Reachability for Branching Concurrent Stochastic Games

Kousha Etessami
School of Informatics, University of Edinburgh, UK
kousha@inf.ed.ac.uk

Emanuel Martinov
School of Informatics, University of Edinburgh, UK
eo.martinov@gmail.com

Alistair Stewart
Department of Computer Science, University of Southern California, Los Angeles, CA, USA
stewart.al@gmail.com

Mihalis Yannakakis
Department of Computer Science, Columbia University, New York City, NY, USA
mihalis@cs.columbia.edu

Abstract

We give polynomial time algorithms for deciding almost-sure and limit-sure reachability in Branching Concurrent Stochastic Games (BCSGs). These are a class of infinite-state imperfect-information stochastic games that generalize both finite-state concurrent stochastic reachability games ([8]) and branching simple stochastic reachability games ([13]).

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases stochastic games, multi-type branching processes, concurrent games, minimax-polynomial equations, reachability, almost-sure, limit-sure

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.115

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version A full preprint is available on arXiv at `https://arxiv.org/abs/1806.03907`.

Funding Research partially supported by an EPSRC studentship EP/N509644/1-1789473 (Martinov), and by NSF Grants CCF-1703925,CCF-1763970 (Yannakakis).

1 Introduction

Branching Processes (BP) are infinite-state stochastic processes that model the stochastic evolution of a population of entities of distinct types. In each generation, every entity of each type t produces, as offsprings, a set of entities of various types in the next generation, according to a given probability distribution on offsprings associated with type t. BPs are fundamental stochastic models that have been used to model phenomena in many fields, including biology (see, e.g., [24]), population genetics ([20]), physics and chemistry (e.g., particle systems, chemical chain reactions), medicine (e.g. cancer growth [2, 28]), marketing, and others. In many cases, the process is not purely stochastic but there is the possibility of taking actions (for example, adjusting the conditions of reactions, applying drug treatments in medicine, advertising in marketing, etc.) which can influence the probabilistic evolution of the process to bias it towards achieving desirable objectives. Some of the factors that affect the reproduction may be controllable (to some extent) while others are not and also may not be sufficiently well-understood to be modeled accurately by specific probability distributions, and thus it may be more appropriate to consider their effect in an adversarial (worst-case)

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 115; pp. 115:1–115:14


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

sense. *Branching Concurrent Stochastic Games* (BCSG) are a natural model to represent such settings. There are two players and, for each type t, each player has a set of available actions, which can affect the reproduction of entities of type t. The evolution of the process starts with some initial population of entities. For each entity of type t, the two players each select (concurrently and independently) an action from their available set (possibly in a randomized manner) and their join choices of actions determine the probability distribution for the offsprings of that entity. The first player represents a controller who can control some of the parameters of the reproduction and the second player represents other parameters that are not controlled and are treated adversarially. The first player wants to select a strategy that optimizes some objective. In this paper we focus on *reachability objectives*, a basic and natural class of objectives. Some types are designated as undesirable (for example, malignant cells), in which case we want to minimize the probability of ever reaching any entity of such a type. Or conversely, some types may be designated as desirable, in which case we want to maximize the probability of reaching an entity of such a type.

BCSGs generalize purely stochastic multi-type BPs as well as Branching Markov Decision Processes (BMDP) and Branching Simple Stochastic Games (BSSG) which were studied for reachability objectives in [13]. In BMDPs there is only one player who aims to maximize or minimize a reachability objective. In BSSGs there are two opposing players, but they control different types. These models were studied previously also under another basic objective, namely optimization of *extinction probability*, i.e., the probability that the process will eventually become extinct, that is, that the population will become empty [11, 15]. We will later discuss in detail the prior results and compare them with the results in this paper.

BCSGs can also be seen as a generalization of finite-state concurrent (stochastic) games [8] (see also [18]); namely they extend such finite-state games with branching. Concurrent games have been used in the verification area to model the dynamics of open systems, where one player represents the system and the other player the environment. Such a system moves sequentially from state to state depending on the actions of the two players (the system and the environment). Branching concurrent games model the more general setting in which processes can spawn new processes that then proceed independently in parallel (e.g., new threads are created and terminated). We note incidentally that even if there are no probabilities in the system itself, in the case of concurrent games, probabilities arise naturally from the fact that the optimal strategies are in general randomized; as a consequence it can be shown that branching concurrent stochastic games are expressively and computationally equivalent to the non-stochastic version (see [15]).

We now summarize our main results and compare and contrast them with previous results on related models. First, we show that a Branching concurrent stochastic game (BCSG), G, with a reachability objective has a well-defined *value*, i.e., given an initial (finite) population μ of entities of various types and a target type t^*, if the sets of all possible (mixed) strategies of the two players are respectively Ψ_1, Ψ_2, and if $\Upsilon_{\sigma,\tau}(\mu, t^*)$ denotes the probability of eventually reaching an entity of type t^* when starting from initial population μ under strategy $\sigma \in \Psi_1$ for player 1 and strategy $\tau \in \Psi_2$ for player 2, then $\inf_{\sigma\in\Psi_1} \sup_{\tau\in\Psi_2} \Upsilon_{\sigma,\tau}(\mu, t^*)$ $= \sup_{\tau\in\Psi_2} \inf_{\sigma\in\Psi_1} \Upsilon_{\sigma,\tau}(\mu, t^*)$, which is the *value*, v^*, of the game. Furthermore, we show that the player who wants to minimize the reachability probability always has an optimal (mixed) *static strategy* that achieves the value, i.e., a strategy σ^* which uses, for all entities of each type t generated over the entire history of the game, the same probability distribution on the available actions for type t, independent of the past history, and which has the property that $v^* = \sup_{\tau\in\Psi_2} \Upsilon_{\sigma^*,\tau}(\mu, t^*)$. The optimal strategy in general has to be mixed (randomized); this is the case even for finite-state concurrent games [8]. On the other hand,

the player that wants to maximize the reachability probability of a BCSG need not have any optimal strategy (whether static or not), and it was known that this holds even for BMDPs, i.e., even when there is only one player [13]. The same holds for finite-state CSGs: the player maximizing reachability probability need not have any optimal strategy [8].

