Timed Network Games with Clocks

Guy Avni[1]
IST Austria, Klosterneuburg, Austria
guy.avni@ist.ac.at

Shibashis Guha[2]
Université Libre de Bruxelles, Brussels, Belgium
shibashis.guha@ulb.ac.be

Orna Kupferman[3]
Hebrew University, Jerusalem, Israel
orna@cs.huji.ac.il

Abstract

Network games are widely used as a model for selfish resource-allocation problems. In the classical model, each player selects a path connecting her source and target vertices. The cost of traversing an edge depends on the *load*; namely, number of players that traverse it. Thus, it abstracts the fact that different users may use a resource at different times and for different durations, which plays an important role in determining the costs of the users in reality. For example, when transmitting packets in a communication network, routing traffic in a road network, or processing a task in a production system, actual sharing and congestion of resources crucially depends on time.

In [13], we introduced *timed network games*, which add a time component to network games. Each vertex v in the network is associated with a cost function, mapping the load on v to the price that a player pays for staying in v for one time unit with this load. Each edge in the network is guarded by the time intervals in which it can be traversed, which forces the players to spend time in the vertices. In this work we significantly extend the way time can be referred to in timed network games. In the model we study, the network is equipped with *clocks*, and, as in timed automata, edges are guarded by constraints on the values of the clocks, and their traversal may involve a reset of some clocks. We argue that the stronger model captures many realistic networks. The addition of clocks breaks the techniques we developed in [13] and we develop new techniques in order to show that positive results on classic network games carry over to the stronger timed setting.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory, Theory of computation → Network formation, Theory of computation → Timed and hybrid models

Keywords and phrases Network games, Timed automata, Nash equilibrium, Equilibrium inefficiency

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.23

Related Version A full version of the paper is available at [14], `http://arxiv.org/abs/1808.04882`.

[1] Supported by the Austrian Science Fund (FWF) under grants S11402-N23 (RiSE/SHiNE), Z211-N23 (Wittgenstein Award), and M2369-N33 (Meitner fellowship).
[2] Supported partially by the ARC project "Non-Zero Sum Game Graphs: Applications to Reactive Synthesis and Beyond" (Fédération Wallonie-Bruxelles).
[3] Supported by the European Research Council (FP7/2007-2013) / ERC grant agreement no 278410.

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 23; pp. 23:1–23:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Network games (NGs, for short) [10, 48, 49] constitute a well studied model of non-cooperative games. The game is played among selfish players on a network, which is a directed graph. Each player has a source and a target vertex, and a strategy is a choice of a path that connects these two vertices. The cost a player pays for an edge depends on the *load* on it, namely the number of players that use the edge, and the total cost is the sum of costs of the edges she uses. In *cost-sharing* games, load has a positive effect on cost: each edge has a cost and the players that use it split the cost among them. Then, in *congestion* games[4], load has a negative effect on cost: each edge has a non-decreasing *latency function* that maps the load on the edge to its cost.

One limitation of NGs is that the cost of using a resource abstracts the fact that different users may use the resource at different times and for different durations. This is a real limitation, as time plays an important role in many real-life settings. For example, in a road or a communication system, congestion only affects cars or messages that use a road or a channel simultaneously. We are interested in settings in which congestion affects the quality of service (QoS) or the way a price is shared by entities using a resource at the *same time* (rather than affecting the travel time). For example, discomfort increases in a crowded train (in congestion games) or price is shared by the passengers in a taxi (in cost-sharing games).

The need to address temporal behaviors has attracted a lot of research in theoretical computer science. Formalisms like temporal logic [46] enable the specification of the temporal ordering of events. Its refinement to formalisms like real-time temporal logic [7], interval temporal logic [42], and timed automata (TAs, for short) [6] enables the specification of *real-time* behaviors. Extensions of TAs include *priced timed automata* (PTAs, for short) that assign costs to real-time behaviors. Thus, PTAs are suitable for reasoning about quality of real-time systems. They lack, however, the capability to reason about multi-agent systems in which the players' choices affect the incurred costs.

We study *timed network games* (TNGs, for short) – a new model that adds a time component to NGs. A TNG is played on a *timed-network* in which edges are labeled by *guards* that specify time restrictions on when the edge can be traversed. Similar to NGs, each player has a source and target vertex, but a strategy is now a *timed path* that specifies, in addition to which vertices are traversed, the amount of time that is spent in each vertex. Players pay for staying in vertices, and the cost of staying in a vertex v in a time interval $I \subseteq \mathbb{R}_{\geq 0}$ is affected by the load in v during I. In [13], we studied a class of TNGs that offered a first extension of NGs to a timed variant in which the reference to time is restricted: the guards on the edges refer only to *global* time, i.e., the time that has elapsed since the beginning of the game. In the model in [13], it is impossible to refer to the duration of certain events that occur during the game, for example, it is not possible to express constraints that require staying exactly one time unit in a vertex. Accordingly, we refer to that class as *global TNGs* (GTNGs, for short).

In this work, we significantly extend the way time can be referred to in TNGs. We do this by adding *clocks* that may be reset along the edges, and by allowing the guards on the edges to refer to the values of all clocks. GTNGs can be viewed as a fragment in which there is only a single clock that is never reset. We demonstrate our model in the following example.

[4] The name *congestion games* is sometimes used to refer to games with general latency functions. We find it more appropriate to use it to refer to games with non-decreasing functions.

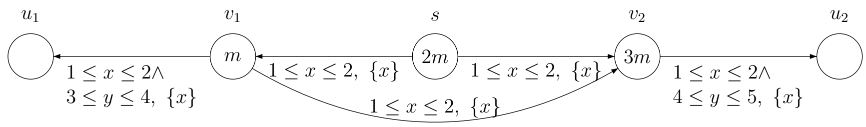

Figure 1 A congestion TNG.

▶ **Example 1.** Consider a setting in which messages are sent through a network of routers. Messages are owned by selfish agents who try to avoid congested routes, where there is a greater chance of loss or corruption. The owners of the messages decide how much time they spend in each router. Using TNGs, we can model constraints on these times, as well as constraints on global events, in particular, arrival time. Note that in some applications, c.f., advertising or security, messages need to patrol the network with a lower bound on their arrival time.

Consider the TNG appearing in Figure 1. The vertices in the TNG model the routers. There are two players that model two agents, each sending a message. The source of both messages is s and the targets are u_1 and u_2, for messages 1 and 2, respectively. The latency functions are described in the vertices, as a function of the load m; e.g., the latency function in v_2 is $\ell_{v_2}(m) = 3m$. Thus, when a single message stays in v_2 the cost for each time unit is 3, and when the two messages visit v_2 simultaneously, the cost for each of them is 6 per unit time. The network has two clocks, x and y. Clock x is reset in each transition and thus is used to impose restrictions on the time that can be spent in each router: since all transitions can be taken when $1 \le x \le 2$, a message stays between 1 and 2 time units in a router. Clock y is never reset, thus it keeps track of the global time. The guards on clock y guarantee that message 1 reaches its destination by time 4 but not before time 3 and message 2 reaches its destination by time 5 but not before time 4.

Suppose the first agent chooses the timed path $(s, 2), (v_1, 1), u_1$, thus message 1 stays in s for two time units and in v_1 for one time unit before reaching its destination u_1. Suppose the second agent chooses the path $(s, 2), (v_1, 2), (v_2, 1), u_2$. Note that crossing an edge is instantaneous. Since both messages stay in the same vertices during the intervals $I_1 = [0, 2]$ and $I_2 = [2, 3]$, the load in the corresponding vertices is 2. During interval I_1, each of the agents pays $|I_1| \cdot \ell_s(2) = 2 \cdot 4$ and during I_2, each pays $|I_2| \cdot \ell_{v_1}(2) = 1 \cdot 2$. Message 2 stays in v_1 alone during the interval $[3, 4]$ and in v_2 during the interval $[4, 5]$, for which it pays 1 and 3, respectively. The total costs are thus 10 and 14.

Before we elaborate on our contribution, let us survey relevant works, namely, extensions of NGs with temporal aspects and extensions of timed-automata to games. Extensions of NGs that involve reasoning about time mostly study a cost model in which the players try to minimize the time of arrival at their destinations (c.f., [36, 39, 47, 45]), where, for example, congestion affects the duration of crossing an edge. These works are different from ours since we consider a QoS cost model. An exception is [36], which studies the QoS costs. A key difference in the models is that there, time is discrete and the players have finitely many strategies. Thus, reductions to classical resource allocation games is straightforward while for TNGs it is not possible, as we elaborate below. Games on timed automata were first studied in [11] in which an algorithm to solve timed games with timed reachability objective was given. The work was later generalized and improved [4, 20, 35, 23]. Average timed games, games with parity objectives, mean-payoff games and energy games have also been studied in the context of timed automata [2, 37, 27, 21, 34]. All the timed games above are two-player zero-sum ongoing games. Prices are fixed and there is no notion of load. Also, the questions studied on these games concern their decidability, namely finding winners and strategies for them. TNGs are not zero-sum games, so winning strategies do not exist. Instead, the problems we study here concern rationality and stability.

