Alternating Reachability Games with Behavioral and Revenue Objectives

Orna Kupferman and Tami Tamir

[1] School of Eng. and Computer Science, The Hebrew University, Israel, Email: orna@cs.huji.ac.il
[2] School of Computer Science, The Interdisciplinary Center, Herzliya, Israel. E-mail: tami@idc.ac.il

Abstract

We introduce and study *alternating reachability games with tolls* (ARGTs). An ARGT is a multi-player game played on a directed graph. Each player has a source vertex and a set of target vertices. The vertices of the graph are partitioned among the players. Thus, each player owns a subset of the vertices. In the beginning of the game, each player places a token on her source vertex. Whenever a token reaches a vertex v, the owner of the token pays a toll to the owner of vertex v, who directs the token to one of the successors of v. The objective of each player combines a *reachability* objective with a *minimal-cost maximal-profit* objective. For the first, the token of the player needs to reach one of her target vertices. For the second, the player aims at decreasing the toll she pays to other players and increasing the toll paid to her due to visits in vertices she owns. ARGTs model settings in which the vertices are owned by entities who also use the network; for example, communication networks in which service providers own the routers and send messages. ARGTs also offer an extension of rational synthesis with rewards to actions. To the best of our knowledge, this model is the first to combine behavioral and revenue objectives.

We study different instances of the game, distinguishing between various network topologies and various levels of overlap among the reachability objectives of the players. We analyze the *stability* of ARGTs, characterizing instances for which a Nash equilibrium is guaranteed to exist, and studying its inefficiency. We also analyze the problems of finding *optimal strategies* for the players and for the society as a whole.

1 Introduction

Synthesis is the automated construction of a system from its specification. In the late 1980s, several researchers realized that the classical approach to system synthesis, where a system is extracted from a proof that the specification is satisfiable, is well suited to *closed* systems, but not to *open* (also called reactive [14]) systems [1, 10, 23]. In reactive systems, the system interacts with the environment, and a correct system should then satisfy the specification with respect to all environments. The right way to approach synthesis of reactive systems is to consider the situation as a game between the system and the environment.

The game is played on a directed graph. When the game starts, a token is placed on a vertex that corresponds to the initial configuration of the system. Some of the vertices in the graph, these that correspond to configurations in which the system sets the values of the output

signals, are owned by the system, and a strategy for the system determines how to direct tokens that visit these vertices. The other vertices are owned by the environment, and a strategy for the environment determines how to direct tokens that visit them. The strategies of the players induce a single trace for the token in the graph. From a technical point of view, while synthesis of closed systems corresponds to solving a 1-player game, namely to solving a reachability query in a graph, the transition to open systems involved a transition to 2-player games, and to the problem of *alternating reachability* [15].

In the last years, researchers have studied enriched settings of the synthesis problem. Two notable enrichments are a transition to a *multi-player setting*, where the interaction involves not just a system and its environment, but an arbitrary number of components [4], and to an *objective-based* interaction (also known as *rational synthesis* [11, 16, 3]), where each of the interacting components has its own specification. From a technical point of view, the vertices of the corresponding graph are now partitioned among the players, their interaction induces a trace in the graph, and the objective of each player is that this trace would satisfy her specification.[1]

Rational synthesis is strongly related to *mechanism design*, studied in game theory and economy [19, 20]. There, the goal is to design a game whose outcome determines real-valued utilities to the participating players. Typical questions about games in traditional game-theory applications concern their stability — whether the players reach an equilibrium, and their social welfare — maximizing the total players' utility [29]. The most common criterion for stability is the existence of a *Nash equilibrium* (NE, for short) [18]: A profile of strategies, one for each player, is an NE if no (single) player can benefit from unilaterally changing her strategy.

The formal-verification community has already adapted many concepts from classical game theory. Beyond the transition to the rational setting, researchers have studied an extension of rational synthesis to quantitative specifications [11, 16], studied the stability of games with Borel winning conditions [9, 7], and studied *repair* of rational-synthesis games, where repair amounts to making the game stable [2]. Still, the utility of a player in all these games depends only on the trace generated during the game and the level of satisfaction of her objective in this trace.

In this work we introduce and study multi-player games in which the utility of a player combines the satisfaction of her objective in the generated trace with a revenue associated with the trace. This revenue involves payments among the players. We define *alternating reachability games with tolls* (ARGTs, for short). An ARGT is played on a directed graph. The vertices of the graph are partitioned among the players. That is, each player owns a subset of the vertices. In addition, each player has a source vertex and a set of target vertices. In the beginning of the game, each player places a token on her source vertex. Whenever a token reaches a vertex v, that is not a target, the owner of the token pays a toll to the owner of vertex v, who directs the token to one of the successors of v. All tokens passing through v must be directed to the same successor. The objective of each player combines a *reachability* objective with a *minimal-cost maximal-profit* objective. For the first, the token of the player needs to reach one of her target vertices. For the second, the player aims at decreasing the toll she pays to other players and increasing the toll paid to her by other players due to visits in vertices she owns.

To the best of our knowledge, ARGTs are the first model to combine behavioral and revenue objectives. Some research on synthesis does take into account the cost of generating the synthesized system, say when the system is composed from a library of components [6], or when the system has a limited budget for sensing the environment [8] or for dealing with a dishonest

---

[1]Different specification formalisms induce different winning conditions. In particular, starting with specifications in LTL, the game graph is induced by a deterministic parity automaton for the specification, and the winning condition is the ω-regular parity condition [23]. See Section 6 for a discussion on our choice of the reachability winning condition and other technical choices.

environment [17]. In our model, however, the revenue is generated by payments among the players and captures dependencies among them.

Beyond the extension of rational synthesis, ARGTs model settings in which vertices in networks are owned by entities who also use the network, and suggests a formal model for "incentive-aware" routing. Computer networks support an enormous number of applications and services such as access to the World Wide Web, shared-use storage servers, email and instant-messaging applications, among many others. The primary purpose of a network is to provide reachability among the applications running on its end hosts. The users specify their reachability objectives, and a routing protocol assigns to every user a path along which it traverses the network [22, 25]. The emergence of the Internet gives rise to the analysis of the above applications using non-cooperative game-theoretic tools [20]. Indeed, the Internet is used and managed by many agents with conflicting incentives, each aiming to maximize its own utility. In particular, Internet service providers (ISPs) typically provide both Internet transit, web hosting, and Usenet service. Thus, the utilities of ISPs combine various objectives that reflect both the quality of service they provide to their clients as well as their revenue. Our model of ARGT captures such combined utilities. Finally, many applications in AI and planning are modeled by multi-agent games that capture the interaction among the underlying components of a system. Extensive research is carried in order to formally specify and reason about these games [4]. Our work here adds to the picture cost and stability considerations.

We first study the problem of finding the social optimum of a given ARGT, which amounts to maximizing the number of players that fulfill their reachability objectives. We show that the problem, which corresponds to *rational alternating reachability*, is NP-complete in general, but can be solved efficiently for ARGTs where the players share the set of target vertices or ARGTS whose underlying graph is a directed acyclic graph (DAG), with a constant number of players.

Next, we analyze the stability of the game, namely, the existence of an NE and its inefficiency. We show that general ARGTs need not have an NE. We characterize ARGTs that are guaranteed to have an NE, namely symmetric ones (the reachability objectives of all players coincide) whose graph is a DAG. Moreover, we show that the problem of deciding whether a given ARGT has an NE is Σ_2^P-complete. We then turn to analyze the inefficiency caused by the selfish behaviour of the players. We show that there are ARGTs for which all the players can fulfill their reachability objective, yet the only stable profiles are those in which no player does. Thus, the price of stability (PoS) is generally unbounded. Still, we point to classes in which some social optimum is stable (that is, PoS=1).

Finally, we consider the best-response problem, in which a single player needs to calculate an optimal strategy given the strategies of the other players. Once again, we distinguish between general ARGTs for which we show that the problem is NP-complete, and symmetric games played on a DAG, for which we present an efficient algorithm.

