2010 Seventh International Conference on the Quantitative Evaluation of Systems

Timed Branching Processes

Ashutosh Trivedi and Dominik Wojtczak
Computing Laboratory, The University of Oxford, UK
Email: {ashutosh.trivedi,dominik.wojtczak}@comlab.ox.ac.uk

***Abstract*—We study Timed Branching Processes (TBPs), a natural extension of (multitype) Branching Processes (BPs) where each entity is equipped with a finite set of private continuous variables, called clocks. Clocks grow uniformly with the same rate and using them various timing constraints can be imposed on the branching rules of the system, e.g. the way an entity reproduces (branches) can depend on its age. In comparison with discrete-time BPs, where all the entities live for a constant amount of time before they branch (and die), and more general continuous-time BPs, where for each entity the amount of time before the branching takes place is governed by an exponential distribution, our model can be seen as an abstraction of continuous-time BPs where we do not know the exact distribution on the time before an entity branches, but rather some time interval when it happens. Allowing an external controller to decide at what point in time the branching takes place permits us to study the best/worst behaviour of the system. For each given instance of TBP, we show how to answer the following questions: What is the supremum probability of extinction for a given initial population? What is the supremum probability that a given population becomes extinct in less than *t* time units? What is the supremum expected number of entities of a given type that will be created before the population becomes extinct?**

***Keywords*-Multitype Branching Processes; Timed Automata; Probabilistic Timed Automata.**

I. Introduction

We study Timed Branching Processes (TBPs), a natural extension of (Multi-typed) Branching Processes (BPs). BPs are a natural model used for studying behaviour of population dynamics. A population consists of entities of various types (possibly many entities of the same type can coexist at the same time) and each of them branches after some time into a set (possibly empty) of entities of various types while disappearing itself.[1] This assumption is natural, for instance, for annual plants that reproduce only at a specific time of the year. The set of offspring of an entity is chosen at random among many possibilities with some fixed distribution that depends only on the type of the entity that has branched. The type can describe fundamental differences between entities, e.g. stem cells are very different from regular cells, or it can correspond to some characteristics of the entities, such as their age or size. Although the entities coexist with each other, the BP model assumes that there is no interaction between them, so how they reproduce and for how long they live is the same as if they were the only entities in the system. This assumption greatly improves the computational complexity of the analysis of such models and is natural in situations where the population exists in an environment that has virtually unlimited resources to sustain the growth of the population, e.g. common situation for bacteria or insects. BPs has wide applicability in modelling various physical phenomena, such as nuclear chain reactions, red blood cell formation, population genetics, population migration, epidemic outbreaks and molecular biology (see, e.g. [1] for many examples of BP models used in biological systems). The lifespan of an entity is always equal to one time unit in discrete-time BPs, and is exponentially distributed for continuous-time BPs. In other words, both of these models assume that an entity of the same type lives and reproduces in the same way regardless of its past and its age. In our model, in TBPs, the evolution happens in real-time just like for the continuous-time model. However, the point of time at which the branching takes place for each entity can be chosen by the controller (i.e. it is chosen nondeterministically). This model can be seen as an abstraction of continuous-time BPs for which we do not know the exact distribution on the time before an entity branches, but rather some lower and upper bound on that time. Allowing an external controller to decide at what point in time the branching takes place, permits us to study the best/worst behaviour of the examined model. Also, in TBPs, the branching rules and the probabilities assigned to them can change throughout the life of an entity. We allow each entity to have a finite set of real-time private clocks and an entity can neither access nor modify clocks of the other entities in the system. An offspring inherits the values of the clocks of its parent, but while doing so, their values can be arbitrarily rearranged or some of them can be reset to 0. Which clocks are reset and the way the rest of them are assigned is common for all the offspring of a given branching action.[2] Notice that, since there is no bound on the number of entities in our system, and each of the entities has some finite number of private clocks, we will need to analyse a system with potentially infinitely many real-time clocks.

The clocks can measure not only the age of an entity or the age of its parent, but also other parameters like its size, weight or in the case an entity is a particle, its energy level. All of them can be assumed to increase uniformly with time and once such a parameter exceeds a certain threshold a splitting of the entity occurs. Of course, in such a case it would be natural to split the values of the clocks equally between its offspring. However, for such systems the problems we study in this paper become undecidable (see our technical report [2] for details)

[1] Actually, it is not necessary to assume that an entity dies immediately after reproduction as long as it reproduces at most once during its lifetime.

[2] It is possible to restrict our model to allow only standard resets of the clocks by introducing new types into the model. However, the number of types would need to increase exponentially in the number of clocks. At the same time, rearrangement of the clocks is very useful when modelling such systems, so instead we allow such operations explicitly in our model.

978-0-7695-4188-4/10 $26.00 © 2010 IEEE
DOI 10.1109/QEST.2010.36

IEEE computer society

and so we will focus on the case where these values are only allowed to be copied or reset to 0.

As a motivating example, let us discuss a simple model of bacteria population. There are two types of bacteria, harmful ones that cause some damage to the organism and harmless ones that do not. Each type evolves differently, but each bacterium has to branch within six minutes after it was born. All harmful bacteria die between the second and fifth minute of their life without any offspring. On the other hand, there are two possibilities for harmless bacteria to evolve. Assuming that a harmless bacterium is at least four and at most six minutes old, it can branch (while disappearing itself) into one harmful and one harmless bacterium with probability 1. As a second possibility, assuming that the harmless bacterium is at least one and at most three minutes old and its grandparent was born no more than five minutes ago (if it existed, otherwise its parent), the bacterium branches with probability $\frac{1}{4}$ into two harmless ones and with the remaining probability, it changes into a harmful bacterium.

We would like to answer the following questions about this model. Is the supremum probability that a population consisting of one harmless bacterium becomes extinct greater than $\frac{2}{3}$ (i.e. can we make the organism recover with probability $\geq \frac{2}{3}$)? Is the supremum probability that this happens within one hour greater than $\frac{1}{2}$ (i.e. can we make this bacteria population extinct within one hour with probability $\geq \frac{1}{2}$)? What is the supremum of the expected number of harmful bacteria created before extinction? And more importantly, for estimating the worst possible expected cumulative damage done to the organism, what is the supremum of the expected value of the sum of lifetime of all harmful bacteria before the whole population becomes extinct? (That is, e.g. time periods when exactly two harmful bacteria coexisted in the system are counted twice.) TBPs are a suitable formalism for modelling evolutions of such populations and we will show how to answer this sort of questions about them.

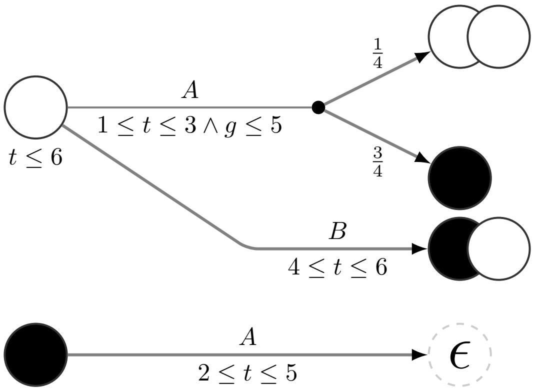

Fig. 1. Reproduction of the example bacteria population. Harmless bacteria are white, harmful are black. Clocks t, p and g correspond to the the age of the bacterium, the age of its parent and the age of its grandparent, respectively. An offspring of an entity inherits the values of its parent clocks after the following reassignment is applied to them $\theta(t, p, g) = (0, t, p)$ (in other words, we shift the values of the clocks one position to the right and reset the age of the new entity to 0). There are two actions available to the controller, A and B.

Also, TBPs can be used to analyse the expected total workload of distributed computations. The goal is to complete a task, which can be split in many different ways into subtasks that can be performed independently on a distributed computer network. Each such subtask can be split further and so on. We do not know exactly how the execution of a given task will proceed. For some of the subtasks it is the computer executing it that decides how to split a given subtask and sometimes this choice is probabilistic. For each subtask, we are given a lower and upper bound on its execution time and possible ways how its execution can proceed. In order to get an upper bound on the expected amount of money we would need to pay to run a given task, assuming that the cost of running a task is proportional to its total workload, we need to compute the worst possible expected total workload for the initial task.

It can be shown that the decision problems that we study in this paper become undecidable if entities coexisting at the same time are allowed to exchange information with each other, for instance by allowing entities when they die to reset some of the private clocks of the other entities. However, due to space constraints and to ease the exposition of the results, we restrain ourselves from defining such a general model in this paper and refer the reader to its full version [2].

A. Related work

The simplest model of BPs, Galton-Watson process ([3]) which is a discrete-time model where all entities are of the same type, dates as far back as 1874 and was used to explain why some aristocratic family surnames became exctinct. The generalisation of this model to multiple types of entities was first studied in 1940s by Kolmogorov and Sevast'yanov ([4]). For an overview of the results known for BPs, see e.g. [5] and [1]. The precise computational complexity of decision problems about the probabilities of extinction of an arbitrary BP was done for the first time in [6]. *Branching Decision Processes* (BDPs), a natural generalisation of BPs to controlled setting was studied before for discrete-time BPs in the OR literature (e.g. [7], [8]) and found applications in manpower planning, controlled queuing networks, management of livestock and epidemic control, among others. The focus of these works was on optimising the expected average or discounted reward over a run of the process, or optimising the population growth rate. In [9] and [10], it was shown that decision problems about probability of extinction of BDPs are polynomial-time many-one reducible to computing the optimal probability of termination for so-called 1-exit Recursive Markov Decision Processes. The computational complexity of approximating the probability of extinction for BDPs follows from these results. Similarly, the computational complexity of computing the best/worst expected total reward before extinction follows from [11].

In the branching processes arising in biology, the probability that an entity will branch is usually dependent on its age. Age-dependent models of BPs were considered before in the literature (see, e.g. [5], [1]), but not in the presence of time constraints on transitions and invariants on states. TBPs can be used to perform the best/worst behaviour analysis of age-dependent continuous-time branching processes.

Timed automata, defined in [12], and its extension to probabilistic setting ([13]) are the most commonly used formalisms for describing systems that evolve in real-time. The use of timing constraints to describe the behaviour of the system is both intuitive and expressive enough to capture a large class of real-time systems occurring in practice, and yet many of the decision problems for these models have moderate computational complexity.