The first question that arises in the context of non-zero-sum games is the existence of *stable outcomes*. In the context of NGs, the most prominent stability concept is that of a (pure) *Nash equilibrium* (NE, for short) [43] – a profile such that no player can decrease her cost by unilaterally deviating from her current strategy.[5] Decentralized decision-making may lead to solutions that are sub-optimal for the society as a whole. The standard measures to quantify the inefficiency incurred due to selfish behavior is the *price of stability* (PoS) [10] and the *price of anarchy* (PoA) [38]. In both measures we compare against the *social optimum* (SO, for short), namely a profile that minimizes the sum of costs of all players. The PoS (PoA, respectively) is the best-case (worst-case) inefficiency of an NE; that is, the ratio between the cost of a best (worst) NE and the SO.

The picture of stability and equilibrium inefficiency for standard NGs is well understood. Every NG has an NE, and in fact these games are *potential games* [48], which have the following stronger property: a *best response sequence* is a sequence of profiles $P_1, P_2, \ldots$ such that, for $i \geq 1$, the profile P_{i+1} is obtained from P_i by letting some player deviate and decrease her personal cost. In finite potential games, every best-response sequence converges to an NE. For k-player cost-sharing NGs, the PoS and PoA are $\log k$ and k, respectively [10]. For congestion games with affine cost functions, PoS ≈ 1.577 [29, 1] and PoA $= \frac{5}{2}$ [30].

In [13], we showed that these positive results carry over to GTNGs. A key technical feature of GTNGs is that since guards refer to global time, it is easy to find an upper bound T on the time by which all players reach their destinations. Proving existence of NE follows from a reduction to NGs, using a zone-like structure [5, 18]. The introduction of clocks with resets breaks the direct reduction to NGs and questions the existence of a bound by which the players arrive at their destinations.[6] To see the difficulty in finding such a bound, consider, for example, a cost-sharing game in which all players, on their paths to their targets, need to stay for one time unit in a "gateway" vertex v that costs 1 (see details in Section 6). Assume also that, for $1 \leq i \leq k$, Player i can only reach v in times that are multiples of p_i, for relatively prime numbers $p_1, \ldots, p_k$. The SO is obtained when all players synchronize their visits to v, and such a synchronization forces them to wait till time $p_1 \cdot \ldots \cdot p_k$, which is exponential in the TNG.

The lack of an upper bound on the global time in TNGs demonstrates that we need a different approach to obtain positive results for general TNGs. We show that TNGs are guaranteed to have an NE. Our proof uses a combination of techniques from real-time models and resource allocation games. Recall that a PTA assigns a price to a timed word. We are able to reduce the *best-response* and the *social-optimum* problems to and from the problem of finding cheapest runs in PTAs [19], showing that the problems are PSPACE-complete. Next, we show that TNGs are potential games. Note that since players have uncountably many strategies, the fact that TNGs are potential games does not immediately imply existence of an NE, as a best-response sequence may not be finite. We show that there is a best-response sequence that terminates in an NE. For this, we first need to show the existence of an integral best-response, which is obtained from the reduction to PTAs. Finally, given a TNG, we find a time T such that there exists an NE in which all players reach their destination by time T.

Due to lack of space, some of the proofs appear in the full version [14].

[5] Throughout this paper, we consider *pure* strategies, as is the case for the vast literature on NGs.
[6] In the full version we show that even with an upper bound on time, a reduction from TNGs to NGs is not likely.

2 Preliminaries

2.1 Resource allocation games and network games

For $k \in \mathbb{N}$, let $[k] = \{1, \ldots, k\}$. A *resource allocation game* (RAG, for short) is $R = \langle k, E, \{\Sigma_i\}_{i \in [k]}, \{\ell_e\}_{e \in E} \rangle$, where $k \in \mathbb{N}$ is the number of players; E is a set of resources; for $i \in [k]$, the set strategies of Player i is $\Sigma_i \subseteq 2^E$; and, for $e \in E$, the *latency function* $\ell_e : [k] \to \mathbb{Q}_{\geq 0}$ maps a load on e to its cost under this load. A *profile* is a choice of a strategy for each player. The set of profiles of R is $\mathit{profiles}(R) = \Sigma_1 \times \ldots \times \Sigma_k$. For $e \in E$, we define the *load* on e in a profile $P = \langle \sigma_1, \ldots, \sigma_k \rangle$, denoted $\mathit{load}_P(e)$, as the number of players using e in P, thus $\mathit{load}_P(e) = |\{i \in [k] : e \in \sigma_i\}|$. The cost a player pays in profile P, denoted $\mathit{cost}_i(P)$, depends on the choices of the other players. We define $\mathit{cost}_i(P) = \sum_{e \in \sigma_i} \ell_e(\mathit{load}_P(e))$.

Network games (NGs, for short) can be viewed as a special case of RAGs where strategies are succinctly represented by means of paths in graphs. An NG is $\mathcal{N} = \langle k, V, E, \{\langle s_i, u_i \rangle\}_{i \in [k]}, \{\ell_e\}_{e \in E} \rangle$, where $\langle V, E \rangle$ is a directed graph; for $i \in [k]$, the vertices s_i and u_i are the *source* and *target* vertices of Player i; and the latency functions are as in RAGs. The set of strategies for Player i is the set of simple paths from s_i to u_i in $\mathcal{N}$. Thus, in NGs, the resources are the edges in the graph.

We distinguish between two types of latency functions. In *cost-sharing games*, the players that visit a vertex share its cost equally. Formally, every $e \in E$ has a cost $c_e \in \mathbb{Q}_{\geq 0}$ and its latency function is $\ell_e(l) = \frac{c_e}{l}$. Note that these latency functions are decreasing, thus the load has a positive effect on the cost. In contrast, in *congestion games*, the cost functions are non-decreasing and so the load has a negative effect on the cost. Typically, the latency functions are restricted to simple functions such as linear latency functions, polynomials, and so forth.

2.2 Timed networks and timed network games

A *clock* is a variable that gets values from $\mathbb{R}_{\geq 0}$ and whose value increases as time elapses. A *reset* of a clock x assigns value 0 to x. A *guard* over a set C of clocks is a conjunction of *clock constraints* of the form $x \sim m$, for $x \in C$, $\sim \in \{\leq, =, \geq\}$, and $m \in \mathbb{N}$. Note that we disallow guards that use the operators $<$ and $>$ (see Remark 4). A guard of the form $\bigwedge_{x \in C} x \geq 0$ is called *true*. The set of guards over C is denoted $\Phi(C)$. A *clock valuation* is an assignment $\kappa : C \to \mathbb{R}_{\geq 0}$. A clock valuation κ *satisfies* a guard g, denoted $\kappa \models g$, if the expression obtained from g by replacing each clock $x \in C$ with the value $\kappa(x)$ is valid.

A *timed network* is a tuple $\mathcal{A} = \langle C, V, E \rangle$, where C is a set of clocks, V is a set of vertices, and $E \subseteq V \times \Phi(C) \times 2^C \times V$ is a set of directed edges in which each edge e is associated with a guard $g \in \Phi(C)$ that should be satisfied when e is traversed and a set $R \subseteq C$ of clocks that are reset along the traversal of e.

