Extrapolation of an Optimal Policy using Statistical Probabilistic Model Checking

Artur Rataj[1] and Bożena Woźna-Szcześniak[2]

[1] IITiS, Polish Academy of Sciences, ul. Bałtycka 5, 44-100 Gliwice, Poland
arturrataj@gmail.com
[2] IMCS, Jan Długosz University
Al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland.
b.wozna@ajd.czest.pl

Abstract. We show how to extrapolate an optimal policy controlling a model, which is itself too large to find the policy directly using probabilistic model checking (PMC). In particular, we look for a global optimal resolution of non–determinism in several small Markov Decision Processes (MDP) using PMC. We then use the resolution to find a respective set of decision boundaries representing the optimal policies found. Then, a hypothesis is formed on an extrapolation of these boundaries to an equivalent boundary in a large MDP. The resulting hypothetical extrapolated decision boundary is statistically approximately verified, whether it indeed represents an optimal policy for the large MDP. The verification either weakens or strengthens the hypothesis. The criterion of the optimality of the policy can be expressed in any modal logic that includes the probabilistic operator $\mathtt{P}_{\sim p}[\cdot]$, and for which a PMC method exists.

Keywords: probabilistic model checking, statistical model checking, non–determinism, optimal policy, extrapolation.

1 Introduction

Probabilistic model checking (PMC) [4] refers to a range of techniques for a formal analysis of a stochastic system, which is usually a state transition system with transitions labelled by probability values.

A policy of a decision maker (an agent), controlling a Markov Decision Process (MDP), resolves a non–deterministic choice, which exist in each state of an MDP in the form of a number of probability distributions over states, of which one is arbitrarily chosen (for details see Sec. 2). An optimal policy [12], in respect to a given property, may in particular correspond to either the minimum or maximum value of the property. In this paper we consider an MDP with properties specified in any modal logic that includes the probabilistic operator $\mathtt{P}_{\sim p}[\cdot]$, for which exists a PMC method. A common example of such a logic, for which efficient model checkers exist, is *Probabilistic Computation Tree Logic* (PCTL) [3].

Statistical probabilistic model checking (SPMC) [13, 8], including Monte Carlo simulation and sampling, involves a generation of a large number of random paths in a stochastic model, evaluating a given property on each path, and finally statistically aggregating all these evaluations in order to approximate a correct value of a property.

We address the issue of an estimation of an optimal policy in a model, which is too large to have that policy found using PMC, and also too large to have that policy estimated using SPMC, if an initial, sufficiently precise approximation ζ of the policy is unknown. In order to obtain ζ, we first find a number of equivalent optimal policies in several scaled-down versions of the large model. Then, we pose a hypothesis on an extrapolation of these policies to the large model. Finally, we strengthen or weaken the hypothesis using SPMC, which approximately verifies, whether the extrapolated policy is optimal by checking whether the policy has the largest fitness, when compared to a number of its close variants.

In particular, we begin with searching for a global optimal resolution of non–determinism [7] in several small MDPs, that model smaller versions of the system we are interested in. We then estimate equations of decision boundaries, each representing one of the obtained optimal policies. A hypothesis is then formed on extrapolating the equations to a large MDP. The resulting hypothetical extrapolated decision boundary is finally approximately verified by estimating if its fitness is locally maximal using a Monte Carlo DTMC simulator.

The paper is constructed as follows. In Sec. 2 we define the formalism used. In Sec. 3 we propose a technique which extrapolates and verifies an optimal policy. In Sec. 4 we present a case study. In the last section we conclude the paper.

2 Preliminaries

Let us define the formalism used throughout the paper. It is fairly standard and follows [4], where the reader will find an in–depth description.

2.1 Discrete–Time Markov Chains

A *discrete-time Markov chain* (DTMC) consists of states that represent instantaneous snapshots of the system at a given time, and has transitions labelled by (discrete) probability distributions over the target states.

