# Recursive Stochastic Games with Positive Rewards

K. Etessami[1], D. Wojtczak[1], and M. Yannakakis[2]

[1] LFCS, School of Informatics, University of Edinburgh
[2] Dept. of Computer Science, 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 Simple Stochastic Games (1-RSSGs) with strictly positive rewards. These are a class of finitely presented countable-state zero-sum stochastic games, with total expected reward objective. They subsume standard finite-state MDPs and Condon's simple stochastic games and correspond to optimization and game versions of several classic stochastic models, with rewards. 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 give polynomial time algorithms for 1-exit Recursive Markov decision processes (1-RMDPs) with positive rewards. Specifically, we show that the exact optimal value of both maximizing and minimizing 1-RMDPs with positive rewards can be computed in polynomial time (this value may be $\infty$). For two-player 1-RSSGs with positive rewards, we prove a "stackless and memoryless" determinacy result, and show that deciding whether the game value is at least a given value $r$ is in NP $\cap$ coNP. We also prove that a simultaneous strategy improvement algorithm converges to the value and optimal strategies for these stochastic games. We observe that 1-RSSG positive reward games are "harder" than finite-state SSGs in several senses.

## 1 Introduction

Markov decision processes and stochastic games are fundamental models in stochastic optimization and game theory (see, e.g., [25, 23, 13]). 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.) 1-RMDPs and 1-RSSGs were studied in [8, 9] 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 $\infty$.

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 [21], biological sequence analysis ([4]), and population biology [17, 16]. Another model that corresponds to a strict subclass of SCFGs is "random walks with back-buttons" studied in [12] as a model of web surfing. See [7] 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 (see full version [11] for details). We are given a context-free grammar where non-terminals are partitioned into three disjoint sets: `random`, `player-1`, and `player-2`. Starting from a designated start non-terminal, $S_{\text{init}}$, we proceed to generate a derivation by choosing a remaining *left-most* non-terminal, $S$, and expanding it. The precise derivation law (left-most, right-most, etc.) doesn't effect the game value in our strictly positive reward setting, but does if we allow 0 rewards. If $S$ belongs to `random`, it is expanded randomly 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 total expected reward (which may be $\infty$), and player 2 wants to minimize it. When we have only one player it is a minimizing or maximizing 1-RMDP.

We assume strictly positive rewards on all transitions (rules) in this paper. This assumption is very natural for modeling 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 en-

dow our games with a number of important robustness properties. In particular, in the above 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 $\infty$) depending on the derivation law (e.g., left-most or right-most). Moreover, for 1-RMDPs, if we allow 0 rewards, then there may not even exist any $\epsilon$-optimal strategies. Furthermore, even in a purely probabilistic setting without players (1-RMCs), with 0 rewards the expected reward can be irrational. 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 open decidability questions, e.g., for optimal reachability probabilities in non-reward 1-RMDPs ([8, 1]). (See the full version [11] for elaboration on these issues.) As we shall show, none of these pathologies arise in our setting with strictly positive rewards.

We show that 1-RMDPs and 1-RSSGs with strictly positive rewards have a value which is either rational (with polynomial bit complexity) or $\infty$, 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 the maximizing and minimizing 1-RMDPs. The two cases 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 $\infty$) 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 ([19]). Whether this strategy improvement algorithm runs in worst-case P-time is open, just like its version for finite-state SSGs.

We observe that these games are essentially "harder" than Condon's finite-state SSG games 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 $\infty$. By contrast, if finite-state SSGs are themselves equipped with strictly positive rewards, we can decide in P-time whether their value is $\infty$. Moreover, it has recently been shown that computing the value of Condon's SSG games is in the complexity class PPAD (see [10] and [20]). The same proof however does not work for 1-RSSG positive reward games, 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 $\infty$. 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 $\infty$.

The tool PReMo [28] implements a number of analyses for RMCs, 1-RMDPs, and 1-RSSGs. Most recently, the strategy improvement algorithm of this paper was implemented and incorporated in the tool. See the PReMo web page ([28]) for very encouraging experimental results based on the algorithms of this paper.