When traversing a path in a timed network, time is spent in vertices, and edges are traversed instantaneously. Accordingly, a *timed path* in $\mathcal{A}$ is a sequence $\eta = \langle \tau_1, e_1 \rangle, \ldots, \langle \tau_n, e_n \rangle \in (\mathbb{R}_{\geq 0} \times E)^*$, describing edges that the path traverses along with their traversal times. The timed path η is *legal* if the edges are successive and the guards associated with them are satisfied. Formally, there is a sequence $\langle v_0, t_0 \rangle, \ldots, \langle v_{n-1}, t_{n-1} \rangle, v_n \in (V \times \mathbb{R}_{\geq 0})^* \cdot V$, describing the vertices that η visits and the time spent in these vertices, such that for every $1 \leq j \leq n$, the following hold: (1) $t_{j-1} = \tau_j - \tau_{j-1}$, with $\tau_0 = 0$, (2) there is $g_j \in \Phi(C)$ and $R_j \subseteq C$, such that $e_j = \langle v_{j-1}, g_j, R_j, v_j \rangle$, (3) there is a clock valuation κ_j that describes the values of the clocks before the incoming edge to vertex v_j is traversed. Thus, $\kappa_1(x) = t_0$, for all $x \in C$, and for $1 < j \leq n$, we distinguish between clocks that are reset when e_{j-1}

is traversed and clocks that are not reset: for $x \in R_{j-1}$, we define $\kappa_j(x) = t_{j-1}$, and for $x \in (C \setminus R_{j-1})$, we define $\kappa_j(x) = \kappa_{j-1}(x) + t_{j-1}$, and (4) for every $1 \leq j \leq n$, we have that $\kappa_j \models g_j$. We sometimes refer to η also as the sequence $\langle v_0, t_0\rangle, \ldots, \langle v_{n-1}, t_{n-1}\rangle, v_n$.

Consider a finite set $T \subseteq \mathbb{R}_{\geq 0}$ of time points. We say that a timed path η is a *T-path* if all edges in η are taken at times in T. Formally, for all $1 \leq j \leq n$, we have that $\tau_j \in T$. We refer to the time at which η *ends* as the time τ_n at which the destination is reached. We say that η is *integral* if $T \subseteq \mathbb{N}$.

A *timed network game* (TNG, for short) extends an NG by imposing constraints on the times at which edges may be traversed. Formally, $\mathcal{T} = \langle k, C, V, E, \{\ell_v\}_{v\in V}, \langle s_i, u_i\rangle_{i\in[k]}\rangle$ includes a set C of clocks, and $\langle C, V, E\rangle$ is a timed network. Recall that while traversing a path in a timed network, time is spent in vertices. Accordingly, the latency functions now apply to vertices, thus $\ell_v : [k] \to \mathbb{Q}_{\geq 0}$ maps a load on vertex v to its cost under this load. Traversing an edge is instantaneous and is free of charge. A strategy for Player i, for $i \in [k]$, is then a legal timed path from s_i to u_i. We assume all players have at least one strategy.

▶ Remark. A possible extension of TNGs is to allow costs on edges. Since edges are traversed instantaneously, these costs would not be affected by load. Such an extension does not affect our results and we leave it out for sake of simplicity. Another possible extension is allowing strict time guards, which we discuss in Remark 4.

The cost Player i pays in profile P, denoted $cost_i(P)$, depends on the vertices in her timed path, the time spent on them, and the load during the visits. In order to define the cost formally, we need some definitions. For a finite set $T \subseteq \mathbb{R}_{\geq 0}$ of time points, we say that a timed path is a T-strategy if it is a T-path. Then, a profile P is a T-profile if it consists only of T-strategies. Let $t_{max} = \max(T)$. For $t \in T$ such that $t < t_{max}$, let $next_T(t)$ be the minimal time point in T that is strictly larger than t. We partition the interval $[0, t_{\max}]$ into a set Υ of sub-intervals $[m, next_T(m)]$ for every $m \in (T \cup \{0\}) \setminus \{t_{max}\}$. We refer to the sub-intervals in Υ as *periods*. Suppose T is the minimal set such that P is a T-profile. Note that Υ is the coarsest partition of $[0, t_{\max}]$ into periods such that no player crosses an edge within a period in Υ. We denote this partition by Υ_P.

For a player $i \in [k]$ and a period $\gamma \in \Upsilon_P$, let $visits_P(i, \gamma)$ be the vertex that Player i visits during period γ. That is, if $\pi_i = \langle v^i_0, t^i_0\rangle, \ldots, \langle v^i_{n_i-1}, t^i_{n_i-1}\rangle, v^i_{n_i}$ is a legal timed path that is a strategy for Player i and $\gamma = [m_1, m_2]$, then $visits_P(i, \gamma)$ is the vertex v^i_j for the index $1 \leq j < n_i$ such that $\tau^i_j \leq m_1 \leq m_2 \leq \tau^i_{j+1}$, and $visits_P(i, \gamma)$ is the vertex v^i_0 if $0 = m_1 \leq m_2 \leq \tau^i_1$. Note that since P is a T-profile, for each period $\gamma \in \Upsilon_P$, the number of players that stay in each vertex v during γ is fixed. Let $load_P(v, \gamma)$ denote this number. Formally $load_P(v, \gamma) = |\{i : visits_P(i, \gamma) = v\}|$. Finally, for a period $\gamma = [m_1, m_2]$, let $|\gamma| = m_2 - m_1$ be the *duration* of γ. Suppose Player i's path ends at time τ^i. Let $\Upsilon^i_P \subseteq \Upsilon_P$ denote the periods that end by time τ_i.

Recall that the latency function $\ell_v : [k] \longrightarrow \mathbb{Q}_{\geq 0}$ maps the number of players that simultaneously visit vertex v to the price that each of them pays per time unit. If $visits_P(i, \gamma) = v$, then the cost of Player i in P, over the period γ is $cost_{\gamma,i}(P) = \ell_v(load_P(v, \gamma)) \cdot |\gamma|$. We define $cost_i(P) = \sum_{\gamma\in\Upsilon^i_P} cost_{\gamma,i}(P)$. The cost of the profile P, denoted $cost(P)$, is the total cost incurred by all the players, i.e., $cost(P) = \sum_{i=1}^{k} cost_i(P)$.

A T-strategy is called an *integral strategy* when $T \subseteq \mathbb{N}$, and similarly for *integral* profile.

A profile $P = \langle \pi_1, \ldots, \pi_k\rangle$ is said to *end* by time τ if for each $i \in [k]$, the strategy π_i ends by time τ. Consider a TNG $\mathcal{T}$ that has a cycle such that a clock x of $\mathcal{T}$ is reset on the cycle. It is not difficult to see that this may lead to $\mathcal{T}$ having infinitely many integral profiles that end by different times. A TNG $\mathcal{T}$ is called *global* if it has a single clock x that is never reset. We use GTNG to indicate that a TNG is global.

As in RAGs, we distinguish between cost-sharing TNGs that have cost-sharing latency functions and congestion TNGs in which the latency functions are non-decreasing.

2.3 Stability and efficiency

Consider a game G. For a profile P and a strategy π of player $i \in [k]$, let $P[i \leftarrow \pi]$ denote the profile obtained from P by replacing the strategy of Player i in P by π. A profile P is said to be a *(pure) Nash equilibrium* (NE) if none of the players in $[k]$ can benefit from a unilateral deviation from her strategy in P to another strategy. Formally, for every Player i and every strategy π for Player i, it holds that $cost_i(P[i \leftarrow \pi]) \geq cost_i(P)$.

A *social optimum* (SO) of a game G is a profile that attains the infimum cost over all profiles. We denote by $SO(G)$ the cost of an SO profile; i.e., $SO(G) = \inf_{P \in profiles(G)} cost(P)$. It is well known that decentralized decision-making may lead to sub-optimal solutions from the point of view of the society as a whole. We quantify the inefficiency incurred due to self-interested behavior by the *price of anarchy* (PoA) [38, 44] and *price of stability* (PoS) [10] measures. The PoA is the worst-case inefficiency of a Nash equilibrium, while the PoS measures the best-case inefficiency of a Nash equilibrium. Note that unlike resource allocation games in which the set of profiles is finite, in TNGs there can be uncountably many NEs, so both PoS and PoA need to be defined using infimum/supremum rather than min/max. Formally,

▶ **Definition 2.** Let $\mathcal{G}$ be a family of games, and let $G \in \mathcal{G}$ be a game in $\mathcal{G}$. Let $\Gamma(\mathrm{G})$ be the set of Nash equilibria of the game G. Assume that $\Gamma(\mathrm{G}) \neq \emptyset$.

- The *price of anarchy* of G is $PoA(G) = \sup_{P \in \Gamma(\mathrm{G})} cost(P)/SO(G)$. 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 $PoS(G) = \inf_{P \in \Gamma(\mathrm{G})} cost(P)/SO(G)$. The *price of stability* of the family of games $\mathcal{G}$ is $PoS(\mathcal{G}) = sup_{G \in \mathcal{G}} PoS(G)$.

