Bidding Games on Markov Decision Processes with Quantitative Reachability Objectives

Guy Avni
University of Haifa
Haifa, Israel
gavni@cs.haifa.ac.il

Martin Kurečka
Masaryk University
Brno, Czechia
martin.kurecka@fi.muni.cz

Kaushik Mallik*
IMDEA Software Institute
Madrid, Spain
kaushik.mallik@imdea.org

Petr Novotný
Masaryk University
Brno, Czechia
petr.novotny@fi.muni.cz

Suman Sadhukhan
University of Haifa
Haifa, Israel
ssadhukh@campus.haifa.ac.il

ABSTRACT

Graph games are fundamental in strategic reasoning of multi-agent systems and their environments. We study a new family of graph games which combine stochastic environmental uncertainties and auction-based interactions among the agents, formalized as bidding games on (finite) *Markov decision processes* (MDP). Normally, on MDPs, a single decision-maker chooses a sequence of actions, producing a probability distribution over infinite paths. In bidding games on MDPs, two players—called the *reachability* and *safety* players—bid for the privilege of choosing the next action at each step. The reachability player's goal is to maximize the probability of reaching a given target vertex, whereas the safety player's goal is to minimize it. These games generalize traditional bidding games on *graphs*, and the existing analysis techniques do not extend. For instance, the central property of bidding games on graphs is the existence of a *threshold* budget, which is the necessary and sufficient budget to guarantee winning for the reachability player. For MDPs, the threshold becomes a *relation* between budgets and probabilities of reaching the target. We devise value-iteration algorithms that approximate thresholds and optimal policies for general MDPs, and compute the exact solutions for acyclic MDPs, and show that finding thresholds is at least as hard as *simple-stochastic games*.

KEYWORDS

Graph Games; Bidding Games; Markov decision processes

ACM Reference Format:
Guy Avni, Martin Kurečka, Kaushik Mallik, Petr Novotný, and Suman Sadhukhan. 2025. Bidding Games on Markov Decision Processes with Quantitative Reachability Objectives. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

*Part of the research was done when the author was at the Institute of Science and Technology Austria (ISTA), Austria.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

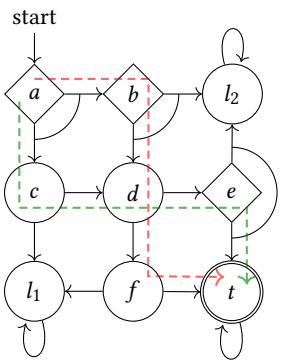

Figure 1: A bidding game on an MDP. The control and random vertices are denoted as circular and diamond-shaped, respectively. The probabilistic transitions (marked with arcs) use the uniform distribution. The target for the reachability player is t. The dashed paths are the viable reachability policies for the setting of Ex. 1.1.

1 INTRODUCTION

Graph games are fundamental for reasoning about strategic interactions between agents in multi-agent systems, with [31, 33, 42] or without [3] external environments. Environments, when present, are commonly modeled using stochastic processes, like *Markov decision processes* (MDP) [41] in reinforcement learning (single-agent), and *stochastic games* in the multi-agent setting [22, 26, 30].

We study games which combine stochastic environments with auction-based interactions among players, formalized as *bidding games on MDP arenas*. An MDP is a graph whose vertices are partitioned into *control* vertices and *random* vertices, and the game involves the players moving a token along the edges of the graph. The rules of the game are as follows. The two players are allocated initial budgets, normalized in a way that their sum is 1. When the token reaches a control vertex, an auction is held to determine who chooses where the token goes next. In these auctions, the players simultaneously submit bids from their available budgets, the higher bidder moves the token and pays his bid amount to the lower bidder. When the token reaches a random vertex, it automatically moves to one of the successors according to the transition probabilities of the MDP (without affecting the budgets of the players).

We consider the *quantitative reachability* objectives, where the goal of the first player, called the *reachability player*, is to maximize the probability that a given target vertex is reached, and the goal of the second player, called the *safety player*, is to minimize it.

Example 1.1. Consider the game in Fig. 1 and the initial budget allocation $\langle 0.75+\epsilon, 0.25-\epsilon \rangle$ for the two players, where $\epsilon \in (0, 0.25)$

is arbitrary. Notice that if the game reaches l_1, l_2, the target t can no longer be reached. We show how the reachability player can reach t with probability at least 0.5. Initially, the token moves randomly from a. If it reaches c, the reachability player avoids l_1 by winning the auction with the bid 0.25 (which exceeds the opponent's budget) and moving to d with new budgets $\langle 0.5+\epsilon, 0.5-\epsilon\rangle$. At d, he bids 0.5, to force the game to e, from which t is reached with probability 0.5. If, on the other hand, the token moves to a from b, with probability 0.5 it moves to l_2 and the reachability player loses. If the token reaches d, since no biddings were made, the budgets remain $\langle 0.75+\epsilon, 0.25-\epsilon\rangle$. At d, the reachability player bids 0.25, proceeds to f with budgets $\langle 0.5+\epsilon, 0.5-\epsilon\rangle$, and then bids 0.5 to force the game to t. Thus, each path b, d, f, t and c, d, e, t, after a, occur with probability 0.5, and the total probability to reach t from a is 0.5.

Bidding games on MDPs generalize traditional bidding games on *graphs*, i.e., on MDPs without any random vertices. Bidding games on graphs have a rich pedigree, going back to the seminal work of Lazarus et al. [34, 35], followed by a series of extensions to various payment schemes [7, 8, 10, 11], non-zero-sum games [36], bidding games with discrete budgets [2, 15, 16, 28], partial-information games [12], and bidding games with charging [6].

We point out a distinction of our setup from the traditional one. In the traditional setup, winning policies are memoryless, i.e., they choose the same successor from each vertex upon winning the bidding. As seen in the MDP in Ex. 1.1, the reachability player's move at d depends on the path used to reach d.

Applications. Our results have an immediate application in *auction-based scheduling* [14], which is a *decentralized* multi-objective decision-making framework. In this framework, we are given an arena, modeling the environment, a pair of specifications φ_0, φ_1, and we want to compute a pair of policies σ_0, σ_1 that will be composed at runtime with the policies bidding against each other at each step for choosing the next action. The goal is to synthesize σ_0, σ_1 such that their runtime composition fulfills $\varphi_0 \wedge \varphi_1$. The advantage is modularity, where the policies can be independently designed, and if one specification changes, only the relevant policy needs to be updated. The synthesis algorithms in auction-based scheduling solve two *independent* zero-sum bidding games for σ_0 and σ_1 (on the same arena). As zero-sum bidding games have been studied only on graphs, auction-based scheduling is restricted to graph arenas until now. Our work will lead to a decentralized solution of multi-objective reachability on MDPs using auction-based scheduling.

Fair resource allocation studies how to allocate a collection of items to a set of agents in a *fair* manner, where various definitions of fairness exist [4, 18]. Bidding games naturally create a fair allocation mechanism [19, 36], namely allocate an initial budget to each agent, fix an ordering of the items, and hold a bidding for each item one by one. Bidding games on MDPs offer *resource-allocation under stochastic uncertainties* [17, 20]: now the agents bid, but the outcome is uncertain (e.g., in online advertisements, the higher bidder gets the ad-slot, but the number of viewers remains stochastic).

Bidding games on graphs. We briefly survey results on traditional bidding games on graphs [34, 35]. A central quantity in these games are the *threshold budgets*: every vertex v is associated with a value

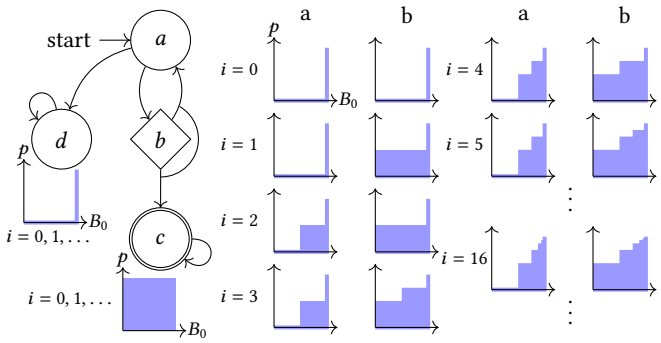

Figure 2: Value iteration for the game on the *left* with the sequence of reachability values (c is the target) on the *right*.

