Improving the Efficiency of the Proxel Method by Using Individual Time Steps

Claudia Krull[1], Robert Buchholz[2], and Graham Horton[1]

[1] Intitut für Simulation und Graphik, Otto-von-Guericke-University, Universitätsplatz 2, Magdeburg 39106, Germany
[2] Institut für Technische und Betriebliche Informationssysteme, Otto-von-Guericke-University, Universitätsplatz 2, Magdeburg 39106, Germany

Abstract. Discrete stochastic models (DSM) are widely used in various application fields today. Proxel-based simulation can outperform discrete event-based approaches in the analysis of small stiff DSM, which can occur for example in reliability modeling. However, when parallel processes with largely differing speed are involved, the faster process determines the small discretization time step, investing far too much effort into the approximation of the slower process. This paper relieves that problem by using individual time steps for each transition and situation. The key problem is to keep semantic consistency when using different time steps for parallel transitions. However, the preservation of the probability mass in every single simulation time step could be achieved. Experiments show that binary step division in conjunction with appropriate subdivision criteria can outperform the original Proxel method significantly. This increases the applicability of Proxels, by enabling the analysis of larger and therefore more realistic models.

Keywords: Proxel-based simulation, discrete-time Markov chains, state space-based simulation, discrete stochastic models.

1 Introduction

Discrete stochastic models (DSM) are used in various application fields such as reliability modeling, manufacturing, planning and control. The analysis of DSM is usually done using discrete event-based simulation. This can become expensive when small stiff models are involved, such as in reliability modeling. Many replications might be necessary to obtain statistically significant results. Proxel-based simulation [1,2] can outperform traditional DES on small stiff models. It deterministically discovers all possible system developments and their probabilities in discrete time steps, avoiding a possibly large number of replications.

However, when transitions with largely differing speed are involved, the original Proxel algorithm cannot perform optimally. A globally fixed constant time step is either too small for a slow transition, investing more computational effort than necessary, or it is too large for a fast transition, producing a too large error. A small constant time step also leads to a more pronounced state space explosion, severely limiting the size of model that can be simulated effectively.

An ideal solution would be to use a different time step for each transition, adjusted to the speed of the transition, or even to the current situation. The paper shows an approach realizing the concept of individual time steps for each transition, these are also called variable time steps. It uses binary step division, providing maximum flexibility and computational efficiency. The key problem to tackle is a semantic inconsistency that arises when using different time steps for parallel transitions. By redistributing the probability completely in every step, the original Proxel-method ensures that the sum of the probability of all possible states at one time is still 1. This is no longer valid when using differently sized time steps in parallel. The solution proposed here computes all probabilities for the smallest time step necessary. For slower transitions this probability is collected in a container, which is processed further along the time scale.

Experimental results indicate that using an appropriate subdivision criterion, variable time steps can increase the accuracy of a Proxel simulation by several orders of magnitude, or accordingly decrease the computation time necessary to obtain the same accuracy. Extrapolation of the results using different thresholds for the subdivision criteria is only of limited applicability. The paper shows, that variable time steps can reduce state space explosion, eventually enabling the simulation of larger models and thereby increasing the applicability of Proxels. The theory and results presented here are the result of a Masters thesis [3].

2 Proxel-Based Simulation

The Proxel-based simulation algorithm was developed by Horton [1] and further improved by Lazarova-Molnar [2]. It is a state space-based simulation method, which is based on the method of supplementary variables [4]. This section gives a brief overview of the basic ideas involved in Proxel simulation.

A so-called Proxel (see Equation (1)) represents a probability element in the expanded state space of the model. It contains the discrete model state dS and the age vector τ of the active or race age transitions as coordinates in the expanded state space. It also includes the current point in simulation time t, the route to this particular Proxel R and the probability of that combination p. Route R and simulation time t are rarely explicitly included in practical implementations, t is usually global and by omitting R, the reachable state space is reduced considerably.

$$P = (S, R, p) = ((dS, \tau, t), R, p) \tag{1}$$

By extending the discrete system states with the discretized age of all currently enabled or otherwise interesting transitions, a non-Markovian process is turned into a discrete-time Markov chain (DTMC) [5]. The one-step transition probability between these DTMC states can be determined dynamically using the so-called instantaneous rate function (IRF), saving the effort of explicitly building the Markov chain.

The IRF as defined in Equation (2) represents the current rate of probability flow from one state to the next [1].