B. Summary of the results

In this paper we show complexity upper bounds for the following problems. The problem of deciding whether the optimal expected probability of extinction of a given initial population is $\geq p$ (p is given in binary) is in EXPSPACE. Moreover, the problem of deciding whether the optimal time-bounded probability of extinction, with time bound T (given in binary), is $\geq p$ is as well in EXPSPACE. Finally, for the total reward objective, which for instance can be used to compute the supremum of the expected number of entities of a given type created before the whole population becomes extinct, the optimal value can be computed exactly in EXPTIME. The exponential blow-up in the computational complexity of all these problems comes from the (boundary) region abstraction of the underlying Timed Automata. On the other hand, if all entities have just one private clock, the computational complexity remains the same as for BDPs without any clocks.

II. DEFINITIONS

A. Preliminaries

1) Notation: We denote by $\mathbb{N}$ the set of non-negative integers, by $\mathbb{R}$ the set of reals and by $\mathbb{R}_{\oplus}$ the set of non-negative reals and $\mathbb{R}_{\oplus}^{\infty} = \mathbb{R}_{\oplus} \cup \{\infty\}$. For $n \in \mathbb{N}$, let $[\![n]\!]_{\mathbb{N}}$ and $[\![n]\!]_{\mathbb{R}}$ denote the sets $\{0, 1, \ldots, n\}$, and $\{r \in \mathbb{R} \,|\, 0 \leq r \leq n\}$ respectively. For a real number $r \in \mathbb{R}$ we write $\lfloor r \rfloor$ for the floor of r, i.e., largest integer $n \in \mathbb{N}$ such that $n \leq r$; and we write $\wr r \int$ for the fractional part of r, i.e. $r - \lfloor r \rfloor$. For an arbitrary set X, we denote by $|X|$ the number of elements in X and by X^* the set of all possible finite sequences (also called lists) of elements from X.

2) Lists: We will use α, β, γ to denote finite lists of elements. For two lists $\alpha = a_1, a_2, \ldots, a_k$ and $\beta = b_1, b_2, \ldots, b_l$: $\alpha \cdot \beta$ denotes their concatenation, in this case $\alpha \cdot \beta = a_1, \ldots, a_k, b_1, \ldots, b_l$; α_i denotes the i-th element of list α, in this case a_i and $|\alpha|$ denotes the length of list α, in this case k. The empty list will be denoted by ϵ ($|\epsilon| = 0$). Given a list $\alpha = a_1, a_2, a_3, \ldots, a_n$ and an element b, we write $\alpha \otimes b$ for the list $(a_1, b), (a_2, b), \ldots, (a_n, b)$. We can also compose this operation, e.g. $\alpha \otimes b \otimes c$ is the list $(a_1, b, c), (a_2, b, c), \ldots, (a_n, b, c)$.

3) Probability Distributions: A *discrete probability distribution* over a countable set Q is a function $\mu : Q \to [0, 1]$ such that $\sum_{q \in Q} \mu(q) = 1$. For a possible uncountable set Q', we define $\mathcal{D}(Q')$ to be the set of functions $\mu : Q' \to [0, 1]$ such that the set *supp*$(\mu) = \{q \in Q \,|\, \mu(q) > 0\}$ is countable and, over *supp*(μ), μ is a distribution. We say that $\mu \in \mathcal{D}(Q)$ is a *point distribution* if $\mu(q) = 1$ for some $q \in Q$.

4) Markov Decision Processes: Markov decision processes (see, e.g. [14]), are well-studied modelling formalism for systems exhibiting nondeterministic and probabilistic behaviour.

Definition 1: A *Markov decision process* (MDP) is a tuple $\mathcal{M} = (S, A, p, r)$ where:

- S is the set of *states*;
- A is the set of *actions*;
- $p : S \times A \to \mathcal{D}(S)$ is a partial function called the *probabilistic transition function*;
- $r : S \times A \to \mathbb{R}$ is the *reward function*.

We say that an MDP $\mathcal{M}$ is *finite* (*discrete*) if both S and A are finite (countable). We write $A(s)$ for the set of actions available at s, i.e. the set of actions a for which $p(s, a)$ is defined. In an MDP $\mathcal{M}$, if the current state is s, then one of the actions in $A(s)$ is chosen nondeterministically and if the chosen action is a then the probability of reaching state $s' \in S$ in the next step is $p(s, a)(s')$ and the reward obtained for this transition is $r(s, a)$.

B. Branching Decision Processes

Our definition of (multitype) Branching Decision Processes (BDPs) is essentially the same as in [7] and [9]. The differences are pointed out at the end of this section.

Definition 2: A *Branching Decision Process* (BDP) is a tuple $\mathcal{B} = (Q, A, p, r)$ where:

- Q is a set of *types*;
- A is a set of *actions*;
- $p : Q \times A \to \mathcal{D}(Q^*)$ is a partial function called the *probabilistic transition function*;
- $r : Q \times A \to \mathbb{R}$ is the *reward function*.

We say that an BDP $\mathcal{B}$ is *finite* if both S and A are finite. We write $A(q)$ for the set of actions available to an entity of type q, i.e., the set of actions a for which $p(q, a)$ is defined. Notice that our definition does not assume that the set of types and actions is finite; in fact the semantics of TBPs will be given as BDPs with uncountably many types.

Let us first describe informally how BDPs evolve. A *configuration* of a BDP $\mathcal{B}$ is a list of elements of Q that we call *entities*. A BDP starts at some initial configuration, $\alpha^0 \in Q^*$, and the controller picks simultaneously for each entity one of the actions available to an entity of its type. In the new configuration each entity is replaced by a list of new entities. These lists are picked according to the probability distribution $p(q, a)$ that depends both on the type of the entity, q, and the action, a, performed on it by the controller. The next configuration, α^1, is a concatenation of all these lists in the same order as these entities occurred in α^0. The process proceeds in exactly the same manner from α^1, moving to α^2, and from there to α^3, etc. Once the state ϵ is reached, i.e. no entities are present in the system, the process will stay in that state forever. Notice that the probability of moving to configuration β from α while performing a list of actions γ is the sum over all possible $\beta_1, \ldots, \beta_l$, where $l = |\alpha|$ and $\beta_1 \cdot \ldots \cdot \beta_l = \beta$, of the product of the probabilities that entity α_i branches into β_i after action γ_i was performed on it.

Definition 3 (Semantics of BDP): The semantics of a Branching Decision Process $\mathcal{B} = (Q, A, p, r)$ is given as an MDP $\mathcal{M}_{\mathcal{B}} = (\mathit{States}_{\mathcal{B}}, \mathit{Actions}_{\mathcal{B}}, \mathit{Prob}_{\mathcal{B}}, \mathit{Reward}_{\mathcal{B}})$ where:

- $\mathit{States}_{\mathcal{B}} = Q^*$ is the set of states;
- $\mathit{Actions}_{\mathcal{B}} = A^*$ is the set of actions;
- $\mathit{Prob}_{\mathcal{B}} : \mathit{States}_{\mathcal{B}} \times \mathit{Actions}_{\mathcal{B}} \to \mathcal{D}(\mathit{States}_{\mathcal{B}})$ is the probabilistic transition function such that for $\alpha \in \mathit{States}_{\mathcal{B}}$ and $\gamma \in \mathit{Actions}_{\mathcal{B}}$ we have that $\mathit{Prob}_{\mathcal{B}}(\alpha, \gamma)$ is defined when $|\gamma| = |\alpha|$ (this length is denoted by l) and $\gamma_i \in A(\alpha_i)$ for each i; moreover
$$\mathit{Prob}_{\mathcal{B}}(\alpha, \gamma)(\beta) = \sum_{\substack{\beta_1,\ldots,\beta_l \in Q^* \\ \beta_1 \cdot \ldots \cdot \beta_l = \beta}} \prod_{i=1}^{l} p(\alpha_i, \gamma_i)(\beta_i),$$
for every $\beta \in \mathit{States}_{\mathcal{B}}$. Notice that for a fixed β, there are only finitely many combinations of β_i-s such that $\beta_1 \cdot \ldots \cdot \beta_l = \beta$, so we sum over a finite set, even if the set Q has uncountably many elements. (Also, such defined $\mathit{Prob}_{\mathcal{B}}(\alpha, \gamma)$ is a probability distribution.)
- $\mathit{Reward}_{\mathcal{B}} : \mathit{States}_{\mathcal{B}} \times \mathit{Actions}_{\mathcal{B}} \to \mathbb{R}_{\oplus}$ is the reward function such that $\mathit{Reward}_{\mathcal{B}}(\alpha, \gamma) = \sum_{i=1}^{l} r(\alpha_i, \gamma_i)$;

Observe that according to this definition $\mathit{Prob}_{\mathcal{B}}(\epsilon, \epsilon)(\epsilon) = 1$ and $\mathit{Reward}_{\mathcal{B}}(\epsilon, \epsilon) = 0$. For a given BDP $\mathcal{B}$ and any of its states $\alpha \in \mathit{States}_{\mathcal{B}}$, we denote by $\mathit{Actions}_{\mathcal{B}}(\alpha)$ the set of actions $\gamma \in \mathit{Actions}_{\mathcal{B}}$ for which $\mathit{Prob}_{\mathcal{B}}(\alpha, \gamma)$ is defined.

Our semantics differs a little bit from the previous equivalent models [7] and [9]. First of all, a configuration of a BDP is represented explicitly as a simple list of entities, as opposed to a vector holding for each type the number of entities of that type in the current population. The vector representation can be a lot more succinct than the explicit one, because these numbers can be represented in binary. Secondly, as a consequence of representing configurations as lists, configuration s_1, s_2 is different than s_2, s_1 as well will be their evolution. In order to make our representation equivalent to the vector notation, we could define an equivalence relation that would group such configurations together. In the BDP model in [9], instead of having probability distribution on the outcomes of an action assigned to each type, the types are partitioned between controlled, for which the action can be chosen be the controller, and probabilistic ones, for which the choice of the next action is probabilistic. It can easily be seen that by introducing auxiliary types and transitions, that model can simulate tightly the other one. None of this differences influences the results presented in this paper.