$\mathit{Th}(v) \in [0, 1]$ such the reachability player wins from v if his initial budget is strictly larger than $\mathit{Th}(v)$, and loses (i.e., the safety player wins) if it is strictly smaller than $\mathit{Th}(v)$. Furthermore, *pure* policies suffice for both players. Interestingly, computing thresholds is equivalent to solving a class of *stochastic games* [26] called *random-turn games* [39]: For a bidding game $\mathcal{G}$, suppose $RT(\mathcal{G})$ represents the random-turn game on $\mathcal{G}$ where who moves the token at each turn is determined uniformly at random. Then the optimal probability with which the reachability player wins in $RT(\mathcal{G})$ from a vertex v equals $1 - \mathit{Th}(v)$. This reduction implies that computing thresholds for bidding games is in NP ∩ co-NP. An opposite reduction is unknown.

Our results: bidding games on MDPs. We prove thresholds exist for bidding games on MDPs, though their shapes become significantly more complex and reasoning becomes more advanced than bidding games on graphs. This is because thresholds are now *binary relations* between budgets and probabilities, where $(B, p) \in \mathit{Th}(v)$ if the reachability player can reach the target with every probability $p' < p$ and every budget $B' > B$, and the safety player can avoid the target with every probability $(1 - p') > (1 - p)$ when the reachability player's budget is $B' < B$. We develop a value-iteration algorithm to find thresholds as described in the next example.

Example 1.2. Fig. 2 illustrates our value-iteration algorithm. Intuitively, the shaded area in the plot at iteration $i \in \mathbb{N}$ depicts all the necessary budgets of the reachability players and the respective probabilities of reaching the target c in at most i steps. For example, for $i = 4$ at a, if the reachability player has a budget in $(0.5, 0.75]$, he can win only one bidding (in a) and reach c with probability up to 0.5 in 2 steps (abc), and if he has a budget in $(0.75, 1]$, he can win two biddings (in a) and reach c with probability up to 0.75 in 4 steps ($ababc$). Every other path to c is longer than 4 steps. In the limit, the "plots" tend to thresholds (Thm. 5.1), which for a is $\{(B, p) \mid \exists n \geq 1 \,.\, B \in (1 - 2^{-n}, 1 - 2^{-(n+1)}] \wedge p \leq 1 - 2^{-n}\}$.

We summarize our results below. We consider the problem of deciding if the reachability player can reach the target in a given MDP with a given probability p using a given budget B. **(I)** For *general MDPs*, the solution remains unknown. Under the assumption that (B, p) is not exactly on the threshold, we show that the problem is decidable. The time and space complexity of our algorithm depends

on the distance ϵ of (B, p) from the threshold (infinity norm), the minimum probability $\delta_{\min}$ of any random edge, and the number of vertices $|V|$, and is given as $O\left(\frac{|V|^2}{\epsilon}\log(1/\epsilon)^3\,\delta_{\min}^{-4|V|}\right)$. Our decision procedure uses an approximated value iteration algorithm to limit the computational complexity. **(II)** For *acyclic MDPs*, the reachability problem is decidable in $O(|V|^{|V|})$ time and space, for tree-shaped MDPs, it is decidable in NP ∩ co-NP.

The above assumption that the point (B, p) does not lie exactly on the threshold is natural in the context of multi-objective decision-making. For instance, for "classical" (non-bidding) multi-objective stochastic games, algorithms for determining the winning player assume that the target payoff vector does not lie on the boundary of the Pareto set of achievable payoffs [5].

The proofs omitted due to limited space can be found in the extended version of this paper [13].

Further Related Work. As mentioned earlier, one of our motivations is solving multi-objective problems on MDPs via the auction-based scheduling approach. The alternative approach is to directly synthesize a single policy achieving an acceptable tradeoff between the individual objectives as was studied for MDPs [23, 29, 38], stochastic games [5, 24], and reinforcement learning [1, 21, 32, 37]. We are the first to study quantitative reachability objectives in bidding games on MDPs. For sure winning, simple reductions to bidding games on graphs are known [9].

2 PRELIMINARIES OF MARKOV DECISION PROCESSES (MDP)

Syntax. An MDP is a tuple $\langle V, V_c, V_r, E, \delta\rangle$, where V is a finite set of vertices, V_c and V_r are the *control* and *random* vertices such that $V_c \cup V_r = V$ and $V_c \cap V_r = \emptyset$, $E\colon V_c \to 2^{V_r}$ is the *control transition function*, and $\delta\colon V_r \to \Delta(V_c)$ is the *random transition function*, where $\Delta(V_c)$ is the set of all probability distributions over V_c. The set of *successors* of vertex v will be denoted as $Succ(v)$, where $Succ(v) \coloneqq E(v)$ if $v \in V_c$ and $Succ(v) \coloneqq \{v' \in V \mid \delta(v)(v') > 0\}$ if $v \in V_r$. A vertex v is called *sink* if $Succ(v) = \{v\}$.

Convention for figures. MDPs are depicted as transition diagrams with circular nodes representing control vertices and diamond-shaped nodes representing random vertices. Target vertices are depicted in double circles. If the random transitions have uniform distributions, they are marked using an arc between them.

Semantics. Semantics of MDPs are summarized below; details can be found in standard textbooks [40]. A *path* of an MDP starting at a given vertex $v \in V$ is a sequence $v^0v^1\ldots$ with $v^0 = v$ and every $v^{i>0}$ being a successor of v^{i-1}. Paths can be either finite or infinite. We write $Paths_{\text{fin}}(M)$ and $Paths_{\text{inf}}(M)$ to denote, respectively, the set of all finite and infinite paths, and write $Paths^c_{\text{fin}}(M)$ to denote the set of all finite paths that end in a control vertex. A *scheduler* is a function $\theta\colon Paths^c_{\text{fin}}(M) \to V$ mapping every finite path $\rho = v^0\ldots v^k$ ending at the control vertex $v^k \in V_c$ to one of its successors; i.e., $\theta(\rho) \in E(v^k)$. Every scheduler θ induces a unique probability distribution $\mathbb{P}^{M,\theta}_v(\cdot)$ over the paths of M with initial vertex v.

Specifications. A *specification* φ over an MDP M is a set of infinite paths of M. We will consider reachability and safety specifications of both bounded and unbounded variants, defined below. Given a set of vertices $T \subseteq V$ called the *target* vertices, and an integer $h > 0$, the *bounded-horizon reachability* specification is the set of paths that visit T *in at most h steps*, i.e., $Reach^{M,h}(T) \coloneqq \{v^0v^1\ldots \in Paths_{\text{inf}}(M) \mid \exists i \le h\,.\, v^i \in T\}$. The (unbounded) *reachability* specification is the set of paths that *eventually* visit T; i.e., $Reach^M(T) \coloneqq \bigcup_h Reach^{M,h}(T)$. Dually, given a set of vertices $U \subseteq V$ called the *unsafe* vertices, and a number $h > 0$, the *bounded-horizon safety* specification is the set of paths that avoid U *for at least h steps*, i.e., $Safe^{M,h}(U) \coloneqq \{v^0v^1\ldots \in Paths_{\text{inf}}(M) \mid \forall i \le h\,.\, v^i \notin U\}$. The (unbounded) *safety* specification is the set of paths that *always* avoid U; i.e., $Safe^M(U) \coloneqq \bigcap_h Safe^{M,h}(U)$. Reachability and safety specifications—with bounded and unbounded horizons—are complementary to each other, i.e., for every $h > 0$, $Reach^{M,h}(T) = V^\omega \setminus Safe^{M,h}(T)$, and $Reach^M(T) = V^\omega \setminus Safe^M(T)$.

3 BIDDING GAMES ON MDP-S

On a given M, we consider a zero-sum "token game" between two players, who will be referred to as the *reachability* and *safety* players. Initially, the token is placed in a given *initial* vertex, and the players are allocated budgets (positive real numbers) whose sum is 1. As convention, we will only specify the reachability player's budget as B, and the safety player's budget will be implicit (i.e., $1 - B$).

Recall that a game constitutes the two players bidding for the privilege of moving the token from the control vertices: the higher bidder chooses the successor and pays the bid amount to the lower bidder.[1] On the other hand, from every random vertex v, the token moves according to the distribution $\delta(v)(w)$, and the budgets of the players remain unaffected. The game continues forever, generating a probability distribution over infinite paths. The reachability player wants to maximize the probability of reaching T, while the safety player wants to minimize it. We formalize this below.