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

The main idea of the Proxel algorithm is a simple iterative approach. The initial Proxel $((S_0, \boldsymbol{\tau_0}, 1))$ contains the initial system state S_0, an all zero age vector τ_0 and the probability 1. All possible successive system states are determined and using the given discrete time step Δ, the corresponding transitions IRF μ and the transitions age in τ their probability at time Δ can be approximated, assuming at most one state change can happen within one time step. This procedure is then repeated for all follow-up Proxels generated at time Δ. Thereby a so-called Proxel tree is generated, which contains all possible system developments at discrete intervals. The number of Proxels at one point in time can become quite large, due to the possible number of combinations of values in the age vector, expanding the system state. This increases memory requirement as well as computation cost.

The basic assumption of the Proxel method is that at most one state change can happen within one time step, therefore the simulation result can only ever be an approximation. The time step plays a central role in the performance and accuracy of the method. The smaller the time step, the more accurate the result is, but also the more pronounced the state space explosion and the more expensive the computation. However, larger time steps can be used to extrapolate more accurate results using Richardson extrapolation [6].

The key problem of Proxels is the state space explosion. Thus, a time step as large as possible is desirable. However, if transitions of differing speed are involved, the faster transition determines the maximum possible step size, resulting in a time step that is smaller than necessary for the slower transition. The example in Figure 1 illustrates that behavior. It shows a small stochastic Petri net with two consecutive transitions that differ in speed by a factor of 10. Both have a normal distribution and the same coefficient of variation. Using the appropriate time step for the faster transition computes the result for the slower transition to an accuracy that is ten times as large. Using the appropriate time step for the slower transition reduces the accuracy of the faster transition by a factor of 10. Either too much is invested, or the error induced is too large. In this example, the ideal time steps of the two transitions would also vary by a factor of 10, approximating both with comparable accuracy and effort.

The constant global time step is one of the key performance factors in Proxel simulation, it influences accuracy, as well as cost. However, the accuracy achieved

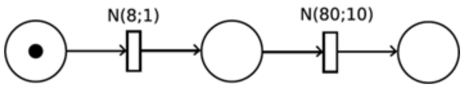

Fig. 1. Simple Example Model with Transitions of Differing Speed

using the same time step can be very different for transitions with largely differing speed. This paper tries to remedy this problem by using a separate time step for each transition, which is adapted to the individual transitions speed.

2.1 Error Sources in Proxel Simulation

Since the goal of individual time steps is to simulate transitions of different speed with comparable accuracy, one needs to be able to estimate the accuracy. Therefore the error sources of Proxels need to be identified and investigated.

BA Error. The first source of error in Proxel-based simulation is an error made knowingly through the basic assumption (BA). This assumption states that no more than one state change can be made within one discrete time step, neglecting the possibility of having two or more consecutive state changes within one step. Since this is one basic property of the simulation algorithm, the effect of the error can only be reduced by reducing the step size, but not by changing any algorithm feature. However, the error can be estimated by calculating the probability of two consecutive state changes in one step (see [3] for further details).

ODE Error. The second known source of error in Proxel-based simulation is the integration method used when estimating the one-step transition probabilities (ODE). The dynamic behavior of a single transition is defined by Equation (3), where $\varPi(t)$ represents the probability of the state at time t and $\mu(t)$ is the value of the instantaneous rate function (IRF) at time t. This means that the rate of change of the probability to stay in a state is proportional to the hazard rate function value and the remaining probability mass at that time. This is an ordinary differential equation and due to discretization the problem stated in (4) needs to be solved. Former Proxel implementations used Eulers method (5) or trapezoid integration (6) (Heuns method), which is not very accurate, but was sufficient for small time steps. When using larger time steps for slower functions, the error induced by an inappropriate integration method can no longer be disregarded. The error can however easily be reduced by using higher order integration methods such as Runge-Kutta. Using embedded Runge-Kutta methods, the error can even be estimated at little extra cost. One example of such a method is the combination of Euler's method and Heun's method, forming the ODE12 integration scheme. A more accurate, but also more expensive method was introduced by Dormand and Prince using fourth and fifth order integration methods, which is simply called ODE45. A more detailed description of the integration methods used in this paper can be found in [3].