*Due to space constraints proofs and discussions are omitted. See full paper [11].*

**Related work.** Two (equivalent) purely probabilistic recursive models, Recursive Markov chains and probabilistic Pushdown Systems (pPDSs) were introduced in [7] and [5], and have been studied in several papers recently. These models were extended to the optimization and game setting of (1)-RMDPs and (1)-RSSGs in [8, 9], and studied further in [1]. As mentioned earlier, the games considered in these earlier papers had the goal of maximizing/minimizing termination or reachability probability, which can be irrational, and for which quantitative decision problems 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 [9]. 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 $\infty$. But the converse does not hold: the expected reward may be $\infty$ 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 (compute the optimal reward), whereas in [9] it was purely qualitative (almost sure termination).

Condon [2] originally studied finite-state SSGs with termination objectives (no rewards), and showed that the quantitative termination decision problem is in NP $\cap$ coNP; it is a well-known open problem whether it is in P. In [3] strategy improvement algorithms for SSGs were studied, based on variants of the classic Hoffman-Karp algorithm [18]. It remains open whether the simultaneous version of strategy improvement runs in P-time. This is also the case for our simultaneous strategy improvement algorithm for 1-RSSGs with positive rewards. (Single-vertex updates per step in strategy improvement is known to require exponentially many steps in the worst case.)

There has been some recent work on augmenting purely probabilistic multi-exit RMCs and pPDSs with rewards in [6]. These results however are for RMCs without players. We in fact show in Theorem 8 that the basic questions about multi-exit RMDPs and RSSGs are undecidable.

A full tech report of this paper appeared in [11]. A recent and independent paper by Gawlitza and Seidl [14] considers monotone linear-min-max equations with potentially negative constant terms (with entirely different motivation 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 rather 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 [15] containment in NP∩coNP, nor in PLS, for the relevant decision and search problems. Nevertheless, there are connections between their work and ours that need to be explored further. In particular, Gawlitza [15] informs us that a modified version of their strategy improvement algorithm can also be used to obtain our P-time upper bound for the LFP, over the non-negative extended reals, for the linear-min and linear-max equations that arise for 1-RMDPs.

Models related to 1-RMDPs have been studied in OR, under the name Branching Markov Decision Chains (a controlled version of multi-type Branching processes). These are close to the single-player SCFG model, with non-negative rewards, but simultaneous derivation law. They were studied by Pliska [24], in a related form by Veinott [27], and extensively by Rothblum and co-authors (e.g., [26]). 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.

## 2 Definitions and Background

Let $\mathbb{R}_{>0} = (0, \infty)$ denote the positive real numbers, $\mathbb{R}_{\geq 0} \doteq [0, \infty)$, $\overline{\mathbb{R}} \doteq [-\infty, \infty]$, $\mathbb{R}^{\infty}_{>0} \doteq (0, \infty]$, and $\mathbb{R}^{\infty}_{\geq 0} \doteq [0, \infty]$. 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 *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 $\mathtt{pl}_i(ex) = 0$ for all $ex \in Ex_i$.
– 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. We assume for vertices $u$ and $v$ there is at most one transition in $\delta$ 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 $u \in \mathtt{pl}_i^{-1}(0)$, $\sum_{\{v' | (u, p_{u,v'}, v', c_{u,v}) \in \delta_i\}} p_{u,v'} = 1$, 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.)