2   Preliminaries

2.1   Alternating Reachability Games with Tolls

An *alternating reachability game with tolls* (ARGT) is $\mathcal{N} = \langle k, V_1, \ldots, V_k, E, \langle s_i, T_i \rangle_{i \in [k]}, \gamma \rangle$, where k is the number of players, $V = \bigcup_{i \in [k]} V_i$ is a set of vertices, partitioned into k sets, and $E \subseteq V \times V$ is a set of edges. For each $i \in [k]$, the pair $\langle s_i, T_i \rangle \in V \times 2^V$ describes the reachability objective for Player i, namely forming a path from her source s_i to some target vertex in T_i. Finally, $\gamma \in \mathbb{N}$ is a *reward*. We say that V_i is the set of vertices *owned* by Player i.

For every $v \in V$, let $succ(v) = \{v' : \langle v, v' \rangle \in E\}$. Let $G_{\mathcal{N}} = \langle V, E \rangle$.

A strategy for Player i is a function $f_i : V_i \rightarrow V \cup \{\bot\}$ such that for all $v \in V_i$, if $succ(v) \neq \varnothing$, then $f_i(v) \in succ(v)$, and otherwise $f_i(v) = \bot$. Intuitively, we assume that the players move tokens along the graph. Before the game starts, Player i places her token on s_i. The strategies of the players describe how each player directs tokens that reach the vertices she owns. The objective of each player is twofold. First, she wants her token to reach one of her target vertices. Second, whenever a token visits a vertex on its way to a target vertex, the owner of the token pays the owner of the vertex. Accordingly, each player wants to minimize the number of vertices owned by other players that her token visits, and to maximize the number of visits of tokens of other players in vertices she owns. Below we formalize this intuition.

A *profile* in $\mathcal{N}$ is a tuple $P = \langle f_1, \ldots, f_k \rangle$ of strategies for the players. For a player $i \in [k]$, let $E_i^P \subseteq E$ be the set of edges that Player i chooses in P. Thus, $E_i^P = E \cap \bigcup_{v \in V_i} \{\langle v, f_i(v) \rangle\}$. Note that the intersection with E is required for the vertices in V_i that have an empty set of successors. For a profile P, let $E^P = \bigcup_{i \in [k]} E_i^P$, and let $G_{\mathcal{N}}^P = \langle V, E^P \rangle$ be the subgraph of $\mathcal{N}$ induced by the strategies in P. Thus, $G_{\mathcal{N}}^P$ consists of all the edges chosen by the players in their strategies in P. Note that each vertex in V has at most one successor in $G_{\mathcal{N}}^P$. That is, $G_{\mathcal{N}}^P$ is a directed graph in which every vertex with $succ(v) \neq \varnothing$ has out-degree 1.

For $i \in [k]$, Player i fulfills her reachability objective in P if the graph $G_{\mathcal{N}}^P$ includes a path from s_i to a vertex in T_i. Note that $G_{\mathcal{N}}^P$ includes a single path $\pi_i^P = v_0, v_1, v_2, \ldots$ with $v_0 = s_i$. Player i fulfills her reachability objective in P iff π_i^P reaches T_i. Then, let $m_i \geqslant 0$ be the minimal index such that $v_{m_i} \in T_i$, and let $visit(i, P) = \{v_0, v_1, \ldots, v_{m_i - 1}\}$ be the set of vertices that Player i traverses in order to fulfill her reachability objective, namely the set of vertices in the prefix of π_i^P up to and excluding the first vertex in T_i. Note that a player may not fulfill her reachability objective in P, in which case $visit(i, P) = \varnothing$.

For a vertex $v \in V$, let $load(v, P)$ denote the number of players that traverse the vertex v in their paths to their target vertices in P. Formally, $load(v, P) = |\{i : v \in visit(i, P)\}|$. For a player $i \in [k]$, we define $cost(i, P) = |visit(i, P)|$ and $profit(i, P) = \sum_{v \in V_i} load(v, P)$. Finally, let $reward(i, P) = \gamma$ if Player i fulfills her reachability objective in P and $reward(i, P) = 0$, otherwise. The total utility of Player i in P combines her behavioral objective, measured by $reward(i, P)$, and her revenue objective, where $revenue(i, P) = profit(i, P) - cost(i, P)$, sums the tolls she pays and collect. Formally, $utility(i, P) = reward(i, P) + revenue(i, P)$. Note that if Player i fulfills her objective, then she gets a reward γ, pays the other players for passing through their vertices, and is paid by players that pass through her vertices. If Player i does not fulfill her objective, then she only gets paid by players that pass through her vertices. The rationale behind this is that in this case, the player will not use the network as a client. In practice, γ is big enough to incentivize a fulfillment of the reachability objectives. Note that if a player owns a vertex in $visit(i, P)$, then she pays to herself, and the associated cost and profit cancel each other in the revenue calculation.

For a profile P, we define the utility of P as the sum of the utilities of the players. Thus, $utility(P) = \sum_{i \in [k]} utility(i, P)$. The following simple observation follows from the fact that in all profiles, the sum of profits of all players is equal to the sum of costs. Thus, only the rewards given to players that fulfill their reachability objectives are not cancelled in the calculation of $utility(P)$.

Observation 1. *For every profile P, we have that $utility(P) = x\gamma$, where $x \leqslant k$ is the number of players that fulfill their reachability objective in P.*

An ARGT $\mathcal{N}$ is a *DAG-ARGT* if the graph $G_{\mathcal{N}}$ is a DAG. It is *symmetric*, if all players share the same reachability objective, and is *common-target*, if they only have the same set of

target vertices. We use SYM and CT to denote the symmetric and common-target instances. If, in addition, the target sets are singletons, we add 1 to the description. So, for example, a DAG-CT1-ARGT is an ARGT whose graph is a DAG and in which all players have the same single target vertex. Beyond the theoretical interest in these classes, DAG instances correspond to systems in which edges reflect progress in time or precedence constraints. SYM and CT instances arise in applications servicing clients with uniform or partially-uniform objectives.

Example 1. Consider the DAG-CT1-ARGT $\mathcal{N}$ presented in Figure 1(i). In this game, $k = 2$, and we denote the two players by a and b. Let $V_a = \{s_a, s_b, w\}$ and $V_b = \{u, v, t\}$. This partition is illustrated in the figure by round (Player a) and rectangle (Player b) vertices. The players objectives are $\langle s_a, t \rangle$ and $\langle s_b, t \rangle$, respectively. Let P_0 be the profile described by the bold edges, thus, bold edges are in E^{P_0}. For example, $f_a^0(s_a) = u$ and $f_b^0(u) = t$. In this profile, both players fulfill their reachability objective, $\pi_a^{P_0} = \langle s_a, u, t \rangle$, and $\pi_b^{P_0} = \langle s_b, u, t \rangle$, In addition, $visit(a, P_0) = \{s_a, u\}$, and $visit(b, P_0) = \{s_b, u\}$. Thus, $load(s_a, P_0) = load(s_b, P_0) = 1$, $load(u, P_0) = 2$, and for every vertex $v' \in \{v, w, t\}$, we have that $load(v', P_0) = 0$. Accordingly, $cost(a, P_0) = cost(b, P_0) = 2$, $profit(a, P_0) = profit(b, P_0) = 2$, and $utility(a, P_0) = utility(b, P_0) = \gamma$.
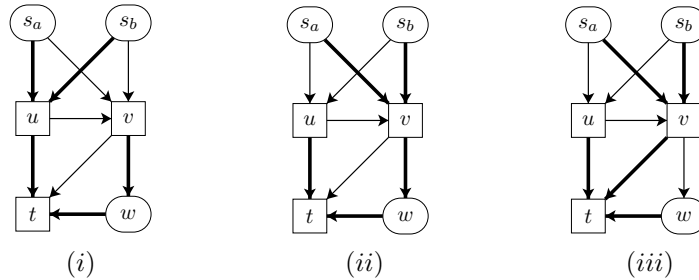

Figure 1: The ARGT $\mathcal{N}$: (i) the profile P_0, (ii) the profile P_1, and (iii) the NE P_2.

