Parity Games on Temporal Graphs

Pete Austin(✉), Sougata Bose, and Patrick Totzke

University of Liverpool, Liverpool, UK
sgpausti@liverpool.ac.uk

Abstract. Temporal graphs are a popular modelling mechanism for dynamic complex systems that extend ordinary graphs with discrete time. Simply put, time progresses one unit per step and the availability of edges can change with time.
We consider the complexity of solving ω-regular games played on temporal graphs where the edge availability is ultimately periodic and fixed a priori.
We show that solving parity games on temporal graphs is decidable in PSPACE, only assuming the edge predicate itself is in PSPACE. A matching lower bound already holds for what we call *punctual* reachability games on static graphs, where one player wants to reach the target at a given, binary encoded, point in time. We further study syntactic restrictions that imply more efficient procedures. In particular, if the edge predicate is in P and is monotonically increasing for one player and decreasing for the other, then the complexity of solving games is only polynomially increased compared to static graphs.

Keywords: Temporal graphs · Reachability Games · Complexity · Timed automata

1 Introduction

Temporal graphs are graphs where the edge relation changes over time. They are often presented as a sequence $G_0, G_1, \ldots$ of graphs over the same set of vertices. We find it convenient to define them as pairs $G = (V, E)$ consisting of a set V of vertices and associated edge availability predicate $E : V^2 \to 2^{\mathbb{N}}$ that determines at which integral times a directed edge can be traversed. This model has been used to analyse dynamic networks and distributed systems in dynamic topologies, such as gossiping and information dissemination [36,24]. There is also a large body of work that considers temporal generalisations of various graph-theoretic notions and properties [32,14,10]. Related algorithmic questions include graph colouring [30], exploration [12], travelling salesman [33], maximum matching [29], and vertex-cover [2]. The edge relation is often deliberately left unspecified and sometimes only assumed to satisfy some weak assumptions about connectedness, frequency, or fairness to study the worst or average cases in uncontrollable environments. Depending on the application, one distinguishes between "online" questions, where the edge availability is revealed stepwise, as

N. Kobayashi and J. Worrell (Eds.): FoSSaCS 2024, LNCS 14574, pp. 79–98, 2024.
https://doi.org/10.1007/978-3-031-57228-9_5

opposed to the “offline” variant where all is given in advance. We refer to [17,31] for overviews of temporal graph theory and its applications.

Two player zero-sum verification games on directed graphs play a central role in formal verification, specifically the reactive synthesis approach [34]. Here, a controllable system and an antagonistic environment are modeled as a game in which two opposing players jointly move a token through a graph. States are either owned by Player 1 (the system) or Player 2 (the environment), and the owner of the current state picks a valid successor. Such a play is won by Player 1 if, and only if, the constructed path satisfies a predetermined *winning condition* that models the desired correctness specification. The winning condition is often given either in a temporal logic such as Linear Temporal Logic (LTL) [35], or directly as ω-automaton whose language is the set of infinite paths considered winning for Player 1. The core algorithmic problem is solving games: to determine which player has a strategy to force a win, and if so, how.

Determining the complexity of solving games on static graphs has a long history and continues to be an active area of research. We refer to [1,13] for introductions on the topic and recall here only that solving reachability games, where Player 1 aims to eventually reach a designated target state, is complete for polynomial time. The precise complexity of solving parity games is a long-standing open question. It is known to be in $\mathsf{UP}\cap\mathsf{coUP}$ [22], and so in particular in NP and coNP, and recent advances have led to quasi-polynomial time algorithms [6,23,26,9,25].

Related Work. Periodic temporal graphs were first studied by Floccchini, Mans, and Santoro in [14], where they show polynomial bounds on the length of explorations (paths covering all vertices). Recently, De Carufel, Flocchini, Santoro, and Simard [10] study Cops & Robber games on periodic temporal graphs. They provide an algorithm for solving one-cop games that is only quadratic in the number of vertices and linear in the period.

Games on temporal graphs with maximal age, or period of some absolute value K given in binary are games on exponentially succinctly presented arenas. Unfolding them up to time K yields an ordinary game on the exponential sized graph which allows to transfer upper bounds, that are not necessarily optimal. In a similar vein, Avni, Ghorpade, and Guha [4] have recently introduced types of games on exponentially succinct arenas called pawn games. Similar to our results, their findings provide improved PSPACE upper bounds for reachability games.

Parity games on temporal graphs are closely related to timed-parity games, which are played on the configuration graphs of timed automata [3]. However, the time in temporal graphs is discrete as opposed to the continuous time semantics in timed automata. Solving timed parity games is complete for EXP[28,8] and the lower bound already holds for reachability games on timed automata with only two clocks [21]. Unfortunately, a direct translation of (games on) temporal graphs to equivalent timed automata games requires at least two clocks: one to hold the global time used to check the edge predicate and one to ensure that time progresses one unit per step.

Contributions. We study the complexity of solving parity games on temporal graphs. As a central variant of independent interest are what we call *punctual* reachability games, that are played on a static graph and player wants to reach a target vertex at a given binary encoded time. We show that solving such games is already hard for PSPACE, which provides a lower bound for all temporal graph games we consider.

As our second, and main result, we show how to solve parity games on (ultimately) periodic temporal graphs. The difficulty to overcome here is that the period may be exponential in the number of vertices and thus a naïvely solving the game on the unfolding only yields algorithms in exponential space. Our approach relies on the existence of polynomially sized summaries that can be verified in PSPACE using punctual reachability games.

We then provide a sufficient syntactic restriction that avoids an increased complexity for game solving. In particular, if the edge predicate is in polynomial time and is monotonically increasing for one player and decreasing for the other, then the cost of solving reachability or parity games on temporal graphs increases only polynomially in the number of vertices compared to the cost of solving these games on static graphs.

None of our upper bounds rely on any particular representation of the edge predicate. Instead, we only require that the representation ensures that checking membership (if an edge is traversable at a given time) has suitably low complexity. That is, our approach to solve parity games only requires that the edge predicate is in PSPACE, and polynomial-time verifiable edge predicates suffice to derive P-time upper bounds for monotone reachability games. These conditions are met for example if the edge predicate is defined as semilinear set given as an explicit union of linear sets (NP in general and in P for singleton sets of periods), or by restricted Presburger formulae: the quantifier-free fragment is in P, the existential fragment is in NP but remains in P if the number of variables is bounded [37]. See for instance [15] and contained references.

The rest of the paper is structured as follows. We recall the necessary notations in Section 2 and then discuss reachability games in Section 3. Section 4 presents the main construction for solving parity games and finally, in Section 5, we discuss improved upper bounds for monotone temporal graphs.

2 Preliminaries

Definition 1 (Temporal Graphs). *A temporal graph $G = (V, E)$ is a directed graph where V are vertices and $E : V^2 \to 2^{\mathbb{N}}$ is the edge availability relation that maps each pair of vertices to the set of times at which the respective directed edge can be traversed. If $i \in E(s,t)$ we call t an* i-successor *of s and write $s \xrightarrow{i} t$.*

The horizon *of a temporal graph is $h(G) = \sup_{s,t \in V}(E(s,t))$, the largest finite time at which any edge is available, or ∞ if no such finite time exists. A temporal graph is* finite *if $h(G) \in \mathbb{N}$ i.e., every edge eventually disappears forever. A temporal graph is* periodic *with period $K \in \mathbb{N}$ if for all nodes $s, t \in V$ it holds that $E(s,t) = E(s,t) + K \cdot \mathbb{N}$. We call G* static *if it has period* 1.

