all players evenly share the cost of the solution that optimizes their common objective. While this is indeed the case in all known symmetric games, we show that, surprisingly, this is not valid for AF-games, in fact already for the class of one-letter, all accepting, unit-cost and single-word instances.

Moreover, we start by showing that a NE need not exist in general symmetric instances.

Theorem 6.1. *Symmetric instances of* AF*-games need not have a NE.*

Proof: We describe a two-player symmetric AF game with no NE. Consider a WFA $\mathcal{A}$ consisting of a single accepting state with two self loops, labeled $(a, 1)$ and $(b, \frac{5}{14} - \epsilon)$. Let n_1 and n_2 be such that $0 \ll n_2 \ll n_1$. We define $L = a^6 + ab^{n_1} + aab^{n_2} + aaab$. We denote the 4 strategies available to each of the players by A, B, C, and D, with $A = (6, 0)$ indicating 6 uses of the a-transition and 0 uses of the b-transition, $B = (1, n_1)$, $C = (2, n_2)$, and $D = (3, 1)$.

In order to show that there is no NE, we go over the profiles in the game and show that none of them is a NE. We start with the profile $\langle A, A \rangle$ in which both players pay $\frac{1}{2}$ as they split the cost of the a-transition evenly. This is not a NE as Player 1 (or, symmetrically, Player 2) would deviate to $\langle B, A \rangle$, where he pays $\frac{1}{7}$ for the a-transition and the full price of the b-transition, which is $\frac{5}{14} - \epsilon$, thus he pays $\frac{1}{2} - \epsilon$. The profile $\langle B, A \rangle$ is not a NE as Player 2 pays $\frac{6}{7}$ for a, and can benefit from joining Player 1 in B, where both players pay $\frac{1}{2} + \frac{1}{2} \cdot (\frac{5}{14} - \epsilon) = 0.678 - \epsilon$.

In profile $\langle B, B \rangle$, both players pay $0.678 - \epsilon$. This is not a NE, as Player 1 would deviate to $\langle C, B \rangle$, where he pays $\frac{2}{3}$ for the a-transition and, as $n_2 \ll n_1$, only ϵ for the b-transition. Again, Player 2 benefits from joining Player 1, thereby reducing his cost from $\frac{1}{3} + \frac{5}{14} - \epsilon \approx 0.69$ to $0.678 - \epsilon$.

In profile $\langle C, C \rangle$, again both players pay $0.678 - \epsilon$. By deviating to $\langle D, C \rangle$, Player 1 reduces his payment to $\frac{3}{5} + \epsilon$. Player 2 benefits from joining Player 1, thereby reducing his cost from $\frac{2}{5} + \frac{5}{14} - \epsilon \approx 0.757$ to $0.678 - \epsilon$.

In profile $\langle D, D \rangle$, both players pay $0.678 - \epsilon$ and when deviating to $\langle A, D \rangle$, Player 1 reduces his payment to $\frac{6}{9} \approx 0.667$. Player 2 benefits from joining him, thereby reducing his cost from $\frac{1}{3} + \frac{5}{14} - \epsilon$ to $\frac{1}{2}$. The two final profiles to consider are $\langle C, A \rangle$ from which Player 1 can benefit from deviating to B, and $\langle D, B \rangle$ from which Player 1 can benefit from deviating to C. The other profiles are symmetric, and we are done. □

We turn to study the equilibrium inefficiency, starting with the PoA. It is easy to see that in symmetric AF games, we have $PoA = k$. This bound is achieved, as in the classic network-formation game, by a network with two parallel edges labeled by a and having costs k and 1. The players all have the same specification $L = \{a\}$. The profile in which all players select the expensive path is a NE. We show that $PoA = k$ is achieved even for weak symmetric instances.

Theorem 6.2. *The PoA equals the number of players, already for weak symmetric instances.*

Proof: Recall that $PoA \leq k$, and we show a lower bound of k. The example is a generalization of the PoA in cost sharing games [2]. For k players, consider the weak instance depicted in Figure 6, where all players have the length k as their objective. Intuitively, the social optimum is attained when all players use the loop $\langle q_0, q_0 \rangle$ and thus $OPT = 1$. This profile in a NE, but there is another, more expensive NE. Consider the profile in which all players use the run $q_0 q_1 \ldots q_k$. Its cost is clearly k. This is a NE because a player has two options to deviate. Either to the run that uses only the loop, which costs 1, or to a run that uses the loop and some prefix of $q_0, q_1, \ldots, q_k$, which costs at least $1 + \frac{1}{k}$. Since he currently pays 1, he has no intention of deviating to either runs. Since $OPT = 1$ and the expensive NE costs k, we get $PoA = k$. □

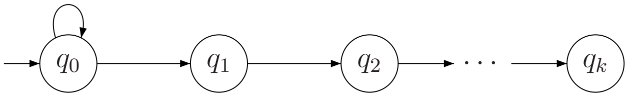

Figure 8: The WFA $\mathcal{A}$ for which a symmetric game with $|L| = 1$ achieves $PoA = k$.

