Recursive Stochastic Games with Positive Rewards*

Kousha Etessami[a], Dominik Wojtczak[b], Mihalis Yannakakis[c]

[a] *School of Informatics, University of Edinburgh*
[b] *Department of Computer Science, University of Liverpool*
[c] *Computer Science Department, Columbia University*

Abstract

We study the complexity of a class of Markov decision processes and, more generally, stochastic games, called 1-exit Recursive Markov Decision Processes (1-RMDPs) and 1-exit Recursive Simple Stochastic Games (1-RSSGs), with *strictly positive* rewards. These are a class of finitely presented countable-state zero-sum turn-based stochastic games that subsume standard finite-state MDPs and Condon's simple stochastic games. They correspond to optimization and game versions of several classic stochastic models, with rewards. In particular, they correspond to the MDP and game versions of multi-type branching processes and stochastic context-free grammars with strictly positive rewards. The goal of the two players in the game is to maximize/minimize the total expected reward generated by a play of the game. Such stochastic models arise naturally as models of probabilistic procedural programs with recursion, and the problems we address are motivated by the goal of analyzing the optimal/pessimal expected running time in such a setting.

We first show that in such games both players have optimal deterministic "stackless and memoryless" optimal strategies. We then provide polynomial-time algorithms for computing the exact optimal expected reward (which may be infinite, but is otherwise rational), and optimal strategies, for both the maximizing and minimizing single-player versions of the game, i.e., for (1-exit) Recursive Markov Decision Processes (1-RMDPs). It follows that the quantitative decision problem for positive reward 1-RSSGs is in NP $\cap$ coNP. We show that Condon's well-known quantitative termination problem for finite-state simple stochastic games (SSGs) which she showed to be in NP $\cap$ coNP reduces to a special case of the reward problem for 1-RSSGs, namely, deciding whether the value is ∞. By contrast, for finite-state SSGs with strictly positive rewards, deciding if this expected reward value is ∞ is solvable in P-time. We also show that there is a simultaneous strategy improvement algorithm that converges in a finite number of steps to the value and optimal strategies of a 1-RSSG with positive rewards.

Keywords: Recursive Markov decision processes, recursive simple stochastic games, recursive Markov chains, total expected reward, stochastic context-free grammars, multi-type branching processes, probabilistic pushdown systems, linear programming, policy iteration and strategy improvement algorithms.

* A preliminary version of this paper appeared in the proceedings of the *ICALP'2008* conference: [15].

Email addresses: kousha@inf.ed.ac.uk (Kousha Etessami), d.wojtczak@liv.ac.uk (Dominik Wojtczak), mihalis@cs.columbia.edu (Mihalis Yannakakis)

1. Introduction

Markov decision processes and stochastic games are fundamental models in stochastic dynamic optimization and game theory (see, e.g., [33, 31, 21]). In this paper, motivated by the goal of analyzing the optimal/pessimal expected running time of probabilistic procedural programs, we study the complexity of a reward-based stochastic game, called *1-exit recursive simple stochastic games* (1-RSSGs), and its 1-player version, *1-exit recursive Markov decision processes* (1-RMDPs). These form a class of (finitely presented) countable-state turn-based zero-sum stochastic games (and MDPs) with strictly positive rewards, and with an undiscounted expected total reward objective.

Intuitively, a 1-RSSG (1-RMDP) consists of a collection of finite-state component SSGs (MDPs), each of which can be viewed as an abstract finite-state procedure (subroutine) of a probabilistic program with potential recursion. Each component procedure has some nodes that are probabilistic and others that are controlled by one or the other of the two players. The component SSGs can call each other in a recursive manner, generating a potentially unbounded call stack, and thereby an infinite state space. The "1-exit" restriction essentially restricts these finite-state subroutines so they do not return a value, unlike multi-exit RSSGs and RMDPs in which they can return distinct values. (We shall show that the multi-exit version of these reward games are undecidable.) An example 1-RSSG with two components A and B is depicted in Figure 1. 1-RMDPs and 1-RSSGs were studied in [17] in a setting without rewards, where the goal of the players was to maximize/minimize the probability of termination. Such termination probabilities can be irrational, and quantitative decision problems for them subsume long standing open problems in exact numerical computation. Here we extend 1-RSSGs and 1-RMDPs to a setting with positive rewards. Note that much of the literature on MDPs and games is based on a reward structure. This paper is a first step toward extending these models to the recursive setting. Interestingly, we show that the associated problems actually become more benign in some respects in this strictly positive reward setting. In particular, the values of our games are either rational, with polynomial bit complexity, or ∞.

The 1-RMDP and 1-RSSG models can also be described as optimization and game versions of several classic stochastic models, including stochastic context-free grammars (SCFGs) and (multi-type) branching processes. These have applications in many areas, including natural language processing [29], biological sequence analysis ([10]), and population biology [25, 24]. Another model that corresponds to a strict subclass of SCFGs is "random walks with back-buttons" studied in [19] as a model of web surfing. See [16] for details on the relationships between these various models.

A 1-RSSG with positive rewards, can be equivalently reformulated as the following game played on a stochastic context-free grammar. We are given a context-free grammar where nonterminals are partitioned into three disjoint sets: `random`, `player-1`, and `player-2`. Starting from a designated start nonterminal, S_{init}, we proceed to generate a (*left-most*) derivation by choosing a remaining (*left-most*) nonterminal, S, and expanding it. The precise derivation law (left-most, right-most, etc.) does not effect the game's value in our strictly positive reward setting, but it would do so if we were to allow 0 rewards on rules/transitions. If S belongs to `random`, it is expanded ran-

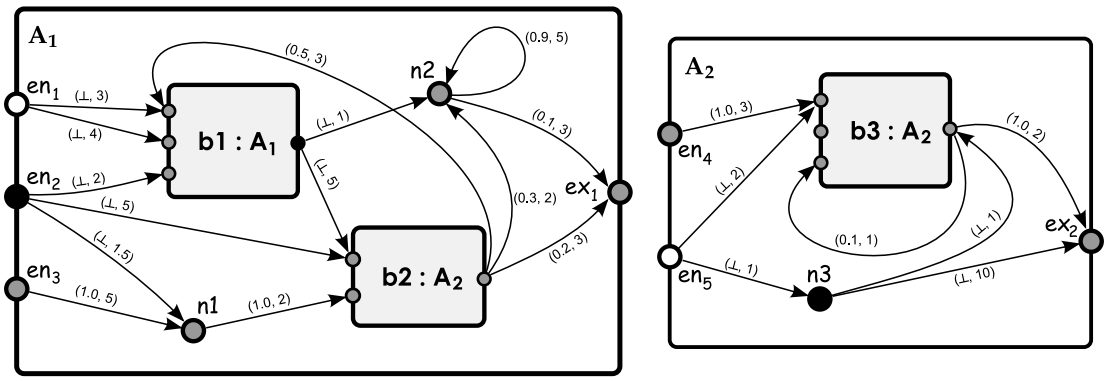

Figure 1: A 1-RSSG example consisting of two components, A_1 and A_2. Black vertices belong to player 1, white to player 2, grey vertices to "nature" (i.e., they are random). Each box (labelled, e.g., b1:A_1) has a name (b1) and is mapped to a component (A_1). Each edge has a label whose first component is $\perp$ for controlled vertices or a probability value for random ones, and the second component is the reward assigned to this edge.

domly by choosing a rule $S \to \alpha$, according to a given probability distribution over the rules whose left hand side is S. If S belongs to `player-i`, then player i chooses which grammar rule to use to expand this S. Each grammar rule also has an associated strictly positive *reward* for player 1, and each time a rule is used during the derivation, player 1 accumulates this associated reward. Player 1 wants to maximize the total expected reward (which may be ∞), and player 2 wants to minimize it. When we have only one player in the game it is either a minimizing or maximizing 1-RMDP.

Let us mention another very closely related model to 1-RMDPs and 1-RSSGs, namely *(multi-type) Branching Markov Decision Processes (BMDPs)* and *Branching Simple Stochastic Games* (BSSGs), which constitute a natural generalization of the purely stochastic *multi-type Branching Processes* ([25]), to the controlled setting of MDPs and SSGs. These stochastic processes are heavily used in population biology and many other areas of applied probability. They model the stochastic evolution of a population of objects of possibly distinct types. In each generation, each object of a given type in the population gives rise to a (possibly empty) set of offspring objects of possibly distinct types in the next generation. In the purely probabilistic setting, the offspring in the next generation for an object of type, t, are determined by a probability distribution associated with the type t. In the controlled and game settings of BMDPs and BSSGs, the players control particular types of objects and can decide between a (finite) set of offspring choices for each object of that type in each generation. Players can use whatever strategy they wish to optimize a given objective. See [17, 13, 14] for more on these models. We mention that the results of this paper yield directly a polynomial time algorithm, given a BMDP, for computing the optimal (maximum or minimum) *expected number of descendants of a given type* for an object of a given type. Speculating somewhat about the applications of such models, they could be useful, for example, if the branching process represents a population of different types of

cells, where some cell types are benign whereas other cell types are malignant. If an adversary (say a foreign virus) can control the reproductive behavior of some cell types, then such a model could be used to compute (in P-time), the worst-case expected total number of malignant cells that can arise, under the worst possible adversary. Alternatively, if medicines could be introduced to control the reproductive behavior of some cell types, then such a model could be used to compute (in P-time), the expected total number of malignant cells that would arise under the best possible medicine (and to compute that "medicine", i.e., an optimal strategy).

We assume *strictly positive* rewards on all transitions (rules) in this paper, and this assumption is essential for our results regarding 1-RMDPs and 1-RSSGs. However, for analyzing, e.g., the optimal expected total number of descendants of given types for BMDPs and BSSGs, we only need the assumption that rewards are non-negative, and all of our results would remain intact (this is essentially because BMDPs operate under a *simultaneous* derivation law, unlike, e.g., context-free grammars with *left-most* derivation). The assumption of strictly positive rewards is very natural for modeling the optimal/pessimal expected running time in probabilistic procedural programs: each discrete step of the program is assumed to cost some non-zero amount of time. Strictly positive rewards also endow our games with a number of important robustness properties. In particular, in the above context-free grammar presentation, with strictly positive rewards these games have the same value regardless of what derivation law is imposed. This is not the case if we also allow 0 rewards on grammar rules. In that case, even in the single-player setting, the game value can be wildly different (e.g., 0 or ∞) depending on the derivation law (e.g., left-most, or right-most, or simultaneous). We shall explain all this in more detail in Section 6.1, using explicit examples of such games presented as context-free grammars.

As we shall show, none of these pathologies arise in the setting with strictly positive rewards. In this case, all derivation rules for the context-free grammar presentation of these games yield precisely the same value. The *left-most* derivation rule is the one that captures precisely 1-RMDPs and 1-RSSGs. We show that 1-RMDPs and 1-RSSGs with strictly positive rewards have a value which is either rational (with polynomial bit complexity) or ∞, and which arises as the least fixed point solution (over the extended reals) of an associated system of linear-min-max equations. Both players do have optimal strategies in these games, and in fact we show the much stronger fact that both players have *stackless and memoryless* (SM) optimal strategies: deterministic strategies that depend only on the current state of the running component, and not on the history or even the stack of pending recursive calls.

We provide polynomial-time algorithms for computing the exact value for both maximizing and minimizing 1-RMDPs with positive rewards, and for computing optimal strategies. The two cases of maximization and minimization are not equivalent and require separate treatment. We show that for the 2-player games (1-RSSGs) deciding whether the game has value at least a given $r \in \mathbb{Q} \cup \{\infty\}$ is in NP $\cap$ coNP. We also describe a practical simultaneous strategy improvement algorithm, analogous to similar algorithms for finite-state stochastic games, and show that it converges to the game value (even if it is ∞) in a finite number of steps. A corollary is that computing the game value and optimal strategies for these games is contained in the class PLS of polynomial local search problems ([27]).

We also observe that these games are "harder" than Condon's finite-state SSG games [7] in the following senses. We reduce Condon's quantitative decision problem for finite-state SSGs to a special case of 1-RSSG games with strictly positive rewards: namely to deciding whether the game value is ∞. By contrast, if finite-state SSGs are themselves equipped with strictly positive rewards, we can decide in P-time whether their value is ∞. Moreover, it has been shown that computing the value of Condon's SSG games is in the complexity class PPAD for which the 2-player Nash Equilibrium problem [6] (and the ≥ 3 player ϵ-NE problem [9]) is complete (see [18] and [28]). The same proof however does not work for 1-RSSGs with positive rewards, and we do not know whether these games are contained in PPAD. Technically, the problem is that in the expected reward setting the domain of the fixed point equations is not compact, and indeed the expected reward is potentially ∞, so the problem cannot in an obvious way be formulated as a Brouwer fixed point problem. In these senses, the 1-RSSG reward games studied in this paper appear to be "harder" than Condon's SSGs, and yet as we show their quantitative decision problems remain in NP $\cap$ coNP. Finally, we show that the more general multi-exit RSSG model is undecidable. Namely, even for single-player multi-exit RMDPs with strictly positive rewards, it is undecidable whether the optimal reward value is ∞.

Applications of these models to the analysis of expected running time of recursive probabilistic programs, as in the tool PReMo ([39]), was the original motivation for the work in this paper. The tool PReMo [39] implements a number of analyses for RMCs, 1-RMDPs, and 1-RSSGs. In particular, the strategy improvement algorithm of this paper was implemented and incorporated in that tool. It is worth noting that it was shown by Friedmann [22] that essentially the same simultaneous strategy improvement algorithm requires exponentially many steps in the worst case to compute the optimal value for parity games and for Condon's finite-state SSGs, and similar results were shown by Fearnley [20] for MDPs with a total or average reward criterion. Despite this worst-case behavior, the algorithm performs very well in practice on a wide range of instances, including for 1-RSSGs. See [37] for some encouraging experimental results showing how simultaneous strategy improvement outperforms some other standard iterative methods for computing the fixed point of max-linear equation systems.

Related work. Two (equivalent) purely probabilistic recursive models, Recursive Markov chains and probabilistic Pushdown Systems (pPDSs) were introduced in [16] and [11], and have been studied in several papers subsequently. These models were extended to the optimization and game setting of (1)-RMDPs and (1)-RSSGs in [17], and studied further in [2, 3, 13, 14]. As mentioned earlier, the games considered in these earlier papers had the goal of maximizing/minimizing termination or reachability probability, which can be irrational. Furthermore, the problems of computing an optimal strategy, as well as quantitative decision problems regarding the optimal probability (e.g., decide whether it exceeds a given rational bound), encounter long standing open problems in numerical computation, even to place their complexity in NP. On the other hand, the qualitative termination decision problem ("is the termination game value exactly 1?") for 1-RMDPs was shown to be in P, and for 1-RSSGs in NP $\cap$ coNP, in [17]. These results are related to the results in the present paper as follows. If termination occurs with probability strictly less than 1 in a strictly positive reward game, then

the expected total reward is ∞. But the converse does not hold: the expected reward may be ∞ even when the game terminates with probability 1, because there can be *null recurrence* in these infinite-state games. Thus, not only do we have to address this discrepancy, but also our goal in this paper is quantitative computation (to compute the optimal reward), whereas in [17] it was purely qualitative (almost sure termination).

The problem of approximating the optimal termination probabilities for 1-RMDPs (and BMDPs) was addressed in [13], which gave efficient algorithms for computing approximately the optimal termination probabilities of 1-RMDPs within any desired accuracy in polynomial time in the size of the given 1-RMDP and the number of bits of precision, and for computing ϵ-optimal strategies. Efficient algorithms for approximately optimizing the reachability probabilities of BMDPs in polynomial time were presented in [14] (these algorithms do not apply however to reachability analysis of 1-RMDPs). Our focus in this paper however is on exact optimization, and the objective is based on rewards rather than termination/reachability probability.

Condon [7] originally studied finite-state SSGs with termination objectives (no rewards), and showed that the quantitative termination decision problem, i.e. determining whether the value of the game (the optimal termination probability) is greater than or equal to a given rational number, is in NP $\cap$ coNP; it is a well-known open problem whether this problem is in P. In [8] strategy improvement algorithms for SSGs were studied, based on variants of the classic Hoffman-Karp algorithm [26]. As noted earlier, more recently it was shown by Friedmann [22] that the simultaneous strategy improvement method requires exponentially many steps in the worst case to compute the optimal value for both parity games and for Condon's finite-state SSGs.

There has been some work on augmenting purely probabilistic multi-exit RMCs and pPDSs with rewards in [12], as well as work on analyzing the distribution of the runtime of RMCs and pPDSs, proving effective tail bounds for it (using polynomial space) [4]. These results however are for purely probabilistic RMCs without players. We in fact show in Theorem 20 that the basic questions for analyzing multi-exit RMDPs and RSSGs with positive rewards are undecidable.