To analyze BCSGs reachability games, we model them by a system of equations $x = P(x)$, called a *minimax Probabilistic Polynomial System* (minimax-PPS for short), where x is a tuple of variables corresponding to the types of the BCSG. There is one equation $x_i = P_i(x)$ for each type t_i, where $P_i(x)$ is the value of a (one-shot) two-player zero-sum matrix game, whose payoff for every pair of actions is given by a polynomial in x whose coefficients are positive and sum to at most 1 (a probabilistic polynomial). The function $P(x)$ defines a monotone operator from $[0,1]^n$ to itself, and thus it has, in particular, a *greatest fixed point* (GFP) g^* in $[0,1]^n$. We show that the coordinates g_i^* of the GFP give the optimal *non-reachability* probabilities for the BCSG game when started with a population that consists of a single entity of type t_i. The value of the game for any initial population μ can be derived easily from the GFP g^* of the minimax-PPS. This generalizes a result in [13], which established an analogous result for the special case of BSSGs. It also follows from our minimax-PPS equational characterization that *quantitative* decision problems for BCSGs, such as deciding whether the reachability game value is $\geq p$ for a given $p \in (0,1)$ are all solvable in PSPACE.

Our main algorithmic results concern the *qualitative* analysis of the reachability problem, that is, the problem of determining whether one of the players can win the game with probability 1, i.e., if the value of the game is 0 or 1. We provide the first polynomial-time algorithms for qualitative reachability analysis for branching concurrent stochastic games. For the value=0 problem, the algorithm and its analysis are very simple. If the value is 0, the algorithm computes an optimal strategy σ^* for the player that wants to minimize the reachability probability, where σ^* is in fact static and deterministic, i.e., it selects for each type, deterministically, a single available action, and guarantees $\Upsilon_{\sigma^*,\tau}(\mu, t^*) = 0$ for all $\tau \in \Psi_2$. If the value is positive then the algorithm computes a static mixed strategy τ for the player maximizing reachability probability that guarantees $\inf_{\sigma\in\Psi_1} \Upsilon_{\sigma,\tau}(\mu, t^*) > 0$.

The value=1 problem is much more complicated. There are two versions of it, because it is possible that the game value is 1 but no strategy for the maximizing player guarantees reachability with probability 1. Thus, we have two versions of the problem. In the first version, called the *almost-sure problem*, we want to decide whether there exists a strategy τ^* for player 2 that guarantees that the target type t^* is reached with probability 1 regardless of the strategy of player 1, i.e., such that $\Upsilon_{\sigma,\tau^*}(\mu, t^*) = 1$ for all $\sigma \in \Psi_1$. In the second version, called the *limit-sure problem*, we want to decide if the value $v^* = \sup_{\tau\in\Psi_2} \inf_{\sigma\in\Psi_1} \Upsilon_{\sigma,\tau}(\mu, t^*)$ is 1, i.e., if for every $\epsilon > 0$ there is a strategy τ_ϵ of player 2 that guarantees that the probability of reaching the target type is at least $1 - \epsilon$ regardless of the strategy σ of player 1; such a strategy τ_ϵ is called *ϵ-optimal*. We provide polynomial-time algorithms for both versions of the problem. The algorithms non-trivially generalize the algorithms of both [8] and [13], both of which address different special subcases of qualitative BCSG reachability. Our positive results on qualitative reachability for BCSG are surprising especially in view of the known major obstacles in resolving the analogous questions for the extinction problem for BCSG, as explained below in the review of related work.

In the almost-sure problem, if the answer is positive, our algorithm constructs (a compact description of) a strategy τ^* of player 2 that achieves value 1; the strategy is a randomized non-static strategy, and this is inherent (i.e., there may not exist a static strategy that achieves value 1). If the answer is negative, our algorithm constructs a (non-static, randomized)

strategy σ for the opposing player 1 such that $\Upsilon_{\sigma,\tau}(\mu, t^*) < 1$ for all strategies τ of player 2. In the limit-sure problem, if the answer is positive, i.e., the value is 1, our algorithm constructs for any given $\epsilon > 0$, a static, randomized ϵ-optimal strategy, i.e., a strategy τ_ϵ such that $\Upsilon_{\sigma,\tau_\epsilon}(\mu, t^*) \geq 1 - \epsilon$ for all $\sigma \in \Psi_1$. If the answer is negative, i.e., the value is < 1, our algorithm constructs a static randomized strategy σ' for player 1 such that $\sup_{\tau \in \Psi_2} \Upsilon_{\sigma',\tau} < 1$. Due to space limits, most proofs are omitted; see the full version.

Related Work. As mentioned, the two works most closely related to ours are [8] and [13]. Our results generalize both. de Alfaro, Henzinger, and Kupferman [8] studied finite-state concurrent (stochastic) games (CSGs) with reachability objectives and provided polynomial time algorithms for their qualitative analysis, both for the almost-sure and the limit-sure reachability problem. See also [22, 19, 21, 5] for more recent results on finite-state CSG reachability; in particular, there are finite-state CSGs with value= 1 for which (near-)optimal strategies need to have some action probabilities that are doubly-exponentially small [22, 5] (unlike simple, turned-based stochastic games), and thus must be represented succinctly to ensure polynomial space. This is of course the case also for branching CSGs, and the optimal or ϵ-optimal strategies constructed by our algorithms are represented compactly so that the algorithms run in polynomial time.

BMDPs and BSSGs with reachability objectives were studied in [13], which provided polynomial-time algorithms for their qualitative analysis, and also gave polynomial time algorithms for approximate quantitative analysis of BMDPs, i.e., approximate computation of the optimal reachability probability for maximizing and minimizing BMDPs, and it showed that this problem for BSSGs is in TFNP. Note that even for finite-state simple stochastic games the question of whether the value of the game can be computed in polynomial time is a well-known long-standing open problem [7]. It was also shown in [13] that the optimal non-reachability probabilities of maximizing or minimizing BMDPs and BSSGs are captured by the GFP of a system of min/max-PPS equations, $x = P(x)$, where each right-hand side $P_i(x)$ is the maximum or minimum of a set of probabilistic polynomials in x; note that these types of equation systems are special cases of minimax-PPSs and are much simpler.