We use the symbols ($N, B, Q, \delta$, 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 node 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 $\Delta$ are defined inductively as follows:

1. $\langle \epsilon, u \rangle \in V$, for $u \in Q$. ($\epsilon$ 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) ([7]); A *1-RSSGs* is a RSSG where every component has one exit, and we likewise define *1-RMDPs* and *1-RMCs*. This entire paper is focused on 1-RSSGs and 1-RMDPs, except for Theorem 8, 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* $\sigma$ 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.) Let $\Psi_i$ denote the set of all strategies for player $i$. A pair of strategies $\sigma \in \Psi_1$ and $\tau \in \Psi_2$ induce 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 total expected reward gained during the $i$'th transition, starting at $\langle \epsilon, u \rangle$ to be given by a random variable $Y_i$. The total $k$-step expected reward is simply $r_u^{k,\sigma,\tau} = E[\sum_{i=1}^{k} Y_i]$. When $k = 0$, we of course have $r_u^{0,\sigma,\tau} = 0$. Given an initial vertex $u$, let $r_u^{*,\sigma,\tau} = \lim_{k\to\infty} r^{k,\sigma,\tau} = E[\sum_{i=1}^{\infty} Y_i] \in [0, \infty]$ 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 ([22]) 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. Nevertheless, we will establish that determinacy does hold for 1-RSSG positive reward games, as part of our proof of Stackless & Memoryless determinacy. For all vertices $u$, let $r_u^* \doteq \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} r_u^{*,\sigma,\tau}$. We show $r_u^* = \inf_{\tau\in\Psi_2} \sup_{\sigma\in\Psi_1} r_u^{*,\sigma,\tau}$, and thus $r_u^*$ is the *value* of the game starting at vertex $u$. 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 strategies for both players.* For a strategy $\sigma \in \Psi_1$, let $r_u^{*,\sigma} = \inf_{\tau\in\Psi_2} r_u^{*,\sigma,\tau}$, and for $\tau \in \Psi_2$, let $r_u^{*,\cdot,\tau} = \sup_{\sigma\in\Psi_1} r_u^{*,\sigma,\tau}$. Call a deterministic strategy *Stackless & Memoryless (SM)* if it depends neither on the history of the game nor on the current call stack, i.e., only depends on the current vertex. Such strategies, for player $i$, can be given by a map $\sigma : V_i \mapsto V$. We call a game *SM-determined* if both players have optimal SM strategies.

In ([8]) 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 $u$, $r_u^* \doteq \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} r_u^{*,\sigma,\tau} = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} r_u^{*,\sigma,\tau}$ (i.e., these games are determined). (6) $\mathbf{r}^* = \lim_{k \to \infty} \mathbf{x}^k$.*

The following is a simple corollary of the proof.

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

## 3 SM-determinacy and strategy improvement

We now prove SM-determinacy, and 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, $\sigma$, 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 3 that this can be done in P-time.) Then, update $\sigma$ to a new SM strategy, $\sigma'$, 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 $\sigma'$ in place of $\sigma$, until no further local improvement is possible, i.e., stop when $\sigma' = \sigma$.
Theorem 2 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 1, both players have optimal SM strategies, i.e., the games are SM-determined.

**Theorem 2.** *(1)* SM-determinacy. *In 1-RSSG positive reward games, both players have optimal SM strategies. (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.*

The proof is intricate, and is given in the full version ([11]). Here we briefly sketch the approach. Fix a SM strategy $\sigma$ for player 1. It can be shown that if $\mathbf{x} = P(\mathbf{x})$ is the linear-min-max equation system for this 1-RSSG, then $\mathbf{r}_u^{*,\sigma} \leq P_u(\mathbf{r}^{*,\sigma})$, for all vertices $u$, and equality fails only on vertices $u_i$ belonging to player 1 such that $\sigma(u_i) = v_i$ is not "locally optimal", i.e., such that there exists some neighbor $w_i$ such that $r_{w_i}^{*,\sigma} + c_{u_i,w_i} > r_{v_i}^{*,\sigma} + c_{u_i,v_i}$. Let $u_1, \ldots, u_n$ be all such