Policies and paths. A policy of a player is a function of the form $[0,1] \times Paths^c_{\text{fin}}(M) \to [0,1] \times V$, mapping every pair of available budget B and finite path $v^0\ldots v^k$ to a pair of a bid value $b \le B$ and a successor of v^k. We will write σ and τ to represent the policy of the reachability and the safety player, respectively.

Suppose we are given an initial vertex v and an initial budget B of the reachability player (recall that the safety player's initial budget will be $1 - B$). We will call the pair $\langle v, B\rangle$ the *initial configuration*. Every pair of policies (σ, τ) and the initial configuration $\langle v, B\rangle$ induce a scheduler $\theta(\sigma, \tau, B)$ as follows: if the current path is $\rho \in Paths^c_{\text{fin}}(M)$ and the current budget of the reachability player is B', then, denoting $\sigma(B', \rho) = (b_R, u)$ and $\tau(1 - B', \rho) = (b_S, w)$, we define the scheduler θ as follows:

- if $b_R \ge b_S$,[2] i.e., if the reachability player wins the bidding, then $\theta(\sigma, \tau, B)(\rho) = u$, and his new budget is $B' - b_R$, and
- if $b_R < b_S$, i.e., if the safety player wins the bidding, then $\theta(\sigma, \tau, B)(\rho) = w$, and the reachability player's new budget is $B' + b_S$.

We will write $\mathbb{P}^{\sigma,\tau}_{v,B}$ instead of $\mathbb{P}^{\theta(\sigma,\tau,B)}_v$ to denote the probability distribution over the set of infinite paths starting at vertex v.

[1] This bidding mechanism is known as Richman bidding in the literature [34]. Other bidding mechanisms also exist, but they are left as part of future works.

[2] We assume, arbitrarily, that ties go in favor of the reachability player. In our proofs, we show that it does not matter how ties are resolved.

Winning conditions. Let $\langle v, B\rangle$ be an initial configuration, φ be a reachability specification (bounded or unbounded), and $p \in [0,1]$ be the *required probability* for the reachability player to satisfy φ. A *winning policy of the reachability player* is a policy σ such that for every policy τ of the safety player, it holds that $\mathbb{P}^{\sigma,\tau}_{v,B}(\varphi) \geq p$. Dually, a *winning policy of the safety player* is a policy τ such that for every policy σ of the reachability player, it holds that $\mathbb{P}^{\sigma,\tau}_{v,B}(\varphi) \leq p$. Winning policies of each player are formalized in a way as if the opponent can see them before choosing the responses. This a standard practice and eliminates the need to capture concurrent actions of players [27]. The sets of winning policies of reachability and safety players will be respectively denoted as $\Pi_{\mathsf{R}}(B,p,v,\varphi)$ and $\Pi_{\mathsf{S}}(B,p,v,\varphi)$, and if φ is clear, as $\Pi_{\mathsf{R}}(B,p,v)$ and $\Pi_{\mathsf{S}}(B,p,v)$.

Thresholds. In traditional reachability bidding games on graphs, where probabilities are unnecessary, the threshold of a vertex v is the budget B such that the reachability player wins from v with every budget $B' > B$, and loses with every budget $B' < B$. In bidding games on MDPs, thresholds generalize to relations over budgets and probabilities: The threshold of v is the set of all pairs (B,p) such that the reachability player wins with every budget greater than B and required probability less than p, and loses with every budget less than B and required probability larger than p.

Definition 3.1 (Threshold). For a given vertex v, the *threshold* of v, written Th_v, is the set of all pairs (B,p) such that $\Pi_{\mathsf{R}}(B',p',v)$ is nonempty whenever $B' > B$ and $p' < p$, and $\Pi_{\mathsf{S}}(B',p',v)$ is nonempty whenever $B' < B$ and $p' > p$.

A central question in traditional bidding games is whether thresholds exist, because then it can be *determined* which of the players will win based on the budget allocation, as long as the budget is not exactly equal to the threshold. In our case, the existence question of thresholds generalizes to the question of *whether the threshold completely separates* the winning points of the two players.

Definition 3.2 (Completely separating thresholds). The threshold of v is *completely separating* if for every point $(B,p) \notin \mathit{Th}_v$,

- there exists $(B',p') \in \mathit{Th}_v$ such that either $B < B'$ and $p > p'$, or $B > B'$ and $p < p'$, and
- exactly one of the sets $\Pi_{\mathsf{R}}(B,p,v)$ and $\Pi_{\mathsf{S}}(B,p,v)$ is nonempty.

One of our main results, Cor. 5.2, asserts that completely separating thresholds indeed exist. Thus, for almost all points, precisely one of the players can win regardless of the opponent's strategy.

The algorithmic question. We define *problem instances* as tuples of the form $\langle M, v, T, B, p\rangle$, where M is an MDP, $\langle v, B\rangle$ is the initial configuration, T is the target, and p is the required probability of satisfying the reachability specification $\mathit{Reach}^M(T)$. The subject of this paper is how to decide who wins in a given problem instance.

Problem 1 (Quantitative reachability). *Let $\langle M, v, T, B, p\rangle$ be a problem instance. For a given $j \in \{\mathsf{R}, \mathsf{S}\}$, decide if the set $\Pi_j(B,p,v,\mathit{Reach}^M(T))$ is nonempty.*

If $\Pi_{\mathsf{R}}(B,p,v,\mathit{Reach}^M(T)) \neq \emptyset$, our decision procedure will produce the witness winning policy for the reachability player as a byproduct; construction of winning policies for the safety player is solved for acyclic MDPs, and remains open for general MDPs. We will assume that T is a set of sinks, which is without loss of any generality since the game ends as soon as T is reached.

4 BOUNDED-HORIZON VALUE ITERATION

We start with the bounded-horizon variant of Prob. 1 with horizon h. In this setting, we propose a 2-dimensional value iteration algorithm for deciding who wins the game.

For reachability, for each vertex v, our algorithm computes a monotonically increasing (with respect to "$\subseteq$") sequence of "values" $\text{r-}val^0_v, \ldots, \text{r-}val^h_v \subseteq [0,1]^2$, where $\text{r-}val^i_v$ will be shown to represent the set of all $(\overline{B}, p)$ such that for every $B > \overline{B}$, the reachability player can reach T from the initial configuration $\langle v, B\rangle$ with probability at least p in at most i steps. Dually, for safety, for each vertex v, our algorithm computes a monotonically decreasing sequence of "values" $\text{s-}val^0_v, \ldots, \text{s-}val^h_v \subseteq [0,1]^2$, where $\text{s-}val^i_v$ will be shown to represent the set of all $(\overline{B}, p)$ such that for every $B < \overline{B}$, the safety player can avoid T from the initial configuration $\langle v, B\rangle$ with probability at least $1-p$ for at least i steps.

Clearly, if v is in T, the target T will be "reached" in zero steps, no matter what (B,p) is, and therefore every $(\overline{B}, p)$ belongs to $\text{r-}val^0_v$. In contrast, if v is *not* in T, the target T will be reached in zero steps only with probability $p = 0$. The points with $\overline{B} = 1$ are trivially included to $\text{r-}val^0_v$ as well, because $\{B \mid B > \overline{B} = 1\} = \emptyset$. By duality, the definition of $\text{s-}val^0_v$ is exactly the opposite.