Another important objective, the probability of *extinction*, has been studied previously for Branching Concurrent Stochastic Games, as well as BMDPs and BSSGs, and the purely stochastic model of Branching Processes (BPs). These branching models under the extinction objective are equivalent to corresponding subclasses of recursive Markov models, called respectively, 1-exit Recursive Concurrent Stochastic Games (1-RCSG), Markov Decision Processes (1-RMDP), and Markov Chains (1-RMC), and related subclasses of probabilistic pushdown processes under a termination objective [16, 12, 17, 11, 15, 10]. The extinction probabilities for these models are captured by the *least fixed point* (LFP) solutions of similar systems of probabilistic polynomial equations; for example, the optimal extinction probabilities of a BCSG are given by the LFP of a minimax-PPS. Polynomial time-algorithms for qualitative analysis, as well as for the approximate computation of the optimal extinction probabilities of Branching MDPs (and 1-RMDPs) were given in [17, 11]. However, negative results were shown also which indicate that the problem is much harder for branching concurrent (or even simple) stochastic games, even for the qualitative extinction problem. Specifically, it was shown in [17] that the qualitative extinction (termination) problem for BSSG (equivalently, 1-RSSG) is at least as hard as the well-known open problem of computing the value of a finite-state simple stochastic game [7]. Furthermore, it was shown in [15] that (both the almost-sure and limit-sure) qualitative extinction problems for BCSGs (equivalently 1-RCSGs) are at least as hard as the square-root sum problem, which is not even known to be

in NP. Thus, *the qualitative reachability problem for BCSGs seems to be very different than the extinction problem for BCSGs*: obtaining analogous results for the qualitative extinction to those of the present paper for qualitative reachability would resolve two major open problems. This is despite the fact that there is a natural "duality" between the extinction and reachability objectives (see [13]).

The equivalence between branching models (like BMDPs, BCSGs) and recursive Markov models (like 1-RMDPs, 1-RCSGs) with respect to extinction does *not* hold for the reachability objective. For example, almost-sure and limit-sure reachability coincide for a BMDP, i.e., if the supremum probability of reaching the target is 1 then there exists a strategy that ensures reachability with probability 1. However, this is not the case for 1-RMDPs. Furthermore, almost-sure reachability for 1-RMDPs can be decided in polynomial time [3, 4], but limit-sure reachability for 1-RMDPs is not even known to be decidable. The qualitative reachability problem for 1-RMDPs and 1-RSSGs (and equivalent probabilistic pushdown models) was studied in [4, 1]. These results do not apply to the corresponding branching models (BMDP, BSSG). Another objective considered in prior work is the *expected total reward* objective for 1-RSSGs ([14]) and 1-RCSGs ([31]) with positive rewards. None of these prior results have any implications for BCSGs with reachability objectives.

For richer objectives beyond reachability or extinction, Chen et. al. [6] studied model checking of purely stochastic branching processes (BPs) with respect to properties expressed by deterministic parity tree automata, and showed that the qualitative problem is in P-time (hence this holds in particular for reachability probability in BPs), and that the quantitative problem of comparing the probability with a rational is in PSPACE. Michalevski and Mio [25] extended this to properties of BPs expressed by "game automata", a subclass of alternating parity tree automata. More recently, Przybyłko and Skrzypczak [27] considered existence and complexity of game values of Branching turn-based (i.e., simple) stochastic games, with regular objectives, where the two players aim to maximize/minimize the probability that the generated labeled tree belongs to a regular language (given by a tree automaton). They showed that (unlike our case of simpler reachability games) already for some basic regular properties these games are not even determined, meaning they do not have a value. They furthermore showed that for a probabilistic turn-based branching game, with a regular tree objective, it is undecidable to compare the value that a given player can force to 1/2; whereas for deterministic turn-based branching games they showed it is decidable and 2-EXPTIME-complete (respectively, EXPTIME-complete), to determine whether the player aiming to satisfy (respectively, falsify) a given regular tree objective has a pure winning strategy. Other past research includes work in operations research on (one-player) Branching MDPs [26, 29, 9]. None of these prior works bear on any of the results on BCSG reachability problems established in this paper.

2 Background

A *Branching Concurrent Stochastic Game* (BCSG) consists of a finite set $V = \{T_1, \ldots T_n\}$ of types, two finite non-empty sets $\Gamma^i_{max}, \Gamma^i_{min} \subseteq \Sigma$ of actions (one for each player) for each type T_i (Σ is a finite action alphabet), and a finite set $R(T_i, a_{max}, a_{min})$ of probabilistic rules associated with each tuple (T_i, a_{max}, a_{min}), where $i \in [n]$, $a_{max} \in \Gamma^i_{max}$, & $a_{min} \in \Gamma^i_{min}$. Each rule $r \in R(T_i, a_{max}, a_{min})$ is a triple (T_i, p_r, α_r), which we can denote by $T_i \xrightarrow{p_r} \alpha_r$, where $\alpha_r \in \mathbb{N}^n$ is a n-vector of natural numbers that denotes a finite multi-set over the set V, and where $p_r \in (0, 1] \cap \mathbb{Q}$ is the probability of the rule r (which we assume to

be a rational number, for computational purposes), where we assume that for all $T_i \in V$ and $a_{max} \in \Gamma^i_{max}$, $a_{min} \in \Gamma^i_{min}$, the rule probabilities in $R(T_i, a_{max}, a_{min})$ sum to 1, i.e., $\sum_{r \in R(T_i, a_{max}, a_{min})} p_r = 1$.

If for all types $T_i \in V$, either $|\Gamma^i_{max}| = 1$ or $|\Gamma^i_{min}| = 1$, then the model is a "turn-based" perfect-information game and is called a *Branching Simple Stochastic Game* (BSSG). If for all $T_i \in V$, $|\Gamma^i_{max}| = 1$ (respectively, $|\Gamma^i_{min}| = 1$), then it is called a *minimizing Branching Markov Decision Process* (BMDP) (respectively, a *maximizing* BMDP). If both $|\Gamma^i_{min}| = 1 = |\Gamma^i_{max}|$ for all $i \in [n]$, then the process is a classic, purely stochastic, *multi-type Branching Process* (BP) ([23]).

A *play* of a BCSG defines a (possibly infinite) node-labeled forest, whose nodes are labeled by the type of the object they represent. A play contains a sequence of "generations", $X_0, X_1, X_2, \ldots$ (one for each integer time $t \geq 0$, corresponding to nodes at depth/level t in the forest). For each $t \in \mathbb{N}$, X_t consists of the population (set of objects of given types), at time t. X_0 is the initial population at generation 0 (these are the roots of the forest). X_{k+1} is obtained from X_k in the following way: for each object e in the set X_k, assuming e has type T_i, both players select simultaneously and independently actions $a_{max} \in \Gamma^i_{max}$, and $a_{min} \in \Gamma^i_{min}$ (or distributions on such actions), according to their strategies; thereafter a rule $r \in R(T_i, a_{max}, a_{min})$ is chosen randomly and independently (for object e) with probability p_r; each such object e in X_k is then replaced by the set of objects specified by the multi-set α_r associated with the corresponding randomly chosen rule r. This process is repeated in each generation, as long as the current generation is not empty, and if for some $k \geq 0$, $X_k = \emptyset$ then we say the process *terminates* or becomes *extinct*.