We now turn to the PoS analysis. We first demonstrate the anomaly of having $PoS > 1$ with the two-player game appearing in Figure 9. All the states in the WFA $\mathcal{A}$ are accepting, and the objectives of both players is a single long word. The social optimum is when both players traverse the loop q_0, q_1, q_0. Its cost is $2 + \epsilon$, so each player pays $1 + \frac{\epsilon}{2}$. This, however, is not a NE, as Player 1 (or, symmetrically, Player 2) prefers to deviate to the run $q_0, q_1, q_1, q_1, \ldots$, where he pays the cost of the loop q_1, q_1 and his share in the transition from q_0 to q_1. We can choose the length of the objective word and ϵ so that this share is smaller than $\frac{\epsilon}{2}$, justifying his deviation. Note that the new situation is not a NE either, as Player 2, who now pays slightly less than 2, is going to join Player 1, resulting in an unfortunate NE in which both players pay 1.5.

It is not hard to extend the example from Figure 9 to $k > 2$ players by changing the 2-valued transition to k, and adjusting ϵ and the lengths of the players accordingly. The social optimum and the only NE are as in the two-player example. Thus, the PoS

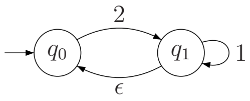

Figure 9: The WFA $\mathcal{A}$ for which the SO in a symmetric game is not a NE.

in the resulting game is $1 + \frac{1}{k}$.

A higher lower bound of $1 + \frac{1}{k-1}$ is shown in the following theorem. Although both bounds tend to 1 as k grows to infinity, this bound is clearly stronger. Also, for $k = 2$, the bound $PoS = 1 + \frac{1}{k-1} = 2$ is tight. We conjecture that $\frac{k}{k-1}$ is tight for every $k > 1$.

Theorem 6.3. *In symmetric k-player games, the PoS is at least $\frac{k}{k-1}$. This already holds for one-letter one-word all-accepting instances.*

Proof: For $k \geq 2$, we describe a family of symmetric games for which the PoS tends to $\frac{k}{k-1}$. For $n \geq 1$, the game $G_{\epsilon,n}$ uses the WFA that is depicted in Figure 10. Note that this is a one-letter instance in which all states are accepting. The players have an identical specification, consisting of a single word w of length $\ell \gg 0$. We choose ℓ and $\epsilon = \epsilon_0 > \ldots > \epsilon_{n-1}$ as follows. Let $C_0, \ldots, C_n$ denote, respectively, the cycles with costs $(k^n + \epsilon_0), (k^{n-1} + \epsilon_1), \ldots, (k + \epsilon_{n-1}), 1$. Let $r_0, \ldots, r_n$ be lasso-runs on w that end in $C_0, \ldots, C_n$, respectively. Consider $0 \leq i \leq n-1$ and let P_i be the profile in which all players choose the run r_i. We choose ℓ and ϵ_i so that Player 1 benefits from deviating from P_i to the run r_{i+1}, thus P_i is not a NE. Note that by deviating from r_i to r_{i+1}, Player 1 pays the same amount for the path leading to C_i. However, his share of the loop C_i decreases drastically as he uses the k^{n-i}-valued transition only once whereas the other players use it close to ℓ times. On the other hand, he now buys the loop C_{i+1} by himself. Thus, the change in his payment is $\frac{1}{k} \cdot (k^{n-i} + \epsilon_i) - (\epsilon' + k^{n-(i+1)} + \epsilon_{i+1})$. We choose ϵ_{i+1} and ℓ so that $\frac{\epsilon_i}{k} > \epsilon' + \epsilon_{i+1}$, thus the deviation is beneficial.