We now consider the case of $i > 0$ and $v \notin T$. We take the perspective of the reachability player; the case of safety is similar. Consider the following two cases. (a) Suppose $v \in V_{\mathsf{r}}$. Since there is no bidding in v, the budgets of the players at v remain unaffected after the transition. For a fixed budget, if p_w is the probability of reaching T in $i-1$ steps from the successor w (of v), then the probability of reaching T in i steps from v becomes $\sum_w p_w \cdot \delta(v)(w)$. (b) Now suppose $v \in V_{\mathsf{c}}$. For a fixed probability p, we can ask for the least budget needed to reach T from v. If B_+ and B_- are the maximum and minimum budgets required from any successor to reach T in $i-1$ steps with probability p, then the budget required at v for i-step reachability is $B = (B_+ + B_-)/2$. This is because the reachability player can bid $(B_+ - B_-)/2$ from his budget B and make sure that, regardless of the outcome of the bidding, he has enough budget in the next step to reach T in $i-1$ steps: either he wins the bidding and has budget B_-, in which case he must move to the vertex associated to B_-, or he loses the bidding and has budget B_+. This is formalized in the value iteration algorithm presented below.

$$\text{r-}val^0_v \coloneqq \begin{cases} [0,1]^2 & \text{if } v \in T, \\ [0,1]\times\{0\} \cup \{1\}\times[0,1] & \text{otherwise,} \end{cases} \tag{1}$$

$$\text{s-}val^0_v \coloneqq \begin{cases} [0,1]\times\{1\} \cup \{0\}\times[0,1] & \text{if } v \in T, \\ [0,1]^2 & \text{otherwise,} \end{cases} \tag{2}$$

and for $i > 0$, $\text{r-}val^i_v \coloneqq \mathcal{T}_v\left(\left\{\text{r-}val^{i-1}_w \mid w \in \mathit{Succ}(v)\right\}\right)$ and $\text{s-}val^i_v = \mathcal{T}_v\left(\left\{\text{s-}val^{i-1}_w \mid w \in \mathit{Succ}(v)\right\}\right)$, where the operator $\mathcal{T}_v$ is defined as follows: If $v \in V_{\mathsf{r}}$,

$$\mathcal{T}_v\left(\{val_w \mid w \in \mathit{Succ}(v)\}\right) \coloneqq \bigcup_{B\in[0,1]} \left\{ \left(B, \sum_{w\in\mathit{Succ}(v)} \delta(v)(w)\cdot p_w \right) \middle| \forall w \in \mathit{Succ}(v)\ .\ (B,p_w) \in val_w \right\} \tag{3}$$

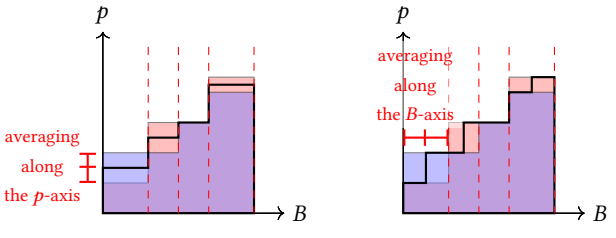

Figure 3: Illustration of the $\mathcal{T}_v$ operator for when $v \in V_r$ (left) and $v \in V_c$ (right). In both cases, we assume there are two successors whose values from the $(i-1)$-th iteration are given as the red and blue regions. The outputs of $\mathcal{T}_v$ is shown as the set with thick boundaries. For $v \in V_r$ (left), we assume uniform transition probabilities (i.e., 0.5 for each successor).

and if $v \in V_c$,

$$\mathcal{T}_v(\{val_w \mid w \in Succ(v)\}) := \bigcup_{p\in[0,1]} \left\{ \left(\frac{B_+ + B_-}{2}, p \right) \middle| \begin{array}{l} B_- \in \{B \mid \exists w \in Succ(v) \,.\, (B,p) \in val_w\}, \\ B_+ \in \{B \mid \forall w \in Succ(v) \,.\, (B,p) \in val_w\} \end{array} \right\} \tag{4}$$

Intuitively, the $\mathcal{T}_v$ operator averages the value sets of the successors along the p axis for random vertices and along the B axis for control vertices. Fig. 3 illustrates this, and Fig. 2 illustrates the value iteration algorithm. The following theorem states that the problem of deciding which player has a winning policy for the horizon i reduces to membership queries for the sets r-val^i and s-val^i computed by the above value iteration algorithm, giving us a sound and complete procedure for determining the winner.

Theorem 4.1. *Let $\langle M, v, T, B, p\rangle$ be a problem instance and $i \in \mathbb{N}$ be the horizon. The following hold:*

(A) $\Pi_R(B, p, v, Reach^{M,i}(T)) \neq \emptyset$ if and only if there exists $\overline{B} < 1$ such that $\overline{B} \leq B$ and $(\overline{B}, p) \in$ r-val^i_v.

(B) $\Pi_S(B, p, v, Reach^{M,i}(T)) \neq \emptyset$ if and only if there exists $\overline{B} > B$ such that $(\overline{B}, p) \in$ s-val^i_v, or $B = 1$ and $(B, p) \in$ s-val^i_v.

For the sake of space, we show the proof of only the "if" direction of claims (A) and (B), moreover assuming that there exists $\overline{B}$ from the statement such that $\overline{B} \neq B$; this will show what the winning policies look like. The full proof is in the extended version [13].

Proof of the "if" directions given $\overline{B} \neq B$. We simultaneously consider both (A) and (B), hence we use *val* to denote either r-*val* or s-*val*. Let $(\overline{B}, p) \in val^i_v$ and suppose the reachability player's initial budget is $B = \overline{B} + s$ with $s > 0$ for r-*val* and $s < 0$ for s-*val*.

The winning policy can be inductively extracted from the computed values val^i_v. If $i = 0$, the claim is trivially true since satisfaction of $Reach^{M,0}(T)$ deos not depend on the budget nor chosen policies. Otherwise, assume $i > 0$ and first discuss the case when $v \in V_c$. In the first step, π identifies B_- and B_+ as defined in (4) such that $(B_- + B_+)/2 = \overline{B}$ and bids $|B_+ - B_-|/2$. If π wins the bidding, it moves the token to the successor w for which $(B_-, p) \in val^{i-1}_w$ and gives the bid to the opponent yielding the new reachability player budget $B' = B_- + s$. Otherwise, the opponent moves the token to any successor w yielding the new budget $B' = B_+ + s$. By definition of B_+, $(B_+, p) \in val^{i-1}_w$ for any choice of w. Therefore, regardless of the new vertex w, π has enough budget to continue according to a policy in $\Pi_j(B', p, w, Reach^{M,i-1}(T))$ which is nonempty by the induction hypothesis (here j is R if *val* is r-*val* and S if *val* is s-*val*).

Now suppose $v \in V_r$. By the inductive definition of val^i_v in (3), for each $w \in Succ(v)$ there exists p_w with $(B, p_w) \in val^{i-1}_w$, such that $\sum_{w\in Succ(v)} \delta(v)(w) \cdot p_w = p$. By the inductive hypothesis, for each $w \in Succ(v)$, $\Pi_j(B, p_w, w, Reach^{M,i-1}(T))$ contains a policy π_w, which can be followed by π regardless of the outcome. □

From the construction of policies in the (partial) proof of Thm. 4.1, it follows that for any given vertex v and any given budget, the choice of the policy will depend on the probability p_v and the horizon length i remaining for satisfying the bounded-horizon specification. It is clear that both p_v and i will depend on the *path* followed to reach v; for instance, in Ex. 1.1, at d, if b was visited earlier, then p_d is 1, whereas if c was visited earlier, then p_d is 0.5 (the horizon i is 2 in both cases). This makes policies implicitly history-dependent.

The computability of the values at each step follows from their finite representations. In particular, the sets r-val^i_v and s-val^i_v have a "staircase form" (see Fig. 3) and can be represented by the corner points of the steps. Before formalizing this, define the order $\prec$ as $(B, p) \prec (B', p')$ if and only if $B \geq B'$ and $p \leq p'$. A $\prec$-downward (or $\prec$-upward) closure of a set S is the set of all points (B, p) such that there exists $(B', p') \in S$ with $(B, p) \prec (B', p')$ (or $(B', p') \prec (B, p)$).

Lemma 4.2. *Let M be an MDP, T be target vertices, and v be a vertex in M. For every $i \in \mathbb{N}$, there exists a finite set $G \subseteq [0, 1]^2$ of at most $3|V|^i$ points such that* r-val^i_v *is the $\prec$-downward closure of G, and* s-val^i_v *is the $\prec$-upward closure of G. Moreover, all boundary points of* r-val^i_v *belong to* s-val^i_v*, and vice versa.*

The sets r-val^0_v and s-val^0_v are downward and upward closures of $\{(0, 0), (0, 1), (1, 1)\}$ or $\{(0, 0), (1, 0), (1, 1)\}$, depending on whether v is in T or not. Thus they indeed have the staircase shape with a single step. The proof of Lem. 4.2 [13] requires showing that the staircase shape propagates through the operator $\mathcal{T}_v$.