The strategies of players can in general be arbitrary functions from any finite *history* tree, to (distributions on) actions, for each object in the current population. The *history* of the process up to time k is a forest of depth k that includes not only the populations $X_0, X_1, \ldots, X_k$, but also all the information regarding past actions and rules applied at each object, and all the parent-child relationships between objects up to generation k. The history can be represented by a forest of depth k, with internal nodes labeled by rules and actions, and whose leaves at level k form the current population X_k. Thus, formally, a strategy of player 1 (player 2, respectively) is a function that maps every finite history (i.e., a labelled forest of some finite depth, k, as above) and each object e in the current population X_k (leaf at depth k) to a probability distribution on the actions Γ^i_{max} (to the actions Γ^i_{min}, respectively), assuming that object e has type T_i.[1]

Let Ψ_1, Ψ_2 be the set of all strategies of players 1, 2. We say that a strategy is *deterministic* if for every history it maps each object e in the current population to a single action with probability 1 (in other words, it does not randomize on actions). We say that a strategy is *static* if for each type $T_i \in V$, and for any object e of type T_i, the player always chooses the same distribution on actions, irrespective of the history.

Different objectives can be considered for BCSGs. This paper considers (existential) *reachability*, where the aim of the players is to maximize/minimize the probability of reaching a generation that contains at least one object of a given target type T_{f^*}. The BCSG reachability game can of course also be viewed as a "non-reachability" ("safety") game, by just reversing the role of the players. We will exploit this alternative view in crucial ways (and this was also exploited in [13] for BSSGs). Given an initial population $\mu \in \mathbb{N}^n$,

[1] Note: this very general notion of a "strategy" permits the action (or distribution on actions) chosen for a given object e to depend not only on e's "ancestors" in the history forest, but also on siblings, cousins, etc., in the entire forest, up to and including the generation of the population that e belongs to.

with $\mu_{f^*} = 0$, and given strategies $\sigma \in \Psi_1, \tau \in \Psi_2$, let $g^*_{\sigma,\tau}(\mu)$ denote the probability that $(X_l)_{f^*} = 0$ for all $l \geq 0$. Let $g^*_{\sigma,*}(\mu) = \inf_{\tau\in\Psi_2} g^*_{\sigma,\tau}(\mu)$, let $g^*_{*,\tau}(\mu) = \sup_{\sigma\in\Psi_1} g^*_{\sigma,\tau}(\mu)$, and let $g^*(\mu) = \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} g^*_{\sigma,\tau}(\mu)$ denote the *value* of the game under the non-reachability objective and for the initial population μ. We will show that these games do have a *value*, meaning $g^*(\mu) = \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} g^*_{\sigma,\tau}(\mu) = \inf_{\tau\in\Psi_2} \sup_{\sigma\in\Psi_1} g^*_{\sigma,\tau}(\mu)$.

In the case where the initial population μ is a single object of some type T_i, then for the value of the game we write g^*_i, and similarly, when one or both of the strategies σ, τ are fixed, we write $(g^*_{\sigma,*})_i$, $(g^*_{*,\tau})_i$, or $(g^*_{\sigma,\tau})_i$. The vector g^* of g^*_i's, is called the vector of the non-reachability values of the game. We will see that, having the vector of g^*_i's, the non-reachability value for any starting population μ can be computed simply as $g^*(\mu) = \prod_i (g^*_i)^{\mu_i}$. So given a BCSG, the goal is to compute the vector g^* of non-reachability values. As our original objective is reachability, we point out that the vector of reachability values is $r^* = \mathbf{1} - g^*$ (where $\mathbf{1}$ is the all-1 vector), and hence the reachability value $r^*(\mu)$ of the game starting with population μ is $r^*(\mu) = 1 - g^*(\mu)$.

We will associate with any given BCSG a system of *minimax probabilistic polynomial equations* (minimax-PPS), $x = P(x)$, for the non-reachability objective. This system will have one variable x_i, and one equation $x_i = P_i(x)$, for each type T_i other than the target type T_{f^*}. We will show that the vector of non-reachability values g^* for different starting types is precisely the Greatest Fixed Point (GFP) solution of the system $x = P(x)$ in $[0,1]^n$. To define these equations, some shorthand notation will be useful. We use x^v to denote the monomial $x_1^{v_1} x_2^{v_2} \cdots x_n^{v_n}$ for an n-vector of variables $x = (x_1, \cdots, x_n)$ and a vector $v \in \mathbb{N}^n$. Considering a multi-variate polynomial $P_i(x) = \sum_{r\in R} p_r x^{\alpha_r}$ for some rational coefficients $p_r, r \in R$, we will call $P_i(x)$ a *probabilistic polynomial*, if $p_r \geq 0$ for all $r \in R$ and $\sum_{r\in R} p_r \leq 1$. A *minimax probabilistic polynomial system of equations* (minimax-PPS), $x = P(x)$, is a system of n equations in n variables $x = (x_1, \ldots, x_n)$, where for each $i \in \{1, \ldots, n\}$, $P_i(x) := Val(A_i(x))$, where $A_i(x)$ is a matrix whose entries are probabilistic polynomials, and $Val(A_i(x))$ is the minimax value of the finite two-player zero-sum game with payoff matrix $A_i(x)$ for each $x \in \mathbb{R}^n$. Note that as special cases, when $A_i(x)$ has only one row or only one column, then $Val(A_i(x))$ is the maximum or minimum of a set of probabilistic polynomials, and when it has only one row and column, then $Val(A_i(x))$ is simply a probabilistic polynomial.

For computational purposes, we assume that all coefficients are rational and that there are no zero terms in the probabilistic polynomials, and we assume the coefficients and non-zero exponents of each term are given in binary. We denote by $|P|$ the total bit encoding length of a system $x = P(x)$ under this representation. Since $P(x)$ defines a monotone function $P : [0,1]^n \to [0,1]^n$, it follows from Tarski's theorem ([30])) that any such system has both a *Least Fixed Point* (LFP) solution $q^* \in [0,1]^n$, and a *Greatest Fixed Point* (GFP) solution, $g^* \in [0,1]^n$. In other words, $q^* = P(q^*)$ and $g^* = P(g^*)$ and moreover, for all $s^* \in [0,1]^n$ such that $s^* = P(s^*)$, we have $q^* \leq s^* \leq g^*$ (coordinate-wise inequality).

