True State-Space Complexity Prediction: By the Proxel-Based Simulation Method

Sanja Lazarova-Molnar
College of Information Technology
United Arab Emirates University
United Arab Emirates
sanja@uaeu.ac.ae

Abstract

All state-space based simulation methods are doomed by the phenomenon of state-space explosion. The condition occurs when the simulation becomes memory-infeasible as simulation time advances due to the large number of states in the model. However, state-space explosion is not something that depends solely on the number of discrete states of the model as typically observed. While this is correct and completely sufficient for Markovian models, it is certainly not a sufficient criterion when models involve non-exponential probability distribution functions. In this paper we discuss the phenomenon of state-space explosion in terms of accurate complexity prediction for a general class of models. Its early diagnosis is especially significant in the case of proxel-based simulation, as it can lead towards hybridization of the method by employing discrete phase approximations for the critical states and transitions. This can significantly reduce the computational complexity of the simulation.

1. Introduction

All state-space based simulation methods encounter the problem of state-space explosion [1]. The state-space based methods build models by enumerating the states of the system being analyzed. It is believed that the state-space explosion problem appears due to the large number of states that typically real-world systems have.

We address this problem from the perspective of the proxel-based method [2], which is a state-space based approach that builds the reachable state-space of the model on-the-fly [3]. The method also suffers from state-space explosion, which is the reason why we need an accurate complexity prediction. Therefore, we offer an approach for predicting the true complexity of a proxel-based simulation as way to counteract and downsize the problem of state-space explosion. As described in [4] the computational complexity of the method can benefit by replacing certain transitions with discrete phases [5]. In some cases the improvement factor was even ca. 100 in terms of computation time. Therefore, we need complexity prediction in order to make adequate decisions about which transitions are to be substituted. Further in the paper we provide brief background on the proxel-based simulation method, followed by our prediction algorithm and supplemented by examples.

2. Preliminaries

Here we introduce the state-explosion problem and the proxel-based method, as necessary for explaining our complexity prediction approach.

2.1 State-space explosion problem

State-space explosion problem appears in the state-space simulation methods because of the large number of states that real systems have. It is also known that the state-space grows exponentially with the size of the model description. This is especially true for Petri Net descriptions that involve many concurrent activities. There have been many approaches that try to downsize this problem. However, most of them are oriented towards CTMCs (continuous-time Markov chains), as described in [3, 6].

2.2 Proxel-based method

The proxel-based method was introduced in [7], and further formalized and analyzed in [2]. The method stands for an implementation of the Cox's method of supplementary variables [8] and deterministically simulates discrete stochastic models. It works by converting a non-Markovian model into a Markovian one using additional variables that track the aging of events. These are then used for calculating the probabilities with which the events can happen at any point in time. Its main strength is that it can be applied to models that contain any probability distribution functions, matching its modeling power to the one of discrete-event simulation [9]. Therefore, it allows for much more realistic modeling of various phenomena. Additionally, the proxel-based method achieves higher accuracy because it does not rely on generating random numbers.

The proxel-based method advances by observing all possible developments of a model, probabilistically weighting each of them. At the end, a transient solution is obtained which contains the probability of each state at every point in time. The probabilities are calculated based on the instantaneous rate function (IRF). The IRF approximates the probability that an event will happen within a predetermined elementary time step, given that it has been pending for a certain amount of time τ (indicated as 'age intensity'). It is calculated from the probability density function (f) and the cumulative distribution function (F) using the following formula:

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

In Figure 1, we present a basic two-state model and the first two levels of its proxel-tree. The latter one describes the advancement of the model in terms of proxels. As mentioned previously, the state vector contains also the 'age intensity' (e.g. (A, 0), (A, Δt)). The initial probability is denoted as 1.0, and the proxel-based method has converted the non-Markovian model into a Markovian one by recording the 'age intensities' in the state variables.

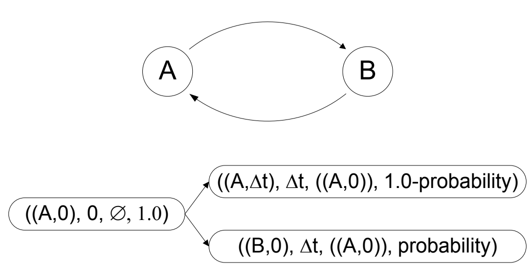

Figure 1. A two state model, and its two-level proxel-tree

In addition, note that proxels that are generated at the same level and represent the same state are combined in one proxel. This proxel carries the sum of the probabilities of the combined proxels as its probability parameter. More details on the proxel-based method and its algorithms can be found in [2].

