Proxels Practically: The Meaning of the Lifetime Factor

Sanja Lazarova-Molnar and Graham Horton*
Otto-von-Guericke Universität Magdeburg
{sanja, graham}@sim-md.de

Abstract

Proxel-based simulation is a new and deterministic approach to analysing discrete stochastic models and studying their behaviour. Unfortunately, as it is the case with the deterministic approaches it suffers from the well-known state-space explosion problem, nevertheless having an advantage with respect to the other approaches. The proxel-based method benefits from the way it generates and stores the state-space, which is performed on-the-fly, meaning that only the truly reachable states are the ones that are stored. This in turn means that on-paper inspection of the model (i.e. its graphical representation in form of a Petri net or a reachability graph) does not determine how complex its behaviour will be in practice, and correspondingly its proxel-based analysis. Therefore, we needed a tactic to estimate the computational complexity of the proxel-based simulation for different models, thus allowing a better strategy for organisation of the data structures being used.

We refer to the factor that determines the actual state-space of one model (the one that is stored and truly reachable) as *lifetime* of the discrete states and formalise its impact on the computational complexity of the proxel-based analysis of stochastic models.

1 Introduction

The proxel-based method is a relatively new method for analysing stochastic models which was introduced in [Hor02], and is based on the method of supplementary variables [Cox55, Ger00]. Until now there has not been developed an approach which would aid the process of complexity prediction of the proxel-based analysis of a given model. In this paper we present a paradigm for which we believe will provide a hint on the complexity of the proxel-based simulation regarding the description of the model that is to be analysed. We refer to this paradigm as a *lifetime of a discrete state* and it denotes the longest time that the model can reside in a given discrete state. This is also the factor that determines the real complexity of one model, as opposed to the discrete state space.

In the following, we provide a brief description of the proxel-based method, after which we describe the lifetime factor and its role in predicting the complexity. Furthermore, we present a simple example which supports and illustrates our claims.

*Fakultät für Informatik, Institut für Simulation and Graphik, D-39016 Magdeburg, Germany

2 Proxel-Based Simulation

The proxel-based method works by observing all of the possible behaviours of the model, each with a determined computable probability, based on the distribution functions which describe the events, as well as the time they have been pending (denoted as age intensity). The unit that stores all necessary information for turning a non-Markovian model into a Markovian one is referred to as *proxel*, which stands for “probability element”. Among others, it contains the discrete state, the age intensities of the possible events in that discrete state, and the probability, resulting into the following structure:

Proxel = (State, Time, Route, Probability), where *State = (Discrete State, Age Intensity Vector).*

The *Route* parameter contains the sequence of states via which the model has reached the actual state. Time advances in discrete steps and the point where the simulation starts is the initial discrete state. From there on, based on the possible state changes (associated with events) new proxels are generated for the subsequent time step. The values of the corresponding age intensities are updated with respect to the event that has caused the state change.

The *probability* is approximated by the IRF (instantaneous rate function) $\mu(\tau)$ [Tri02], integrated along the time step, where τ is the age intensity of the active state change i.e.

$$probability = \int_{t}^{t+\Delta t} \mu(x)dx, \text{ approximated by } probability = \mu(t) \times \Delta t, \qquad (1)$$

which in this case we interpret as the probability that the state change has happened within the interval $[t, t + \Delta t)$. The IRF is computed from the distribution functions (CDF and PDF) as follows:

$$\mu(\tau) = \frac{f(\tau)}{1 - F(\tau)}. \qquad (2)$$

More on the proxel-based method can be found in [Hor02, LMH05, LM05].

3 Lifetime Computation

Prior to the proxel-based analysis, if the analysis is to be carried out on a bounded state space (i.e. the model has a limited number of discrete states), then a preprocessing step can be carried out for computing the lifetimes of the discrete states. The computed lifetimes provide a preview of the computational complexity and the memory requirements of the concrete proxel-based analysis, and are used for computing the keys of the proxels in the binary tree which is the data structure used for storing the proxels. A unique key is assigned to every proxel, which is computed based on the state that the proxel represents, i.e. the combination of the discrete state and the age intensity vector. Therefore, it is very helpful to be able to predict the largest value that each age intensity can have.

When finite support distributions are associated with the state changes in a model, then it is predictable that the model can spend a limited amount of time in each discrete state. However, when the state changes are distributed according to infinite support distributions, then theoretically the model can spend an infinite amount of time in each discrete state. The probabilities for staying in the discrete states, in general, decrease as time increases. At a certain point in time, they become so small that we can treat them as negligible. We decide that the probabilities for staying in the discrete states are small enough when they become smaller than the predefined minimum probability threshold ϵ, which is usually around 10^{-15}. This makes it possible to determine simulation characteristic lifetimes of the discrete states.