$$\frac{d\varPi}{dt} = \varPi'(t) = -\varPi(t) * \mu(t) \tag{3}$$

$$\varPi(t+\varDelta) = \varPi(t) + k * \varDelta \tag{4}$$

$$k = -\varPi(t) \times \mu(t) \tag{5}$$

$$k = -\varPi(t) \times \frac{1}{2}(\mu(t) + \mu(t+\varDelta) \tag{6}$$

ND Error. A third source of error has been examined in detail for the first time in [3]. It can be called error through the non-deterministic behavior of the method (ND). Proxels approximate the continuous flow of probability between two states by a step function, causing an unnatural behavior, which can have unexpected side effects. One of these effects is, that there is one time step delay in activation of the transitions leaving a subsequent state, even though when in real continuous time they would have been activated almost as soon as the simulation started. This error can be estimated by a more detailed computation of the transitional behavior within the time step. Experiments showed that it can be reduced by simply initializing the age of a new state to $\frac{1}{2}\Delta$ instead of 0 as was done before. This takes into account that the subsequent transition could in reality have been activated anytime within the time step and not at its end. This is still just an approximation, but showed to be efficient and effective.

One primary goal of variable time steps is to control and reduce these errors. Only then can larger time steps be used without unwanted loss of accuracy.

3 Introducing Variable Time Steps

This section first defines requirements of a Proxel algorithm using individual (variable) time steps. Then some basic properties of the algorithm and the new algorithm itself is described. The final part of the section details on the time step subdivision criteria as a central item of the VTS algorithm.

3.1 Requirements of a VTS Algorithm

A suitable algorithm for variable time steps (VTS) should ideally fulfill the following requirements:

- It can choose the ideal step size for each transition and current setting. Only then can the full potential of VTS be exploited.
- It can dynamically determine the step size during runtime using model and transition properties. This is more flexible than a pre-computation step, since it can also react to changing circumstances during runtime.
- It does not have too much overhead over constant time steps (CTS), regarding computation time and memory requirement. Too much overhead would reduce the advantage of VTS compared to CTS.
- It is still able to conserve probability by completely distributing it to all subsequent Proxels. It is necessary to locate the total probability mass in each time step of the simulation to compute statistics and result measures.

Some of these requirements are contradicting, such as maximum flexibility and minimum overhead, therefore we are looking for a good trade-off between them.

3.2 Properties of the VTS Algorithm

This section describes two key decisions made regarding the current implementation, which also distinguish it from a former attempt to use variable time steps in Proxel simulation [7].

The first decision was taken regarding the method of time step division. A choice providing maximum flexibility would be an arbitrary time division. This is however not suitable, since merging Proxels with the same age at the same point in simulation time is one central way to counter state space explosion. This would however no longer be possible using arbitrary step division. The compromise between computational efficiency and flexibility taken in this paper is to choose a binary step division scheme. This means for each transition leaving a Proxel to decide whether the current time step is small enough, according to some criterion, or whether it should be bisected and simulated in two steps. This ensures that Proxels can be merged at higher level division points, maximizing the number of joining points. Another effect is that it reduces the question of the ideal time step to a simple decision problem, which can be applied recursively. Therefore the decision can be made during runtime based on current model and transition properties.

The second decision was to chose the time step separately for each transition leaving a Proxel. This allows for maximum flexibility at additional computational effort. However, choosing different time steps for parallel transitions can result in semantic problems. It is no longer intuitively clear, how the probability should be distributed to the different child Proxels, which can now exist at different points in time. The conservation of the probability mass needs to be ensured at each simulation time step, enabling the consistent computation of transient statistics at the smallest steps taken. The solution chosen in this paper is to calculate the probability leaving the parent Proxel for the smallest time step chosen. For the transitions using longer time steps it is stored in containers for later handling.

3.3 New Proxel Simulation Using Variable Time Steps

This section briefly describes the current algorithm for Proxel-based simulation using individual time steps.

The simulation starts with an initial Proxel with probability 1, all transition ages set to 0 and in the initial discrete system state, as in the original Proxel algorithm [2]. Then the algorithm computes all possible follow-up Proxels, resulting from the firing of the transitions active in this state. In contrast to the original algorithm the time step, and therefore the point in future simulation time where the Proxel is created, is determined individually for each transition. Since binary step division is used, the algorithm starts with a single time step reaching to the end of simulation time and bisects it recursively. The time step is then fixed for each transition out of the current Proxel and repeated until no more probability is left in the original Proxel's marking. For each Proxel generated this scheme is repeated, except that the maximum time step for a transition is the next larger step taken on the higher levels. This ensures the possibility of merging Proxels later on in the simulation time.