C. Strategies

A *path* of a BDP $\mathcal{B}$ is a finite or infinite sequence $\pi = \alpha^0, (\gamma^1, \alpha^1), (\gamma^2, \alpha^2), \ldots \in \mathit{States}_{\mathcal{B}} \times ((\mathit{Actions}_{\mathcal{B}} \times \mathit{States}_{\mathcal{B}})^* \cup (\mathit{Actions}_{\mathcal{B}} \times \mathit{States}_{\mathcal{B}})^\omega)$, consisting of the initial state and a finite or infinite sequence of pairs of an action and a state, such that $\mathit{Prob}_{\mathcal{B}}(\alpha^i, \gamma^i)(\alpha^{i+1}) > 0$ for any $0 \le i \le |\pi|$, where $|\pi|$ is the number of actions taken during path π ($|\pi| = \infty$ if the path is infinite). For any path π we will denote by $\pi_{A(i)} (= \gamma^i)$ the i-th action taken during the path π, by $\pi_{S(i)} (= \alpha^i)$ the i-th state visited, where $\pi_{S(0)} (= \alpha^0)$ is the initial state, and by $\pi(i) (= \alpha^0, (\gamma^1, \alpha^1), \ldots, (\gamma^i, \alpha^i))$ the first i action-state pairs of the path π.

We will call a path of infinite (finite) length a *run* (*finite path*). We write $\mathit{Runs}_{\mathcal{B}}$ ($\mathit{FPath}_{\mathcal{B}}$) for the sets of all runs (finite paths) and $\mathit{Runs}_{\mathcal{B},\alpha}$ ($\mathit{FPath}_{\mathcal{B},\alpha}$) for the sets of all runs (finite paths) that start at a given initial state $\alpha \in \mathit{States}_{\mathcal{B}}$, i.e. paths π such that $\pi_{S(0)} = \alpha$. For a finite path π, its last state will be denoted by $\mathit{last}(\pi)$.

A *strategy* in BDP $\mathcal{B}$ is a function $\sigma : \mathit{FPath}_{\mathcal{B}} \to \mathcal{D}(\mathit{Actions}_{\mathcal{B}})$, such that for all $\pi \in \mathit{FPath}_{\mathcal{B}}$: $\mathit{supp}(\sigma(\pi)) \subseteq \mathit{Actions}_{\mathcal{B}}(\mathit{last}(\pi))$. We write $\Sigma_{\mathcal{B}}$ for the set of all strategies. A strategy σ is *pure* if $\sigma(\pi)$ is a point distribution for all $\pi \in \mathit{FPath}_{\mathcal{B}}$ for which σ is defined, while it is *memoryless* if for all $\pi, \pi' \in \mathit{FPath}_{\mathcal{B}}$ we have that $\mathit{last}(\pi)=\mathit{last}(\pi')$ implies $\sigma(\pi)=\sigma(\pi')$. A pure strategy can be identified with a function $\sigma : \mathit{FPath}_{\mathcal{B}} \to \mathit{Actions}_{\mathcal{B}}$. A strategy is called *static*, if it is pure and for any state and all entities of the same type in that state it picks the same action, i.e. for an arbitrary $\pi \in \mathit{FPath}_{\mathcal{B}}$, we have $\sigma(\pi)_i = \sigma(\pi)_j$ whenever $\mathit{last}(\pi)_i = \mathit{last}(\pi)_j$. A static and memoryless strategy is essentially a function of the form $\sigma : \mathit{States}_{\mathcal{B}} \to \mathit{Actions}_{\mathcal{B}}$.

A strategy $\sigma \in \Sigma_{\mathcal{B}}$ and an initial state α induce a probability measure over the set of runs of BDP $\mathcal{B}$ in the following way: the basic open sets of $\mathit{Runs}_{\mathcal{B}}$ are of the form $\pi \cdot (\mathit{Actions}_{\mathcal{B}} \times \mathit{States}_{\mathcal{B}})^\omega$, where $\pi \in \mathit{FPath}_{\mathcal{B}}$, and the measure of this open set is equal to $\prod_{i=0}^{|\pi|-1} \sigma(\pi(i))(\pi_{A(i+1)}) \cdot \mathit{Prob}_{\mathcal{B}}(\pi_{S(i)}, \pi_{A(i+1)})(\pi_{S(i+1)})$ if $\pi_{S(0)} = \alpha$ and equal to 0 otherwise. It is a classical result of measure theory that this extends to a unique measure over all Borel subsets of $\mathit{Runs}_{\mathcal{B}}$ and we will denote this measure by $P^{\sigma}_{\mathcal{B},\alpha}$.

Let $X : \mathit{Runs}_{\mathcal{B}} \to \mathbb{R}^{\infty}_{\oplus}$ be an arbitrary measurable function in respect to $P^{\sigma}_{\mathcal{B},\alpha}$. The expected value of X under strategy σ when starting at α is defined as $\mathbb{E}^{\sigma}_{\mathcal{B},\alpha}\{X\} = \int_{\mathit{Runs}_{\mathcal{B}}} X \, \mathrm{d}P^{\sigma}_{\mathcal{B},\alpha}$ (which can be ∞ even if $P^{\sigma}_{\mathcal{B},\alpha}(X = \infty) = 0$). The supremum expected value of X in $\mathcal{B}$ when starting at α is defined as $\mathcal{V}^*(\alpha)(X) = \sup_{\sigma \in \Sigma_{\mathcal{B}}} \mathbb{E}^{\sigma}_{\mathcal{B},\alpha}\{X\}$. A strategy, $\widehat{\sigma}$, is said to be optimal if $\mathbb{E}^{\widehat{\sigma}}_{\mathcal{B},\alpha}\{X\} = \mathcal{V}^*(\alpha)(X)$ and ε-optimal if $\mathbb{E}^{\widehat{\sigma}}_{\mathcal{B},\alpha}\{X\} \ge \mathcal{V}^*(\alpha)(X) - \varepsilon$. It is common for the objective of the controller to be optimisation of the probability that a run of $\mathcal{B}$ belongs to a given Borel set B. In such a case, we can just use the indicator function $\chi_B(r) = 1$ if $r \in B$ and $= 0$ otherwise, as the function X.

D. Clocks, clock valuations, regions and zones.

We fix a constant $K \in \mathbb{N}$ and finite set of *clocks* $\mathcal{C}$. A (K-bounded) *clock valuation* is a function $\nu : \mathcal{C} \to [\![K]\!]_{\mathbb{R}}$ and we write V for the set of clock valuations.

Assumption 1: Although clocks are usually allowed to take arbitrary non-negative values, we have restricted their values to be bounded by the constant K. This restriction is for technical convenience and comes without significant loss of generality.

If $\nu \in V$ and $t \in \mathbb{R}_{\oplus}$, we write $\nu{+}t$ for the clock valuation defined by $(\nu{+}t)(c) = \nu(c){+}t$ if $\nu(c){+}t \le K$ and $(\nu{+}t)(c) = K$ otherwise, for all $c \in \mathcal{C}$. Let $\Theta(\mathcal{C})$ be the set of all functions from $\mathcal{C}$ to $(\mathcal{C} \cup \{0\})$. For a function $\theta \in \Theta(\mathcal{C})$, by $\nu\langle\theta\rangle$ we denote the clock valuation ν' such that $\nu'(c) = \nu(\theta(c))$ if $\theta(c) \ne 0$ and $\nu'(c) = 0$ otherwise.

The set of *clock constraints* over $\mathcal{C}$ is the set of conjunctions of *simple constraints*, which are constraints of the form $c \bowtie i$ or $c{-}c' \bowtie i$, where $c, c' \in \mathcal{C}$, $i \in [\![K]\!]_{\mathbb{N}}$, and $\bowtie \in \{<, >, =, \leq, \geq\}$. For every $\nu \in V$, let $\mathrm{SCC}(\nu)$ be the finite set of simple constraints which hold in ν.

A *clock region* is a maximal set $\zeta \subseteq V$, such that $\mathrm{SCC}(\nu){=}\mathrm{SCC}(\nu')$ for all $\nu, \nu' \in \zeta$. We write $\mathcal{R}$ for the set of clock regions. Every clock region is an equivalence class of the indistinguishability-by-clock-constraints relation, and vice versa. Note that ν and ν' are in the same clock region if and only if the integer parts of the clocks and the partial orders of the clocks, determined by their fractional parts, are the same in ν and ν'. We write $[\nu]$ for the clock region of ν and, if $\zeta{=}[\nu]$, write $\zeta\langle\theta\rangle$ for the clock region $[\nu\langle\theta\rangle]$.

A *clock zone* is a convex set of clock valuations, which is a union of a set of clock regions. We write $\mathcal{Z}$ for the set of clock zones. For any clock zone W and clock valuation ν, we use the notation $\nu \in W$ to denote that $[\nu] \in W$. A set of clock valuations is a clock zone if and only if it is definable by a clock constraint. For $W \subseteq V$, we write $\overline{W}$ for the smallest closed set in V containing W. Observe that, for every clock zone W, the set $\overline{W}$ is also a clock zone.

III. Timed Branching Processes

Now we are in the position to formally introduce *timed branching processes*.

Definition 4 (Timed Branching Processes): A *(multitype) Timed Branching Process* (TBP) is a tuple $\mathcal{T} = (\mathcal{C}, Q, Act, Inv, E, \delta, r)$, where:

- $\mathcal{C}$ is a finite set of bounded clocks,
- $Q = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_n\}$ is a finite set of types;
- Act is a finite set of *actions*;
- $Inv : Q \to \mathcal{Z}$ is the *invariant condition*;
- $E : Q{\times}Act \to \mathcal{Z}$ is the *action enabledness function*;
- $\delta : Q{\times}Act \to \mathcal{D}(\Theta(\mathcal{C}){\times}Q^*)$ is the *transition probability function*; and
- $r : Q \cup (Q \times Act) \to \mathbb{R}_{\oplus}$ is the *reward function*.

When we consider a TBP as an input of an algorithm, its size should be understood as the sum of the sizes of encodings of Q, $\mathcal{C}$, Inv, Act, E, and δ. For computational purposes, we assume that all probabilities are rational, and their numerators and denominators are all written in binary.