2.2   Game Theory, Definitions and Notations

Consider an ARGT $\mathcal{N}$, profile P of $\mathcal{N}$, a player $i \in [k]$, and a strategy f_i' for Player i. We use (P_{-i}, f_i') to denote the profile obtained from P by replacing the strategy of Player i by f_i'. For a given profile P, a *best response* (BR) for Player i is a strategy f_i' that maximizes $utility(i, (P_{-i}, f_i'))$. In the *BR problem*, we are given an ARGT $\mathcal{N}$, a profile P, and a player $i \in [k]$, and we want to find a strategy f_i' such that the utility of Player i in (P_{-i}, f_i') is maximal.

A profile P is said to be a *(pure) Nash equilibrium* (NE)[2] if none of the players in $[k]$ can benefit from an unilateral deviation from her strategy in P. In other words, for every $i \in [k]$ and every strategy f_i', it holds that $utility(i, (P_{-i}, f_i')) \leqslant utility(i, P)$. The set of NEs of the game $\mathcal{N}$ is denoted by $\Gamma(\mathcal{N})$.

A *social optimum* (SO) of an ARGT $\mathcal{N}$ is a profile that attains the maximal utility. We denote by $OPT(\mathcal{N})$ the utility of an SO profile; i.e., $OPT(\mathcal{N}) = \max_P utility(P)$. By Observation 1, we have that $OPT(\mathcal{N})$ is $x\gamma$, where x is the maximal number of players that can

---

[2]Throughout this paper, we consider *pure* strategies. Unlike mixed strategies, pure strategies may not be random or drawn from a distribution.

simultaneously fulfill their reachability objectives. In the *SO problem*, we are given an ARGT $\mathcal{N}$ and a threshold $\kappa \geqslant 0$, and we need to decide whether $OPT(\mathcal{N}) \geqslant \kappa$.

A social optimum may be achieved by a centralized authority and need not be an NE. The following parameters measure the inefficiency caused as a result of the players' selfish interests. First, the *price of anarchy* (PoA) [21] of $\mathcal{N}$ is the ratio between the utility of a social optimum and the utility of a worst NE. That is, $\text{PoA}(\mathcal{N}) = \max_{P \in \Gamma(\mathcal{N})} OPT(\mathcal{N})/utility(P)$. Then, the *price of stability* (PoS) [5] of an ARGT $\mathcal{N}$ is the ratio between the utility of a social optimum and the utility of a best NE. That is, $\text{PoS}(\mathcal{N}) = \min_{P \in \Gamma(\mathcal{N})} OPT(\mathcal{N})/utility(P)$.

Example 2. Consider again the ARGT presented in Example 1. The profile P_0 described in the example is not an NE. Figure 1(*ii*) presents the profile P_1 resulting from a deviation of Player a from P_0. By changing her strategy in the source vertices, specifically, by setting $f_a^1(s_a) = f_a^1(s_b) = v$, Player a forces Player b to visit in vertex w and increases her revenue. In P_1, we have that $utility(a, P_1) = \gamma + 1$, while $utility(b, P_1) = \gamma - 1$.

The profile P_1 is not an NE either. Figure 1(*iii*) presents the profile P_2 resulting from a deviation of Player b from P_1. By changing her strategy in v, specifically, by setting $f_b^2(v) = t$, Player b can avoid the visit in w and reduces her cost. In P_2, we have that $utility(a, P_2) = utility(b, P_2) = \gamma$. The profile P_2 is an NE. Note that $f_b(u)$ does not affect the players' utilities, as no path from the source vertices in $G_{\mathcal{N}}^{P_2}$ visits u.

We note that $\mathcal{N}$ would converge to a different NE if Player b is the first to respond to P_0. By setting $f_b(u) = v$, Player a is forced to visit both u and v on her way to t. However, Player a will then respond by setting $f_a(s_a) = f_a(s_b) = v$. Finally, Player b will respond by setting $f_b(v) = t$, and save the visit in w. The resulting profile is an NE that differs from P_2 in $f_b(u)$.

3   The Social-Optimum Problem

In this section we study the SO problem. By Observation 1, in the SO problem we only care for the behavioral objectives, and solve a *rational alternating reachability* problem. That is, the problem of maximizing the number of players fulfilling their reachability objective. More formally, given a graph and a set of reachability objectives, assign a single successor to each vertex in a way that maximizes the number of reachability objectives fulfilled in the induced graph. Some of the algorithmic challenges arising in our setting are closely related to the problem of finding k disjoint paths in directed graphs. This problem is known to be NP-complete for every $k \geqslant 2$ [12]. However, for DAGs, an $O(|E||V|)$-time algorithm for $k = 2$ was given in [27], and was generalized later for every $k \geqslant 2$ to an $O(|E||V|^{k-1})$-time algorithm [12].

Theorem 1. *The SO problem for ARGT is NP-complete. NP-hardness holds already for general graphs with 2 players and for DAG-ARGTs with $O(|V|)$ players.*

Proof: Membership in NP is easy, as given a witness profile P, it is possible to calculate the number of players that fulfill their reachability objectives in P in polynomial time.

We turn to prove the hardness result for 2 players. Consider a directed graph $G = \langle V, E \rangle$ and two vertices $u, v \in V$. By [12], the problem of deciding whether there are vertex-disjoint paths from u to v and from v to u (that is, except for u and v, the two paths share no vertex) is NP-hard. We describe a reduction from this problem to the SO problem in a 2-player ARGT. Given $G = \langle V, E \rangle$, we define $\mathcal{N} = \langle 2, V, \varnothing, E, \{\langle u, v \rangle, \langle v, u \rangle\}, 1 \rangle$. Thus, all the vertices belong to Player 1, her reachability objective is to form a path from u to v, and the reachability objective of Player 2 is to form a path from v to u.

We prove that $OPT(\mathcal{N}) \geqslant 2$ iff there are vertex-disjoint paths from u to v and from v to u. First, if such paths exist, then the function $f_1 : V \rightarrow V$ that assigns to each vertex that participate in one of the paths its successor in the single path it belongs to and map other vertices to arbitrary successors, if a strategy for Player 1, and every profile P in which Player 1 follows f_1 is such that both players fulfill their reachability objectives in P. Hence, by Observation 1, we have $utility(P) = 2$ and $OPT(\mathcal{N}) \geqslant 2$.

For the other direction, assume that $OPT(\mathcal{N}) \geqslant 2$, and let f_1 be a strategy of Player 1 in a profile P with $utility(P) = 2$. By Observation 1, both players fulfill their reachability objectives in P. Thus, E^P includes a path form u to v and from v to u. We prove that these paths are vertex disjoint. Assume by way of contradiction that there is a vertex $w \in V \backslash \{u, v\}$ that participate in both paths. The graph $G_{\mathcal{N}}^P$ has a single path from w. If without loss of generality, this path reaches v, then $G_{\mathcal{N}}^P$ includes a cycle from v back to itself via w, and thus $G_{\mathcal{N}}^P$ does not contain a path from u to v, contradicting the assumption.

To prove NP-hardness for DAG-ARGTs, we describe a reduction from SAT. Given a CNF propositional formula θ with m clauses, we construct a DAG-ARGT $\mathcal{N}_\theta$ with $m + 1$ players such that $OPT(\mathcal{N}_\theta) \geqslant m$ iff θ is satisfiable. Let $X = \{x_1, \ldots, x_n\}$ be a set of variables and let $\theta = \bigwedge_{1 \leqslant i \leqslant m} C_i$, with $C_i = l_i^1 \vee l_i^2 \vee \cdots l_i^{m_i}$, where l_i^j, for $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant m_i$, is a literal, namely a variable in X or its negation.