The algorithmic handling of different parallel time steps is pictured in Figure 2 (right). The SPN of the model being simulated is shown in Figure 2 (left). The model has three parallel transitions, the transition leading to place B is the fastest of the three, the transition to place C is two times slower and the

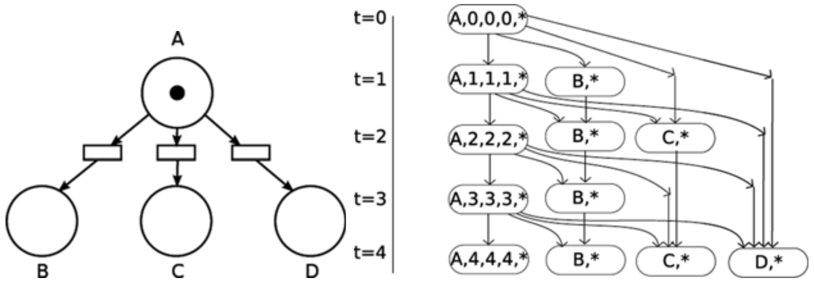

Fig. 2. Example SPN of Model with Three Parallel Transitions of Differing Speed (left) and Computing Successor Proxels for Example Model (right)

transition leading to state D four times slower than AB. Figure 2 (right) shows the start of the Proxel tree resulting from the simulation of this SPN. Each Proxel contains the following information: the current discrete system state, the age of the three activated transitions in state A and an asterisk as a placeholder for the probability of that expanded system state.

The discrete time steps chosen for the three transitions to places B, C, and D are 1, 2 and 4. The Proxels for staying in place A and entering place B are created in every time step. The probability to leave state A for the states C and D is also computed for each step of size 1, however the Proxels are only created at time 2 and 4 respectively. The probability leaving A in one time step is stored in these containers until the appropriate time step is reached.

In this way one can ensure that the sum of the probability of all current and future Proxels created out of a single Proxel corresponds to the Proxels original probability. In each time step the location of the total probability mass is clearly defined. The additional computational effort to compute all outgoing probability at the minimal step needs to be invested to ensure semantic unambiguousness and conservation of the probability mass. To reduce the effort of determining the memory location and retrieval of future Proxels, the definition of a Proxel is extended by a list of redirectors and redirector recursion depth. With the help of these, the target Proxel for each transitions probability can be determine using constant effort. The global storage of the Proxels in different time steps is realized by a hierarchy of Proxel containers, one for each level of the recursion. Experiments showed that no more than 20 recursion levels were required to simulate the tested models to a sufficient accuracy.

3.4 Time Step Subdivision Criteria

One crucial point of a good algorithm for variable time steps using binary step division is a good subdivision criterion. Five different criteria were developed in [3], which will be described here and tested in the experiments section 4. The main goal is to produce comparable accuracy for the simulation of transitions with differing speed. Therefore, the subdivision criteria try to achieve a similar error in every simulation time step. All of the criteria use certain thresholds to determine whether the time step should be subdivided or not.

GLOBAL_PROB. A naive approach of a flexible time step size would be to set a global threshold for the maximum probability of a Proxel (GLOBAL_PROB). This would mean to subdivide a time step until the resulting child Proxels have a probability below the given threshold. This seems to be a reasonable criterion, since the smaller the overall probability of a Proxel the smaller the possible contribution to the global simulation error. This however fails when processing the follow-up Proxels created for the initial Proxel. All of these have a probability below the global threshold, therefore their outgoing transitions can be simulated using the maximum possible time step, since the probability of the subsequent Proxels can never exceed that of the parent Proxel. This is not the desired behavior, and therefore (GLOBAL_PROB) is not applicable.

TRANS_PROB. Another simple idea is to limit the probability of a one-step state transition (TRANS_PROB). This threshold limits the fraction of probability that can leave a Proxel (discrete system state) within one time step. This also makes sense, since using smaller time steps also reduces the possible error made in one step. However, as the probability of the original Proxel is reduced, the fraction reduces accordingly. If in each step half of the probability is allowed to leave the Proxel, it will theoretically never loose all of its probability and the time steps will become smaller, eventually preventing an advancing of simulation time beyond the support of the distribution. The Proxel algorithm however prevents this loop, since it discards Proxels below a given probability threshold. Therefore, (TRANS_PROB) will be investigated further.

ORIG_PROB. The third criterion limits the amount of probability leaving a Proxel within one step to a given fraction of the original Proxel probability (ORIG_PROB). This ensures that the time step is not decreased indefinitely as can happen for TRANS_PROB and that all of the probability leaves the Proxel eventually. It also ensures similar accuracy in all successor Proxels and therefore is a promising candidate for a subdivision criterion.