A *lifetime* of a discrete state determines the longest time that the model can spend in that discrete state regarding the concrete simulation parameters ϵ, and is calculated based on the distribution functions that are associated with the active state changes in the actual discrete state. The procedure for the calculating the lifetimes is described in Algorithm 1. In this algorithm for simplicity reasons we use the most simple approximation method for the IRF, which is based on the starting point of each interval.

Algorithm 1: Computing Lifetimes of Discrete States

```
Input: Δt, t_max
foreach discrete state DS in the model M do
    t = 0;
    lifetime(DS) = 0;
    prob_exit(DS) = 0;
    prob_stay(DS) = 1.0;
    for t = 0 to t_max in steps of Δt do
        foreach active state change SC in DS do
            prob_exit(DS, SC) = μ_SC(t) × Δt;
            prob_exit(DS) = prob_exit(DS) + prob_exit(DS, SC);
        end
        prob_stay(DS) = prob_stay(DS) × (1.0 − prob_exit(DS));
        if prob_stay(DS) < ε then
            lifetime(DS) = t;
        end
    end
    if lifetime(DS) = 0 then
        lifetime(DS) = t_max;
    end
end
```

The symbols used in the algorithm have the following meanings:

- t_{max} is the maximum simulation time and Δt is the size of the time step,
- $prob_{exit}(DS)$ is the total probability for exiting the discrete state DS,
- $prob_{exit}(DS, SC)$ is the probability for exiting the discrete state DS through the state change SC,
- $prob_{stay}(DS)$ is the probability for not leaving the discrete state DS,
- $\mu_{SC}(t)$ is the value of the instantaneous rate function of the random variable that describes the state change SC, having an age intensity of t.

The algorithm works by calculating the probabilities for leaving the disrete states through any of the state changes at every discrete time step (lines 8 and 9) and thereby the probability for not leaving the discrete state (line 11), until the end of simulation time t_{max} is reached (line 6). If the probability for not leaving the discrete state is less than ϵ (line 12), that means that at that point in time there is a probability of zero or negligible probability to stay in the same discrete state. Therefore the point in time at which that happens is assigned as the lifetime of the actual discrete state (line 13). If that point in time, where the probability for staying in the same discrete state is negligible, is not reached within the maximum simulation time t_{max}, then the maximum simulation time t_{max} is assigned as the lifetime of the actual state for the actual proxel-based simulation (lines 16 and 17).
The lifetimes of the discrete states in a model directly influence and determine the complexity of the simulation. The reason for that is that they are a factor that determines the real state space of the model in terms of proxels, thereby considering only the truly reachable states. If the model has no more than one concurrently active state changes with different activation times, then we propose considering the sum of the lifetimes of all states in a model as a measure for the complexity of that model's proxel-based analysis. In cases where this condition does not hold, the number of supplementary variables in each state has to be considered too, as it increases the number of their possible combinations when generating states i.e. proxels. The presence of age memory state changes is a typical case of having more than one active state changes with different activation times. In that case we define a *maximum number of states generated from a discrete state* as follows.

Definition 1 (Maximum number of states generated from a discrete state (MNSDS)) *Maximum number of states generated from a discrete state* (MNSDS) *is the maximum number of combinations of the state vector, considering the lifetimes of the discrete states.*

MNSDS is equal to the ratio of the lifetime of the discrete state and Δt when there are no more than one active state changes with different activation times. If that is not the case MNSDS is calculated as a product of the lifetime of the actual discrete state and the lifetime(s) of the state(s) in which the age memory state change(s) is (are) active. MNSDS is directly used for calculating the key of each proxel.
The effect that lifetimes have on the computational complexity of the proxel-based simulation is supported and illustrated with concrete examples in the next section.

4 Experiments

As it is pointed out in the previous section, the number of concurrently active state changes together with the characteristics and the parameters of the random variables affects the computational complexity of the proxel-based simulation. Furthermore, the addition of concurrently active state changes is not a constraining factor of the proxel-based simulation. It can be seen as a beneficial or neutral one, unless the state changes are activated at different points in time.
The experiments presented in this section demonstrate the relation of the computational complexity of the proxel-based simulation of one model and the sum of the lifetimes of its

discrete states. The model that we choose for this set of experiments excludes age memory state changes, which are present in the models from the experiments shown in [LM05] and in that case the complexity is higher, as illustrated.