A direct consequence of Lem. 4.2 is that the sets r-val^i_v and s-val^i_v can be computed in exponential time and space for every i.

Corollary 4.3. *The sets* r-val^i_v *and* s-val^i_v *for each v and i can be computed in $O(|V|^i)$ time and $O(|V|^i)$ space.*

Another consequence is determinacy, i.e., one of the players always fulfills the respective bounded-horizon specification.

Corollary 4.4. *Let $\langle M, v, T, B, p\rangle$ be an arbitrary problem instance. For every i, the point (B, p) belongs to at least one of the sets* r-val^i_v *and* s-val^i_v.

5 FROM BOUNDED TO UNBOUNDED HORIZON

5.1 Limiting Behavior of Value Iteration

If we continue the bounded-horizon value iteration for increasing horizon, we obtain the following values in the limit:

$$\text{r-}val^*_v := \text{cl}\left[\bigcup_{i=0}^{\infty} \text{r-}val^i_v\right] \text{ and } \text{s-}val^*_v := \bigcap_{i=0}^{\infty} \text{s-}val^i_v,$$

where cl $[S]$ denotes the closure of a set S in the Euclidean metric; note that s-val^*_v is closed by construction. In this section, we establish a connection between r-val^*_v, s-val^*_v and the true values that are winning for the respective player in the unbounded horizon setting. In particular, we relate the limit sets to the threshold $\mathcal{T}h_v$, and present an algorithm for Prob. 1 that runs in doubly exponential time. Our proof also constructs the winning policy for the reachability player, if one exists; the construction of the safety player's winning policy remains open. In the following, we will use the notation $\langle S \rangle$ to denote the interior of the set S.

Theorem 5.1. *Let $\langle M, v, T, B, p \rangle$ be a problem instance. The following hold:*

(A) $(B,p) \in \langle \text{r-}val^*_v \rangle \Rightarrow \Pi_{\mathrm{R}}(B,p,v,Reach^M(T)) \neq \emptyset$,
(B) $(B,p) \in \langle \text{s-}val^*_v \rangle \Rightarrow \Pi_{\mathrm{S}}(B,p,v,Reach^M(T)) \neq \emptyset$.

Proof. Let $(B,p) \in \langle \text{r-}val^*_v \rangle$. Then, for some $i \in \mathbb{N}$, it is already in the interior of r-val^i_v. Since every reachability policy winning on a finite horizon also wins on the infinite horizon, Thm. 4.1 implies $\Pi_{\mathrm{R}}(B,p,v,Reach^M(T))$ is non-empty.

Now, assume (B,p) is in the interior of s-val^*_v, so there exists $\overline{B} = B + s$ for some $s > 0$ with $(\overline{B},p) \in$ s-val^*_v. The safety player follows a policy τ that, besides the current budget, maintains a requested probability of avoiding T, initially p. Since s-val^*_v is a fixpoint of $\mathcal{T}$ [13], the safety player, when in $v \in V_{\mathrm{c}}$, determines B_+ and B_- from eq. (4) such that $(B_- + B_+)/2 = \overline{B}$ and bids $(B_- - B_+)/2$. If she wins, the budget increases to $B_- - s$ and she moves to w where $(B_-,p) \in$ s-val^*_w. Otherwise, the budget decreases to $B_+ - s$ or less and the reachability player can select any successor w. By definition of B_+, $(B_+,p) \in$ s-val^*_w regardless of the choice.

For $v \in V_{\mathrm{r}}$, the safety player determines p_w for each successor w such that $\sum_{w \in Succ(v)} \delta(v)(w) \cdot p_w = p$ and $(\overline{B},p_w) \in$ s-val^*_w. Upon moving to w, she updates the requested probability to p_w.

Since each step preserves the invariant $(\overline{B}',p') \in$ s-val^*_w, where (B',p') is either (B_-,p), (B_+,p), or $(\overline{B},p_w)$, the above rule can iterate indefinitely. Moreover, as $(B',p') \in$ s-val^*_w implies $(B',p') \in$ s-val^i_w for all $i \in \mathbb{N}$, policy τ essentially coincides with the policy from Thm. 4.1 for any finite i. Thus, $\tau \in \Pi_{\mathrm{S}}(B,p,v,Reach^{M,i}(T))$ for every i. If there was a reachability player policy σ such that $\mathbb{P}^{\sigma,\tau}_{v,B}(Reach^M(T)) = q > p$, then, for some k, we must have $q \geq \mathbb{P}^{\sigma,\tau}_{v,B}(Reach^{M,k}(T)) > p$, contradicting $\tau \in \Pi_{\mathrm{S}}(B,p,v,Reach^{M,k}(T))$. Thus, no such σ exists, proving τ is in $\Pi_{\mathrm{S}}(B,p,v,Reach^M(T))$. □

We now characterize the threshold and show that it is completely separating, thereby establishing determinacy.

Corollary 5.2 (Determinacy). *For every MDP M, target vertices T, and vertex v in M, it holds that* r-$val^*_v \cap$ s-$val^*_v = \mathcal{T}h_v$. *Moreover, the threshold is completely separating.*

5.2 On the Decidability of the Quantitative Reachability Problem

From Thm. 5.1, it follows that Problem 1 reduces to the membership problem of deciding which of the sets r-val^*_v and s-val^*_v contain the given pair (B,p). Unfortunately, the decidability of this question remains open. If we were using the value iteration trying to answer decidability, the difficulty comes from the situation when $(B,p) \in \mathcal{T}h_v$, because the true $\mathcal{T}h_v$ is obtained only in the limit, and we are not guaranteed to decide the membership of (B,p) in finite time. To circumvent this "edge case," we make the following assumption.

Assumption 1. *The problem instance $\langle M, v, T, B, p \rangle$ is such that the given pair (B,p) does not belong to $\mathcal{T}h_v$.*

In other words, we assume that the pair (B,p) lies either in the *interior* of r-val^*_v or in the *interior* of s-val^*_v. For every point $(B,p) \in \langle \text{r-}val^*_v \rangle$, Lem. 5.3 below provides an upper bound on the iteration index after which (B,p) will be included inside r-val^i_v; and excluded from s-val^i_v, respectively.

We first introduce some notation. We use $\delta_{\min}$ to denote the minimum positive transition probability in M unless all transitions are non-probabilistic; in that case we set $\delta_{\min} = \frac{1}{2}$. We further denote by $d_\infty(x,y)$ and $d_\infty(x,Y)$ the L_∞ distance of a point x to another point y and to the set Y. Of importance is the situation when either $x \in$ r-val^*_v and $Y =$ s-val^*_v, or $x \in$ s-val^*_v and $Y =$ r-val^*_v, in which case $d_\infty(x,Y)$ is called the *distance of x from v's threshold*. Given the sets $X, Y \subseteq [0,1]^2$, the Hausdorff distance $d_h(X,Y)$ between X and Y is the largest distance one needs to travel starting from any point in either X or Y to reach the closest point in the other set. Formally,

$$d_h(X,Y) := \max\left\{ \max_{y \in Y} \min_{x \in X} \|x-y\|_\infty, \max_{x \in X} \min_{y \in Y} \|x-y\|_\infty \right\}.$$

It can be shown that there exist optimal policies that admit a short path to T regardless of the game history, and that after exponentially many random choices, such a path is traversed with sufficiently high probability. This idea yields the following lemma.

Lemma 5.3. *Let $\langle M, v, T, B, p \rangle$ be a problem instance and suppose (B,p) is in the interior of* r-val^*_v *with its distance from v's threshold being $\epsilon > 0$. Then for every n_ϵ such that*

$$n_\epsilon \geq 4 \log(2/\epsilon)\, |V| \delta_{\min}^{-2|V|},$$

it holds that $(B,p) \in$ r-$val^{n_\epsilon}_v$.

We obtain the following algorithm for Prob. 1 under Assump. 1.

Algorithm: AlgExact
Repeat for $i = 0, 1, 2, \ldots$:
(1) If $(B,p) \in$ r-val^i_v, return "$(B,p) \in \langle \text{r-}val^*_v \rangle$".
(2) If $d_\infty((B,p), \text{r-}val^i_v) > 2e^{-i\delta_{\min}^{2|V|}/4|V|}$, return "$(B,p) \in \langle \text{s-}val^*_v \rangle$."

