Otto-von-Guericke-Universität Magdeburg

Department of Simulation and Graphics

Master's Thesis

Use of Proxels in the Analysis of Rare Events

Amine Mouzannar

Magdeburg, 2018

Supervisor:

Dr. Claudia Krull
claudia@isg.cs.ovgu.de

Mouzannar, Amine:

Use of Proxels in the Analysis of Rare Events

Master's Thesis, Otto-von-Guericke-Universität Magdeburg, 2018.

Abstract

Rare events can describe many real-world problems which exist in small probability. Discrete event techniques are widely applied to analyze such rare event probability, but it requires many replications to estimate accurate results.

Importance Sampling and Importance Splitting/RESTART are popular variance reduction methods aim to increase the occurrence of rare event in the simulation to reduce the simulation runtime.

In contrast, proxel-based simulation is a state space based simulation, which discovers all system states during the simulation. The proxel-based method may be a competitive method to the variance reduction methods and suitable to analyze models containing rare events.

To show whether the proxel-based model is suitable for rare event simulation and it is a competitive method to the variance reduction methods. The proxel-based method was implemented with often studied academic rare event reliability and queuing system models.

During the implementation, some challenges were encountered where the probability cut-off was used to overcome those challenges. The proxel-based results were compared to results obtained using the RESTART method, where the proxel-based method showed promising and competitive results compared to the RESTART method.

Acknowledgements

I would like first to express my great appreciation to my supervisor Dr. Claudia Krull for her support, guidance, and a lot of feedback during my thesis and my studies.

I would like to thank my parents and my friends for the continuous support they have provided me during my studies.

Contents

List of Figures

List of Tables

Glossary

CDF	Cumulative Distribution Function
eDSPN	extended Deterministic and Stochastic Petri Net
GATS	Global Acceptable Time Step Size
IRF	Instantaneous Rate Function
IS	Importance Sampling
MC	Monte Carlo
PDE	Partial Differential Equation
PDF	Probability Density Function
RATS	Reasonable Acceptable Time Step Size
RESTART	REpetitive Simulation Trails After Reaching Thresholds
SAN	Stochastic Activity Network
SCPN	Stochastic Colored Petri Nets
SPN	Stochastic Petri Net
SPNP	Stochastic Petri Net Package

1. Introduction

This chapter represents the introduction of the thesis, which is organized as follows: Section 1.1 represents the background and the motivation of the thesis. Section 1.2 describes the thesis goals. The last section represents the thesis outline.

1.1 Motivation

Simulation is a useful modeling tool that enables to validate how a specific system evolves over time. Simulation has many different definitions that fit its main purpose. A general description can be, simulation is a modeling method to monitor, analyze, or investigate how a specific system behaves over time [RN16].

Jerry Banks defined simulation: ”Simulation is the imitation of the operation of a real-world process or system over time” [Ban05]. Moreover, Eugen Lamers described the goal of simulation with accurate estimating of a specific probability within a time interval or to obtain an approximate probability of a particular event to happen as fast as possible [Lam08].

Simulation process involves a good understanding of the complex system. However, the study of the system as it behaves over time is done using a simulation model. The simulation model is defined with the set of assumption and rules described with mathematical and logical relationships between the system components [RN16].

The simulation model helps to predict how a specific system will perform under certain conditions or even before the system is built. Thus, it is possible to predict the performance of a particular system with different conditions [RN16].

Simulation has many advantages and disadvantages. Some of them can be founded in [Ban05]. In short, main simulation advantage is to help users to understand how a system operates using certain input parameters, also "What if" questions can be answered during the simulation process. However, simulation has some limitations, for instance, some models are difficult to develop where expert knowledge is required. Furthermore, simulation can be sometimes expensive and time-consuming [Ban05].

According to Roger McHaney, simulation approaches can be categorized into four different categories [McH09]:

1. **Continuous Simulation:**

   Continuous simulation is a continuous process, where a system can be modeled by a set of differential equations to represent how the system continuously evolve over time. Predator-prey models are good example models for such simulation approach.

2. **Monte Carlo Simulation (MC):**

   Monte Carlo simulation is a probability simulation method to estimate a probability where time has no role based on the use of random numbers. Monte Carlo method rely on random sampling to estimate its results.

3. **Discrete Event Simulation:**

   Discrete event simulation analyze a system as it evolve over time where the state change of the system is represented as events in time. Discrete event simulation is described more in Chapter 2.

4. **Agent-based Modeling:**

   Mathematical modeling method of multiply connected agents to predict their interact phenomenon. In other words, a complex modeling approach to analyze complex systems, which consists of multiple interacting agents.

Some real-world problems exist in small probability; such probability can be extremely small e.g 10^{-15} or less and can result in serious consequences. For instance, failure in a nuclear power plant leads to financial and human losses [RT09, PSW05].

Some of the application areas of rare event simulation are listed in [RT09, PSW05] as follows:

- Nuclear physics, e.g. atomic accident

- Security systems, e.g. false alarms in radar
- Aircraft, spacecraft, e.g. Technical defects
- Mathematical Finance and Insurance Risk, e.g. ruins
- Manufacturing systems, e.g. breakdowns

In most of the rare event problems, the mathematical model can be complicated to be calculated analytically due to its complexity. Estimating a probability which is unlikely to happen can be time-consuming and ineffective [RT09, GUD15].

Solving rare event problems with standard discrete event simulation requires long simulation runtime. After all, it may still require many replications to estimate accurate results in the form of confidence intervals [LM05].

One way to solve such models is to make those rare events more frequent during the simulation. Importance Sampling and Importance Splitting/RESTART are popular variance reduction methods aim to make the increase the occurrence of those events during the simulation [PSW05]. Importance Sampling and Importance Splitting/RESTART are discussed in Chapter 2.