3 Lifetimes of discrete states and computational complexity

Observing both models presented in Figure 2, the immediate impression is that the model shown in Figure 2(b) is more complex than the one in Figure 2(a). This is so because it contains twice as many discrete states and trice as many transitions. This conclusion is not necessarily true (and actually wrong and misleading) and our goal is to show that, as well as define a measure for a more accurate assessment of the complexity of discrete stochastic models. As a matter of fact, more transitions can even mean lower complexity of the model, which we will explain further.

As previously described, the term *state* defines the discrete state of the model including the age intensity, i.e. the time that the model spent in it prior to a transition to another state. Accordingly we define a *lifetime* of a discrete state as the longest time a model can spend in one discrete state, i.e. the longest reachable age intensity of one discrete state. The lifetime has to be observed dependent on the probability distribution functions that describe the activities related to the particular discrete state.

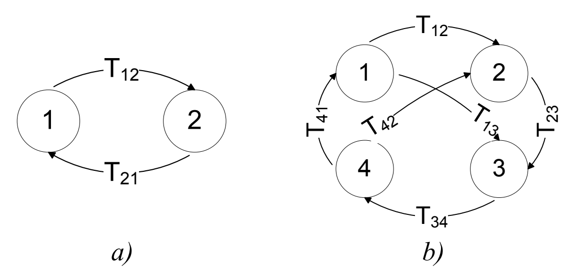

Figure 2. Two sample models with different discrete state spaces

We claim that the discrete state space (which is equivalent to a reachability graph in Petri net terminology) is not a single factor that determines the complexity of models' proxel-based simulation. The sum of the lifetimes of all discrete state is what defines the computational complexity of the simulation. We will show this by an example in the next section.

We propose the following algorithm for computing the complexity of a proxel-based simulation of a discrete stochastic model. It is based on our algorithm for computing the lifetimes of the discrete states in a model [2].

```
Input: Δt, t_max
Output: number_proxels
sum_lifetime = 0;
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 transition T enabled in DS do
            prob_exit(DS, T) = μ_T(t) × Δt;
            prob_exit(DS) = prob_exit(DS) + prob_exit(DS, T);
        end
        prob_stay(DS) = prob_stay(DS) × (1 − prob_exit(DS));
        if prob_stay(DS) < ε then
            lifetime(DS) = t;
        end
    end
    if lifetime(DS) = 0 then
        lifetime(DS) = t_max;
    end
    sum_lifetime = sum_lifetime + lifetime(DS);
end
number_proxels = sum_lifetime / Δt;
```

Symbols in the algorithm have the following meanings:

- ε is the predetermined negligible-probability threshold, typically 10^{-15} in our experiments,
- t_{max} is the maximum simulation time and Δt is the size of the time step,
- *probexit*(*DS*) is the total probability for exiting the discrete state *DS*,
- *probexit*(*DS*, *T*) is the probability for exiting the discrete state *DS* through the transition *T*,
- *probstay*(*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 transition *T*, having an age intensity of *t*,
- *sum_lifetime* is the sum the lifetimes of all discrete states in the model, and
- *number_proxels* is the total number of proxels generated for the specific model.

Our algorithm as an output produces an upper bound of the maximum number of proxels generated at each simulation step for a specific model which equals to the number of nodes in the proxel-tree. The proxel-based method generates the truly reachable state-space on-the-fly which only contains the tangible states. Therefore this figure can be viewed as a prediction for the true complexity of one discrete stochastic model. This is a more accurate measure than the number of states and transitions in a discrete state space or a reachability graph, as it contains more information about the model, i.e. it involves the probability distribution functions (in the algorithm used in form of an instantaneous rate function $\mu_T(t)$).

In the following we provide a demonstration of our approach on the two sample models from Figure 2. This illustrates that the true complexity of one model cannot be deduced by a simple observation of its discrete state space.

4. Experiments and results

We base our experiments on the models shown in Figure 2. The proxel-based method does not impose any restrictions regarding to the types of probability distribution functions that can be used. The probability distribution functions that we use in our models are the following:

- in the 2-state model:

  T_{12} ~ Normal (10, 1),
  T_{21} ~ Uniform (10, 20), and

- in the 4-state model:

  T_{12} ~ Normal (1.0, 0.01),
  T_{23} ~ Uniform (0.5, 1.0),
  T_{34} ~ Uniform (1.0, 3.0),
  T_{41} ~ Uniform (1.0, 1.5),
  T_{42} ~ Normal (2.0, 0.01),
  T_{42} ~ Uniform (1.0, 2.0).