The ARGT $\mathcal{N}_\theta$ (see Figure 2) consists of $m + 3n$ vertices. There are n "variable vertices" $x_1, \ldots, x_n$, with vertex x_i branching into two "literal vertices" x_i^T and x_i^F, associated with the literals x_1 and $\overline{x_i}$, respectively. Player $m + 1$ owns the variable vertices, thus a strategy for her corresponds to a truth assignment to the variables. In addition, there are m "clause vertices" $C_1, \ldots, C_m$, with C_i being owned by Player i. A clause vertex C_i branches into the variable vertices of variables that appear in its literals. For example, in the figure, $C_2 = x_1 \vee \overline{x_2}$ and the vertex C_2 branches into the vertices x_1 and x_2. The source of Player i is C_i, and her set T_i of target vertices is the set of literal vertices associated with the literals in C_i. For example, $T_2 = \{x_1^T, x_2^F\}$. Finally, the reward for fulfilling a reachability objective is 1. For simplicity, we assume that Player $m+1$ does not have a reachability objective (alternatively, we could proceed with an objective satisfied by an empty path, say $\langle x_1, \{x_1\}\rangle$, or an objective that is equal to an objective of one of the clause players, and then require $OPT(\mathcal{N}_\theta)$ to be at least $m + 1$).

We prove that θ is satisfiable iff $OPT(\mathcal{N}_\theta) \geqslant m$. Assume that θ is satisfiable. Consider a profile in which Player $m + 1$ chooses a satisfying assignment in her strategy from the variable vertices and the clause players each choose a variable in a satisfied literal. It is easy to see that all the clause players fulfill their reachability objectives. Thus, by Observation 1, we have $OPT(\mathcal{N}_\theta) = m$. Assume that $OPT(\mathcal{N}_\theta) \geqslant m$, thus there is a profile P in which all m clause players fulfill their reachability objectives, thus in E^P there are paths from C_i to one of its literals for all $1 \leqslant i \leqslant m$. By the construction of $\mathcal{N}_\theta$, these paths have to go via variables vertices, and the strategy of Player $m + 1$ in these vertices induce a satisfying assignment to θ. $\square$

In Theorem 2 below, we show that combing the two classes for which we could show NP-hardness in Theorem 1 results in a class for which the SO problem can be solved in polynomial time. In Theorem 3, we point to another class of ARGTs, namely CT-ARGTs, for which the problem can be solved efficiently. The proof is based on detecting all source vertices from which some target vertex is reachable.

Theorem 2. *The SO problem for DAG-ARGT and a constant number of players can be solved in polynomial time.*
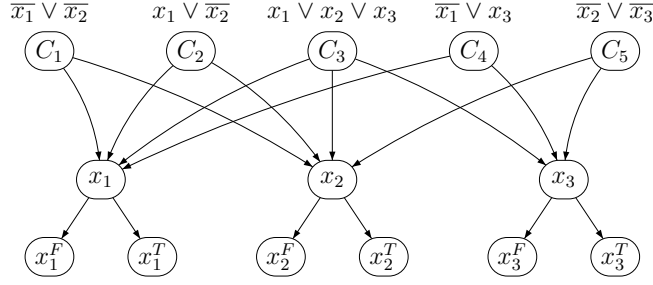
Figure 2: The DAG-ARGT $\mathcal{N}_\theta$ for $\theta = (\overline{x_1} \vee \overline{x_2}) \wedge (x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_3) \wedge (\overline{x_2} \vee \overline{x_3})$.

Proof: For simplicity, we assume that the set of target vertices of all players are singletons; that is $T_i = \{t_i\}$, for all $i \in [k]$. The general case can then be reduced to $\Pi_{i\in[k]}|T_i|$ executions of the algorithm.

Given $\mathcal{N}$ and an integer κ, we check whether $OPT(\mathcal{N}) \geq \kappa$ as follows. First, as k is a constant, so is κ. Hence, so is the number of subsets $W \subseteq [k]$ of size κ. Our algorithm goes over all subsets $W \subseteq [k]$ of size κ, and checks whether all the players in W can fulfill their objectives. That is, if $\Pi = \{\langle s_{i_1}, t_{i_1}\rangle, \ldots, \langle s_{i_{|W|}}, t_{i_{|W|}}\rangle\}$ is the set of objectives of the players in W, then we need to check whether there are paths that fulfill Π such that all the vertices in the induced subgraph of $G_\mathcal{N}$ have out-degree at most 1.

By [12], the problem of finding a constant number of vertex-disjoint paths in a DAG can be solved in polynomial time. Specifically, the algorithm in [12] gets as input a DAG G and a set $\{\langle u_1, u_1'\rangle, \ldots, \langle u_q, u_q'\rangle\}$ of q source-target pairs, and decides whether there are q vertex-disjoint paths in G that connect each source with its target. Our problem is similar, except that the paths fulfilling Π need not be disjoint, but their union form a forest of in-trees (directed trees in which internal nodes have out-degree 1).

In order to reduce our problem to the one in [12] we go over all possible ways to describe a forest of in-trees satisfying Π as a collection of at most q vertex-disjoint paths. Formally, let $S = \{s_{i_1}, \ldots, s_{i_{|W|}}\}$ and $T = \{t_{i_1}, \ldots, t_{i_{|W|}}\}$. Then, a *scheme* for Π is a revised set of objectives $\Pi' = \{\langle s_{i_1}, t_{i_1}'\rangle, \ldots, \langle s_{i_{|W|}}, t_{i_{|W|}}'\rangle\}$ such that for every $j \in W$, we have that $t_{i_j}' \in S \cup T$, and Π is fulfilled iff Π' is fulfilled. Intuitively, t_{i_j}' is the first vertex in $S \cup T$ that the path from s_{i_j} to t_{i_j} visits. Consequently, there are paths that fulfill Π such that all induced subgraph of $G_\mathcal{N}$ is a forest of in-trees iff there is a scheme Π' for Π such that there are $|W|$ vertex-disjoint paths in $G_\mathcal{N}$ that connects its sources and targets. Clearly, since $|W|$ is fixed, so is the number of schemes for Π. Our algorithm goes over all these schemes and apply the algorithm in [12] for each of them. □

Theorem 3. *The SO problem for CT-ARGTs can be solved in linear time.*

Proof: Given $G_\mathcal{N}$, add a new "dummy-target" vertex t', and $|T|$ edges connecting every vertex in T to t'. Now, by performing $BFS(t')$ on the graph with reverse edge directions, we can identify all sources s_i from which t' is reachable. The set of edges forming the BFS tree, in their original direction, induces a valid strategy in which all the players for which the reachability objective can be fulfilled, are directed to some target. □

4   Stability of ARGTs

In this section we study the stability of ARGTs. First, in Section 4.1, we show that general ARGTs need not have an NE. We give a precise characterization of ARGTs that are guaranteed to have an NE, namely DAG-SYM-ARGTs. Indeed, we describe a linear-time algorithm for calculating an NE in such ARGTs, and we also show that non-symmetric ARGTs need not have an NE, even when their graph is a DAG, and that so do ARGTs that include cycles, even when they are symmetric. Moreover, we show that the problem of deciding whether a given ARGT has an NE is Σ_2^P-complete. Then, in Section 4.2, we study the PoA and Pos of ARGTs and show that they are both unbounded in the general case.

4.1   An NE and its Existence

Theorem 4. *ARGTs need not have an NE. In fact, there is a 2-player DAG-CT1-ARGT with no NE, as well as a 2-player SYM1-ARGT with no NE.*

Proof: We start with an example of a DAG-CT1-ARGT with no NE. Consider the ARGT $\mathcal{N}_1$ described in Figure 3. Players a and b, each own her source vertex s_a and s_b, respectively, and has to reach the target vertex t. The identity of the owner of the vertex t is not important, and so is the reward γ for fulfilling reachability objectives. For simplicity of presentation, edges in the figure correspond to chains of vertices in $\mathcal{N}_1$: each edge is labeled by a word over $\{a, b\}$, indicating the owners of the vertices in the chain. For example, the edge labeled a^2b^2 from s_a to t stands for a chain consisting of four edges with the first two vertices owned by Player a (the first of them being s_a), followed by two vertices owned by Player b, and then the vertex t.