An independent paper by Gawlitza and Seidl [23] considers monotone linear-min-max equations with potentially negative constant terms (with entirely different motivation coming from abstract interpretation), and studies a different kind of strategy improvement algorithm for computing their least fixed point solution over the *full* extended reals. Their work is related to ours, but in subtle ways. In particular their notion of LFP over the extended reals may yield negative values or even $-\infty$, and they assume that "strategies" (choices for the max and min operators) are memoryless, rather than proving a (memoryless) determinacy result. Moreover, their strategy improvement algorithm requires a particular initial strategy (otherwise, it can fail) and thus is not directly formulable as a local search. Unlike our results, their results apparently do not yield containment in NP $\cap$ coNP for the relevant decision problems (only containment in NP is known, see [23]). Nevertheless, there appear to be close connections between their work and ours that could be explored further.

Models related to 1-RMDPs have been studied in Operations Research under the name Branching Markov Decision Chains (a controlled version of multi-type Branching processes). These are close to Branching Markov Decision Processes with non-negative rewards and to the single-player SCFG model, but with simultaneous deriva-

tion law. They were studied by Pliska [32], in a related form by Veinott [36], and extensively by Rothblum and co-authors (e.g., [34]). Besides the restriction to simultaneous derivation, these models were restricted to the single-player MDP case, and to simplify their analysis they were typically assumed to be "transient" (i.e., the expected number of visits to a node was assumed to be finite under all strategies). None of these works yield a P-time algorithm for optimal expected rewards for 1-RMDPs with positive rewards. Although we do not directly appeal to any of these results (and in particular to the eigenvalue characterisations they typically involve), our results are related and further generalize a related model to a 2-player setting.

Another work [5], studies the problem of finding a strategy that minimizes the expected number of transitions taken before termination for a given *one-counter Markov Decision Process* (OC-MDP). The OC-MDP model can be seen as a special subclass of RMDPs, but it is not comparable with 1-RMDPs: there are (countable state) Markov decision processes generated by 1-RMDPs that cannot be generated by OC-MDPs and *vice versa*. It was shown in [5] that an ϵ-optimal strategy for the objective of minimizing the expected number of transitions taken can be computed in time linear in $1/\epsilon$ and exponential in the encoding size of the OC-MDP, and that finding such a strategy cannot be done in polynomial time unless P=NP.

Finally, the model studied in this paper has been extended to a model with time constraints [35] and concurrent game setting in [38]. In the former, each transition has an associated time constraint and can only be taken if this constraint is satisfied. The results in this paper generalize well to such a setting with an exponential blow-up in the computational complexity of the problems studied. In the latter, at each step both players make a (possibly probabilistic) choice among the ones available to him at the current node. The next state and reward generated by this step is dependent on the selected pair. Such games do not belong to the class of perfect-information games, because the choices are made currently and independently of each other. In this concurrent setting, optimal and even ϵ-optimal strategy may have to be probabilistic. Furthermore, optimal strategies may require infinite amount of memory in general, but it was shown in [38] that both players have ϵ-optimal probabilistic stackless & memoryless strategies and can be found using a natural generalization of the strategy improvement algorithm presented in this paper. The quantitative decision questions regarding the value of such a game as well as checking whether the value is infinite can be answered in PSPACE and turns out to be as hard as the square root sum problem.

Organization of the paper. The rest of the paper is organized as follows. Section 2 gives basic definitions of the models and the problems studied, and presents relevant background. It shows also how to construct from a given 1-RSSG with positive rewards a linear min-max system of equations whose least fixed point gives the optimal rewards for every starting vertex. Section 3 shows that both players have stackless-memoryless optimal strategies, and also proves that a strategy improvement algorithm can be used to compute optimal strategies. Section 4 presents polynomial-time algorithms for both minimizing and maximizing 1-RMDPs with positive rewards. It also shows that the problem for 1-RSSGs is in NP $\cap$ coNP, and furthermore the qualitative problem of determining if the optimal reward in infinite is at least as hard as Condon's quantitative problem for finite-state SSGs. In Section 5 we show that the problem for multi-

exit RMDPs is undecidable. Section 6 explains the close relationship of 1-RMDPs and 1-RSSGs with positive rewards with the analogous reward models of Branching MDPs and games (BMDPs and BSSGs) and Stochastic context-free grammar MDPs and games.

2. Definitions and Background

Let $\mathbb{R}_{>0} = (0, \infty)$ denote the positive real numbers, $\mathbb{R}_{\geq 0} = [0, \infty)$, $\overline{\mathbb{R}} = [-\infty, \infty]$, $\mathbb{R}^{\infty}_{>0} = (0, \infty]$, and $\mathbb{R}^{\infty}_{\geq 0} = [0, \infty]$. The extended reals $\overline{\mathbb{R}}$ have the natural total order. We assume the following usual arithmetic conventions on the non-negative extended reals $\mathbb{R}^{\infty}_{\geq 0}$: $a \cdot \infty = \infty$, for any $a \in \mathbb{R}^{\infty}_{>0}$; $0 \cdot \infty = 0$; $a + \infty = \infty$, for any $a \in \mathbb{R}^{\infty}_{\geq 0}$. This extends naturally to matrix arithmetic over $\mathbb{R}^{\infty}_{\geq 0}$.

We first define general multi-exit RSSGs (for which basic reward problems turn out to be undecidable). Later, we will confine these to the 1-exit case, 1-RSSGs. A visual depiction of a RSSG is given in Figure 1 and its detailed description follows after the formal definition.

A *Recursive Simple Stochastic Game (RSSG) with positive rewards* is a tuple $A = (A_1, \ldots, A_k)$, where each *component* $A_i = (N_i, B_i, Y_i, En_i, Ex_i, \mathtt{pl}_i, \delta_i, \xi_i)$ consists of:

- A set N_i of *nodes*, with a distinguished subset En_i of *entry* nodes and a (disjoint) subset Ex_i of *exit* nodes.
- A set B_i of *boxes*, and a mapping $Y_i : B_i \mapsto \{1, \ldots, k\}$ that assigns to every box (the index of) a component. To each box $b \in B_i$, we associate a set of *call ports*, $Call_b = \{(b, en) \mid en \in En_{Y(b)}\}$, and a set of *return ports*, $Ret_b = \{(b, ex) \mid ex \in Ex_{Y(b)}\}$. Let $Call^i = \cup_{b \in B_i} Call_b$, $Ret^i = \cup_{b \in B_i} Ret_b$, and let $Q_i = N_i \cup Call^i \cup Ret^i$ be the set of all nodes, call ports and return ports; we refer to these as the *vertices* of component A_i.
- A mapping $\mathtt{pl}_i : Q_i \mapsto \{0, 1, 2\}$ that assigns to every vertex a *player* (Player 0 represents "chance" or "nature"). We assume $\mathtt{pl}_i(u) = 0$ for all $u \in Call^i \cup Ex_i$.
- A transition relation $\delta_i \subseteq (Q_i \times (\mathbb{R}_{>0} \cup \{\bot\}) \times Q_i \times \mathbb{R}_{>0})$, where for each tuple $(u, x, v, c_{u,v}) \in \delta_i$, the source $u \in (N_i \setminus Ex_i) \cup Ret^i$, the destination $v \in (N_i \setminus En_i) \cup Call^i$, and x is either (i) $p_{u,v} \in (0, 1]$ (the transition probability) if $\mathtt{pl}_i(u) = 0$, or (ii) $x = \bot$ if $\mathtt{pl}_i(u) = 1$ or 2; and $c_{u,v} \in \mathbb{R}_{>0}$ is the positive reward associated with this transition. A transition $(u, x, v, c_{u,v}) \in \delta_i$ can be viewed as an edge from vertex u to vertex v with label $(x, c_{u,v})$, and the transition relation δ_i as a set of labelled edges on the vertices of A_i (see Figure 1). We assume that for any two vertices, u and v, there is at most one transition in δ_i from u to v. For computational purposes we assume the given probabilities $p_{u,v}$ and rewards $c_{u,v}$ are rational. Probabilities must also satisfy consistency: for every $u \in \mathtt{pl}_i^{-1}(0)$, $\sum_{\{v' \mid (u, p_{u,v'}, v', c_{u,v'}) \in \delta_i\}} p_{u,v'} = 1$, unless u is a call port or exit node, neither of which have outgoing transitions, in which case by default $\sum_{v'} p_{u,v'} = 0$.
- Finally, the mapping $\xi_i : Call_i \mapsto \mathbb{R}_{>0}$ maps each call port u in the component to a positive rational value $c_u = \xi(u)$. (This mapping reflects the "cost" of a

function call, but is not strictly necessary. This cost can be 0 and all our results would still hold.)

Example. The example RSSG in Figure 1 has two components A_1 and A_2, i.e., $k = 2$. Component A_1 has three entry nodes (en_1, en_2 and en_3), two internal nodes ($n1$ and $n2$) and one exit node ex_1. Component A_2 has two entry nodes (en_4 and en_5), one internal nodes ($n3$) and one exit node ex_2.

Component A_1 has two boxes $b1$ and $b2$ mapped to A_1 and A_2, respectively. As box $b1$ is mapped to A_1, i.e., $Y(b1) = 1$, it has three entry ports (($b1, en_1$), ($b1, en_2$), ($b1, en_3$)) and one exit port ($b1, ex_1$). Box $b2$ is mapped to A_2, i.e., $Y(b2) = 2$, and has two entry ports (($b2, en_4$), ($b2, en_5$)) and one exit port (($b2, ex_2$)). Component A_2 has only one box $b3$ mapped to A_1, i.e., $Y(b3) = 1$, which has three entry ports (($b3, en_1$), ($b3, en_2$), ($b3, en_3$)) and one exit port (($b3, ex_1$)).

We can see that nodes en_2, $(b1, ex_1)$ and n_3 belong to player 1, i.e., $\mathtt{pl}(en) = \mathtt{pl}((b1, ex_1)) = \mathtt{pl}(n_3) = 1$. Nodes en_1 and en_5 belong to player 2, i.e., $\mathtt{pl}(en_1) = \mathtt{pl}(en_5) = 2$. For all other nodes x we have $\mathtt{pl}_i(x) = 0$, i.e., they are random nodes.

An example of a probabilistic transition is $(n2, 0.1, ex_1, 3) \in \delta_1$, which gives reward 3 to Player 1 with probability 0.1. In the figure, it is represented by an arrow from $n2$ to ex_1 with $(0.1, 3)$ as its label. An example of a controlled transition is $(en_5, \bot, n3, 1) \in \delta_2$, which gives reward 1 to Player 1 if Player 2, while at en_5, decides to use it. In the figure, it is represented by an arrow from en_5 to $n3$ with $(\bot, 1)$ as its label. In this particular example $\xi \equiv 0$. □

We use the symbols N, B, Q, δ, etc., without a subscript, to denote the union over all components. Thus, e.g., $N = \cup_{i=1}^{k} N_i$ is the set of all nodes of A, $\delta = \cup_{i=1}^{k} \delta_i$ the set of all transitions, etc. Let $n(u) = \{v \mid (u, \bot, v, c_{u,v}) \in \delta\}$ denote the neighbors of u if u is a player 1 or player 2 vertex and $n(u) = \{v \mid (u, p_{u,v}, v, c_{u,v}) \in \delta\}$ otherwise. An RSSG A defines a global denumerable simple stochastic game, with rewards, $M_A = (V = V_0 \cup V_1 \cup V_2, \Delta, \mathtt{pl})$ as follows. The global *states* $V \subseteq B^* \times Q$ of M_A are pairs of the form $\langle \beta, u \rangle$, where $\beta \in B^*$ is a (possibly empty) sequence of boxes and $u \in Q$ is a *vertex* of A. The states $V \subseteq B^* \times Q$ and transitions Δ are defined inductively as follows:

1. $\langle \epsilon, u \rangle \in V$, for $u \in Q$. (ϵ denotes the empty string.)
2. if $\langle \beta, u \rangle \in V$ & $(u, x, v, c) \in \delta$, then $\langle \beta, v \rangle \in V$ and $(\langle \beta, u \rangle, x, \langle \beta, v \rangle, c) \in \Delta$.
3. if $\langle \beta, (b, en) \rangle \in V$ & $(b, en) \in Call_b$, then $\langle \beta b, en \rangle \in V$ & $(\langle \beta, (b, en) \rangle, 1, \langle \beta b, en \rangle, \xi((b, en))) \in \Delta$.
4. if $\langle \beta b, ex \rangle \in V$ & $(b, ex) \in Ret_b$, then $\langle \beta, (b, ex) \rangle \in V$ & $(\langle \beta b, ex \rangle, 1, \langle \beta, (b, ex) \rangle, 0) \in \Delta$.

The mapping $\mathtt{pl} : V \mapsto \{0, 1, 2\}$ is given as follows: $\mathtt{pl}(\langle \beta, u \rangle) = \mathtt{pl}(u)$ if u is in $Q \setminus (Call \cup Ex)$, and $\mathtt{pl}(\langle \beta, u \rangle) = 0$ if $u \in Call \cup Ex$. The set of states V is partitioned into V_0, V_1, and V_2, where $V_i = \mathtt{pl}^{-1}(i)$. We consider M_A with various *initial states* of the form $\langle \epsilon, u \rangle$, denoting this by M_A^u. Some states of M_A are *terminating states* and have no outgoing transitions. These are states $\langle \epsilon, ex \rangle$, where ex is an exit node. An RSSG where $V_2 = \emptyset$ ($V_1 = \emptyset$) is called a maximizing (minimizing, respectively) *Recursive Markov Decision Process* (RMDP); an RSSG where $V_1 \cup V_2 = \emptyset$ is called a *Recursive Markov Chain* (RMC) ([16]). A *1-RSSGs* is a RSSG where every component has one exit, and we likewise define *1-RMDPs* and *1-RMCs*. (The example RSSG in Figure

1 is in fact a 1-RSSG, because each component has just one exit.) This entire paper is focused on 1-RSSGs and 1-RMDPs, except for Theorem 20, where we show that multi-exit RMDP reward games are undecidable.

In a (1-)RSSG with positive rewards the goal of player 1 (maximizer) is to maximize the total expected reward gained during a play of the game, and the goal of player 2 (minimizer) is to minimize this. A *strategy* σ for player i, $i \in \{1,2\}$, is a function $\sigma : V^*V_i \mapsto V$, where, given the history $ws \in V^*V_i$ of play so far, with $s \in V_i$ (i.e., it is player i's turn to play a move), $\sigma(ws) = s'$ determines the next move of player i, where $(s, \bot, s', c) \in \Delta$. (We could also allow randomized strategies, but this won't be necessary, as we shall see.)

A special class of strategies extensively used later in this paper are *Stackless & Memoryless (SM)* strategies. These are strategies that are deterministic, and depend neither on the history of the game nor on the current call stack. In other words, these strategies only depend on the current vertex. Such strategies, for player i, can clearly be specified by a function $\sigma : V_i \mapsto V$.

Let Ψ_i denote the set of all strategies for player i. A pair of strategies $\sigma \in \Psi_1$ and $\tau \in \Psi_2$ induces in a straightforward way a Markov chain $M_A^{\sigma,\tau} = (V^*, \Delta')$, whose set of states is the set V^* of histories. Let $r_u^{k,\sigma,\tau}$ denote the expected reward in k steps in $M_A^{\sigma,\tau}$, starting at initial state $\langle \epsilon, u \rangle$. Formally, we can define the reward gained during the i'th transition, starting at $\langle \epsilon, u \rangle$ to be given by a random variable C_i. The total k-step expected reward is simply $r_u^{k,\sigma,\tau} = E[\sum_{i=1}^{k} C_i]$. When $k = 0$, we of course have $r_u^{0,\sigma,\tau} = 0$. Given an initial vertex u, let $r_u^{*,\sigma,\tau} = \lim_{k\to\infty} r^{k,\sigma,\tau} = E[\sum_{i=1}^{\infty} C_i] \in [0,\infty]$ denote the total expected reward obtained in a run of $M_A^{\sigma,\tau}$, starting at initial state $\langle \epsilon, u \rangle$. Clearly, this sum may diverge, thus $r^{*,\sigma,\tau} \in [0,\infty]$. Note that, because of the positive constraint on the rewards out of all transitions, the sum will be finite if and only if the expected number of steps until the run terminates is finite.

We now want to associate a "value" to 1-RSSG games. Unlike 1-RSSGs with termination probability objectives, it unfortunately does not follow directly from general determinacy results such as Martin's Blackwell determinacy ([30]) that these games are determined, because those determinacy results require a Borel payoff function to be bounded, whereas the payoff function for us is unbounded. Instead, let us define for all vertices u $r_u^* \doteq \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} r_u^{*,\sigma,\tau}$. Also, for a strategy $\sigma \in \Psi_1$, let $r_u^{*,\sigma} \doteq \inf_{\tau\in\Psi_2} r_u^{*,\sigma,\tau}$, and for $\tau \in \Psi_2$, let $r_u^{*,\cdot,\tau} \doteq \sup_{\sigma\in\Psi_1} r_u^{*,\sigma,\tau}$. Player 1's strategy σ is called ϵ-*optimal* if $r_u^{*,\sigma} \geq r_u^* - \epsilon$ and is *optimal* if it is 0-optimal; similarly for player 2's strategies. A game is *determined* if for every $\epsilon > 0$ both players have ϵ-optimal strategies and we call it *SM-determined* if both players have optimal SM strategies. We will first show that $r_u^* = \inf_{\tau\in\Psi_2} \sup_{\sigma\in\Psi_1} r_u^{*,\sigma,\tau}$, so our games are determined and r_u^* is the *value* of the game starting at vertex u, and later that our games are in fact also SM determined.