Informally, the behaviour of a TBP is as follows. A configuration of a TBP, just like for BDPs, is a list of entities, α, and time passes before an available action is triggered, after which a discrete probabilistic transition occurs. Each entity is equipped with a finite set of private clocks $\mathcal{C}$. The state of each entity can be represented as a tuple $s = (q, \nu) \in Q \times V$, where q is the type of this entity and ν the valuation of its private clocks. As noted before, the values of all the clocks are bounded from above by a constant K.[3] For each such a tuple, $s = (q, \nu)$, we let $\mathit{Type}(s) = q$ and $\mathit{Valn}(s) = \nu$. During the delay, d, the reward obtained is the sum of the rewards for each entity α_i multiplied by the amount of time that has passed, i.e. $d \cdot \sum_i r(\mathit{Type}(\alpha_i))$, plus the sum of all the rewards for the actions performed at that time. Time passage is available only if for all entities α_i in α the invariant condition $Inv(\mathit{Type}(\alpha_i))$ is satisfied while time elapses, and an action a can be performed on entity α_i after time d elapses only if it is enabled after time elapse, i.e. if $\nu_i{+}d \in E(\alpha_i, a)$. Both the time and the action are chosen nondeterministically. If the action a is chosen to be performed on an entity of type q, the set of new entities, β, and the rearrangement of their parent's clocks, θ, is chosen with a fixed probability $\delta[q, a](\theta, \beta)$. All entities on which an action was performed, are removed from the current configuration and replaced with the set of new entities they branched into (this set can be empty). The valuation of the private clocks of the new entities in β is set to the valuation of the private clocks of their parent at the moment the branching took place after the rearrangement and reset operator θ was applied to it. In general, a strategy can specify multiple actions to be performed simultaneously on any subset of the entities that are alive at a given moment of time.

There are in fact two possible ways of defining the semantics of TBPs. The just described behaviour corresponds to *synchronous semantics*, where for each moment of time we keep the list of all the entities alive at that time. (Of course it suffices to keep track of the configuration only for the time moments when a discrete action is performed on at least one entity.) Moreover, the private clocks of all the entities in such a model progress at the same pace and are advanced by the same amount equal to the time delay between consecutive configurations. To give a better understanding of the synchronous semantics, we present here an example run of the TBP from Figure 1. In this example, $X(t, p, g)$ denotes an entity of type X whose age, its parent's age age and its grandparent's age are equal to t, p, g, respectively. Moreover, $\to_{d=x}$ represents a time delay of x time units, and $\to^{X,Y}$ corresponds to performing discrete action X on the first entity and action Y on the second one. If, e.g. $X = \epsilon$ then no action is performed on the first entity at that moment of time. An example run that starts with the initial population of one harmless bacterium can look as follows: $W(0,0,0) \to_{d=4} W(4,4,4) \to^{B} B(0,4,4), W(0,4,4) \to_{d=2} B(2,6,6), W(2,6,6) \to^{\epsilon,A} B(2,6,6), B(0,2,6) \to_{d=1} B(3,7,7), B(1,3,7) \to_{A,\epsilon} \epsilon \cdot B(1,3,7) \to_{d=3} B(4,6,10) \to^{A} \epsilon$ and the process stays in ϵ from that point on. Assuming that the strategy used by the the controller was pure, the probability of this particular run occurring is $\frac{3}{4}$.

Using the synchronous semantics it is easy to keep track of the size of the population and its age distribution at any moment of time, so for instance it is easy to express formally a property that the population becomes extinct before time T. Moreover, this semantics would be necessary if the objective of the controller was, e.g. optimisation of the number of different types of entities at a given moment of time. However, for the objectives studied in this paper, this is not necessary and simpler *asynchronous semantics* is sufficient. This alternative approach is to allow the entities to perform time delays

[3] This assumption is very natural for models of biological systems. On the other hand, it can be shown that TBPs with unbounded clock valuations can be analysed by splitting each type into exponentially many, in the number of clocks, new types.

of unequal lengths. In this model, the entities at the i-th consecutive configuration of the system are exactly the i-th generation of the branching process, i.e. all of these entities have exactly $i-1$ ancestors. Because of the differences in the lifespan of their ancestors, such a representation of the state of the system can lead to a situation where two entities are in the same configuration, although one of them will die before the other one was born. However, as we will prove later, for any TBP and any of the objectives studied in this paper there exists a static and memoryless strategy that is (ε-)optimal. This means that for any entity at any given time there exists an optimal action that does not depend on the future nor the past of any other entity in the system (and in fact it does not even depend on the past of this entity). Hence, the optimal value of any of these objectives, when starting with the same initial population, is the same in both semantics. At the same time, the asynchronous semantics has a big advantage over the synchronous semantics, because it allows us to represent the semantics of a TBP as a BDP (instead of MDP), which simplifies greatly the notation. Furthermore, we will be able to apply region abstraction and boundary region abstraction directly to this BDP in order to obtain a finite BDP (with the number of types being exponential in the size of the original TBP). This allow us to use directly all the computational complexity results established for finite BDPs, which streamlines the proofs. To give a better understanding of the asynchronous semantics, we present now a run of the TBP from Figure 1, which directly corresponds to the run we gave for the synchronous semantics. Here $\rightarrow_{d=x,d=y}$ represents a time delay of x time unit for the first entity and y time units for the second one. $W(0,0,0) \rightarrow_{d=4} W(4,4,4) \rightarrow^{B} B(0,4,4), W(0,4,4) \rightarrow_{d=3,d=2} B(3,7,7), W(2,6,6) \rightarrow^{A,A} \epsilon \cdot B(0,2,6) \rightarrow_{d=4} B(4,6,10) \rightarrow^{A} \epsilon$ and the process stays in ϵ from that point on. Again, assuming that the strategy used was pure, the probability of this particular run occurring is exactly the same as before, i.e. $\frac{3}{4}$.

Now formally, the (asynchronous) semantics of a TBP is given by a BDP which has both an (uncountably) infinite number of states and an (uncountably) infinite number of transitions. An action in this semantics will be represented as a pair $a = (d, c) \in \mathbb{R}_{\oplus} \times Act$, where d is the time delay to be performed before executing action c on this entity. For an action $a = (d, c) \in \mathbb{R}_{\oplus} \times Act$, we let $\mathit{Delay}(a) = d$ and $\mathit{Action}(a) = c$.

Definition 5 (Timed Branching Processes (Semantics)): Let $\mathcal{T} = (\mathcal{C}, Q, Act, Inv, E, \delta, r)$ be a TBP. The semantics of $\mathcal{T}$ is the BDP $[\![\mathcal{T}]\!] = (Q', A', p', r')$ where

- $Q' \subseteq Q \times V$ is a set of types such that for any $s \in Q'$ we have $\mathit{Valn}(s) \in Inv(\mathit{Type}(s))$;
- $A' = \mathbb{R}_{\oplus} \times Act$ is the set of *timed actions*;
- $p' : Q' \times A' \rightarrow \mathcal{D}({Q'}^{*})$ is the probabilistic transition function such that $p'(s, a) = \mu$ if and only if the following holds: $\nu{+}d \in Inv(\mathit{Type}(s))$ for all $d \in [0, \mathit{Delay}(a)]$; $(\nu{+}\mathit{Delay}(a)) \in E(\mathit{Type}(s), \mathit{Action}(a))$; and all $\beta \in Q^{*}$:

$$\mu(\beta \otimes \nu') = \sum_{\substack{\theta \in \Theta(\mathcal{C}) \text{ s.t.} \\ \nu' = (\nu + \mathit{Delay}(a))\langle\theta\rangle}} \delta[\mathit{Type}(s), \mathit{Action}(a)](\theta, \beta);$$

- $r' : Q' \times A' \rightarrow \mathbb{R}_{\oplus}$ is the reward function where

$$r'(s,a) = r(\mathit{Type}(s)) \cdot \mathit{Delay}(a) + r(\mathit{Type}(s), \mathit{Action}(s)).$$

IV. Abstractions of Timed Branching Processes

In this section we generalise the notion of Region Abstraction [12] and Boundary Region Abstraction [15] to the context of TBPs, and discuss properties of these abstractions.

A. Region Abstraction

Let us again fix a finite set of clocks $\mathcal{C}$ and an upper bound on their value $K \in \mathbb{N}$; this defines the set of possible valuations of clocks, V. For $\zeta, \zeta' \in \mathcal{R}$, where $\mathcal{R}$ is the set of all regions of V, we say that clock region ζ' is in the future of clock region ζ, or that ζ is in the past of ζ', if there are $\nu \in \zeta$, $\nu' \in \zeta'$ and delay $d \in \mathbb{R}_{\oplus}$ such that $\nu' = \nu{+}d$; we then write $\zeta \rightarrow_{*} \zeta'$. We say that ζ' is the *time successor* of ζ if $\zeta \rightarrow_{*} \zeta'$, $\zeta{\neq}\zeta'$, and $\zeta \rightarrow_{*} \zeta'' \rightarrow_{*} \zeta'$ implies $\zeta''{=}\zeta$ or $\zeta''{=}\zeta'$ and write $\zeta \rightarrow \zeta'$ and $\zeta' \leftarrow \zeta$. Time successor definition is extended to n-th time successor in a natural way: we say that ζ' is the n-th successor of ζ, and write $\zeta \rightarrow_{+n} \zeta'$, if there is a sequence of regions $\langle \zeta_1, \zeta_2, \ldots, \zeta_n \rangle$ such that $\zeta_1{=}\zeta$, $\zeta_n{=}\zeta'$ and $\zeta_i \rightarrow \zeta_{i+1}$ for every $1{\leq}i{<}n$.

Definition 6: (Region Abstraction of a TBP) Let $\mathcal{T} = (\mathcal{C}, Q, Act, Inv, E, \delta, r)$ be a TBP. The region abstraction of $\mathcal{T}$ is the finite BDP $\mathcal{T}_{\text{RA}} = (Q_{\text{RA}}, A_{\text{RA}}, p_{\text{RA}}, r_{\text{RA}})$ where:

- $Q_{\text{RA}} \subseteq Q \times \mathcal{R}$ is a set of types such that for any $(q, \zeta) \in Q_{\text{RA}}$ we have $\zeta \subseteq Inv(q)$;
- $A_{\text{RA}} \subseteq \mathbb{N} \times Act$ is the set of actions, such that if $(n, a) \in A_{\text{RA}}$ then $n \leq (2 \cdot |\mathcal{C}|)^{K}$ where K is the upper bound on the value of clocks;
- $p_{\text{RA}} : Q_{\text{RA}} \times A_{\text{RA}} \rightarrow \mathcal{D}({Q_{\text{RA}}}^{*})$ is the probabilistic transition function such that for $(q, \zeta) \in Q_{\text{RA}}$ and $(n, a) \in A_{\text{RA}}$ we have $p_{\text{RA}}((q, \zeta), (n, a)) = \mu$ if and only if the following holds: $\zeta \rightarrow_{+n} \zeta_n$; $\zeta_n \subseteq E(q, a)$; and for all $\beta \in Q^{*}$:

$$\mu(\beta \otimes \zeta') = \sum_{\theta \in \Theta(\mathcal{C}) \wedge \zeta_n\langle\theta\rangle = \zeta'} \delta[q, a](\theta, \beta).$$

- the reward function $r_{\text{RA}} : Q_{\text{RA}} \times A_{\text{RA}} \rightarrow \mathbb{N}$ is such that $r_{R}((q, \zeta), (n, a)) = n$.