Player a has to choose between two strategies, corresponding to choosing the left or the right chain leaving s_a. Player b has to choose between two strategies, corresponding to choosing the top or the bottom chain connecting s_b to t. Thus, the game has four possible profiles. Table 1 describes the utility of the players in the possible profiles. In all four profiles, both players fulfill their reachability objectives. However, their revenues vary. It can be verified that no NE exists, as players benefit from moving among the profiles in the table in a counter-clockwise direction. For example, in the (LEFT,TOP)-profile, every player pays 1 to the other player (for using the ba chain). Therefore, they both have utility γ. By deviating to the (LEFT,BOTTOM)-profile, Player b increases her profit to 3 and her cost to 2, resulting in utility $\gamma - 1$ for Player a and $\gamma + 1$ for Player b.
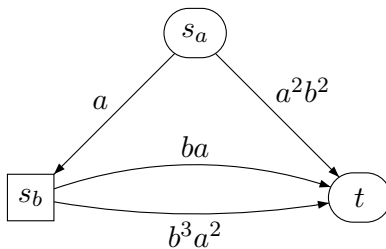
$b\backslash a$	LEFT	RIGHT
TOP	γ, γ	$\gamma + 1, \gamma - 1$
BOTTOM	$\gamma + 1, \gamma - 1$	γ, γ

Figure 3: A 2-player DAG-CT1-ARGT with no NE.

Table 1.   Players utilities. Every entry describes the utility of Player b followed by the utility of Player a.

We continue to an example of a SYM1-ARGT with no NE. Consider the ARGT $\mathcal{N}_2$ presented in Figure 4. As in Figure 3, edges in the figure correspond to chains of vertices and each edge

is labeled by a word over $\{a, b\}$, indicating the owners of the vertices in the chain. Note that $s \in V_a$ and $\{u, v\} \subseteq V_b$. Also, t is a sink and all other vertices have out-degree 1. Thus, a strategy is for Player a is fully defined by $f_a(s)$, and a strategy for Player b is fully defined by $f_b(u)$ and $f_b(v)$. Table 2 presents the players' utilities in the possible profiles. For simplicity, we denote a strategy by the edges each player selects to E^P. Note that if Player b's strategy is $(u, v), (v, u)$, then the players do not fulfill their reachability objective and their utility is 0. Every other profile induces one of the following (s, t)-paths: $p_1 = \langle s, u, v, t \rangle$, $p_2 = \langle s, v, t \rangle$, $p_3 = \langle s, v, u, t \rangle$, and $p_4 = \langle s, u, t \rangle$.

The path $p_1 = \langle s, u, v, t \rangle$ consists of 7 edges, out of which Player a owns 3, and Player b owns 4. Therefore, in the corresponding profile, the utility of a is $\gamma - 1$ and the utility of b is $\gamma + 1$. The calculation is similar for the other paths.

It is easy to verify that $\mathcal{N}_2$ has no NE. If Player b's strategy is $(u, v), (v, u)$, she can deviate to a strategy that includes either (u, t) or (v, t) – in which the reachability objective is fulfilled. For every other profile, note that Player a will deviate from p_1 to p_2 and from p_3 to p_4, while Player b will deviate from p_2 to p_3 and from p_4 to p_1. Thus, $\mathcal{N}_2$ does not have a NE profile. $\square$

Figure 4: A 2-player SYM-ARGT with no NE.
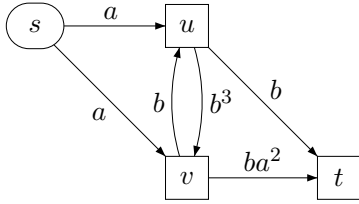
$b \backslash a$	(s, u)	(s, v)
$(u, v), (v, t)$	$p_1 : (\gamma + 1, \gamma - 1)$	$p_2 : (\gamma - 2, \gamma + 2)$
$(u, v), (v, u)$	$none : (0, 0)$	$none : (0, 0)$
$(u, t), (v, u)$	$p_4 : (\gamma, \gamma)$	$p_3 : (\gamma + 1, \gamma - 1)$
$(u, t), (v, t)$	$p_4 : (\gamma, \gamma)$	$p_2 : (\gamma - 2, \gamma + 2)$

Table 2. Players utilities in $\mathcal{N}_2$. Every entry specifies the path induced by the strategy, the utility of Player b, and the utility of Player a.

The examples in Theorem 4 leave open the question of the existence of an NE in an ARGT that is both DAG and symmetric. For this class we have some good news: We show that in DAG-SYM-ARGT, an NE always exists, and can be calculated efficiently.

Theorem 5. *Every DAG-SYM-ARGT has an NE, which can be calculated in linear time.*

Proof: Consider a DAG-SYM-ARGT $\mathcal{N} = \langle k, V_1, \ldots, V_k, E, \langle s, T \rangle, \gamma \rangle$. Let $v_1, \ldots, v_n$ be a topological sort of V, thus for all indices $1 \leq i < j \leq n$, we have that v_i is not reachable from v_j. We assume that all the vertices in V are reachable from s, as non-reachable vertices can be removed. Also, $v_n \in T$, as otherwise, all vertices that appear after the last target vertex in the sort can be removed. Accordingly, we assume that the sort is such that $v_1 = s$ and $v_n \in T$. For $1 \leq \ell \leq n$, let $\mathcal{N}_\ell$ be the ARGT obtained from $\mathcal{N}$ by restricting its graph to the subgraph of $G_\mathcal{N}$ induced by $\{v_\ell, \ldots, v_n\}$, and defining the reachability objective to be $\langle v_\ell, T \rangle$. Starting with $\ell = n$, we proceed by induction on ℓ and construct an NE in the ARGT $\mathcal{N}_\ell$. In particular, an NE in $\mathcal{N}_1$ is an NE in $\mathcal{N}$.

The calculation of the NE for $\mathcal{N}_\ell$ proceeds as follows. First, for $\ell = n$, since $v_n \in T$, a profile in which the strategies induce an empty path is an NE in the game $\mathcal{N}_n$, where $s = v_n$. For the induction step, assume that an NE was already calculated for each of the ARGTs $\mathcal{N}_n, \ldots, \mathcal{N}_{\ell+1}$. We proceed to calculate an NE for $\mathcal{N}_\ell$. Let i be the player who owns v_ℓ, that is, $v_\ell \in V_i$. For every edge $\langle v_\ell, u \rangle$, the topological sort implies that the ARGT $\mathcal{N}_u$ was analyzed already. Let P_u be the NE calculated for $\mathcal{N}_u$. Let u' be the vertex for which $\langle v_\ell, u' \rangle \in E$ and the utility of Player i in the game $\mathcal{N}_{u'}$ is maximal. We claim that the profile in which $f_i(v_\ell) = u'$, and in all

other vertices all players follow their strategies in $P_{u'}$ is an NE in $\mathcal{N}_\ell$. To see this, note that a beneficial deviation of Player j from her strategy $f_j(v_{\ell'})$ in $P_{u'}$, for $\ell' > \ell$, induces a beneficial deviation for her in $P_{u'}$, contradicting the stability of $P_{u'}$. Finally, a deviation of Player i from $f_i(v_\ell)$ is not beneficial, as $f_i(v_\ell)$ was selected to be most profitable for her.

The time analysis is straightforward. The topological sort takes time $O(|V| + |E|)$. Then, in the NE calculation every vertex and every edge are considered once, and the whole process takes linear time.   □

The fact an ARGT need not have an NE gives rise to the problem of deciding whether a given ARGT has an NE. We show that the problem is complete in Σ_2^P, namely the class of problems that can be solved by a nondeterministic polynomial Turing machine that has an oracle to some NP-complete problem.