We are interested in the following problem: *Given A, a 1-RSSG (or 1-RMDP), and given a vertex u in A, compute r_u^* if it is finite, or else declare that $r_u^* = \infty$. Also, compute optimal SM strategies for both players.*

In [17] we defined a monotone system of *nonlinear* min-max equations for the value of the termination probability game on 1-RSSGs, and showed that its *Least Fixed Point* solution yields the desired probabilities. Here we show we can adapt this to obtain

analogous *linear* min-max systems in the setting of positive reward 1-RSSGs. We use a variable x_u for each unknown r^*_u. Let $\mathbf{x}$ be the vector of all $x_u, u \in Q$. The system has one equation of the form $x_u = P_u(\mathbf{x})$ for each vertex u. Suppose that u is in component A_i with (unique) exit ex. There are 5 cases based on the "*Type*" of u.

1. $Type_0$: $u = ex$. In this case: $x_u = 0$.
2. $Type_{rand}$: $\mathtt{pl}(u) = 0$ & $u \in (N_i \setminus \{ex\}) \cup Ret^i$: $x_u = \sum_{v \in n(u)} p_{u,v}(x_v + c_{u,v})$.
3. $Type_{call}$: $u = (b, en)$ is a call port: $x_{(b,en)} = x_{en} + x_{(b,ex')} + c_u$, where $ex' \in Ex_{Y(b)}$ is the unique exit of $A_{Y(b)}$.
4. $Type_{max}$: $\mathtt{pl}(u) = 1$ and $u \in (N_i \setminus \{ex\}) \cup Ret^i$: $x_u = \max_{v \in n(u)}(x_v + c_{u,v})$
5. $Type_{min}$: $\mathtt{pl}(u) = 2$ and $u \in (N_i \setminus \{ex\}) \cup Ret^i$: $x_u = \min_{v \in n(u)}(x_v + c_{u,v})$

We denote the system in vector form by $\mathbf{x} = P(\mathbf{x})$. Given a 1-RSSG, we can easily construct its associated system in linear time. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \leq \mathbf{y}$ means $x_j \leq y_j$ for every j. Let $\mathbf{r}^* \in \mathbb{R}^n$ denote the n-vector of r^*_u's. Let $\mathbf{0}$ denote an all 0 vector, and define $\mathbf{x}^0 = \mathbf{0}$, $\mathbf{x}^{k+1} = P^{k+1}(\mathbf{0}) = P(\mathbf{x}^k)$, for $k \geq 0$.

Theorem 1. ***1.*** *The map $P : \overline{\mathbb{R}}^n \to \overline{\mathbb{R}}^n$ is monotone on $\mathbb{R}^\infty_{\geq 0}$ and $\mathbf{0} \leq \mathbf{x}^k \leq \mathbf{x}^{k+1}$ for $k \geq 0$.*

2. *$\mathbf{r}^* = P(\mathbf{r}^*)$.*

3. *For all $k \geq 0$, $\mathbf{x}^k \leq \mathbf{r}^*$.*

4. *For all $\mathbf{r}' \in \mathbb{R}^\infty_{\geq 0}$, if $\mathbf{r}' = P(\mathbf{r}')$, then $\mathbf{r}^* \leq \mathbf{r}'$.*

5. *For all vertices u,*

$$r^*_u \doteq \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} r^{*,\sigma,\tau}_u = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} r^{*,\sigma,\tau}_u.$$

(In other words, these games are determined.)

6. *$\mathbf{r}^* = \lim_{k \to \infty} \mathbf{x}^k$.*

Proof.

1. All equations in the system $P(x)$ are min-max linear with non-negative coefficients and constants, and hence are monotone.
2. The proof that $\mathbf{r}^* = P(\mathbf{r}^*)$ is similar to the one for 1-RSSG termination games from [17], but it uses in a crucial way the fact that rewards on all transitions are strictly positive.
   (a) For $u = ex \in Type_0$, $\mathbf{r}^*_u = 0$, so it fulfills the corresponding equation $x_u = 0$.

(b) For $u \in Type_{rand}$, from the definition $r^*_u = \sup_\sigma \inf_\tau r^{*,\sigma,\tau}_u$ it follows that $\mathbf{r}^*_u = \sum_{v \in n(u)} p_{u,v}(\mathbf{r}^*_v + c_{u,v})$. Note that this holds even when some of the expected rewards are infinite, because if $p_{u,v} > 0$ and the game starting at v has infinite reward value, then this is also the case starting at u.

(c) For $u \in Type_{call}$, where $u = (b, en)$ is a call port. We claim that

$$\mathbf{r}^*_u = \mathbf{r}^*_{en} + \mathbf{r}^*_{(b,ex')} + c_u \tag{1}$$

where ex' is the unique exit of $Y(b)$. First, for any pair of strategies σ and τ of player 1 and 2, respectively, we define two random variables. Let $K^{\sigma,\tau}$ be equal to the total accumulated reward until a play of the game M_A starting at u, and using strategies σ and τ, exits the box b, i.e. reaches (b, ex') in the same (empty) calling context (i.e., with the same (empty) call stack), and let $L^{\sigma,\tau}$ be the total accumulated reward thereafter (if the play never leaves b then $L^{\sigma,\tau}$ is defined to be 0). Also, let T be the event that the game exits the box b and T' be its complement. Finally, let $\mathrm{P}^{\sigma,\tau}(F)$ be the probability of the event F occurring in the Markov chain $M^{\sigma,\tau}_u$. From the definition, $\mathbf{r}^*_u = \sup_\sigma \inf_\tau \mathrm{E}(K^{\sigma,\tau} + L^{\sigma,\tau}) = \sup_\sigma \inf_\tau \mathrm{E}K^{\sigma,\tau} + \mathrm{E}L^{\sigma,\tau} = \sup_\sigma \inf_\tau \mathrm{E}(K^{\sigma,\tau}|T) \cdot \mathrm{P}^{\sigma,\tau}(T) + \mathrm{E}(K^{\sigma,\tau}|T') \cdot \mathrm{P}^{\sigma,\tau}(T') + \mathrm{E}(L^{\sigma,\tau}|T) \cdot \mathrm{P}^{\sigma,\tau}(T) + \mathrm{E}(L^{\sigma,\tau}|T') \cdot \mathrm{P}^{\sigma,\tau}(T') = \sup_\sigma \inf_\tau \mathrm{E}(K^{\sigma,\tau}|T) \cdot \mathrm{P}^{\sigma,\tau}(T) + \infty \cdot \mathrm{P}^{\sigma,\tau}(T') + \mathrm{E}(L^{\sigma,\tau}|T) \cdot \mathrm{P}^{\sigma,\tau}(T) + 0 \cdot \mathrm{P}^{\sigma,\tau}(T')$, because the event T' implies that the game never stops and from the assumption that all rewards are strictly positive $\mathrm{E}(K^{\sigma,\tau}|T')$ has to be ∞ then. We now claim that the last expression is in fact equal to $\sup_\sigma \inf_\tau \mathrm{E}(K^{\sigma,\tau}) + \mathrm{E}(L^{\sigma,\tau}|T)$. This is because equality holds if $\mathrm{P}^{\sigma,\tau}(T) = 1$ and otherwise we have $\mathrm{P}^{\sigma,\tau}(T') > 0$ which implies that both expressions are ∞ and so are equal. Now, $\sup_\sigma \inf_\tau \mathrm{E}(K^{\sigma,\tau}) + \mathrm{E}(L^{\sigma,\tau}|T) = \sup_\sigma \inf_\tau \mathrm{E}(K^{\sigma,\tau}) + \sup_\sigma \inf_\tau \mathrm{E}(L^{\sigma,\tau}|T)$, because any pair of player 1's strategies σ_1 and σ_2 that are (ϵ-)optimal for $\inf_\tau \mathrm{E}(K^{\sigma_1,\tau})$ and $\inf_\tau \mathrm{E}(L^{\sigma_2,\tau}|T)$, respectively, can be easily composed into a single strategy σ that is (ϵ-)optimal for $\inf_\tau \mathrm{E}(K^{\sigma,\tau}) + \mathrm{E}(L^{\sigma,\tau}|T)$ and *vice versa*. Finally, $\sup_\sigma \inf_\tau \mathrm{E}(K^{\sigma,\tau}) = c_u + \mathbf{r}^*_{en}$, because $K^{\sigma,\tau}$ only accumulates reward from the moment the game enters and until it leaves box b, and the structure of the game between these two moments is isomorphic to a game starting at en. Similarly, $\sup_\sigma \inf_\tau \mathrm{E}(L^{\sigma,\tau}|T) = \mathbf{r}^*_{(b,ex')}$, because the event T implies that the game reaches (b, ex') at some point, $L^{\sigma,\tau}$ accumulates reward only from that moment on and the structure of the game from that point is isomorphic to a game starting at (b, ex').

(d) For $u \in Type_{\max}$, we know that $\mathbf{r}^*_u \geq \mathbf{r}^*_v + c_{u,v}$ for any $v \in n(u)$, because otherwise the *max* player would be able to increase his expected reward by taking the transition to the node v in the first step. On the other hand, we also have that $\mathbf{r}^*_u \leq \mathbf{r}^*_v + c_{u,v}$ for some $v \in n(u)$, as otherwise no matter what transition player *max* picks from u, the *min* player has a strategy such that *max* would not be able to obtain the expected total reward $\mathbf{r}^*_u$.

(e) For $u \in Type_{\min}$, we know that $\mathbf{r}^*_u \leq \mathbf{r}^*_v + c_{u,v}$ for all $v \in n(u)$, because otherwise it would be better for the *min* player to take the transition leading

to the node v and giving the *max* player expected reward $\mathbf{r}^*_v + c_{u,v}$ which is lower than $\mathbf{r}^*_u$. However, for some $v \in n(u)$ it has to be $\mathbf{r}^* \geq \mathbf{r}^*_v + c_{u,v}$, as otherwise player *max* could always obtain expected reward higher than $\mathbf{r}^*_u$ no matter what *min* player does.

3. Note that P is monotonic, and $\mathbf{r}^*$ is a fixed point of P. Since $x^0 = \mathbf{0} \leq \mathbf{r}^*$, it follows by induction on k that $\mathbf{x}^k \leq \mathbf{r}^*$, for all $k \geq 0$.

4. Consider any fixed point $\mathbf{r}'$ of the equation system $P(\mathbf{x})$. We will prove that $\mathbf{r}^* \leq \mathbf{r}'$. Let us denote by τ^* a strategy for the *minimizer* that picks for each vertex the successor with the minimum value in $\mathbf{r}'$, i.e., for each state $s = \langle \beta, u \rangle$, where u belongs to player 2 (*minimizer*) nodes, we choose $\tau^*(s) = \arg\min_{v \in n(u)} \mathbf{r}'_v + c_{u,v}$ (breaking ties lexicographically).

   Lemma 2. *For all strategies $\sigma \in \Psi_1$ of player 1, and for all $k \geq 0$, $\mathbf{r}^{k,\sigma,\tau^*} \leq \mathbf{r}'$.*

   Proof. Base case $\mathbf{r}^{0,\sigma,\tau^*} = \mathbf{0} \leq \mathbf{r}'$ is trivial.

   (a) $u = ex$, then $\mathbf{r}^{k,\sigma,\tau^*}_u = 0 = \mathbf{r}'_u$ for all $k \geq 0$.

   (b) $u \in Type_{rand}$ is a random node and after we define a strategy $\sigma'(\theta) = \sigma(\langle \varepsilon, u \rangle \theta)$ we get:

   $$\mathbf{r}^{k+1,\sigma,\tau^*}_u = \sum_{v \in n(u)} p_{u,v}(\mathbf{r}^{k,\sigma',\tau^*}_v + c_{u,v}) \leq \sum_{v \in n(u)} p_{u,v}(\mathbf{r}'_v + c_{u,v}) = \mathbf{r}'_u$$

   based on the inductive assumption and the fact that $\mathbf{r}'$ is a fixed point of $P(\mathbf{x})$.

   (c) If $u = (b, en)$ is an entry en of the box b then we claim

   $$\mathbf{r}^{k+1,\sigma,\tau^*}_u \leq \max_\rho \mathbf{r}^{k,\rho,\tau^*}_{en} + \max_\rho \mathbf{r}^{k,\rho,\tau^*}_{(b,ex')} + c_u \tag{2}$$

   where (b, ex') is the only return port of box b. To see this, note that in any specific trajectory, the total reward gained in $k+1$ steps starting at call port (b, en) is c_u plus the remaining reward, which is split into two parts: that gained in i steps inside box b, and the rest gained in j steps after returning from box b, and such that $i + j = k$. Thus clearly the total expected reward in $k + 1$ steps starting at u is no more than c_u plus the expected reward in k steps starting inside box b (i.e., starting at the entry en of $Y(b)$) plus the expected gain in k steps starting at (b, ex'). We now have

   $$\max_\rho \mathbf{r}^{k,\rho,\tau^*}_{en} + \max_\rho \mathbf{r}^{k,\rho,\tau^*}_{(b,ex')} + c_u \leq \mathbf{r}'_{en} + \mathbf{r}'_{(b,ex')} + c_u = \mathbf{r}'_u \tag{3}$$

   by inductive assumption, and by the fact that $\mathbf{r}'$ is a fixed point of $P(\mathbf{x})$. So, combining equations (2) and (3), we have $\mathbf{r}^{k+1,\sigma,\tau^*}_u \leq \mathbf{r}'_u$.

(d) For $u \in Type_{max}$ we claim

$$\mathbf{r}_u^{k+1,\sigma,\tau^*} \leq \max_{v \in n(u)} \mathbf{r}_v^{k,\sigma',\tau^*} + c_{u,v}$$

because the player has to move to some neighbor v of $\langle \varepsilon, u \rangle$ in one step, and thus it cannot gain more that $\mathbf{r}^{k,\sigma',\tau^*}$, where σ' is defined from σ in the same way as for $Type_{rand}$. Thus

$$\mathbf{r}_u^{k+1,\sigma,\tau^*} \leq \max_{v \in n(u)} \mathbf{r}_v^{k,\sigma',\tau^*} + c_{u,v} \leq \max_{v \in n(u)} \mathbf{r}'_v + c_{u,v} = \mathbf{r}'_u$$

(e) For $u \in Type_{min}$ we know that $\tau^*(u) = \arg\min_{v \in n(u)}(\mathbf{r}'_u + c_{u,v}) = v^*$, so:

$$\mathbf{r}_u^{k+1,\sigma,\tau^*} = \mathbf{r}_{v^*}^{k,\sigma',\tau^*} + c_{u,v^*} \leq \mathbf{r}'_{v^*} + c_{u,v^*} = \min_{v \in n(u)} (\mathbf{r}'_v + c_{u,v}) = \mathbf{r}'_u$$

□

Now by the lemma we have $\mathbf{r}_u^{*,\sigma,\tau^*} = \lim_{k\to\infty} \mathbf{r}_u^{k,\sigma,\tau^*} \leq \mathbf{r}'_u$ for every vertex u and for any *max* player strategy σ, so $\sup_\sigma \mathbf{r}_u^{*,\sigma,\tau^*} \leq \mathbf{r}'_u$. Thus for all vertices u:

$$\mathbf{r}_u^* = \sup_\sigma \inf_\tau \mathbf{r}_u^{*,\sigma,\tau} \leq \inf_\tau \sup_\sigma \mathbf{r}_u^{*,\sigma,\tau} \leq \sup_\sigma \mathbf{r}_u^{*,\sigma,\tau^*} \leq \mathbf{r}'_u \tag{4}$$

5. In equation (4) above, choose $\mathbf{r}' = \mathbf{r}^*$. Then we have, for all vertices u,

$$\sup_\sigma \inf_\tau \mathbf{r}_u^{*,\sigma,\tau} = \inf_\tau \sup_\sigma \mathbf{r}_u^{*,\sigma,\tau}.$$

6. We know that $\mathbf{z} = \lim_{k\to\infty} \mathbf{x}^k$ exists in $[0, \infty]$, because it is a monotonically non-decreasing sequence (note some entries may be infinite). In fact we have $\mathbf{z} = \lim_{k\to\infty} P^{k+1}(\mathbf{0}) = P(\lim_{k\to\infty} P^k(\mathbf{0}))$, and thus $\mathbf{z}$ is a fixed point of the equation $P(\mathbf{x}) = \mathbf{x}$. So from (4) we have $\mathbf{r}^* \leq \lim_{k\to\infty} \mathbf{x}^k$. Since $\mathbf{x}^k \leq \mathbf{r}^*$ for all $k \geq 0$, $\lim_{k\to\infty} \mathbf{x}^k \leq \mathbf{r}^*$ and thus $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{r}^*$.

