Computing Average Response Time*

Krishnendu Chatterjee[1], Thomas A. Henzinger[1], and Jan Otop[2]

[1] IST Austria
[2] University of Wrocław

Abstract. Responsiveness —the requirement that every request to a system be eventually handled— is one of the fundamental liveness properties of a reactive system. Average response time is a quantitative measure for the responsiveness requirement used commonly in performance evaluation. We show how average response time can be computed on state-transition graphs, on Markov chains, and on game graphs. In all three cases, we give polynomial-time algorithms.

1. Introduction

Graphs and their generalizations provide the mathematical framework for modeling the behavior of reactive systems. The vertices of the graph represent states of the system, the edges represent transitions, and paths of the graph represent traces of the system. The two classical extensions of the graph model for reactive systems are with (i) probabilities and (ii) interaction with an adversary. In the presence of stochasticity in the system, from every vertex there is a probability distribution of transitions to the next vertex, and this gives rise to a Markov chain. In the presence of an adversary, the vertices of the graph are partitioned into vertices that are controlled by the proponent and vertices that are controlled by the opponent, and the choice of outgoing transition from a vertex is decided, respectively, by the proponent or the opponent. This gives rise to two player games on graphs. While graphs represent closed systems, games on graphs represent systems that interact with an adversarial environment, and Markov chains represent probabilistic systems. Thus, graphs, games on graphs, and Markov chains are fundamental models for the behavior of reactive systems.

One of the fundamental liveness properties in system analysis is the *responsiveness* property, which requires that every request of a system component is eventually granted. The responsiveness property is a qualitative property that classifies every trace of the system as correct or incorrect. In contrast to qualitative properties, the performance evaluation of systems requires quantitative measures on traces. A quantitative property assigns a real number to every trace,

* This research was supported in part by the Austrian Science Fund (FWF) under grants S11402-N23,S11407-N23 (RiSE/SHiNE) and Z211-N23 (Wittgenstein Award), ERC Start grant (279307: Graph Games), Vienna Science and Technology Fund (WWTF) through project ICT15-003 and by the National Science Centre (NCN), Poland under grant 2014/15/D/ST6/04543.

in contrast to the Boolean values ("correct" vs. "incorrect") assigned by qualitative properties. A basic quantitative property is the *mean-payoff* property, where every transition of the system is assigned a cost. The mean payoff of a trace is the limit (inferior) of the sequence of average costs c_n (i.e., the "long-run average"), where for every $n > 0$, the average cost c_n is computed over the finite prefix of length n of the trace. Building upon the mean-payoff property, we consider a quantitative version of the responsiveness property, the *average response time* (ART), defined as follows: for every request, the response time for the request is the number of steps to the next grant, and the ART of a trace is the long-run average of all response times of the trace. If there are only finitely many request-grant pairs, than the ART of the trace is a finite average. If there is a request without a subsequent grant, or if an infinite sequence of response times has no upper bound, then the ART is infinite. In this way, the ART property differs from the mean-payoff property, because the mean payoff of a trace is always bounded by the maximum cost of a transition.

The ART of a trace is a natural quantitative measure of the responsiveness, and thus a basic system property for performance evaluation [15]. For graphs, we are interested in the minimal and maximal ART over all traces (i.e., all infinite paths of the graph). For Markov chains, we are interested in the expected value of the ART. For games on graphs, we are interested in the optimal strategy of a system to make the ART as small or as large as possible, no matter how the environment behaves. The ART that is achieved by an optimal strategy of a proponent who tries to make the ART as small as possible (the "minimizer") against an optimal strategy of an opponent who tries to make the ART as large as possible (the "maximizer") is called the ART value of the game.

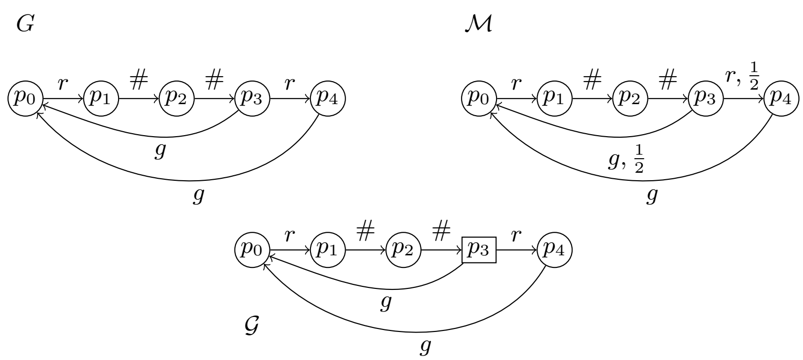

Fig. 1. Three models of a reactive system: the graph G, the Markov chain $\mathcal{M}$, and the game graph $\mathcal{G}$. Transitions in the Markov chain $\mathcal{M}$ are labeled with probabilities; we omit the probability 1 on the unique outgoing transitions from the vertexes p_0, p_1, p_2, and p_4. In the game graph $\mathcal{G}$, circled positions belong to the minimizing proponent, whereas the squared position p_3 belongs to the maximizing opponent.

Example 1. Figure 1 presents the three models G, $\mathcal{M}$, and $\mathcal{G}$ with transitions labeled by the following actions: requests r, grants g, and other instructions $\#$. The graph G has two simple cycles: $C_1 = p_0p_1p_2p_3$ and $C_2 = p_0p_1p_2p_3p_4$. The cycles C_1, C_2 yield respectively the sequences of actions $r\#\#g$ and $r\#\#rg$. Thus, the ART of C_1 is 3 and the ART of C_2 is $\frac{4+1}{2}$. Any infinite path can be partitioned into cycles C_1 and C_2, and hence the minimal ART of G is $\frac{5}{2}$ and the maximal ART of G is 3.

The Markov chain $\mathcal{M}$ results from the graph G and hence we observe that both cycles C_1 and C_2 occur with equal probability $\frac{1}{2}$. Therefore, the expected ART of $\mathcal{M}$ is $(\frac{1}{2} \cdot \frac{5}{2}) + (\frac{1}{2} \cdot 3) = \frac{11}{4}$.

Finally, the game arena $\mathcal{G}$ results from G by assigning p_3 to the player that attempts to maximize the ART. Thus, the ART value of $\mathcal{G}$ is 3, as the maximizer can always pick the move from p_3 to p_0. Interestingly, to maximize the ART, the opponent does not postpone the grant by moving from p_3 to p_4, but rather issues immediately a grant, which prevents the emission of a promptly satisfied request at (p_3, p_4). Such a promptly satisfied request would decrease the ART and thus the maximizing opponent is better off by issuing the grant quickly.

In summary, the minimal and maximal ART are easily and naturally defined numerical values of a labeled graph, the expected ART is the corresponding value of a labeled Markov chain, and the ART value is the corresponding quantity for a labeled 2-player game graph. In this paper, we present algorithms for computing these four values.

Automata provide a natural framework for specifying qualitative properties. Their extension, weighted automata, provide a framework for expressing quantitative properties [2, 9]. While weighted finite automata with mean-payoff measure [2] cannot express the ART property [6], extensions of weighted finite automata with *nesting* have been proposed in [5–8] as a quantitative specification framework that can express the ART property. These works focus on solving the quantitative emptiness and universality questions for entire classes of weighted finite automata [5–7], as well as on the evaluation of such automata classes with respect to probability distributions over words [8]. However, the solution and complexity of computing the specific ART property for graphs (minimal and maximal ART), games on graphs (ART value), and Markov chains (expected ART) has not been studied before.

In this work we consider the specific problem of computing the ART property for graphs, game graphs, and Markov chains. Our main result is that for all three models the ART property can be computed in polynomial time. The precise computational complexities differ for the various models (see Theorem 3, Theorem 4, and Theorem 5). If we compare our results to previous results for the class of nested weighted finite automata that can express the ART property, we see the following: (a) while solving automaton emptiness is similar in flavor to computing ART on graphs, for general nested weighted automata the resulting complexities are PSPACE and higher, whereas we present polynomial-time algorithms; (b) for Markov chains our results are easily derived from results of [8];

and (c) to the best of our knowledge, the problem of computing ART for games on graphs has not been studied before.