MEAN. The fourth criterion is based on a finding documented in [2], where it was stated that the maximum time step allowed should not exceed one half of the fastest transitions mean value. Generalizing this we define the MEAN criterion: the time step size is not allowed to exceed the fraction k of the transitions distribution mean. This rather simple criterion ensures about equal accuracy of all distribution approximation, scales with the transitions speed and prevents an indefinite subdivision of the time step.

UNANIMOUS. The fifth criterion developed in [3] is based on the three error sources of Proxel-based simulation (see Section 2.1), which were described in Section 2.1. The criterion UNANIMOUS is used in such a way, that it no longer subdivides a time step when all three error methods do not exceed a given threshold anymore. This should directly limit the error made in each time step to the given threshold.

All criteria except the GLOBAL_PROB seem to be suitable candidates for a valid subdivision criterion. Therefore they need to be tested for their applicability and performance regarding runtime and accuracy. All of these criteria are based on threshold values. Therefore, an extrapolation to a small threshold value using rougher estimates computed using larger thresholds should be theoretically possible. A formal proof of linear convergence of the results when reducing the threshold k in a given criterion was not possible. Therefore experiments were conducted to test the applicability of the Richardson extrapolation.

4 Experimental Results

This section describes some of the experiments conducted using the newly developed Proxel-based algorithm using variable time steps. The experiments are selected from [3], for more detailed results and further experiments the reader is referred there.

The first set of experiments (see Section 4.1) tests several combinations of subdivision criteria and integration methods. We expect that at least some of the criteria can reliably outperform constant time steps by attaining better accuracy with comparable computational effort. The second set of experiments (see Section 4.2) tests the applicability of Richardson extrapolation using the thresholds of the subdivision criteria.

The general experiment setting used was the following:

– Trapezoid integration for the approximation of one-step transition probability. (see Equation (6))
– As an exception, for the UNANIMOUS criterion embedded Runge-Kutta methods (ODE12, ODE45) were necessary to estimate the integration error per step.
– The age of a newly created Proxel was initialized with $\frac{1}{2}\Delta$, to reduce the ND-error (see Section 2.1).
– Proxel cutoff probability was set to 10^{-15}, meaning that Proxels with a probability below that threshold were discarded, in order to dampen state space explosion.

In earlier papers the comparison criterion between different approaches was always the effort invested and accuracy obtained using a given time step size. Since we are now looking at dynamically determined time step sizes, the performance criterion needs to be generalized. The comparison between different algorithm configurations happens according to the accuracy that could be obtained over the computational effort invested. The accuracy is determined by the error of the Proxel result compared to the analytically obtained solution. The models chosen for testing were all simple enough to obtain an analytical solution for the results of interest. Neither the solution accuracy nor the computational effort can be controlled directly, both are a result of algorithm runtime behavior.

Thus, the algorithms can not be compared at specific points and the individual measurements need to be interpolated to yield comparable plots.

4.1 Which Subdivision Criterion to Use?

The first experiment was conducted using a simple chain-like model (see Figure 3) with five successive transitions of decreasing mean value. The distributions of the successive state transitions are the following: $N(1; 0,25)$, $N(4; 0,5)$, $N(9; 0,75)$, $N(16; 1)$ and $N(25; 1,25)$. The result measure of interest in the chain model was the average time of the last transition firing.

The accuracy over runtime results for the criteria ORIG_PROB, MEAN and TRANS_PROB as well as for constant time steps (CTS) are depicted in Figure 4. The results of all other criteria tested were omitted from the diagram, because they did not converge to the actual result value (which was determined analytically). In this diagram the lowest curve shows the most efficient algorithm, since here the least effort was needed to achieve a certain level of accuracy.

The constant time steps exhibit a reliable behavior, the more computational effort is invested, the smaller the error of the result. Only the ORIG_PROB criterion can outperform the constant time steps, and it does so by about two orders of magnitude. Hence, the plot looks as if it were a straight line. The MEAN criterion can almost compete with the performance of CTS, but has a slightly higher effort to obtain comparable accuracy. The TRANS_PROB criterion behaved reliably, increasing the accuracy with increased effort, but could not reach the efficiency of constant time steps.

The second experiment was conducted using a warranty model based on a real life Proxel application also discussed in [2]. The reachability graph of the model is shown in Figure 5. The goal is to compute the warranty cost for a given configuration of year and mileage based warranty expiration, a given failure distribution function and cost per failure. The measure of interest in the warranty model was the average cost incurred within the warranty period.