vertices belonging to player 1. Associate a parameter $t_i \in \mathbb{R}^{\infty}_{\geq 0}$ with each such vertex $u_i$, creating a parametrized game $A(\mathbf{t})$, in which whenever the vertex $u_i$ is encountered player 1 gains additional reward $t_i$ and the game then terminates. Let $g_{u,\tau}(\mathbf{t})$ denote the expected reward of this parametrized game starting at vertex $u$, when player 1 uses SM strategy $\sigma$ and player 2 uses SM strategy $\tau$. Let $f_u(\mathbf{t}) = \min_\tau g_{u,\tau}(\mathbf{t})$. The vector $\mathbf{t}^\sigma$, where $t_i^\sigma = r_{u_i}^{*,\sigma}$, is a fixed point of $f_u(\mathbf{t})$, for every vertex $u$, and so is $\mathbf{t}^{\sigma'}$ where $\sigma'$ is any SM strategy consistent with $\sigma$ on all vertices other than the $u_i$'s. The functions $g_{u,\tau}(\mathbf{t})$ is continuous and nondecreasing over $[0,\infty]^n$, and expressible as an infinite sum of *linear* terms with non-negative coefficients. Using these properties of $g_{u,\tau}$, and their implications for $f_u$, we show that if $\sigma'$ is the SM strategy obtained by locally improving the strategy $\sigma$ at the $u_i$'s, by letting $\sigma'(u_i) := w_i$, then $t_i^\sigma = \mathbf{r}_{u_i}^{*,\sigma} < \mathbf{r}_{u_i}^{*,\sigma'} = \mathbf{t}_i^{\sigma'}$, and thus also $\mathbf{r}_z^{*,\sigma} = f_z(\mathbf{t}^\sigma) \leq f_z(\mathbf{t}^{\sigma'}) = \mathbf{r}_z^{*,\sigma'}$, for any vertex $z$. Thus, switching to $\sigma'$ does not decrease the value at any vertex, and increases it on all the switched vertices $u_i$. There are only finitely many SM strategies, thus after finitely many iterations we reach a SM strategy, $\sigma$, where no improvement is possible. This $\sigma$ must be optimal. Since each local improvement step can be done in P-time and increases sum total reward, the problem is in PLS. □

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

**Theorem 3.** *There is a P-time algorithm for computing the exact optimal value (including the possible value $\infty$) 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.

### 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 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 indexed by any $v \notin U$ removed.

**Proposition 1.** *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 $\mathbf{r}' = \mathbf{r}^*|_U$.*

**Theorem 4.** *We can compute $\mathbf{r}^*$ for the max-linear equation system $P$, including the values that are infinite, in time polynomial in the size of the 1-RMDP.*

*Proof.* Build 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 1 (IV), that at least one of the values of $\mathbf{r}^*|_{S_i}$ is $\infty$. So by Proposition 1 (II), all are. We can then mark all variables in $S_i$ as $\infty$, and also mark all variables in the SCCs that can reach $S_i$ in $SCC(G)$ as $\infty$. 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 $\infty$. We label the variables not in $U$ with $\infty$, obtaining the vector $\mathbf{r}^*$. □

## 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 1.** *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).*

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 find and remove all variables $x_u$ such that $r^*_u = \infty$.* By removing these 0 and $\infty$ variables from $P$ we obtain a new system $P'$, and a new LP, min-LP$'$.

**Lemma 2.** *If $\infty$ 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 1, player 2 has an optimal SM strategy, call it $\tau$, which yields the finite optimal reward vector $r^*$. Once strategy $\tau$ is fixed, we can define a new equation system $P'_\tau(\mathbf{x}) = A_\tau \mathbf{x} + b_\tau$, where $A_\tau$ is a nonnegative matrix and $b_\tau$ is a vector of average rewards per single step from each node, obtained under strategy $\tau$. 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 2.** *(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).*

Now pick an optimal *SM* strategy $\tau$ 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 5.** *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 2, we can compute the exact value (even if $\infty$) of minimizing reward 1-RMDPs in P-time.*

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

**Theorem 6.** *Deciding whether the value $r^*_u$ of a 1-RSSG positive reward game is $\geq a$ for a given $a \in [0, \infty]$, is in NP $\cap$ coNP.*

This is immediate from P-time upper bounds for 1-RMDPs, and SM-determinacy: guess a player's SM strategy, and compute the value for the remaining 1-RMDP.