From the results of Alur and Dill [16] the following proposition follows.

Proposition 1: . The size of $\mathcal{T}_{\text{RA}}$ is exponential in the size of $\mathcal{T}$.

B. Boundary Region Abstraction

We say that a clock region ζ is *thin* if $[\nu] \neq [\nu{+}\varepsilon]$ for every $\nu \in \zeta$ and $\varepsilon{>}0$ and *thick* otherwise. We write $\mathcal{R}_{\text{Thin}}$ and $\mathcal{R}_{\text{Thick}}$ for the sets of thin and thick regions, respectively. Observe that if $\zeta \in \mathcal{R}_{\text{Thick}}$ then, for any $\nu \in \zeta$, there exists $\varepsilon > 0$, such that $[\nu] = [\nu{+}\varepsilon]$ and the time successor of a thin region is thick, and vice versa.

For any $\nu \in V$, $b \in [\![K]\!]_{\mathbb{N}}$ and $c \in \mathcal{C}$ we define $\mathsf{time}(\nu, (b, c)) \stackrel{\text{def}}{=} b{-}\nu(c)$ if $\nu(c){\leq}b$, and $\mathsf{time}(\nu, (b, c)) \stackrel{\text{def}}{=} 0$ if $\nu(c){>}b$. Intuitively, $\mathsf{time}(\nu, (b, c))$ returns the amount of time that must elapse in ν before the clock c reaches the integer value b. Observe that, for every $\zeta' \in \mathcal{R}_{\text{Thin}}$, there exists $b \in \llbracket K \rrbracket_{\mathbb{N}}$ and $c \in \mathcal{C}$, such that, for every $\zeta \in \mathcal{R}$ in the past of ζ', we have that $\nu \in \zeta$ implies $\nu{+}(b{-}\nu(c)) \in \zeta'$; and we write $\zeta \rightarrow_{b,c} \zeta'$. We say ν is in the *closure* of ζ, if $\nu \in \overline{\zeta}$.

The boundary region abstraction is motivated by the following. Consider an action $a \in Act$, an entity (q, ν), and $\zeta \rightarrow_* \zeta'$ such that $\nu \in \zeta$ and action q is enable in clock region of ζ', i.e. $\zeta' \subseteq E(q, a)$.

- If $\zeta' \in \mathcal{R}_{\text{Thick}}$, then there are infinitely many $d \in \mathbb{R}_{\oplus}$ such that $\nu{+}d \in \zeta'$. However, amongst all such d's, for one of the boundaries of ζ', the closer $\nu{+}d$ is to this boundary, the 'better' the timed action (d, a) becomes for the controller's objective. However, since ζ' is a thick region, the set $\{d \in \mathbb{R}_{\oplus} \mid \nu{+}d \in \zeta'\}$ is an open interval, and hence does not contain its boundary values. Observe that the infimum equals $b_{\mathsf{inf}}{-}\nu(c_{\mathsf{inf}})$ where $\zeta \rightarrow_{b_{\mathsf{inf}}, c_{\mathsf{inf}}} \zeta_{\mathsf{inf}} \rightarrow \zeta'$ and the supremum equals $b_{\mathsf{sup}}{-}\nu(c_{\mathsf{sup}})$ where $\zeta \rightarrow_{b_{\mathsf{sup}}, c_{\mathsf{sup}}} \zeta_{\mathsf{sup}} \leftarrow \zeta'$. In the boundary region abstraction we include these 'best' timed actions through the actions $(b_{\mathsf{inf}}, c_{\mathsf{inf}}, a, \zeta')$ and $(b_{\mathsf{sup}}, c_{\mathsf{sup}}, a, \zeta')$.
- If $\zeta' \in \mathcal{R}_{\text{Thin}}$, then there exists a unique $d \in \mathbb{R}_{\oplus}$ such that $(\ell, \nu{+}d) \in \zeta'$. Moreover since ζ' is a thin region, there exists a clock $c \in C$ and a number $b \in \mathbb{N}$ such that $\zeta \rightarrow_{b,c} \zeta'$ and $d = b{-}\nu(c)$. In the boundary region abstraction we summarise this 'best' timed action from region ζ via region ζ' through the action (b, c, a, ζ').

Based on this intuition the boundary region abstraction (BRA) is defined as follows.

Definition 7: (Boundary Region Abstraction of a TBP) Let $\mathcal{T} = (\mathcal{C}, Q, Act, Inv, E, \delta, r)$ be a TBP. The boundary region abstraction of $\mathcal{T}$ is a (potentially infinite) BDP $\mathcal{T}_{\text{BRA}} = (Q_{\text{BRA}}, A_{\text{BRA}}, p_{\text{BRA}}, r_{\text{BRA}})$ where:

- $Q_{\text{BRA}} \subseteq Q \times V \times \mathcal{R}$ is a set of types such that for any $(q, \nu, \zeta) \in Q_{\text{BRA}}$ we have $\zeta \subseteq Inv(q)$ and $\nu \in \overline{\zeta}$.
- $A_{\text{BRA}} \subseteq \llbracket K \rrbracket_{\mathbb{N}} {\times} \mathcal{C} {\times} Act {\times} \mathcal{R}$ is the set of boundary timed actions;
- $p_{\text{BRA}} : Q_{\text{BRA}} \times A_{\text{BRA}} \rightarrow \mathcal{D}({Q_{\text{BRA}}}^*)$ is the probabilistic transition function such that for $(q, \nu, \zeta) \in Q_{\text{BRA}}$ and $(b, c, a, \zeta_a) \in A_{\text{BRA}}$ we have that $p_{\text{BRA}}((q, \nu, \zeta), (b, c, a, \zeta_a)) = \mu$ if and only if for all $\beta \in Q^*$:

$$\mu(\beta{\otimes}\nu'{\otimes}\zeta') = \sum_{\theta \in \Theta(\mathcal{C}) \wedge \nu_a\langle\theta\rangle = \nu' \wedge \zeta_a\langle\theta\rangle = \zeta'} \delta[\ell, a](\theta, \beta)$$

  where $\nu_a = \nu{+}\mathsf{time}(\nu, (b, c))$, $\zeta_a \subseteq E(q, a)$ and one of the following conditions holds:
    - $\zeta \rightarrow_{b,c} \zeta_a$
    - $\zeta \rightarrow_{b,c} \zeta_{\mathsf{inf}} \rightarrow \zeta_a$ for some ζ_{inf}
    - $\zeta \rightarrow_{b,c} \zeta_{\mathsf{sup}} \leftarrow \zeta_a$ for some ζ_{sup}.
- $r_{\text{BRA}}((q, t, \nu, \zeta), (b, c, a, \zeta_a)) = r(q) \cdot \mathsf{time}(\nu, (b, c)) + r(q, a)$.

Although the boundary region abstraction $\mathcal{T}_{\text{BRA}}$ is not a finite branching process, for any initial configuration $\alpha \in Q^*_{\text{BRA}}$ of finite length one can restrict attention to a finite branching process due to the following observation.

Lemma 2: In boundary region abstraction $\mathcal{T}_{\text{BRA}}$ of a TBP $\mathcal{T}$, the set of types of population originating from a fixed type $q \in Q_{\text{BRA}}$ is finite.

Proof: For a clock valuation ν we define its fractional signature $\lbag \nu \rbag$ to be the sequence $(f_0, f_1, \ldots, f_m)$, such that $f_0 = 0$, $f_i < f_j$ if $i < j$, for all $i, j \leq m$, and $f_1, f_2, \ldots, f_m$ are all the non-zero fractional parts of clock values in the clock valuation ν. In other words, for every $i \geq 1$, there is a clock c, such that $\lbag \nu(c) \rbag = f_i$, and for every clock $c \in \mathcal{C}$, there is $i \leq m$, such that $\lbag \nu(c) \rbag = f_i$. Let $(f_0, f_1, \ldots, f_m)$ be the fractional signature $\lbag \nu \rbag$.

For a nonnegative integer $k \leq m$ we define the k-shift of a fractional signature $(f_0, f_1, \ldots, f_m)$ as the fractional signature $(f'_k, f'_{k+1}, \ldots, f'_m, f'_0, \ldots, f'_{k-1})$ such that for all non-negative integers $i \leq m$ we have $f'_i = \lbag f_i + 1 - f_k \rbag$. We say that a fractional signature $(f'_0, f'_1, \ldots, f'_n)$ is a subsequence of another fractional signature $(f_0, f_1, \ldots, f_m)$ if $n \leq m$ and for all nonnegative integers $i < n$ we have $f'_i \leq f'_{i+1}$; and for every nonnegative integer $i \leq n$ there exists a nonnegative integer $j \leq m$ such that $f'_i = f_j$. Since taking boundary moves either result in a subsequence of fractional signature (potentially in the case of clock resets) or in a k-shift of the fractional signature, the following proposition is immediate.

Proposition 3: In a boundary region abstraction $\mathcal{T}_{\text{BRA}}$ if an entity $(q', \nu', \zeta') \in Q_{\text{BRA}}$ is reachable from a configuration $(q, \nu, \zeta) \in S_{\text{BRA}}$, then fractional signature of ν' is a k-shift of a subsequence of the fractional signature of ν.