□

The following is a simple corollary of the proof.

Corollary 3. *In 1-RSSG positive reward games, the* minimizer *has an optimal deterministic Stackless and Memoryless (SM) strategy.*

Proof. It is enough to consider the strategy τ^*, from Part 4 of Theorem 1, when we let $\mathbf{r}' = \mathbf{r}^*$. For then, by equation (4), we have $r_u^* = \sup_\sigma r^{*,\sigma,\tau^*} = \inf_\tau \sup_\sigma r^{*,\sigma,\tau}$. □

Note that for a 1-RMC (i.e., without players) with positive rewards, the vector $\mathbf{r}^* \in (\mathbb{R}^\infty_{\geq 0})^n$ of expected total rewards is the LFP of a system $x = Ax + b$, for some non-negative matrix $A \in \mathbb{R}^{n\times n}$, $A \geq 0$, and a positive vector $b > 0$. We will exploit this fact later in various proofs.

3. SM-determinacy and strategy improvement

We now prove SM-determinacy for 1-RSSGs with positive rewards, and we also show that strategy improvement can be used to compute the values and optimal strategies for 1-RSSG positive reward games. Consider the following *(simultaneous) strategy improvement* algorithm.

Initialization: Pick some (any) SM strategy, σ, for player 1 (maximizer).
Iteration step: First compute the optimal value, $r_u^{*,\sigma}$, starting from every vertex, u, in the resulting minimizing 1-RMDP. (We show in Theorem 8 that this can be done in P-time.) Then, update σ to a new SM strategy, σ', as follows. For each vertex $u \in Type_{max}$, if $\sigma(u) = v$ and u has a neighbor $w \neq v$, such that $r_w^{*,\sigma} + c_{u,w} > r_v^{*,\sigma} + c_{u,v}$, let $\sigma'(u) := w$ (e.g., choose a w that maximizes $r_w^{*,\sigma} + c_{u,w}$). Otherwise, let $\sigma'(u) := \sigma(u)$.
Repeat the iteration step, using the new σ' in place of σ, until no further local improvement is possible, i.e., stop when $\sigma' = \sigma$.

Theorem 6 below shows that this algorithm always halts, and produces an optimal final SM strategy for player 1. (The proof shows it works even if we switch any non-empty subset of improvable vertices in each iteration.) Combined with Corollary 3, both players have optimal SM strategies, i.e., the games are SM-determined.

Recall that for a 1-RMC (i.e., without players) with positive rewards, the vector $\mathbf{r}^*$ of expected total rewards is the LFP of a system $x = Ax + b$, for some non-negative matrix $A \in \mathbb{R}^{n\times n}$, $A \geq 0$, and a positive vector $b > 0$. In the proof of Theorem 6 we shall need the following basic fact about matrix inequalities.[1]

Lemma 4. *For any $x \in \mathbb{R}^n_{\geq 0}$, $A \in (\mathbb{R}^\infty_{\geq 0})^{n\times n}$ and $b \in (\mathbb{R}^\infty_{>0})^n$, if $x \leq Ax + b$ then $x \leq (\sum_{k=0}^\infty A^k)b$. This holds even if for some indices i we have $b_i = 0$, as long as the entries in any such row i of the matrix A are all zero.*

Proof.

Let $D = \sum_{k=0}^\infty A^k$ and $y = Db$. We have to prove that $x \leq y$. Some of the entries of D can be infinite. Let $R = \{r_1, r_2, \ldots, r_m\}$ be the set of indices of the rows of D that contain at least one ∞ entry. For every $r \in R$, $y_r = \sum_{i=1}^n D_{r,i}b_i$. Since $b_i > 0$ for all i and $D_{r,i}$ is ∞ for at least one i, we have $y_r = \infty$ and so $x_r \leq y_r$ is trivially fulfilled for every $r \in R$. Now let us construct a new matrix A' by zeroing all rows of A that are in R. Similarly, let x' be the vector x where entries x_r for all $r \in R$ were zeroed. Let $D' = \sum_{k=0}^\infty A'^k$.

We will prove that $x' \leq A'x' + b$. For entries $r \in R$, it is trivial as $(A'x')_r + b_r = 0 + b_r \geq 0 = x'_r$. If $r \notin R$ then $x'_r = x_r$ and

$$(A'x')_r = \sum_{i=1}^n A'_{r,i}x'_i = \sum_{\{i|A'_{r,i}>0\}} A'_{r,i}x'_i$$

[1] Note that if we assume both that $A \in (\mathbb{R}_{\geq 0})^{n\times n}$ and that $(\sum_{k=0}^\infty A^k)$ converges, the lemma is trivial: we have $(I - A)^{-1} = (\sum_{k=0}^\infty A^k)$, and thus $x \leq Ax + b \Rightarrow x - Ax \leq b \Rightarrow (I - A)x \leq b \Rightarrow x \leq (I - A)^{-1}b$. But we need this lemma even when $(\sum_{k=0}^\infty A^k)$ is not convergent.

Proposition 5. *If $A_{i,j} > 0$, and for some k we have $D_{j,k} = \infty$ then $D_{i,k} = \infty$.*

Proof. We have that $D = I + AD$ and so $D_{i,k} = \delta_{ik} + \sum_{l=1}^{n} A_{i,l} D_{l,k} \geq A_{i,j} D_{j,k} = \infty$. (where δ_{ik} is equal to 1 if $i = k$ and 0 otherwise). □

Suppose that $r \notin R$. If for some i, $x'_i \neq x_i$, then $i \in R$ and we must have $D_{i,j} = \infty$ for some j. If $A'_{r,i} > 0$ then $A_{r,i} = A'_{r,i}$, and from Proposition 5 we get that $D_{r,j} = \infty$, which contradicts the fact that $r \notin R$. Thus for $r \notin R$, and for i such that $A'_{r,i} > 0$, we must have $x'_i = x_i$ and $A'_{r,i} = A_{r,i}$. Thus $(A'x')_r + b_r = (Ax)_r + b_r \geq x_r = x'_r$ for all $r \notin R$. Hence we can conclude that $x'_r \leq (A'x')_r + b_r$ for all r.

We will now prove that $\lim_{k\to\infty} A'^k = 0$. For contradiction, note that if we had $\lim_{k\to\infty}(A'^k)_{i,j} \neq 0$ for some i, j then it must be the case that $D'_{i,j} = \infty$, because $(A'^k)_{i,j} \geq 0$ for all k, and for some $\epsilon > 0$ and infinitely many k, $(A')^k_{i,j} > \epsilon$. Since $A' \leq A$, we get that $A'^k \leq A^k$ for any $k \geq 0$ and thus $\sum_{k=0}^{\infty} A'^k \leq \sum_{k=0}^{\infty} A^k$. Therefore if $D'_{i,j} = \infty$ then $D_{i,j} = \infty$, but this means that all entries in the i-th row of A were zeroed in order to obtain A'. However if the i-th row in A' has only zeroes, then so does the i-th row in A'^k for any k. This would contradict the assumption that $\lim_{k\to\infty}(A'^k)_{i,j} \neq 0$.

Now, substituting x' by $A'x' + b$ in $x' \leq A'x' + b$, we get that $x' \leq A'x' + b \leq A'(A'x' + b) + b = A'^2x' + A'b + b \leq A'^2(A'x' + b) + A'b + b = A'^3x' + (A'^2 + A' + I)b$ and by iterating we can see that $x' \leq A'^{l+1}x' + (\sum_{k=0}^{l}(A')^k)b$ for any $l \geq 0$. All entries of x' are finite and $\lim_{k\to\infty} A'^k = 0$, so by taking the limit $l \to \infty$ we get $x' \leq (\sum_{k=0}^{\infty}(A')^k)b \leq (\sum_{k=0}^{\infty}(A)^k)b = y$. This shows that also for $r \notin R$ we have $x_r \leq y_r$, which concludes the proof that $x \leq y$.

Finally, we show now that we can also handle the case when for some indices i, $b_i = 0$ as long as each such a i-th row in A contains only 0s. We proceed by induction on the number, d, of indices i such that $b_i = 0$. For the base case $d = 0$, the claim was already proved. For the inductive case, suppose $d > 0$, and let i be the smallest such index. Since we assume $Ax + b \geq x$, it must be that $x_i = 0$. For any matrix M, let M' denote the matrix obtained by removing the i-th row and the i-th column from M. Similarly, for a vector v by v' denote the vector v with the i-th entry removed. If $x_i = 0$, then $M'x' = (Mx)'$ for any matrix M. Also, since the i-th row of A contains only 0s we have that $(A')^k = (A^k)'$ for any $k \geq 0$ and so $\sum_{k=0}^{\infty}(A')^k = (\sum_{k=0}^{\infty} A^k)'$. Now, from $Ax + b \geq x$ we get $(Ax + b)' \geq x'$ and so $A'x' + b' \geq x'$. But it is easy to see that A' and b' have the same property as before: if $b'_j = 0$ then the j-th row of A' consists of only 0s. Moreover, there are now $d - 1$ such indices. Thus, from the inductive hypothesis, $x' \leq (\sum_{k=0}^{\infty}(A')^k)b' = (\sum_{k=0}^{\infty} A^k)'b' = ((\sum_{k=0}^{\infty} A^k)b)'$, and because the inequality is trivial for the i-th position of x, we conclude that $x \leq (\sum_{k=0}^{\infty} A^k)b$. □

Theorem 6. *(1)* SM-determinacy. *In 1-RSSG positive reward games, both players have optimal SM strategies (and thus by Corollary 3 these games are SM determined).*
(2) Strategy Improvement. *Moreover, we can compute the value and optimal SM strategies using the above simultaneous strategy improvement algorithm.*
(3) Computing the value and optimal strategies in these games is contained in the class PLS.

Proof. Let σ be any SM strategy for player 1. Consider $\mathbf{r}_u^{*,\sigma} = \inf_{\tau\in\Psi_2} \mathbf{r}_u^{*,\sigma,\tau}$. (Note that some entries in the vector $\mathbf{r}^{*,\sigma}$ may be ∞.) First, note that if $\mathbf{r}^{*,\sigma} = P(\mathbf{r}^{*,\sigma})$ then

$\mathbf{r}^{*,\sigma} = \mathbf{r}^*$. This is because, by Theorem 1, $\mathbf{r}^* \leq \mathbf{r}^{*,\sigma}$, and on the other hand, σ is just one strategy for player 1, and for every vertex u, $\mathbf{r}^*_u = \sup_{\sigma' \in \Psi_1} \mathbf{r}^{*,\sigma'}_u \geq \mathbf{r}^{*,\sigma}_u$. Now we claim that, for all vertices u such that $u \notin Type_{\max}$, $\mathbf{r}^{*,\sigma}_u$ satisfies its equation in $\mathbf{x} = P(\mathbf{x})$. In other words, $\mathbf{r}^{*,\sigma}_u = P_u(\mathbf{r}^{*,\sigma})$. To see this, note that for vertices u of Types $\{0, call, rand\}$, no choice of either player is involved and the equation holds by definition of $\mathbf{r}^{*,\sigma}$ (In particular, the expected reward value at a call u is c_u plus the sum of the expected reward values of the game starting at the entry inside the box, and the game starting at the return port.) For nodes $u \in Type_{\min}$, we have the equation $\mathbf{x}_u = \min_{v \in n(u)} \mathbf{x}_v + c_{u,v}$. But note that the best minimizer can do against strategy σ, starting at $\langle \epsilon, u \rangle$, is to move to a neighboring vertex v such that $v = \arg\min_{v \in n(u)}(\mathbf{r}^{*,\sigma}_v + c_{u,v})$. Thus, the only equations that may fail are those for $u \in Type_{max}$, $\mathbf{x}_u = \max_{v \in n(u)}(\mathbf{x}_v + c_{u,v})$. Suppose $\sigma(u) = v$, for some neighbor v. Clearly then, $\mathbf{r}^{*,\sigma}_u = \mathbf{r}^{*,\sigma}_v + c_{u,v}$. Thus, $\mathbf{r}^{*,\sigma}_u \leq \max_{v' \in n(u)}(\mathbf{r}^{*,\sigma}_{v'} + c_{u,v'})$. Thus equality fails iff there is another vertex $w \neq v$, with $(u, \perp, w) \in \delta$, such that $\mathbf{r}^{*,\sigma}_v + c_{u,v} < \mathbf{r}^{*,\sigma}_w + c_{u,w}$.

Suppose now that the nodes $(u_1, u_2, \ldots u_n)$ are all those nodes where the *SM* strategy σ is not locally optimal, i.e., for $i = 1, 2, \ldots, n$, $\sigma(u_i) = v_i$, and thus $\mathbf{r}^{*,\sigma}_{u_i} = \mathbf{r}^{*,\sigma}_{v_i} + c_{u_i,v_i}$, but there is some w_i such that $\mathbf{r}^{*,\sigma}_{v_i} + c_{u_i,v_i} < \mathbf{r}^{*,\sigma}_{w_i} + c_{u_i,w_i}$. Let $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and similarly define $\mathbf{v}$ and $\mathbf{w}$. Consider now a revised *SM* strategy σ', which is identical to σ, except that $\sigma'(u_i) = w_i$ for all i. Next, consider a parametrized 1-exit RSSG, $A(\mathbf{t})$ where $\mathbf{t} = (t_1, t_2, \ldots, t_n)$, which is identical to A, except that all edges out of vertices u_i are removed, and replaced by a single probability 1 edge labeled by reward t_i, to the exit of the same component node u_i is in. Fixing the value of the vector $\mathbf{t} \in [0, \infty]^n$ determines an 1-RSSG, $A(\mathbf{t})$. Note that if we restrict *SM* strategies σ or σ' to vertices other than those in $\mathbf{u}$, then they both define the same *SM* strategy for the 1-RSSG $A(\mathbf{t})$. Define $r^{*,\sigma,\tau,\mathbf{t}}_z$ to be the expected total reward starting from $\langle \epsilon, z \rangle$ in the Markov chain $M^{z,\sigma,\tau}_{A(\mathbf{t})}$. Now, for each vertex z, define the function $f_z(\mathbf{t}) = \inf_{\tau \in \Psi_2} r^{*,\sigma,\tau,\mathbf{t}}_z$. In other words, $f_z(\mathbf{t})$ is the infimum of the expected rewards, over all strategies of player 2, starting at $\langle \epsilon, z \rangle$ in $A(\mathbf{t})$. This reward is parametrized by $\mathbf{t}$. Now, let $\mathbf{t}^\sigma$ be a vector such that $\mathbf{t}^\sigma_{u_i} = \mathbf{r}^{*,\sigma}_{u_i}$, and observe that $f_z(\mathbf{t}^\sigma) = \mathbf{r}^{*,\sigma}_z$ for every z. This is so because any strategy for minimizing the total reward starting from z would, upon hitting a state $\langle \beta, u_i \rangle$ in some arbitrary context β, be best off minimizing the total expected reward starting from $\langle \beta, u_i \rangle$ until that context is exited, (and unless the minimizer has a strategy that assures the context is exited with probability 1, the expected reward will be ∞).

Note that, by Corollary 3, in the 1-RSSG reward game on $A(\mathbf{t})$, for any values in vector $\mathbf{t}$, and any start vertex z, minimizer has an optimal SM strategy $\tau_{z,\mathbf{t}}$, such that $\tau_{z,\mathbf{t}} = \arg\min_{\tau \in \Psi_2} r^{*,\sigma,\tau,\mathbf{t}}_z$. Let $g_{(z,\tau)}(\mathbf{t}) = r^{*,\sigma,\tau,\mathbf{t}}_z$. Note that $f_z(\mathbf{t}) = \min_\tau g_{z,\tau}(\mathbf{t})$, where the minimum is over *SM* strategies. Now, note that the function $g_{z,\tau}(\mathbf{t})$ is the expected reward in a positive reward 1-RMC starting from a particular vertex, and it is given by $g_{z,\tau}(\mathbf{t}) = (\lim_{k \to \infty} R^k(\mathbf{0}))_z$ for a linear system $\mathbf{x} = R(\mathbf{x})$ with non-negative coefficients in R, where $R(\mathbf{x}) = A_{\sigma,\tau}\mathbf{x} + \mathbf{b}^{\sigma,\tau}(\mathbf{t})$, for some nonnegative matrix $A_{\sigma,\tau}$, and vector $\mathbf{b}^{\sigma,\tau}(\mathbf{t})$ which describes the average 1-step rewards from each vertex. All of these 1-step rewards are positive, except that at positions u_i the entry is the variable t_i, i.e., $\mathbf{b}^{\sigma,\tau}_{u_i}(\mathbf{t}) = t_i$. (Note that for all i the u_i'th row vector of $A_{\sigma,\tau}$ is all zero.) Simple iteration then shows that $g_{z,\tau}(\mathbf{t}) = \lim_{k \to \infty} R^k(\mathbf{0})_z = ((\sum_{k=0}^\infty A^k_{\sigma,\tau})\mathbf{b}^{\sigma,\tau}(\mathbf{t}))_z$. (Note that if $\lim_{k \to \infty} A^k_{\sigma,\tau} = 0$, then $(\sum_{k=0}^\infty A^k_{\sigma,\tau}) = (I - A_{\sigma,\tau})^{-1}$.) Now $g_{z,\tau}(\mathbf{t})$ has the following properties: it is a continuous, nondecreasing, and linear function of $\mathbf{t} \in [0, \infty]^n$, and

for $\mathbf{t} \in [0,\infty]^n$, $g_{z,\tau}(\mathbf{t}) \in [0,\infty]$. Specifically, we can think of it as a function $g_{z,\tau}(\mathbf{t}) = \boldsymbol{\alpha}^{z,\tau}\mathbf{t} + \beta^{z,\tau}$, where $\boldsymbol{\alpha}^{z,\tau} = (\alpha_1^{z,\tau}, \alpha_2^{z,\tau}, \ldots, \alpha_n^{z,\tau})$ and $\alpha_i^{z,\tau}, \beta^{z,\tau} \in [0,\infty]$.