Naturally, one can unfold a temporal graph into its *expansion* up to some time $T \in \mathbb{N} \cup \{\infty\}$, which is the graph with nodes $V \times \{0, 1, \ldots, T\}$ and directed edges $(s, i) \rightarrow (t, i + 1)$ iff $i \in E(s, t)$.

In order for algorithmic questions to be interesting, we assume that temporal graphs are given in a format that is more succinct than the expansion up to their horizon or period. We only require that the representation ensures that checking if an edge is traversable at a given time can be done reasonably efficiently.

We will henceforth use formulae in the existential fragment of Presburger arithmetic, the first-order theory over natural numbers with equality and addition. That is, the ∃PA formula $\Phi_{s,t}(x)$ with one free variable x represents the set of times at which an edge from s to t is available as $E(s, t) = \{n \mid \Phi_{s,t}(n) \equiv \mathit{true}\}$. We use common syntactic sugar including inequality and multiplication with (binary encoded) constants. For instance, $\Phi_{s,t}(x) \overset{\text{def}}{=} 5 \leq x \wedge x \leq 10$ means the edge is available at times $\{5, 6, 7, 8, 9, 10\}$; and $\Phi_{s,t}(x) \overset{\text{def}}{=} \exists y.(x = y \cdot 7) \wedge \neg(x \leq 100)$ means multiples of 7 greater than 100.

Definition 2 (Parity Games). *A* parity game *is a zero-sum game played by two opposing players on a directed graph. Formally, the game is given by a game graph $G = (V, E)$, a partitioning $V = V_1 \uplus V_2$ of vertices into those owned by Player 1 and Player 2 respectively, and a colouring $col : V \rightarrow C$ of vertices into a finite set $C \subsetneq \mathbb{N}$ of colours.*

The game starts with a token on an initial vertex $s_0 \in V$ and proceeds in turns where in round i, the owner of the vertex occupied by the token moves it to some successor. This way both players jointly agree on an infinite path $\rho = s_0 s_1 \ldots$ called a play. *A play is winning for Player 1 if* $\max\{c \mid \forall i \exists j.\, col(s_j) = c\}$*, the maximum colour seen infinitely often, is even.*

A strategy *for Player i is a recipe for how to move. Formally, it is a function $\sigma_i : V^* V_i \rightarrow V$ from finite paths ending in a vertex s in V_i to some successor. We call σ* positional *if $\sigma(\pi s) = \sigma(\pi' s)$ for any two prefixes $\pi, \pi' \in V^*$. A strategy is* winning from vertex *s if Player i wins every play that starts in vertex s and during which all decisions are made according to σ.*

We call a vertex s winning for Player i if there exists a winning strategy from s, and call the subset of all such vertices the *winning region* for that player. Parity games enjoy the following property (See [13, Theorem 15] for details).

Proposition 1. *Parity games are uniformly positionally determined: For every game $(V = V_1 \uplus V_2, E, col)$ there is a pair σ_1, σ_2 of positional strategies so that σ_i is winning for Player i from every vertex in the winning region of Player i.*

A *temporal parity game* is a parity game played on the infinite expansion of a temporal graph $G = (V, E)$, where the ownership and colouring of vertices are given with respect to the underlying directed graph $V = V_1 \uplus V_2$ and $col : V \rightarrow C$. The ownership and colouring are lifted to the expansion so that vertices in $V_i \times \mathbb{N}$ are owned by Player i and vertex (s, n) has colour $col(s)$.

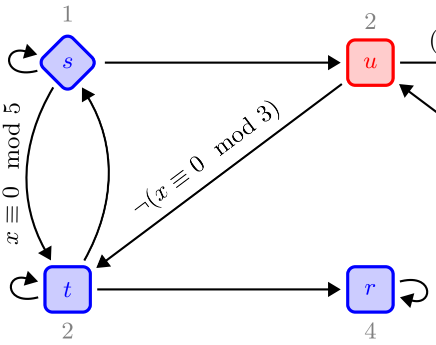

Fig. 1: An example of a temporal parity game. Player 1 controls the diamond vertices $V_1 = \{s, v\}$ and Player 2 controls square vertices $V_2 = \{r, t, u, w\}$. Edge labels are Presburger formulae constraints denoting when an edge is available; edges without constraints are always available. The grey label next to each node denotes its colour. E.g., $col(s) = 1 \in C = \{1, 2, 3, 4\}$.

Example 1. Consider the temporal parity game shown in Fig. 1. We will draw Player 1 states as diamond and those controlled by Player 2 as squares and sometimes write modulo expressions to define the edge availability. For example, the constraint on the edge from u to v can be written as the ∃PA-formula as $\exists y.(x = 3y) \vee (x = 3y + 1)$ and so this edge is available at times $0, 1, 3, 4, 6, \ldots$. The temporal graph underlying this game has period 15.

Player 1 has a winning strategy starting from (s, i) in the expansion by staying in state s until time $i' \geq i$ with $i' \equiv 0 \mod 5$ and then following the edge to $(t, i' + 1)$. If Player 2 ever chooses to move to r, he is trapped in an even-coloured cycle; if he stays in t forever, then the resulting game sees only colour 2 and is losing for him. Otherwise, if the game continues at $(s, i' + 2)$, Player 1 repeats as above (and wins plays that see both states s and t. The example shows that Player 1 s strategies depend on the time and are not positional in the vertices alone, even if the winning set has period 1. Indeed, the only possible vertex-positional strategy (cycle in s) is losing.

The vertices $\{s, t\}$ shaded in blue represent the vertex from which Player 1 can win starting at any time, following the strategy described above. From the vertices shaded in red, Player 2 can win starting at certain times. For example, Player 2 has a winning strategy from (u, i) if, and only if, $i \equiv 0 \mod 3$ or $i \equiv 1 \mod 3$ by moving to $(v, i + 1)$. Notice that this edge is not available, and thus Player 2 is forced to move to t at times $x \equiv 2 \mod 3$. In particular therefore, Player 1 wins from $(v, 0)$. The winning region for Player 1 is $\{(s, k), (t, k), (r, k), (u, 3k + 2), (v, 3k), (w, 3k + 1) \mid k \in \mathbb{N}\}$.

The algorithmic question we consider is determining the set of vertices from which Player 1 wins starting at time 0.

3 Reachability Games

We discuss a variant of temporal games that turns out to be central both for upper and lower bounds for solving games on temporal graphs.

We call these *punctual reachability games*, which are played on a static graph and Player 1 aims to reach the target precisely at a target time.