G. Horton proposed a new computational simulation approach in [Hor02]. This new approach is called Proxel-based analysis. This approach is easier to understand and implement compared to the variance reduction methods, where no differential equations are needed [LMH03b].

Proxel-based simulation discovers all system states in the simulation. For this reason, we believe that proxel-based simulation is a competitive approach to the existing rare event simulation methods, where all events in the model have the same importance [LMH03b].

For this purpose, we implement the proxel-based method to some commonly studied rare event models from the rare event simulation literature. And the results obtained using the proxel-based are compared to the one obtained by the RESTART technique (see Section 2.7).

1.2 Thesis Goals

In this section, we present the goals of this thesis, where the primary goal we are trying to achieve is to show whether that proxel-based method is a competitive method to the RESTART method.

In that manner, this thesis should answer several questions to achieve the primary goal, which is whether proxel-based is competitive. Those questions can be listed as follows:

1. How accurate the proxel-based method compared to the RESTART method?
2. Is the proxel-based method a competitive approach regarding the computational runtime?
3. Is the proxel-based method efficient to analyze non-Markovian rare event model?

Based on the answers to the above-asked questions we can show whether the proxel-based method is competitive approach to RESTART method.

1.3 Thesis Outline

The thesis is organized as follows:

- **Chapter 2**: provides the theoretical background, where we review some advantages and disadvantages of several rare events simulation approaches.
- **Chapter 3**: presents a detailed implementation done in this work.
- **Chapter 4**: discusses and evaluates the implementation results.
- **Chapter 5**: represents the summary and conclusion of this thesis, followed by the future work.

2. Background

The purpose of this chapter is to represent the general background of the thesis. The background is organized as follows:

First, we present an overview of discrete stochastic models, including Stochastic Petri Net (SPN). Second, we represent the discrete event simulation approach as a simulation approach for the rare event.

Third, we discuss the commonly used rare event simulation methods: Importance Sampling and Importance Splitting. Fourth, we present our proxel-based method followed by some of the rare event simulation tools in the next section.

Last, we discuss the rare event simulation methods reviewed in this chapter. And based on this discussion, we represent an introduction to the implementation Chapter 3.

2.1 Discrete Stochastic Models

Stochastic models are a mathematical modeling representation of a system, which can describe various real-world processes. Those stochastic models consist of connected events, where the state change in the model occurs depends on a random process. The random process can be discrete or continuous [LM05].

Example of a discrete model can be people waiting in a queue in front of an ATM, where the number people in the queue and the busy status of the ATM are considered to be discrete. Predator-prey can be an excellent example of continuous models [LM05, McH09].

The random variables are described by a distribution function, which can be modeled with the Probability Density Function "PDF" or the Cumulative Distribution Function "CDF" [LM05].

- $F_X(x)$ represent the Cumulative Distribution Function, described as:

$$F_X(x) = Pr\{X \leq x\} \tag{2.1}$$

- $f_X(x)$ represent the Probability Density Function, described as:

$$f_X(x) = \frac{d}{dx} F_X(x) \tag{2.2}$$

Instantaneous Rate Function "IRF" can represent the rate within the event might happen in a time interval τ. Instantaneous rate function is sometimes called the hazard rate function, and it is calculated as follows:

$$\mu(\tau) = \frac{f(\tau)}{1 - F(\tau)} \tag{2.3}$$

Analysis of discrete stochastic models :

Discrete stochastic models can be solved using transient and steady-state solution analysis. The difference between those two analysis type that steady-state solution is the solution as the model state at equilibrium, where no change is occurring anymore. While the transient probability describes the model as it still evolves over time in the form of probabilities before it reaches equilibrium state [LM05].

2.1.1 Stochastic Petri Nets

Stochastic Petri Nets (SPNs) are graphical modeling tool proposed by Carl Adam Petri in 1962 [Pet62]. They are a graphical and mathematical representation method to model and evaluate discrete stochastic models. SPNs can describe many complex systems in a more straightforward way using the state and state changes models [Zim07].

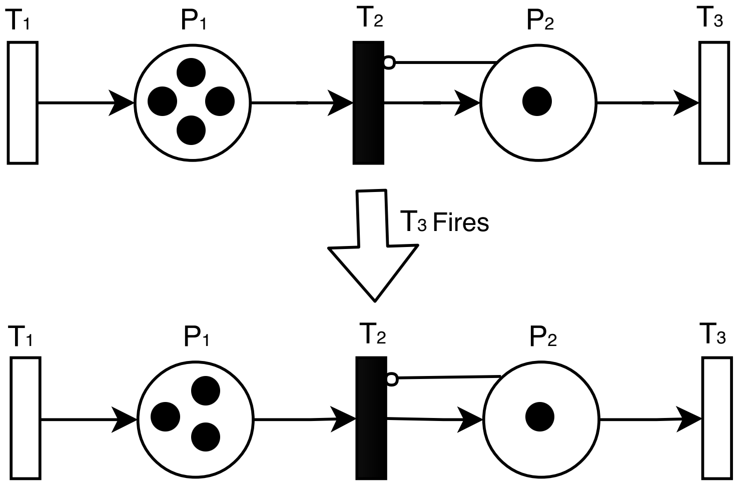

Figure 2.1: Stochastic Petri Nets model example

Figure 2.1 is a Stochastic Petri Nets model example for a single queue with one server, the figure is adapted from [LM05].

- Places are represented by circles. P_1 represent the queue and P_2 corresponds to the server.
- Transitions are drawn as bars. T_1 is a timed transition corresponds to a new customer arrives at the system. T_2 is Immediate transition corresponds to a customer move from the queue to being served when the server is free. While T_3 is a timed transition corresponds to a customer leaving the system.
- Arcs, represented by arrow connecting places and transitions
- Tokens are drawn as filled circles. In our example tokens corresponds to customers in the system.

As seen in Figure 2.1, after a customer service is completed, which is represented by transition T_3 fired. T_2 immediately fire where a customer move from P_1 to P_2.