3 The Best-Response and the Social-Optimum Problems

Consider a TNG $\mathcal{T} = \langle k, C, V, E, \{\ell_v\}_{v \in V}, \langle s_i, u_i \rangle_{i \in [k]} \rangle$. In the *best-response problem* (BR problem, for short), we ask how a player reacts to a choice of strategies of the other players. Formally, let $\pi_1, \ldots, \pi_{k-1}$ be a choice of integral[7] strategies for Players $1, \ldots, k-1$ in $\mathcal{T}$. We look for a strategy π_k that minimizes $cost_k(\langle \pi_1, \ldots, \pi_k \rangle)$. The choice of allowing Player k to react is arbitrary and is done for convenience of notation. In the *social optimum problem* (SOPT problem, for short), we seek a profile that maximizes the social welfare, or in other words, minimizes the sum of players' costs.

In this section we describe *priced timed automata* (PTAs, for short) [9, 17] and show that while they are different from TNGs both in terms of the model and the questions asked on it, they offer a useful framework for reasoning about TNGs. In particular, we solve the BR and SOPT problems by reductions to problems about PTAs.

3.1 From TNGs to priced timed automata

A PTA [9, 17] is $\mathcal{P} = \langle C, V, E, \{r_v\}_{v \in V} \rangle$, where $\langle C, V, E \rangle$ is a timed network and $r_v \in \mathbb{Q}_{\geq 0}$ is the *rate* of vertex $v \in V$. Intuitively, the rate r_v specifies the cost of staying in v for a duration of one time unit. Thus, a timed path $\eta = \langle v_0, t_0 \rangle, \ldots, \langle v_n, t_n \rangle, v_{n+1}$ in a PTA has a

[7] We choose integral strategies since strategies with irrational times cannot be represented as part of the input; for strategies that use rational times, the best response problem can be solved with little modification in the proof of Theorem 4.

price, denoted $price(\eta)$, which is $\sum_{0 \leq j \leq n} r_v \cdot t_v$. The size of $\mathcal{P}$ is $|V| + |E|$ plus the number of bits needed in the binary encoding of the numbers appearing in guards and rates in $\mathcal{P}$. [8]

Consider a PTA $\mathcal{P}$ and two vertices s and u. Let $paths(s, u)$ be the set of timed paths from s to u. We are interested in cheapest timed paths in $paths(s, u)$. A priori, there is no reason to assume that the minimal price is attained, thus we are interested in the optimal price, denoted $opt(s, u)$, which we define to be $\inf\{price(\eta) : \eta \in paths(s, u)\}$. The corresponding decision problem, called the *cost optimal reachability problem* (COR, for short) takes in addition a threshold μ, and the goal is to decide whether $opt(s, t) \leq \mu$. Recall that we do not allow the guards to use the operators $<$ and $>$.

▶ **Theorem 3.** *[19, 32] The COR problem is PSPACE-complete for PTAs with two or more clocks. Moreover, the optimal price is attained by an integral path, i.e., there is an integral path $\eta \in paths(s, u)$ with $price(\eta) = opt(s, u)$.*

In Sections 3.2 and 3.3 below, we reduce problems on TNGs to problems on PTAs. The reductions allow us to obtain properties on strategies and profiles in TNGs using results on PTAs, which we later use in combination with techniques for NGs in order to solve problems on TNGs.

3.2 The best-response problem

▶ **Theorem 4.** *Consider a TNG $\mathcal{T}$ with n clocks and integral strategies $\pi_1, \ldots, \pi_{k-1}$ for Players $1, \ldots, k-1$. There is a PTA $\mathcal{P}$ with $n+1$ clocks and two vertices v and u such that there is a one-to-one cost-preserving correspondence between strategies for Player k in $\mathcal{T}$ and timed paths from v to u: for every strategy π_k in $\mathcal{T}$ and its corresponding path η in $\mathcal{P}$, we have $cost_k(\langle \pi_1, \ldots, \pi_k \rangle) = price(\eta)$.*

Proof. We describe the intuition of the reduction and the details can be found in the full version. Consider a TNG $\mathcal{T} = \langle k, V, E, C, \{\ell_v\}_{v \in V}, \langle s_i, u_i \rangle_{i \in [k]} \rangle$, where $C = \{x_1, \ldots, x_m\}$. Let $Q = \langle \pi_1, \ldots, \pi_{k-1} \rangle$ be a choice of timed paths for Players $1, \ldots, k-1$. Note that Q can be seen as a profile in a game that is obtained from $\mathcal{T}$ by removing Player k, and we use the definitions for profiles on Q in the expected manner. Let $T \subseteq \mathbb{Q}$ be the minimal set of time points for which all the strategies in Q are T-strategies. Consider two consecutive time points $a, b \in T$, i.e., there is no $c \in T$ with $a < c < b$. Then, there are players that cross edges at times a and b, and no player crosses an edge at time points in the interval (a, b). Moreover, let t_{max} be the latest time in T, then t_{max} is the latest time at which a player reaches her destination. Let Υ_Q be a partition of $[0, t_{max}]$ according to T. We obtain Υ'_Q from Υ_Q by adding the interval $[t_{\max}, \infty)$.

A key observation is that the load on all the vertices is unchanged during every interval in Υ'_Q. For a vertex $v \in V$ and $\delta \in \Upsilon_Q$, the cost Player k pays per unit time for using v in the interval δ is $\ell_v(load_Q(v, \delta) + 1)$. On the other hand, since all $k-1$ players reach their destination by time t_{max}, the load on v after t_{max} is 0, and the cost Player k pays for using it then is $\ell_v(1)$.

The PTA $\mathcal{P}$ that we construct has $|\Upsilon'_Q|$ copies of $\mathcal{T}$, thus its vertices are $V \times \Upsilon'_Q$. Let $\delta_0 = [0, b] \in \Upsilon'_Q$ be the first interval. We consider paths from the vertex $v = \langle s_k, \delta_0 \rangle$, which is the copy of Player k's source in the first copy of $\mathcal{T}$, to a target u, which is a new vertex we add and whose only incoming edges are from vertices of the form $\langle u_k, \delta \rangle$, namely,

[8] In general, PTAs have rates on transitions and strict time guards, which we do not need here.

the copies of the target vertex u_k of Player k. We construct $\mathcal{P}$ such that each such path η from v to u in $\mathcal{P}$ corresponds to a legal strategy π_k for Player k in $\mathcal{T}$, and such that $cost_k(\langle \pi_1, \ldots, \pi_{k-1}, \pi_k \rangle) = price(\eta)$. The main difference between the copies are the vertices' costs, which depend on the load as in the above. We refer to the n clocks in $\mathcal{T}$ as *local clocks*. In each copy of $\mathcal{P}$, we use the local clocks and their guards in $\mathcal{T}$ as well as an additional *global* clock that is never reset to keep track of global time. Let $\delta = [a, b] \in \Upsilon_Q$ and $\delta' = [b, c] \in \Upsilon'_Q$ be the following interval. Let $\mathcal{T}_\delta$ and $\mathcal{T}_{\delta'}$ be the copies of $\mathcal{T}$ that corresponds to the respective intervals. The local clocks guarantee that a path in $\mathcal{T}_\delta$ is a legal path in $\mathcal{T}$. The global clock allows us to make sure that (1) proceeding from $\mathcal{T}_\delta$ to $\mathcal{T}_{\delta'}$ can only occur precisely at time b, and (2) proceeding from $\langle u_k, \delta \rangle$ in $\mathcal{T}_\delta$ to the target u can only occur at a time in the interval δ. ◀

We conclude with the computational complexity of the BR problem. The decision-problem variant gets as input a TNG $\mathcal{T}$, integral strategies $\pi_1, \ldots, \pi_{k-1}$ for Players $1, \ldots,$ $k-1$, and a value μ, and the goal is to decide whether Player k has a strategy π_k such that $cost_k(\langle \pi_1, \ldots, \pi_k \rangle) \leq \mu$. Theorem 4 implies a reduction from the BR problem to the COR problem and a reduction in the other direction is easy since PTAs can be seen as TNGs with a single player. For one-clock instances, we show that the BR problem is NP-hard by a reduction from the subset-sum problem. Note the contrast with the COR problem in one-clock instances, which is NLOGSPACE-complete [41]. The proof of the following theorem can be found in the full version.

▶ **Theorem 5.** *The BR problem is PSPACE-complete for TNGs with two or more clocks. For one-clock cost-sharing and congestion TNGs it is in PSPACE and NP-hard.*

Proof. We reduce the BR problem to and from the COR problem, which is PSPACE-complete for PTAs with at least two clocks [19]. A PTA can be seen as a one-player TNG, thus the BR problem for TNGs with two or more clocks is PSPACE-hard. For the upper bound, given a TNG $\mathcal{T}$, strategies $Q = \langle \pi_1, \ldots, \pi_{k-1} \rangle$ for Players $1, \ldots, k-1$, and a threshold μ, we construct a PTA $\mathcal{P}$ as in the proof of Theorem 4. Note that the size of $\mathcal{P}$ is polynomial in the size of the input and that $\mathcal{P}$ has one more clock than $\mathcal{T}$. An optimal path in $\mathcal{P}$ is a best response for Player k, and such a path can be found in PSPACE.