The result graph includes results for constant time steps and the following subdivision criteria: ORIG_PROB, MEAN, TRANS_PROB, UNANIMOUS using ODE12 embedded integration method (Unanimous-ODE12) and ODE45 embedded integration method (Unanimous-ODE45). In this experiment again, CTS showed reliable behavior, even though not as linearly-looking as for the chain model. The constant time steps were only outperformed by the MEAN criterion and by ORIG_PROB, with the latter one being the most efficient criterion overall. The two settings of the UNANIMOUS criterion could not outperform CTS,

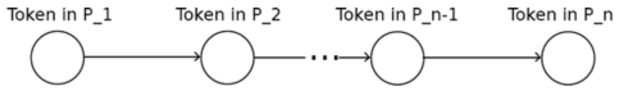

Fig. 3. Reachability Graph of Chain Model

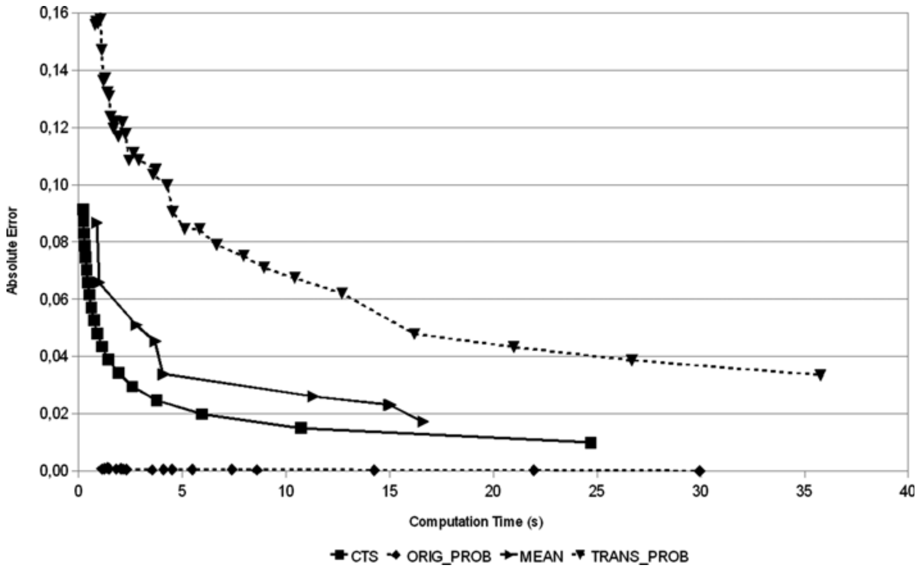

Fig. 4. Accuracy Obtained Over Computation Time for Chain Model and Different Subdivision Criteria

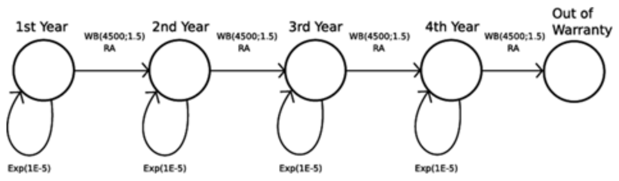

Fig. 5. Reachability Graph of Warranty Model

even though the combination with ODE45 is comparable to constant time steps regarding the results.

Result discussion. The experiments comparing the different time step subdivision schemes showed that few of them could reliably outperform constant time steps. For further experiments refer to [3]. Only the ORIG_PROB criterion was better than CTS in all experiments. MEAN was competitive, but not always better than CTS. The TRANS_PROB criterion performed worse than CTS on some models. The UNANIMOUS criterion did not converge to the analytical result for any model and can not be used reliably. This poor performance is probably due to a combination of still poor understanding of the exact effects of the different errors and resulting improper estimation. Furthermore it is not clear how to combine three totally different error effects using only one threshold, or what the combination of three different thresholds could be. A competitive UNANIMOUS criterion requires further research effort on these topics. Overall the results show