In addition to the mentioned above, some additional rules are used to describe the model as it behaves over time. Stochastic Petri Nets represent a powerful modeling tool to model many real-world processes in simplified models [LM05].

2.2 Discrete Event Simulation

Discrete event simulation is a straightforward approach used to analyze discrete models. In other words, it is a simulation approach to model systems where the state variable evolve at discrete points of time [LM05, Ban05].

In discrete event simulation, the system is first translated to a conceptual representation, where random variable describes the events in the model. Once the system model is converted and validated, the model is then simulated several times. After the simulation, the simulation results are represented in the form of confidence interval [LM05, L'E90].

Discrete event simulation has some limitations when the model of interest contains a rare event. For instance, estimating a rare event probability requires long simulation runtime to achieve a reasonable accuracy probability [Zim07].

As overall approach discrete event simulation is a straightforward approach to understand. However, some models require a large number of replications such as in rare events models, or models with large state space [LM05, Zim07].

2.3 Importance Sampling

Monte Carlo simulation has some limitation when an event of interest has very small probability. Estimating a tiny probability with Monte Carlo algorithm requires a large number of samples to detect the occurrence of the rare event [LMT09].

Importance Sampling (IS) overcomes the limitation of Standard Monte Carlo simulation when a probability of interest is a rare event. IS is a popular variance reduction technique to estimate the rare event probability, by which it makes the rare event in a system of less rare and more frequent [LMT09].

The basic idea of Importance Sampling is to change the original probability distribution of the model by a new one to increase the occurrence of the rare event, which is more important for the simulation [GI89, LMT09].

2.3.1 Algorithm

Importance Sampling (IS) is not scope of this work (see Section 2.7). Therefore, we will not explain the whole algorithm but a more detailed description of the IS algorithm can be found in [AHO95, GW97, LMT09, AMD+].

Importance Sampling is a variance reduction technique which aims to achieve accurate estimation through short simulation time. Importance Sampling idea is to sample more often particular random variables in the simulation, which has more influence on the parameter being evaluated [SSG97].

This is done through changing the original probability distribution of the model to one which encourages the important values. The probability estimation is done using the likelihood ratio estimator. The efficiency of the Importance Sampling method is mostly relying on the choice of biased distribution [SSG97].

2.3.2 Advantages and Disadvantages

Importance Sampling is a robust approach to overcome Monte Carlo or discrete event simulation limitations when dealing with rare event. However, it has some advantages and disadvantages, some of them are listed in [Gar00, MSM09, LMT09]. Some of those advantages and disadvantages are:

1. Advantages
   - Overcomes the limitation of Monte Carlo simulation when estimating a rare event probability.
   - Can be applied to various of models.
2. Disadvantages
   - Difficult to use, selection of the optimal change of measure requires a deep knowledge of the model.

As a summary, the primary challenge to the Importance Sampling method lies in the choice of the new distribution function. Although Importance Sampling has some promising published results, but we focus on the RESTART method (see Section 2.7).

2.4 Importance Splitting/RESTART

Importance Splitting is another variance reduction technique. It was first discussed in 1951 by Herman Kahn and T. E. Harris [KH51] as an alternative approach for Importance Sampling. The idea first was to split the state space into importance regions, and a number of retrials are performed after reaching the importance region [Gar00].

In 1991, RESTART (REpetitive Simulation Trails After Reaching Thresholds) was introduced by M. and J. Villen-Altamirano in [VAVA91]. RESTART was first proposed in 1970 by Bayes [Bay70] but under the name of Importance Sampling [VAVA99, GG02].

There are some discussions whether RESTART technique belongs to the same Importance Splitting class, or it is included in the Importance Splitting technique. M.

and J. Villen-Altamiran pointed out that RESTART does not belong to the same Importance Splitting class for two reasons mentioned in Table 2.1 [VAVA99].

Importance Splitting	RESTART
Retrials continues until the end of simulation	Retrials continues until leaving the importance region
Retrials are made only at the first stored point of the main trial	Retrials are made on every stored point of the main trial

Table 2.1: Importance Splitting and RESTART Comparison

According to M. and J. Villen-Altamiran, RESTART has some advantages over the Importance Splitting method. The Importance Splitting method has some limitation where each retrial can leave the importance region during the simulation, which prevents steady-state simulation and limit this method to transient state simulation [VAVA99].

RESTART algorithm has been explained in several papers, among these papers [VAVA91, VAVA99, VAVA06, VA07, VA09, VAVA11]. RESTART method estimates the rare event probability through executing a number of retrials after reaching a specific state of interest where the rare event is more likely to occur [VAVA99].

2.4.1 Algorithm

Let's assume that S is the state space and A is the rare event. The space state is divided into nested sub sequence regions C_i and $(C_1 \supset C_2 \supset ... \supset C_M)$, whereas i is bigger then the more importance of the region. The space state is divided according to a defined importance function Φ and thresholds T_i, where $(1 \leq i \leq M)$ and $\Phi \geq T_i$ [VA09].

The choice of the importance function is defined by the thresholds, where the importance function is chosen in a way it encourages reaching the rare event after passing the thresholds T_i [VA09].

B_i and D_i are two more events defined as follows :

- B_i occurs when a system enters the importance region.
- D_i occurs when a system leaves the importance region.