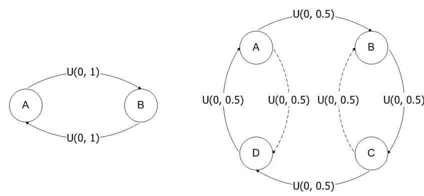
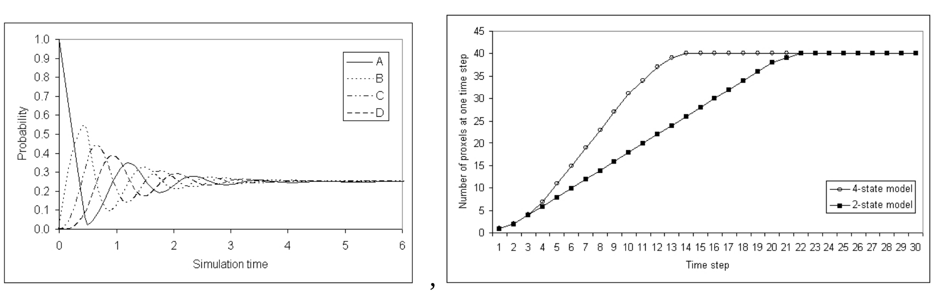


Figure 1: Two models with equal proxel complexities

Let us observe the models shown in Figure 1, excluding the state changes illustrated by dashed lines. The model with two discrete states seems on the first view as less complex than the four-state one. However, when analysed using the proxel-based method, both models have equal complexities because the sums of the lifetimes of their discrete states are equal. We chose this example, because the uniform distribution function has the nice property of having a finite support, making it simpler to determine the lifetimes of the discrete states.

When chosen $\Delta t = 0.05$, then the lifetimes of both discrete states in the two-state model are $20\Delta t$, resulting into a maximal number of 40 proxels. In the four-state model, the lifetimes of the discrete states are $10\Delta t$ resulting again into a maximal number of 20 proxels. If we now add the state changes represented by the dashed lines to the four-state model, then the lifetimes of the discrete states *A* and *C* shorten insignificantly (because of the increased probability of leaving the discrete state). This results again into almost the same computational complexity i.e. number of proxels, which is 36. The computation times of both simulations are same too. Transient solution of the four-state model excluding the dashed state changes is shown in Figure 2a. The model has a steady-state solution: $Pr(A) = Pr(B) = Pr(C) = Pr(D) = 0.25$.



(a) Solution values for the four-state model from Figure 1 excluding the dash-lined state changes

(b) Comparison of the numbers of proxels generated at each time step for the two models from Figure 1, excluding the dash-lined state changes

Figure 2: Results from the experiments

In Figure 2b the comparison of the number of proxels generated at each discrete time step for the 2-state and 4-state models is shown. It is evident that the 4-state model reaches the maximum number of proxels sooner than the 2-state one, which is because of the greater number of state changes which generate accordingly more proxels at each time step.
This experiments show that the discrete state space of one model is not a measure for the complexity of its proxel-based simulation. Instead, the distributions associated with the state changes must be taken into account and this can be directly used for predicting the complexity.

5 Summary and Outlook

The paradigm of a lifetime, as presented here, can be an important factor in predicting the complexity of the proxel-based simulation of a given model. This in turn means that it aids the completeness of the proxel-based method. We believe that the "a priori" computation of lifetimes can also serve in optimising the data structures used for storing the proxels, as well as for deciding when to use phase-type approximations for substituting some of the distributions [IH05].

References

[Cox55] D. R. Cox. The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables. *Proceedings Cambridge Philosophical Society*, 51(3):433–441, 1955.

[Ger00] Reinhard German. *Performance Analysis of Communication Systems. Modelling with Non-Markovian Stochastic Petri Nets*. John Wiley & Sons, Ltd, 2000.

[Hor02] Graham Horton. A new paradigm for the numerical simulation of stochastic Petri nets with general firing times. In *Proceedings of European Simulation Symposium, Dresden, October 2002*. SCS Verlag, 2002.

[IH05] Claudia Isensee and Graham Horton. Approximation of discrete phase-type distributions. In *Proceedings of Annual Simulation Symposium 2005 in San Diego, USA*, apr 2005.

[LM05] Sanja Lazarova-Molnar. *The Proxel-Based Method: Formalisation, Analysis and Applications*. PhD thesis, aug 2005.

[LMH05] Sanja Lazarova-Molnar and Graham Horton. Description framework for proxel-based simulation of a general class of stochastic models. In *Summer Computer Simulation Conference 2005*, jul 2005.

[Tri02] Kishor S. Trivedi. *Probability and statistics with reliability, queuing and computer science applications*. John Wiley and Sons Ltd., Chichester, UK, UK, 2002.