We claim that the only NE is when all players use the run r_n. We note that a profile in which a player uses two cycles C_i and C_j, cannot be a NE as either moving the uses from C_i to C_j is beneficial, or the other way around. So, we focus on profiles

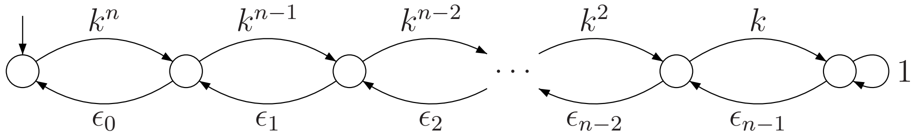

Figure 10: The network of the identical-specification game $G_{\epsilon,n}$, in which PoS tends to $\frac{k}{k-1}$.

in which all players use a strategy in $\{r_1, \dots, r_n\}$. The case when all players select the same run r_i, for $1 \leq i \leq n-1$, is taken care of in the above. Consider a profile $P = \langle r_{i_1}, r_{i_2}, \dots, r_{i_k} \rangle$, where $1 \leq i_1, i_2, \dots, i_k \leq n$. Wlog, assume that $i_1 \leq i_2 \leq \dots \leq i_k$. Using a similar argument as above, Player 1 benefits from deviating to r_{i_1+1}. Indeed, if $i_2 > i_1$, then he is the only player using C_{i_1}, so he pays roughly k^{n-i_1} for it. By deviating, he exchanges this cost for paying at most (roughly) $k^{n-i_1}/k + k^{n-(i_1+1)}$, which is clearly beneficial. On the other hand, if $i_2 = i_1$, then he pays at least $k^{n-i_1}/(k-1)$ for the i_1-th cycle, which he exchanges by deviating, to paying $\epsilon + k^{n-(i_1+1)}$, which is again beneficial, and we are done. □

Finally, we note that our hardness result in Theorem 4.3 implies that finding the social optimum in a symmetric AF-game is NP-complete. Indeed, since the social optimum is the cheapest run on some word in L, finding the best-response in a one-player game is equivalent to finding the social optimum in a symmetric game. This is contrast with other cost-sharing and congestion game (e.g. [22], where the social optimum in symmetric games can be computed using a reduction to max-flow).

Acknowledgments. We thank Michal Feldman, Noam Nisan, and Michael Schapira for helpful discussions.

References

[1] S. Albers, S. Elits, E. Even-Dar, Y. Mansour, and L. Roditty. On Nash Equilibria for a Network Creation Game. In *Proc. 17th SODA*, pages 89-98, 2006.

[2] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. 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. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[4] B. Aminof, O. Kupferman, and R. Lampert, Reasoning about online algorithms with weighted automata, *ACM Transactions on Algorithms*, 6(2), 2010.

[5] B. Alpern and F.B. Schneider. Recognizing safety and liveness. *Distributed computing*, 2:117–126, 1987.

[6] G. Avni and O. Kupferman, Synthesis from Component Libraries with Costs, In *Proc. 25th CONCUR*, pages 156–172, 2014.

[7] G. Avni, O. Kupferman, T. Tamir, Network-Formation Games with Regular Objectives In *Proc. 17th FoSSaCS*, pages 119–133, 2014.

[8] G. Avni, O. Kupferman, T. Tamir, From Reachability to Temporal Specifications in Cost-Sharing Games, In *Proc. 7th IJCAR*, pages 1–15, 2014.

[9] G. Avni, O. Kupferman, T. Tamir, Congestion Games with Multisets of Resources and Applications in Synthesis, In *Proc. 35th FSTTCS*, pages 365–379, 2015.

[10] T. Brihaye, V. Bruyère, J. De Pril, and H. Gimbert. On subgame perfection in quantitative reachability games. *Logical Methods in Computer Science*, 9(1), 2012.

[11] K. Chatterjee. Nash equilibrium for upward-closed objectives. In *Proc. 15th CSL*, LNCS 4207, pages 271–286. Springer, 2006.

[12] K. Chatterjee, T. A. Henzinger, and M. Jurdzinski. Games with secure equilibria. *Theoretical Computer Science*, 365(1-2):67–82, 2006.

[13] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. In *Proc. 18th CONCUR*, pages 59–73, 2007.

[14] K. Chatterjee, R. Majumdar, and M. Jurdzinski. On Nash equilibria in stochastic games. In *Proc. 13th CSL*, LNCS 3210, pages 26–40. Springer, 2004.

[15] H. Chen and T. Roughgarden. Network Design with Weighted Players, *Theory of Computing Systems*, 45(2), 302–324, 2009.

[16] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses. Selfish Routing in Capacitated Networks. *Mathematics of Operations Research* 29: 961–976, 2004.