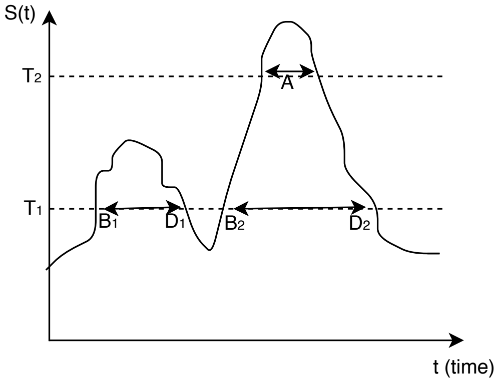

Figure 2.2: Normal simulation

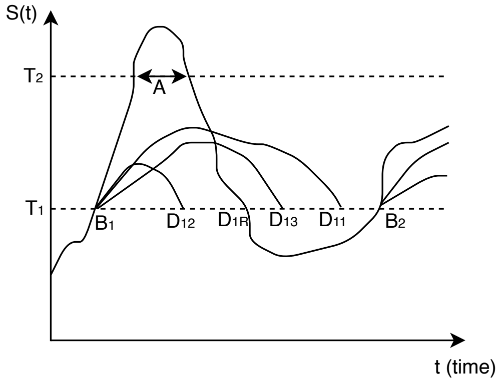

Figure 2.3: Simulation with RESTART

The rare event probability *Pr(A)* is defined in the simulation as the system's probability being in the rare set.

Figure 2.2 shows a normal simulation, where a system evolve over time while Figure 2.3 shows a simulation using RESTART. Figure 2.2 and Figure 2.3 are adapted from [VAVA91].

RESTART is described in this section based on [VAVA91, VAVA99, VAVA06, VA07, VA09, VAVA11].

RESTART works as follows:

1. The simulation proceeds until the end of the simulation, where the path is called the main trial.

2. After event B_1 happens in the main trial, the state is stored, and all retrials begin from the stored event B_1.

3. After event D_1 happens, the state is restored from the saved event B_1 and [B_1, D_1) interval is simulated again.

4. The steps (2 & 3) are performed R_1 times, each time starting with the same B_1 but with different ending event D_{11}, D_{12}, ..., D_{1R_1}.

5. After event D_{1R_i} happens, the simulation proceed until another event B_2 to happens.

6. The above process is applied as shown in Figure 2.3.

2.4.2 Advantages and Disadvantages

Although RESTART is more straightforward method to be implement on more models compared to Importance Sampling, it still has some drawbacks. The advantages and disadvantages are derived from [Gar00].

1. Advantages
   - Faster approach and easier to implement to various of models compared to the Importance Sampling method.
2. Disadvantages
   - Importance function can be complicated for some models.
   - The offspring of a split have the same history until the simulation enter another importance region.

As a summary of this section, RESTART method has advantages over the Importance Splitting method. A further comparison between RESTART and Importance Sampling will be presented in Section 2.6.

2.5 Proxel-based Simulation

A new computational simulation approach was proposed by G. Horton in [Hor02]. This new approach is called Proxel-based analysis [Hor02].

The Proxel-based approach may be an ambitious alternative approach to the discrete event simulation. It is easier approach compared to variance reduction approaches, where differential equations are not needed [LMH03b]. Proxel-based method can be applied to analyze several discrete stochastic models categories [LM05].

The proxel-based method allows tracking the probability of a particular event to happen during a specific time interval, where all the system states are discovered in the simulation. Each proxel has a certain amount of information to determine how a model behave from one step to another [LMH03b, LMH03a].

Proxel can be represented as follows [LM05]:

$$Proxel = \big((DiscreteState, AgeVector), Time, Route, Probability\big)$$

- Discrete State: describe the notation of ”tangible marking”, represented with ”m”.
- Age Vector: is the time information of a certain state change, which indicate the time a state change has been active. It is represented with ”τ”.
- Time: corresponds the global simulation time, represented with ”t”.
- Route: describe the path in which the model reached the certain state.
- Probability: is the probability of the combination of all of a discrete state, an age vector, a global simulation time, and a route(s) of being in a given state. It is represented with ”Pr”.

Example

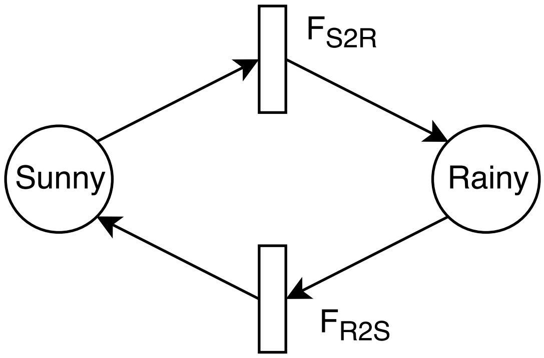

Figure 2.4: Weather model SPN

To illustrate the proxel-based method, lets assume a simple weather example (Figure 2.4) with two states: either ”Sunny” or ”Rainy”. Also, two change transitions F_{S2R} and F_{R2S} which corresponds to the state change from Sunny to Rainy or vice versa. This example is adapted from [LM05].

Figure 2.5 illustrate the weather example in a proxel tree. The proxel tree represents the proxel generation process. ”S” corresponds to state ”Sunny” and ”R” corresponds to state ”Rainy”. First, a proxel-based method starts always with an initial proxel with age vector and route equal to zero and probability equal to 1, where it is the only proxel at the time t=0 [LM05].

In the next time step, new proxels are generated from the initial proxel. If transition didn’t fire the age vector is incremented by the time step, otherwise it is reset to