2. Preliminaries

We present notions and notations used throughout the paper. We begin with models of reactive systems: graphs, games and Markov chains (Section 2.1). Then, we present basic objectives studied with these models (Section 2.2), which lead to computational questions (Section 2.3). Finally, we recall previous results on computational questions for mean-payoff objectives (Section 2.4).

2.1 Models

Game arena. A *game arena* $\mathcal{G}$ is a tuple (V, V_1, V_2, E) where (V, E) is a finite graph, (V_1, V_2) is a partition of V into positions of Player 1 and Player 2, respectively. To present results in a uniform way, we consider graphs as arenas with all positions belonging to one player, i.e., we identify $(V, V, \emptyset, E)$ (resp., $(V, V, \emptyset, E)$) with (V, E). We assume (for technical convenience) that for every position $v \in V$ there is at least one outgoing edge.

Game plays. A game on an arena $\mathcal{G}$ is played as follows: a token is placed at a starting position, and whenever the token is at a Player-1 position, then Player 1 chooses an outgoing edge to move the token, and when the token is at a Player-2 position, then Player 2 does likewise. As a consequence we obtain an infinite sequence of positions, which is called a *play*, and *strategies* are recipes to extend finite prefix of plays (i.e., the recipes to describe how to move tokens). We formally define them below.

Strategies and plays. Given a game arena $\mathcal{G}$, a function $\sigma_1 : V^* \cdot V_1 \mapsto V$ (resp., $\sigma_2 : V^* \cdot V_2 \mapsto V$) is a *strategy* for Player 1 (resp., Player 2) on $\mathcal{G}$ iff $\sigma_j(v_0 v_1 \dots v_k) = v$ implies $(v_k, v) \in E$. In other words, given a finite sequence of positions that ends at a Player-1 position (representing the history of interactions), a strategy for Player 1 chooses the next position respecting the edge relationship (to move the token). We denote the set of all strategies for Player 1 (resp., Player 2) on $\mathcal{G}$ by $\mathcal{S}_1[\mathcal{G}]$ (resp., $\mathcal{S}_2[\mathcal{G}]$). A strategy σ_i has *finite memory* if there exist a finite set $\mathcal{M}$, $m_0 \in \mathcal{M}$, and functions $f : \mathcal{M} \times V \to \mathcal{M}$ and $g : \mathcal{M} \times V_i \to V$ such that for all $\boldsymbol{v} = v_0 v_1 \dots v_k$ with $v_i \in V$, we have $\sigma_i(\boldsymbol{v}) = g(f(\dots(f(f(m_0, v_0), v_1) \dots, v_{k-1}), v_k)$. The *memory* of σ_i is said to be $|\mathcal{M}|$, while if $|\mathcal{M}| = 1$, then σ_i is called *memoryless*. Informally, a memoryless strategy does not depend on the history, but only on the current position. A pair of strategies σ_1, σ_2 on $\mathcal{G}$, along with a starting position v, defines a *play* $\pi(\sigma_1, \sigma_2, v)$, which is a word over V. The play $\pi(\sigma_1, \sigma_2, v) = v_0 v_1 \dots$ is defined inductively as follows: (a) $v_0 = v$; (b) $v_{i+1} = \sigma_1(v_0 \dots v_i)$ if $v_i \in V_1$; and (c) $v_{i+1} = \sigma_2(v_0 \dots v_i)$ if $v_i \in V_2$. We define $\Pi(\mathcal{G})$ as the set of all plays on $\mathcal{G}$. Since every position has at least one outgoing edge, every play is indeed infinite.

Labeled Markov chains. A *(labeled) Markov chain* is a tuple $\langle \Sigma, S, s_0, E \rangle$, where Σ is the alphabet of letters, S is a finite set of states, s_0 is an initial state,

$E : S \times \Sigma \times S \mapsto [0,1]$ is the edge probability function, which for every $s \in S$ satisfies that $\sum_{a\in\Sigma, s'\in S} E(s,a,s') = 1$.

Distributions given by Markov chains. Consider a Markov chain $\mathcal{M}$. For every finite word u, the probability of u, denoted $\mathbb{P}_{\mathcal{M}}(u)$, w.r.t. the Markov chain $\mathcal{M}$ is the sum of probabilities of paths labeled by u, where the probability of a path is the product of probabilities of its edges. For basic open sets $u \cdot \Sigma^\omega = \{uw : w \in \Sigma^\omega\}$, we have $\mathbb{P}_{\mathcal{M}}(u \cdot \Sigma^\omega) = \mathbb{P}_{\mathcal{M}}(u)$, and then the probability measure over infinite words defined by $\mathcal{M}$ is the unique extension of the above measure (by Carathéodory's extension theorem [11]). We will denote the unique probability measure defined by $\mathcal{M}$ as $\mathbb{P}_{\mathcal{M}}$.

2.2 Objectives

We consider two types of objectives: quantitative and Boolean. In the following definitions, we consider a game arena $\mathcal{G} = (V, V_1, V_2, E)$.

Quantitative objectives. A *quantitative objective* in general is a Borel measurable function $f : \Pi(\mathcal{G}) \mapsto \mathbb{R} \cup \{-\infty, \infty\}$. Player 1 (called also Minimizer) plays in a way to construct a play π of a possibly small value $f(\pi)$, whereas Player 2 (called also Maximizer) attempts to maximize $f(\pi)$. The minimal value of the game which Player 1 can ensure (called the lower value) is defined as $\underline{\mathsf{val}}(f,v) = \inf_{\sigma_1\in\mathcal{S}_1[\mathcal{G}]} \sup_{\sigma_2\in\mathcal{S}_2[\mathcal{G}]} f(\pi(\sigma_1,\sigma_2,v))$. Player 2 on the other hand can ensure that the value of the game is at least the upper value, denoted as $\overline{\mathsf{val}}(f,v) = \sup_{\sigma_2\in\mathcal{S}_2[\mathcal{G}]} \inf_{\sigma_1\in\mathcal{S}_1[\mathcal{G}]} f(\pi(\sigma_1,\sigma_2,v))$. By Borel determinacy [14], the upper and lower values coincide with respect to f, hence we call their value, the value of the game, and denote it by $\mathsf{val}(f,v)$.

Optimal strategies. Consider a quantitative objective f. A strategy σ for Player 1 (resp., Player 2) is called *optimal* for a position v if and only if we have $\sup_{\sigma_2\in\mathcal{S}_2[\mathcal{G}]} f(\pi(\sigma,\sigma_2,v)) = \mathsf{val}(f,v)$ (resp., $\inf_{\sigma_1\in\mathcal{S}_1[\mathcal{G}]} f(\pi(\sigma_1,\sigma,v)) = \mathsf{val}(f,v)$).

Mean-payoff objectives. The mean payoff objective is defined by a labeling $\mathtt{wt} : E \mapsto \mathbb{Z}$ of edges E on $\mathcal{G}$ with integers. Given a labeling $\mathtt{wt}$ and a play $\pi = v_0 v_1 \ldots$ on $\mathcal{G}$ we define $\textsc{LimAvgInf}^{\mathtt{wt}}(\pi) = \liminf_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \mathtt{wt}(v_{i-1}, v_i)$. We skip the superscript $\mathtt{wt}$, if it is clear from the context.

Average response time objectives. We define the *average response time* (ART) objective based on an *action labeling* act $: E \to \{r, g, \#\}$ that assigns actions to moves. Given a play π on $\mathcal{G}$, we define $\mathsf{rt}_i[\pi]$ as the number of positions between the i-th edge labeled with a request and the first following edge labeled with a grant; if there are no grants past the i-th request, we put $\mathsf{rt}_i[\pi] = \infty$. For a play π with infinite number of requests and grants, we define the quantitative objective $\mathrm{ART}(\pi) = \liminf_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \mathsf{rt}_i[\pi]$. Finally, we put restrictions on the game arena, discussed below, to avoid plays with finitely many requests.