Definition 1. *A DTMC is a tuple* $\mathcal{D} = (S, s^\iota, T, \mathrm{AP}, L)$*, where* S *is a finite set of states,* s^ι *is the initial state,* $T : S \times S \to [0, 1]$ *is a transition probability function such that* $\sum_{s' \in S} T(s, s') = 1$ *for all* $s \in S$*,* AP *is a set of atomic propositions, and* $L : S \to 2^{\mathrm{AP}}$ *is a valuation function which assigns to every state* $s \in S$ *a set* $L(s)$ *of atomic propositions that are assumed to be true at that state.*

Observe that each transition represents the possibility to evolve from one state to another. Moreover, for a state $s \in S$ of $\mathcal{D}$, the probability of moving to a state $s' \in S$ in one discrete step is given by $T(s, s')$. Further, a *path* of $\mathcal{D}$ is an infinite sequence $\omega = s_0, s_1, s_2, \ldots$ of states such that $T(s_i, s_{i+1}) > 0$ for all $i \geq 0$. Each path of $\mathcal{D}$ provides one possible evolution of the Markov chain.

Properties of DTMCs can be written in *Probabilistic Computation Tree Logic* (PCTL) [3], a probabilistic extension of the temporal logic CTL [1].

Definition 2 (Syntax). *Let $a \in \mathrm{AP}$ be an atomic proposition, $p \in [0,1]$ a probability bound, $k \in \mathbb{N}$ and $\sim \in \{<, \leq, \geq, >\}$. The syntax of PCTL is defined inductively as follows:*

$$\phi ::= \mathit{true} \mid a \mid \neg\phi \mid \phi \wedge \phi \mid \mathtt{P}_{\sim p}[\psi], \quad \psi ::= \mathtt{X}\phi \mid \phi\,\mathtt{U}\phi \mid \phi\,\mathtt{U}^{\leq k}\phi$$

PCTL formulae are interpreted over the states of a DTMC or *Markov decision processes* (see next section). We say that a state $s \in S$ satisfies a PCTL formula ϕ, denoted $\mathcal{D}, s \models \phi$, if ϕ is true at the state s. Intuitively, a state s satisfies the basic state formula $\mathtt{P}_{\sim p}[\psi]$ if the probability of taking a path from s satisfying *path formula* ψ meets the bound $\sim p$. Further, the path formula $\mathtt{X}\phi$ (operator *neXt*) is true, if ϕ is satisfied in the next state; the path formula $\phi_1\mathtt{U}\phi_2$ (operator *until*) is true, if ϕ_2 is eventually satisfied and ϕ_1 is true until then; the path formula $\phi_1\mathtt{U}^{\leq k}\phi_2$ (operator *bounded until*) is true, if ϕ_2 is satisfied within k discrete steps and ϕ_1 is true until then.

In practice, it is common to write formulae of the following kind: $\mathtt{P}_{=?}[\psi]$, which asks "*what is the probability of ψ to be true*". Also, the following useful operators can be derived from the above PCTL syntax: $\mathtt{F}\phi ::= \mathit{true}\,\mathtt{U}\,\phi$ (*eventually* ϕ becomes true) and $\mathtt{G}\phi ::= \neg\mathtt{F}\neg\phi$ (ϕ is true *globally*), and a bounded variants of these.

2.2 Markov decision processes

A *Markov decision process* (MDP), like DTMC, consists of states, representing possible configurations of the system being modelled, and transitions between states occur in discrete time-steps. However, at each state the system (decision maker) may choose any action that is available in this state, and then non–deterministically move into a new state, while providing the decision maker a corresponding probability.

Definition 3. *An MDP is a tuple $\mathcal{M} = (S, s^\iota, Act, \mu, \mathrm{AP}, L)$, where:*

- *S, s^ι, AP and $L : S \to 2^{\mathrm{AP}}$ are defined as for DTMCs,*
- *Act is a finite set of actions,*
- *$\mu : S \times Act \to Dist(s)$ is the (partial) transition probability function, with $Dist(S)$ denoting the set of all discrete probability distributions over S.*

Observe that for each state $s \in S$, the successor state is determined in two stages: firstly, an available action $a \in Act$ (i.e. one for which $\mu(s, a)$ is defined) is non–deterministically selected; secondly, the successor is randomly chosen according to the probability distribution $\mu(s, a)$.

To reason formally about the behaviour of MDPs, normally, the notation of *policies* is used. A *policy* resolves all of the non–deterministic choices in an MDP.