The final case to consider is TNGs with one clock. We show that the BR problem is NP-hard for such instances using a reduction from the *subset-sum* problem. The input to that problem is a set of natural numbers $A = \{a_1, \ldots, a_n\}$ and $\mu \in \mathbb{N}$, and the goal is to decide whether there is a subset of A whose sum is μ. We start with the cost-sharing case. The game we construct is a two-player game on a network that is depicted in Figure 2. Player 2 has a unique strategy that visits vertex v_{n+1} in the time interval $[\mu, \mu + 1]$. A Player 1 strategy π corresponds to a choice of a subset of A. Player 1's source is v_1 and her target is u_2. The vertex v_{n+1} is the only vertex that has a cost, which is 1, and the other vertices cost 0. For $1 \leq i \leq n$, Player 1 needs to choose between staying in vertex v_i for a duration of a_i time units, and exiting the vertex through the top edge, or staying 0 time units, and exiting the vertex through the bottom edge. Finally, she must stay in v_{n+1} for exactly one time unit. The cost Player 1 pays for v_{n+1} depends on the load. If she stays there in the global time interval $[\mu, \mu + 1]$, she pays 1/2, and otherwise she pays 1. Thus, Player 1 has a strategy with which she pays 1/2 iff there is a subset of A whose sum is μ, and we are done.

The reduction for congestion games is similar. Recall that in congestion games, the cost increases with the load, thus a player would aim at using a vertex together with as few other players as possible. The network is the same as the one used above. Instead of two

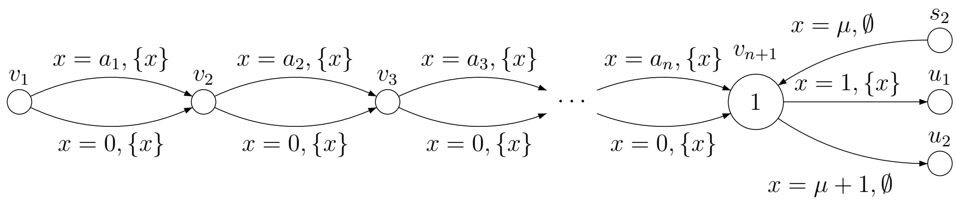

Figure 2 NP-hardness proof of best response problem in one clock TNG.

players, we use three players, where Players 2 and 3 have a unique strategy each. Player 2 must stay in v_{n+1} in the time interval $[0, \mu]$ and Player 3 must stay there during the interval $[\mu + 1, \sum_{1 \leq i \leq n} a_i]$. As in the above, Player 1 has a strategy in which she uses v_{n+1} alone in the time interval $[\mu, \mu + 1]$ iff there is a subset of A whose sum is μ. ◀

3.3 The social-optimum problem

▶ **Theorem 6.** *Consider a TNG $\mathcal{T} = \langle k, C, V, E, \{\ell_v\}_{v \in V}, \langle s_i, u_i \rangle_{i \in [k]} \rangle$. There is a PTA $\mathcal{P}$ with $k \cdot |C|$ clocks, $|V|^k$ vertices, and two vertices $\bar{s}$ and $\bar{u}$ such that there is a one-to-one cost-preserving correspondence between profiles in $\mathcal{T}$ and paths from $\bar{s}$ to $\bar{u}$; namely, for a profile P and its corresponding path η_P, we have $cost(P) = price(\eta_P)$.*

Proof. We describe the intuition of the construction and the details can be found in the full version. Recall that the social optimum is obtained when the players do not act selfishly, rather they cooperate to find the profile that minimizes their sum of costs. Let $\mathcal{T} = \langle k, C, V, E, \{\ell_v\}_{v \in V}, \langle s_i, u_i \rangle_{i \in [k]} \rangle$. We construct a PTA $\mathcal{P}$ by taking k copies of $\mathcal{T}$. For $i \in [k]$, the i-th copy is used to keep track of the timed path that Player i uses. We need k copies of the clocks of $\mathcal{T}$ to guarantee that the individual paths are legal. Recall that the players' goal is to minimize their total cost, thus for each point in time, the price they pay in $\mathcal{P}$ is the sum of their individual costs in $\mathcal{T}$. More formally, consider a vertex $\bar{v} = \langle v_1, \ldots, v_k \rangle$ in $\mathcal{P}$ and let $S_{\bar{v}} \subseteq V$ be the set of vertices that appear in $\bar{v}$. Then, the load on a vertex $v \in S_{\bar{v}}$ in $\bar{v}$ is $load_{\bar{v}}(v) = |\{i : v_i = v\}|$, and the rate of $\bar{v}$ is $\sum_{v \in S_{\bar{v}}} \ell_v(load_{\bar{v}}(v))$. The cost of the social optimum in $\mathcal{T}$ coincides with the price of the optimal timed path in $\mathcal{P}$ from $\langle s_1, \ldots, s_k \rangle$ to the vertex $\langle u_1, \ldots, u_k \rangle$, i.e., the vertices that respectively correspond to the sources and targets of all players. ◀

We turn to study the complexity of the SOPT problem. In the decision-problem variant, we are given a TNG $\mathcal{T}$ and a value μ and the goal is to decide whether there is a profile P in $\mathcal{T}$ with $cost(P) \leq \mu$. Theorem 6 implies a reduction from the SOPT problem to the COR problem, and, as in the BR problem, the other direction is trivial. For one-clock instances, we use the same NP-hardness proof as in the BR problem. The details can be found in the full version.

▶ **Theorem 7.** *The SOPT problem is PSPACE-complete for at least two clocks and it is NP-hard for TNGs with one clock.*

4 Existence of a Nash Equilibrium

The first question that arises in the context of games is the existence of an NE. In [13], we showed that GTNGs are guaranteed to have an NE by reducing every GTNG to an NG. We strengthen the result by showing that every TNG has an NE.

In order to prove existence, we combine techniques from NGs and use the reduction to PTA in Theorem 4. A standard method for finding an NE is showing that a *best-response sequence* converges: Starting from some profile $P = \langle \pi_1, \ldots, \pi_k \rangle$, one searches for a player that can benefit from a unilateral deviation. If no such player exists, then P is an NE and we are done. Otherwise, let π'_i be a beneficial deviation for Player i, i.e., $cost_i(P) > cost_i(P[i \leftarrow \pi'_i])$. The profile $P[i \leftarrow \pi'_i]$ is considered next and the above procedure repeats.

A *potential function* for a game is a function Ψ that maps profiles to costs, such that the following holds: for every profile $P = \langle \pi_1, \ldots, \pi_k \rangle$, $i \in [k]$, and strategy π'_i for Player i, we have $\Psi(P) - \Psi(P[i \leftarrow \pi'_i]) = cost_i(P) - cost_i(P[i \leftarrow \pi'_i])$, i.e., the change in potential equals the change in cost of the deviating player. A game is a *potential game* if it has a potential function. In a potential game with finitely many profiles, since the potential of every profile is non-negative and in every step of a best-response sequence the potential strictly decreases, every best-response sequence terminates in an NE. It is well-known that RAGs are potential games [48] and since they are finite, this implies that an NE always exists.

The idea of our proof is as follows. First, we show that TNGs are potential games, which does not imply existence of NE since TNGs have infinitely many profiles. Then, we focus on a specific best-response sequence that starts from an integral profile and allows the players to deviate only to integral strategies. Finally, we define *normalized TNGs* and show how to *normalize* a TNG in a way that preserves existence of NE. For normalized TNGs, we show that the potential reduces at least by 1 along each step in the best-response sequence, thus it converges to an NE.