The G-R condition. Observe that the value of a play with infinitely many requests and finitely many grants is infinite, i.e., if Player 1 cannot enforce infinitely many grants, he looses. For plays with finitely many requests, there are several ways to define the value of the play: the average over finitely many requests, or Player 1 (resp., Player 2) wins unconditionally. As we are interested in plays with infinitely many requests, we assume the *grant-request* condition (G-R) on games arenas stating that: every grant is followed by a request in the next step. Then, a sequence with infinitely many grants has infinitely many requests, and if there are finitely many requests, then the last request is never granted and the ART is infinite.

The G-R condition eliminates corner cases, and allows us to focus on the core of the problem. Still, our construction can be adapted to work without this condition (Remark 1).

Quantitative objectives as random variables. The quantitative objectives are measurable functions mapping paths to reals, and thus can be interpreted as random variables w.r.t. the probabilistic space we consider. Given a Markov chain $\mathcal{M}$ and a value function f, we consider the following fundamental quantities:

1. **Expected value**: $\mathbb{E}_f(\mathcal{M})$ is the expected value of the random variable defined by the quantitative objective f w.r.t. the probability measure defined by the Markov chain $\mathcal{M}$.
2. **(Cumulative) distribution**: $\mathbb{D}_{\mathcal{M},f}(\lambda) = \mathbb{P}_{\mathcal{M}}(\{\pi : f(\pi) \leq \lambda\})$ is the cumulative distribution function of the random variable defined by f w.r.t. the probability measure defined by the Markov chain $\mathcal{M}$.

Boolean objectives. A Boolean objective is a function $\Phi : \Pi(\mathcal{G}) \mapsto \{0,1\}$. We consider two types of Boolean objectives: Büchi and threshold. Büchi objectives Φ_B are defined by a subset F of the positions of the arena. Then, $\Phi_B(\pi) = 1$ iff some position from F occurs infinitely often in π. Threshold objectives are defined by imposing a threshold on a quantitative objective, i.e., given a quantitative objective f and a threshold $\boldsymbol{\theta}$, we consider the set of winning plays to be $\{\pi \in \Pi(\mathcal{G}) : f(\pi) \leq \boldsymbol{\theta}\}$, all plays π whose value does not exceed $\boldsymbol{\theta}$. We define the threshold variants of the quantitative objectives $\textsc{LimAvgInf}$, ART as $\textsc{LimAvgInf}^{\leq\lambda} = \{\pi \mid \textsc{LimAvgInf}(\pi) \leq \lambda\}$, and $\text{ART}^{\leq\lambda} = \{\pi \mid \text{ART}(\pi) \leq \lambda\}$.

Winning strategies. A strategy σ_1 (resp., σ_2) is *winning* for Player 1 (resp., Player 2) from a position v iff for all strategies σ_2 for Player 2 (resp., all strategies σ_1 for Player 1), the play π defined by σ_1, σ_2 given v satisfies $\Phi(\pi) = 1$ (resp., $\Phi(\pi) = 0$).

2.3 Computational questions

We present questions, which we study in this paper.

Computational questions for games. Given a Boolean objective Φ (resp., quantitative objective f), a game arena $\mathcal{G}$ and a starting position s_0, we consider the following basic computational questions:

- The *game question* asks to determine the player that has the winning strategy for Φ starting from position s_0.
- The *value question* asks to compute $\mathsf{val}(f, s_0)$.

Computational questions for Markov chains. Given a quantitative objective f and a Markov chain $\mathcal{M}$, we consider the following basic computational questions:

– The *expected question* asks to compute $\mathbb{E}_f(\mathcal{M})$.
– The *distribution question* asks, given a threshold λ, to compute $\mathbb{D}_{\mathcal{M},f}(\lambda)$.

2.4 Previous results

We present existing results on the computational questions for two-player games and Markov chains with mean-payoff objectives. The computational questions for ART objectives have not been studied before; we study ART objectives in the following sections.

Mean-payoff games admit pseudo-polynomial algorithms for solving games and computing the value of the game [1, 16]. The complexity is given w.r.t. the set of positions V, the set of moves E and the maximal absolute value W of the labeling `wt`.

Theorem 1 ([10, 16, 1]). *The following assertions hold:*
– *The game question for mean-payoff games can be solved in $O(|V| \cdot |E| \cdot W)$ time. The winner has a memoryless winning strategy.*
– *The value of a mean-payoff game can be computed in $O(|V|^2 \cdot |E| \cdot W \cdot \log(W \cdot V))$ time. Both players admit optimal memoryless strategies.*

For Markov chains with mean-payoff objectives, basic computational questions can be solved in polynomial time. These questions are solved by reductions to linear programming (LP), and hence the exact complexity depends on the exact complexity of LP. To avoid the discussion on the wide-range of methods to solve LP, we only give the size of the LP instance produced by the reductions.

Theorem 2 ([12]). *For $\mathcal{M} = (\Sigma, S, s_0, E)$, the expected question and the distribution question can be computed in polynomial time, by reduction to linear programming with $|S|$ variables and $|S| + |E| + 1$ constraints.*

3. Games and Graphs with ART Objectives

In this section we study one- and two-player games with the average response time (ART) objective. We establish polynomial-time algorithms to determine the winner in these games as well as polynomial time algorithms for computing the value of the game. We begin with examples showing that both players require memory to play optimally. Then, we establish polynomial-time complexity of two-player games with ART objectives (Section 3.2). Finally, we discuss one-player games with ART objectives (Section 3.3), which inherit a polynomial-time algorithm from the two-player case. We, however, establish better bounds both on the complexity and the memory required to play optimally.

3.1 Memory requirement for ART objectives

We begin with an example showing that Player 1 needs memory to win even in a one-player game with an ART objective.

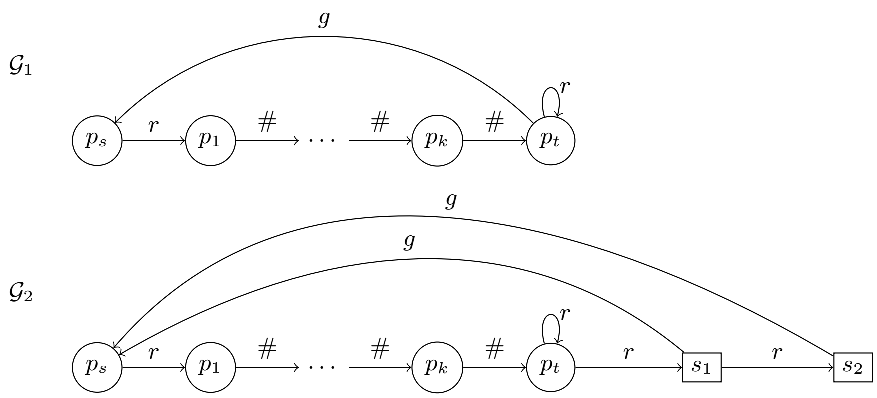

Fig. 2. Examples of games arenas: $\mathcal{G}_1$ where Player 1 requires finite memory to play optimally, and $\mathcal{G}_2$ where Player 2 requires finite memory as well. Circle positions are owned by Player 1 and square ones are owned by Player 2.

Example 2. Consider a game arena $\mathcal{G}_1$ depicted in Figure 2. Player 1 has two memoryless strategies. In the first, he stays forever in p_t, which results in the infinite average response time. In the second, in p_t he always moves to p_s. Observe that this case the average response time is $k + 1$.