For convenience in proofs and algorithms throughout the paper and to simplify the structure of the matrices involved, we shall observe that minimax-PPSs can always be cast in the following normal form. A minimax-PPS in *simple normal form* (SNF), $x = P(x)$, is a system of n equations in n variables $\{x_1, \cdots, x_n\}$, where each $P_i(x)$ for $i = 1, 2, \ldots, n$ is one of three forms:

- FORM L: $P_i(x) = a_{i,0} + \sum_{j=1}^n a_{i,j} x_j$, where for all j, $a_{i,j} \geq 0$, and $\sum_{j=0}^n a_{i,j} \leq 1$
- FORM Q: $P_i(x) = x_j x_k$ for some j, k
- FORM M: $P_i(x) = Val(A_i(x))$, where $A_i(x)$ is a $(n_i \times m_i)$ matrix, such that for all $j \in [n_i]$ and $k \in [m_i]$, the entry $(A_i(x))_{j,k} \in \{x_1, \ldots, x_n\} \cup \{1\}$.

We shall often assume a minimax-PPS in its SNF form, and say that a variable x_i is "of form/type" L, Q, or M, meaning that $P_i(x)$ has the corresponding form.

▶ **Proposition 1** (informal statement; see full version for formal statement and proof)**.** *Every minimax-PPS, $x = P(x)$, can be transformed in P-time to an "equivalent" minimax-PPS, $y = Q(y)$ in SNF form, such that $|Q| \in O(|P|)$.*

Thus, for the rest of this paper *we may assume, without loss of generality, that all minimax-PPSs are in SNF normal form.*

3 Non-reachability values for BCSGs and the Greatest Fixed Point

We show that for a given BCSG with a target type T_{f^*}, a minimax-PPS, $x = P(x)$, can be constructed such that its *greatest fixed point* (GFP) $g^* \in [0,1]^n$ is precisely the vector g^* of non-reachability values for the BCSG. For simplicity, from now on let us call a *maximizer* (respectively, a *minimizer*) the player that aims to *maximize* (respectively, *minimize*) the probability of *not* reaching the target type. That is, we swap the roles of the players for the benefit of less confusion in analysing the minimax-PPS which captures non-reachability values in its GFP.

For each type $T_i \neq T_{f^*}$, the minimax-PPS will have an associated variable x_i and an equation $x_i = P_i(x)$, and the $P_i(x)$ is defined as follows. For each action $a_{max} \in \Gamma^i_{max}$ of the maximizer (i.e., the player aiming to maximize the probability of *not* reaching the target) and action $a_{min} \in \Gamma^i_{min}$ of the minimizer, in T_i, let $R'(T_i, a_{max}, a_{min}) = \{r \in R(T_i, a_{max}, a_{min}) \mid (\alpha_r)_{f^*} = 0\}$ be the set of probabilistic rules r for type T_i and players' action pair (a_{max}, a_{min}) that generate a multi-set α_r which does not contain an object of the target type. For each action pair for T_i, there is a probabilistic polynomial $q_{i,a_{max},a_{min}}(x) := \sum_{r \in R'(T_i,a_{max},a_{min})} p_r x^{\alpha_r}$. Now we let $P_i(x) \equiv Val(A_i(x))$ be the value of a finite zero-sum game with matrix $A_i(x)$, where the matrix is constructed as follows: (1) rows belong to the max player in the minimax-PPS (i.e., the player trying to maximize the non-reachability probability), and columns belong to the min player; (2) for each row and column (i.e., pair of actions (a_{max}, a_{min})) the matrix entry $A_i(x)_{a_{max},a_{min}}$ is the corresponding probabilistic polynomial $q_{i,a_{max},a_{min}}(x)$.

The following theorem captures the fact that the optimal *non-reachability values* g^* in the BCSG game exist (meaning these game do have a value) and correspond to the GFP of the minimax-PPS $x = P(x)$ that was just defined.

▶ **Theorem 2.** *The non-reachability game values $g^* \in [0,1]^n$ of a BCSG reachability game exist, and correspond to the GFP of the minimax-PPS, $x = P(x)$, in $[0,1]^n$. That is, $g^* = P(g^*)$, and for all other fixed points $g' = P(g')$ in $[0,1]^n$, it holds that $g' \leq g^*$. Moreover, for an initial population μ, the optimal non-reachability value is $g^*(\mu) = \prod_i (g^*_i)^{\mu_i}$ and the game is determined, i.e., $g^*(\mu) = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} g^*_{\sigma,\tau}(\mu) = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} g^*_{\sigma,\tau}(\mu)$. Finally, the player maximizing non-reachability probability in the BCSG has a (mixed) static optimal strategy.*

▶ **Corollary 3.** *Given a BCSG reachability game, and a probability $p \in (0,1)$, deciding whether the game value is $\geq p$ is in PSPACE.*

The PSPACE upper bound follows from Theorem 2, by appealing to decision procedures for the (existential) theory of reals to answer quantitative questions about the GFP of the corresponding minimax-PPS equations. This is entirely analogous to very similar arguments in [15, 13, 16], so we do not elaborate. Any substantial improvement on PSPACE for such quantitative decision problems would require a major breakthrough on exact numerical computation, even for BPs or BMDPs (see [16, 13, 15]).

4 P-time algorithm for almost-sure reachability for BCSGs

Before considering almost-sure reachability, we first show there is a simple P-time algorithm for computing the variables x_i with value $q_i^* = 1$ for the GFP in a given minimax-PPS, $x = P(x)$, or in other words, for a given BCSG, deciding whether the reachability value, starting with an object of a given type T_i, is 0. The algorithm is easy, amounts to an AND-OR graph reachability analysis, and is very similar to the algorithm given for deciding $q_i^* = 1$ for BSSGs in [13]. Let $W = \{x_1, \ldots, x_n\}$ denote the set of all variables of the minimax-PPS. The algorithm is given in Figure 1.

