# Fast Simulation Without Randomness: A Simulation Tool Combining Proxels and Discrete Phases

Claudia Isensee, Graham Horton
claudia@sim-md.de, graham@sim-md.de
Institut für Simulation und Graphik, Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany

## Abstract

The simulation of discrete stochastic models using state space-based methods has recently become practical through the proxel-based simulation algorithm. Proxels implement the method of supplementary variables and generate a DTMC of the model's state space. However, due to state space explosion the method does not yet work efficiently on larger models. By using discrete phase-type approximations, which are also DTMCs, for some distributions the state space can be reduced significantly and larger models can be simulated with comparable accuracy. The tool presented here enables a user to selectively replace general distribution functions by phase approximations and simulate the resulting model. This is one step further towards a general purpose proxel and phase-based simulator.

## 1 Introduction

This paper will introduce a tool implementing a proxel and phase-based simulation algorithm. This section will motivate the work and show previous developments. The following section will describe the developed tool. Some experiments and a conclusion form the last part of the paper.

### 1.1 Motivation

Discrete stochastic models are usually simulated using discrete event simulation. In some cases this stochastic approach does not lead to satisfactory results in reasonable computation time. Especially models involving rare events require many replications, which leads to very high computational cost.

State space-based methods such as the recently developed proxel-based simulation algorithm [2] do not suffer from this drawback, since they are deterministic. Proxels are based on the method of supplementary variables [1] and extend a model's state space by including information about the age of the currently enabled and some other transitions. The model is turned into a discrete-time Markov chain by calculating the transition probabilities between the states on-the-fly using the instantaneous rate function. However, due to the exploration of all possible states of the model and the use of one Markov chain state per distribution time step, this method suffers from state space explosion. Therefore, it usually only performs well on small models.

Another way to turn general distributions into Markov chains is by approximating them with phase-type distributions [6]. In contrast to proxels, discrete phases use a Markov chain structure where the probability to remain in a state can be non-zero. They can therefore represent distributions with fewer states than the number of time steps of the discretization. The disadvantage of discrete phase approximations is that they have to be precomputed and integrated in the model before the actual simulation can start. An additional degree of freedom is the number of phases used, where a larger number leads to better fits, but also to more computation time for the approximation.

Since proxels and discrete phases are both DTMC structures, they can be combined easily, which is shown in [4]. By reducing the number of states needed to represent a non-Markovian distribution, discrete phases can lessen state space explosion, and therefore reduce the memory and computation time needed for proxel-based simulation, thereby making the simulation of larger models more feasible. However, this gain can be outweighed by the time needed for the computation of the phase-type approximation, especially when large phase numbers would be required to approximate a distribution accurately.

Until now, when simulating using a combination of proxels and phases, the model had to be statically specified including the precomputed phase approximations. In order to use the simulation method efficiently, a flexible tool is needed that can simulate using both proxels and phases. More importantly, it should be possible to replace an arbitrary distribution within a model with a phase-type approximation dynamically by the user. In order to aid the user in the decision whether to use proxels or phases, and which approximation to choose, recommendations should be made during the process. This paper will introduce a general purpose simulator using both proxels and phases, with a graphical user interface.

### 1.2 Previous Work

The proxel-based simulation method was introduced in [2]. Proxel simulation has been included in a general purpose Petri net simulator, which is described in [7]. However the tool is not publicly available, and does not yet incorporate the use of discrete phases. Some special purpose tools have been developed such as [5]. In [3] an approximation algorithm for discrete phase-type distributions was introduced. The formal inclusion of phases into the proxel paradigm was described in [4].

## 2 Tool Description

This section describes the developed simulation tool and its different elements by going through the process of simulating a small model. The process involves the input of the reachability graph, the substitution of some transitions by discrete phase-type approximations, the simulation itself and the visualization of the results. As a demonstration model the test case from [4] is taken, due to its low complexity.

## 2.1 Model Input

The input and simulation process is controlled through a main dialog. Besides opening input files, controlling the approximation and simulation steps, the dialog also enables the user to change the global simulation time step and the maximum simulation time. The current implementation takes a reachability graph as input. This is due to the fact that no Petri net editor was readily available, which covered all necessary modeling elements and could be easily connected to a newly developed simulation tool. Developing a better interface for model input is a future task and will further enhance the usability of the developed tool. Because of the selected input format, the reachability graph of the model has to be finite, in order to simulate it with this tool.

## 2.2 Approximating and Replacing Distributions

The next step is to replace certain distributions by discrete phase approximations. The distributions that are used in the model can be displayed in a *Distributions* dialog that also contains suggestions on whether to replace a distribution by a phase-type approximation or not. Regardless of that suggestion, the user still has the option to do otherwise. When having approximated a general distribution with a phase-type one, there is also the possibility to reset to the original distribution.