Consider a finite-memory strategy, in which, each time Player 1 moves from p_k to p_t, he loops n times in position p_t, and then moves to p_s. This strategy gives a play, which repeats infinitely a cycle of length $k+n+1$, with $n+1$ requests and response times $k+n+1$ for the request issued in (p_s, p_1), and $n, n-1, \ldots, 1$ for requests issued in the loop (p_t, p_t). The ART in this case is $\frac{k+n+1+0.5 \cdot n \cdot (n+1)}{n+1}$, which attains the minimum when $n + 1$ is approximately $\sqrt{2k}$. In such a case, the ART is approximately $\sqrt{2k} + 0.5$, which is smaller than $k + 1$.

Based on Example 2, we can show that Player 2 also requires memory to win against Player 1.

Example 3. Consider a game arena $\mathcal{G}_2$ from Figure 2, which extends $\mathcal{G}_1$ from Example 2. Recall that Player 1 to play optimally has to loop $\sqrt{2k} + 1$ times at position p_t in $\mathcal{G}_1$. Therefore, if Player 1 loops less than $\sqrt{2k}$ times at p_t in $\mathcal{G}_2$, then Player 2 maximizes the average response time by going immediately for a grant, i.e., moving from s_1 to p_s. However, if Player 1 loops more than $\sqrt{2k}$ times at p_t, then Player 2 is better off by delaying a grant even trough issuing a request, i.e., moving from s_1 to s_2 and then to p_s. To play such a strategy, Player 2 requires approximately $\sqrt{2k}$ memory.

3.2 Two-player games with ART objectives

We present the main result of this section.

Theorem 3. *The following assertions hold:*

(1) Let $\lambda = \frac{p}{q}$ with $p, q \in \mathbb{Z}$. The two-player game question with the $\mathrm{ART}^{\leq\lambda}$ *objective can be solved in $O(|V|^7 \cdot |E| \cdot \min(|q|, |V|^3))$ time. The winner has a winning strategy with memory bounded by $2|V|^2$.*
(2) The value of two-player games with quantitative ART *objectives can be computed in $O(|V|^{10} \cdot |E| \cdot \log(|V|))$ time. Both players admit optimal strategies with memory bounded by $2|V|^2$.*

In the remaining part of this section, we prove Theorem 3.
Key ideas. We prove Theorem 3 by reduction to mean-payoff games. We highlight some key ideas of the proof.

1. First, note that for mean-payoff games memoryless strategies are sufficient (see Theorem 1), and, in contrast, memory is required for both players for ART objectives (see Example 2 and Example 3). We present a reduction of games with ART objectives to mean-payoff games that involves a polynomial blow-up, and a blow-up is unavoidable due to the memory requirement.
2. As illustrated in Example 3, both players use memory to track the number of *pending* requests, i.e., the number of requests since the last grant. In the reduction, we encode the number of pending requests in the game arena $\mathcal{G}^N$ (Definition 1). We show that it suffices to count up to $2|V|^2$ pending requests (Lemma 4), which yields $2|V|^2$ upper bound on the required memory to play optimally.
3. The general algorithms for mean-payoff games are pseudo-polynomial. In our reduction, the weights in the game arena $\mathcal{G}^N$ correspond to the number of pending requests, and hence they are bounded by $2|V|^2$. Thus for our reduction the weights are polynomial, and the existing algorithms for mean-payoff games [16, 1] work in polynomial time when applied to our reduction.

Simple case: thresholds $\lambda > |V|$. We proceed with the proof of Theorem 3. First, we observe that games with $\mathrm{ART}^{\leq\lambda}$ objectives can be solved easily if the threshold λ is greater than the number of the positions $|V|$. In such a case Player 1 plays Büchi game to reach grant infinitely often. If Player 1 can win in the Büchi game, he has a memoryless strategy that ensures that ART does not exceed $|V|$. Otherwise, if he fails, Player 2 can force ART to be infinite with a memoryless strategy. Büchi games can be solved in $O(|V|^2)$ time [3, 4], and hence:

Lemma 1. *Let $\mathcal{G} = (V, V_1, V_2, E)$ be a game arena, act be an action labeling, $\lambda \in \mathbb{Q}$. If $\lambda > |V|$, then the game with the objective* $\mathrm{ART}^{\leq\lambda}$ *can be solved in $O(|V|^2)$ time and the winner has a memoryless winning strategy.*

In the following we consider thresholds λ bounded by $|V|$.

Consider a play π. We define the number of pending requests at position i, denoted by $\mathsf{pr}_i[\pi]$, as the number of edges labeled with a request since the last edge labeled with a grant. Observe that, if j is a position of a grant and there are k requests up to position i, then $\sum_{i=1}^{k} \mathsf{rt}_i[\pi] = \sum_{i=1}^{j} \mathsf{pr}_i[\pi]$. Using this observation, we reduce games with the average response time objective to games with the mean-payoff of pending requests. We encode the number of pending requests in the game. To ensure that the game arena is finite, we compute pending requests up to some bound N. The average of (bounded to N) pending requests underapproximates the average response time (Lemma 2). Later on, we show that for N big enough, both values coincide.

Definition 1 (Arenas $\mathcal{G}^N$). *Consider a game arena $\mathcal{G} = (V, V_1, V_2, E)$, an action labeling $act : E \to \{r, g, \#\}$, and $N > 0$. We define a game arena $\mathcal{G}^N$ and a weight labeling wt_λ such that $\mathcal{G}^N = (V^N, V_1^N, V_2^N, E^N)$ and*

$(\mathbf{V^N}, \mathbf{V_1^N}, \mathbf{V_2^N})$: *$V^N = V \times \{0, \ldots, N\}$, $V_1^N = V_1 \times \{0, \ldots, N\}$, and $V_2^N = V_2 \times \{0, \ldots, N\}$,*

$(\mathbf{E^N})$: *for all $v_1, v_2 \in V$, $x, y \in \{0, \ldots, N\}$ we have $(\langle v_1, x\rangle, \langle v_2, y\rangle) \in E^N$ iff $(v_1, v_2) \in E$ and either*
- *$act(v_1, v_2) = r$ and $y = \min(x + 1, N)$, or*
- *$act(v_1, v_2) = g$ and $y = 0$, or*
- *$act(v_1, v_2) = \#$ and $x = y$.*

$(\mathbf{wt}_\lambda)$: *for all $(\langle v_1, x\rangle, \langle v_2, y\rangle) \in E^N$, we define*
- *$wt_\lambda(\langle v_1, x\rangle, \langle v_2, y\rangle) = x$ if $act(v_1, v_2) = r$, and*
- *$wt_\lambda(\langle v_1, x\rangle, \langle v_2, y\rangle) = x + \lambda$ if $act(v_1, v_2) \in \{g, \#\}$.*

Key ideas. Observe that for every play π on an arena $\mathcal{G}$ there exists a unique corresponding play π' in the arena $\mathcal{G}^N$ and vice versa. Indeed, given a play $\pi = v_0v_1v_2$ on $\mathcal{G}$, we transform it into the play π' on $\mathcal{G}^N$ by annotating positions of π' with the number of pending requests bounded to N, i.e., the play $\pi' = (v_0, 0)(v_1, \min(\mathsf{pr}_1[\pi], N))(v_2, \min(\mathsf{pr}_2[\pi], N))\ldots$. To transform a play π' on $\mathcal{G}^N$ to the corresponding play on $\mathcal{G}$ we project out the second component in each position of π'. Finally, we observe that if a play π' is eventually contained in $V \times \{0, \ldots, N - 1\}$, then it records actual numbers of pending requests, not restricted by N, and hence $\mathrm{ART}(\pi) \le \lambda$ if and only if $\textsc{LimAvgInf}^{\mathrm{wt}_\lambda}(\pi') \le \lambda$.