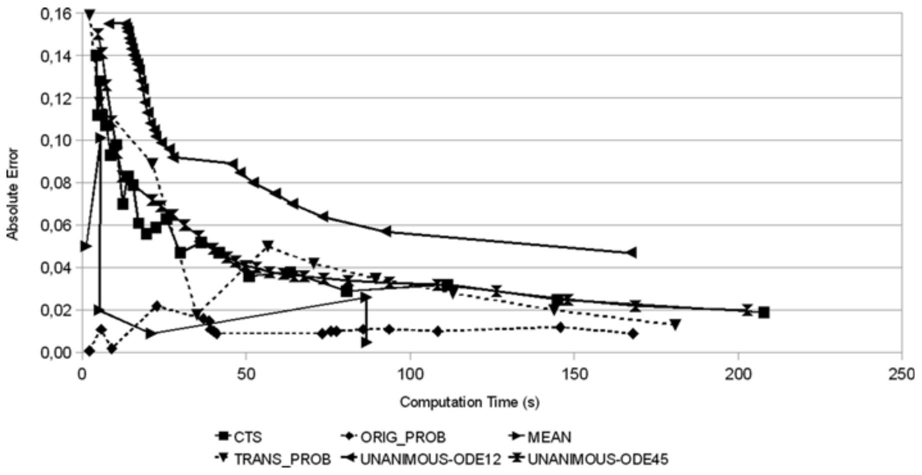

Fig. 6. Accuracy Obtained over Computation Time for Warranty Model and Different Subdivision Criteria

that individual (variable) time steps can outperform constant time steps, when an appropriate time division criterion is chosen (here ORIG_PROB).

4.2 Applicability of Richardson Extrapolation

This section presents some experimental results in the applicability of Richardson extrapolation. Only MEAN and ORIG_PROB were tested, due to their constant and reliable performance in earlier experiments. The extrapolation was done using the threshold value k for the fraction of the transitions distribution mean (MEAN criterion) and the fraction of the original Proxel probability (ORIG_PROB criterion). Both thresholds were varied between 1 and 0, where 1 resulted in too rough results and 0 was excluded for obvious reasons. Again these experiments are only a selection of the results shown in [3]. The graphs show the actual result measures for each of the models for threshold values between 0 and 1, and for better resolution of small values also between 0 and 0.3.

The convergence behavior was tested on the chain and warranty models already described in the previous section. Using the MEAN subdivision criterion, the results converge to the analytical solution, however not linearly. The solutions of the warranty model alternate around the real value in successively smaller jumps (see Figure 8) and the solution of the chain model converges as a step function (see Figure 7). This is not a problem of the criterion, but a side effect of the choice of binary subdivision of the steps. The model transitions change step size by an order of 2 only at certain values of k and stay stable in between. This behavior is not perfect, but using an intelligent choice of threshold pairs (k and $\frac{1}{2}k$) extrapolation should be possible using the MEAN criterion.

The convergence behavior of the ORIG_PROB criterion is less reliable. Even though the results do converge toward the analytical solution, this happens in

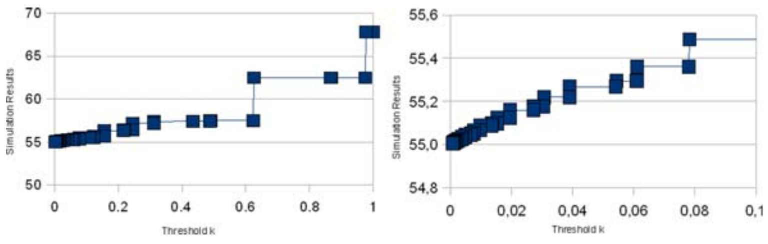

Fig. 7. Result Extrapolation for Chain Model and MEAN Criterion

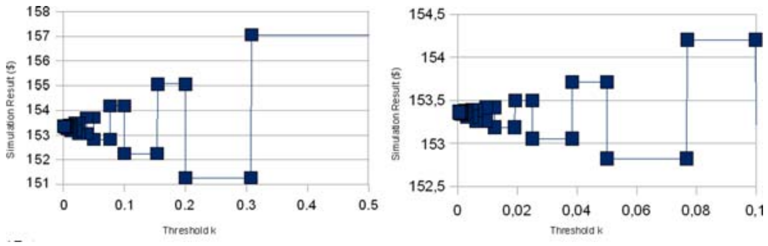

Fig. 8. Result Extrapolation for Warranty Cost and MEAN Criterion

an erratic fashion. The result of the chain model converges in seemingly unpredictable jumps (Figure 9), making extrapolation more a lottery game than reliable. The result of the warranty model seems to converge almost linearly for larger threshold values (Figure 10), the magnification of the smaller values however also shows unpredictable behavior. Therefore, extrapolation using the ORIG_PROB criterion is not reliable, an arbitrary combination of points might point to the wrong final result.