Definition 3. *A* punctual *reachability game $G = (V, E, s_0, F)$ is a game played on a static graph with vertices $V = V_1 \uplus V_2$, edges $E \subseteq V^2$, an initial state s_0 and set of target vertices $F \subseteq V$. An additional parameter is a target time $T \in \mathbb{N}$ given in binary. Player 1 wins a play if and only if a vertex in F is reached at time T.*

Punctual reachability games are really just a reformulation of the membership problem for alternating finite automata (AFA) [7] over a unary input alphabet. Player 1 wins the punctual reachability game with target T if, and only if, the word a^T is accepted by the AFA described by the game graph. Checking if a given unary word a^T is accepted by an AFA is complete for polynomial time if T is given in unary [20]. We first observe that it is PSPACE-hard if T is given in binary. We write in the terminology of punctual reachability games but the main argument is by reduction from the emptiness problem for unary AFA, which is PSPACE-compete [18,19]. We rely on the fact that the shortest word accepted by an AFA is at most exponential in the number of states.

Lemma 1. *Let $G = (V, E, s_0, F)$ be a reachability game on a static graph. If there exist $T \in \mathbb{N}$ so that Player 1 wins the punctual reachability game at target time T, then there exists some such $T \leq 2^{|V|}$.*

Proof. Assume towards contradiction that $T \geq 2^{|V|}$ is the smallest number such that Player 1 wins the punctual reachability game and consider some winning strategy σ. For any time $k \leq T$ we can consider the set $S_k \subseteq V$ of vertices occupied on any branch of length k on σ. By the pigeonhole principle, we observe $k < k' \leq T$ with $S_k = S_{k'}$, which allows to create a strategy σ' that follows σ until time k, then continues (and wins) according to σ as if it had just seen a length k' history leading to the same vertex. This shows that there exists a winning strategy for target time $T - (k - k')$, which contradicts the assumption. □

A lower bound for solving punctual reachability games is now immediate.

Lemma 2. *Solving punctual reachability games with target time T encoded in binary is* PSPACE*-hard.*

Proof. We reduce the non-emptiness problem of AFA over unary alphabets. In our terminology this is the decision problem if, for a given a reachability game $G = (V, E, s_0, F)$ there exists some $T \in \mathbb{N}$ so that Player 1 wins the punctual reachability game at target time T. This problem is PSPACE-complete [18].

By Lemma 1, positive instances can be witnessed by a small target $T \leq 2^{|V|}$ and so we know that it is PSPACE-hard to determine the existence of such a small target time that allows Player 1 to win.

Consider now the punctual reachability game G' that extends G by a new initial vertex s_0' that is owned by Player 1 and which has a self-loop as well as an edge to the original initial vertex s_0 with target time $T' \overset{\text{def}}{=} 2^{|V|}$. In G', Player 1 selects some number $T \leq T'$ by waiting in the initial vertex for $T' - T$ steps and then starts the game G with the target time T. Therefore, Player 1 wins in G' for target T' if, and only if, she wins for some $T \leq 2^{|V|}$ in G. □

Corollary 1. *Solving reachability games on finite temporal graphs is* PSPACE-*hard.*

Proof. We reduce the punctual reachability game with target T to an ordinary reachability game on a finite temporal graph. This can be done by introducing a new vertex u as the only target vertex, so that it is only reachable via edges from vertices in F at time exactly T. That is $E(s,u) \overset{\text{def}}{=} \{T\}$ and $E(s,t) = [0,T]$ for all $s,t \in V \setminus \{u\}$. Now Player 1 wins the reachability game for target u if, and only if, she wins the punctual reachability game with target F at time T. □

A matching PSPACE upper bound for solving punctual reachability games, as well as reachability games on finite temporal graphs can be achieved by computing the winning region backwards as follows.[1] For any game graph with vertices $V = V_1 \uplus V_2$, set $S \subseteq V$ and $i \in \{1,2\}$, let $Pre_i(S) \subseteq V$ denote the set of vertices from which Player i can force to reach S in one step.

$$Pre_i(S) \overset{\text{def}}{=} \{v \in V_i \mid \exists (v,v') \in E.v' \in S\} \cup \{v \in V_{1-i} \mid \forall (v,v') \in E.v' \in S\}$$

A straightforward induction on the duration T shows that Player i wins the punctual reachability game with target time T from vertex s if, and only if $s \in Pre_i^T(F)$, the T-fold iteration of Pre_i applied to the target set F.

Notice that knowledge of $Pre_i(S)$ is sufficient to compute $Pre_i^{k+1}(S)$. By iteratively unfolding the definition of Pre_i^k, we can compute $Pre_1^T(F)$ from $Pre_1^0(F) = F$ in polynomial space[2]. Together with Lemma 2 we conclude the following.

Theorem 1. *Solving punctual reachability games with target time T encoded in binary is* PSPACE-*complete.*

[1] For readers familiar with reachability games, the notion $Pre_1(S)$ above is very similar to, but not the same as the k-step attractor of S: The former contains states from which Player 1 can force to see the target in *exactly* k steps, whereas the latter contains those where the target is reachable in k *or fewer* steps.

[2] To be precise, naïvely unfolding the definition requires $\mathcal{O}(T + |V|^2)$ time, exponential in (the binary encoded input) T, and $\mathcal{O}(|V| + \log(T))$ space to memorise the current set $Pre_k \subseteq V$ as well as the time $k \leq T$ in binary.

The same approach works for reachability games on finite *temporal* graphs if applied to the expansion up to horizon $h(G)$, leading to the same time and space complexity upper bounds. The only difference is that computing $Pre_1^k(F \times \{T\})$ requires to check edge availability at time $T - k$.

Theorem 2. *Solving reachability games on finite temporal graphs is* PSPACE*-complete.*

Proof. Consider a temporal game with vertices $V = V_1 \uplus V_2$, edges $E : V^2 \to 2^{\mathbb{N}}$ target vertices $F \subseteq V$ and where $T = h(G)$ is the latest time an edge is available. We want to check if starting in an initial state s_0 at time 0, Player 1 can force to reach F at time T. In other words, for the game played on the expansion up to time T we want to decide if $(s_0, 0)$ is contained in $Pre_1^T(F \times \{T\})$.

By definition of the expansion, we have $Pre_1(S \times \{n\}) \subseteq V \times \{n-1\}$ for all $S \subseteq V$ and $n \leq T$. Since we can check the availability of an edge at time n in polynomial space, we can iteratively compute $Pre_1^n(F \times \{T\})$ backwards, starting with $Pre_1^0(F \times \{T\}) = F \times \{T\}$, and only memorising the current iteration $n \leq T$ and a set $W_n \subseteq V$ representing $Pre_1^n(F \times \{T\}) = W_n \times \{T - n\}$. □

4 Parity Games

We consider Parity games played on periodic temporal graphs. As input we take a temporal graph $G = (V, E)$ with period K, a partitioning $V = V_1 \uplus V_2$ of the vertices, as well as a colouring $col : V \to C$ that associates a colour out of a finite set $C \subset \mathbb{N}$ of colours to every state.

It will be convenient to write $col(\pi) \overset{\text{def}}{=} \max\{col(s_i) \mid 0 \leq i \leq k\}$ for the maximal colour of any vertex visited along a finite path $\pi = (s_0, 0)(s_1, 1) \ldots (s_k, k)$. The following relations R_s^σ capture the guarantees provided by a strategy σ if followed for one full period from vertex s.