We choose to run the proxel simulation for both models with a time step $\Delta t = 0.02$ up to simulation time $t = 20$. This is sufficient to demonstrate that the complexity of the 4-state model is higher than the one of the 2-state model.

Using the algorithm that we presented, we calculated the lifetimes of the discrete states for both models, as presented in Tables 1 and 2. In both tables we see the sums of lifetimes of all discrete states in the models. This is the number that reflects the true complexity of the proxel-based simulation for the selected models. Roughly, it can be said that the complexity of the 2-state model is double the complexity of the 4-state model, which is the opposite of what is expected.

This example proves that the probability distribution functions are essential in determining the complexity of one discrete stochastic model. In addition to this, the higher number of transitions in a model can also imply lower complexity, as this means that each state has more ways to exit, and therefore a limited lifetime. It only takes one uniformly distributed transition to limit the lifetime of a discrete state very precisely.

If we include the size of the time step ($\Delta t = 0.02$), then we will get the accurate figure of the maximum number of proxels that can be generated at one time step. For the 2-state model it is 40/0.02 = 2000, whereas for the 4-state model it is 20.5/0.02 = 1025. This, of course represents an upper limit. The proxel-based simulation can actually benefit from some probability distribution functions (e. g. exponential) by not having to employ age variables, which reduces the number of proxels that are actually generated.

The actual numbers of proxels generated at time t = 20 are 375 for the 4-state model and 1100 for the 2-state model. If simulated long enough these numbers become closer to the predicted upper bounds. In Figure 3 we can observe the cumulative number of proxels generated with respect to the simulation steps. The number of proxels grows with polynomial complexity in the order of the branching factor of the proxel tree until it reaches the maximum number of proxels generated at a time step. Afterwards its growth is linear as the number of proxels generated does not change.

In Figure 4 we see the plot of the number of proxels generated at a time step as a function of the number of steps. It is very interesting to observe the abrupt behavior that develops due to the involvement of uniform probability distributions. Namely, these distributions limit the time range during which a state transition can occur.

In Figures 5 and 6 the transient solutions of the two models from Figure 2 are shown. This, one again, illustrates the ability of the proxel-based method to generate smooth transient solutions within relatively short computing time intervals. In our case, each simulation took ca. one second.

Table 1. Lifetimes of the discrete states of the 2-state model from Figure 2(a).

Discrete State	Lifetime
1	20
2	20
Total	40

Table 2. Lifetimes of the discrete states of the 4-State model from Figure 2(b):

Discrete State	Lifetime
1	5
2	4
3	6
4	5.5
Total	20.5

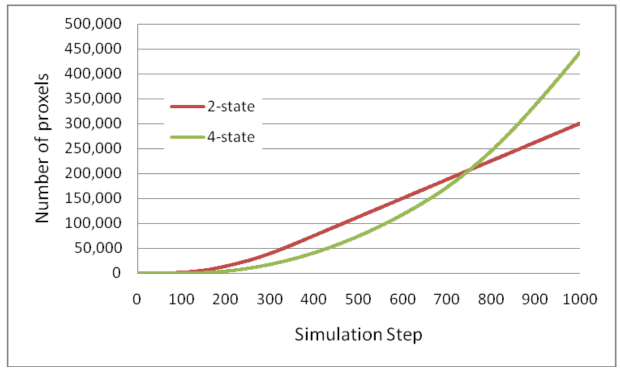

Figure 3. Number of proxels generated as a function of the simulation step for the 2- and 4-state models from Figure 2.

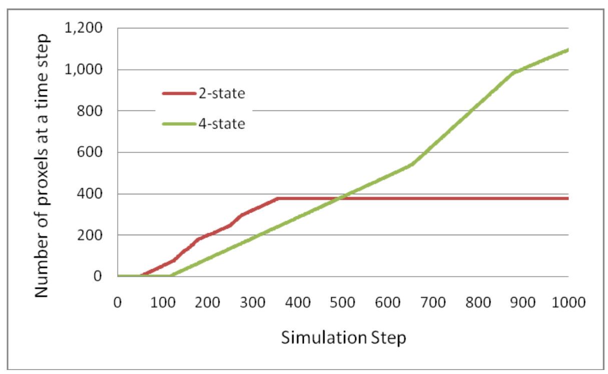

Figure 4. Number of proxels generated at a time step as a function of the simulation step for the 2- and 4-state models from Figure 2.