Let $\mathbf{g}^\tau(\mathbf{t}) = (g_{w_1,\tau}(\mathbf{t}'), g_{w_2,\tau}(\mathbf{t}'), \ldots, g_{w_n,\tau}(\mathbf{t}'))$ where $\mathbf{t}' = \mathbf{t} + \mathbf{c}^{\mathbf{u},\mathbf{w}}$ and let $\mathbf{c}^{\mathbf{u},\mathbf{w}} = (c_{u_1,w_1}, c_{u_2,w_2}, \ldots, c_{u_n,w_n})$. Note $\mathbf{t} \in (-c_{u_1,w_1}, \infty] \times (-c_{u_2,w_2}, \infty] \times \ldots \times (-c_{u_n,w_n}, \infty]$. We can represent $\mathbf{g}^\tau(\mathbf{t})$ as $D^\tau\mathbf{t} + \mathbf{d}^\tau$, where $D^\tau = [\boldsymbol{\alpha}^{w_1,\tau}; \boldsymbol{\alpha}^{w_2,\tau}; \ldots; \boldsymbol{\alpha}^{w_n,\tau}]$ and $\mathbf{d}_j^\tau = \sum_{i=0}^n \alpha_i^{w_j,\tau} c_{u_i,w_i} + \beta^{w_j,\tau}$. Note that if $\mathbf{d}_j^\tau = 0$ then it has to be $\boldsymbol{\alpha}^{w_j,\tau} = \mathbf{0}$ and $\beta^{w_j,\tau} = 0$, because $c_{u_i,w_i} > 0$ for all i.

Consider function $\mathbf{f}(\mathbf{t}) = \min_\tau \mathbf{g}^\tau(\mathbf{t})$. This is well defined, since whatever the values in $\mathbf{t}$, the *min* player always has, by Corollary 3, an optimal *SM* strategy τ^* in $A(\mathbf{t})$ such that for any strategy σ of the *max* player, and any strategy τ of the *min* player, and all z we have $r_z^{*,\sigma,\tau^*,\mathbf{t}} \leq r_z^{*,\sigma,\tau,\mathbf{t}}$. Note that $\mathbf{f}(\mathbf{t}) = (f_{w_1}(\mathbf{t} + \mathbf{c}^{\mathbf{u},\mathbf{w}}), f_{w_2}(\mathbf{t} + \mathbf{c}^{\mathbf{u},\mathbf{w}}), \ldots, f_{w_n}(\mathbf{t} + \mathbf{c}^{\mathbf{u},\mathbf{w}}))$.

Lemma 7. *If $\mathbf{f}(\mathbf{t}) > \mathbf{t}$ for some finite vector $\mathbf{t}$, then for any fixed point $\mathbf{t}^*$ of $\mathbf{f}$, $\mathbf{t} \leq \mathbf{t}^*$.*

Proof. Suppose that $\mathbf{t}^*$ is some fixed point of $\mathbf{f}$. Since $\mathbf{f}(\mathbf{t}^*) = \min_\tau \mathbf{g}^\tau(\mathbf{t}^*)$, for some τ^* we have $\mathbf{g}^{\tau^*}(\mathbf{t}^*) = \mathbf{t}^*$. From the fact that $\mathbf{f}(\mathbf{t}) > \mathbf{t}$, we get that for all τ we have $\mathbf{g}^\tau(\mathbf{t}) > \mathbf{t}$. In particular we have $\mathbf{g}^{\tau^*}(\mathbf{t}) > \mathbf{t}$, which means that $D^{\tau^*}\mathbf{t} + \mathbf{d}^{\tau^*} > \mathbf{t}$. Now, for all i, either $\mathbf{d}_i^{\tau^*} = 0$ and the i-th row in D^{τ^*} is all zeroes, or $\mathbf{d}_i^{\tau^*} > 0$, thus from Lemma 4 we can conclude that $\mathbf{t} \leq \sum_{k=0}^\infty (D^{\tau^*})^k \mathbf{d}^{\tau^*}$. However, letting $h(\mathbf{t}) = \mathbf{g}^{\tau^*}(\mathbf{t}) = D^{\tau^*}\mathbf{t} + \mathbf{d}^{\tau^*}$ be the linear operator on $[0,\infty]^n$, note that the least fixed point solution (in $[0,\infty]^n$) of $h(\mathbf{t})$ is $\mathbf{t}_0 = \lim_{k\to\infty} h^{k+1}(\mathbf{0}) = \lim_{k\to\infty} D^{\tau^*} h^k(\mathbf{0}) + \mathbf{d}^{\tau^*} = \sum_{k=0}^\infty (D^{\tau^*})^k \mathbf{d}^{\tau^*}$. Thus, any other fixed point of h has to be greater than $\mathbf{t}_0$ and in particular $\mathbf{t}^* \geq \mathbf{t}_0 \geq \mathbf{t}$. □

Now, we know that $\mathbf{f}(\mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}})_i = f_{w_i}(\mathbf{t}^\sigma) = \mathbf{r}_{w_i}^{*,\sigma} > \mathbf{r}_{v_i}^{*,\sigma} + c_{u_i,v_i} - c_{u_i,w_i} = \mathbf{r}_{u_i}^{*,\sigma} - c_{u_i,w_i} = (\mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}})_i$ which proves that $\mathbf{f}(\mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}}) > \mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}}$. Therefore, by Lemma 7, any fixed point of $\mathbf{f}$ has to be greater or equal to $\mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}}$. Also, if we switch strategy σ to σ', then $\mathbf{t}^{\sigma'} - \mathbf{c}^{\mathbf{u},\mathbf{w}}$ is a fixed point of $\mathbf{f}$ because $\mathbf{f}(\mathbf{t}^{\sigma'} - \mathbf{c}^{\mathbf{u},\mathbf{w}})_i = f_{w_i}(\mathbf{t}^{\sigma'}) = \mathbf{r}_{w_i}^{*,\sigma'} = \mathbf{r}_{u_i}^{*,\sigma'} - c_{u_i,w_i} = (\mathbf{t}^{\sigma'} - \mathbf{c}^{\mathbf{u},\mathbf{w}})_i$. Thus $\mathbf{t}^\sigma \leq \mathbf{t}^{\sigma'}$. Since $\mathbf{f}$ is non-decreasing, then $\mathbf{r}_z^{*,\sigma'} = f_z(\mathbf{t}^{\sigma'}) \geq f_z(\mathbf{t}^\sigma) = \mathbf{r}_z^{*,\sigma}$ for any z, and for $u_1, u_2, \ldots, u_n$ the inequality is strict: $\mathbf{r}_{u_i}^{*,\sigma'} - c_{u_i,w_i} = \mathbf{r}_{w_i}^{*,\sigma'} \geq \mathbf{r}_{w_i}^{*,\sigma} > \mathbf{r}_{v_i}^{*,\sigma} + c_{u_i,v_i} - c_{u_i,w_i} = \mathbf{r}_{u_i}^{*,\sigma} - c_{u_i,w_i}$.

Thus, switching to the new *SM* strategy σ', we get $\mathbf{r}^{*,\sigma'}$ which dominates $\mathbf{r}^{*,\sigma}$, and is strictly greater in some coordinates, including all the u_i's. There are finitely many *SM* strategies, thus repeating this we eventually reach some *SM* strategy σ^* that can't be improved. Thus $\mathbf{r}^{*,\sigma^*} = P(\mathbf{r}^{*,\sigma^*})$, and by our earlier claim $\mathbf{r}^{*,\sigma^*} = \mathbf{r}^*$. Thus, maximizer has an optimal *SM* strategy, arrived at via simultaneous strategy improvement.

Since each local improvement step can be done in P-time and increases the sum total reward, the problem is in PLS. □

4. The complexity of reward 1-RMDPs and 1-RSSGs

Theorem 8. *There is a P-time algorithm for computing the exact optimal value (including the possible value ∞) of a 1-RMDP with positive rewards, in both the case where the single player aims to maximize, or to minimize, the total reward.*

We consider maximizing and minimizing 1-RMDPs separately.

4.1. Maximizing reward 1-RMDPs

We are given a maximizing reward 1-RMDP (i.e., no $Type_{\min}$ nodes in the 1-RSSG). Let us call the following LP "*max-LP* ":
Minimize $\sum_{u \in Q} x_u$
Subject to:

$x_u = 0$	for all $u \in Type_0$
$x_u \geq \sum_{v \in n(u)} p_{u,v}(x_v + c_{u,v})$	for all $u \in Type_{rand}$
$x_u \geq x_{en} + x_{(b,ex')} + c_u$	for all $u = (b, en) \in Type_{call}$; ex' is the exit of $Y(b)$.
$x_u \geq (x_v + c_{u,v})$	for all $u \in Type_{\max}$ and all $v \in n(u)$
$x_u \geq 0$	for all vertices $u \in Q$

We show that, when the value vector $\mathbf{r}^*$ is finite, it is precisely the optimal solution to the above max-LP, and furthermore that we can use this LP to find and eliminate vertices u for which $r^*_u = \infty$. Note that if $\mathbf{r}^*$ is finite then it fulfills all the constraints of the max-LP, and thus it is a feasible solution. We will show that it must then also be an optimal feasible solution. We first have to detect vertices u such that $\mathbf{r}^*_u = \infty$. For the max-linear equation system P, we define the underlying directed *dependency graph* G, where the nodes are the set of vertices, Q, and there is an edge in G from u to v if and only if the variable x_v occurs on the right hand side in the equation defining variable x_u in P. We can decompose this graph in linear time into strongly connected components (SCCs) and get an SCC DAG, $SCC(G)$, where the set of nodes are SCCs of G, and an edge goes from one SCC A to another B, iff there is an edge in G from some node in A to some node in B. We will call a subset $U \subseteq Q$ of vertices *proper* if all vertices reachable in G from the vertices in U are already in U. We also use U to refer to the corresponding set of variables. Clearly, such a proper set U must be a union of SCCs, and the equations restricted to variables in U do not use any variables outside of U, so they constitute a proper equation system on their own. For any proper subset U of G, we will denote by max-LP$|_U$ a subset of equations of max-LP, restricted to the constraints corresponding to variables in U and with the new objective $\sum_{u \in U} x_u$. Analogously we define $P|_U$, and let $\mathbf{x}|_U$ be the vector $\mathbf{x}$ with entries outside of U removed.

Proposition 9. *Let U be any proper subset of vertices.*
(I) The vector $\mathbf{r}^*|_U$ is the LFP of $P|_U$.
(II) If $r^*_u = \infty$ for some vertex u in an SCC S of G, then $r^*_v = \infty$ for all $v \in S$.
(III) If $\mathbf{r}'$ is an optimal bounded solution to max-LP$|_U$*, then $\mathbf{r}'$ is a fixed point of $P|_U$.*
(IV) If max-LP$|_U$ has a bounded optimal feasible solution $\mathbf{r}'$, then $\mathbf{r}' = \mathbf{r}^*|_U$.

Proof. Part (I) follows immediately from the definitions. Part (II) follows by induction on the length of the shortest path from any vertex $v \in S$ to u. In particular, if $x_v = \max\{x_w, \ldots\}$, and $r^*_w = \infty$, then $r^*_v = \infty$, and likewise for other vertex types. For part (III), observe that for each vertex $u \in Type_{\max}$, if $\mathbf{r}'$ is an optimal bounded solution of the max-LP, then at least one of the constraints $x_u \geq x_v + c_{u,v}$ holds *tightly*, i.e., $x_u = x_v + c_{u,v}$. For otherwise, we could decrease the value of x_u, letting $x_u = \max_{v \in n(u)}(x_v + c_{u,v})$, and still satisfy all constraints. The fact that the other types of inequalities are satisfied tightly follows similarly. For part (IV), if max-LP$|_U$ has a feasible bounded solution, then the optimal (minimum) solution $\mathbf{r}'$ is bounded. From part (III), we know $\mathbf{r}'$ is a

fixed point of $P|_U$, but then from the objective function of max-LP$|_U$, we know that $\mathbf{r}'$ is the LFP of $P|_U$, so we must have $\mathbf{r}' = \mathbf{r}^*|_U$. □

Theorem 10. *We can compute* $\mathbf{r}^*$ *for the max-linear equation system P, including the values that are infinite, in time polynomial in the encoding size of the 1-RMDP.*

Proof. Build the dependency graph G of P and decompose it into SCC DAG $SCC(G)$. We will find the LFP solution to P, bottom-up starting at a bottom SCC, S_1. We solve max-LP$|_{S_1}$ using a P-time LP algorithm. If the LP is feasible then the optimal (minimum) value is bounded, and we plug in the values of the (unique) optimal solution as constants in all other constraints of max-LP. We know this optimal solution is equal to $\mathbf{r}^*|_{S_1}$, since S_1 is *proper*. We do the same, in bottom-up order, for remaining SCCs S_2, $\ldots$, S_l. If at any point after adding the new constraints corresponding to the variables in an SCC S_i, the LP is *infeasible*, we know from Proposition 9 (IV), that at least one of the values of $\mathbf{r}^*|_{S_i}$ is ∞. So by Proposition 9 (II), all of them are. We can then mark all variables in S_i as ∞, and also mark all variables in the SCCs that can reach S_i in $SCC(G)$ as ∞. Also, at each step we add to a set U the SCCs that have finite optimal values. At the end we have a maximal *proper* such set U, i.e., every variable outside of U has value ∞. We label the variables not in U with ∞, obtaining the vector $\mathbf{r}^*$. All of this can be done easily in polynomial time. □

Algorithm 1 summarizes all the steps necessary to compute the optimal solution for maximizing 1-RMDPs with positive rewards.

4.2. Minimizing reward 1-RMDPs

Given a minimizing reward 1-RMDP (i.e., no $Type_{\max}$ nodes) we want to compute $\mathbf{r}^*$. Call the following LP "*min-LP:* "
Maximize $\sum_{u \in Q} x_u$
Subject to:

$x_u = 0$	for all $u \in Type_0$
$x_u \leq \sum_{v \in n(u)} p_{u,v}(x_v + c_{u,v})$	for all $u \in Type_{rand}$
$x_u \leq x_{en} + x_{(b,ex')} + c_u$	for all $u = (b, en) \in Type_{call}$; ex' is the exit of $Y(b)$.
$x_u \leq (x_v + c_{u,v})$	for all $u \in Type_{\min}$ and all $v \in n(u)$
$x_u \geq 0$	for all vertices $u \in Q$

Lemma 11. *For any* proper *set U, if an optimal solution* $\mathbf{x}$ *to* min-LP$|_U$ *is bounded, it is a fixed point of the min-linear operator* $P|_U$*. Thus, if min-LP*$|_U$ *has a bounded optimal feasible solution then* $\mathbf{r}^*|_U$ *is bounded (i.e., is a real vector).*

Proof. Note that if an optimal solution $\mathbf{x}$ to *min-LP*$|_U$ is bounded then for each vertex $u \in Type_{\min}$, for at least one of the constraints $x_u \leq x_v + c_{u,v}$ we have equality, i.e., $x_u = x_v + c_{u,v}$. For otherwise, we could increase the value of x_u, letting $x_u = \min_{v \in n(u)}(x_v + c_{u,v})$, and still satisfy all the constraints. Similarly the equality holds for all the other types of vertices. Therefore, $\mathbf{x}$ is a fixed point of $P|_U$ and because we showed $\mathbf{r}^*|_U$ to be the least fixed point of $P|_U$, $\mathbf{r}^*|_U$ has to be bounded as well. □

Algorithm 1: An algorithm for computing the optimal expected reward in maximizing 1-RMDP with positive rewards.

Input: A maximizing 1-RMDP with positive rewards $A = (A_1, \ldots, A_k)$, where $A_i = (N_i, B_i, Y_i, En_i, Ex_i, \mathtt{pl}_i, \delta_i, \xi_i)$ and $Q = \cup_i Q_i$ is the set of vertices.