▶ **Theorem 8.** *TNGs are potential games.*

Proof. Consider a TNG $\mathcal{T} = \langle k, C, V, E, \{\ell_v\}_{v \in V}, \langle s_i, u_i \rangle_{i \in [k]} \rangle$. Recall that for a profile P, the set of intervals that are used in P is Υ_P. We define a potential function Ψ that is an adaptation of Rosenthal's potential function [48] to TNGs. We decompose the definition of Ψ into smaller components, which will be helpful later on. For every $\gamma \in \Upsilon_P$ and $v \in V$, we define $\Psi_{\gamma,v}(P) = \sum_{j=1}^{load_P(v,\gamma)} |\gamma| \cdot \ell_v(j)$, that is, we take the sum of $|\gamma| \cdot \ell_v(j)$ for all $j \in [load_P(v, \gamma)]$. We define $\Psi_\gamma(P) = \sum_{v \in V} \Psi_{\gamma,v}(P)$, and we define $\Psi(P) = \sum_{\gamma \in \Upsilon_P} \Psi_\gamma(P)$. Let for some $i \in [k]$, we have P' to be a profile that is obtained by an unilateral deviation of Player i to a strictly beneficial strategy π'_i from her current strategy in P, that is $P' = P[i \leftarrow \pi']$ for some $i \in [k]$. In the full version, we show that $\Psi(P) - \Psi(P') = cost_i(P) - cost_i(P')$. ◀

Recall from Theorem 4, that given a TNG, a profile P and an index i, we find the best response of Player i by constructing a PTA. If P is an integral profile, from Theorem 3, we have that the best response of Player i also leads to an integer profile. Thus we have the following lemma.

▶ **Lemma 9.** *Consider a TNG $\mathcal{T}$ and an integral profile P. For $i \in [k]$, if Player i has a beneficial deviation from P, then she has an integral beneficial deviation.*

The last ingredient of the proof gives a lower bound for the difference in cost that is achieved in a beneficial integral deviation for some player $i \in [k]$, which in turn bounds the change in potential.

We first need to introduce a *normalized* form of TNGs. Recall that the latency function in a TNG $\mathcal{T}$ is of the form $\ell_v : [k] \to \mathbb{Q}_{\geq 0}$. In a normalized TNG all the latency functions map loads to natural numbers, thus for every vertex $v \in V$, we have $\ell_v : [k] \to \mathbb{N}$. Constructing a normalized TNG from a TNG is easy. Let L be the least common multiple of the denominators of the elements in the set $\{\ell_v(l) : v \in V \text{ and } l \in [k]\}$. For every latency function ℓ_v and every $l \in [k]$, we construct a new latency function ℓ'_v by $\ell'_v(l) = \ell_v(l) \cdot L$.

Consider a TNG $\mathcal{T}$ and let $\mathcal{T}'$ be the normalized TNG that is constructed from $\mathcal{T}$. It is not hard to see that for every profile P and $i \in [k]$, we have $cost_i(P)$ in $\mathcal{T}'$ is $L \cdot cost_i(P)$ in $\mathcal{T}$. We can thus restrict attention to normalized TNGs as the existence of NE and convergence of best-response sequence in $\mathcal{T}'$ implies the same properties in $\mathcal{T}$. In order to show that a best-response sequence converges in TNGs, we bound the change of potential in each best-response step by observing that in normalized TNGs, the cost a player pays is an integer.

▶ **Lemma 10.** *Let $\mathcal{T}$ be a normalized TNG, $P = \langle \pi_1, \ldots, \pi_k \rangle$ be an integral profile in $\mathcal{T}$, and π'_i be a beneficial integral deviation for Player i, for some $i \in [k]$. Then, $cost_i(P) - cost_i(P[i \leftarrow \pi'_i]) \geq 1$.*

We can now prove the main result in this section.

▶ **Theorem 11.** *Every TNG has an integral NE. Moreover, from an integral profile P, there is a best-response sequence that converges to an integral NE.*

Proof. Lemma 9 allows us to restrict attention to integral deviations. Indeed, consider an integral profile P. Lemma 9 implies that if no player has a beneficial integral deviation from P, then P is an NE in $\mathcal{T}$. We start best-response sequence from some integral profile P_I and allow the players to deviate with integral strategies only. Consider a profile P and let P' be a profile that is obtained from P by a deviation of Player i. Recall from Theorem 8 that $cost_i(P) - cost_i(P') = \Psi(P) - \Psi(P')$. Lemma 10 implies that when the deviation is beneficial, we have $\Psi(P) - \Psi(P') \geq 1$. Since the potential is non-negative, the best-response sequence above converges within $\Psi(P_I)$ steps. ◀

▶ Remark. A TNG that allows $<$ and $>$ operators on the guards is not guaranteed to have an NE. Indeed, in a PTA, which can be seen as a one-player TNG, strict guards imply that an optimal timed path may not be achieved. In turn, this means that an NE does not exist. To overcome this issue, we use ϵ-NE, for $\epsilon > 0$; an ϵ-deviation is one that improves the payoff of a player at least by ϵ, and an ϵ-NE is a profile in which no player has a ϵ-deviation. Our techniques can be adapted to show that ϵ-NE exist in TNGs with strict guards. The proof uses the results of [19] that show that an ϵ-optimal timed path exists in PTAs. The proof technique for existence of NE in TNGs with non-strict guards can then be adapted to the strict-guard case.

5 Equilibrium Inefficiency

In this section we address the problem of measuring the degradation in social welfare due to selfish behavior, which is measured by the PoS and PoA measures. We show that the upper bounds from RAGs on these two measures apply to TNGs. For cost-sharing TNGs, we show that the PoS and PoA are at most $\log k$ and k, respectively, as it is in cost-sharing RAGs. Matching lower bounds were given in [13] already for GTNGs. For congestion TNGs with affine latency functions, we show that the PoS and PoA are $1 + \sqrt{(3)}/3 \approx 1.577$ and $\frac{5}{2}$, respectively, as it is in congestion RAGs. Again, a matching lower bound for PoA is shown in [13] for GTNGs, and a matching lower bound for the PoS remains open. Let $\mathcal{F}$ denote a family of latency functions and $\mathcal{F}$-TNGs and $\mathcal{F}$-RAGs denote, respectively, the family of TNGs and RAGs that use latency functions from this family.

▶ **Theorem 12.** *Consider a family of latency functions $\mathcal{F}$. We have $PoS(\mathcal{F}\text{-}TNGs) \leq PoS(\mathcal{F}\text{-}RAGs)$ and $PoA(\mathcal{F}\text{-}TNGs) \leq PoA(\mathcal{F}\text{-}RAGs)$. In particular, the PoS and PoA for cost-sharing TNGs with k players is at most $\log(k)$ and k, respectively, and for congestion TNGs with affine latency functions it is at most roughly 1.577 and $\frac{5}{2}$ respectively.*

Proof. We prove for PoS in cost-sharing games and the other proofs are similar. Consider a TNG $\mathcal{T}$ and let $N^1, N^2, \ldots$ be a sequence of NEs whose cost tends to $c^* = \inf_{P \in \Gamma(\mathcal{T})} cost(P)$. Let O be a social optimum profile in $\mathcal{T}$, which exists due to Theorem 6. Thus, $PoS(\mathcal{T}) = \lim_{j \to \infty} cost(N^j)/\ cost(O)$. We show that each element in the sequence is bounded above by PoS(cost-sharing RAGs), which implies that $PoS(\mathcal{T}) \leq PoS$(cost-sharing RAGs), and hence PoS(cost-sharing TNGs) $\leq PoS$(cost-sharing RAGs). In the full version, for each $j \geq 1$, we construct a RAG $\mathcal{R}_j$ that has $PoS(\mathcal{R}_j) = cost(N^j)/cost(O)$, and since $\mathcal{R}_j$ is a cost-sharing RAG, we have $PoS(\mathcal{R}_j) \leq PoS$(cost- sharing RAGs), and we are done. ◀

6 Time Bounds

Recall that due to resets of clocks, the time by which a profile ends can be potentially unbounded. It is interesting to know, given a TNG, whether there are time bounds within which some interesting profiles like an NE and an SO are guaranteed to exist. Earlier we showed that every TNG is guaranteed to have an integral NE (Theorem 11) and an integral SO (Theorem 6). In this section we give bounds on the time by which such profiles end. That is, given a TNG $\mathcal{T}$, we find $t_{NE}(\mathcal{T}), T_{SO}(\mathcal{T}) \in \mathbb{Q}_{\geq 0}$ such that an integral NE N and an integral SO O exist in $\mathcal{T}$ in which the players reach their destinations by time $t_{NE}(\mathcal{T})$ and $T_{SO}(\mathcal{T})$ respectively.

We start by showing a time bound on an optimal timed path in a PTA, and then proceed to TNGs.