Definition 4. *For a strategy σ and vertex $s \in V$ define $R_s^\sigma \subseteq V \times C$ be the relation containing $(t, c) \in R_s^\sigma$ if, and only if, there exists a finite play $\pi = (s, 0) \ldots (t, K)$ consistent with σ, that starts in s at time 0, ends in t at time K, and the maximum colour seen on the way is $col(\pi) = c$. We call R_s^σ the* summary *of s with respect to strategy σ.*

*A relation $B \subseteq V \times C$ is s-*realisable *if there is a strategy σ with $B = R_s^\sigma$.*

Example 2. Consider the game in Fig. 2 where vertex $u \in V_2$ has colour 2 and all other vertices have colour 1. The graph has period $K = 2$. The relations $\{(t, 1)\}$ and $\{(t, 2), (t', 2)\}$ are s-realisable, as witnessed by the strategies $\sigma(s) = t$ and $\sigma(s) = u$), respectively. However, $\{(t, 2)\}$ is not s-realisable as no Player 1 strategy guarantees to visit s then u then t.

Lemma 3. *Checking s-realisability is in* PSPACE*. That is, one can verify in polynomial space for a given temporal Parity game, state $s \in V$ and relation $B \subseteq V \times C$ whether B is s-realisable.*

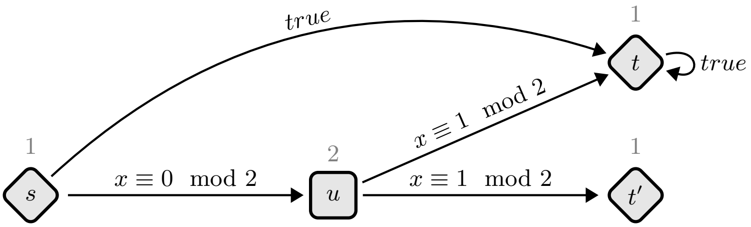

Fig. 2: The game from Example 2. Labels on vertices and edges denote colours and available times, respectively. The graph has period 2. In two rounds, Player 1 can force to end in t having seen colour 1, or in either t or t' but having seen a better colour 2.

Proof. We reduce checking realisability to solving a reachability game on a temporal graph that is only polynomially larger. More precisely, given a game $G = (V, E, col)$ consider the game $G' = (V', E', col')$ over vertices $V' \overset{\text{def}}{=} V \times C$ that keep track of the maximum colour seen so far. That is, the ownership of vertices and colours are lifted directly as $(s, c) \in V'_1 \iff s \in V_1$ and $col'(s, c) \overset{\text{def}}{=} col(s)$, and for any $i \in \mathbb{N}$, $s, t, s_0 \in V$, $c, d \in C$, we let (t, d) be an i-successor of (s, c) if, and only if, both t is an i-successor of s and $d = \max\{c, col(t)\}$.

Consider some relation $B \subseteq V \times C$. We have that B is s-realisable if, and only if, Player 1 wins the punctual reachability game on G' from vertex $(s, col(s))$ at time 0, towards target vertices $B \subseteq V'$ at target time K. Indeed, any winning Player 1 strategy in this game witnesses that B is s-realisable and vice versa. By Theorem 2, the existence of such a winning strategy can be verified in polynomial space by backwards-computing the winning region. □

The following defines a small, and PSPACE-verifiable certificate for Player 1 to win the parity game on a periodic temporal graph.

Definition 5 (Certificates). *Given temporal parity game (V, E, col) with period K, a* certificate *for Player 1 winning the game from initial vertex $s_0 \in V$ is a multigraph where the vertex set $V' \subseteq V$ contains s_0, and edges $E' \subseteq V' \times C \times V'$ are labelled by colours, such that*

1. *For every $s \in V'$, the set $Post(s) \overset{def}{=} \{(t, c) \mid (s, c, t) \in E'\}$ is s-realisable.*
2. *The maximal colour on every cycle reachable from s_0 is even.*

Notice that condition 1 implies that no vertex in a certificate is a deadlock. A certificate intuitively allows to derive Player 1 strategies based on those witnessing the realisability condition.

Example 3. Consider the game from Example 1 played on the temporal graph with period 15. A certificate for Player 1 winning from state v at time 0 is depicted in Fig. 3. Indeed, the Player 1 strategy mentioned in Example 1 (aim

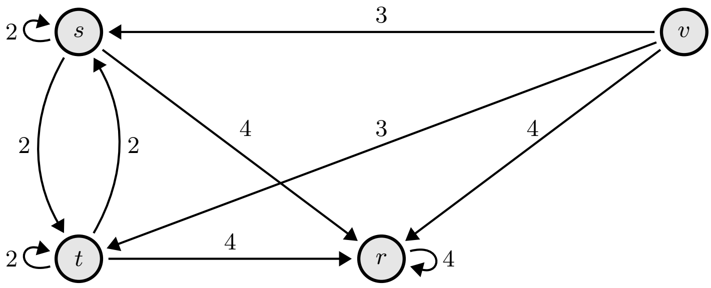

Fig. 3: A certificate that Player 1 wins the game in Example 1 from state v at time 0.

to alternate between s and t) witnesses that $Post(v) = \{(s,3),(t,3),(r,4)\}$ is v-realisable because it allows Player 1 to enforce that after $K = 15$ steps from v, the game ends up in one of those states via paths whose colour is dominated by $col(v) = 3$ or $col(r) = 4$.

Lemma 4. *Player 1 wins the parity game on G from vertex s_0 if, and only if, there exists a certificate.*

Proof. For the backward implication we argue that a certificate C allows to derive a winning strategy for Player 1 in the parity game G. By the realisability assumption (1), for each vertex $s \in V$ there must exist a Player 1 strategy σ_s with $R_s^{\sigma_s} = Post(s)$ that tells her how to play in G for K rounds if the starting time is a multiple of K. Moreover, suppose she plays according to σ_s for K rounds and let t and c be the vertex reached and maximal colour seen on the way. Then by definition of the summaries, $(t,c) \in R_s^{\sigma_s} = Post(s)$ and so in the certificate C there must be some edge $s \xrightarrow{c} t$.

Suppose Player 1 continues to play in G like this forever: From time $i \cdot K$ to $(i+1) \cdot K$ she plays according to some strategy σ_{s_i} determined by the vertex s_i reached at time $i \cdot K$. Any consistent infinite play ρ in G, chosen by her opponent, describes an infinite walk ρ' in C such that the colour seen in any step $i \in \mathbb{N}$ of ρ' is precisely the dominant colour on ρ between rounds iK and $(i+1)K$. Therefore the dominant colours seen infinitely often on ρ and ρ' are the same and, by certificate condition (2) on the colouring of cycles, even. We conclude that the constructed strategy for Player 1 is winning.

For the forward implication, assume that Player 1 wins the game on G from vertex s at time 0. Since the game G is played on a temporal graph with period K, its expansion up to time $K-1$ is an ordinary parity game on a static graph with vertices $V \times \{0,1,\dots,K-1\}$ where the second component indicates the time modulo K. Therefore, by positional determinacy of parity games (Proposition 1), we can assume that Player 1 wins in G using a strategy σ that is itself periodic. That is, $\sigma(hv) = \sigma(h'v)$ for any two histories h, h' of lengths $|h| \equiv |h'| \mod K$. Moreover, we can safely assume that σ is uniform, meaning that it is winning from any vertex $(s, 0)$ for which a winning strategy exists. Such a strategy induces a multigraph $C = (V, E')$ where the edge relation is defined by $(s, c, t) \in E' \iff (t, c) \in R^{\sigma}_{s}$. It remains to show the second condition for C to be a certificate, namely that any cycle in C, reachable from the initial vertex s_0, has an even maximal colour. Suppose otherwise, that C contains a reachable cycle whose maximal colour is odd. Then there must be play in G that is consistent with σ and which sees the same (odd) colour infinitely often. But this contradicts the assumption that σ was winning in G in the first place. □

Our main theorem is now an easy consequence of the existence of small certificates.