Output: For all $u \in Q$, $x^*_u = r^*_u$, which is the optimal value from $\langle \varepsilon, u \rangle$ in A.

Construct the dependency graph, i.e., the digraph $G = (V, E)$ that has nodes $V = Q$ (the set of vertices of A), and edges E consisting of $\{(u, v) \mid u \in Q, v \in n(u)\}$ (the edges of A) and for each call port $u = (b, en)$ we also include edges (u, en) and $(u, (b, ex'))$ where ex' is the exit of $Y(b)$.

Find a bottom-up SCC decomposition of G and denote it by $(V_1, \ldots, V_l)$.

for $i = 1, \ldots, l$ **do**

  if there is an edge from a node of V_i to a node $v \in V_j$, $j < i$ where $x^*_v = \infty$ **then**

    set $x^*_u = \infty$ for all $u \in V_i$

  else

    Solve the following linear program in variables $x_u, u \in V_i$; for all occurrences below of $x_v, v \in V_j$ with $j < i$ we substitute the previously computed values x^*_v.

    Minimize $\sum_{u \in V_i} x_u$

    Subject to:

$x_u = 0$	for all $u \in Type_0 \cap V_i$
$x_u \geq \sum_{v \in n(u)} p_{u,v}(x_v + c_{u,v})$	for all $u \in Type_{rand} \cap V_i$
$x_u \geq x_{en} + x_{(b,ex')} + c_u$	for all $u = (b, en) \in Type_{call} \cap V_i$ where ex' is the exit of $Y(b)$.
$x_u \geq (x_v + c_{u,v})$	for all $u \in Type_{\max} \cap V_i$ and all $v \in n(u)$
$x_u \geq 0$	for all vertices $u \in V_i$

    If the above program is infeasible then set $x^*_u = \infty$ for all $u \in V_i$. Otherwise set the values of x^*_u for all $u \in V_i$ to the just found optimal solution.

From min-LP we can remove variables $x_u \in Type_0$, by substituting their occurrences with 0. *Assume, for now, that we can also (efficiently) find all variables x_u such that $r^*_u = \infty$.* By removing these variables, and eliminating appropriately their occurrences in all the constraints where they occur, we obtain a new operator P', and a new LP, min-LP$'$.

Lemma 12. *If ∞ and 0 nodes have been removed, i.e., if $\mathbf{r}^* \in (0, \infty)^n$, then $\mathbf{r}^*$ is the unique optimal feasible solution of min-LP$'$.*

Proof. By Corollary 3, player 2 has an optimal SM strategy, call it τ, which yields the finite optimal reward vector r^*. Once strategy τ is fixed, we can define a new equation system $P'_\tau(\mathbf{x}) = A_\tau \mathbf{x} + b_\tau$, where A_τ is a nonnegative matrix and b_τ is a vector of average rewards per single step from each node, obtained under strategy τ. We then have $\mathbf{r}^* = \lim_{k\to\infty}(P'_\tau)^k(0)$, i.e., $\mathbf{r}^*$ is the LFP of $x = P'(x)$.

Proposition 13. *(I)* $\mathbf{r}^* = (\sum_{k=0}^{\infty} A_\tau^k) b_\tau$.
(II) If $\mathbf{r}^*$ is finite, then $\lim_{k\to\infty} A_\tau^k = 0$, and thus $(I - A_\tau)^{-1} = \sum_{i=0}^{\infty}(A_\tau)^i$ exists (i.e., is a finite real matrix).

Proof. (I): $\mathbf{r}^* = \lim_{k\to\infty}(P'_\tau)^{k+1}(0) = \lim_{k\to\infty} A_\tau (P'_\tau)^k(0) + b_\tau = \lim_{k\to\infty}(\sum_{i=0}^{k}(A_\tau)^k) b_\tau$. (This holds regardless of whether $\mathbf{r}^*$ is finite. We shall use this fact in a subsequent proof.)
(II): since $\mathbf{r}^* = P'_\tau(\mathbf{r}^*)$, we have, for any $k \geq 0$, $\mathbf{r}^* = A_\tau^k \mathbf{r}^* + (I + A_\tau + A_\tau^2 + \ldots + A_\tau^{k-1}) b_\tau$. The second part of the right hand side, in the limit, is equal to $\mathbf{r}^*$, thus $A_\tau^k \mathbf{r}^*$ in the limit is an all-zero vector. It follows that the limit of A_τ^k is an all-zero matrix since all the entries/rewards in $\mathbf{r}^*$ are positive (we have already removed 0 entries). □

Now pick an optimal *SM* strategy τ for player 2 that yields the finite $\mathbf{r}^*$. We know that $\mathbf{r}^* = (I - A_\tau)^{-1} b_\tau$. Note that $\mathbf{r}^*$ is a feasible solution of the min-LP$'$. We show that for any feasible solution $\mathbf{r}$ to min-LP$'$, $\mathbf{r} \leq \mathbf{r}^*$. From the LP we can see that $\mathbf{r} \leq A_\tau \mathbf{r} + b_\tau$ (because this is just a subset of the constraints) and in other words $(I - A_\tau)\mathbf{r} \leq b_\tau$. We know that $(I - A_\tau)^{-1}$ exists and is non-negative (and finite), so multiply both sides by $(I - A_\tau)^{-1}$ to get $\mathbf{r} \leq (I - A_\tau)^{-1} b_\tau = \mathbf{r}^*$. Thus $\mathbf{r}^*$ is the optimal feasible solution of min-LP$'$. □

For $u \in Q$, consider the LP: **Maximize** x_u, **subject to:** the same constraints as min-LP, except, again, remove all variables $x_v \in Type_0$. Call this u-min-LP$'$.

Theorem 14. *In a minimizing 1-RMDP, for all vertices u, value $\mathbf{r}^*_u$ is finite iff u-min-LP$'$ is feasible and bounded. Thus, combined with Lemma 12, we can compute the exact value (even if ∞) of minimizing reward 1-RMDPs in P-time.*

We first need some preliminary claims. Let W be the set of vertices u such that u-min-LP$'$ is bounded and let S be the minimum *proper* set such that $W \subseteq S$. From min-LP remove all the constraints for variables outside of the set S and remove the variables of $Type_0$ in the same way as before. Call this set of constraints LP_S.

Proposition 15. *For any two vectors $\mathbf{x} = [x_1, x_2, \ldots, x_n], \mathbf{y} = [y_1, y_2, \ldots, y_n]$ and vector $\mathbf{z} = \max(\mathbf{x}, \mathbf{y}) = [\max(x_1, y_1), \max(x_2, y_2), \ldots, \max(x_n, y_n)]$, and subset $A \subseteq \{1, 2, \ldots, n\}$, and constants $p_{ij} \geq 0, c_{i,j} \geq 0$ we have that:*

1. *if vectors* $\mathbf{x}, \mathbf{y}$ *fulfill a linear constraint* $\widetilde{x_i} \leq \sum_{j \in A} p_{ij}(\widetilde{x_j} + c_{i,j})$ *then so does* $\mathbf{z}$
2. *if vectors* $\mathbf{x}, \mathbf{y}$ *fulfill a constraint* $\widetilde{x_i} \leq \min_{j \in A}(\widetilde{x_j} + c_{i,j})$ *then so does* $\mathbf{z}$

Proof.

1. Function *max* is monotonic, hence if $x_i \leq x_j$ and $y_i \leq y_j$, then $\max(x_i, y_i) \leq \max(x_j, y_j)$. Thus $\max(x_i, y_i) \leq \max\left(\sum_{j \in A} p_{ij}(x_j + c_{i,j}), \sum_{j \in A} p_{ij}(y_j + c_{i,j})\right)$ based on the fact that they fulfill the underlying constraint. However we know that for all j we have that $x_j \leq \max(x_j, y_j) = z_j$ and $y_j \leq \max(x_j, y_j) = z_j$, hence $\sum_{j \in A} p_{ij}(x_j + c_{i,j}) \leq \sum_{j \in A} p_{ij}(z_j + c_{i,j})$ and $\sum_{j \in A} p_{ij}(y_j + c_{i,j}) \leq \sum_{j \in A} p_{ij}(z_j + c_{i,j})$, which means that $z_i = \max(x_i, y_i) \leq \max\left(\sum_{j \in A} p_{ij}(x_j + c_{i,j}), \sum_{j \in A} p_{ij}(y_j + c_{i,j})\right) \leq \sum_{j \in A} p_{ij}(z_j + c_{i,j})$
2. Again we know that $\max(x_i, y_i) \leq \max\left(\min_{j \in A}(x_j + c_{i,j}), \min_{j \in A}(y_j + c_{i,j})\right)$ and for all j we have $x_j + c_{i,j} \leq z_j + c_{i,j}$ and $y_j + c_{i,j} \leq z_j + c_{i,j}$. We also know that the *min* function is monotonic, hence $\min_{j \in A}(x_j + c_{i,j}) \leq \min_{j \in A}(z_j + c_{i,j}) \geq \min_{j \in A}(y_j + c_{i,j})$. This means that $z_i = \max(x_i, y_i) \leq \max\left(\min_{j \in A}(x_j + c_{i,j}), \min_{j \in A}(y_j + c_{i,j})\right) \leq \min_{j \in A}(z_j + c_{i,j})$.

□

Corollary 16. *For any two feasible solutions* $\mathbf{x}, \mathbf{y}$ *to* LP_S *we have that* $\mathbf{z} = \max(\mathbf{x}, \mathbf{y}) = [\max_i(\mathbf{x}_i, \mathbf{y}_i)]$ *(vector with entries being the maximum of the respective entries in* $\mathbf{x}$ *and* $\mathbf{y}$*) is a feasible solution to* LP_S *as well.*

Proof. (**of Theorem 14.**)

($\Rightarrow$) First let us show that for any u if $\mathbf{r}^*_u$ is finite, then u-min-LP$'$ has to be feasible and bounded. Feasibility is easy as an all zero vector $\mathbf{0}$ fulfills all the constraints in u-min-LP$'$.

Now pick the optimal *SM* strategy τ for the *min* player that yields the optimal reward vector $\mathbf{r}^*$ and take any feasible vector $\mathbf{x}$. From the u-min-LP$'$ we can see that $\mathbf{x} \leq A_\tau \mathbf{x} + b_\tau$ (because this is just a subset of the constraints). Since we removed all zero reward nodes, i.e., exits of components, then all entries of b_τ are positive and from Lemma 4 we can get that $\mathbf{x} \leq (\sum_{k=0}^{\infty} A_\tau^k) b_\tau$. However by Proposition 13 (I) (which holds regardless of whether $\mathbf{r}^*$ is finite) this means that $\mathbf{x} \leq \mathbf{r}^*$ for any feasible $\mathbf{x}$.

For contradiction, assume u-min-LP$'$ was feasible but unbounded. Then there would exist a sequence of feasible vectors $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \ldots$ such that $\lim_{k \to \infty} \mathbf{x}^k_u = \infty$. But we know that $\mathbf{x}^k \leq \mathbf{r}^*$ for all k, thus $\mathbf{r}^*_u$ would have to be infinite, contradicting our assumption.

($\Leftarrow$) Now let us show that if u-min-LP$'$ is feasible and bounded then $\mathbf{r}^*_u$ has to be finite. Consider an LP with LP_S constraints and with the objective: **Maximize** $\sum_{u \in W} x_u$. Call it *W-min-LP* and for any optimal solution $\mathbf{x}^*$ denote by $\overline{\mathbf{x}}^*$ the vector filled with values from $\mathbf{x}^*$ for $u \in W$ and ∞ for all $u \in S \setminus W$. Notice that $\overline{\mathbf{x}}^*$ is unique, because if the value of two optimal vectors $\mathbf{x}$ and $\mathbf{x}'$ differ at an entry $u \in W$, then $\max(\mathbf{x}, \mathbf{x}')$ is also feasible thanks to Corollary 16, and this would contradict their optimality.

Lemma 17. *The vector $\overline{\mathbf{x}}^*$ is a fixed point of $P|_S$.*

Proof. Since for every x_u, $u \in W$, u-min-LP′ is bounded, and we removed from u-min-LP′ only the constraints that these variables do not depend on (even in a transitive way), the maximum value of x_u cannot possibly increase after we remove these constraints, because that would mean x_u could have been assigned a higher value in u-min-LP′. Hence the LP W-min-LP is feasible and bounded.

Now we show that for an optimal solution $\mathbf{x}^*$, no constraint with a variable x_u, $u \in W$, on the left hand side can hold tightly (i.e., with equality) when there is a variable x_v, $v \in S \setminus W$, on the right hand side. Let us take some optimal solution $\mathbf{x}^*$ to W-min-LP. Notice that $S \setminus W = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices whose corresponding variables are unbounded, i.e., v_i-min-LP is unbounded. We know that for each of them there is a sequence of feasible solutions $\mathbf{x_1}^{v_i}, \mathbf{x_2}^{v_i}, \mathbf{x_3}^{v_i}, \ldots$ to v_i-min-LP (the bold subscripts denote the position in this sequence, not inside the vector), such that the value of entry x_{v_i} in this sequence of vectors is nondecreasing and becomes arbitrarily large. If we project this sequence onto the variables in S then $\mathbf{x_1}^{v_i}|_S, \mathbf{x_2}^{v_i}|_S, \mathbf{x_3}^{v_i}|_S, \ldots$ is a sequence of feasible solutions to W-min-LP, such that x_{v_i} becomes arbitrarily large. Now construct a sequence of vectors $\mathbf{x}'_i = \max(\mathbf{x}^*, \mathbf{x_i}^{v_1}|_S, \mathbf{x_i}^{v_2}|_S, \ldots, \mathbf{x_i}^{v_n}|_S)$. By Corollary 16 we know that all vectors in this sequence are feasible solutions to W-min-LP. We also know that all of them are optimal solutions, because we always take the maximum of the entries, including the ones in the optimal solution $\mathbf{x}^*$. So we obtain as high a value of the objective function $\sum_{u \in W} x_u$ as before, and we cannot improve this value as it would contradict the assumption that $\mathbf{x}^*$ was optimal. Now notice three things:

1. Since every variable x_u, $u \in W$, is bounded, at some point in this sequence, we will reach a point such that the r.h.s. of any constraint which involves some variable x_v, $v \in S \setminus W$, will be larger than the highest possible value of all variables corresponding to vertices in W. This means that at that point there cannot be a constraint that holds with equality such that x_u, $u \in W$, is the l.h.s. and where there is a variable x_v, $v \in S \setminus W$, on the r.h.s.

2. For all k, for every x_u, $u \in W$, there has to be some constraint with x_u on the l.h.s. such that $\mathbf{x}'_k$ satisfies this constraint tightly, with equality, because otherwise we could increase the value of x_u without altering the value of any other variables, to obtain a larger value for the objective, which would contradict the optimality of $\mathbf{x}'_k$.

3. All variables x_v, $v \in S \setminus W$, become arbitrarily large in this sequence, thus it cannot be the case that there are only variables corresponding to vertices in W on the r.h.s. in any constraint with x_v on the l.h.s. (that would force this variable to be bounded).

Using these facts, we can see that for a large enough k, from the vector $\mathbf{x}'_k$ we can construct a vector $\overline{\mathbf{x}}^*$ which a fixed point of $P|_S$. We do so by setting the value of all variables x_v, $v \in S \setminus W$ to ∞, and leaving the value of all variables x_u, $u \in W$, unchanged in $\mathbf{x}'_k$. The claim that $\overline{\mathbf{x}}^*$ is a fixed point of $P|_S$ follows because for every variable x_u, $u \in W$, of type $Type_{rand}$ or $Type_{call}$, $\mathbf{x}'_k$ satisfies the correlated constraint with x_u on

the l.h.s. with equality, and this can only be the case if the r.h.s. of that constraint contains only variables corresponding to vertices in W, and thus $\overline{\mathbf{x}}^*$ also satisfies this constraint with equality. Likewise, for variables x_u, $u \in W$, of type $Type_{min}$, for $\mathbf{x}'_k$ all constraints such that x_u is the l.h.s. and there is at least one variable corresponding to a vertex in $S \setminus W$ on the r.h.s., must hold with strict inequality. Hence, since equality must hold in $\mathbf{x}'_k$ for one of the constraints involving x_u on the l.h.s., there must exist one such constraint such that the r.h.s. only involves variables corresponding to vertices in W. Thus, equality also holds for these constraints for $\overline{\mathbf{x}}^*$ for these variables. Thus $\overline{\mathbf{x}}^*$ satisfies the corresponding *min* equation in $P|_S$. Also for variables in x_v, $v \in S \setminus W$, all the equations in $P|_S$ will clearly be fulfilled after setting their values to ∞, because both sides of the equations where x_v occurs have at least one variable corresponding to a vertex in $S \setminus W$, and that makes the value of both sides of this equation ∞. □

Now finally we can finish the proof of Theorem 14, using the previous lemma. Since we know that $\mathbf{r}^*|_S$ is the LFP of the operator $P|_S$, it must be that $\mathbf{r}^*|_S \leq \overline{\mathbf{x}}^*$, which means that for all $u \in W$ we have that $\mathbf{r}^*_u|_S \leq \overline{\mathbf{x}}^*_u = \mathbf{x}^*_u$, which is finite. □

Algorithm 2 summarizes the steps needed to compute the optimal values in a minimizing 1-RMDP with positive rewards. Note that the variables equal to 0 are not removed as this is not really needed. Also note that the linear programs are feasible: $x = 0$ is a feasible solution. As compared with Algorithm 1, Algorithm 2 has to solve more and larger linear programs unless the dependency graph G of P is strongly connected.