1. Initialize $S := \{x_i \in W \mid P_i(\mathbf{1}) < 1\}$.
2. Repeat until no change has occurred:
   a. if there is a variable $x_i \notin S$ of form L or Q such that $P_i(x)$ contains a variable already in S, then add x_i to S.
   b. if there is a variable $x_i \notin S$ of form M such that for every action $a_{max} \in \Gamma^i_{max}$, there exists an action $a_{min} \in \Gamma^i_{min}$, such that $A_i(x)_{(a_{max},a_{min})} \in S$, then add x_i to S.
3. Output the set $\bar{S} := W - S$.

Figure 1 Simple P-time algorithm for computing the set of types with reachability value 0 in a given BCSG, or equivalently the set of variables $\{x_i \mid q_i^* = 1\}$ of the associated minimax-PPS.

▶ **Proposition 4.** *There is a P-time algorithm that, given a BCSG or equivalently a corresponding minimax-PPS, $x = P(x)$, with n variables and with GFP $q^* \in [0,1]^n$, and given $i \in [n]$, determines whether $q_i^* = 1$ or $q_i^* < 1$. In case $q_i^* = 1$, the algorithm can produce a deterministic static strategy σ for the max player (maximizing non-reachability) that forces $q_i^* = 1$. Otherwise, if $q_i^* < 1$, the algorithm can produce a mixed static strategy τ for the min player (minimizing non-reachability) that guarantees $q_i^* < 1$.*

We are now ready to present a P-time algorithm for almost-sure reachability for BCSGs. First, as a preprocessing step, we apply the algorithm of Proposition 4, which identifies in P-time all the variables x_i where $q_i^* = 1$. We then remove these variables from the system, substituting the value 1 in their place. We then simplify and reduce the resulting SNF-form minimax-PPS into a reduced form, with GFP $q^* < 1$. Note that the resulting reduced SNF-form minimax-PPS may contain some variables x_j of form M, whose corresponding matrix $A_j(x)$ has some entries that contain the value 1 rather than a variable (because we substituted 1 for removed variables x_j, where $q_j^* = 1$). Note also that in the reduced SNF-form minimax-PPS each variable x_i of form Q has an associated quadratic equation $x_i = x_j x_k$, because if one of the variables (say x_k) on the right-hand side was set to 1 during preprocessing, the resulting equation ($x_i = x_j$) would have been declared to have form L in the reduced minimax-PPS. We henceforth assume that the minimax-PPS is in SNF-form, with $q^* < 1$, and we let X be its set of (remaining) variables. We now apply the algorithm of Figure 2 to the minimax-PPS with $q^* < 1$, which identifies the variables x_i in the minimax-PPS (equivalently, the types in the BCSG), from which we can almost-surely reach the target type T_{f^*} (i.e., $q_i^* = 0$ *and* there is a strategy τ^* for the player minimizing non-reachability probability that achieves this value, no matter what the other player does).

▶ **Theorem 5.** *Given a BCSG with minimax-PPS, $x = P(x)$, such that the GFP $q^* < \mathbf{1}$, the algorithm in Figure 2 terminates in polynomial time and returns the set of variables $\{x_i \in X \mid \exists \tau \in \Psi_2\ (q^*_{*,\tau})_i = 0\}$.*

1. Initialize $S := \{x_i \in X \mid P_i(\mathbf{0}) > 0$, that is $P_i(x)$ has a constant term $\}$.
   Let $\gamma_0^i := \Gamma_{min}^i$ for every variable $x_i \in X - S$. Let $t := 1$.
2. Repeat until no change has occurred to S:
   a. if there is a variable $x_i \in X - S$ of form L where $P_i(x)$ contains a variable already in S, then add x_i to S.
   b. if there is a variable $x_i \in X - S$ of form Q where both variables in $P_i(x)$ are already in S, then add x_i to S.
   c. if there is a variable $x_i \in X - S$ of form M and if for all $a_{min} \in \Gamma_{min}^i$, there exists a $a_{max} \in \Gamma_{max}^i$ such that $A_i(x)_{(a_{max},a_{min})} \in S \cup \{1\}$, then add x_i to S.
3. For each $x_i \in X - S$ of form M, let:
   $\gamma_t^i := \{a_{min} \in \gamma_{t-1}^i \mid \forall a_{max} \in \Gamma_{max}^i,\ A_i(x)_{(a_{max},a_{min})} \notin S \cup \{1\}\}$. (Note that $\gamma_t^i \subseteq \gamma_{t-1}^i$.)
4. Let $F := \{x_i \in X - S \mid P_i(\mathbf{1}) < 1$, or $P_i(x)$ is of form Q $\}$
5. Repeat until no change has occurred to F:
   a. if there is a variable $x_i \in X - (S \cup F)$ of form L where $P_i(x)$ contains a variable already in F, then add x_i to F.
   b. if there is a variable $x_i \in X - (S \cup F)$ of form M such that for $\forall a_{max} \in \Gamma_{max}^i$, there is a min player's action $a_{min} \in \gamma_t^i$ such that $A_i(x)_{(a_{max},a_{min})} \in F$, then add x_i to F.
6. If $X = S \cup F$, **return** F, and halt.
7. Else, let $S := X - F$, $t := t + 1$, and go to step 2.

Figure 2 P-time algorithm for computing almost-sure reachability types $\{x_i \mid \exists \tau \in \Psi_2\ (g_{*,\tau}^*)_i = 0\}$ for a minimax-PPS (in SNF), associated with a given BCSG.

The proof is in the full version. Here we give very brief intuition for why the algorithm works. The set S will accumulate variables $x_i \in X$, such that regardless of the strategy $\tau \in \Psi_2$ of the player minimizing non-reachability (i.e., maximizing reachability), the probability of reaching the target type is < 1. The loop in Step (2.) is a basic "attractor set" construction that adds to S any variable x_i that should be in S by virtue of prior membership in S of variables (types) occurring in $P_i(x)$. In step (3.), for each variable $x_i \in X - S$ we maintain the remaining "useful" set of actions $\gamma_t^i \subseteq \Gamma_{min}^i$ that can avoid the set S (or extinction). The loop in Step (5.) accumulates a set $F \subseteq X - S$, such that for every $x_i \in F$ there is a strategy $\tau \in \Psi_2$ to either reach the target or a branching (quadratic) type in F, with positive probability, regardless of the opponent's strategy. The key assertion is this: if in step (6.) we find all variables are already either in S or in F, we are done; F must be the set of types from which we can force almost-sure reachability of the target type; otherwise, all variables in $X - (F \cup S)$ can be added to S. The reason this assertion holds is not obvious (see the detailed proof). The proof of the theorem also yields the following:

▶ **Corollary 6.** *Let F be the set of variables output by the algorithm in Figure 2.*

1. *Let $S = X - F$. There is a randomized non-static strategy $\hat{\sigma}$ for the max player (maximizing non-reachability) such that for all $x_i \in S$, and for all strategies τ of the min player (minimizing non-reachability), starting with one object of type T_i the probability of reaching the target type is < 1.*
2. *There is a randomized non-static strategy $\hat{\tau}$ for the min player (minimizing non-reachability) such that for all strategies σ of the max player (maximizing non-reachability), and for all $x_i \in F$, starting at one object of type T_i the probability of reaching the target type is 1.*

The non-static strategies $\hat{\sigma}$ and $\hat{\tau}$ mentioned above both have suitably compact descriptions (as functions that map finite histories to distributions over actions for entities in the current population) and can be described in such a compact form in polynomial time, as a function of the encoding size of the input BCSG.