It is easy to see that AlgExact is sound: If it returns that $(B,p) \in$ r-val^*_v, i.e., if (1) happens before (2), then soundness follows from the fact that r-$val^i_v \subseteq$ r-val^*_v for each i. If it returns that $(B,p) \in$ s-val^*_v, i.e., if (2) happens before (1), then it follows from Lemma 5.3 that (B,p) is farther away from r-val^i_v than the threshold of v. Therefore, r-val^*_v does not contain (B,p), and from Cor. 5.2, it follows that s-val^*_v must contain (B,p). Despite soundness, AlgExact is a semi-decision procedure, because, in general, neither (1) nor (2) is guaranteed to be triggered after finitely many i, as has been explained earlier. Luckily, termination is guaranteed when Assump. 1 holds, in which case AlgExact is sound and complete.

Theorem 5.4. *AlgExact is a semi-decision procedure for Prob. 1, and is sound and complete when Assump. 1 is fulfilled by the given*

problem instance. For the latter case, if the distance between (B, p) *and* v*'s threshold is* $\epsilon > 0$*, then AlgExact terminates in at most* $O\left(\log(1/\epsilon)\,|V|\delta_{\min}^{-2|V|}\right)$ *iterations, yielding the space and time complexity* $|V|^{O\left(\log(1/\epsilon)|V|\delta_{\min}^{-2|V|}\right)}$.

5.3 Abstraction for Complexity Reduction

To reduce the high computational complexity of AlgExact, we present an approximation algorithm that is sound and complete in the sense of Thm. 5.4, while the running time is exponential in $|V|$. The high complexity of AlgExact stems from the arbitrarily high precision of the value set representations. The idea of our algorithm is to discretize the two dimensions—budget and probability—using a fixed finite precision. The approximate value iteration begins by rounding off r-val_v^0 or s-val_v^0 (whether rounding up or down depends on the dimension and the value set) to the closest discrete level along each dimension. At each subsequent step, first the $\mathcal{T}_v$ operator (defined in (4) and (3)) is applied on the approximate values from the previous iteration. Since $\mathcal{T}_v$ may produce value sets with higher precision than the chosen one, the obtained sets are further rounded off again to the closest discrete level along each dimension. We show that the procedure yields a sufficiently low approximation error that depends on the number of iterations and the chosen precision. We formalize the abstract safety value iteration below; the abstract reachability values are later obtained through duality.

Let $\alpha \in (0, 1]$ be a constant parameter, called the *grid size*, such that $1/\alpha$ is an integer. We define a uniform grid X that divides the value space $[0, 1]^2$ in squares of length α, i.e.,

$$X := \{[p\alpha, (p+1)\alpha] \times [q\alpha, (q+1)\alpha] \mid 0 \le p, q < 1/\alpha\}.$$

Given a relation $S \subseteq [0, 1]^2$ over the budgets and probabilities, define $\Delta_\alpha(S)$ as the *over-approximation* of S using the elements of X, i.e., $\Delta_\alpha(S) := \bigcup \{x \in X \mid x \cap S \neq \emptyset\}$.

For every $v \in V$, the abstract safety value iteration computes the sequence $[\![\text{s-}val]\!]_v^0 \supseteq [\![\text{s-}val]\!]_v^1 \supseteq \ldots$, where

$$[\![\text{s-}val]\!]_v^0 := \Delta_\alpha(\text{s-}val_v^0),$$
$$\forall i > 0: \quad [\![\text{s-}val]\!]_v^i := \Delta_\alpha \circ \mathcal{T}_v\left(\left\{[\![\text{s-}val]\!]_w^{i-1} \mid w \in Succ(v)\right\}\right).$$

Henceforth, we will refer to $[\![\text{s-}val]\!]_v^i$ as *abstract* values, and s-val_v^i as *concrete* values. The following lemma shows that the abstract values always over-approximate the concrete values, and the approximation error remain bounded by $\alpha(i+1)$ for every i.

Lemma 5.5. *For every* $v \in V$ *and every* i, $[\![\text{s-}val]\!]_v^i \supseteq \text{s-}val_v^i$*, and, moreover,* $d_h([\![\text{s-}val]\!]_v^i, \text{s-}val_v^i) \le \alpha(i+1)$*. The abstract value* $[\![\text{s-}val]\!]^i$ *can be computed in* $2|V|^2 i/\alpha$ *time and* $2|V|/\alpha$ *space.*

Given an abstract safety value, we can define an abstract reachability value $[\![\text{r-}val]\!]_v^i := [0, 1]^2 \setminus [\![\text{r-}val]\!]_v^i \cup \text{r-}val_v^0$. Note that $[\![\text{r-}val]\!]_v^i$ is under-approximation of r-val_v^i since both, the complement of $[\![\text{s-}val]\!]_v^i$ and r-val_v^0, are subsets of r-val_v^i. Moreover, whenever $d_h\left([\![\text{s-}val]\!]_v^i, \text{s-}val_v^i\right) \le \overline{\epsilon}$ then $d_h\left([\![\text{r-}val]\!]_v^i, \text{r-}val_v^i\right) \le \overline{\epsilon}$.

We now present AlgApprox, an improved algorithm for Prob. 1. AlgApprox uses the same principle as AlgExact, but instead of using the exact value iteration, uses the approximate counterpart along with an iterated refinement of the resolution of the grid.

Algorithm: AlgApprox
Repeat for $h = 0, 1, 2, \ldots$:

(1) Set $\overline{\epsilon} := 2^{-h}$, $n := \lceil 4|V| \log\left(\frac{2}{\overline{\epsilon}}\right) \delta_{\min}^{-2|V|} \rceil$, $\alpha := \overline{\epsilon}/n$.
(2) Compute $[\![\text{s-}val]\!]_v^n$ for a given α and its complement $[\![\text{r-}val]\!]_v^n$.
(3) If $(B, p) \in [\![\text{r-}val]\!]_v^n$, return "$(B, p)$ is in $\langle \text{r-}val_v^* \rangle$."
(4) If $d_\infty\left((B, p), [\![\text{r-}val]\!]_v^n\right) \ge 2\overline{\epsilon}$, return "$(B, p)$ is in $\langle \text{s-}val_v^* \rangle$."

The algorithm AlgApprox iteratively looks for a precision $\overline{\epsilon}$ that is sufficient to decide whether (B, p) is in r-val_v^* or s-val_v^*. Since $[\![\text{r-}val]\!]_v^n \subseteq \text{r-}val_v^n \subseteq \text{r-}val_v^*$, every time the algorithm returns from Step (3), the decision is sound. For a given precision $\overline{\epsilon}$, the algorithm computes enough steps of the abstract value iteration on a fine-enough grid to ensure that: a) the abstract value $[\![\text{r-}val_v^n]\!]$ is at most $\overline{\epsilon}$-far from r-val_v^n, and b) the concrete value r-val_v^n is at most $\overline{\epsilon}$-far from r-val_v^*. Hence the decision on line (4) is also sound, by the triangular inequality. Finally, provided that (B, p) is in ϵ-distance from the threshold, the algorithm will eventually set $\overline{\epsilon} \le \frac{\epsilon}{2}$, decide by (3) or (4), and terminate.

Theorem 5.6. *The restriction of Prob. 1 to inputs satisfying Assump. 1 lies in EXPTIME. In particular, AlgApprox is a sound and complete algorithm when Assump. 1 is fulfilled by the given problem instance, and provided the distance between* (B, p) *and* v*'s threshold is* $\epsilon > 0$*, then AlgApprox terminates in at most* $h = \lceil \log(1/\epsilon) \rceil + 1$ *iterations yielding the time complexity*

$$O\left(\frac{|V|^4}{\epsilon} \log(1/\epsilon)^3\, \delta_{\min}^{-4|V|}\right).$$

5.4 Lower Complexity Bounds

We show that the exact problem for general MDPs is at least as hard as computing values in simple stochastic games (SSG), which is known to be in NP ∩ co-NP, and whether it belongs to P remains a long-standing open problem. SSGs generalize MDPs by partitioning control vertices into two sets based on ownership by two players, referred to as Player 0 and Player 1. Formally, an SSG is a tuple $\langle V, V_0, V_1, V_r, E, \delta \rangle$, where V_0 and V_1 are the vertices owned by Player 0 and Player 1, respectively. The components V, V_r, E, and δ are adapted from MDPs as follows: $V = V_0 \cup V_1 \cup V_r$, $E: V_0 \cup V_1 \to V_r$, and $\delta: V_r \to \Delta(V_0 \cup V_1)$. Without loss of generality, we assume every SSG has a designated initial vertex in V_1 and that every path belongs to $(V_1 V_r V_0 V_r)^\omega$, ensuring an even alternation of vertex types, which can be achieved with polynomial blowup.