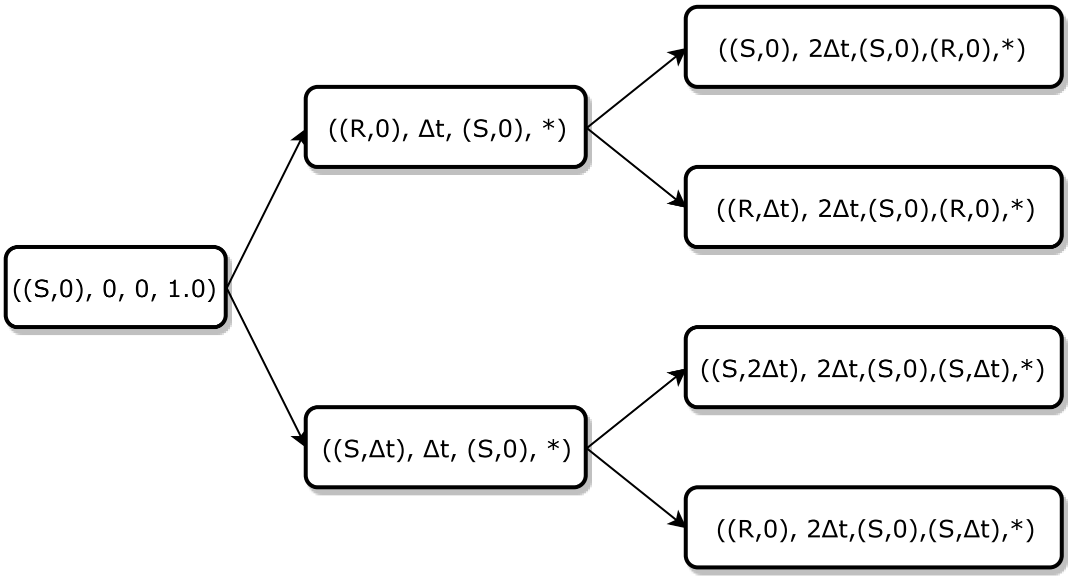

Figure 2.5: Proxel tree of the weather model

zero. The following process is repeated. In our current example, two proxels are generated from the initial proxel as shown in the figure.

Probabilities are ignored in the figure above for simplicity reasons. However, the probabilities in proxel tree are calculated as follows :

- Probability created after the transition x fired

$$Pr = \mu_x(t) \times \Delta t \times Pr_{parent} \tag{2.4}$$

  where $\mu(x)$ is instantaneous rate function.

$$\int_{t_1}^{t_1+\Delta t} \mu(x)dx \tag{2.5}$$

- Probability created after no transition fired, where $\sum_x Pr_x$ is the sum of all probabilities of all other proxel at that time are given and Pr_{parent} the probability of the parent's proxel from which proxel is generated.

$$Pr = Pr_{parent} - \sum_x Pr_x \tag{2.6}$$

2.5.1 Proxel Algorithm discussion

Determination of the Size of Δt

Choosing the time step size is an essential factor in proxel-based simulation. For small models, the decision of the time step can be chosen quickly. However, as the model gets more complex, the choice of the time step gets more complicated [LM05].

A reasonable choice should record all the state change in the system. Therefore, the decision of the time step should be taken carefully relying on the distribution functions in the model [LM05].

Some rules were suggested by Sanja Lazarova-Molnar in [LM05] to calculate the acceptable time step in a model, as well as the globally acceptable time step. For instance, the relatively acceptable time step for every state change RATS(SC) can be calculated as follows:

- Uniform(a,b) $\quad \text{RATS(SC)} = \frac{a+b}{4}$.
- Exponential(λ) $\quad RATS(SC) = \frac{1}{2\lambda}$.
- Normal(μ, σ) $\quad RATS(SC) = \frac{\mu}{2}$.
- Weibull(α, β) $\quad RATS(SC) = \frac{\alpha}{2} \times \Gamma(\frac{1}{\beta} + 1)$.

After checking the acceptable time step for every state change in the system, the global acceptable time step (GATS) can be calculated as follows:

$$GATS(M) = \min_{\forall SC \in M} RATS(SC) \tag{2.7}$$

M corresponds to the discrete stochastic model, and SC to the state change.

When choosing the time step, it should be taken into consideration that the smaller the time step, the longer runtime to reach the steady-state. However, choosing smaller time step would achieve a highly accurate solution [LM05].

Merging proxels

One feature of the proxel-based method is merging proxels. In other words, when a new proxel is being generated, and it happens that another proxel has the same discrete state and age vector in the same simulation time. In this case, both the proxels are merged into one proxel with the same discrete state and age vector while the probabilities of both proxels are added together [LM05, KBH09].

However, one important rule must be fulfilled to merge proxels, which is the route of the proxel should be ignored [KBH09]. To illustrate this feature, we consider the same weather example Figure 2.4.