Moreover, under the control of a particular policy, the behaviour of an MDP is fully probabilistic and, as is for DTMCs, one can define a probability space over the possible paths through the model. Further, it is possible to reason about the best– or worst–case system behaviour by quantifying over all possible policies: for example, it is possible to compute the minimum or maximum probability of a PCTL property. Finally, the notion of an *optimal policy* can be used with a property value optimised by a model checker. For example, `Prism` [5] can find an optimal policy with regard to the minimum or maximum possible probability of a PCTL property [6].

Properties of MDPs can also be written in PCTL, yet with an implicit quantification over policies. For example, the $\mathtt{P}_{=?}$ operator used for DTMCs is replaced with two variants $\mathtt{P}_{\min=?}$ (the minimum probability) and $\mathtt{P}_{\max=?}$ (the maximum probability).

3 Approximate extrapolation of policy

Let $\mathcal{M} = (S, s^\iota, Act, \mu, \mathrm{AP}, L)$ be an MDP. We first define an MDP *with classified binary choices* (MDPCBC) as follows.

Definition 4. *An MDPCBC is a tuple* $\mathcal{X} = (S, s^\iota, Act, \mu, \mathrm{AP}, L)$*, where:*

- S*,* s^ι*,* AP *and* $L : S \to 2^{\mathrm{AP}}$ *are defined as for MDPs,*
- $Act = \bigcup_{i=1}^{\alpha} Act_i$ *is a finite set of actions that is divided into* α *disjoint classes* Act_i *according to their meaning as understood in the modelled phenomenon. Moreover, each class* Act_i *is divided into two disjoint sets of the same size:* $Act_i{\downarrow}$ *and* $Act_i{\uparrow}$*.*
- $\mu : S \times Act \to Dist(s)$ *is the (partial) transition probability function, with* $Dist(S)$ *denoting the set of all discrete probability distributions over* S*, and the following property: any non–deterministic binary choice* q *contains exactly two actions* $a_q{\downarrow}$ *and* $a_q{\uparrow}$ *such that* $a_q{\downarrow} \in Act_i{\downarrow}$ *and* $a_q{\uparrow} \in Act_i{\uparrow}$*.*

For example, if $Act = Act_1 \cup Act_2$, then Act_1 might represent choices of either a black or a white ball and Act_2 might represent a decisions if to continue a loop of choosing the balls or, on the contrary, stop the process. Further, if a non–deterministic binary choice represents a class of actions “a choice of either a black or a white ball”, then $Act_1{\downarrow}$ would contain only choices of the black ball and $Act_1{\uparrow}$ would contain only choices of the white ball.

Now we define the notation of a decision boundary.

Definition 5. *Let* $\mathcal{X} = (S, s^\iota, Act, \mu, \mathrm{AP}, L)$ *be an MDPCBC, and let the binary non–deterministic choices between actions belonging to* Act_i *be available at certain states* $S_i \subseteq S$*. A* decision boundary *is a function* $D_i : S_i \to \{\mathit{false}, \mathit{true}\}$ *such that if there is a non–deterministic choice between actions* $a_q{\downarrow}$ *and* $a_q{\uparrow}$*, then the agent chooses* $a_q{\downarrow}$ *if* $D_i(s) = \mathit{false}$*, and* $a_q{\uparrow}$ *otherwise.*

Thus, a decision boundary determines a certain policy controlling an MDPCBC, which then becomes a DTMC, further called a *Markov chain with classified binary choices* (MCCBC).

As seen, thanks to the division of *Act* into classes, we have a number of class–specific decision boundaries, which individually may be easier to describe mathematically. On the contrary, mixing actions with vastly different meanings might produce a common decision boundary which is hard to analyse.

3.1 Method

Let there be a set $\mathscr{X} = \{X^1_{\text{small}}, \ldots, X^J_{\text{small}}, X_{\text{large}}\}$ of MDPCBCs, instantiated from a common template, but with different values of the parameter N, equal respectively to $\{N^1_{\text{small}}, \ldots, N^J_{\text{small}}, N_{\text{large}}\}$. The parameter does not influence on the nature of the problem, but merely represents a scale of the problem. N can be e.g. the number of philosophers in the Dining Philosophers Problem [2]. We want to estimate an optimal policy of X_{large}. Yet it is impossible to find X_{large} directly using PMC (e.g. implemented in `Prism`), due to extensive computational complexity and memory requirements. Therefore, we will attempt to extrapolate J optimal policies of J respective X^j_{small}, $j = 1, \ldots, J$.