Lemma 2. *Let $\mathcal{G}$ be a game arena and act be an action labeling. For every play π on $\mathcal{G}$ and the corresponding play π' on $\mathcal{G}^N$, we have*
1. *$\mathrm{ART}(\pi) \le \lambda$ implies $\textsc{LimAvgInf}^{wt_\lambda}(\pi') \le \lambda$, and*
2. *if π' eventually stays in $V \times \{0, \ldots, N - 1\}$, then $\mathrm{ART}(\pi) \le \lambda$ if and only if $\textsc{LimAvgInf}^{wt_\lambda}(\pi') \le \lambda$.*

Proof. Consider $k > 0$ and $\epsilon \ge 0$. Let g_k be the position of the first grant following the k-th request. We show that (*) $\frac{1}{k}\sum_{i=1}^{k} \mathsf{rt}_i[\pi] \le \lambda + \epsilon$ implies $\frac{1}{g_k}\sum_{i=1}^{g_k} \mathrm{wt}_\lambda(\pi')[i] \le \lambda + \epsilon$.

Assume that $\frac{1}{k}\sum_{i=1}^{k} \mathsf{rt}_i[\pi] \le \lambda + \epsilon$, then by simple transformation we get $(\sum_{i=1}^{k} \mathsf{rt}_i[\pi]) + (g_k - k)\cdot(\lambda + \epsilon) \le g_k(\lambda + \epsilon)$. Now observe that at the position corresponding to a grant the sum of response times is equal to the sum of pending requests over all positions, i.e., $\sum_{i=1}^{k} \mathsf{rt}_i[\pi] = \sum_{i=1}^{g_k} \mathsf{pr}_i[\pi]$. Recall that $\mathrm{wt}_\lambda(\pi')[i] = min(N, \mathsf{pr}_i[\pi]) + \lambda$ if $\mathrm{act}(v_{i-1}, v_i) \neq r$, and $\mathrm{wt}_\lambda(\pi')[i] = min(N, \mathsf{pr}_i[\pi])$ otherwise. Therefore,

$$\sum_{i=1}^{k} \mathsf{rt}_i[\pi] + (g_k - k)(\lambda + \epsilon) \ge \sum_{i=1}^{g_k} \mathrm{wt}_\lambda(\pi')[i] + (g_k - k)\epsilon \ge \sum_{i=1}^{g_k} \mathrm{wt}_\lambda(\pi')[i]$$

Finally, $\frac{1}{g_k} \sum_{i=1}^{g_k} \mathrm{wt}_\lambda(\pi')[i] \leq \lambda + \epsilon$.

If $\mathrm{ART}(\pi) \leq \lambda$, then there exists a sequence $p[1], p[2], \ldots$ such that for every $n > 0$ we have $\frac{1}{p[n]} \sum_{i=1}^{p[n]} \mathsf{rt}_i[\pi] \leq \lambda + \frac{1}{n}$. Observe that due to (*) for all $n > 0$ we have $\frac{1}{g_{p[n]}} \sum_{i=1}^{g_{p[n]}} \mathrm{wt}_\lambda(\pi')[i] \leq \lambda + \frac{1}{n}$, and hence $\textsc{LimAvgInf}_\lambda^{\mathrm{wt}}(\pi') \leq \lambda$.

Now, assume that (**) past position K, the play π' is contained in $V \times \{0, \ldots, N-1\}$. We first assume that $K = 1$. Consider k and $\epsilon \geq 0$ such that $\frac{1}{g_k} \sum_{i=1}^{g_k} \mathrm{wt}_\lambda(\pi')[i] \leq \lambda + \epsilon$. Then, $\sum_{i=1}^{g_k} \mathrm{wt}_\lambda(\pi')[i] - g_k(\lambda + \epsilon) \leq 0$. Again, $\sum_{i=1}^{k} \mathsf{rt}_i[\pi] = \sum_{i=1}^{g_k} \mathsf{pr}_i[\pi]$. However, condition (**) implies that for $i \geq K = 1$ we have $\mathrm{wt}_\lambda(\pi')[i] = \mathsf{pr}_i[\pi] + \lambda$ if $\mathrm{act}(v_{i-1}, v_i) \neq r$, and $\mathrm{wt}_\lambda(\pi')[i] = \mathsf{pr}_i[\pi]$ otherwise. Therefore, $\sum_{i=1}^{g_k} \mathrm{wt}_\lambda(\pi')[i] = \sum_{i=1}^{k} \mathsf{rt}_i[\pi] + (g_k - k)\lambda$. Finally, $\frac{1}{k} \sum_{i=1}^{k} \mathsf{rt}_i[\pi] \leq \lambda + \frac{g_k}{k} \epsilon$.

Now, if $\textsc{LimAvgInf}_\lambda^{\mathrm{wt}}(\pi') \leq \lambda$, then there exists a sequence $p[1], p[2], \ldots$ such that for all $n > 0$ we have $\frac{1}{g_{p[n]}} \sum_{i=1}^{g_{p[n]}} \mathrm{wt}_\lambda(\pi')[i] \leq \lambda + \frac{1}{n}$. It follows that for all $n > 0$ we have $\frac{1}{p[n]} \sum_{i=1}^{p[n]} \mathsf{rt}_i[\pi] \leq \lambda + \frac{g_{p[n]}}{p[n]} \frac{1}{n}$. We claim that $\frac{g_k}{k}$ is bounded by a constant independent of n, and hence $\mathrm{ART}(\pi) \leq \lambda$. To show that $\frac{g_{p[n]}}{p[n]}$ is bounded by a constant, consider $m_r, m_g, m_\#$ denoting the number of respectively requests, grants and null instructions up to position $g_{p[n]}$. Observe that $g_{p[n]} = m_r + m_g + m_\#$ and $m_r = p[n]$. Condition (G-R), i.e., every grant is immediately followed by a request, implies that $m_g \leq m_r$. Again, by condition (G-R) all moves labeled with $\#$ follow some pending request and hence the weight of such moves is at least $\lambda + 1$. Therefore, to have $\frac{1}{g_{p[n]}} \sum_{i=1}^{g_{p[n]}} \mathrm{wt}_\lambda(\pi')[i] \leq \lambda + 0.5$ (for $n > 2$), the following inequality must hold $m_\# < (4\lambda + 2) \cdot p[n]$. Thus, $\frac{g_{p[n]}}{p[n]} \leq \frac{m_\# + 2p[n]}{p[n]} \leq 4(\lambda + 1)$, i.e., $\frac{g_k}{k}$ is bounded.

Finally, note that even if $K > 0$, a finite prefix does not affect the limit of $\frac{1}{p[n]} \sum_{i=1}^{p[n]} \mathsf{rt}_i[\pi]$. □

Lemma 2 implies that winning with the $\mathrm{ART}^{\leq \lambda}$ objective on $\mathcal{G}$ implies winning with the objective $\textsc{LimAvgInf}^{\leq \lambda}$ on $\mathcal{G}^N$ for every N. Next, we prove a cutoff result saying that for $N \geq 2|V|^2$, winning on $\mathcal{G}^N$ with the objective $\textsc{LimAvgInf}^{\leq \lambda}$ is equivalent to winning with the $\mathrm{ART}^{\leq \lambda}$ objective on $\mathcal{G}$.

Key ideas. To prove the cutoff result, we consider a winning strategy for Player 1 on $\mathcal{G}^N$ with the objective $\textsc{LimAvgInf}^{\leq \lambda}$. Without loss of generality, we can assume that this strategy is memoryless [10]. We show that every for $N > 2|\mathcal{G}|^2$, every memoryless winning strategy on $\mathcal{G}^N$ that wins for $\textsc{LimAvgInf}^{\leq \lambda}$ must ensure that each play eventually stays in $V \times \{0, \ldots, N-1\}$. Therefore, such a strategy is also winning for $\mathrm{ART}^{\leq \lambda}$ on $\mathcal{G}$ (Lemma 2).