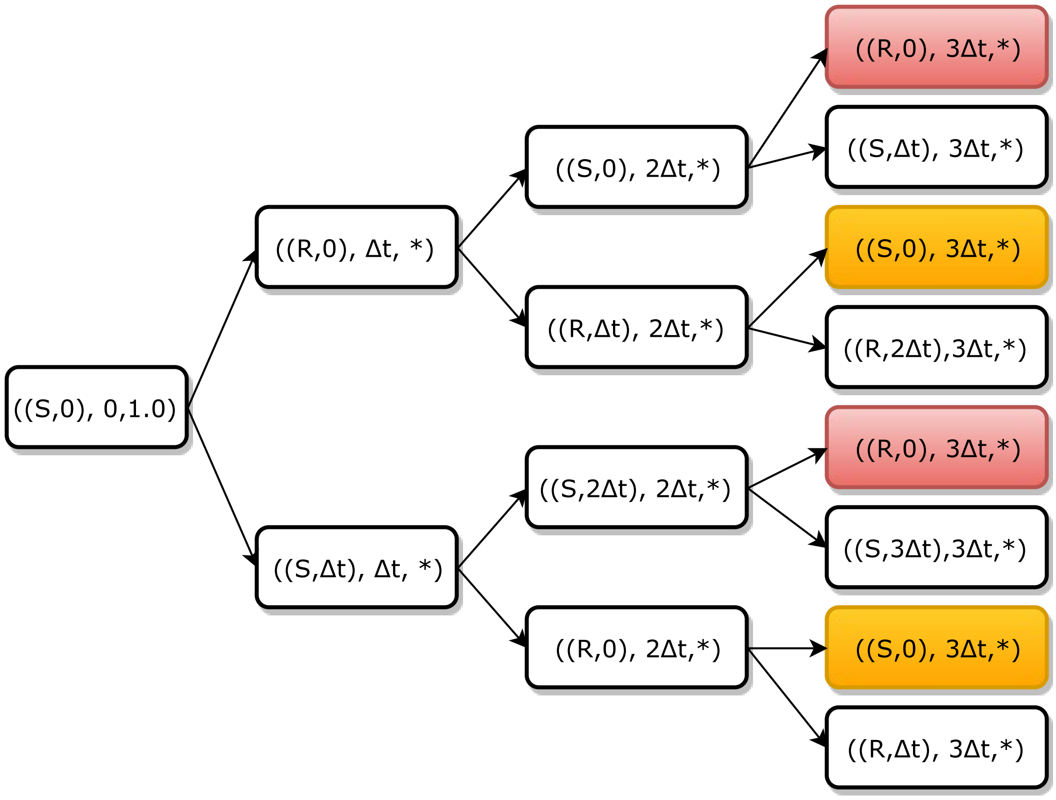

Figure 2.6: Merging proxels example

Figure 2.6 is the proxel tree generation for the weather example, whereas seen in the model the route of the proxel is avoided. Probabilities are ignored for simplicity reasons. At simulation time t= $3\Delta t$, two proxels are being generated having the same discrete state and age vector. Therefore, at simulation time $4\Delta t$, both of the proxels will be merged, and two proxels will be generated from the merged proxels.

Non-Markovian Models

Another particular feature in the proxel-based method is solving non-Markovian models. Proxel-based method turns non-Markovian models to Markovian ones by integrating the age information in the state definition. The proxel-based method calculates the probability for the state changes through extending the state to the most likely state change in time [LM05, KBH09]. Sanja Lazarova-Molnar provided an example to explain this feature in more details in [LM05].

Threshold-based pruning

The proxel-based method generates proxels exponentially with respect to the number of the discrete time step, which sometimes leads to a huge amount of proxels as the simulation time increase. This may cause complexity in the proxel-based algorithm [Hor02].

Graham Horton proposed the threshold-based pruning as a bounding method for the proxel algorithm [Hor02]. A specific probability threshold is defined for generating new proxels, where proxels are generated if the maximum simulation time is not reached and the generated proxel's probability is greater than a defined threshold [Hor02, KBH09].

Estimating very small probability using the proxel-based method requires more memory and longer computational runtime [Hor02, KBH09]. Therefore, the threshold-based pruning technique can be a useful method to support the simulation in estimate rare event probability.

2.5.2 Advantages and disadvantages

The goal of this thesis is to show whether the proxel-based method is a competitive approach to the existing rare event simulation approaches when a system of interest contains rare event. The listed advantages and disadvantages of the proxel-based method are taken from [LM05].

1. Advantages
   - All events in the simulation has the same importance.
   - Flexible and able to simulate different classes of stochastic models.
   - Differential equations are not needed.
2. Disadvantages
   - Problem with the state space of a model, known as the state space explosion.

The proxel-based method has many unique features, which make this method very flexible and easy to implement. Nevertheless, proxel has one known disadvantage that due to its deterministic nature. As seen in Figure 2.6, the state space expands on each time step, which makes solving a large-scale model using the proxel-based method not suggested [LM05].

However, more discussion about proxel advantages and limitation will be presented further in this thesis.

2.6 Rare Event Simulation Tools

In this section, we present some of the simulation tools used to analyze rare event models. Many simulation tools support designing Stochastic Petri Nets models and computing its performance analysis, such as GreatSPN [BBC+09], SPNP [HTT00], SimGine [KAAJB13], CPN tools [JKW07], Möbius [CGK+09], TimeNet [GKZH95], UltraSAN [CFJ+91], and WebSPN [LSP16].

However, among the mentioned tools few support rare event simulation, which will be discussed in this section. For rare event simulation, Importance Splitting, RESTART, or Importance Sampling (IS) are required to speed up the simulation and estimate rare event in reasonable time.

UltraSAN

UltraSAN is a simulation tool developed by the department of electrical and computer engineering, at the University of Arizona Tucson [CFJ+91]. UltraSAN tool uses stochastic activity network (SAN) to estimate a probability in a system, SAN is an extension of Stochastic Petri Nets. UltraSAN supports the simulation of the rare event with Importance Sampling [OS94].

SPNP

Stochastic Petri Net package (SPNP) is another simulation tool to analyze Stochastic Petri Nets models. It is developed at the Duke University. This modeling tool supports the simulation of non-Markovian SPNs, an analytic numerical solution of Markovian models, and Fluid Stochastic Petri Nets [HTT00]. SPNP support the simulation of the rare event using Importance Splitting [TT00].

TimeNet

TimeNet is a modeling framework which supports RESTART method [Kel96]. It was first developed in [GKZH95] as a modeling tool for Stochastic Petri Nets. The first version was developed at the Technical University of Berlin, Germany.

Nowadays, TimeNet is supported by the Software Engineering group at the Technical University of Ilmenau, Germany [Tim]. TimeNet supports the simulation of

Extended Deterministic and Stochastic Petri Nets eDSPNs, Stochastic Colored Petri Nets SCPNs, and Markov Chain. Also, TimeNet supports the simulation of the rare events using the RESTART method. TimeNet GUI is developed in Java, and some other solution modules are done in C++ and C [GKZH95, Tim].

TimeNet has many promising published results [Zim06, Zim10, ZRWL16, Zim18].

2.7 Existing Rare Event Methods Discussion

In this chapter, different rare event simulation approaches are described, and some of their advantages and disadvantages are discussed. However, rare event models are still problematic. For instance, many replications are required in discrete-event simulation so that the rare events occur, nonetheless, sometimes they never happen in the simulation [LM05].

Moreover, standard Monte Carlo simulation has limitations estimating rare event probability, where accurate estimation using the standard Monte Carlo simulation is not possible [RR08].