Theorem 3. *Solving parity games on periodic temporal graphs is* PSPACE-*complete.*

Proof. Hardness already holds for reachability games Lemma 2. For the upper bound we show membership in NPSPACE and use Savitch's theorem. By Lemma 4 it suffices to guess and verify a candidate certificate C. These are by definition polynomial in the number of vertices and colours in the given temporal parity game. Verifying the cycle condition (2) is trivial in polynomial time and verifying the realisability condition (1) is in PSPACE by Lemma 3. □

Remark 1. The PSPACE upper bound in Theorem 3 can easily be extended to games on temporal graphs that are *ultimately* periodic, meaning that there exist $T, K \in \mathbb{N}$ so that for all $n \geq T$, $s \xrightarrow{n} t$ implies $s \xrightarrow{n+K} t$. Such games can be solved by first considering the periodic suffix according to Theorem 3 thereby computing the winning region for Player 1 at time exactly T, and then solving the temporal reachability game with horizon T.

5 Monotonicity

In this section, we consider the effects of monotonicity assumptions on the edge relation with respect to time on the complexity of solving reachability games. We first show that reachability games remain PSPACE-hard even if the edge relation is decreasing (or increasing) with time. We then give a fragment for which the problem becomes solvable in polynomial time.

Increasing and Decreasing temporal graphs: Let the edge between vertices $u, v \in V$ of a temporal graph be referred to as *decreasing* if $u \xrightarrow{i+1} v$ implies $u \xrightarrow{i} v$ for all $i \in \mathbb{N}$, i.e. edges can only disappear over time. Similarly, call the edge *increasing* if for all $i \in \mathbb{N}$ we have that $u \xrightarrow{i} v$ implies $u \xrightarrow{i+1} v$; i.e. an edge available at current time continues to be available in the future. A temporal graph is decreasing (increasing) if all its edges are. We assume that the times at which edge availability changes are given in binary. More specifically, every edge is given as inequality constraint of the form $\Phi_{u,v}(x) \overset{\text{def}}{=} x \leq n$ (respectively $x \geq n$) for some $n \in \mathbb{N}$.

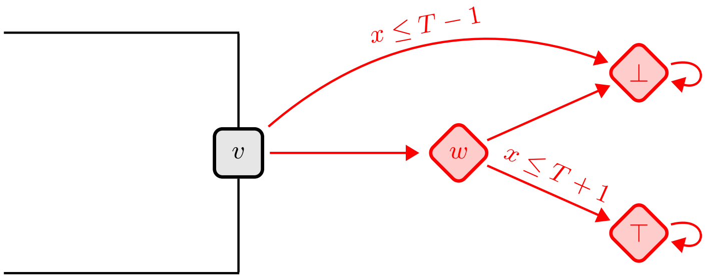

Fig. 4: Reduction from a punctual reachability game to a reachability game on a temporal graph that is finite and decreasing, see Theorem 4. Components added are shown in red.

Although both restrictions imply that the graph is ultimately static, we observe that solving reachability games on such monotonically increasing or decreasing temporal graphs remains PSPACE-complete.

Theorem 4. *Solving reachability and Parity games on decreasing (respectively increasing) temporal graphs is* PSPACE*-complete.*

Proof. The upper bound holds for parity games as the description of the temporal graph explicitly includes a maximal time T from which the graph becomes static. One can therefore solve the Parity game for the static suffix graph (in NP) and then apply the PSPACE procedure (Theorem 2) to solve for temporal reachability towards the winning region at time T. Alternatively, the same upper bound also follows from Theorem 3 and Remark 1.

For the lower bound we reduce from punctual reachability games which are PSPACE-hard by Lemma 2. Consider a (static) graph G and a target time $T \in \mathbb{N}$ given in binary. Without loss of generality, assume that the target vertex v has no outgoing edges. We convert G into a temporal graph G' with $V' = V \cup \{w, \top, \perp\}$, $V_1' = (V_1 \setminus \{v\}) \cup \{w\}$, $V_2' = V' \setminus V_1'$ and new target $\top$. The vertex $\perp$ is a sink state and the original target vertex v is now controlled by Player 2. Edge availabilities are $v \xrightarrow{x} \perp$ if $x \leq T-1$, $v \xrightarrow{x} w$ if $x \leq T+1$, $w \xrightarrow{x} \top$ if $x \leq T+1$, and all other edges disappear after time $T+1$. The constructed temporal graph is finite and decreasing. See Fig. 4. The construction ensures that the only way to reach $\top$ is to reach v at time T, w at time $T+1$ and take the edge from w to $\top$ at time $T+1$. Player 1 wins in G' if and only if she wins the punctual reachability game on G.

A similar reduction works in the case of increasing temporal graphs by switching the ownership of vertices v and w. The vertex v, now controlled by Player 1 has the edge $v \xrightarrow{x} w$ at times $x \geq T$ and the edge $v \longrightarrow \perp$ at all times. The vertex w now controlled by Player 2 has the edge $w \longrightarrow \top$ available at all times but the edge $w \xrightarrow{x} \perp$ becomes available at time $x \geq T+2$. □

Declining and improving temporal games: We now consider the restriction where all edges controlled by one player are increasing and those of the over player are decreasing. Taking the perspective of the system Player 1, we call a game on a temporal graph *declining* if all edges $u \longrightarrow v$ with $u \in V_1$ are decreasing and all edges $u \longrightarrow v$ with $u \in V_2$ are increasing. Note that *declining* is a property of the game and not the graph as the definition requires a distinction based on ownership of vertices, which is specified by the game and not the underlying graph. From now on, we refer to such games as declining temporal reachability (or parity) games. Notice that Player 1 has fewer, and Player 2 has more choices to move at later times. Analogously, call the game *improving* if the conditions are reversed, i.e., all edges $u \longrightarrow v$ with $u \in V_1$ are increasing and all edges $u \longrightarrow v$ with $u \in V_2$ are decreasing.

We show that declining (and improving) temporal reachability games can be solved in polynomial time.

Theorem 5. *Solving declining (respectively improving) temporal reachability games is in* P.