The proof of the lemma follows from this proposition as the set Q and $\mathcal{R}$ are finite sets, and the set of k-shifts of subsequences of the fractional signatures of a valuation is finite. ■

Definition 8: (Boundary Region Abstraction for Fixed Initial Population) The reachable branching-process of a boundary region abstraction $\mathcal{T}_{\text{BRA}} = (Q_{\text{BRA}}, A_{\text{BRA}}, p_{\text{BRA}}, r_{\text{BRA}})$ from a configuration $\alpha \in Q^*_{\text{BRA}}$, is the finite BDP $\mathcal{T}^{\alpha}_{\text{BRA}} = (Q^{\alpha}_{\text{BRA}}, A_{\text{BRA}}, p_{\text{BRA}}, r_{\text{BRA}})$, where $Q^{\alpha}_{\text{BRA}} \subseteq Q_{\text{BRA}}$ is a finite set of types such that $(q, \nu, \zeta) \in Q_{\text{BRA}}$ if $\lbag \nu \rbag$ is a k-shift of a subsequence of $\lbag \nu_i \rbag$ where $\alpha_i = (q_i, \nu_i, \zeta_i)$ for some $i \leq |\alpha|$.

The following result is a slight modification of Proposition 1.

Proposition 4: The size of $\mathcal{T}^{\alpha}_{\text{BRA}}$ is exponential in the size of $\mathcal{T}$ and α.

V. Results

We omit subscripts $\mathcal{B}$, e.g. in *States*$_{\mathcal{B}}$, $\Sigma_{\mathcal{B}}$, etc., when the corresponding BDP is clear by context.

A. Expected Total Reward

For a given BDP $\mathcal{B}$ and $N \geq 0$ we define $\text{Total}_N(\pi)$, the cumulative reward of a run π after N steps, as $\text{Total}_N(\pi) = \sum_{i=0}^{N-1} Reward(\pi_{S(i)}, \pi_{A(i+1)})$. For a configuration $\alpha \in States$, and a strategy $\sigma \in \Sigma$, let $\text{ETotal}_N(\mathcal{B}, \alpha, \sigma)$ be the *N-step expected total reward* defined as $\text{ETotal}_N(\mathcal{B}, \alpha, \sigma) = \mathbb{E}^{\sigma}_{\mathcal{B},\alpha}\big\{\text{Total}_N\big\}$ and the *expected total reward* be $\text{ETotal}(\mathcal{B}, \alpha, \sigma) = \lim_{N\to\infty} \text{ETotal}_N(\mathcal{B}, \alpha, \sigma)$. This last value can potentially be ∞. It is straightforward to show that if σ is static and memoryless then for any $\alpha \in \mathit{States}$:

$$\mathrm{ETotal}_N(\mathcal{B},\alpha,\sigma)=\sum_{i=1}^{|\alpha|}\mathrm{ETotal}_N(\mathcal{B},\alpha_i,\sigma)$$

and by taking the limit as N goes to infinity we also get

$$\mathrm{ETotal}^*(\mathcal{B},\alpha,\sigma)=\sum_{i=1}^{|\alpha|}\mathrm{ETotal}^*(\mathcal{B},\alpha,\sigma).$$

For each starting state α, we would like to compute the *optimal expected reward* over all strategies of a TBP starting at α, denoted by $\mathrm{ETotal}^*(\mathcal{B},\alpha)$, i.e. $\mathrm{ETotal}^*(\mathcal{B},\alpha) = \sup_{\sigma\in\Sigma_{\mathcal{B}}}\mathrm{ETotal}(\mathcal{B},\alpha,\sigma)$.

a) Optimality Equations: For a BDP $\mathcal{B}$ we define the following (uncountable) set of equations $\mathcal{E}_{\mathrm{ETot}}(\mathcal{B})$.

- $\mathcal{X}(\alpha)=0$ if $\alpha=\epsilon$, and
- $\mathcal{X}(\alpha)=\sup\limits_{\gamma\in \mathit{Actions}(\alpha)}\big\{\mathrm{Rwd}(\alpha,\gamma)+\sum\limits_{\beta\in \mathit{States}}\mathit{Prob}(\alpha,\gamma)(\beta)\cdot \mathcal{X}(\beta)\big\}$, otherwise. Here $\mathrm{Rwd}(\alpha,\gamma)$ is a shorthand for $\sum_{i=1}^{|\alpha|} r(\mathit{Type}(\alpha_i))\cdot \mathit{Delay}(\gamma_i)+r(\mathit{Type}(\alpha_i),\mathit{Action}(\gamma_i))$.

We can represent this equation system as $x=\Phi(x)$ where $x:\mathit{States}\to\mathbb{R}^{\infty}_{\oplus}$ and $\Phi:(\mathit{States}\to\mathbb{R}^{\infty}_{\oplus})\to(\mathit{States}\to\mathbb{R}^{\infty}_{\oplus})$ is an operator that is easily defined by the right-hand sides (RHS) of equation system $\mathcal{E}_{\mathrm{ETot}}(\mathcal{B})$.

- $\Phi(x)(\alpha)=0$ if $\alpha=\epsilon$, and
- $\Phi(x)(\alpha)=\sup\limits_{\gamma\in \mathit{Actions}(\alpha)}\big\{\mathrm{Rwd}(\alpha,\gamma)+\sum\limits_{\beta\in \mathit{States}}\mathit{Prob}(\alpha,\gamma)(\beta)\cdot x(\beta)\big\}$, otherwise.

We say that a function $x:\mathit{States}\to\mathbb{R}^{\infty}_{\oplus}$ is a solution of the optimality equations $\mathcal{E}_{\mathrm{ETot}}(\mathcal{B})$, and we write $x\models\mathcal{E}_{\mathrm{ETot}}(\mathcal{B})$, if x is a fixed point of Φ. We can define a partial order on these functions as follows: $x\le y$ iff $x(\alpha)\le y(\alpha)$ for all $\alpha\in\mathit{States}$. The bottom element in this order is $\mathbf{0}$, where $\mathbf{0}(\alpha)=0$ for all α. Now, from Knaster-Tarski theorem, we know that Φ has a least fixed point and in fact the following holds.

Theorem 5: If x^* is the least fixed point of Φ then $x^*(\alpha)=\mathrm{ETotal}^*(\mathcal{B},\alpha)$ for every $\alpha\in\mathit{States}$.

Furthermore, this least fixed point can be defined in a more constructive way.

Theorem 6: The least fixed point of Φ is equal to $\lim_{N\to\infty}\Phi^N(\mathbf{0})$, where $\Phi^N(\mathbf{0})$ is equal to the N-th application of the operator Φ to $\mathbf{0}$.

Using these two theorems we can show the following.

Corollary 7: For any $\alpha\in\mathit{States}$, $\mathrm{ETotal}^*(\mathcal{B},\alpha)=\sum_{i=1}^{|\alpha|}\mathrm{ETotal}^*(\mathcal{B},\alpha_i)$.

b) Finite BDPs.: Although [11] does not deal with BDPs directly, rather with so-called 1-exit Recursive Simple Stochastic Games with positive rewards, there is a tight connection between these two models. As pointed out in that paper, the results presented there hold for Branching Markov Decision Processes with nonnegative rewards, which is essentially the same model as BDPs with the expected total reward objective studied in this section.

Theorem 8 (Expected Total Reward [11]): The supremum expected total reward for a finite BDP $\mathcal{B}$ can be computed exactly (this value is always rational or ∞) in PTIME, and there always exists an optimal static and memoryless strategy.

c) Timed Branching Processes.: Let $\mathcal{T}$ be a TBP and $\mathcal{T}_{\text{BRA}}$ be its boundary region abstraction. To map a state of a TBP to the corresponding state in the boundary region graph we define the operator $(\!|\cdot|\!)$ as follows: for $\alpha\in\mathit{States}_{[\![\mathcal{T}]\!]}$ we define $(\!|\alpha|\!)\in\mathit{States}_{\mathcal{T}_{\text{BRA}}}$ to be the configuration such that $(\!|\alpha|\!)_i=(\mathit{Type}(\alpha_i),\mathit{Valn}(\alpha_i),[\mathit{Valn}(\alpha_i)])$.

We show in Theorem 9 that the problem of computing expected total reward of a TBP $\mathcal{T}$ can be reduced to a similar problem on the corresponding boundary region abstraction. Although boundary region graph $\mathcal{T}_{\text{BRA}}$ is not a finite BDP, we showed in Lemma 2 that for a population $\alpha\in\mathit{States}$ of fixed (finite) size the resulting BDP $\mathcal{T}^{\alpha}_{\text{BRA}}$ has finitely many types.

Theorem 9: For every configuration $\alpha\in\mathit{States}_{[\![\mathcal{T}]\!]}$ we have that $\mathrm{ETotal}^*([\![\mathcal{T}]\!],\alpha)=\mathrm{ETotal}^*(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!))$. Moreover, for an arbitrary $\varepsilon>0$, we can construct an ε-optimal static and memoryless strategy for TBP $[\![\mathcal{T}]\!]$ from any optimal static and memoryless strategy for its boundary region abstraction $\mathcal{T}_{\text{BRA}}$.

Proof (sketch): In order to prove $\mathrm{ETotal}^*([\![\mathcal{T}]\!],\alpha)=\mathrm{ETotal}^*(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!))$, we show that for the operators $\Phi_{[\![\mathcal{T}]\!]}$ and $\Phi_{\mathcal{T}_{\text{BRA}}}$, corresponding to BDPs $[\![\mathcal{T}]\!]$ and $\mathcal{T}_{\text{BRA}}$, respectively, $\Phi^N_{[\![\mathcal{T}]\!]}(\mathbf{0})(\alpha)=\Phi^N_{\mathcal{T}_{\text{BRA}}}(\mathbf{0})((\!|\alpha|\!))$ holds for all N. First, we prove it for a single entity $\alpha=(q,\nu)$ and then use the fact that $\Phi^N_{[\![\mathcal{T}]\!]}(\mathbf{0})(\alpha)=\sum_i\Phi^N_{\mathcal{T}_{\text{BRA}}}(\mathbf{0})(\alpha_i)$ and $\Phi^N_{[\![\mathcal{T}]\!]}(\mathbf{0})(\alpha)=\sum_i\Phi^N_{\mathcal{T}_{\text{BRA}}}(\mathbf{0})(\alpha_i)$ holds all N and all α; these proofs are by induction on N. Finally, $\mathrm{ETotal}^*([\![\mathcal{T}]\!],\alpha)=\mathrm{ETotal}^*(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!))$ follows as a direct consequence of Theorems 5 and 6.