A Player j *policy* maps each vertex $v \in V_j$ to one of its successors. Suppose Player 0 is the reachability player aiming to reach a given target $T_G \subseteq V$, while Player 1 is the safety player trying to avoid it. The *value* of G is the maximum probability p such that Player 0 can reach T_G with probability at least p against any policy of Player 1. For detailed formal definitions, we refer to Condon [25].

Theorem 5.7. *For any SSG* G *with initial vertex* v*, there exists a bidding game* G_B *whose size is polynomial in the size of* G*, and such that the value of the reachability objective in* G *is equal to the minimal* p *that satisfies* $(\frac{1}{3}, p) \in \text{s-}val_v^*$ *in* G_B.

We provide the idea behind the proof; details are in the extended version [13]. For an SSG G, we construct the bidding game G_B by

adding an extra edge (v, v') to each vertex $v \in V_0 \cup V_1$ where v' is a new sink. The set of target vertices in G_B contains the vertices in T_B, and moreover contains every newly introduced sink state that is connected to a state in V_1. In other words, from the vertices in V_1 and V_0 in G_B, the reachability and safety players, respectively, can immediately win by moving to the connected sink vertex. This setup forces the safety and reachability players to outbid their opponents when $v \in V_1$ and $v \in V_0$, respectively, as losing the bid would result in an immediate loss. To analyze this game, we examine the achievable safety and reachability probabilities for an initial budget of $1/3-\varepsilon_0$. If both players bid optimally, the budget oscillates between $1/3 - \varepsilon_t$ and $2/3 - \varepsilon_t$, with the deviation ε_t growing at each step. While the reachability player mimics his strategy from G, the safety player can eventually force a transition to a sink once ε_t is large enough. However, this can take arbitrarily many steps depending on ε_0, allowing the reachability player to win with a probability arbitrarily close to the value of G as $\varepsilon_0 \to 0$.

6 BIDDING GAMES ON RESTRICTED MDP-S

Now we consider the special cases of Prob. 1 for acyclic and tree-shaped MDPs, which are MDPs whose underlying transition graphs are acyclic (with loops on sinks) and rooted trees, respectively.

6.1 Acyclic MDPs

Contrary to general MDPs, the value iteration algorithm (from Sec. 5) converges in at most $|V|$ iterations for acyclic MDPs. This implies that AlgExact will always terminate in at most $|V|$ iterations.

Lemma 6.1. *For acyclic MDPs, for every vertex v, $\text{r-}val_v^{|V|} = \text{r-}val_v^*$ and $\text{s-}val_v^{|V|} = \text{s-}val_v^*$.*

Lem. 6.1 implies that Prob. 1 is in EXPTIME for acyclic MDPs.

Theorem 6.2. *For acyclic MDPs, AlgExact is a sound and complete algorithm for Prob. 1 and terminates in at most $|V|$ iterations, yielding the space and time complexity $O\left(|V|^{|V|}\right)$.*

6.2 Tree-Shaped MDPs

In case the MDP is tree-shaped, we can solve Prob. 1 in NP ∩ co-NP. The main idea of the proof is to find a certificate of the fact that (B, p) belongs to $\text{r-}val_v^{|V|}$. According to the inductive definition of $\text{r-}val_v^{|V|}$ by (3) and (4), the presence of (B, p) in $\text{r-}val_v^{|V|}$ can be witnessed by a point $(B_w, p_w) \in \text{r-}val_w^{|V|-1}$ for every $w \in Succ(v)$. A similar witness can be constructed to show that each of the points (B_w, p_w) belongs to $\text{r-}val_w^{|V|-1}$, and the process can be repeated recursively up to the base case $(B_u, p_u) \in \text{r-}val_u^0$. Therefore, whenever (B, p) belongs to $\text{r-}val_v^*$, there exists a certificate of this fact in the form of a finite set of points satisfying the relations in (3) and (4). If M is a tree, the certificate moreover contains a single point (B_u, p_u) for every vertex $u \in V$. The whole certificate then satisfies the following constraints:

$$B_v \le B,\ p_v \ge p$$

$$\forall u \in V_c\ \exists u^-, u^+ \in Succ(u):\ B_u = \frac{B_{u^-} + B_{u^+}}{2},\ p_u \le \min_{w \in Succ(u)} p_w, B_{u^+} \ge \max_{w \in Succ(u)} B_w$$

$$\forall u \in V_r:\ B_u \ge \max_{w \in Succ(u)} B_w,\ p_u = \sum_{w \in Succ(u)} \delta(u)(w) \cdot p_w$$

$$\forall t \in T:\ 0 \le B_t \le 1,\ 0 \le p_t \le 1$$

$$\forall z \in Z:\ B_z = 1 \text{ or } p_z = 0,$$

where Z is the set of leaves not in T. By fixing a choice of u^- and u^+ for every control vertex u, and a choice of which of the two equalities should hold for each $z \in Z$, we create a concrete linear program. The point (B, p) belongs to $\text{r-}val_v^*$ if and only if there is a choice that makes the linear program feasible. The same idea can be used to prove a point belongs to $\text{s-}val_v^*$. Combining this with Thm. 4.1, we obtain the following upper complexity bounds.

Theorem 6.3. *For tree-shaped MDPs, Prob. 1 is in NP ∩ co-NP.*

7 CONCLUSIONS AND FUTURE WORK

We studied bidding games on MDPs with quantitative reachability and safety specifications. We show that thresholds are binary relations over budgets and probabilities. This makes their computation significantly more challenging than traditional bidding games on graphs, for which thresholds are scalars (budgets). We developed a new value iteration algorithm for approximating the threshold up to arbitrary precision, and showed how it can be used to decide whether a given initial budget B suffices to win with probability at least p, assuming (B, p) is not on the threshold (Assump. 1). In acyclic and tree-shaped MDPs, Assump. 1 is not required and the decision procedure becomes significantly more efficient.

A number of questions remain open: Is Prob. 1 decidable without Assump. 1? What are the exact complexities? (There is a big gap between the upper and lower complexity bounds.) Furthermore, several interesting extensions can be considered, namely extensions to richer classes of specifications (like ω-regular and mean-payoff) and extensions to different forms of bidding mechanisms (like poorman and taxman, both with and without charging). Another interesting question is the equivalence with stochastic models (recall that bidding games are equivalent to random-turn games). This is still unclear, because even if the threshold budget were simulated by random turn assignments, this randomness would not "blend" with the existing randomness (in the random transitions) in the MDP, and we would obtain stochastic games with two sources of probabilities, which have not been studied to the best of our knowledge. Finally, the foundation of auction-based scheduling [14] on MDPs is now ready, and it will be interesting to investigate how policy synthesis for multi-objective MDPs can benefit from it.

ACKNOWLEDGMENTS

We thank Alon Krymgand for suggesting the proof technique used for proving Thm. 5.7, which simplified our original proof. G. Avni and S. Sadhukhan were supported by the ISF grant no. 1679/21, M. Kurečka and P. Novotný were supported by the Czech Science Foundation grant no. GA23-06963S, and K. Mallik was supported by the ERC project ERC-2020-AdG 101020093.

REFERENCES

[1] Axel Abels, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. 2019. Dynamic Weights in Multi-Objective Deep Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 11–20. http://proceedings.mlr.press/v97/abels19a.html

[2] M. Aghajohari, G. Avni, and T. A. Henzinger. 2019. Determinacy in Discrete-Bidding Infinite-Duration Games. In *Proc. 30th CONCUR (LIPIcs, Vol. 140)*. 20:1–20:17.

[3] R. Alur, T. A. Henzinger, and O. Kupferman. 2002. Alternating-time temporal logic. *J. ACM* 49, 5 (2002), 672–713.

[4] G. Amanatidis, G. Birmpas, A. Filos-Ratsikas, and A. A. Voudouris. 2022. Fair Division of Indivisible Goods: A Survey. In *Proc. 31st IJCAI*, Luc De Raedt (Ed.). ijcai.org, 5385–5393.