Theorem 6. *The problem of deciding whether an NE exists in a given ARGT is Σ_2^P-complete. Hardness in Σ_2^P holds already for DAG-ARGTs.*

Proof: We start with the upper bound. Consider an ARGT $\mathcal{N}$ and a profile P in $\mathcal{N}$. Checking whether P is not an NE can be done in NP. Indeed, the check can be done by guessing a beneficial deviation of one of the players and checking that it is indeed beneficial (in Theorem 10, we formalize this further, by means of the complexity of the BR problem). Consequently, as P can be guessed, deciding whether an NE exists can be done by a nondeterministic polynomial-time Turing machine with an NP oracle.

We continue to the lower bound and describe a reduction from QBF$_2$: satisfiability for quantified Boolean formulas with 2 alternations of quantifiers, where the most external quantifier is "exists". Let θ be a Boolean propositional formula in CNF over the variables $X = \{x_1, \ldots, x_{n_1}\}$ and $Y = \{y_1, \ldots, y_{n_2}\}$, and let $\psi = \exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_l \theta$. We construct an ARGT $\mathcal{N}_\psi$ such that N_ψ has an NE iff ψ is satisfiable.

Recall the reduction from SAT to the SO problem described in the proof of Theorem 1. There, a single player determines the assignment to the variables. In the reduction here, we have one player, referred to as Player X, that determines the assignment to the variables $x_1, \ldots, x_n$ and one player, referred to as Player Y, that determines the assignment to the variables $y_1, \ldots, y_l$. The ARGT $\mathcal{N}_\psi$ contains m "no-NE gadgets", $\mathcal{N}_1, \ldots, \mathcal{N}_m$, one for each clause, in which no NE exists. The construction of $\mathcal{N}_\psi$ is such that when ψ is not satisfiable, Player Y has an incentive to deviate to a profile in which the utility of at least one clause players goes below the utility she can get if she changes her strategy to one that proceed to her no-NE gadget.

We now describe the reduction in more detail (see Figure 5). Given ψ, there are $2m + 2$ players: m clause players $C_1, \ldots, C_m$, as in Theorem 1, Player X, Player Y, and m dummy players $D_1, \ldots, D_m$, who play in the no-NE gadgets. In addition to the literal vertices that correspond to literals in C_i, the set of target vertices for Player C_i contain the vertices t_i and t, explained below.

A clause player C_i may proceed to the no-NE gadget, where her utility is $\gamma - 2$ or $\gamma - 3$. The gadgets are similar to the game described in the proof of Theorem 4, except that the game on $\mathcal{N}_i$ is played between C_i and D_i (hence the c and d labels on the edges), and is preceded by a chain of 2 vertices owned by Player D_i. Accordingly, the utility of the players have as in Theorem 4, with an increase of 2 to D_i and a decrease of 2 to C_i. Hence, the utility of C_i if she takes the transition to $\mathcal{N}_i$ is $\gamma - 2$ or $\gamma - 3$, depending on the strategy of the dummy player D_i. As in Theorem 4, the objective of the dummy player is to get to t_i from the vertex below it. A clause player may also proceed to one of the variables in its literals, hoping that Player X or Y would proceed to one of the literal vertices in her targets.

All the literal vertices $y_1^F, y_1^T, \ldots, y_l^F, y_l^T$ are owned by Player Y, and have an edge to a chain of length 3, leading to the vertex t, which is in the target set of all clause players. That is, when a clause player C_i hopes to fulfill her reachability objective by choosing a Y variable y, and Player Y does not assign to y a value that satisfies the clause C_i, then Player C_i still fulfills her reachability objective, but she does so by traversing the chain to t, paying 4 to Player Y. The incentive of Player Y is then to maximize the number of clause players that reach t. Once, however, a clause player reaches t, making her utility $\gamma - 4$, she has an inventive to deviate to her no-NE gadget, where her utility is $\gamma - 2$ or $\gamma - 3$.
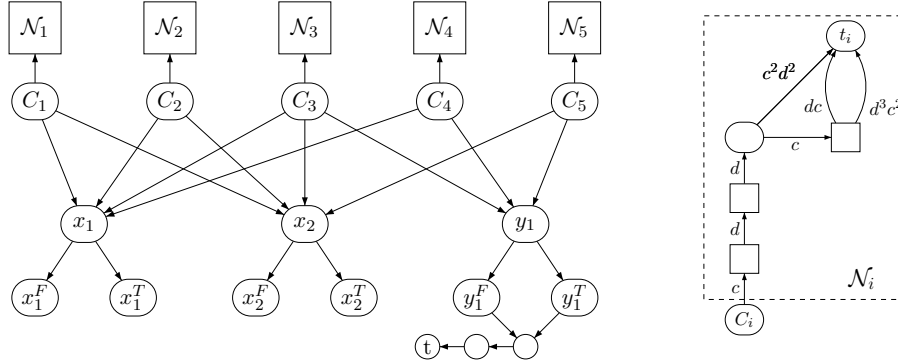

Figure 5: The DAG-ARGT $\mathcal{N}_\psi$ for $\psi = \exists x_1 \exists x_2 \forall y_1 (\overline{x_1} \vee \overline{x_2}) \wedge (x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_2 \vee y_1) \wedge (\overline{x_1} \vee y_1) \wedge (\overline{x_2} \vee \overline{y_1})$. The "no-NE gadget" $\mathcal{N}_i$ appears on the right.

We prove that $\mathcal{N}_\psi$ has an NE iff ψ is satisfiable. Assume first that ψ is satisfiable, and let P be a profile in which Player X assigns to the variables in X an assignment such that for every assignment to the variables in Y, the formula θ holds. Since ψ is satisfiable, such an assignment to the variables in X exists. Also, the clause players proceed to the variable vertices that appear in the literal that is satisfied in their clause, and all dummy players D_i take the dc chain to t_i. We claim that P is an NE. The utilities in the profile P are as follows: each of the clause players gets the reward γ, pays 1 to Player X or Y, and is being paid 1 from her dummy player. Then, Players X or Y get revenues according to the number of clause players that satisfy their reachability objectives via variables in X or Y. This revenue is between 0 to m. Deviating from this strategy is not beneficial, as no matter how Player Y deviates, the clause players continue to satisfy their reachability objective without getting to the vertex t. Finally, as Player C_i is not aiming for t_i, the dummy players have no incentive to change their strategies, which gives them utility $\gamma - 1$.

Assume now that ψ is not satisfiable. We argue that $\mathcal{N}_\psi$ has no NE. Consider a profile P. We distinguish between two cases. First, if some clause player C_i proceeds up to $\mathcal{N}_i$, then, as in the proof of Theorem 4, the players C_i and D_i would perform an infinite BR dynamics, and P is not an NE. Assume then that all clause players proceed to variable vertices. Let g be the assignment to the variables in $X \cup Y$ that is induced by the strategies of Player X and Player Y. If g does not satisfy θ, then at least one clause player C_i does not fulfill her reachability objective of fulfills it by reaching the target vertex t, paying 4 to Player Y. Thus, at least one clause player has utility at most $\gamma - 4$, and would deviate to her no-NE gadget, where her utility is $\gamma - 3$ or $\gamma - 2$. Thus, P is not an NE. Now, if g does satisfy θ, then, as ψ is not satisfiable, there is an assignment g' that does not satisfy θ and is obtained from g by changing only the assignment of variables in Y. By deviating to a strategy that corresponds to

g' Player Y causes at least one clause player C_i to fulfill her reachability objective by reaching t, increasing the payment of C_i to Player Y. Thus, Player Y would deviate, and again P is not an NE. □

4.2   Stability Inefficiency

In this section we study the price of anarchy (PoA) and price of stability (PoS) in ARGTs. Our news is bad, showing that they are both unbounded. Since in all games we have that PoA $\geqslant$ PoS, the PoA analysis is redundant. However, since the PoS analysis is complicated, and includes the network achieving unbounded PoA as a component, we present both. The PoA analysis also completes our analysis of DAG-SYM-ARGTs for which we show that $PoS = 1$.