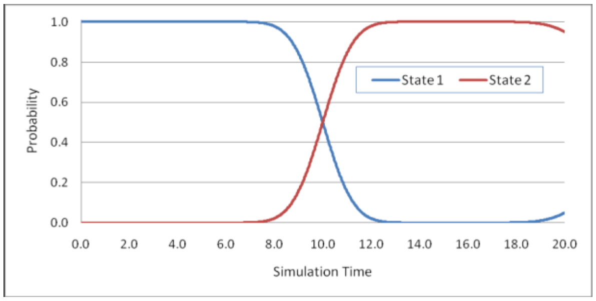

Figure 5. Transient solution of the model from Figure 2(a).

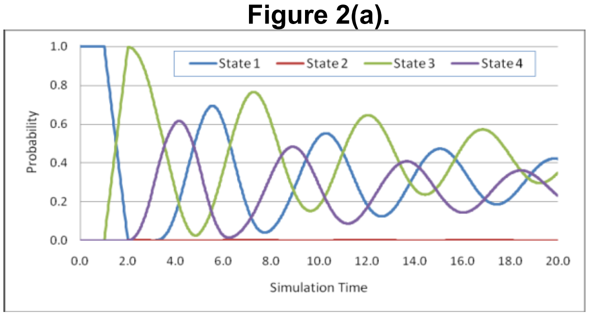

Figure 6. Transient solution of the model from Figure 2(b).

5. Summary and outlook

The paper is motivated by the need of computational complexity prediction of the proxel-based method. We need a more precise technique to determine the true complexity of one discrete stochastic model, as opposed to only the number of discrete states and transitions (which is in fact sufficient only when Markovian models are in question). This initiated the development of an algorithm to compute the so-called lifetimes of each discrete state in a model to ultimately reflect the complexity of the model as the sum of the lifetimes of all discrete states in the model.

The only drawback of our approach currently is that it assumes that all transitions are enabling, i.e. that their clocks restart as soon as they are activated. In our further research we plan to enhance our algorithm, such as to be able to handle aging transitions as well.

Additionally, we believe that the differences in rates should also play an important role in determining the computational complexity. These differences actually dictate the size of the time step. Also, the higher the difference in the rates is, the higher the complexity of the simulation will be as the size of the time step will have to be small enough to capture all transition rates.

We believe that our approach, even though related to the proxel-based method, determines the computational complexity that can be beneficial for other simulation methods as well; especially the state-space based ones. In addition it can aid in spotting bottlenecks in models and applying special approaches to them. For instance, detection of such problems in the proxel-based simulation leads to the substitution of the problematic transitions by discrete phases and thereby improving the efficiency of the simulation. In the same way the presented approach can be useful to PDE (partial differential equations)-based approaches because their complexity depends on the same factors as the proxel-based method, i.e. size of the time step and lifetimes of the discrete states.

10. References

[1] A. Valmari, "A stubborn attack on state explosion," *Formal Methods in System Design*, vol. 1, pp. 297-322, 1992.

[2] S. Lazarova-Molnar, "The Proxel-Based Method: Formalisation, Analysis and Applications," in *Faculty of Informatics*, vol. Ph.D. Magdeburg: University of Magdeburg, 2005.

[3] D. D. Deavours and W. H. Sanders, "“On-the-fly” solution techniques for stochastic Petrinets and extensions," *Software Engineering, IEEE Transactions on*, vol. 24, pp. 889-902, 1998.

[4] C. Isensee, S. Lazarova-Molnar, and G. Horton, "Combining Proxels and Discrete Phases," *Proceedings of ICMSAO*, 2005.

[5] A. Bobbio, A. Horvath, M. Scarpa, and M. Telek, "Acyclic discrete phase type distributions: Properties and a parameter estimation algorithm," *Performance Evaluation*, vol. 54, pp. 1-32, 2003.

[6] W. J. Knottenbelt, "Parallel performance analysis of large Markov models," *Unpublished PhD thesis, Imperial College, University of London*, 1999.

[7] G. Horton, "A new paradigm for the numerical simulation of stochastic Petri nets with general firing times," *Proceedings of the European Simulation Symposium*, 2002.

[8] D. R. Cox, "The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables," *Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 433-441, 1955.

[9] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-event system simulation*: Prentice Hall Budapest, Hungary, 2001.