Proof. We first give the proof for declining games. Consider the reachability game on the expansion with vertices $V \times \mathbb{N}$ such that the target set is $F \times \mathbb{N}$. For $k \in \mathbb{N}$ let $W_k \subseteq V$ be the set of those vertices u such that Player 1 has a winning strategy from (u, k). We first show that

$$W_{i+1} \subseteq W_i \tag{1}$$

For sake of contradiction, suppose there exists $u \in W_{i+1} \setminus W_i$. Let σ^1_{i+1} be a (positional) winning strategy from $(u, i+1)$ for Player 1 in the expansion. Since $u \notin W_i$, by positional determinacy of reachability games (Proposition 1), Player 2 has a winning strategy σ^2_i from (u, i). Consider a strategy σ^1_i for Player 1, such that for all $v \in V_1$, $\sigma^1_i(v, k) \overset{\text{def}}{=} \sigma^1_{i+1}(v, k+1)$, for all $k \geq i$. Similarly, let σ^2_{i+1} be the strategy for Player 2, such that for all $v \in V_2$, $\sigma^2_{i+1}(v, k+1) = \sigma^2_i(v, k)$, for all $k \geq i$, Note that this is well defined because by definition of declining games, i.e, $v \xrightarrow{k+1} u$ implies $v \xrightarrow{k} u$ for all $v \in V_1$, and $v \xrightarrow{k} u$ implies $v \xrightarrow{k+1} u$, for all $v \in V_2$. Starting from the vertex $(u, i+1)$, the pair of strategies $(\sigma^1_{i+1}, \sigma^2_{i+1})$ defines a unique play π_{i+1}, which is winning for Player 1. Similarly, the pair of strategies (σ^1_i, σ^2_i) define a play π_i which is winning for Player 2 starting from (u, i). However, the two plays visit the same set of states, particularly, (v, k) is visited in π_i if and only if $(v, k+1)$ is visited in π_{i+1}. Therefore, either both are winning for Player 1 or both are losing for Player 2, which is a contradiction. Let $N \subseteq \mathbb{N}$ be the set of times at which the graph changes, i.e.

$$N = \{c \mid \exists \varPhi_{u,v}(x) = x \lhd c, \text{ where } \lhd \in \{\leq, \geq\}\}$$

Let $m \overset{\text{def}}{=} \max(N)$ be the latest time any edge availability changes. We show that $W_m = W_k$ for all $k \geq m$. To see this, note that W_m is equal to the winning region for Player 1 in the (static) reachability game played on $G_m = (V, E_m)$,

Algorithm 1 Algorithm for declining games with set of change times N and $m = \max(N)$

```
$W \leftarrow \text{Solve}(G_m)$                          $\triangleright$ Computes Player 1 winning region in $G_m$
while $N \neq \emptyset$ do
    $n \leftarrow max(N)$
    if $(Pre_1(W \times \{n\}) = W$ then
        $N \leftarrow N \setminus n$                        $\triangleright$ Accelerate to next change time
    else
        $W \leftarrow Pre_1(W)$
        $N \leftarrow N \cup \{n-1\} \setminus \{n\}$
    end if
end while
```

where $E_m = \{(u,v) \mid u \xrightarrow{m} v\}$. Consider a (positional) winning strategy σ_m for Player 1 in G_m and define a positional strategy $\sigma(v,k) = \sigma_m(v)$, for $k \geq m$. Since the graph is static after time m, this is well defined. Starting from a vertex (u,k), a vertex $(v, k+k')$ is visited on a σ-consistent path if and only if there is a σ_m-consistent path $u \longrightarrow_{k'} v$. Therefore, σ is a winning strategy from any vertex (v,k) such that $k \geq m$ and $v \in W_m$. Moreover, the set W_m can be computed in time $\mathcal{O}(|V|^2)$ by solving the reachability game on G_m [13, Theorem 12].

To solve reachability on declining temporal games, we can first compute the winning region W_m in the stabilised game G_m. This means $W_m \times [m, \infty)$ is winning for Player 1. To win the declining temporal reachability game, Player 1 can play the punctual reachability game with target set W_m at target time m. The winning region for Player 1 at time 0 can therefore be computed as $Pre_1^m(W_m \times \{m\})$ as outlined in the proof of Theorem 2. Note that naïvely this only gives a PSPACE upper bound as in the worst case, we would compute Pre_1 an exponential (m) times.

To overcome this, note that in the expansion graph $Pre_1^i(W_m \times \{m\}) = W_{m-i} \times \{m-i\}$. According to Eq. (1), $W_{m-i} \subseteq W_{m-i'}$ for $i' > i$. Let i, i' be such that $m-i$ and $m-i'$ are both consecutive change points, i.e, $m-i, m-i' \in N$ and $\forall \ell \in N. \ell < m-i' \vee \ell > m-i$. Since the edge availability of the graph does not change between time $m-i'$ and $m-i$, we have $W_{m-i-1} = W_{m-i}$ implies $W_{m-i'} = W_{m-i}$. Therefore, we can accelerate the Pre_1 computation and directly move to the time step $m-i'$, i.e, the i'th iteration in the computation. This case is illustrated at time $n' = m-i'$ in Fig. 5.

With this change, our algorithm runs the Pre_1 computation at most $|V|+|N|$, as each Pre_1 computation either corresponds to a step a time in N when the graph changes, or a step in which the winning region grows such as at time n in Fig. 5. Since each Pre_1 computation can be done in polynomial time, we get a PTIME algorithm in this case, shown in Algorithm 1.

The case for improving temporal reachability games can be solved similarly. Instead of computing the winning region for Player 1 in G_m, we start with computing the winning region W_m^2 for Player 2 in G_m and switch the roles of Player 1 and Player 2, i.e, Player 2 has the punctual reachability objective with

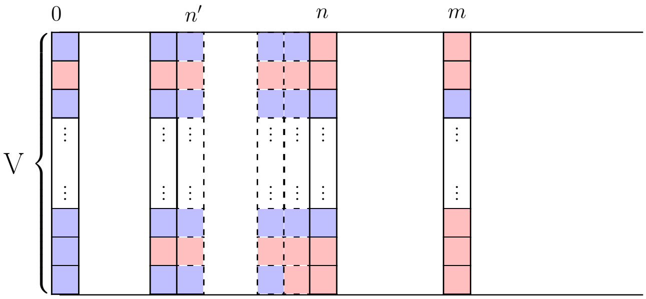

Fig. 5: Illustration of Algorithm 1. The blue vertices at time i denote the winning region W_i for Player 1. The times $n, n' \in N$ and Pre_1 computation at change point n increases the winning region but is stable at time n'.

target set W^2_m and target time m, which can be solved as above. This gives us an algorithm to compute the winning region for Player 2 and by determinacy of reachability games on infinite graphs, we can compute the winning region for Player 1 at time 0 as well. □

Remark 2. Algorithm 1 also works for parity objectives by changing step 1, where Solve(G_m) would amount to solving the parity game on the static graph G_m. This can be done in quasi-polynomial time and therefore gives a quasi-polynomial time algorithm to solve declining (improving) temporal parity games and in particular, gives membership in the complexity class $\mathsf{NP} \cap \mathsf{coNP}$.

Since the declining (improving) restriction on games on temporal graphs allow for improved algorithms, a natural question is to try to lift this approach to a larger class of games on temporal graphs. Note that the above restrictions are a special case of eventually periodic temporal graphs with a prefix of time m followed by a periodic graph with period 1. Now, we consider temporal graphs of period $K > 1$ such that the game arena is declining (improving) within each period. Formally, a game on a temporal graph G is *periodically declining* (improving) if there exists a period K such that for all $k \in \mathbb{N}$, $k \in E(u, v)$ if and only if $k+K \in E(u, v)$; and the game on the finite temporal graph resulting from G by making the graph constant from time K onwards, is declining (improving). We prove that this case is PSPACE-hard, even with reachability objectives.