Importance Sampling and Importance Splitting/RESTART are widely used approaches for rare event simulation. To show that the proxel-based method is a competitive method it is necessary to create a comparison criterion.

Glasserman et al. mentioned in [GHSZ99] that Importance Sampling can estimate worse results than standard MC simulation if the change of measure was chosen improperly. Moreover, they stated that finding the right change of measure become complicated as the model of interest gets complex, and finding the right change of measure requires a rough approximation of the rare event probability [GHSZ99].

According to Manuel and José Villén-Altamirano, RESTART method is more robust and flexible approach compared to Importance Sampling, where using RESTART it is possible to set more adequate thresholds in the system also the simulation does not rely on any particular feature of the system [VAVA11].

Zimmermann et al. implemented Important Sampling and RESTART on the same system failure model in [ZRWL16]. Although both of the applied methods serve the same goal in estimating the rare event probability. However, implementing RESTART method has shown to be easier than Importance Sampling.

Marnix Garvels and Dirk Kroese compared RESTART with Importance Sampling implementation in [GK98]. The authors pointed out that RESTART is more robust estimator, where Importance Sampling is more challenging to implement and requires more optimization parameters [GK98].

As already mentioned in Section 2.6, RESTART has been implemented in TimeNet. Thus, it is easier to estimate rare event probabilities using a TimeNet graphic interface after providing the Stochastic Petri Nets model.

The proxel-based method gives all events in the simulation the same importance, where each proxel has a certain amount of information to determine how a model behave from one step to another [LM05].

A comparison of the proxel-based method and discrete event simulation was made by Horton and Lazarova in [LMH03a]. A model containing rare event was simulated using both of the simulation methods. The discrete event simulation was performed on SIMPLEX3, where the computational runtime took almost 25 minutes. In contrast, using the proxel-based method the computational runtime was 4.2 seconds using time step Δt=0.05 [LMH03a].

Nevertheless, talking about a comparison between discrete event simulation and proxel-based is not fair and not a sufficient comparison. Therefore, A further comparison will be discussed in the next chapters between RESTART method, which is widely used for rare event simulation, and the proxel-based method.

3. Implementation

The reader should be now familiar with the goal and the motivation of the thesis described in previous chapters. This chapter will provide a further discussion about the implementation done in this thesis.

3.1 Overview

Queuing system and reliability models represent commonly studied models in the rare event community. Queuing models such as single queue model and tandem queuing system are good representations for numerous technical systems, communication networks, logistics, etc [RN16, Hei95].

An important commonly studied rare event in the queuing system can be described by the long waiting time or by the buffer overload. While in the reliability models, the rare event is described as the failure of the system.

In this chapter, we explain the implementation of the proxel-based method in rare event models in queuing systems and reliability.

This chapter is organized as follows:

- **Section 3.2** describes the RESTART and proxel-based implementation of a system failure model.
- **Section 3.3** presents the proxel-based method implementation of queuing models, then in a tandem queue model.
- **Section 3.4** describes the rare event probability calculation in the all of the models discussed in this paper.

- **Section 3.5** represents a summary of the implementation, including the difficulties and challenges.

3.2 System Failure Model

System failure models are often studied models in the analysis of rare events. It can be described by the reliability of a system to remain functional under certain conditions. The rare event in such models is represented by the failure of the system, which is designed for high reliability [Hei95].

In this section, we describe both RESTART and proxel-based methods implementation of a system failure model.

Zimmermann et al. implemented both RESTART and Importance Sampling algorithms of a system failure model in [ZRWL16]. Figure 3.1 represents the model described in the paper, where N corresponds the number of components in the system, which may fail.

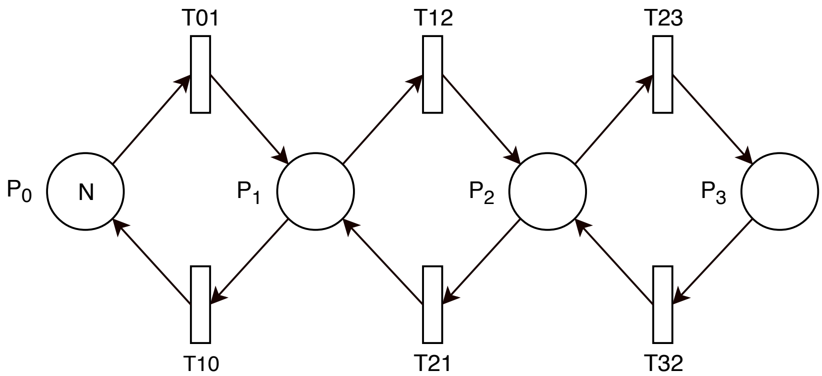

Figure 3.1: System failure model SPN

P0, P1, P2, and P3 represent the four states of the components, which describes the stage of deterioration in the system.

T01, T12, T23 represent the components failure transitions in the model. While, components repair is described with the opposite transitions T10, T21, and T32.

We are interested in the probability of the failure of the system, which is described as having at least one component in place P3.

3.2.1 RESTART Implementation in TimeNet

As explained in Chapter 2, RESTART is included in TimeNet. In this section, we present the RESTART implementation of the system failure model in TimeNet.

Repair timed transitions T10, T21, and T32 are exponentially distributed with repair rate of 1. And the failure timed transitions T01, T12, and T23 are exponentially distributed with rate ε, ε values varies between 1.0E-01 and 1.0E-03.

All timed transitions are set to be infinite server semantic, which means that the timed transitions are affected by the number of components in the system places (P0, P1, P2, P3).