Theorem 7. *The PoA in ARGTs is unbounded. For every integer k and reward γ, there is a k-player DAG-SYM1-ARGT $\mathcal{N}$ such that $OPT(\mathcal{N}) = k\gamma$, yet the utility of some NE is 0.*

Proof: Consider the DAG-SYM1-ARGT depicted in Figure 6 (left). Let $V_1 = \{s, z, t\}$, $V_2 = \{u\}$, and $V_i = \varnothing$, for all $2 < i \leqslant k$. In the SO P, we have $f_1(s) = u$ and $f_2(u) = t$. Thus, all players fulfill their reachability objectives and $utility(P) = k\gamma$. On the other hand, the profile P' in which $f'_1(s) = z$ and $f'_2(u) = z$ is an NE, as no path to t can be formed if only one player changes her strategy. For P', we have $utility(P') = 0$. □

Theorem 8. *The PoS in ARGTs is unbounded. For every integer k and reward γ, there is a k-player DAG-CT1-ARGT $\mathcal{N}$ such that $OPT(\mathcal{N}) = k\gamma$, yet the utility of the best NE is 0.*

Proof: We first describe a game for $k = 2$ players, then we show how additional players can be added in a way that preserves the analysis. Consider the DAG-CT1-ARGT $\mathcal{N}$ in Figure 6 (right). The ARGT starts with the "no-NE" ARGT from the proof of Theorem 4. While there the target vertex is v, here, once v is reached, the players still have to reach t, thus traversing the ARGT from the proof of Theorem 7. Accordingly, the space of strategies of each player combines space of strategies she has in the two underlying ARGTs.
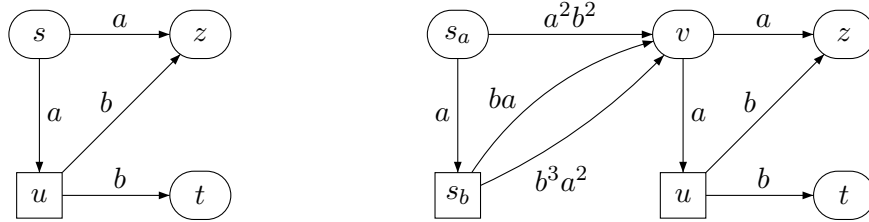

Figure 6: ARGTs with unbounded PoA (left) and unbounded PoS (right).

In more details, Player a has four different strategies: choosing LEFT or RIGHT from s_a, and choosing u or z from v. Similarly, Player b has four strategies: choosing TOP or BOTTOM from s_b, and choosing t or z from u.

Note that t is reachable from both s_a and s_b, thus $OPT(\mathcal{N}) = 2\gamma$. We show that in every NE of $\mathcal{N}$, both players choose to direct tokens to z, resulting in no player fulfilling her reachability objective and total utility 0. First, observe that every profile in which $f_a(v) = f_b(u) = z$ is an NE.

Consider now a profile in which $f_a(v) = z$ and $f_b(u) = t$. Independent of $f_a(s_a)$ and $f_b(s_b)$, by setting $f_a(v) = u$, both players reach t and gain the reachability reward; therefore, Player a has a beneficial move. Similarly, if $f_a(v) = u$ and $f_b(u) = z$, then Player b has a beneficial move. We conclude that there is no NE of $\mathcal{N}$ in which only one of the players directs the tokens to z.

Now, consider a profile in which $f_a(v) = u$ and $f_b(u) = t$, that is, the subpath $\langle v, u, t \rangle$ is valid. In every such profile, as in the ARGT from the proof of Theorem 4, every player has two strategies for her source vertex. Moreover, since the subpath $\langle v, u, t \rangle$ includes exactly one vertex owned by each of the players, the utilities of the player as a function of their strategies in s_a and s_b are identical to those given in Table 1, and thus no NE exists. We conclude that the only NEs are these in which $f_a(v) = f_b(u) = z$, whose total utility is 0.

For $k > 2$ players, the game can be extended by adding dummy players whose reachability objective is $\langle v, \{t\} \rangle$, and who do not own any vertex in the game. These players do not affect the above stability analysis. Their reachability objectives are all fulfilled in the SO, yet none of them is fulfilled in any NE. $\square$

As shown in Theorem 7, the PoA in DAG-SYM1-ARGTs is unlimited. We now show that, on the positive side, every DAG-SYM-ARGTs has a stable optimal profile:

Theorem 9. *The PoS in DAG-SYM-ARGT is* 1.

Proof: Let P be an SO profile of a DAG-SYM-ARGT $\mathcal{N}$. Recall that in SYM-ARGTs, we have that $OPT(\mathcal{N}) \in \{0, k\gamma\}$, and is 0 iff there is no path from s to any target vertex $t \in T$. If $OPT(\mathcal{N}) = 0$, then clearly, $\text{PoS}(\mathcal{N}) = 1$. Otherwise, consider the NE produced by the algorithm presented in the proof of Theorem 5. We claim that this NE has utility $k\gamma$. Recall that $v_1, \ldots, v_n$ is a topological sort of V, and the algorithm keeps, for every $1 \leqslant \ell \leqslant n$, an NE for the ARGT $\mathcal{N}_\ell$ induced by $\{v_\ell, \ldots, v_n\}$, where the reachability objective is $\langle v_\ell, T \rangle$.

In the NE calculation, every player selects for $f_i(v_\ell)$ the successor that maximizes her utility. Since this utility depends on the existence of a path that fulfills the players' reachability objective, every player selects a successor from which such a path exists. In particular, the source vertex selects the first edge of an (s, t)-path in $G_{\mathcal{N}}$, and the NE produced has utility $k\gamma$, implying that the PoS is 1. $\square$

5   The Best-Response Problem

In this section we study the computational complexity of the the BR problem.

Theorem 10. *The BR problem is NP-complete. NP-hardness holds already for* 2-*player SYM1-ARGTs.*

Proof: Membership in NP (for the corresponding decision problem, namely deciding whether there is a strategy f'_i for Player i such that $utility(i, (P_{-i}, f'_i)) \geqslant \kappa$, for a given threshold κ) is easy, as given a witness strategy f'_i, we can calculate $utility(i, (P_{-i}, f'_i))$ in polynomial time.

For hardness in NP we describe a reduction from the *Hamiltonian-path* problem. Given a graph $G = \langle V, E \rangle$ and two vertices s and t in G, the Hamiltonian path problem is to decide whether there exists a simple (s, t)-path in G that visits all $|V|$ vertices. Consider the symmetric ARGT $\mathcal{N} = \langle 2, V_1, V_2, E, \langle s, t \rangle, \gamma \rangle$ in which $V_1 = V$ and $V_2 = \varnothing$, and arbitrary γ. Thus, the revenue of Player 1 in a profile P is the length of the single (s, t)-path in G^P. It is not hard to prove that Player 1 has a strategy leading to profile P in which $utility(1, P) = \gamma + |V| - 1$ if and only if the graph G has a Hamiltonian path from s to t. $\square$

Theorem 11. *The BR problem can be solved in linear time in DAG-SYM-ARGTs.*

Proof: Consider a strategy P. Since there is only one source, the set of edges E^P induced by the strategies in P includes exactly one path from s. W.l.o.g. we assume that for every $v \in T$, we have $succ(v) = \varnothing$.

Let π^P be the (s,t)-path the players use in a profile P. Since all the players use the same path, then for all $v \in V$, it holds that $load(v, P) \in \{0, k\}$. If $t \notin T$ then for every player i, $utility(i, P) = 0$. Assume $t \in T$, then, since the game is symmetric, $profit(i, P) = k \cdot |V_i \cap \pi^P|$, $cost(i, P) = |\pi^P|$, and $utility(i, P) = \gamma + k \cdot |V_i \cap \pi^P| - |\pi^P|$. Recall that $G_{\mathcal{N}}^P = \langle V, E^P \rangle$ is the subgraph of $G_{\mathcal{N}}$ induced by the strategies in P. Let $E' = E^P \backslash E_i^P$ be the set of edges selected by all players but i, and let $E_i = (V_i \times V) \cap E$ be the set of edges leaving vertices in V_i.