▶ **Lemma 13.** *Consider a PTA $\mathcal{P} = \langle C, V, E, \{r_v\}_{v \in V}\rangle$, and let χ be the largest constant appearing in the guards on the edges of $\mathcal{P}$. Then, for every $s, u \in V$, there is an integral optimal timed path from s to u that ends by time $|V| \cdot (\chi + 2)^{|C|}$.*

Proof. Consider an optimal integral timed path η in $\mathcal{P}$ that ends in the earliest time and includes no loop that is traversed instantaneously. Let $v_0, \ldots, v_n$ be the sequence of vertices that η traverses, and, for $0 \leq i < n$, let κ_i be the clock valuation before exiting the vertex v_i. Since η is integral, κ_i assigns integral values to clocks. Note that since the largest constant appearing in a guard in $\mathcal{P}$ is χ, the guards in $\mathcal{P}$ cannot differentiate between clock values greater than χ. We abstract away such values and define the *restriction* of a clock valuation κ_i to be $\beta_i : C \to (\{0\} \cup [\chi] \cup \{\top\})$ by setting, for $x \in C$, the value $\beta_i(x) = \kappa_i(x)$, when $\kappa_i(x) \leq \chi$, and $\beta_i(x) = \top$, when $\kappa_i(x) > \chi$. Assume towards contradiction that η ends after time $|V| \cdot (\chi + 2)^{|C|}$. Then, there are $0 \leq i < j < n$ such that $\langle v_i, \beta_i\rangle = \langle v_j, \beta_j\rangle$. Let $\eta = \eta_1 \cdot \eta_2 \cdot \eta_3$ be a partition of η such that η_2 is the sub-path between the i-th and j-th indices. Consider the path $\eta' = \eta'_1 \cdot \eta'_3$ that is obtained from η by removing the sub-path η_2. First, note that η' is a legal path. Indeed, the restrictions of the clock valuations in η_1 and η_3 match these in η'_1 and η'_3, that is, $\eta' = \eta_1 \cdot \eta_3$. Second, since we assume that traversing the loop η_2 is not instantaneous, we know that η' ends before η. Moreover, since the rates in $\mathcal{P}$ are non-negative, we have $price(\eta') \leq price(\eta)$, and we reach a contradiction to the fact that η is an optimal timed path that ends earliest. ◀

▶ **Theorem 14.** *For a k-player TNG $\mathcal{T}$ with a set V of vertices and a set C of clocks, there exists an SO that ends by time $\mathcal{O}(|V|^k \cdot \chi^{k|C|})$, where χ is the maximum constant appearing in $\mathcal{T}$. For every $k \geq 1$, there is a k-player (cost-sharing and congestion) TNG $\mathcal{T}_k$ such that $\mathcal{T}_k$ has $\mathcal{O}(k)$ states, the boundaries in the guards in $\mathcal{T}_k$ are bounded by $\mathcal{O}(k \log k)$, and any SO in $\mathcal{T}_k$ requires time $2^{\Omega(k)}$.*

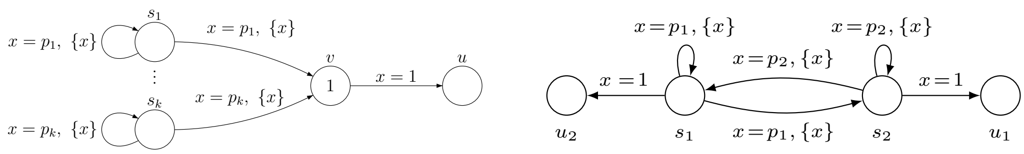

Figure 3 The time required for the SO is not polynomial.

Proof. We start with the upper bound. Consider a TNG $\mathcal{T}$ with a set V of vertices and a set C of clocks. By Theorem 6, we can construct a PTA $\mathcal{P}$ with $|V|^k$ vertices and $k|C|$ clocks such that a social optimum of $\mathcal{T}$ is an optimal timed path in $\mathcal{P}$. Applying Lemma 13, we are done.

We turn to the lower bounds. We show that for every $k \geq 1$, there is a k-player (cost-sharing and congestion) TNG $\mathcal{T}_k$ such that $\mathcal{T}_k$ has $\mathcal{O}(k)$ states, the boundaries in the guards in $\mathcal{T}_k$ are bounded by $\mathcal{O}(k \log k)$, and any SO in $\mathcal{T}_k$ requires time $2^{\Omega(k)}$.

Consider the k-player cost-sharing TNG appearing on the left of Figure 3. Let $p_1, \ldots, p_k$ be relatively prime (e.g., the the first k prime numbers). All the vertices in the TNG have cost 0, except for v, which has some positive cost function. Each player i has to spend one time unit in v in her path from s_i to u. In an SO, all k players spend this one time unit simultaneously, which forces them all to reach v at time $\prod_{1 \leq i \leq k} p_i$. Since the i-th prime number is $O(i \log i)$ and the product of the first i prime numbers is $2^{\Omega(i)}$, we are done. We note that we could define the TNG also with no free vertices, that is vertics with 0 cost, by setting the cost in v to be much higher than those in the source vertices.

For congestion games, the example is more complicated. We start with the case of two players. Consider the congestion TNG appearing on the right of Figure 3. Assume that p_1 and p_2 are relatively prime, $r_{s_1}(1) = r_{s_2}(1) = 0$, and $r_{s_1}(2) = r_{s_2}(2) = 1$. In the SO, the two players avoid each other in their paths from s_i to u_i, and the way to do so is to wait $p_1 \cdot p_2$ time units before the edge from s_i to s_{3-i} is traversed. In the full version,we generalize this example to k players. Again, we could define the TNG with no free vertices. ◀

We proceed to derive a time bound for the existence of an NE. For a TNG $\mathcal{T}$, let $L_{\mathcal{T}} \in \mathbb{N}$ be the smallest number such that multiplying the latency functions by $L_{\mathcal{T}}$ results in a normalized TNG. Recall the $SO(\mathcal{T})$ is the cost of a social optimum in $\mathcal{T}$.

▶ **Theorem 15.** *Consider a TNG $\mathcal{T}$ with k players, played on a timed network $\langle V, E, C\rangle$, and let χ be the maximum constant appearing in a guard. Then, there is an NE in $\mathcal{T}$ that ends by time $\mathcal{O}(\varphi \cdot |V| \cdot \chi^{|C|} + |V|^k \cdot \chi^{k|C|})$, where $\varphi = L_{\mathcal{T}} \cdot SO(\mathcal{T})$ for congestion TNGs and $\varphi = L_{\mathcal{T}} \cdot \log(k) \cdot SO(\mathcal{T})$ for cost-sharing TNGs.*

Proof. Recall the proof of Theorem 11 that shows that every TNG has an integral NE: we choose an initial integral profile P and perform integral best-response moves until an NE is reached. The number of iterations is bounded by the potential $\Psi(P)$ of P. We start the best-response sequence from a social-optimum profile O that ends earliest. By Theorem 14, there is such a profile that ends by time $\mathcal{O}(|V|^k \cdot \chi^{k|C|})$. Let $\varphi = L_{\mathcal{T}} \cdot SO(\mathcal{T})$ in the case of congestion TNGs and $\varphi = L_{\mathcal{T}} \cdot (\ln(k) + 1) \cdot SO(\mathcal{T})$ in the case of cost-sharing TNGs. It is not hard to show that $\Psi(O) \leq \varphi$.

Next, we bound the time that is added in a best-response step. We recall the construction in Theorem 4 of the PTA $\mathcal{P}$ for finding a best-response move. Consider a TNG $\mathcal{T}$ and a profile of strategies P, where, w.l.o.g., we look for a best-response for Player k. Suppose the strategies of Players $1, \ldots, k-1$ take transitions at times $\tau_1, \ldots, \tau_n$. We construct a PTA $\mathcal{P}$

with $n+1$ copies of $\mathcal{T}$. For $1 \leq i \leq n+1$, an optimal path in $\mathcal{P}$ starts in the first copy and moves from copy i to copy $(i+1)$ at time τ_i. We use the additional "global" clock to enforce these transitions. A key observation is that in the last copy, this additional clock is never used. Thus, the largest constant in a guard in the last copy coincides with χ, the largest constant appearing in $\mathcal{T}$. Let η be an optimal path in $\mathcal{P}$ and π_k the corresponding strategy for Player k. We distinguish between two cases. If η does not enter the last copy of $\mathcal{P}$, then it ends before time τ_n, namely the latest time at which a player reaches her destination. Then, the profile $P[k \leftarrow \pi_k]$ ends no later than P. In the second case, the path η ends in the last copy of $\mathcal{P}$. We view the last copy of $\mathcal{P}$ as a PTA. By Lemma 13, the time at which η ends is within $|V| \cdot (\chi+2)^{|C|}$ since its entrance into the copy, which is τ_n. Then, $P[i \leftarrow \pi_k]$ ends at most $|V| \cdot (\chi+2)^{|C|}$ time units after P. To conclude, the best-response sequence terminates in an NE that ends by time $\mathcal{O}(\varphi \cdot |V| \cdot (\chi+2)^{|C|} + |V|^k \cdot \chi^{k|C|})$. ◀

7 Discussion and Future Work

The model of TNGs studied in this paper extends the model of GTNGs introduced in [13] by adding clocks. From a practical point of view, the addition of clocks makes TNGs significantly more expressive than GTNGs and enables them to model the behavior of many systems that cannot be modeled using GTNGs. From a theoretical point of view, the analysis of TNGs poses different and difficult technical challenges. In the case of GTNGs, a main tool for obtaining positive results is a reduction between GTNGs and NGs. Here, in order to obtain positive results we need to combine techniques from NGs and PTAs.