[17] N. Daniele, F. Guinchiglia, and M.Y. Vardi. Improved automata generation for linear temporal logic. In *Proc. 11th CAV*, LNCS 1633, pages 249–260. Springer, 1999.

[18] M. Droste, W. Kuich, and H. Vogler (eds.), Handbook of Weighted Automata, Springer, 2009.

[19] C. Dwork and M. Naor. Pricing via Processing or Combatting Junk Mail, In *Proc. 12th CRYPTO*, pages 139–177, 1992.

[20] D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In *Proc. 16th TACAS*, LNCS 6015, pages 190–204. Springer, 2010.

[21] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker. On a network creation game. In *Proc. 22nd PODC*, pages 347-351, 2003.

[22] A. Fabrikant, C. Papadimitriou, and K. Talwarl, The Complexity of Pure Nash Equilibria, *Proc. 36th STOC*, pages 604–612, 2004.

[23] M. Feldman and T. Tamir. Conflicting Congestion Effects in Resource Allocation Games. Journal of *Operations Research* 60(3), pages 529–540, 2012.

[24] P. von Falkenhausen and T. Harks. Optimal Cost Sharing Protocols for Scheduling Games. In *Proc. 12th EC*, pages 285-294, 2011.

[25] G. de Giacomo and M. Y. Vardi. Automata-Theoretic Approach to Planning for Temporally Extended Goals, In *European Conferences on Planning*, pages 226–238, 1999.

[26] D. Harel and A. Pnueli. On the development of reactive systems. In *Logics and Models of Concurrent Systems*, volume F-13 of *NATO Advanced Summer Institutes*, pages 477–498. Springer, 1985.

[27] T. Harks, M. Klimm, and B. Peis Resource Competition on Integral Polymatroids *Proc. 10th WINE*, LNCS 8877, pages 189–202. Springer, 2014.

[28] S. Herzog, S. Shenker, and D. Estrin. Sharing the “Cost” of Multicast Trees: An Axiomatic Analysis. *IEEE/ACM Transactions on Networking*, 1997.

[29] K-I. Ko and C-L. Lin. On the complexity of min-max optimization problems and their approximation. In *Minimax and Applications*, volume 4 of *Nonconvex Optimization and Its Applications*, pages 219–239. Springer, 1995.

[30] E. Koutsoupias and C. Papadimitriou. Worst-case Equilibria. *Computer Science Review*,3(2): 65–69, 2009.

[31] O. Kupferman and T. Tamir. Coping with selfish on-going behaviors. *Information and Computation*, 210:1–12, 2012.

[32] M. Mavronicolas, I. Milchtaich, B. Monien, and K. Tiemann. Congestion Games with Player-specific Constants. In *Proc 32nd MFCS*, pp. 633–644, 2007.

[33] I. Milchtaich. Weighted Congestion Games With Separable Preferences. *Games and Economic Behavior*, 67, 750-757, 2009.

[34] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

[35] D. Monderer and L. Shapley. Potential Games. *Games and Economic Behavior*, 14:124–143, 1996.

[36] H. Moulin and S. Shenker. Strategyproof Sharing of Submodular Costs: Budget Balance Versus Efficiency. *Journal of Economic Theory*, 18: 511–533, 2001.

[37] C. A. Meyers Network Flow Problems and Congestion Games: Complexity and Ap- proximation Results Ph.D. dissertation, MIT, 2006.

[38] C. Papadimitriou. Algorithms, Games, and the Internet. In *Proc 33rd STOC*, pages 749–753, 2001.

[39] R. Paes Leme, V. Syrgkanis, E. Tardos. The curse of simultaneity. *Innovations in Theoretical Computer Science (ITCS), pages 60-67, 2012.*

[40] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing IEEE J. on Selected Areas in Communication*, Issue on Copyright and Privacy Protection, 1998.*

[41] R. W. Rosenthal. A Class of Games Possessing Pure-Strategy Nash Equilibria. International Journal of Game Theory*, 2: 65–67, 1973.*

[42] R. W. Rosenthal. A The network equilibrium problem in integers. Networks*, 3: 53–59, 1973.*

[43] E. Tardos and T. Wexler. Network Formation Games and the Potential Function Method, In Algorithmic Game Theory*, Cambridge University Press, 2007.*

[44] B. Vöcking. In N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani, eds., Algorithmic Game Theory. Chapter 20: Selfish Load Balancing. *Cambridge University Press, 2007.*