Theorem 6. *Solving periodically declining (improving) temporal reachability games is* PSPACE*-complete.*

Proof. The upper bound follows from the general case of parity games on periodic temporal graphs in Theorem 3. The lower bound is by reduction from punctual reachability games. See Fig. 6. Given a (static) graph G with target state v and target time T, we obtain a periodically declining game G' with period $K = T+1$, vertices $V \cup \{w, \bot, \top\}$, new target $\top$, such that $V'_1 = V_1 \cup \{w, \bot, \top\}$ and $V'_2 = V_2$.

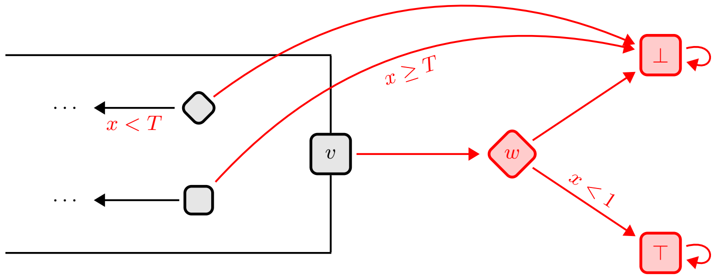

Fig. 6: Reduction from a punctual reachability game to a reachability game on a temporal graphs that is periodic and declining, see Theorem 6. Parts added are shown in red.

We assume without loss of generality that the original target v is a Player 1 vertex, i.e, $v \in V_1$.

We describe the edge availability in G' up to the period $K = T + 1$. For all edges (s, t) of the original graph G, such that $s \in V_1$, the edge $s \xrightarrow{x} t$ is available if and only if $x < T$. Moreover for all $s \in V_1 \setminus \{v\}$, there is a new edge $s \xrightarrow{x} \bot$ available at all times $x \leq T$. For all $s \in V_2$, there is an edge $s \xrightarrow{x} t$ is available at all times (until end of period) and $s \xrightarrow{x} \bot$ is available after time $x \geq T$. These edges ensure that if a play in the original punctual reachability game ends in a vertex of the game other than v at time T, then Player 2 can force the play to reach the sink state $\bot$ and win.

From the original target v, there is an edge to the new state w at all times. From the state w, there are edges $w \longrightarrow \bot$ at all times and $w \xrightarrow{x} \top$ if $x = 0$. If the state w is reached at time k such that $1 < k < T + 1$, then the play is forced to go to $\bot$. The only winning strategy for Player 1 is to reach v at time T, w at time $T + 1$ at which the time is reset due to periodicity. The edge $w \xrightarrow{T+1} \top$ is now available for Player 1 and they can reach the new target $\top$.

The lower bound for the case of periodically increasing temporal reachability games follows by the same construction and using the duality between improving and declining games on temporal graphs. Given a punctual reachability game G with vertices $V = V_1 \uplus V_2$ with target set F, we obtain the dual punctual reachability game $\hat{G}$ with same target time by first switch the ownership of vertices, i.e, $\hat{V}_i = V_{3-i}$, $i \in \{1, 2\}$ and make the new target as $V \setminus F$. It is easy to see that Player 1 wins G if and only if Player 2 wins $\hat{G}$.

Applying the same construction as shown in Fig. 6 to $\hat{G}$, we obtain a periodically declining temporal reachability game $\hat{G}'$, preserving the winner. Now switching the ownership of vertices in $\hat{G}'$ yields a periodically improving temporal reachability game G' which is winning for Player 1 if and only if Player 1 wins G. □

6 Conclusion

In this work we showed that parity games on ultimately periodic temporal graphs are solvable in polynomial space. The lower bound already holds for the very special case of punctual reachability games, and the PSPACE upper bound, which improves on the naïve exponential-space algorithm on the unfolded graph, is achieved by proving the existence of small, PSPACE-verifiable certificates.

We stress again that all constructions are effective no matter how the temporal graphs are defined, as long as checking edge availability for binary encoded times is no obstacle. In the paper we use edge constraints given in the existential fragment of Presburger arithmetic but alternate representations, for example using compressed binary strings of length $h(G)$ given as Straight-Line Programs [5, Section 3] would equally work. Checking existence of edge at time i would correspond to querying whether the i^{th} bit is 1 or not which is P-complete [27, Theorem 1].

The games considered here are somewhat orthogonal to parity games played on the configuration graphs of timed automata, where time is continuous, and constraints are *quantifier-free* formulae involving possibly more than one variable (clocks). Solving parity games on timed automata with two clocks is complete for EXP but is in P if there is at most one one clock [38] [16, Contribution 3(d)]. Games on temporal graphs with quantifier-free constraints corresponds to a subclass of timed automata games with two-clocks, with intermediate complexity of PSPACE. This is because translating a temporal graph game to a timed automata game requires two clocks: one to hold the global time used to check the edge predicate and one to ensure that time progresses one unit per step.

An interesting continuation of the work presented here would be to consider mean-payoff games [11] played on temporal graphs, possibly with dynamic step-rewards depending on the time. If rewards are constant but the edge availability is dynamic, then our arguments for improved algorithms on declining/improving graphs would easily transfer. However, the PSPACE upper bound using summaries seems trickier, particularly checking realisability of suitable certificates.

Acknowledgements This work was supported by the Engineering and Physical Sciences Research Council (EPSRC), grant EP/V025848/1. We thank Viktor Zamaraev and Sven Schewe for fruitful discussions and constructive feedback.

References

1. Automata Logics, and Infinite Games: A Guide to Current Research. Springer-Verlag (2002). https://doi.org/10.1007/3-540-36387-4
2. Akrida, E.C., Mertzios, G.B., Spirakis, P.G., Zamaraev, V.: Temporal vertex cover with a sliding time window. Journal of Computer and System Sciences **107**, 108–123 (2020). https://doi.org/https://doi.org/10.1016/j.jcss.2019.08.002
3. Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. **126**(2), 183 – 235 (1994). https://doi.org/10.1016/0304-3975(94)90010-8