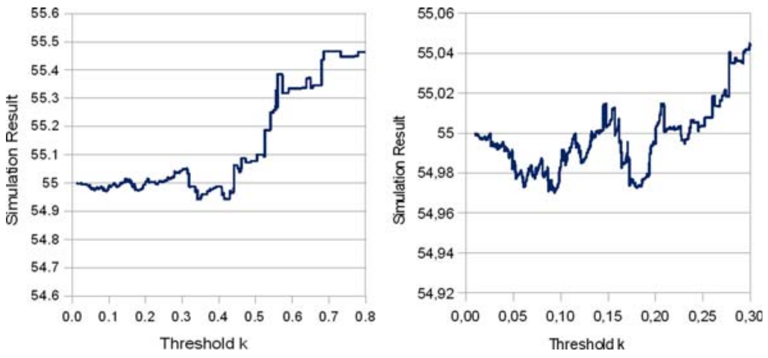

Fig. 9. Result Extrapolation for Chain Model and ORIG_PROB Criterion

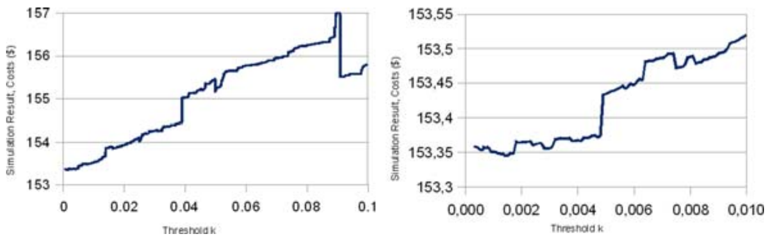

Fig. 10. Result Extrapolation for Warranty Cost and ORIG_PROB Criterion

Result discussion. The experiments on extrapolation using the threshold values of the subdivision criteria were only partially successful. The simulation results did not converge linearly for either one of the two criteria. Using the MEAN criterion, a good choice of the threshold values to extrapolate is a pair of values k and $\frac{1}{2}k$. Only then can one be sure that all model transitions are using twice the step size for k than for $\frac{1}{2}k$. The results obtained using the ORIG_PROB subdivision criterion converged erratically, and do not seem to exhibit any structure. One problem could be that the binary step division forces the transition step size to change in jumps instead of a smooth transition, however that is a key point of the VTS algorithm presented and should not be changed.

5 Conclusion and Outlook

The paper showed an approach to tackle the problem of state space explosion in Proxel-based simulation. Instead of the original constant time steps, time step size was chosen dynamically for each single transition at runtime, enabling an optimal step size in each situation. The choice for binary subdivision resulted in an efficient algorithm with little computational overhead compared to constant time steps. However, the choice of subdivision criterion is crucial to the performance. Only two of the tested criteria could reliably outperform constant time steps on the tested stiff models. The better of the two criteria could achieve the same accuracy about 100 times faster than the constant time step algorithm. Therefore the goal of the paper has been reached. Based on the experiments and further experiments in [3] the extension of individual or variable time steps increases the competitiveness of Proxels in real world applications, making the algorithm feasible for larger and more realistic models.

Future work. More work is required in the future to enhance the competitiveness of variable time steps in Proxels. A better error estimation could lead to more efficient subdivision criteria, as could a structural analysis of the model to be simulated. To dampen state space explosion further when the time steps get very small, one could start merging Proxels with similar, not just exactly matching age vectors. The choice of integration method can also affect the algorithm performance and should be observed more closely in the future.

References

1. Horton, G.: A new paradigm for the numerical simulation of stochastic petri nets with general firing times. In: Proceedings of the European Simulation Symposium 2002, pp. 129–136. SCS European Publishing House (2002)
2. Lazarova-Molnar, S.: The Proxel-Based Method: Formalisation, Analysis and Applications. PhD thesis, Otto-von-Guericke-University Magdeburg (2005)
3. Buchholz, R.: Improving the Efficiency of the Proxel Method by using Variable Time Steps. Master's thesis, Otto-von-Guericke-University Magdeburg (2008)
4. German, R., Lindemann, C.: Analysis of stochastic petri nets by the method of supplementary variables. In: Proceedings of Performance Evaluation, vol. 20, pp. 317–335 (1994)
5. Bolch, G., Greiner, S., de Meer, H., Trivedi, K.S.: Queuing Networks and Markov Chains. John Wiley & Sons, New York (1998)
6. Richardson, L.: The deferred approach to the limit. part i. single lattice. Philosophical Transactions of the Royal Society of London, Series A 226, 817–823 (1927)
7. Wickborn, F., Horton, G.: Feasible state space simulation: Variable time steps for the proxel method. In: Proceedings of the 2nd Balkan Conference in Informatics, Ohrid, Macedonia, pp. 446–453 (2005)