The strategies need to be represented compactly because some of the actions probabilities may be doubly-exponentially small (or doubly-exponentially close to 1), and this is inherent.

5 P-time algorithm for limit-sure reachability for BCSGs

In this section we focus on the limit-sure reachability problem, i.e., starting with one object of a type T_i, decide whether the reachability value is 1. Since we translate reachability into non-reachability when analysing the corresponding minimax-PPS, we are asking whether there exists a sequence of strategies $\langle \tau^*_{\epsilon_j} \mid j \in \mathbb{N} \rangle$ for the min player, such that $\forall j \in \mathbb{N}$, $\epsilon_j > \epsilon_{j+1} > 0$, and where $\lim_{j\to\infty} \epsilon_j = 0$, such that the strategy $\tau^*_{\epsilon_j}$ forces non-reachability probability to be at most ϵ_j, regardless of the strategy σ used by the max player. In other words, for a given starting object of type T_i, we ask whether $\inf_{\tau\in\Psi_2}(g^*_{*,\tau})_i = 0$.

1. Initialize $S := \{x_i \in X \mid P_i(\mathbf{0}) > 0$, that is $P_i(x)$ has a constant term $\}$.
2. Repeat until no change has occurred to S:
   a. if there is a variable $x_i \in X - S$ of form L where $P_i(x)$ contains a variable already in S, then add x_i to S.
   b. if there is a variable $x_i \in X - S$ of form Q where both variables in $P_i(x)$ are already in S, then add x_i to S.
   c. if there is a variable $x_i \in X - S$ of form M and if for all $a_{min} \in \Gamma^i_{min}$, there exists $a_{max} \in \Gamma^i_{max}$ such that $A_i(x)_{(a_{max},a_{min})} \in S \cup \{1\}$, then add x_i to S.
3. Let $F := \{x_i \in X - S \mid P_i(\mathbf{1}) < 1$, or $P_i(x)$ is of form Q $\}$
4. Repeat until no change has occurred to F:
   a. if there is a variable $x_i \in X - (S \cup F)$ of form L where $P_i(x)$ contains a variable already in F, then add x_i to F.
   b. if there is a variable $x_i \in X - (S \cup F)$ of form M and if the following procedure returns "Yes", then add x_i to F.
      i. Set $L_0 := \emptyset$, $B_0 := \emptyset$, $k := 0$. Let $O := X - (S \cup F)$.
      ii. Repeat:
         - $k := k + 1$.
         - $L_k := \{a_{min} \in \Gamma^i_{min} - \bigcup_{j=0}^{k-1} L_j \mid \forall a_{max} \in \Gamma^i_{max} - B_{k-1},\ A_i(x)_{(a_{max},a_{min})} \in F \cup O\}$.
         - $B_k := B_{k-1} \cup \{a_{max} \in \Gamma^i_{max} - B_{k-1} \mid \exists a_{min} \in L_k \ s.t.\ A_i(x)_{(a_{max},a_{min})} \in F\}$.

         Until $B_k = B_{k-1}$.
      iii. Return: "Yes" if $B_k = \Gamma^i_{max}$, and "No" otherwise.
5. If $X = S \cup F$, **return** F, and halt.
6. Else, let $S := X - F$, and go to step 2.

Figure 3 P-time algorithm for computing the set of types that satisfy limit-sure reachability in a given BCSG, i.e., the set of variables $\{x_i \mid g^*_i = 0\}$ in the associated minimax-PPS.

Again, as in the almost-sure case, we first, as a preprocessing step, use the P-time algorithm from Proposition 4 to remove all variables x_i such that $g^*_i = 1$, and we substitute 1 for these variables in the remaining equations. We hence obtain a reduced SNF-form

minimax-PPS, for which we can assume $g^* < 1$. The set of all remaining variables in the SNF-form minimax-PPS is again denoted by X. Thereafter, we apply the algorithm in Figure 3, which computes the set of variables, x_i, such that $g^*_i = 0$. In other words, we compute the set of types, such that starting from one object of that type the value of the reachability game is 1.

▶ **Theorem 7.** *Given a BCSG with minimax-PPS, $x = P(x)$, with GFP $g^* < \mathbf{1}$, the algorithm in Figure 3 terminates in polynomial time, and returns the set of variables $\{x_i \in X \mid g^*_i = 0\}$.*

The proof is in the full version. We again give very brief intuition for why the algorithm works. The algorithm is similar in structure to that of almost-sure reachability, but differs in crucial aspects. Firstly, step (2.) now accumulates the variables x_i for which we know $g^*_i > 0$. In step (3.), as before, the set F is initialized to variables $x_i \in X - S$ (types) where either $P_i(x)$ is quadratic, or the target type is generated with positive probability in the next step, i.e., $(P_i(\mathbf{1}) < 1)$. In step (4.) the algorithm makes crucial use of a "limit-escape" set construction (inside inner loop 4.(b)), a variant of which was used by de Alfaro, Henzinger, and Kupferman in [8], in the context of finite-state CSGs, but which has been adapted here to the context of BCSGs. This loop adds a variable x_i to the set F whenever it is the case that, for any $\epsilon > 0$, the player minimizing non-reachability can play a distribution on actions (depending on ϵ) at any object e of type T_i such that (regardless of the action distribution of the other player) the probability that e produces an offspring in the next generation whose type is in S is at most ϵ times the probability that it produces an offspring that is already in F. Unlike the almost-sure case (where we maintain remaining sets of "useful" actions $\gamma^i_t \subset \Gamma^i_{min}$ that can avoid the set S), in the limit-sure case the player minimizing non-reachability may not have any (distribution on) actions that entirely avoid the set S, but it nevertheless may have a series of distributions on actions that make the ratio of the probability of the "bad" event of generating an offspring in S, compared to the probability of the "good" event of generating an offspring in F, arbitrarily small. Similar to the case of almost-sure reachability, a key assertion is that if in step (5.) all variables in X are already in $S \cup F$ then we are done: F consists of precisely those variables (types) that satisfy limit-sure reachability; otherwise, we can add $X - (S \cup F)$ to the set S. The reason why this holds is again subtle (see the detailed proof in the full version).