**Theorem 7.** *Condon's quantitative termination problem for finite SSGs reduces in P-time to the problem of deciding whether $r^*_u = \infty$.*

By contrast, for finite-state SSGs with strictly positive rewards, we can decide in P-time whether the value is $\infty$, because this is the case iff the value of the corresponding termination game is not 1. Deciding whether an SSG termination game has value 1 is in P-time (see, e.g., [9]).

Finally, we show undecidability for multi-exit RMDPs and RSSGs.

**Theorem 8.** *For multi-exit positive reward RMDPs it is undecidable to distinguish whether the optimal expected reward for a node is finite or $\infty$.*

**Acknowledgement.** Research partly supported by NSF grant CCF-0728736.

## References


1. T. Brázdil, V. Brozek, V. Forejt, and A. Kucera. Reachability in recursive markov decision processes. In *Proc. 17th Int. CONCUR*, pages 358–374, 2006.
2. A. Condon. The complexity of stochastic games. *Inf.&Comp.*, 96:203–224, 1992.
3. A. Condon and M. Melekopoglou. On the complexity of the policy iteration algorithm for stochastic games. *ORSA Journal on Computing*, 6(2), 1994.
4. R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic models of Proteins and Nucleic Acids*. Cambridge U. Press, 1999.
5. J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. In *LICS*, pages 12–21, 2004.
6. J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: expectations and variances. In *Proc. of 20th IEEE LICS'05*, 2005.
7. K. Etessami and M. Yannakakis. Recursive markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proc. of 22nd STACS*, 2005. (See full version at: `http://homepages.inf.ed.ac.uk/kousha/bib_index.html`).
8. K. Etessami and M. Yannakakis. Recursive markov decision processes and recursive stochastic games. In *Proc. 32nd ICALP*, 2005.
9. K. Etessami and M. Yannakakis. Efficient qualitative analysis of classes of recursive markov decision processes and simple stochastic games. In *Proc. 23rd STACS*, 2006.
10. K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. In *Proc. of 48th IEEE FOCS*, 2007.
11. K. Etessami, D. Wojtczak, and M. Yannakakis. Recursive stochastic games with positive rewards. Tech report EDI-INF-RR-1224, July, 2007.
12. R. Fagin, A. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons". In *STOC*, 2000.
13. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.
14. T. Gawlitza and H. Seidl. Precise relational invariants through strategy iteration. In *Proc. of 16th CSL*, 2007.
15. T. Gawlitza. Personal communication. April, 2008.
16. P. Haccou, P. Jagers, and V. A. Vatutin. *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambridge U. Press, 2005.
17. T. E. Harris. *The Theory of Branching Processes*. Springer-Verlag, 1963.
18. A. Hoffman and R. Karp. On nonterminating stochastic games. *Manag. Sci.*, 12:359–370, 1966.
19. D. S. Johnson, C. Papadimitriou, and M. Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988.
20. B. Juba. On the hardness of simple stochastic games. Master's thesis, CMU, 2006.
21. C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
22. D. A. Martin. Determinacy of Blackwell games. *J. Sym. Log.*, 63:1565–1581, 1998.
23. A. Neyman and S. Sorin, ed. *Stochastic Games and Applications*. Kluwer, 2003.
24. S. Pliska. Optimization of multitype branching processes. *Management Sci.*, 23:117–124, 1976/77.
25. M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.
26. U. Rothblum and P. Whittle. Growth optimality for branching Markov decision chains. *Math. Oper. Res.*, 7(4):582–601, 1982.
27. A. F. Veinott. Discrete dynamic programming with sensitive discount optimality criteria. *Ann. Math. Statist.*, 40:1635–1660, 1969.
28. D. Wojtczak and K. Etessami. Premo: an analyzer for probabilistic recursive models. In *Proc. of TACAS*, 2007. Tool web page: `http://groups.inf.ed.ac.uk/premo/`.