We left several open problems. In Theorem 11, we describe a method for finding an integral NE through a sequence of BR moves. We leave open the complexity of finding an NE in TNGs. For the upper bound, we conjecture that there is a PSPACE algorithm for the problem. For the lower bound, we would need to find an appropriate complexity class of search problems and show hardness for that class. For example, PLS [31], which lies "close" to P, and includes the problem of finding an NE in NGs, consists of search problems in which a local search, e.g., a BR sequence, terminates. Unlike NGs, where a BR can be found in polynomial time, in TNGs, the problem is PSPACE-complete. To the best of our knowledge, complexity classes for search problems that are higher than PLS were not studied. Further we show that the BR and SO problems for one-clock TNGs is in PSPACE and is NP-hard, leaving open the tight complexity.

This work belongs to a line of works that transfer concepts and ideas between the areas of formal verification and algorithmic game theory: logics for specifying multi-agent systems [8, 26], studies of equilibria in games related to synthesis and repair problems [25, 24, 33, 3], and of non-zero-sum games in formal verification [28, 22]. This line of work also includes efficient reasoning about NGs with huge networks [40, 12], an extension of NGs to objectives that are richer than reachability [16], and NGs in which the players select their paths dynamically [15]. For future work, we plan to apply the real-time behavior of TNGs to these last two concepts; namely, TNGs in which the players' objectives are given as a specification that is more general than simple reachability or TNGs in which the players reveal their choice of timed path in steps, bringing TNGs closer to the timed games of [11, 2].

References

1 S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann. Exact price of anarchy for polynomial congestion games. *SIAM J. Comput.*, 40(5):1211–1233, 2011.

2 L. Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In *Proc. 14th Int. Conf. on Concurrency Theory*, pages 142–156, 2003.

3 S. Almagor, G. Avni, and O. Kupferman. Repairing multi-player games. In *Proc. 26th Int. Conf. on Concurrency Theory*, volume 42 of *LIPIcs*, pages 325–339, 2015.

4 R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *Proc. 31st Int. Colloq. on Automata, Languages, and Programming*, pages 122–133, 2004.

5 R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

6 R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–236, 1994.

7 R. Alur and T. Henzinger. Real-time logics: complexity and expressiveness. In *Proc. 5th IEEE Symp. on Logic in Computer Science*, pages 390–401, 1990.

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

9 R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. *Theoretical Computer Science*, 318(3):297–322, 2004.

10 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.

11 E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *Proc 2nd International Workshop on Hybrid Systems: Computation and Control*, pages 19–30, London, UK, UK, 1999. Springer-Verlag.

12 G. Avni, S. Guha, and O. Kupferman. An abstraction-refinement methodology for reasoning about network games. In *Proc. 33rd Int. Joint Conf. on Artificial Intelligence*, pages 70–76, 2017.

13 G. Avni, S. Guha, and O. Kupferman. Timed network games. In *42nd Int. Symp. on Mathematical Foundations of Computer Science*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2017.

14 G. Avni, S. Guha, and O. Kupferman. Timed network games with clocks. *CoRR*, abs/1808.04882, 2018. `arXiv:1808.04882`.

15 G. Avni, T.A. Henzinger, and O. Kupferman. Dynamic resource allocation games. In *Proc. 9th International Symposium on Algorithmic Game Theory*, volume 9928 of *Lecture Notes in Computer Science*, pages 153–166, 2016.

16 G. Avni, O. Kupferman, and T. Tamir. Network-formation games with regular objectives. *Information and Computation*, 251:165–178, 2016.

17 G. Behrmann, A. A. Fehnker, T. Hune, K.G. Larsen, P. Pettersson, J. Romijn, and F. W. Vaandrager. Minimum-cost reachability for priced timed automata. In *Proc 4th International Workshop on Hybrid Systems: Computation and Control*, pages 147–161, London, UK, 2001. Springer-Verlag.

18 J. Bengtsson and W.Yi. Timed automata: Semantics, algorithms and tools. In *Lectures on Concurrency and Petri Nets, Advances in Petri Nets*, pages 87–124, 2003.

19 P. Bouyer, T. Brihaye, V. Bruyère, and J-F. Raskin. On the optimal reachability problem of weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, oct 2007.

20 P. Bouyer, F. Cassez, E. Fleury, and K.G. Larsen. Optimal strategies in priced timed game automata. In *Proc. 24th Conf. on Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, pages 148–160, 2004.

21 R. Brenguier, F. Cassez, and J-F Raskin. Energy and mean-payoff timed games. In *Proc 17th International Workshop on Hybrid Systems: Computation and Control*, pages 283–292, 2014.

22 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.

23 T. Brihaye, G. Geeraerts, S. N. Krishna, L. Manasa, B. Monmege, and A. Trivedi. Adding negative prices to priced timed games. In *Proc. 25th Int. Conf. on Concurrency Theory*, pages 560–575, 2014.

24 K. Chatterjee. Nash equilibrium for upward-closed objectives. In *Proc. 15th Annual Conf. of the European Association for Computer Science Logic*, volume 4207 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2006.

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

26 K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. In *Proc. 18th Int. Conf. on Concurrency Theory*, pages 59–73, 2007.

27 K. Chatterjee, T. A. Henzinger, and V. S. Prabhu. Timed parity games: Complexity and robustness. *Logical Methods in Computer Science*, 7(4), 2011.

28 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*, volume 3210 of *Lecture Notes in Computer Science*, pages 26–40. Springer, 2004.

29 G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *ESA*, pages 59–70, 2005.

30 G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proc. 37th ACM Symp. on Theory of Computing*, pages 67–73, 2005.

31 A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *Proc. 36th ACM Symp. on Theory of Computing*, pages 604–612, 2004.

32 J. Fearnley and M. Jurdziński. Reachability in two-clock timed automata is pspace-complete. In *Proc. 40th Int. Colloq. on Automata, Languages, and Programming*, pages 212–223, Berlin, Heidelberg, 2013. Springer-Verlag.

33 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*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010.

34 S. Guha, M. Jurdzinski, S. N. Krishna, and A. Trivedi. Mean-payoff games on timed automata. In *Proc. 36th Conf. on Foundations of Software Technology and Theoretical Computer Science*, pages 44:1–44:14, 2016.

35 T. Dueholm Hansen, R. Ibsen-Jensen, and P. Bro Miltersen. A faster algorithm for solving one-clock priced timed games. In *Proc. 24th Int. Conf. on Concurrency Theory*, pages 531–545, 2013.

36 M. Hoefer, V. S. Mirrokni, H. Röglin, and S. Teng. Competitive routing over time. *Theor. Comput. Sci.*, 412(39):5420–5432, 2011. `doi:10.1016/j.tcs.2011.05.055`.

37 M. Jurdzinski and A. Trivedi. Average-time games. In *Proc. 28th Conf. on Foundations of Software Technology and Theoretical Computer Science*, pages 340–351, 2008.

38 E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009.

39 E. Koutsoupias and K. Papakonstantinopoulou. Contention issues in congestion games. In *Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming - Volume Part II*, ICALP'12, pages 623–635. Springer-Verlag, 2012.

40 O. Kupferman and T. Tamir. Hierarchical network formation games. In *Proc. 23rd Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 10205 of *Lecture Notes in Computer Science*, pages 229–246. Springer, 2017.

41 F. Laroussinie, N. Markey, and P. Schnoebelen. Model checking timed automata with one or two clocks. In *Proc. 15th Int. Conf. on Concurrency Theory*, pages 387–401, 2004.

42 B.C. Moszkowski and Z. Manna. Reasoning in interval temporal logic. In *Logics of Programs*, volume 164 of *Lecture Notes in Computer Science*, pages 371–382. Springer, 1983.

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

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

45 M. Penn, M. Polukarov, and M. Tennenholtz. Random order congestion games. *Mathematics of Operations Research*, 34(3):706–725, 2009.

46 A. Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13:45–60, 1981.

47 K. Ronald and S. Martin. Nash equilibria and the price of anarchy for flows over time. *Theoretical Computer Science*, 49(1):71–97, 2011.

48 R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

49 T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.