Lemma 3. *Let $\mathcal{G} = (V, V_1, V_2, E)$ be a game arena, act be an action labeling, $\lambda \in \mathbb{Q}$ and let $N \geq 2|V|\lambda$. If Player 1 wins on $\mathcal{G}^N$ with the objective $\textsc{LimAvgInf}^{\leq \lambda}$, then he has a memoryless winning strategy that ensures that each play eventually stays in $V \times \{0, \ldots, N-1\}$.*

Proof. If Player 1 wins on $\mathcal{G}^N$ with the objective $\textsc{LimAvgInf}^{\leq \lambda}$, then he has a memoryless winning strategy σ_1 [10]. Assume towards contradiction that for some play π consistent with σ_1 some position from $G \times \{N\}$ is reachable infinitely often. Consider a graph $\mathcal{G}^N[\sigma_1]$ obtained from arena $\mathcal{G}^N$ by fixing edges of Player 1 according to strategy σ. The nodes of $\mathcal{G}^N[\sigma_1]$ are all positions of $\mathcal{G}^N$. We observe that $\mathcal{G}^N[\sigma_1]$ has a cycle C that contains a position from $G \times \{N\}$ and its length is bounded by the number of nodes of $\mathcal{G}^N[\sigma_1]$, i.e., $|C| \leq |V| \cdot N$.

If cycle C does not contain grants, then C is contained in $G \times \{N\}$, and hence the average weight in C is at least $N > \lambda$. Thus, σ_1 is not winning. Therefore, C contains grants and hence it visits nodes from $G \times \{0\}$. Thus, it contains at least one node from each set $G \times \{i\}$ for $i = 0, 1, \ldots, N$. This gives us that C contains N transitions with weights $0, 1, \ldots, N$. The remaining transitions have the weight at least 0. It follows that the average weight of the cycle C is at least

$$\frac{1}{|C|}(\frac{N \cdot (N+1)}{2}) = \frac{N+1}{2|V|} \geq \frac{2|V|\lambda + 1}{2|V|} > \lambda$$

Thus, σ_1 is not winning. A contradiction. □

We are ready to prove Theorem 3.

Proof (of Theorem 3). Let $\mathcal{G} = (V, V_1, V_2, E)$. If $\lambda > |V|$, then by Lemma 1, we can decide in $O(|V|^2)$ time whether Player 1 has a winning strategy, and if he does he has a memoryless winning strategy.

Assume that $\lambda \leq |V|$ and let $N = 2|V|^2$. Lemmas 2 and 3 imply the following condition (**):

(***) Player 1 wins on $\mathcal{G}$ with the objective $\text{ART}^{\leq \lambda}$ if and only if Player 1 wins on $\mathcal{G}^N$ with the objective $\text{LimAvgInf}^{\leq \lambda}$.

For the implication from left to right, consider a winning strategy σ on $\mathcal{G}$ with the objective $\text{ART}^{\leq \lambda}$. Player 1 can use the strategy σ to play on $\mathcal{G}^N$ with the objective $\text{LimAvgInf}^{\leq \lambda}$. Indeed, consider a play π on $\mathcal{G}$ consistent with σ such that $\text{ART}(\pi) \leq \lambda$. Then, Lemma 2 states that for the corresponding play π' on $\mathcal{G}^N$ we have $\text{LimAvgInf}^{\text{wt}_\lambda}(\pi') \leq \lambda$. Now, to show the implication from right to left we consider a winning strategy σ on $\mathcal{G}^N$. By Lemma 3, we can assume that σ is memoryless and each play eventually stays in $V \times \{0, \ldots, N-1\}$. Let σ' be a projection of σ on the first component V, i.e., σ' is a strategy on $\mathcal{G}$. Observe that (2) of Lemma 2 implies that each play consistent with σ' is winning for $\text{ART}^{\leq \lambda}$ and hence σ' is a winning strategy on $\mathcal{G}$ with the objective $\text{ART}^{\leq \lambda}$. Since σ is memoryless, the memory of σ' is $N = 2|V|^2$.

Condition (***) implies that, if any player can win with the $\text{ART}^{\leq \lambda}$ objective, then the memory necessary to win is bounded by $2|V|^2$. In particular, for the minimal threshold λ_0, for which Player 1 has a winning strategy, he has a winning strategy with memory bounded by $2|V|^2$. This strategy is the optimal strategy for the quantitative ART objective on $\mathcal{G}$, and hence Player 1 admits optimal strategies with memory bounded by $2|V|^2$. Similarly, for any $n > 0$ and the objective $\text{ART}^{\leq \lambda_0 - \frac{1}{n}}$, Player 2 has a winning strategy with memory bounded by $2|V|^2$. There are finitely many such strategies and some strategy

σ_o occurs infinitely often. This strategy σ_o is optimal for Player 2 and it has memory bounded by $2|V|^2$.

We now discuss the value of the minimal threshold, for which Player 1 has a winning strategy, which is the value of the game. Consider a strategy σ for Player 1 with memory bounded by $2|V|^2$. We construct a graph G for Player 2 resulting from fixing in $\mathcal{G}$ all choices of Player 1 according to σ and storing its memory. Such a graph has $2|V|^3$ vertexes and no cycles without a grant, as otherwise Player 2 wins for every $\lambda > 0$. Now, the ART in that graph can be computed as follows. We examine all simple cycles in G that begin with a move labeled with a grant, compute the maximal ART over all such cycles, and denote it by T. Observe that the maximal ART over all paths in G equals T. Indeed, we can construct a path of the ART equal T, and conversely any (finite) path can be split into simple cycles that begin with a move labeled with a grant. Therefore, ART over finite prefixes of any infinite paths does not exceed T. Now, observe that simple cycles in G have length bounded by $2|V|^3$ and hence T is a rational number of the form $\frac{p}{q}$, where $q \leq 2|V|^3$. Now, the value of the ART game on $\mathcal{G}$ is the minimum over values of ART on graphs resulting from fixing a strategy σ with memory $2|V|^2$ for Player 1. Therefore, the value of ART game on $\mathcal{G}$ is a rational number of the form $\frac{p}{q}$, where $q \leq 2|V|^3$ and $p < 2|V|^4$.

The game on $\mathcal{G}^N$ with the objective $\textsc{LimAvgInf}^{\leq \lambda}$ can be solved in time $O(nmM)$, where n (resp., m) is the number of positions (resp., moves) of $\mathcal{G}^N$ and M is the bound on the absolute values of weights in $\mathcal{G}^N$ [1]. Recall that $n = |V|N$, $m = |E|N$. Theorem 1 assumes integer weights, and hence for $\lambda = \frac{p}{q}$, we need to multiply all weights by q. However, if $q > 2|V^3|$, the above discussion implies that we can approximate λ by the greatest fraction $\frac{p}{2|V|^3}$ and hence $M = N \cdot \min(q, 2|V^3|)$. Thus, the game can be solved in $O((|V|N) \cdot (|E|N) \cdot N \cdot \min(q, 2|V^3|)) = O(|V|^7 \cdot |E| \cdot \min(q, |V^3|))$. Finally, using the binary search on the possible values of λ and $\text{ART}^{\leq \lambda}$ objective we can find the value of the ART game on $\mathcal{G}$ in $O(|V|^{10}|E| \log(|V|))$. □

3.3 Graphs with ART objectives

In the previous section, we established a polynomial-time algorithm for two-player games with ART objectives. However, if we restrict games to a single player case, we can improve the polynomial bounds.

1. First, the blow-up in the reduction to mean-payoff games is only quadratic in the one-player case.
2. Second, one-player mean-payoff games can be solved in $O(|V||E|)$ time, which is better than pseudo-polynomial bound $O(|V||E|W)$ for two-player mean-payoff games.
3. Third, in one-player case, we establish linear bounds on memory necessary to play optimally (resp., win) with ART objectives (resp., $\text{ART}^{\leq \lambda}$ objectives), which is better than the quadratic bound in the two-player case.