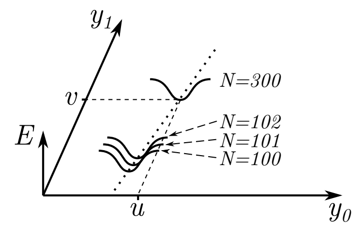
Let the decision boundary for class i in X^j_{small} be $D^j_i(s)$, $s \in S_i$, and let us pose a hypothesis on how $D^j_i(s)$ can be merged into a single function. The hypothesis is represented by $D_i(s, N)$, which generalises all $D^j_i(s)$ so that $D^j_i(s) = D_i(s, N^j_{\text{small}})$, $j = 1, \ldots, J$. This allows for obtaining a hypothetical extrapolated decision boundary $D_i(s, N_{\text{large}})$, controlling X_{large}. Finally, we strengthen or refute the posed hypothesis, by locally verifying the fitness of $D_i(s, N_{\text{large}})$ using SPMC.

3.2 Arbitrariness

Let any state in S be represented by a tuple $V = (x_1, \ldots, x_d)$, a so–called state vector, each of its elements represents some specific phenomenon in a modelled system. For example V might have an interpretation (temperature, precipitation). Let us build a real coordinate metric space $\mathscr{S}$ such that any state s is mapped to a point V_s in $\mathscr{S}$, having coordinates V. We do that because we hope that points close in $\mathscr{S}$ may intuitively represent similar situations in the modelled system, thus it is less likely that a decision boundary goes between them. This hopefully simplifies both the shape and semantic interpretation of $D_i(s, N)$.

Due to, amongst others, a finite $||S||$, the extrapolation to $D_i(s, N_{\text{large}})$ might be imprecise. Consider the following. See that $||V_s|| = ||S||$ is finite, and thus we can find some $\epsilon > 0$ which is equal to the closest distance between all possible pairs (V_{s_1}, V_{s_2}), $s_1, s_2 \in S, s_1 \neq s_2$. Therefore, there is an infinite number of decision boundaries representing a single policy. Let any decision boundary be represented by a vector of parameters $C(N) = (y^N_1, \ldots, y^N_\beta)$, where $\beta \in \mathbb{N}^+$ is a constant specific to a method of the representation. For any class i, we can extrapolate $D_i(s, N)$ by a respective extrapolation of J vectors $C_i(N^j_{\text{small}})$ to a single vector $C_i(N_{\text{large}})$, the latter determining $D_i(s, N_{\text{large}})$. Yet, as $C(N)$ is arbitrary due to $\epsilon > 0$, so is $C_i(N_{\text{large}})$. Moreover, as the extrapolation may augment the arbitrariness, the multiple possible values of $C_i(N_{\text{large}})$ may in turn represent multiple $D_i(s, N_{\text{large}})$.

Fig. 1. A schematic example of a trajectory of $C_1(N) = (y_0, y_1)$ in a parameterised Euclidean space $\mathbb{R}^2$. Schematically depicted minimum error E of a decision maker translates to an optimal policy of an MDPCBC instantiated with a given N.



Consider the example in Fig. 1(a). $C_1(100)$, $C_1(101)$ and $C_1(102)$ determine a common segment. We extrapolate that segment with a line in order to find $C_1(300) = (u, v)$, which in turn determines hypothetical $D_1(s, 300)$. Yet, as MDPCBCs have a finite number of states, $C_1(300)$ can not be determined precisely: minor arbitrariness in the placement of the segment scale up roughly affinely with the distance to the segment. We thus see, that there is at least a single reason for $D_i(s, N_{\text{large}})$ to be a hit–and–miss when it comes to an estimation of a strategy for X_{large}.

4 Case study

Let us study an example – an optimal policy of choosing a coin. There are two coins, a fair one and an unbalanced one. They have the probabilities of the outcome of heads equal to respectively 0.5 and 0.6. A decision maker flips a coin N times, deciding before each flip, which of the two coins to choose (for simplicity, N is even). What is the best policy of maximising the probability of drawing $N/2$ heads within these flips? The criterion of optimality is thus

$$P_{\text{opt}} = \mathtt{P}_{\text{max}=?}[\mathtt{F}(f = N \wedge h = \frac{N}{2})] \tag{1}$$

Let N be the scaling parameter discussed in Sec. 3.