The proof of the theorem also yields the following corollary:

▶ **Corollary 8.** *Suppose the algorithm in Figure 3 outputs the set F when it terminates. Let $S := X - F$.*

1. *There is a randomized static strategy $\hat{\sigma}$ for the max player (maximizing non-reachability) such that for all variables $x_i \in S$, we have $(g^*_{\hat{\sigma},*})_i > 0$.*
2. *For all $\epsilon > 0$, there is a randomized static strategy τ_ϵ, for the min player (minimizing non-reachability), such that for all variables $x_i \in F$, $(g^*_{*,\tau_\epsilon})_i \leq \epsilon$.*

The static strategies $\hat{\sigma}$ and τ_ϵ mentioned above can both be described, in a suitably compact form, in polynomial time, as a function of the encoding size of the input BCSG. However, these static strategies, specifically in the case of τ_ϵ, involve probabilities that are double-exponentially small (and double-exponentially close to 1), as a function of the encoding size of the BCSG, so these probabilities have to be encoded in a suitable succinct notation in order for the output to have polynomial encoding size.

References

1 R. Bonnet, S. Kiefer, and A. W. Lin. Analysis of Probabilistic Basic Parallel Processes. In *Proc. of FoSSaCS'14*, pages 43–57, 2014.

2 Bozic and et. al. Evolutionary dynamics of cancer in response to targeted combination therapy. *eLife*, 2:e00747, 2013.

3 T. Brázdil, V. Brozek, V. Forejt, and A. Kucera. Reachability in recursive markov decision processes. *Inf. Comput.*, 206(5):520–537, 2008.

4 T. Brázdil, V. Brozek, A. Kucera, and J. Obdrzálek. Qualitative reachability in stochastic BPA games. *Inf. Comput.*, 209(8):1160–1183, 2011.

5 K. Chatterjee, K. A. Hansen, and R. Ibsen-Jensen. Strategy complexity of concurrent safety games. In *Proc. of 42nd Inter. Symp. on Math. Found. of Computer Science (MFCS)*, volume 83 of *LIPIcs*, pages 55:1–55:13, 2017.

6 T. Chen, K. Dräger, and S. Kiefer. Model Checking Stochastic Branching Processes. In *Proc. of MFCS'12*, volume 7464 of *Springer LNCS*, pages 271–282, 2012.

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

8 L. de Alfaro, T. A. Henzinger, and O. Kupferman. Concurrent reachability games. *Theoretical Computer Science*, 386(3):188–217, 2007. (Conference version in FOCS'98).

9 E. Denardo and U. Rothblum. Totally expanding multiplicative systems. *Linear Algebra Appl.*, 406:142–158, 2005.

10 J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. *Logical Methods in Computer Science*, 2(1):1–31, 2006.

11 K. Etessami, A. Stewart, and M. Yannakakis. Polynomial-time algorithms for Branching Markov Decision Processes, and probabilistic min(max) polynomial Bellman equations. In *Proc. of 39th Int. Coll. on Automata, Languages and Programming (ICALP)*, 2012. (All references are to the full preprint Arxiv:1202.4789).

12 K. Etessami, A. Stewart, and M. Yannakakis. A polynomial-time algorithm for computing extinction probabilities of multitype branching processes. *SIAM J. Computing*, 46(5):1515–1553, 2017. (Conference version in STOC'12).

13 K. Etessami, A. Stewart, and M. Yannakakis. Greatest Fixed Points of Probabilistic Min/Max Polynomial Equations, and Reachability for Branching Markov Decision Processes. *Information and Computation*, 261:355–382, 2018. (special issue for ICALP'15).

14 K. Etessami, D. Wojtczak, and M. Yannakakis. Recursive stochastic games with positive rewards. In *Proc. of 35th ICALP (1)*, volume 5125 of *Springer LNCS*, pages 711–723, 2008.

15 K. Etessami and M. Yannakakis. Recursive Concurrent Stochastic Games. *Logical Methods in Computer Science*, 4(4), 2008.

16 K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. *Journal of the ACM*, 56(1), 2009.

17 K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. *Journal of the ACM*, 62(2):1–69, 2015.

18 H. Everett. Recursive games. *Contributions to the Theory of Games*, 3(39):47–78, 1957.

19 S. K. S Frederiksen and P. B. Miltersen. Approximating the Value of a Concurrent Reachability Game in the Polynomial Time Hierarchy. In *Proc. of 24th ISAAC*, pages 457–467, 2013.

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

21 K. A. Hansen, R. Ibsen-Jensen, and P. B. Miltersen. The Complexity of Solving Reachability Games using Value and Strategy Iteration. *Theory Comput. Syst.*, 55(2):380–403, 2014.

22 K. A. Hansen, M. Koucky, and P. B. Miltersen. Winning Concurrent Reachability Games Requires Doubly-Exponential Patience. In *Proc. of 24th Annual IEEE Symp. on Logic in Computer Science*, pages 332–341, 2009.

23 T. E. Harris. *The Theory of Branching Processes*. Springer-Verlag, 1963.

24 M. Kimmel and D. E. Axelrod. *Branching processes in biology*. Springer, 2002.

25 H. Michalewski and M. Mio. On the problem of computing the probability of regular sets of trees. In *Proc. of FSTTCS'15*, pages 489–502, 2015.

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

27 M. Przybyłko and M. Skrzypczak. On the complexity of branching games with regular conditions. In *Proc. of MFCS'16*, volume 78 of *LIPIcs*, 2016.

28 G. Reiter, I. Bozic, K. Chatterjee, and M. A. Nowak. TTP: Tool for tumor progression. In *Proc. of CAV'2013*, volume 8044 of *Springer LNCS*, pages 101–106, 2013.

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

30 A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955.

31 D. Wojtczak. Expected termination time in BPA games. In *Proc of ATVA'2013*, pages 303–318, 2013.