Theorem 4. *The following assertions hold:*

(1) The one-player game question for games $(V, V, \emptyset, E)$ (resp., $(V, \emptyset, V, E)$) with $\mathrm{ART}^{\leq\lambda}$ objective can be solved in $O(|V|^3|E|)$ time. Player 1 (resp., Player 2) has a winning strategy with memory bounded by $2|V|$ (resp., $|V|$).
(2) The value of one-player games (graphs) with quantitative ART *objective can be computed in $O(|V|^3|E|\log(|V|))$ time. Player 1 (resp., Player 2) admits optimal strategies with memory bounded by $2|V|$ (resp., $|V|$).*

The main improvement is the cutoff result for the one-player case (Lemma 4), which is a counterpart of Lemma 3. We prove this result by a pumping argument. Having Lemma 4, we establish the complexity of one-player games with the $\mathrm{ART}^{\leq\lambda}$ objective.

Lemma 4. *Let $\mathcal{G} = (V, E)$ an a one-player game arena, act be an action labeling, $\lambda \in \mathbb{Q}$ and let $N > |V| + \lambda$. If there exists a play π on $\mathcal{G}^N$ satisfying the objective $\textsc{LimAvgInf}^{\leq\lambda}$, then there exists a memoryless play satisfying the objective $\textsc{LimAvgInf}^{\leq\lambda}$ that stays in $V \times \{0, \ldots, N-1\}$.*

Proof. Let π be a play such that $\textsc{LimAvgInf}(\pi) \leq \lambda$. Observe that when the number of pending requests exceeds λ, then the weight of every move until the following grant exceeds λ. Therefore, shortening the blocks of π in which the number of pending requests exceeds λ decreases all the partial averages. More precisely, let i be a position at which the number of pending requests exceeds λ and $j > i$ be the position of the following grant. Assume that $j - i > |V|$. Then, we can project $\pi[i, j]$ onto its first component (positions of $\mathcal{G}$), remove all the cycles, and lift the resulting path to the path of $\mathcal{G}^N$ starting in $\pi[i]$. We call this final path ρ and we observe that $|\rho| \leq |V|$ and $\pi' = \pi[1, i - i]\rho\pi[j + 1, \infty]$ is a play on $\mathcal{G}^N$ such that all the partial averages are bounded by the partial averages of π. Finally, the number of pending grants between i and $i + |\rho|$ is bounded by $\lambda + |V|$. We list all the positions i, where the number of pending requests exceeds λ, and we iteratively apply the above procedure to all these positions. In the result we obtain a play π^F that satisfies the objective $\textsc{LimAvgInf}^{\leq\lambda}$ and the number of pending requests is always bounded by $|V| + \lambda$, i.e., π_F stays in $V \times \{0, \ldots, N-1\}$. □

We are ready to prove Theorem 4.

Proof (of Theorem 4). **(1)**: Let $\mathcal{G} = (V, E)$ as all positions belong to one player. If $\lambda > |V|$, then by Lemma 1, we can decide in $O(|V|^2)$ the game question, and if the player has a winning strategy, then he has a memoryless winning strategy.

Assume that $\lambda \leq |V|$ and let $N = 2|V|$. First, we consider the case of all positions belonging to Player 1. Lemmas 2 and 4 imply that, in one-player games, winning on $\mathcal{G}$ with the objective $\mathrm{ART}^{\leq\lambda}$ is equivalent to winning on $\mathcal{G}^N$ with the objective $\textsc{LimAvgInf}^{\leq\lambda}$. Moreover, the winning strategy for $\mathcal{G}$ can be obtained from the winning strategy on $\mathcal{G}^N$ by projecting out the second component. Since $\textsc{LimAvgInf}^{\leq\lambda}$ admits memoryless winning strategies, to win on $\mathcal{G}$ with the objective $\mathrm{ART}^{\leq\lambda}$ it suffices to consider strategies with memory bounded by $N = 2|V|$. To decide whether Player 1 wins we prune $\mathcal{G}^N$ to positions reachable from the given initial position, which takes $O(|V|N+|E|N)$ time, and we compute the minimal mean cycle [13] in time $O(|V|N \cdot |E|N) = O(|V|^3|E|)$.

Now, consider the case of all positions belonging to Player 2. If there exists a cycle in $\mathcal{G}$ that does not contain grants, then Player 2 can win against $\mathrm{ART}^{\leq \lambda}$ objective for any λ. We can check the existence of such a cycle in $O(|V|+|E|)$ and Player 2 has a memoryless winning strategy in that case. Otherwise, if every cycle contains at least one grant, the number of pending requests is bounded by $|V|$, and hence Player 2 requires $|V|$ memory. Thus, for $N = |V|+1$, all plays are contained in $V \times \{0, \ldots, N-1\}$ and Lemma 2 implies that Player 2 wins against $\mathrm{ART}^{\leq \lambda}$ objective if and only if she wins on $\mathcal{G}^N$ against the objective $\textsc{LimAvgInf}^{\leq \lambda}$. Now, we prune $\mathcal{G}^N$ to positions reachable from a given initial position, which takes $O(|V|N+|E|N)$ time, and we compute the maximal mean cycle [13] in time $O(|V|N \cdot |E|N) = O(|V|^3|E|)$. This maximal cycle has the average grater than λ if and only if Player 2 wins on $\mathcal{G}^N$ against $\textsc{LimAvgInf}^{\leq \lambda}$. The latter is equivalent to Player 2 winning on $\mathcal{G}$ against $\mathrm{ART}^{\leq \lambda}$ objective.

(2): We present the argument for Player 1, as the reasoning for Player 2 is virtually the same. For every $\lambda > 0$, if Player 1 has a winning strategy with $\mathrm{ART}^{\leq \lambda}$, then he has a winning strategy with memory $2|V|$. There are finitely many such strategies and one of them achieves the value of the game. A one-player strategy amounts to a single play, which is a lasso of length bounded by $2|V|^2$. Therefore, the minimal threshold λ_0 such that Player 1 has a winning strategy with $\mathrm{ART}^{\leq \lambda_0}$ belongs to a finite set of rationals $\{\frac{p}{q} \mid p, q \in \mathbb{N}, q \leq 2|V|^2, p \leq 2|V|^3\}$. Therefore, using the binary search and the decision procedure from (1), we can find the minimal λ_0, which is the value of the ART game in $O(|V|^3|E|\log(|V|))$. Finally, observe that the strategy for Player 1 for $\mathrm{ART}^{\leq \lambda_0}$ is the optimal strategy for him. Thus, Player 1 admits optimal strategies with memory bounded by $2|V|$. □

3.4 Discussion

We discuss the applicability and significance of the results on ART objective.

Remark 1 (Discussion on the G-R condition). We have introduced the G-R condition for technical simplicity. We can, however, eliminate it. Observe that the G-R condition has been used only in Lemma 2, which relates plays with ART objectives on $\mathcal{G} = (V, V_1, V_2, E)$ and plays with mean-payoff objectives on $\mathcal{G}^N$.

First, without the G-R condition, there can be plays, in which eventually there are no pending requests. Assume that such plays are winning for Player 1. Then, we proceed as follows:

- We show that (*) if Player 2 has a winning, strategy she has a winning strategy such that length of blocks (of positions) with no pending requests are bounded by $|V|$.
- We redefine $\mathcal{G}^N$ such that after $|\mathcal{G}|$ steps with no pending requests Player 1 wins. The size of such modified arena is $|V| \cdot N + |V|^2$.

– We prove the analogue of Lemma 2 for the modified $\mathcal{G}^N$. Observe that the current proof of Lemma 2 works even if we only assume that blocks (of positions) with no pending requests are bounded by $|\mathcal{G}|$.

The above construction also works if Player 2 wins on plays, in which eventually there are no pending requests.

Remark 2 (Discussion on complexity). In this work, our goal is to establish the first polynomial-time algorithms computing the ART property for game graphs and graphs. The complexities of the polynomial upper bounds we establish are quite high ($\widetilde{O}(|V|^{10} \cdot |E|)$ for game graphs, $\widetilde{O}(|V|^3 \cdot |E|)$ for graphs), and likely to be non-optimal. Our algorithms for games are based on reductions to mean-payoff games, where memoryless strategies are sufficient. We show that quadratic size memory is sufficient for ART objectives. Hence a reduction to mean-payoff games, which encodes memory in the state space, gives rise to a game with $|V|^3$ vertices, $|V|^2 \cdot |E|$ edges, and $W = |V|^2$, and then applying the best-known algorithms for mean-payoff games already gives a high polynomial complexity. Obtaining algorithms with better theoretical bounds as well as practical approaches are interesting directions for future work.