Let us first verify a small variant of the model, with $N = N_{\text{small}} = 100$, using `Prism`'s PMC capabilities. `Prism` is able to compute the optimal policy of a decision maker which has a full knowledge about the system, and the computed policy is given as a transition matrix of a DTMC, which in the case of our model is also an MCCBC. The policy is visualised in Fig. 2(a), where f and h are parts of the vector state which are equal to, respectively, the number of tosses so far and the number of heads drawn so far.

Let $Act_1\!\uparrow$ be a choice of the fair coin. The decision boundary, approximated by visually interpreting Fig. 2(a), is

$$h \gtrsim 0.55\,(f - (9 \pm 0.5)) \tag{2}$$

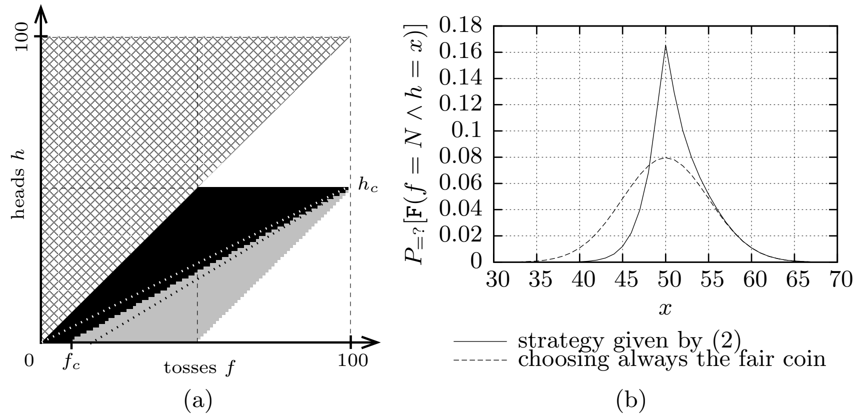


Fig. 2. (a) Visualisation of the optimal policy for $N = N_{\text{small}} = 100$. Gridded region depicts unreachable states, white region designates, that a successful policy is no more possible or a maximum number of tosses is reached, black and grey regions mean respectively "chose the fair or the unbalanced coin". White dashed line represents $h = f/2$, black dashed line shows an example variation of the decision boundary for a different f_c. (b) Probability distributions of h for two different strategies at $f = N = N_{\text{small}}$.

Fig. 2(b) depicts probability distributions of h after all N_{small} tosses. As seen, (2) makes (1) almost twice as large if compared to a state–agnostic approach of always choosing the fair coin.

4.1 A single small MDPCBC

Firstly, we will attempt to extrapolate from only (2), i.e. let $J = 1$. Assuming limited computational resources for finding optimal policies of small models, $J = 1$ enables us to use the largest possible N^j_{small}.

Extrapolation. In order to extrapolate from a single point, we will form some supporting hypotheses. It is easy to see that at the state $(f = 0, h = 0)$ the decision maker should choose the fair coin for any N. This is because he is more afraid of an excessively large h, rather than of the number of heads drawn being too small, as the single available unbalanced coin leans towards heads, and thus it can be used to reduce the deficiency of h. The latter also says, that a high deficiency of h leads to the choice of the unbalanced coin. Therefore, we know that at some $(f \geq f_c, h = 0)$, $f_c > 0$, the unbalanced coin is chosen, and that for any N, the fair coin is chosen for $(0 \leq f < f_c, h = 0)$. We also know that at $(f = N - 1, h = N/2 - 1)$ the decision maker should maximise the probability of drawing a head in order to reach $(f = N, h = N/2)$ (the success is assured only if a head will be drawn, an unbalanced coin is chosen), and that he would minimise that probability for $(f = N - 1, h = N/2)$ (the success is assured only if a head will not be drawn, a fair coin is thus chosen). Therefore, we know that for any N the decision boundary goes between these two states.