One can approximate a given general distribution using the default settings of the approximation algorithm. The only alterable parameters using this option are the approximation method and the number of phases. Suggestions are given for these two values that usually result in a good fit. These recommendations are based on experimental data and consider the parameters of the distribution to be approximated and the discretization time step. This time step has to be the same as the overall simulation time step. The advanced approximation option should only be used when familiar with the internals of the approximation algorithm DPH [3].

Once the approximation has been calculated the resulting error is displayed. A graphical interpretation of that result is presented as shown in Figure 2.2 (left). It shows the discretized PDF of the input distribution and the output of the discrete phase approximation. This gives the user a visual clue to the accuracy of the approximation. If satisfied, the user can then replace the original distribution by the calculated approximation.

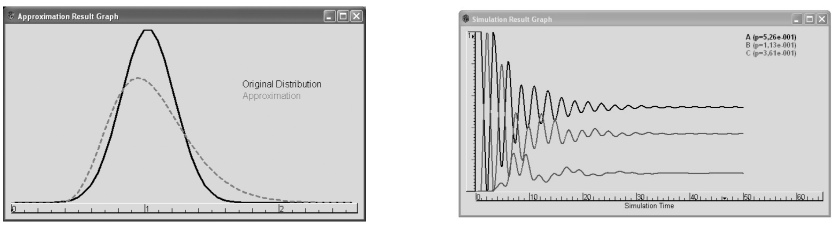


Figure 1: Approximation and Result Graph Window

### 2.3 Model Analysis and Results

The simulation itself is started through the main dialog. The model is fully specified through the reachability graph and the representations of the distributions. The proxel and phase simulation algorithm is described in detail in [4]. The simulation results are output into a text file, that can be further analyzed, and they can also be displayed graphically. The *Simulation Result Graph* window shows the probability graphs of all model states and can optionally also display the final state probabilities, see Figure 2.2 (right). A large state space can make this diagram complex, and therefore the user can also choose which states to display in the result graph. For detailed evaluation of the results however, the output text file will be more useful.

## 3 Experiment

In this section, experiments will be shown that illustrate how the use of discrete phases can reduce the computation time of state space-based simulation, making it possible to simulate larger models. The Petri net of the model used is shown in Figure 2 (left). The system is a simple fast food restaurant with two servers; a drive in and a counter. The two types of customers (people and cars) arrive through the transitions *AP* and *AC* which are exponentially distributed ($AP \sim Exp(1)$ $AC \sim Exp(0.5)$) and are served through the generally distributed transitions *SP* and *SC*. When there are no cars in the queue, the drive in server also serves people and vice versa. Three different configurations where tested that differ in the maximum queue lengths for cars (X) and people (Y) and therefore result in state spaces of different size:

- Configuration 1 - X=2, Y=5 $\rightarrow$ state space size: 13 (Figure 2 (right))
- Configuration 2 - X=3, Y=7 $\rightarrow$ state space size: 24
- Configuration 3 - X=3, Y=10 $\rightarrow$ state space size: 33

The service time transitions have the following distributions:

- Experiment 1 - $SP \sim N(1, 0.2), SC \sim N(2, 0.2)$
- Experiment 2 - $SP \sim W(1, 2), SC \sim W(2, 2)$

When using phases for experiment 1, the Normal distribution $N(1, 0.2)$ is replaced by a 8-phase approximation and $N(2, 0.2)$ with a 16-phase approximation. For the second experiment, both Weibull distributions were replaced by a phase approximation of order 4. These approximations are sufficiently close to the original distribution functions.

The simulation was run until time $25$ and the simulation time step was $0.1$. The statistical results of the experiments are shown in Table 1. The table on the left shows the results using only proxels and the table on the right shows the results when using phases for the two non-Markovian distributions. The column **#Prox** contains the number of proxels generated during the simulation process and **Time** shows the total time in seconds needed for the

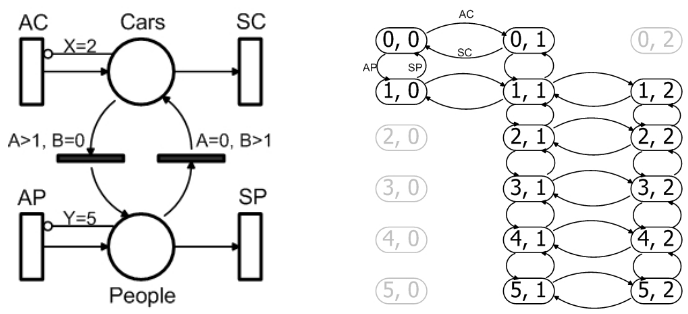


Figure 2: Petri net and reachability graph of a simple fast food restaurant model

simulation and, in case of the phases, also for the approximation. The columns **tSim** and **tAppr** show the time needed to perform the actual simulation and the approximation of the general distributions. The number in brackets behind the configuration denotes the size of the state space.