We reduce the problem of finding a best-response strategy for Player i to the problem of finding the longest path in a weighted DAG. Consider the DAG $G_i^P = \langle V, E' \cup E_i \rangle$. The edges of G_i^P are weighted as follows: For every edge $e \in E'$, set $w(e) = -1$. For every edge $e \in E_i$, set $w(e) = k - 1$.

Claim 1. *For every $t \in T$, the weighted DAG G_i^P includes a path from s to t whose weight is W, iff Player i can respond to P_{-i} by a strategy f' such that $utility(i, (P_{-i}, f_i')) = \gamma + W$.*

Proof: Recall that the edge set of G_i^P is defined such that for every vertex $v \notin V_i$, exactly one edge leaves v and for every vertex $v \in V_i$, all the edges outgoing from v in $G_{\mathcal{N}}$ are in G_i^P. Let π be an (s,t)-path in G_i^P whose weight is W. For every edge $(v, u) \in \pi \cap E_i$, let $f_i'(v) = u$. That is, Player i respond to P_{-i} in a way that leads all the players to use the path π. In the resulting profile, the revenue of Player i is $k \cdot |\pi \cap E_i|$ and its cost is $|\pi|$. The edge weights are defined such that the weigh of π is $W = (k-1) \cdot |\pi \cap E_i| - 1 \cdot |\pi \cap E'| = k \cdot |\pi \cap E_i| - |\pi|$. Adding the reachability reward, we conclude that Player i has a strategy f_i' such that $utility(i, (P_{-i}, f_i')) = \gamma + W$.

For the other direction, assume that Player i can respond to P_{-i} by a strategy f' such that $utility(i, (P_{-i}, f_i')) = \gamma + W$. Such a strategy corresponds to an (s,t)-path π for some $t \in T$. The utility of Player i consists of the reachability reward γ, and cost and revenue of total value W. Every edge in $\pi \cap E_i$ contributes revenue k, and every edge in $\pi \cap E'$ incurs cost 1. These edge correspond to edges in G_i^P having weights $k - 1$ and -1 respectively. Thus, π is an (s,t)-path in G_i^P whose weight is W.   □

In particular, the claim implies that by solving the longest path problem on G_i^P, and selecting the target vertex $t \in T$ achieving the longest (s,t)-path, we get an optimal strategy for Player i. If no vertex from T is reachable from s in G_i^P, then Player i does not have a unilateral deviation from P that improve her, and all other players', utilities.   □

6   Discussion

We introduced and studied ARGTs – a formal model for reasoning about interactions in which the underlying components have both behavioral and revenue objectives. We showed that this combination of objectives makes the games less stable and more complex to reason about. Thus, the news is mostly bad, yet we were able to point to interesting and practical classes for which the games are stable and reasoning about them can be done efficiently.

Different types of games can model different schemes of interaction among the components. One distinction is between *turn-based* and *concurrent* games. In the first, a single player chooses an action and determines the successor vertex in each step of the interaction. In the second, all

players choose actions in all steps [4]. Different specification formalisms induce different winning conditions. In particular, in the context of nonterminating on-going behaviors, much research is focused on ω-*regular winning conditions* [13]. Finally, games may also involve incomplete information [24] or stochastic transitions or strategies [26]. This work studies a setting in which the winning condition is reachability, the scheduling mechanism is turn-based, strategies are not stochastic, and players have full observability. One clear future work is to extend ARGTs to other schemes of interaction.

Additional interesting extensions of ARGTs consider the players' utility scheme. First, it is natural to allow weighted tolls by adding weights to vertices or edges. The weighs can correspond to actual expenses or prioritize different actions. This extension is of particular interest in the context of rational synthesis. Indeed, there, the turn-based setting typically schedules the players in some round-robin fashion, balancing the profits of the players. Second, it is natural to consider models in which a congestion on an edge affects the cost of players traversing it. ARGTs with weighted edges and cost-sharing are of great relevancy also in the context of *network-formation* – a well-studied game used for the analysis of Internet applications, such as routing in computer networks [5, 28].

Our negative results about stability motivate a study of ARGTs' repair: the ability to achieve stability or lower the PoS by an authority that restricts the strategies of the players. It is not hard to see that all ARGTs can be repaired to achieve the social optimum. Indeed, an aggressive repair may leave available only strategies that constitute an SO. It is interesting to study the trade-offs between the aggressiveness of the repair and the utility of the NE it achieves.

Finally, different types of strategies for the players can capture additional settings. In our ARGTs, a strategy $f_i : V_i \to V$ for Player i describes, for each vertex $v \in V_i$, how the player extends a path that reaches v. The extension is independent of the identity of the player whose token visits v. In a setting with *identified tokens*, a strategy for Player i is $f_i : V_i \times [k] \to V$ that describes, for each vertex $v \in V_i$ and player $j \in [k]$, how the player extends a path of Player j. Our initial results indicate that the identified-token setting is more stable, and reasoning about it is less complex. For example, the SO problem can be solved in PTIME.

References

[1] M. Abadi, L. Lamport, and P. Wolper. Realizable and unrealizable concurrent program specifications. In *Proc. 25th Int. Colloq. on Automata, Languages, and Programming*, 1989.

[2] S. Almagor, G. Avni, and O. Kupferman. Automatic generation of quality specifications. In *Proc. 26th Int. Conf. on Concurrency Theory*, 2015.

[3] S. Almagor, O. Kupferman, and G. Perelli. Synthesis of controllable nash equilibria in quantitative objective game. In *Proc. 27th Int. Joint Conf. on Artificial Intelligence*, pages 35–41, 2018.

[4] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[5] 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.

[6] G. Avni and O. Kupferman. Synthesis from component libraries with costs. In *Proc. 25th Int. Conf. on Concurrency Theory*, 2014.

[7] P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Pure Nash equilibria in concurrent deterministic games. *Logical Methods in Computer Science*, 11(2), 2015.

[8] K. Chatterjee, R. Majumdar, and T. A. Henzinger. Controller synthesis with budget constraints. In *Proc 11th International Workshop on Hybrid Systems: Computation and Control*, 2008.

[9] K. Chatterjee, R. Majumdar, and M. Jurdzinski. On Nash equilibria in stochastic games. In *Proc. 13th Annual Conf. of the European Association for Computer Science Logic*, 2004.

[10] D.L. Dill. *Trace theory for automatic hierarchical verification of speed independent circuits*. MIT Press, 1989.

[11] D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In *Proc. 16th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, 2010.

[12] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111 – 121, 1980.

[13] E. Grädel, W. Thomas, and T. Wilke. *Automata, Logics, and Infinite Games: A Guide to Current Research*, 2002.

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

[15] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[16] O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. *Annals of Mathematics and Artificial Intelligence*, 78(1):3–20, 2016.

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

[18] J.F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, 1950.

[19] N. Nisan and A. Ronen. Algorithmic mechanism design. In *Games and Economic Behavior*, 35:166–196, 2001.

[20] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[21] C. H. Papadimitriou. Algorithms, games, and the internet. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 749–753, 2001.

[22] C.E. Perkins. *Ad Hoc Networking*. Addison-Wesley Professional, 1 edition, 2008.

[23] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th ACM Symp. on Principles of Programming Languages*, pages 179–190, 1989.

[24] J.H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and Systems Science*, 29:274–301, 1984.

[25] E.M. Royer and C-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, 1999.

[26] L.S. Shapley. Stochastic games. In *Proc. of the National Academy of Science*, volume 39, 1953.

[27] Y. Shiloach and Y. Perl. Finding two disjoint paths between two pairs of vertices in a graph. *Journal of the ACM*, 25(1):1–9, 1978.

[28] E. Tardos and T. Wexler. *Algorithmic Game Theory*. Cambridge University Press, 2007. Chapter 19: Network Formation Games and the Potential Function Method.

[29] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1953.