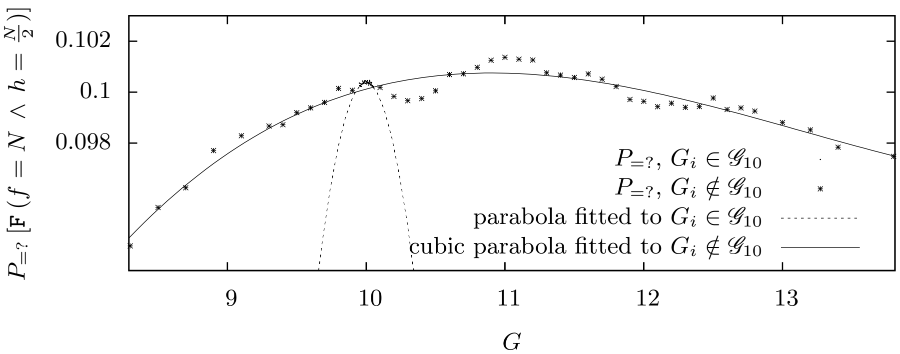


Fig. 3. Estimation of P_{opt} for $N = N_{\text{large}}$, $5 \cdot 10^7$ samples per MCCBC.

On the basis of (2) we can guess that the decision boundary is a segment. Let us guess that the placement of the segment scales affinely with N in the sense that $f_c \approx N/G$, where G is a constant. This hypothesis is trivially represented by the following decision boundary:

$$h > h_1(f) = \left(\frac{N}{2} - 1/2\right) \frac{f - f_c}{N - 1 - f_c} = \left(\frac{N}{2} - 1/2\right) \frac{f - N/G}{N - 1 - N/G}$$

thus

$$D_i(s, N) = \begin{cases} G < \dfrac{N}{N-1} \dfrac{2h + 1 - N}{2h - f} & \text{if } h < f/2 \\ \mathit{true} & \text{if } h \geq f/2 \end{cases} \tag{3}$$

Verification. Using (2) we can estimate $G = N^1_{\text{small}}/f_c \approx 100/(9 \pm 0.5) \underset{\sim}{\in} \mathscr{G} = \langle 10.5; 11.8 \rangle$. Checking statistically P_{opt} in an MDPCBC controlled by (3) for $N = N_{\text{large}} = 5000$ and $G \approx 11 \in \mathscr{G}$ yields the diagram in Fig. 3. An MCCBC with the highest P_{opt} is found for $G = G_1 = 11$, which agrees with the estimation, and we may thus strengthen the hypothesis.

Local maxima are seen in the diagram. For example, we statistically checked MCCBCs for a dense set of values of G in a set $\mathscr{G}_{10}$ such that $\forall_{G_i \in \mathscr{G}_{10}} 0.95 \leq G_i \leq 10.05, ||\mathscr{G}_{10}|| = 51$, then we fitted a parabola as seen in the figure, to show that a local gradient–descent optimiser might be trapped in a local maximum around $G \approx 10$, thus strengthening a respective false hypothesis.

We thus see, that an interval of G wider than the one spanned over $\mathscr{G}_{10}$ should be sampled by such an optimiser. This leads to a considerable numerical complexity of the resulting SPMC, as we need to use as much as $5 \cdot 10^7$ samples per a single estimation in order to get a 99% confidence level of $\approx 2.3 \cdot 10^{-4}$, estimated using Asymptotic Confidence Interval [10].

A less precise extrapolation might increase the said complexity even more. For example,

– a less precise initial estimation of G might result in a larger number of samples to be gathered, in order to localise the maximum around $G = 11$;
– if we would merely assume, that the decision boundary is a segment, whose both ends are unknown and with no known relation, and not that only f_c is unknown as we did so far, we would then need to search within at least a two–dimensional optimisation space. For example one with the optimised parameters (G, H) such that the linear boundary of $D_1(N_{\mathrm{large}})$ intersects a pair of points $(f_c = N_{\mathrm{large}}/G, 0)$, $(N_{\mathrm{large}} - 1, H)$.

Obviously, in general, an MDPCBC with $N = N_{\mathrm{large}}$ might become unverifiable using both PMC and SPMC, if the extrapolation from PMC–checked models were insufficiently precise or unknown at all.

4.2 Several small MDPCBCs

We will attempt to estimate $D_1(N_{\mathrm{large}})$ using several small MDPCBCs. We won't use the reasoning from Sec. 4.1, but instead, an optimiser will analyse a trajectory of parameters representing different decision boundaries.