[5] Pranav Ashok, Krishnendu Chatterjee, Jan Křetínský, Maximilian Weininger, and Tobias Winkler. 2020. Approximating Values of Generalized-Reachability Stochastic Games. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science* (Saarbrücken, Germany) *(LICS '20)*. Association for Computing Machinery, New York, NY, USA, 102–115. https://doi.org/10.1145/3373718.3394761

[6] G. Avni, E. Kafshdar Goharshady, T. A. Henzinger, and K. Mallik. 2024. Bidding Games with Charging. In *Proc. 35th CONCUR (LIPIcs, Vol. 311)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 8:1–8:17.

[7] G. Avni, T. A. Henzinger, and V. Chonev. 2019. Infinite-Duration Bidding Games. *J. ACM* 66, 4 (2019), 31:1–31:29.

[8] G. Avni, T. A. Henzinger, and R. Ibsen-Jensen. 2018. Infinite-Duration Poorman-Bidding Games. In *Proc. 14th WINE (LNCS, Vol. 11316)*. Springer, 21–36.

[9] G. Avni, T. A. Henzinger, R. Ibsen-Jensen, and P. Novotný. 2019. Bidding Games on Markov Decision Processes. In *Proc. 13th RP*. 1–12.

[10] G. Avni, T. A. Henzinger, and Đ. Žikelić. 2019. Bidding Mechanisms in Graph Games. In *Proc. 44th MFCS (LIPIcs, Vol. 138)*. 11:1–11:13.

[11] G. Avni, I. Jecker, and Đ. Žikelić. 2021. Infinite-Duration All-Pay Bidding Games. In *Proc. 32nd SODA*. 617–636.

[12] G. Avni, I. Jecker, and D. Zikelic. 2023. Bidding Graph Games with Partially-Observable Budgets. In *Proc. 37th AAAI*.

[13] Guy Avni, Martin Kurečka, Kaushik Mallik, Petr Novotnỳ, and Suman Sadhukhan. 2024. Bidding Games on Markov Decision Processes with Quantitative Reachability Objectives. *arXiv preprint arXiv:2412.19609* (2024).

[14] G. Avni, K. Mallik, and S. Sadhukhan. 2024. Auction-Based Scheduling. In *Proc 30th TACAS (Lecture Notes in Computer Science, Vol. 14572)*. Springer, 153–172.

[15] G. Avni, T. Meggendorfer, S. Sadhukhan, J. Tkadlec, and Đ. Zikelic. 2023. Reachability Poorman Discrete-Bidding Games. In *Proc. 26th ECAI (Frontiers in Artificial Intelligence and Applications, Vol. 372)*. IOS Press, 141–148.

[16] G. Avni and S. Sadhukhan. 2022. Computing Threshold Budgets in Discrete-Bidding Games. In *Proc. 42nd FSTTCS (LIPIcs, Vol. 250)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 30:1–30:18.

[17] H. Aziz, A. Filos-Ratsikas, J. Chen, S. Mackenzie, and N. Mattei. 2016. Egalitarianism of Random Assignment Mechanisms: (Extended Abstract). In *Proc. 15th AAMAS*. ACM, 1267–1268.

[18] H. Aziz, B. Li, He. Moulin, and X. Wu. 2022. Algorithmic fair allocation of indivisible items: a survey and new questions. *SIGecom Exch.* 20, 1 (2022), 24–40.

[19] M. Babaioff, T. Ezra, and U. Feige. 2021. Fair-Share Allocations for Agents with Arbitrary Entitlements. In *Proc. 21st EC*. ACM, 127.

[20] M. Babaioff, T. Ezra, and U. Feige. 2022. On Best-of-Both-Worlds Fair-Share Allocations. In *Proc. 18th WINE (LNCS, Vol. 13778)*. Springer, 237–255.

[21] Leon Barrett and Srini Narayanan. 2008. Learning All Optimal Policies with Multiple Criteria. In *Proceedings of the 25th International Conference on Machine Learning* (Helsinki, Finland) *(ICML '08)*. Association for Computing Machinery, New York, NY, USA, 41–47. https://doi.org/10.1145/1390156.1390162

[22] K. Chatterjee and T. A. Henzinger. 2012. A survey of stochastic ω-regular games. *J. Comput. Syst. Sci.* 78, 2 (2012), 394–413.

[23] K. Chatterjee, R. Majumdar, and T. A. Henzinger. 2006. Markov Decision Processes with Multiple Objectives. In *Proc. 23rd STACS*. 325–336.

[24] Taolue Chen, Vojtech Forejt, Marta Z. Kwiatkowska, Aistis Simaitis, and Clemens Wiltsche. 2013. On Stochastic Games with Multiple Objectives. In *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 8087)*, Krishnendu Chatterjee and Jirí Sgall (Eds.). Springer, 266–277. https://doi.org/10.1007/978-3-642-40313-2_25

[25] A. Condon. 1990. On Algorithms for Simple Stochastic Games. In *Proc. DIMACS*. 51–72.

[26] A. Condon. 1992. The Complexity of Stochastic Games. *Inf. Comput.* 96, 2 (1992), 203–224.

[27] Luca De Alfaro and Thomas A Henzinger. 2000. Concurrent omega-regular games. In *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 99CB36332)*. IEEE, 141–154.

[28] M. Develin and S. Payne. 2010. Discrete Bidding Games. *The Electronic Journal of Combinatorics* 17, 1 (2010), R85.

[29] Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi, and Mihalis Yannakakis. 2008. Multi-Objective Model Checking of Markov Decision Processes. *Log. Methods Comput. Sci.* 4, 4 (2008). https://doi.org/10.2168/LMCS-4(4:8)2008

[30] J. Filar and K. Vrieze. 1997. *Competitive Markov decision processes*. Springer Verlag.

[31] D. Fisman, O. Kupferman, and Y. Lustig. 2010. Rational Synthesis. In *Proc. 16th TACAS*. 190–204.

[32] Conor F. Hayes, Roxana Radulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel de Oliveira Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A practical guide to multi-objective reinforcement learning and planning. *Auton. Agents Multi Agent Syst.* 36, 1 (2022), 26. https://doi.org/10.1007/S10458-022-09552-Y

[33] O. Kupferman, G. Perelli, and M. Y. Vardi. 2016. Synthesis with rational environments. *Ann. Math. Artif. Intell.* 78, 1 (2016), 3–20.

[34] A. J. Lazarus, D. E. Loeb, J. G. Propp, W. R. Stromquist, and D. H. Ullman. 1999. Combinatorial Games under Auction Play. *Games and Economic Behavior* 27, 2 (1999), 229–264.

[35] A. J. Lazarus, D. E. Loeb, J. G. Propp, and D. Ullman. 1996. Richman Games. *Games of No Chance* 29 (1996), 439–449.

[36] R. Meir, G. Kalai, and M. Tennenholtz. 2018. Bidding games and efficient allocations. *Games and Economic Behavior* 112 (2018), 166–193. https://doi.org/10.1016/j.geb.2018.08.005

[37] Kristof Van Moffaert and Ann Nowé. 2014. Multi-Objective Reinforcement Learning using Sets of Pareto Dominating Policies. *Journal of Machine Learning Research* 15, 107 (2014), 3663–3692. http://jmlr.org/papers/v15/vanmoffaert14a.html

[38] Michael Painter, Bruno Lacerda, and Nick Hawes. 2020. Convex Hull Monte-Carlo Tree-Search. In *Proceedings of the 30th International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI Press, 217–225.

[39] Y. Peres, O. Schramm, S. Sheffield, and D. Bruce Wilson. 2007. Random-Turn Hex and Other Selection Games. *The American Mathematical Monthly* 114, 5 (2007), 373–387.

[40] Martin L Puterman. 1990. Markov decision processes. *Handbooks in operations research and management science* 2 (1990), 331–434.

[41] R. S. Sutton and A. G. Barto. 1998. *Reinforcement learning - an introduction*. MIT Press.

[42] M. J. Wooldridge, J. Gutierrez, P. Harrenstein, E. Marchioni, G. Perelli, and A. Toumi. 2016. Rational Verification: From Model Checking to Equilibrium Checking. In *Proc. of the 30th AAAI*. 4184–4191.