Algorithm 2: An algorithm for computing the optimal expected reward in minimizing 1-RMDP with positive rewards.

Input: A minimizing 1-RMDP with positive rewards $A = (A_1, \ldots, A_k)$, where $A_i = (N_i, B_i, Y_i, En_i, Ex_i, \mathtt{pl}_i, \delta_i, \xi_i)$, and $Q = \cup_i Q_i$ is the set of vertices.

Output: For all $u \in Q$, $x^*_u = r^*_u$, which is the optimal value from $\langle \varepsilon, u \rangle$ in A.

1 **for** $w \in Q$ **do**

2 Solve the following linear program.

Maximize x_w

Subject to:

$x_u = 0$	for all $u \in Type_0$
$x_u \leq \sum_{v \in n(u)} p_{u,v}(x_v + c_{u,v})$	for all $u \in Type_{rand}$
$x_u \leq x_{en} + x_{(b,ex')} + c_u$	for all $u = (b, en) \in Type_{call}$ where ex' is the exit of $Y(b)$.
$x_u \leq (x_v + c_{u,v})$	for all $u \in Type_{\max}$ and all $v \in n(u)$
$x_u \geq 0$	for all vertices $u \in Q$

3 If the above program is unbounded then set $x^*_w = \infty$ and otherwise set x^*_w to be the optimal value of its objective.

4.3. Complexity of (1-)RSSGs with positive rewards

Theorem 18. *Deciding whether the value r^*_u of a given 1-RSSG positive reward game is $\geq q$ for a given $q \in [0, \infty]$, is in NP ∩ coNP.*

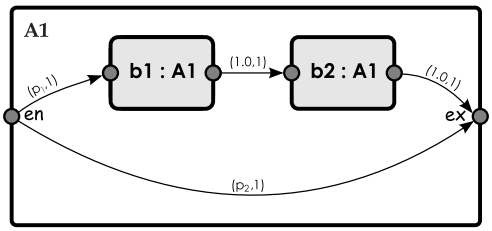

Figure 2: Standard 1-RMC gadget used in proof of Theorem 19

Proof. Both the membership in NP and membership in coNP follow from the P-time upper bounds for 1-RMDPs and SM-determinacy: For membership in NP, guess a SM strategy for the maximizing player, compute the value for the resulting minimizing 1-RMDP and verify that it is $\geq q$. For membership in coNP, guess a SM strategy for the minimizing player, compute the value for the resulting maximizing 1-RMDP and verify that it is $< q$. □

We will show now that the qualitative problem of testing whether the maximizing player can achieve infinite reward in a 1-RSSG is at least as hard as Condon's quantitative decision problem for finite SSGs. Recall that in Condon's problem, we are given a finite SSG, without any rewards, with a designated starting state u and target terminal state t. The objective of the maximizing player is to maximize the probability that the trajectory starting at u eventually reaches state t, and the objective of the minimizing player is to minimize this probability. The quantitative problem is, given finite SSG G and rational number q, is the value of the game $\leq q$? It is well-known (and easy to see) that the problem is polynomially equivalent to the special case that $q = 1/2$. The problem is in NP $\cap$ coNP, and it is a long-standing open question whether it is in P or not [7].

Theorem 19. *Condon's quantitative termination problem for finite SSGs reduces in P-time to the problem of deciding whether $r^*_u = \infty$.*

Proof. Consider the standard 1-RMC from [16], depicted in Figure 2. From the entry, *en*, this 1-RMC goes with probability p_1 to a sequence of two boxes labeled by the same component and with probability p_2 goes to the exit. We assume $p_1 + p_2 = 1$. As shown in ([16], Theorem 3), in this 1-RMC the probability of termination starting at $\langle \epsilon, en \rangle$ is = 1 if and only if $p_2 \geq 1/2$.

Now, given a finite SSG, G with a starting node u and target terminal node t, do the following: First "clean up" G by removing all nodes where the min player (player 2) has a strategy to achieve probability 0 of the trajectory reaching t. We can do this in polynomial time. If u is among these nodes, we would already be done, so assume it is not. The revised SSG will have two designated terminal nodes, the old terminal node t, labeled "1", and another new terminal node labeled "0". From every node v in the revised SSG which does not carry full probability on its outedges, we direct all the "residual" probability to "0", i.e., we add an edge from v to "0" and assign probability $p_{v,}$"0" $= 1 - \sum_w p_{v,w}$ to it, where the sum is over all remaining nodes w

in the SSG. In the resulting finite SSG, we know that if the max player plays with an optimal memoryless strategy (which it has), and the min player plays arbitrarily with a memoryless strategy, there is no bottom SCC in the resulting finite Markov chain other than the two designated terminating nodes "0" and "1". In other words, all the probability exits the system, as long as the maximizing player plays optimally. Note also that, importantly, the "expected time" that it takes for the probability to exit the system when max player plays optimally is finite (because there are no "null recurrent" nodes in a finite Markov chain).

Another way to put this fact is as follows: consider the resulting SSG to be a finite reward SSG with reward 1 on each transition, and switch the role of the max and min player, and now the goal of the max player is to maximize the total reward before termination (at either exit), and that of the min player is to minimize it. Translating the above to this setting, the "cleaned up" SSG has the property that the min player has a memoryless strategy using which, no matter what the maximizer does, the total reward will be finite: we will terminate, at "0" or at "1", in finite expected time (because there are no "null recurrent" nodes in finite Markov chains, and both players have optimal memoryless strategies).

Now, take the remaining finite SSG, call it G'. Just put a copy of G' at the entry of the component A_1 of the 1-RMC, identifying the entry en with the initial node, u, of G'. Take every edge that is directed into the terminal node "1" of G', and instead direct it to the exit ex of the component A_1. Next, take every edge that is directed into the terminal "0" node and direct it to the first call, (b_1, en) of the left box b_1. Both boxes map to the unique component A_1. Call this 1-RSSG A.

We now claim that $q^*_u \leq 1/2$ in the finite SSG G' for terminating at the terminal "1" iff $r^*_u = \infty$ for expected reward value in the resulting reward 1-RSSG, A (recall: with the role min and max reversed, and with all transitions having reward 1).

The reason is as follows: we know that in A the minimizer has at least one SM strategy that obtains finite reward inside any copy of G', and it must play one such strategy each time it goes through G' if it wants to avoid payoff ∞.

Now, there are only a finite number of SM strategies for minimizer inside G' which yield a finite expected reward (after an optimal response by the maximizer). Let $D \in [0, \infty)$ be the maximum finite expected reward among those SM strategies. Also, no matter what the two players do, we know we will earn reward at least 1, each time we go through G'. So, each time going through G' we accumulate a reward $D' \in [1, D]$. Therefore, from the point of view of trying to make sure the total expected reward is finite, it is really of no relevance what the specific value of D' is when we go through G'. Rather, what is important is whether we "visit" a copy of G', i.e., a copy of the entry u, infinitely often.

Now, to make sure that the expected number of times u is visited is finite, the minimizer must in fact maximize the probability of terminating at "1", and thus minimize the probability of termination at "0". In addition, the minimizer must also make sure that the expected reward inside G' is finite, but this we know it can do while maximizing the probability of terminating at "1". Thus, the total reward $r^*_u = \infty$ precisely when the value of the SSG termination game G' is $\leq 1/2$. □

By contrast, for finite-state SSGs with strictly positive rewards, we can decide in P-

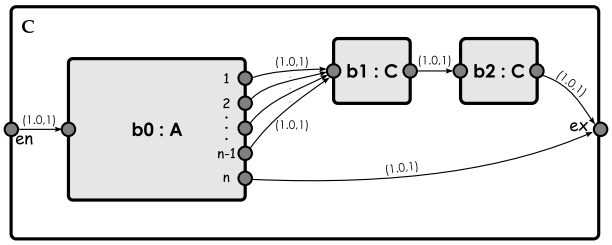

Figure 3: Multi-exit reward RMDP: undecidability

time whether the value is ∞, because this is the case iff the value of the corresponding termination game is not 1. This is basically because null-recurrence is not possible in finite state spaces. Deciding whether an SSG termination game has value 1 is in P-time (see, e.g., [17]).

5. Multi-exit RMDPs with positive rewards

In this section we show undecidability for multi-exit *minimizing* reward RMDPs.

Theorem 20. *For a given multi-exit positive reward* minimizing *RMDP, it is undecidable to distinguish whether the infimum expected total reward value starting at a given node is finite or* ∞.

Proof. We will use the construction of a component named A in the proof of Theorem 10.2 in [17]. This single-entry n-exit component relates RMDPs with n exits with Probabilistic Finite Automata (PFA) with n states. More precisely the supremum probability of termination at the n-th exit starting at the entry of A is equal to the supremum probability with which the correlated PFA accepts some word. It was proved in [1] that deciding whether a given PFA with 46 states accepts any word with probability greater than $\frac{1}{2}$ is undecidable. This means it is undecidable to resolve whether the supremum probability of termination at the n-th exit ($n = 46$) in the correlated RMDP A is greater than $\frac{1}{2}$.

To prove that it is also undecidable to resolve whether the infimum expected total reward starting from a given node in a RMDP with positive rewards is finite or ∞, we will combine the RMDP A with a gadget 1-RMDP C, as can be seen at Figure 3. Let us denote by p the supremum probability of termination at the n-th exit of the component A labeling box $b0$. We will argue that $p > 1/2$ iff the infimum expected total reward for the reward 1-RMDP C is finite.

To consider A inside a reward RMDP, we will place the same positive reward, say 1, on all transitions inside the component A.

We will need several observations about the component A from the proof of Theorem 23 in [17]. Firstly, one property of A is the following: for any strategy that yields probability > 0 of exiting from the n-th exit of component A, it must be the case that

the total probability of exiting from one of the exits of component A is 1. It is easy to verify this fact based on the structure of component A given in [17].

Now, first suppose $p > 1/2$. It follows from the previously mentioned fact that in the reward game the minimizer has a strategy with which to exit from A with probability 1, and simultaneously to exit from the n-th exit with probability $> 1/2$. Therefore, note that component C, under an optimal strategy played inside box $b0$, acts like the previously used gadget from Figure 2, in which the probability of exiting directly is p. For this gadget, if $p > 1/2$, we know that the resulting expected time until termination is finite.

Moreover, the component A from [17] has the following additional property: if $p > 1/2$, then the corresponding PFA accepts a finite word w with probability $p > 1/2$, and we can furthermore use the word w as a strategy σ_w in A such that starting at the entry of A, the strategy σ_w will exit A with probability 1, and will exit from the n-the exit with probability $p > 1/2$, and will exit from A in finite expected time $2|w|$. Thus the expected time taken until termination inside A, i.e., inside the box $b0$, is finite and hence the total expected time until termination starting at the entry of C is also finite.

Next suppose that $p \leq 1/2$. Then in C we either stay inside a copy of $b0$ (A) with non-zero probability, in which case the total reward is infinite, or else we exit from the n-th exit of every copy of A with probability $\leq 1/2$ and we exit from the other exits with probability $\geq 1/2$. It follows easily from the properties of the gadget in C that the expected termination time is infinite in such a case. Thus if we can decide whether the infimum expected total reward starting at the entry of C is finite or not, we can also decide whether the supremum termination probability p at the n-th exit of A is greater than $\frac{1}{2}$, which we know is undecidable. □

Theorem 20 leaves open whether it is undecidable to determine whether the *supremum* expected total reward for a given multi-exit positive reward *maximizing* RMDP is infinite.

A natural approach to attempt to prove such an undecidability result is to use the undecidability result established in [17] for determining whether the *minimum* termination probability for a multi-exit RMDP is = 1. However, there is a technical difficultly with attempting to adapt the proof of that result to the setting of maximizing positive reward RMDPs, which we have not been able to overcome.

We conjecture that indeed determining whether the *supremum* expected total reward for a given multi-exit positive reward *maximizing* RMDP is ∞ is undecidable, but we leave this as an open problem.

6. BSSGs, SCFG games with positive rewards, and equivalence to 1-RSSGs with positive rewards

In this section we explain the close relationship between 1-RMDPs and 1-RSSGs with positive reward, and MDP and stochastic game extensions of context-free grammars and branching processes, with both positive and non-negative rewards.

A *stochastic context-free grammar (SCFG) game*, is given by $G = (V, R, X_{start})$, where $V = V_0 \cup V_1 \cup V_2$ is a set of nonterminals, which is partitioned into three disjoint sets: V_0 are the probabilistic nonterminals (controlled by nature), V_1 and V_2,

the nonterminals controlled by players 1 and 2, respectively. $X_{start} \in V$ is the start nonterminal. R is a set of rules, where each rule $r \in R$ has the form $r = (X, p_r, c_r, Z_r)$, where $X \in V$, and if $X \in V_0$ then $p_r \in [0, 1]$ is a (rational) probability, otherwise, if $X \in V_i$, $i > 0$, then $p_r = \perp$, $c_r \in \mathbb{Q}_{\leq 0}$ is a rational positive reward (or non-negative reward, if we allow 0 reward), and $Z_r \in V^*$ is a (possibly empty) string of nonterminals. A rule $r = (X, p_r, c_r, Z_r)$ is often written also as $X \overset{(p_r,c_r)}{\rightarrow} Z_r$, where X is the left-hand side, Z_r the right-hand side, and (p_r, c_r) the label of the rule. For each nonterminal, X, let $R_X \subseteq R$ denote the set of rules that have X on the left hand side. For each $X \in V_0$ we have $\sum_{r=(X,p_r,c_r,Z_r)\in R_X} p_r = 1$.

The (countable) set of states of the game is a subset of V^*, i.e., strings of nonterminals. The precise game depends on the specific *derivation law* we use for the grammar, e.g., *left-most*, *right-most*, or *simultaneous*. The derivation rule that captures 1-RMDPs and 1-RSSGs exactly is the *left-most* derivation law, so we first describe the game corresponding to left-most derivation.[2] Again, states of the game are sequences of nonterminals. The game begins in the state X_{start}. In each round, if the state is $\mathcal{S} = X_1 \dots X_k$, then we proceed, by using a left-most derivation law, as follows: choose a rule $r = (X_1, p_r, c_r, Z_r) \in R_{X_1}$. If $X_1 \in V_0$ the rule r is chosen probabilistically among the rules r in R_{X_1}, according to the probabilities p_r. If $X_1 \in V_i$, $i \in \{1, 2\}$, then the rule r is chosen by player i. After the choice is made, the play moves to the new state $Z_r X_2 \dots X_k$. The reward gained in that round by player 1 is c_r. The game continues until (and unless) we reach the empty-string state $\mathcal{S} = \epsilon$. The total reward gained by player 1 is the sum total of the rewards over every round. A strategy for player $d \in \{1, 2\}$ is a mapping that, given the history of play ending in state $XW \in V^*$, where $X \in V_d$, maps it to a rule $r \in R_X$.[3] Fixing strategies for the two players, we obtain a (denumerable) reward Markov chain whose states are (a subset of) V^*, the total reward is a random variable defined over the trajectories (runs) of this Markov chain. Player 1's goal is to maximize the expected total reward, and player 2's goal is to minimize it.

Let us consider the games corresponding to other derivation laws. Specifically, the game with *right-most* derivation law is simply the mirror image of the one with left-most derivation: states are sequences of nonterminals, and in each round the remaining *right-most* nonterminal in the current derivation state $\mathcal{S}$ is expanded, either by the choice of the player who controls it, or probabilistically according to the given distribution, if it is a random nonterminal.

Finally, let us note that the game corresponding to the *simultaneous* derivation law is a bit different. Again, a state is a sequence of nonterminals. However, in the case of simultaneous derivation, in each round *all* remaining nonterminals in the current state $\mathcal{S}$ are expanded, by letting the player who controls each nonterminal choose a corresponding rule (or by choosing the rule randomly according to the given distribution, if that nonterminal is random).

It is worth pointing out that these games with the simultaneous derivation law are

[2] The game with left-most derivation is also equivalent to a BPA stochastic game with rewards; see, e.g., [2] and [3] where qualitative questions about BPA games without rewards were considered.

[3] We could more generally define strategies that can yield probability distributions on the next rule, but this won't be necessary, since indeed deterministic "stackless and memoryless" strategies are already optimal.

essentially equivalent to a *Branching Simple Stochastic Game* (BSSG), as defined and considered in [17], but with non-negative rewards and a total reward objective (as opposed to the objective of optimizing extinction probability). Note that, unlike left-most and right-most derivation, this definition of the game with simultaneous derivation law does not immediately yield a perfect information game (because the two players are not aware of each others' choices in each round). Nevertheless, just like 1-RMDPs and 1-RSSGs, these games have a value that arises as the LFP of a max-min-linear monotone system of equations (which are basically the corresponding Bellman equations in the 1-player BMDP case), and this is so even when 0 rewards are allowed.