Extrapolation. We apply a Nelder–Mead Simplex gradient–descent optimiser [9] to a linear combination of f, h, N and 1, with the goal of minimising the number of wrong choices in models with $N = 80, 100$ and 120, i.e. $J = 3$. We obtain a hypothetical generalised decision boundary

$$550.876f - 1001.63\,h - 50.1596\,N - 15.0067 \lesssim 0$$

thus for $N = N_{\mathrm{large}}$

$$h \gtrsim h_2(f) = 0.549980(f - 455.298), \quad G \approx G_2 = 10.982 \tag{4}$$

Testing (4) against the three MCCBC matrices returned by `Prism`, it turns out that this boundary always allows for a right choice within the three MDPCBCs in question. It may be a hint, that the said linear combination has been a right choice.

Verification. Due a finite number of states in the MDPCBCs from which we have extrapolated, we may expect imprecisions in (4) as discussed in Sec. 3, especially that we extrapolate from $N^j_{\mathrm{small}} \approx 100$ to $N_{\mathrm{large}} \approx 5000$, i.e. to a model with a number of tosses about 50 times as large on average.

We know from the reasoning in Sec. 4.1, that $N/2 - 1 \leq h(N-1) < N/2$. For $N = N_{\mathrm{large}}$, $h_2(N_{\mathrm{large}} - 1) \approx 2498.95$, and is thus too small by ≈ 0.05, a difference which seems to be fairly precise for an extrapolation that distant. Yet, we will correct that imprecision by placing the segment correctly at $f = N_{\mathrm{large}} - 1$, and then extracting from (4) merely the value of G_2. This boils down to the reuse of the diagram in Fig. 3. Given the low number of statistically verified MCCBCs in the diagram and the considerable size of the 99% confidence interval, given in Sec. 4.1, it can be stated that $G_2 \approx G_1$. We may thus strengthen the hypothesis.

5 Discussion

We are working on a tool for automating the presented extrapolation. As opposed to the example in the case study, it would apply a number of extrapolating functions beside a linear combination, in order to choose the best extrapolated policy. For example, the tool could support an extrapolation of oscillating functions like in [11], in order to deal with models of physical systems involving periodicity. For example, the scaling parameter might represent a rotational speed of an element, and the policy would minimise a standing wave in the supporting construction.

References

1. Clarke, E.M., Grumberg, O., Peled, D.: Model checking. MIT press (1999)
2. Dijkstra, E.W.: Hierarchical ordering of sequential processes. Acta Informatica **1**(2), 115–138 (1971)
3. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal aspects of computing **6**(5), 512–535 (1994)
4. Kwiatkowska, M., Norman, G., Parker, D.: Advances and challenges of probabilistic model checking. In: Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on, pp. 1691–1698. IEEE (2010)
5. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: G. Gopalakrishnan, S. Qadeer (eds.) Proc. 23rd International Conference on Computer Aided Verification (CAV'11), *LNCS*, vol. 6806, pp. 585–591. Springer (2011)
6. Kwiatkowska, M., Parker, D.: Advances in probabilistic model checking. In: T. Nipkow, O. Grumberg, B. Hauptmann (eds.) Software Safety and Security - Tools for Analysis and Verification, *NATO Science for Peace and Security Series - D: Information and Communication Security*, vol. 33, pp. 126–151. IOS Press (2012)
7. Kwiatkowska, M., Parker, D.: Automated verification and strategy synthesis for probabilistic systems. In: D.V. Hung, M. Ogawa (eds.) Proc. 11th International Symposium on Automated Technology for Verification and Analysis (ATVA'13), *LNCS*, vol. 8172, pp. 5–22. Springer (2013)
8. Legay, A., Delahaye, B., Bensalem, S.: Statistical model checking: An overview. In: International Conference on Runtime Verification, pp. 122–135. Springer (2010)
9. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. The Computer Journal **7**(4), 308–313 (1965)
10. Nimal, V.: Statistical Approaches for Probabilistic Model Checking. Master's thesis, Oxford University (2010)
11. Rataj, A.: Fractional genetic programming for a more gradual evolution. In: Proceedings of the 22nd International Workshop on Concurrency, Specification and Programming, Warsaw, Poland, pp. 371–382 (2013)
12. Sondik, E.J.: The optimal control of partially observable markov processes. Tech. rep., DTIC Document (1971)
13. Younes, H., Kwiatkowska, M., Norman, G., Parker, D.: Numerical vs. statistical probabilistic model checking. International Journal on Software Tools for Technology Transfer (STTT) **8**(3), 216–228 (2006)