4. Avni, G., Ghorpade, P., Guha, S.: A Game of Pawns. In: International Conference on Concurrency Theory. Leibniz International Proceedings in Informatics (LIPIcs), vol. 279, pp. 16:1–16:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2023). https://doi.org/10.4230/LIPIcs.CONCUR.2023.16
5. Babai, L., Szemeredi, E.: On the complexity of matrix group problems i. In: Annual Symposium on Foundations of Computer Science. pp. 229–240 (1984). https://doi.org/10.1109/SFCS.1984.715919
6. Calude, C.S., Jain, S., Khoussainov, B., Li, W., Stephan, F.: Deciding parity games in quasipolynomial time. In: Symposium on Theory of Computing. pp. 252–263 (2017). https://doi.org/10.1145/3055399.3055409
7. Chandra, A.K., Kozen, D.C., Stockmeyer, L.J.: Alternation. Journal of the ACM (JACM) **28**(1), 114–133 (1981). https://doi.org/10.1145/322234.322243
8. Chatterjee, K., Henzinger, T.A., Prabhu, V.S.: Timed Parity Games: Complexity and Robustness. Logical Methods in Computer Science **Volume 7, Issue 4** (Dec 2011). https://doi.org/10.2168/LMCS-7(4:8)2011
9. Colcombet, T., Fijalkow, N.: Universal Graphs and Good for Games Automata: New Tools for Infinite Duration Games. In: International Conference on Foundations of Software Science and Computational Structures. LNCS, vol. 11425, pp. 1–26. Springer (2019). https://doi.org/10.1007/978-3-030-17127-8_1
10. De Carufel, J.L., Flocchini, P., Santoro, N., Simard, F.: Cops & robber on periodic temporal graphs: Characterization and improved bounds. In: Structural Information and Communication Complexity. pp. 386–405. Springer Nature Switzerland (2023). https://doi.org/10.1007/978-3-031-32733-9_17
11. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. International Journal of Game Theory **8**(2), 109–113 (Jun 1979). https://doi.org/10.1007/BF01768705
12. Erlebach, T., Hoffmann, M., Kammer, F.: On temporal graph exploration. Journal of Computer and System Sciences **119**, 1–18 (2021). https://doi.org/https://doi.org/10.1016/j.jcss.2021.01.005
13. Fijalkow, N., Bertrand, N., Bouyer-Decitre, P., Brenguier, R., Carayol, A., Fearnley, J., Gimbert, H., Horn, F., Ibsen-Jensen, R., Markey, N., Monmege, B., Novotný, P., Randour, M., Sankur, O., Schmitz, S., Serre, O., Skomra, M.: Games on graphs (2023). https://doi.org/10.48550/arXiv.2305.10546
14. Flocchini, P., Mans, B., Santoro, N.: Exploration of periodically varying graphs. In: Algorithms and Computation. pp. 534–543. Springer Berlin Heidelberg (2009). https://doi.org/10.1007/978-3-642-10631-6_55
15. Haase, C.: A survival guide to presburger arithmetic. SIGLOG News **5**(3), 67–82 (2018). https://doi.org/10.1145/3242953.3242964
16. Hansen, T.D., Ibsen-Jensen, R., Miltersen, P.B.: A faster algorithm for solving one-clock priced timed games (2013). https://doi.org/10.1007/978-3-642-40184-8_37
17. Holme, P., Saramäki, J.: Temporal Network Theory (01 2019). https://doi.org/10.1007/978-3-030-23495-9
18. Holzer, M.: On emptiness and counting for alternating finite automata. In: International Conference on Developments in Language Theory. pp. 88–97 (1995)
19. Janćar, P., Sawa, Z.: A note on emptiness for alternating finite automata with a one-letter alphabet. Inf. Process. Lett. **104**(5), 164–167 (2007). https://doi.org/10.1016/j.ipl.2007.06.006
20. Jiang, T., Ravikumar, B.: A note on the space complexity of some decision problems for finite automata. Inf. Process. Lett. **40**(1), 25–31 (1991). https://doi.org/https://doi.org/10.1016/S0020-0190(05)80006-7

21. Jurdziński, M., Trivedi, A.: Reachability-time games on timed automata. In: International Colloquium on Automata, Languages and Programming. pp. 838–849. Springer Berlin Heidelberg (2007). https://doi.org/10.1007/978-3-540-73420-8_72
22. Jurdziński, M.: Deciding the winner in parity games is in up $\cap$ co-up. Inf. Process. Lett. **68**(3), 119–124 (1998). https://doi.org/https://doi.org/10.1016/S0020-0190(98)00150-1
23. Jurdziński, M., Lazić, R.: Succinct Progress Measures for Solving Parity Games. In: Annual IEEE Symposium on Logic in Computer Science. pp. 1–9. IEEE Computer Society (2017). https://doi.org/10.1109/LICS.2017.8005092
24. Kuhn, F., Lynch, N., Oshman, R.: Distributed computation in dynamic networks. In: Symposium on Theory of Computing. p. 513–522. STOC '10, Association for Computing Machinery (2010). https://doi.org/10.1145/1806689.1806760
25. Lehtinen, K., Parys, P., Schewe, S., Wojtczak, D.: A Recursive Approach to Solving Parity Games in Quasipolynomial Time. Logical Methods in Computer Science **18**(1), 8:1–18 (2022). https://doi.org/10.46298/lmcs-18(1:8)2022
26. Lehtinen, K., Boker, U.: Register Games. Logical Methods in Computer Science **16**(2), 6:1–6:25 (2020). https://doi.org/10.23638/LMCS-16(2:6)2020
27. Lifshits, Y., Lohrey, M.: Querying and embedding compressed texts. In: International Symposium on Mathematical Foundations of Computer Science. pp. 681–692. Springer Berlin Heidelberg (2006). https://doi.org/10.1007/11821069_59
28. Maler, O., Pnueli, A., Sifakis, J.: On the synthesis of discrete controllers for timed systems. In: International Symposium on Theoretical Aspects of Computer Science. pp. 229–242. Springer Berlin Heidelberg (1995). https://doi.org/10.1007/3-540-59042-0_76
29. Mertzios, G.B., Molter, H., Niedermeier, R., Zamaraev, V., Zschoche, P.: Computing maximum matchings in temporal graphs. Journal of Computer and System Sciences **137**, 1–19 (2023). https://doi.org/https://doi.org/10.1016/j.jcss.2023.04.005
30. Mertzios, G.B., Molter, H., Zamaraev, V.: Sliding window temporal graph coloring. Journal of Computer and System Sciences **120**, 97–115 (2021). https://doi.org/https://doi.org/10.1016/j.jcss.2021.03.005
31. Michail, O.: An Introduction to Temporal Graphs: An Algorithmic Perspective, pp. 308–343. Springer International Publishing (2015). https://doi.org/10.1007/978-3-319-24024-4_18
32. Michail, O., Chatzigiannakis, I., Spirakis, P.G.: Causality, influence, and computation in possibly disconnected synchronous dynamic networks. Journal of Parallel and Distributed Computing **74**(1), 2016–2026 (2014). https://doi.org/10.1016/j.jpdc.2013.07.007
33. Michail, O., Spirakis, P.G.: Traveling salesman problems in temporal graphs. In: International Symposium on Mathematical Foundations of Computer Science. pp. 553–564. Springer Berlin Heidelberg (2014). https://doi.org/10.1016/j.tcs.2016.04.006
34. Pnueli, A., Rosner, R.: On the synthesis of a reactive module. In: Annual Symposium on Principles of Programming Languages. p. 179–190. POPL '89, Association for Computing Machinery (1989). https://doi.org/10.1145/75277.75293
35. Pnueli, A.: The temporal logic of programs. In: Annual Symposium on Foundations of Computer Science. p. 46–57. SFCS '77, IEEE Computer Society (1977). https://doi.org/10.1109/SFCS.1977.32

36. Ravi, R.: Rapid rumor ramification: Approximating the minimum broadcast time. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 202–213 (1994). https://doi.org/10.1109/SFCS.1994.365693
37. Scarpellini, B.: Complexity of subcases of presburger arithmetic. Transactions of the American Mathematical Society **284**, 203–218 (1984). https://doi.org/10.1090/s0002-9947-1984-0742421-9
38. Trivedi, A.: Competitive optimisation on timed automata. Ph.D. thesis, University of Warwick (April 2009)