For the second part, let us recall that a static and memoryless strategy can essentially be interpreted as a function, $\sigma:Q\to A$, from types to actions. From Theorem 8 we know that there exists an optimal static and memoryless strategy in $\mathcal{T}_{\text{BRA}}$ and let $\sigma_{\mathcal{T}_{\text{BRA}}}:Q_{\text{BRA}}\to A_{\text{BRA}}$ be such a strategy. Because $\sigma_{\mathcal{T}_{\text{BRA}}}$ is optimal we know that $\mathrm{ETotal}(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!),\sigma_{\mathcal{T}_{\text{BRA}}})=\mathrm{ETotal}^*(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!))=\mathrm{ETotal}^*([\![\mathcal{T}]\!],\alpha)$. Let $\eta>0$ be a constant to be fixed later. We define a corresponding to $\sigma_{\mathcal{T}_{\text{BRA}}}$ static and memoryless strategy $\sigma^{\eta}_{[\![\mathcal{T}]\!]}:(Q\times V)\to(\mathbb{R}_{\oplus}\times \mathit{Act})$ in $[\![\mathcal{T}]\!]$ as follows: $\sigma^{\eta}_{[\![\mathcal{T}]\!]}(q,\nu)=(t,a)$ if $\sigma_{\mathcal{T}_{\text{BRA}}}(q,\nu,[\nu])=(b,c,a,\zeta)$ and t is such that $\nu+t\in\zeta$ and

$$r(q)\cdot t-r(q,a)\ge(1-\eta)\cdot\big(r(q)\cdot\mathsf{time}(\nu,(b,c))-r(q,a)\big)\quad(1)$$

Such a t can always be chosen. Observe that there are many such strategies for every $\sigma_{\mathcal{T}_{\text{BRA}}}$ and η, and let $\sigma^{\eta}_{[\![\mathcal{T}]\!]}$ be one of them. We first prove by induction on N using inequality (1) that for a single entity $\alpha=(q,\nu)$ we have $\mathrm{ETotal}_N([\![\mathcal{T}]\!],\alpha,\sigma^{\eta}_{[\![\mathcal{T}]\!]})\ge(1-\eta)\cdot\mathrm{ETotal}_N(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!),\sigma_{\mathcal{T}_{\text{BRA}}})$. For general $\alpha\in\mathit{States}$ we can do the following: $\mathrm{ETotal}_N([\![\mathcal{T}]\!],\alpha,\sigma^{\eta}_{[\![\mathcal{T}]\!]})=\sum_{i=0}^{|\alpha|}\mathrm{ETotal}_N([\![\mathcal{T}]\!],\alpha_i,\sigma^{\eta}_{[\![\mathcal{T}]\!]})\ge\sum_{i=0}^{|\alpha|}(1-\eta)\cdot\mathrm{ETotal}_N(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!)_i,\sigma_{\mathcal{T}_{\text{BRA}}})=(1-\eta)\cdot\mathrm{ETotal}_N(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!),\sigma_{\mathcal{T}_{\text{BRA}}})$. Taking the limit as N tends to ∞ we get that

$$\mathrm{ETotal}([\![\mathcal{T}]\!],\alpha,\sigma^{\eta}_{[\![\mathcal{T}]\!]})\ge(1-\eta)\cdot\mathrm{ETotal}(\mathcal{T}_{\text{BRA}},(\!|\alpha|\!),\sigma_{\mathcal{T}_{\text{BRA}}}).$$

Now, if $\mathrm{ETotal}^*([\![\mathcal{T}]\!],\alpha)=0$ or $\mathrm{ETotal}^*([\![\mathcal{T}]\!],\alpha)=\infty$ then by setting, e.g. $\eta=\frac{1}{2}$ we get that $\mathrm{ETotal}([\![\mathcal{T}]\!],\alpha,\sigma^{\eta}_{[\![\mathcal{T}]\!]})\ge 0$ in the former and $\mathrm{ETotal}([\![\mathcal{T}]\!],\alpha,\sigma^{\eta}_{[\![\mathcal{T}]\!]})\ge\infty$ in the latter, so in fact we obtain an optimal strategy $\sigma^{\eta}_{[\![\mathcal{T}]\!]}$ in both of these cases. Otherwise, if $\mathrm{ETotal}^*([\![\mathcal{T}]\!],\alpha)$ is finite and strictly positive, we set $\eta=\varepsilon/\mathrm{ETotal}^*([\![\mathcal{T}]\!],\alpha)$ and obtain that

$\mathrm{ETotal}([\![\mathcal{T}]\!], \alpha, \sigma^{\eta}_{[\![\mathcal{T}]\!]}) \geq (1-\eta) \cdot \mathrm{ETotal}(\mathcal{T}_{\text{BRA}}, (\!|\alpha|\!), \sigma_{\mathcal{T}_{\text{BRA}}}) = (1 - \varepsilon / \mathrm{ETotal}^*([\![\mathcal{T}]\!], \alpha)) \cdot \mathrm{ETotal}^*([\![\mathcal{T}]\!], \alpha) = \mathrm{ETotal}^*([\![\mathcal{T}]\!], \alpha) - \varepsilon$. Hence, $\sigma^{\eta}_{[\![\mathcal{T}]\!]}$ is ε-optimal. ■

B. Extinction Probability

For a given BDP $\mathcal{B}$ and $N \geq 0$, we define $\mathrm{XP}_N(\pi)$, the indictor function of extinction within N steps of a run π as

$$\mathrm{XP}_N(\pi) = \begin{cases} 1 & \text{if } \pi_{S(i)} = \epsilon \text{ for some } i \leq N \\ 0 & \text{otherwise} \end{cases}$$

For a configuration $\alpha \in \mathit{States}$, and a strategy $\sigma \in \Sigma$, let $\mathrm{XP}_N(\mathcal{B}, \alpha, \sigma)$ be the *N-step extinction probability* defined as $\mathrm{XP}_N(\mathcal{B}, \alpha, \sigma) = \mathbb{E}^{\sigma}_{\mathcal{B},\alpha}\{\mathrm{XP}_N\}$ and the *extinction probability* be defined as $\mathrm{XP}(\mathcal{B}, \alpha, \sigma) = \lim_{N\to\infty} \mathrm{XP}_N(\mathcal{B}, \alpha, \sigma)$. For each starting state α, we would like to compute the *optimal extinction probability* over all strategies of a TBP starting at α, denoted by $\mathrm{XP}^*(\mathcal{B}, \alpha)$, i.e. $\mathrm{XP}^*(\mathcal{B}, \alpha) = \sup_{\sigma \in \Sigma_{\mathcal{B}}} \mathrm{XP}(\mathcal{B}, \alpha, \sigma)$.

a) Optimality Equations: For a BDP $\mathcal{B}$ we define the following (uncountable) set of equations $\mathcal{E}_{\mathrm{XP}}(\mathcal{B})$:

- $\mathcal{X}(\alpha) = 1$ if $\alpha = \epsilon$, and
- $\mathcal{X}(\alpha) = \sup\limits_{\gamma \in \mathit{Actions}(\alpha)} \sum\limits_{\beta \in \mathit{States}} \mathit{Prob}(\alpha, \gamma)(\beta) \cdot \mathcal{X}(\beta)$, otherwise.

We can represent this equation system as $x = \Psi(x)$ where $x : \mathit{States} \to \mathbb{R}^{\infty}_{\oplus}$ and $\Psi : (\mathit{States} \to \mathbb{R}^{\infty}_{\oplus}) \to (\mathit{States} \to \mathbb{R}^{\infty}_{\oplus})$ is an operator that is easily defined by the right-hand sides (RHS) of equation system $\mathcal{E}_{\mathrm{XP}}(\mathcal{B})$.

- $\Psi(x)(\alpha) = 1$ if $\alpha = \epsilon$, and
- $\Psi(x)(\alpha) = \sup\limits_{\gamma \in \mathit{Actions}(\alpha)} \big\{ \sum\limits_{\beta \in \mathit{States}} \mathit{Prob}(\alpha, \gamma)(\beta) \cdot x(\beta) \big\}$, otherwise.

Just like for the expected total reward objective, we know that this equation system has a least fixed point and the following analogous theorems hold.

Theorem 10: If x^* is the least fixed point of Ψ then $x^*(\alpha) = \mathrm{XP}^*(\mathcal{B}, \alpha)$ for every $\alpha \in \mathit{States}$.

Theorem 11: The least fixed point of Ψ is equal to $\lim_{N\to\infty} \Psi^N(\mathbf{0})$.

Corollary 12: For any $\alpha \in \mathit{States}$ we have $\mathrm{XP}^*(\mathcal{B}, \alpha) = \prod_{i=1}^{|\alpha|} \mathrm{XP}^*(\mathcal{B}, \alpha_i)$.

b) Finite BDPs.:

Theorem 13 (Finite BDPs[9], [10]): The problem of deciding whether the supremum probability of extinction of a given population is $\geq p$ is in PSPACE. Deciding whether this value is 0, 1 or in between is in PTIME. Moreover, there always exists an optimal static and memoryless strategy that maximises the probability of extinction.

c) Timed Branching Processes.: Let $\mathcal{T}$ be a TBP and $\mathcal{T}_{\text{RA}}$ be its region abstraction. To map a state of a TBP to the corresponding state in the region graph we define the operator $[\cdot]$ as follows: for $\alpha \in \mathit{States}_{[\![\mathcal{T}]\!]}$ we define $[\alpha] \in \mathit{States}_{\mathcal{T}_{\text{RA}}}$ to be the configuration such that $[\alpha]_i = (\mathit{Type}(\alpha_i), [\mathit{Valn}(\alpha_i)])$.

We show in Theorem 14 that the problem of computing the extinction probability of a TBP $\mathcal{T}$ can be reduced to the same problem for its corresponding region abstraction (a finite BDP). This reduction is simpler than showing this for the boundary region abstraction that we used for the expected total reward objective (for which this simpler region abstraction does not work).

Theorem 14: For every configuration $\alpha \in \mathit{States}_{[\![\mathcal{T}]\!]}$ we have that $\mathrm{XP}^*([\![\mathcal{T}]\!], \alpha) = \mathrm{XP}^*(\mathcal{T}_{\text{RA}}, [\alpha])$. Moreover, an optimal static and memoryless strategy in $\mathcal{T}_{\text{RA}}$ gives an optimal static and memoryless strategy for its TBP $[\![\mathcal{T}]\!]$.