4. Markov Chains

In this section, we discuss Markov chains with ART objectives. We establish polynomial-time algorithms for both the expected value and the distribution questions.

Polynomial-time algorithms for Markov chains with objectives given by nested weighted automata (which can express the ART property) has been established in [8]. Hence, below we present the key ideas to obtain a simple algorithm for ART properties. We omit formal and detailed proofs, which are consequences of the results established in [8].

Key ideas. We present the key ideas for both cases.

1. *The expected question.* Consider a labeled Markov chain $\mathcal{M} = \langle \Sigma, S, s_0, E \rangle$, where $\Sigma = \{r, g, \#\}$. To compute the expected value $\mathbb{E}_{\mathrm{ART}}(\mathcal{M})$, we first compute the labeling of transitions $\mathtt{wt}$ of $\mathcal{M}$ such that for all $s_1, s_2 \in S$, we put $\mathtt{wt}((s_1, g, s_2)) = \mathtt{wt}((s_1, \#, s_2)) = \bot$, i.e., no weight, and $\mathtt{wt}((s_1, r, s_2))$ is the expected number of steps to reach a grant. This labelling can be computed in polynomial time in $|\mathcal{M}|$, by reduction to linear programming with $|S|$ variables and $|S| + |E| + 1$ constraints [12]. Then, we compute the expected value of the mean-payoff objective $\textsc{LimAvgInf}^{\mathtt{wt}}$ on $\mathcal{M}$, i.e., $\mathbb{E}_{\textsc{LimAvgInf}}(\mathcal{M})$. The value $\mathbb{E}_{\textsc{LimAvgInf}}(\mathcal{M})$ can be computed in polynomial time. Again, it is computed by reduction to linear programming with $|S|$ variables and $|S|+|E|+1$ constraints [12]. Finally, we return $\mathbb{E}_{\textsc{LimAvgInf}}(\mathcal{M})$ as $\mathbb{E}_{\mathrm{ART}}(\mathcal{M})$. The key aspect of the correctness proof is that the values $\mathbb{E}_{\textsc{LimAvgInf}}(\mathcal{M})$ and $\mathbb{E}_{\mathrm{ART}}(\mathcal{M})$ are equal, which follows from [8, Lemma 26].
2. *The distribution question.* To compute the distribution question, we first discuss the case of Markov chains $\mathcal{M}$ consisting of a single recurrent set, i.e.,

almost all paths visit all states infinitely often. In such a case, the Boolean objective $\mathrm{ART}^{\leq\lambda}$ is a tail event [11] and its probability is either 0 or 1, i.e., for every λ almost all plays satisfy $\mathrm{ART}^{\leq\lambda}$ or almost all plays violate it. Therefore, almost all plays have the same value, which is $\mathbb{E}_{\mathrm{ART}}(\mathcal{M})$. In the general case, we can find in $\mathcal{M}$ subsets $R_1, \ldots, R_k$, which are recurrent sets, i.e., among paths that enter R_i, almost all paths visit all states of R_i infinitely often. We compute all recurrent sets $R_1, \ldots, R_k$ of $\mathcal{M}$ in $O(|S| + |E|)$ time. Then, we compute (in polynomial time) the probabilities $p_1, \ldots, p_k$ of reaching each of these sets from s_0, and expected values $\mathbb{E}_{\mathrm{ART}}(R_1), \ldots, \mathbb{E}_{\mathrm{ART}}(R_k)$, where $\mathbb{E}_{\mathrm{ART}}(R_i)$ is the expected average response time of the Markov chain $(\Sigma, R_i, s_0^i, E \cap R_i \times R_i)$ with some $s_0^i \in R_i$. Probabilities $p_1, \ldots, p_k$ can be computed using linear programming as well. Finally, $\mathbb{D}_{\mathcal{M},\mathrm{ART}}(\lambda)$, the probability of the set of plays below threshold λ, is the sum of probabilities of reaching the recurrent sets with expected values below λ, i.e., $\mathbb{D}_{\mathcal{M},\mathrm{ART}}(\lambda) = \sum\{p_i \mid \mathbb{E}_{\mathrm{ART}}(R_i) \leq \lambda\}$. The correctness proof follows from [8, Lemma 27].

Theorem 5. *Consider a Markov chain $\mathcal{M} = \langle \Sigma, S, s_0, E\rangle$ and $\lambda \in \mathbb{Q}$.*

- *The expected value $\mathbb{E}_{\mathrm{ART}}(\mathcal{M})$ for the* ART *objective can be computed in polynomial time, by a reduction that takes $O(|S| + |E|)$ time and produces two instances of linear programming each with $|S|$ variables and $|S|+|E|+1$ constraints.*
- *The cumulative distribution $\mathbb{D}_{\mathcal{M},\mathrm{ART}}(\lambda)$ for the* ART *objective can be computed in polynomial time, by a reduction that takes $O(|S| + |E|)$ time and produces three instances of linear programming each with $|S|$ variables and $|S| + |E| + 1$ constraints.*

5. Conclusions

Average response time (ART) is a fundamental quantitative property of reactive systems. We presented the first algorithms that are designed specifically for computing ART values on graphs, game graphs, and Markov chains. All our algorithms are polynomial time. There are several interesting directions for future work. First, while our main objective was to establish polynomial-time upper bounds, algorithms of better complexity may be possible (Remark 2). Second, the problems of computing ART values for more general graph models such as Markov decision processes (i.e., graphs with both probabilistic and nonprobabilistic vertices) and stochastic games (i.e., graphs with probabilistic vertices, Player-1 vertices, and Player-2 vertices) are still open. Finally, the value computation problems remain open for interesting generalizations of the ART property such as the more general ART property which counts the number of tick events between request and grant events, rather than counting the number of all transitions between requests and subsequent grants. While these generalizations of the ART property and of the underlying graph models appear modest, the algorithms presented in this paper cannot be generalized directly to these cases.

References

1. Lubos Brim, Jakub Chaloupka, Laurent Doyen, Raffaella Gentilini, and Jean-François Raskin. Faster algorithms for mean-payoff games. *Formal Methods in System Design*, 38(2):97–118, 2011.
2. Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM TOCL*, 11(4):23, 2010.
3. Krishnendu Chatterjee and Monika Henzinger. An $O(n^2)$ time algorithm for alternating Büchi games. In *SODA 2012*, pages 1386–1399. ACM-SIAM, 2012.
4. Krishnendu Chatterjee and Monika Henzinger. Efficient and dynamic algorithms for alternating Büchi games and maximal end-component decomposition. *J. ACM*, 61(3):15:1–15:40, 2014.
5. Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Bidirectional nested weighted automata. To appear at CONCUR 2017.
6. Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Nested weighted automata. In *LICS 2015*, pages 725–737, 2015.
7. Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Nested weighted limit-average automata of bounded width. In *MFCS 2016*, pages 24:1–24:14, 2016.
8. Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Quantitative automata under probabilistic semantics. In *LICS 2016*, pages 76–85, 2016.
9. Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. Springer, 1st edition, 2009.
10. Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, 1979.
11. W. Feller. *An introduction to probability theory and its applications*. Wiley, 1971.
12. Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer, 1996.
13. Richard M Karp. A characterization of the minimum cycle mean in a digraph. *Discrete mathematics*, 23(3):309–311, 1978.
14. Donald A Martin. Borel determinacy. *Annals of Mathematics*, pages 363–371, 1975.
15. John A McDermid. *Software engineer's reference book*. Elsevier, 2013.
16. Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1):343–359, 1996.