| | Using only Proxels | | Using Proxels & Phases | | | |
|---|---|---|---|---|---|---|
| | **#Proxels** | **Time** | **#Proxels** | **Time** | **tSim** | **tAppr** |
| | *Exp1* ($SC/SP \sim N$) | | *Exp1* ($SC/SP \sim N$) | | | |
| *Conf1* (13) | $962,019$ | $\mathbf{14.28}$ | $308,975$ | $\mathbf{1.95}$ | $1.64$ | $0.31$ |
| *Conf2* (24) | $2,256,137$ | $\mathbf{39.39}$ | $638,108$ | $\mathbf{4.87}$ | $4.56$ | $0.31$ |
| *Conf3* (33) | $3,371,027$ | $\mathbf{62,14}$ | $902,432$ | $\mathbf{7.89}$ | $7.58$ | $0.31$ |
| | *Exp2* ($SC/SP \sim W$) | | *Exp2* ($SC/SP \sim W$) | | | |
| *Conf1* (13) | $4,611,478$ | $\mathbf{50.56}$ | $41,522$ | $\mathbf{0.25}$ | $0.22$ | $0.03$ |
| *Conf2* (24) | $9,528,552$ | $\mathbf{143.91}$ | $84,224$ | $\mathbf{0.56}$ | $0.53$ | $0.03$ |
| *Conf3* (33) | $13,509,663$ | $\mathbf{231.82}$ | $118,552$ | $\mathbf{0.89}$ | $0.86$ | $0.03$ |

Table 1: Experiment results using only proxel and a combination of proxels and phases

As Table 1 shows, the larger the state space of the model, the more the state space explosion slows down the proxel computation. In Experiment 1, the computation time increases from $14$ seconds for *Conf1* to $62$ seconds for *Conf3*. When using phases for the non-Markovian distributions, the simulation time also increases (from $1.6$ to $7.6$ seconds), but it is still much less than when using proxels alone. Furthermore, the time needed for the approximation does not depend on the size of the state space, and its influence on the overall time needed decreases with growing state space. In Experiment 2 the benefit from using phases is even higher. The simulation time needed when using proxels for *Conf1* is already $50$ seconds. Because the two Weibull distributions can be approximated very well using only a small number of phases, this is reduced to under a second when using phases.

This experiment illustrates how proxel-based simulation can become infeasible when the state space of the model is too large. The use of phase-type approximations for some non-Markovian distributions can reduce the state space of a model significantly, and can thereby make proxel and phase-based simulation more attractive for larger models. The gain in using phases over proxels is larger, when fewer phases are needed relatively to the time steps of the original distribution.

## 4 Summary and Outlook

This paper introduced a tool for the proxel and phase-based simulation of stochastic models. The tool has a graphical user interface and enables the user to selectively replace generally distributed transitions within a model by phase-type approximations. Experiments also show the benefit that can be gained when using discrete phases. The state space of a model is reduced significantly and a model that could otherwise not be effectively simulated using proxels becomes feasible when using discrete phases.

The major drawback of the developed tool is the restriction to reachability graphs as input format. This results in a restriction to models with a finite state space. Proxels in general can also simulate models with an infinite state space. In order to allow for such models the input format of the tool needs to be changed. A graphical editor for model input would be even more user friendly, and should enhance the usability of the tool significantly.

## References


[1] *Cox, D.:* ”The analysis of non-Markov stochastic processes by the inclusion of supplementary variables.” Proc. Camb. Phil. Soc. (Math. And Phys. Sciences) 51, 443-441. 1955.

[2] *Horton, G.:* ”A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times,” European Simulation Multiconference, Darmstadt, June 2002. SCS European Publishing House, 2002.

[3] *Isensee C., Horton G.:* ”Approximation of Discrete Phase-Type Distributions,” 38th Annual Simulation Symposium, San Diego, CA, USA, 2005.

[4] *Isensee C., Horton G.:* ”Combining Proxels and Discrete Phases,” International Conference on Modeling, Simulation and Applied Optimization 2005. Sharjah, U.A.E. February 2005.

[5] *Lazarova-Molnar, S., Horton, G.:* ”Proxel-Based Simulation of a Warranty Model,” European Simulation Multiconference 2004, SCS European Publishing House 2004.

[6] Neuts, M. F., ”Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach,” The John Hopkins University Press, Baltimore, 1981.

[7] *Wickborn, F., Heller, S., Engelhard, F, Horton, G.:* ”A General-Purpose Proxel Simulator for an Industrial Application ,” 18. Symposium Simulationstechnik (ASIM 2005), Erlangen, Germany, 2005.