Thus, the value for the 1-player MDP version of these simultaneous expansion games on SCFGs can be computed in P-time, even when 0 rewards are allowed. Furthermore, both players in these games have *static* optimal strategies meaning they have optimal strategies that are deterministic and which, irrespective of history or context, always expand any specific nonterminal N belonging to the given player using the same rule. Static strategies are the moral equivalent of deterministic stackless memoryless strategies for 1-RSSGs. The proofs of the above facts, which we will not provide in detail here, are fairly simple variations on the proofs of Theorems 1, 6, 8, and 18 regarding 1-RMDPs and 1-RSSGs with positive rewards.

For 1-RMDPs and 1-RSSGs, the only place where we used in a crucial way the fact that there are only *strictly positive* rewards on transitions, was in the proof of Theorem 1, establishing the correspondence between the values starting at each vertex of the 1-RSSG reward game and the LFP solution of the corresponding system of max-min-linear equations (see, e.g., part (2.c.) of that proof).

The reason why we do not require strictly positive rewards on grammar rules to establish such a correspondence in the setting with *simultaneous* derivation law is because, regardless whether some rules have reward 0 or not, with the simultaneous derivation law the total reward value obtained starting from a particular nonterminal controlled, e.g., by the maximizing player, will indeed equal the maximum over all rules associated with that nonterminal, of the sum total reward of values starting at the nonterminal occurrences on the right hand side of that rule.

However, if we were using left-most derivation, with 0 rewards on rules the correspondence to the LFP of the equations would in general fail.

Let us now explain why 1-RSSGs with positive rewards (non-negative rewards, respectively) are basically equivalent to SCFG games with *left-most* derivation law and with positive rewards (non-negative rewards, respectively). The proof is similar to the proof of equivalence between 1-RMCs and SCFGs with respect to probability of termination in [16], Theorem 2.3.

Given a SCFG game G with positive rewards we can construct a 1-RSSG A such that there is a correspondence between the states and strategies of the players, and such the two games have the same value. The 1-RSSG A has one component A_i for every nonterminal X_i of G, the component A_i has one entry en_i and one exit ex_i, the entry en_i has the same type 0, 1, or 2 as the corresponding nonterminal X_i. For each rule $r = (X_i, p_r, c_r, Z_r)$ of each nonterminal X_i in G, the corresponding component A_i contains a path from the entry en_i to the exit ex_i consisting of a sequence of a boxes that are mapped in order to the nonterminals in Z_r. That is, if $Z_r = X_{i_1} \dots X_{i_k}$, then the path contains k boxes $b_1, \dots, b_k$, where b_j is mapped to A_{i_j} for $j = 1, \dots, k$, the entry

node en_i has a transition to the call port of box b_1 with label $(p_r, c_r/(k+1))$, and the return port of each box b_j has a transition to the call port of the next box b_{j+1} (or the exit ex_i if $j = k$) with label $(1, c_r/(k+1))$; if $k = 0$, i.e., if $Z_r = \epsilon$, then there is a direct transition $en_i \to ex_i$ with label (p_r, c_r). The starting state of the 1-RSSG A is the entry of the component corresponding to the starting nonterminal X_{start} of the SCFG game G.

Conversely, given a 1-RSSG A with positive rewards, we can construct an equivalent SCFG game G as follows. For each vertex u of A that is not an exit, there is a corresponding nonterminal X_u in G which has the same type 0, 1, or 2 as the vertex u. For every transition $u \to v$ of A, there is a corresponding rule $X_u \to X_v$ in G if v is not an exit, or $X_u \to \epsilon$ if v is an exit; the label of the rule is the same as the label of the transition. For every call port $u = (b, en)$ of A, where b is a box that is mapped to a component A_i with exit ex_i, the SCFG game G contains a corresponding rule $X_{(b,en)} \overset{(1,c_u)}{\to} X_{en}X_{(b,ex_i)}$. Considering the equation system $x = P(x)$ for the 1-RSSG A (see Theorem 1) it is easy to see that the value of the 1-RSSG A starting at any vertex is equal to the value of the SCFG game starting at the corresponding nonterminal, and there is also a correspondence between the players' optimal strategies in the two games.

6.1. Some illustrative examples formulated as SCFG games, and further explanation of the role of 0 *rewards.*

We now describe some examples of 1-RSSG games using the simple (and expressively equivalent) formulation of these games as SCFG games.

We first use some examples formulated as SCFG games with left-most derivation to illustrate, as discussed in the introduction, that the condition of *strictly positive* rewards on rules/transitions is essential to avoid various pathological cases arising in such infinite-state games with rewards.

Indeed, consider the purely deterministic context-free grammar given by the rules: $\{X \overset{(\bot,0)}{\mapsto} XY \ ; \ X \overset{(\bot,0)}{\mapsto} \epsilon \ ; \ Y \overset{(\bot,7)}{\mapsto} \epsilon \}$, where X and Y are nonterminals belonging to the maximizing player, player 1. The notation is as follows: the pair (p, c) of quantities labelling a rule denotes the probability, p, of that rule firing, and the reward, c, accumulated for each use of that rule during a derivation, but when the nonterminal belongs to player 1 or 2, instead of a probability p we have the label $\bot$. In this example all nonterminals belong to player 1. Suppose the start nonterminal is X. If the deterministic game proceeds by left-most derivation, it is easy to see that there is no optimal strategy for maximizing player 1's total payoff. Indeed, there aren't even any ϵ-optimal strategies, because the supremum is ∞. In fact, if player 1 uses the rule $X \overset{(\bot,0)}{\mapsto} XY$, n times, to expand the left-most X in the derivation, and then uses $X \overset{(\bot,0)}{\mapsto} \epsilon$, and finally uses $Y \overset{(\bot,7)}{\mapsto} \epsilon$, n times to expand all n remaining Y nonterminals, the total reward is $7*n$. But no single strategy will gain a total reward of ∞. Note in particular that any "stackless and memoryless" strategy, which always picks one fixed rule for each nonterminal, regardless of the history of play and the remaining nonterminals (the "stack"), is the worst strategy possible: its total reward is 0. By contrast, if we require simultaneous expansion of all remaining nonterminals in each round, then there is a single "stackless

and memoryless" strategy that gains infinite reward, namely: in each round expand every copy of X using $X \overset{(\bot,0)}{\mapsto} XY$, and (simultaneously) expand every copy of Y using its unique rule. Clearly, after $n \geq 1$ rounds we accumulate $7 * (n-1)$ reward by doing this. Thus the total reward will be ∞.

Similarly, consider the simple grammar $\{X \overset{(\bot,0)}{\mapsto} XY \ ; \ Y \overset{(\bot,1)}{\mapsto} Y \}$, where, again, both nonterminals X, Y are controlled by the maximizing player, and X is the start nonterminal. Under the left-most derivation law, clearly the maximum reward is 0, whereas under the right-most or simultaneous derivation law, the total reward is ∞. So, the supremum total (expected) reward is not robust with respect to the derivation law, and can wildly differ depending on the derivation law, when 0 rewards are allowed on rules.

This is not the case when only strictly positive rewards are allowed on rules: in that case all derivation laws yield the same value for the resulting game.

Now let us consider a basic example with strictly positive rewards: consider the SCFG with rewards given by the following grammar rules: $\{X \overset{(1/3,3)}{\mapsto} XX \ ; \ X \overset{(2/3,2)}{\mapsto} \epsilon\}$. Here X is the only nonterminal. Consider now a random left-most derivation of this grammar, starting from the nonterminal X. What is the expected total reward accumulated during the entire derivation? It is not hard to see that if we let x denote the total expected reward, then x must satisfy the following equation: $x = (1/3 * (3 + (x + x))) + (2/3 * 2) = (2/3)x + (7/3)$. Therefore, the total expected reward is the unique solution to this equation, namely $x^* = 7$. Note that, in general, such a derivation may not terminate with probability 1, and that the expected reward need not be finite (consider the same grammar with modified probabilities: $\{X \overset{(2/3,3)}{\mapsto} XX \ ; \ X \overset{(1/3,2)}{\mapsto} \epsilon\}$).

As we have just seen, for 1-RMDPs (i.e., context-free MDPs with left-most derivation law), if we allow 0 rewards, then there may not even exist any ϵ-optimal strategies. Furthermore, even in a purely probabilistic setting without players (1-RMCs), with 0 rewards the expected total reward can be irrational. This follows from the fact that for 1-RMCs and SCFGs the total probability of termination can be irrational (see [16]), combined with the fact that we can easily use 0 rewards (in the left-most derivation setting for SCFGs) to encode the total *probability of termination* as the expected total reward. To do this, we simply do the following: add a new start nonterminal, S', to the grammar, as well as a new nonterminal Y. If the old start nonterminal was S, add the new rule $S' \rightarrow S\,Y$ to the grammar, with probability 1 and reward 0, and also add a rule $Y \rightarrow \epsilon$, with probability 1 and reward 1. Assign reward 0 to every rule of the old grammar. It is easy to check that the expected total reward for such a SCFG, using left-most derivation, and starting at the new start nonterminal S', is precisely the probability of eventual termination in the original SCFG.

When 0 rewards are allowed in the left-most derivation setting, even the decidability of determining whether the supremum expected reward for 1-RMDPs is greater than a given rational value is open, and subsumes other simpler open decidability questions, e.g., for the supremum reachability probability in non-reward 1-RMDPs. It is not even known whether it is decidable whether this *supremum* reachability probability is 1 (see [17]), whereas it was shown in [2] that it is decidable, in fact in polynomial time, whether there exists some strategy which achieves probability of termination equal to

1. (See also [3], where the two-player stochastic game version of qualitative reachability problems was considered.) We remark that in the case of BMDPs, the supremum reachability probability is 1 if and only if there is a strategy that achieves it, and this can be decided in polynomial time [14]. However, note that the equivalence between 1-RMDP and BMDP with respect to the extinction probability does *not* carry over to the reachability probability, for essentially the same reason that it does not hold in the reward model with 0 rewards.

Let us now explain further the reason why 0 rewards play a crucial role in the setting with left-most derivation (and thus for 1-RMDPs and 1-RSSGs). If we consider a derivation tree of the context-free grammar associated with such a game with rewards, then if we allow 0 rewards on rules it is entirely possible that a non-terminating infinite subtree (branch) of the derivation tree may nevertheless yield finite total reward. This however can not happen when all rules have strictly positive rewards associated with them: in that case any infinite derivation tree must yield ∞ as its total reward. Thus, in that setting all derivations that yield the same tree yield the same total reward, regardless of the derivation law.

If we instead adopt the simultaneous derivation law, where we expand all remaining nonterminals in each step of the derivation, then no pathologies arise as a result of 0 rewards on rules, and all of our results hold. In particular, the least fixed point solution of the corresponding max/min-linear equations, directly analogous to those described in Section 2 for 1-RMDPs and 1-RSSGs, characterizes the values of such a game starting at each nonterminal. The simultaneous derivation law corresponds naturally to the setting of Branching Markov Decision Processes (BMDPs) and BSSGs (see [17, 13]), and thus as already explained at the beginning of this section, the analogues of Theorems 1, 6, 8, and 18 hold also for BMDPs and BSSGs with non-negative rewards (including 0 rewards).

Acknowledgement. Research partly supported by NSF grants CCF-1017955, CCF-1703925, CCF-1763970, and EPSRC grant EP/G050112/2.

References.

References

[1] V. Blondel and V. Canterini. Undecidable problems for probabilistic automata of fixed dimension. *Theory of Computing Systems*, 36:231–245, 2003.

[2] T. Brázdil, V. Brozek, V. Forejt, and A. Kucera. Reachability in recursive Markov decision processes. *Information and Computation*, 206(5), pages 520–537, 2008.

[3] T. Brázdil, V. Brozek, A. Kucera, and J. Obdrzálek. Qualitative reachability in stochastic BPA games. *Inf. and Comp.*, 209(8), pages 1160–1183, 2011.

[4] T. Brázdil, S. Kiefer, A. Kucera, and I. Hutarova Varekova. Runtime analysis of probabilistic programs with unbounded recursion. *J. Comp. Sys. Sc*, 81(1), pp. 288-310, 2015.

[5] T. Brázdil, A. Kucera, P. Novotný and D. Wojtczak. Minimizing Expected Termination Time in One-Counter Markov Decision Processes. *Proc. of 39th ICALP'12*, 2012.

[6] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. In *Journal of ACM*, 56(3), 2009.

[7] A. Condon. The complexity of stochastic games. *Inf. & Comp.*, 96(2):203–224, 1992.

[8] A. Condon and M. Melekopoglou. On the complexity of the policy iteration algorithm for stochastic games. *ORSA Journal on Computing*, 6(2), 1994.

[9] C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a Nash equilibrium. In *SIAM J. on Computing*, 39(1), pp. 195-259, 2009.

[10] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic models of Proteins and Nucleic Acids*. Cambridge U. Press, 1999.

[11] J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. In *Logical Methods in Computer Science*, 2(1), pp. 1-31, 2006.

[12] J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: expectations and variances. In *Proc. of 20th IEEE LICS'05*, 2005.

[13] K. Etessami, A. Stewart, and M. Yannakakis. Polynomial-time algorithms for branching Markov decision processes, and probabilistic min(max) polynomial Bellman equations. In *Proc. 39th Int. Coll. on Automata, Languages and Programming (ICALP)*, 2012. (Fuller preprint on ArXiv:1202.4789).

[14] K. Etessami, A. Stewart, and M. Yannakakis. Greatest fixed points of probabilistic min/max polynomial equations, and reachability for branching Markov decision processes. In *Proc. 42nd Int. Coll. on Automata, Languages and Programming (ICALP)*, 2015. (Full preprint on ArXiv:1502.05533).

[15] K. Etessami, D. Wojtczak, and M. Yannakakis. Recursive stochastic games with positive rewards. In *Proceedings of ICALP'08*, volume 5125 of *LNCS*, pages 711–723. Springer, 2008.

[16] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. *Journal of ACM*, 56(1), 2009.

[17] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. *Journal of the ACM*, 62 (2), 69 pages, 2015.

[18] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. In *SIAM J. on Computing*, 39(6), pp. 2531-2597, 2010.

[19] R. Fagin, A. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons" (extended abstract). In *ACM Symp. on Theory of Computing*, pages 484–493, 2000.

[20] J. Fearnley. Exponential lower bounds for policy iteration. *Proc. Int. Coll. on Automata, Languages and Programming (ICALP)*, 551-562, 2010.

[21] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.

[22] O. Friedmann. An exponential lower bound for the latest deterministic strategy iteration algorithms. *Logical Methods in Computer Science*, 7(3), 2011.

[23] T. M. Gawlitza and H. Seidl. Solving systems of rational equations through strategy iteration. *ACM Trans. Program. Lang. Syst.*, 33(3):11, 2011.

[24] P. Haccou, P. Jagers, and V. A. Vatutin. *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambridge U. Press, 2005.

[25] T. E. Harris. *The Theory of Branching Processes*. Springer-Verlag, 1963.

[26] A. J. Hoffman and R. M. Karp. On nonterminating stochastic games. *Management Sci.*, 12:359–370, 1966.

[27] D. S. Johnson, C. Papadimitriou, and M. Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988.

[28] B. Juba. On the hardness of simple stochastic games. Master's thesis, CMU, 2006.

[29] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[30] D. A. Martin. Determinacy of Blackwell games. *J. Symb. Logic*, 63(4):1565–1581, 1998.

[31] A. Neyman and S. Sorin, editors. *Stochastic Games and Applications*. NATO ASI Series, Kluwer, 2003.

[32] S. Pliska. Optimization of multitype branching processes. *Management Sci.*, 23(2):117–124, 1976/77.

[33] M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[34] U. Rothblum and P. Whittle. Growth optimality for branching Markov decision chains. *Math. Oper. Res.*, 7(4):582–601, 1982.

[35] A. Trivedi, and D. Wojtczak. Timed branching processes. In *Proc. of 7th International Conference on Quantitative Evaluation of Systems (QEST)*, pages 219–228, 2010.

[36] A. F. Veinott. Discrete dynamic programming with sensitive discount optimality criteria. *Ann. Math. Statist.*, 40:1635–1660, 1969.

[37] Dominik Wojtczak. *Recursive Probabilistic Models: efficient analysis and implementation*. PhD thesis, School of Informatics, University of Edinburgh, 2009.

[38] Dominik Wojtczak. Expected termination time in BPA games. In *Proc. of International Symposium on Automated Technology for Verification and Analysis*, pages 303–318), 2013.

[39] D. Wojtczak and K. Etessami. PReMo: an analyzer for probabilistic recursive models. In *Proc. 13th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 66–71, 2007.