C. Time-Bounded Extinction Probability.

For a given TBP $\mathcal{T} = (\mathcal{C}, Q, \mathit{Act}, \mathit{Inv}, E, \delta, r)$, an initial configuration $\alpha \in \mathit{States}_{[\![\mathcal{T}]\!]}$ the time-bounded extinction probability is the supremum probability of extinction of the entire population within $T \in \mathbb{N}$ units of time. This problem is a variant of the extinction probability and can be solved by slightly modifying the TBP $\mathcal{T}$ to the TBP $\mathcal{T}' = (\mathcal{C}', Q', \mathit{Act}', \mathit{Inv}', E', \delta', r')$, where:

- $\mathcal{C}' = \mathcal{C} \cup \{c^\star\}$ (where $c^\star \notin \mathcal{C}$) is a finite set of bounded clocks with the bound $K' = \max\{K, T\}$;
- $Q' = Q \cup \{\star\}$ (where $\star \notin Q$) is a finite set of types;
- $\mathit{Act}' = \mathit{Act} \cup \{a^\star\}$ (where $a^\star \notin \mathit{Act}$) is a finite set of *actions*;
- $\mathit{Inv}' : Q' \to \mathcal{Z}'$ (here $\mathcal{Z}'$ is the set of zones on $\mathcal{C}'$) is the *invariant condition* such that
  - $\mathit{Inv}'(\Lambda) = \mathit{Inv}(\Lambda) \cap \{\nu(c^\star) \leq T\}$ if $\Lambda \neq \star$, and
  - $\mathit{Inv}'(\Lambda) = \emptyset$ otherwise;
- $E' : Q' \times \mathit{Act}' \to \mathcal{Z}'$ is the *action enabledness function* such that
  - $E'(\Lambda, a) = E(\Lambda, a)$ if $\Lambda \neq \star$ and $a \neq a^\star$,
  - $E'(\Lambda, a) = \{\nu : \nu(c^\star) = T\}$ if $a = a^\star$;
- $\delta' : Q' \times \mathit{Act}' \to \mathcal{D}(\Theta(\mathcal{C}) \times Q'^*)$ is the *transition probability function* such that
  - $\delta'(\Lambda, a) = \delta(\Lambda, a)$ if $\Lambda \neq \star$ and $a \neq a^\star$, otherwise
  - $\delta'(\Lambda, a)(\theta, \star) = 1$ s.t. $\theta \equiv \mathbf{0}$;

  and
- $r' : Q' \cup (Q' \times \mathit{Act}') \to \mathbb{R}_{\oplus}$ is the *reward function* such that:
  - $r'(\Lambda) = r(\Lambda)$ if $\Lambda \neq \star$, and $r'(\star) = 0$,
  - $r'(\Lambda, a) = r(\Lambda, a)$ if $\Lambda \neq \star$, and $a \neq a^\star$,
  - $r'(\Lambda, a) = 0$ otherwise.

The TBP $\mathcal{T}'$ is similar to $\mathcal{T}$, however it has one special clock $c^\star$ that is never reset. Every type in $\mathcal{T}'$ has invariant declaring that the value of clock $c^\star$ should not exceed T. Moreover $\mathcal{T}'$ has a degenerate type $\star$ which spawns itself after T time units with probability 1 while resetting all the clocks, and hence never goes extinct. Every type can make a transition to the degenerate type $\star$ when the value of clock $c^\star$ is equal to T. Hence after T time units, all the types present in the population can either reproduce within 0 time-units or make a transition to type $\star$. It is straightforward to verify that the supremum time-bounded extinction probability in $\mathcal{T}$ for some initial configuration is equal to supremum extinction probability in $\mathcal{T}'$.

D. Complexity results

We showed in the previous subsections how to reduce, using region abstraction or boundary region abstraction, the problem of computing the optimal probability of extinction, time-bounded probability of extinction and total expected reward for TBPs to the respective problems for finite-type BDPs with the number of types exponential in the size of the TBP. Furthermore, it follows from Corollaries 7 and 12, that it suffices to compute the optimal expected total reward and optimal probability of extinction for initial populations consisting of just a single entity, because the optimal value for any other initial configuration can be easily computed based on them. This gives us the following computational complexity upper bounds for the decision problems for TBPs.

Theorem 15: For a given TBP $\mathcal{T}$, its arbitrary entity with all clocks set to zero as the initial configuration and an arbitrary $p \in \mathbb{Q}$ the following holds. (We assume that the length of the input is $|\mathcal{T}|$ + the number of bits required to represent the numerator and denominator of p.)

1) There exists an optimal static and memoryless strategy that maximises the probability of extinction of the initial population. Deciding whether this optimal value is $\geq p$ is in EXPSPACE and whether it is 0, 1 or in between is in EXPTIME.
2) For the time-bounded probability of extinction with time bound T (given in binary) we have the following. We can transform TBP $\mathcal{T}$ in polynomial time into TBP $\mathcal{T}'$, such that the optimal probability of exaction in $\mathcal{T}'$ is the same as the optimal time-bounded probability of extinction in $\mathcal{T}$ with time bound T. The problem of deciding whether this optimal value is $\geq p$ is in EXPSPACE and whether it is 0, 1 or in between is in EXPTIME. Furthermore, there exists an optimal static and memoryless strategy in $\mathcal{T}'$ that maximises this value. (On the other hand, it can happen that all optimal strategies for this objective in $\mathcal{T}$ are not memoryless, but this is not important, because in the end we focus only on $\mathcal{T}'$.)
3) There exists an optimal static and memoryless strategy that maximises the total expected reward before extinction of the initial population. The optimal value can be computed exactly (or decided whether it is ∞) in EXPTIME.

Proof:

1) After the region abstraction is applied to $\mathcal{T}$, the claim follows from Theorem 14 and Theorem 13.
2) The construction given in Section V-C, transforms TBP $\mathcal{T}$ into TBP $\mathcal{T}'$ with one extra clock and T being the upper bound on this clock's value, for which the optimal probability of extinction of the initial population in $\mathcal{T}'$ is the same as the optimal time-bounded, with time bound T, probability of extinction of the same initial population in $\mathcal{T}$. Since the size of the region graph (which is a BDP) of $\mathcal{T}'$ is $\mathcal{O}(T)$ times larger than the region graph of $\mathcal{T}$, it follows again from Theorem 13 that deciding whether the optimal time-bounded probability of extinction with time bound T (given in binary) is $\geq p$ is in EXPSPACE.
3) After the boundary region abstraction is applied to $\mathcal{T}$, the claim follows from Theorem 9 and Theorem 8.

■

It follows from results [17] for Probabilistic Timed Automata that all these problems are EXPTIME-hard, hence in fact the last one is EXPTIME-complete. Moreover, it follows from [18] that if all entities have just one private clock, the computational complexity remains the same as for BDPs without any clocks.

Finally, in order to approximate within i bits of precision the value of the optimal (time-bounded) extinction probability for a given $\mathcal{T}$, initial population α (and time bound T), one can perform a binary search in the interval of all possible values of this quantity, i.e. [0,1]. First, we would ask whether the optimal value is $\geq \frac{1}{2}$, if it is smaller we would ask whether it is $\geq \frac{1}{4}$ and otherwise we would ask whether it is $\geq \frac{3}{4}$, etc. After i such questions, we get to know the i most significant bits of this optimal value.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which significantly helped to improve this paper. The authors are supported in part by EPSRC grants EP/F001096 and EP/G050112/1.

REFERENCES

[1] P. J. Patsy Haccou and V. A. Vatutin, *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambrdige University Press, 2005.

[2] A. Trivedi and D. Wojtczak, "Timed branching processes," *Oxford University Computing Laboratory technical report, RR-10-08*, 2010.

[3] H. W. Watson and F. Galton, *On the probability of the extinction of families*, 1874, vol. 4.

[4] A. N. Kolmogorov and B. A. Sevastyanov, "The calculation of final probabilities for branching random processes," *Doklady Akad. Nauk. U.S.S.R. (N.S.)*, vol. 56, pp. 783–786, 1947.

[5] T. E. Harris, *The Theory of branching processes*. Springer, Berlin, 1963.

[6] K. Etessami and M. Yannakakis, "Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations," *J. ACM*, vol. 56, no. 1, 2009.

[7] S. R. Pliska, "Optimization of multitype branching processes," *Management Science*, vol. 23, no. 2, pp. 117–124, 1976.

[8] U. G. Rothblum and P. Whittle, "Growth Optimality for Branching Markov Decision Chains," *Mathematics of Operations Research*, vol. 7, no. 4, pp. 582–601, 1982.

[9] K. Etessami and M. Yannakakis, "Recursive Markov decision processes and recursive stochastic games," in *Proc. of ICALP'05*, 2005, pp. 891–903, full version at: http://homepages.inf.ed.ac.uk/kousha/j_sub_rmdp_rssg.pdf.

[10] ——, "Efficient qualitative analysis of classes of recursive Markov decision processes and simple stochastic games," in *Proc. of STACS'06*. Springer, 2006.

[11] K. Etessami, D. Wojtczak, and M. Yannakakis, "Recursive stochastic games with positive rewards," in *Proc. of ICALP'08 (Track A)*, 2008, pp. 711–723.

[12] R. Alur and D. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[13] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston, "Automatic verification of real-time systems with discrete probability distributions," *Theoretical Computer Science*, vol. 282, 2002.

[14] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.

[15] M. Jurdziński and A. Trivedi, "Concavely-priced timed automata," in *Proc. of FORMATS'08*, ser. LNCS, vol. 5215. Springer, 2008.

[16] R. Alur and D. Dill, "Automata for modeling real-time systems," in *Proc. of ICALP'90*, ser. LNCS, vol. 443. Springer, 1990.

[17] M. Jurdziński, J. Sproston, and F. Laroussinie, "Model checking probabilistic timed automata with one or two clocks," *Logical Methods in Computer Science*, vol. 4, no. 3, 2008.

[18] F. Laroussinie, N. Markey, and P. Schnoebelen, "Model checking timed automata with one or two clocks," in *Proc. of CONCUR'04*, ser. LNCS, vol. 3170. Springer, 2004.