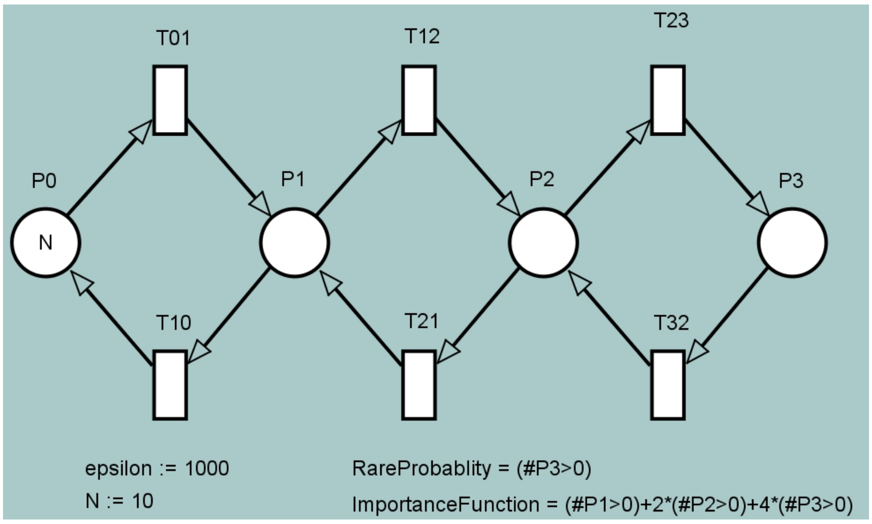

Figure 3.2: System failure model in TimeNet

Figure 3.2 is a screenshot from TimeNet. ε value in the figure is 1000 due to that TimeNet uses average delays instead of rates for all transitions, which means ε is equal to $\frac{1}{1000}$ = 1.00E-03.

The number of components N is equal to 10. Two equations measure are set by the authors: the probability of the rare event and the Importance function. The rare event is defined so that at least one component N fails, which is represented by (P3>0).

The Importance function is an indicator to show the closeness to the rare event, which is defined with (#P1>0)+2*(#P2>0)+4*(#P3>0). We will use the automatic RESTART from TimeNet to estimate the rare event probabilities.

Figure 3.3 shows the necessary steps to start a stationary RESTART simulation in TimeNet, which estimate the steady-state rare event probability using RESTART algorithm [ZK]. The splitting factor is by default equal to 4. The maximum RESTART threshold is set to 3, and the Importance function is set to be a heuristic function. The heuristic function and threshold are importance to direct the simulation to the important region of interest.

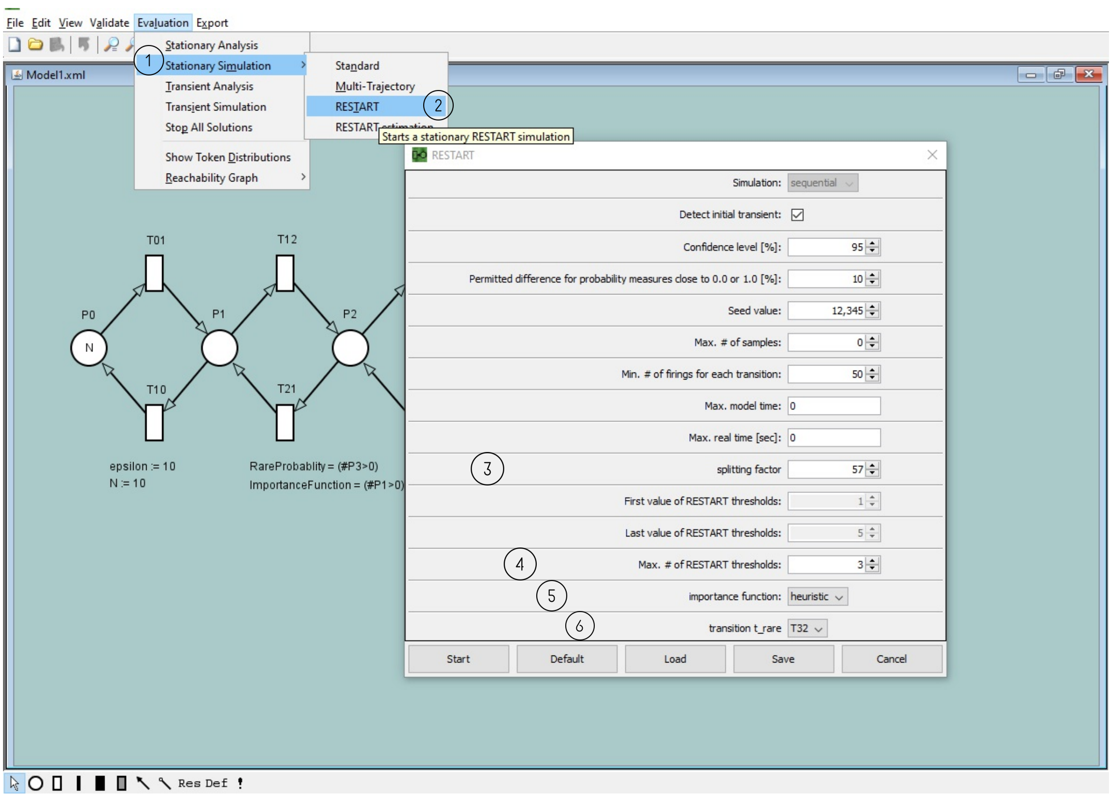

Figure 3.3: Automatic RESTART in TimeNet

3.2.2 Proxel-Based Implementation

In this section, we describe the proxel-based implementation of the system failure model.

To speed up the simulation, the route of the proxels will be avoided in the proxel-based method. This will reduce generating new proxels, where some proxels are merged as explained in Chapter 2.

The discrete state of a proxel consists of the number of components in each of the places: P0, P1, P2, and P3. All transitions in the model are exponentially distributed. Therefore, the age vector is ignored since the hazard function is constant.

As a result, proxels are represented with the number of components in each of the state places, associated with both of simulation time "t", and the probability "Pr". Proxels are represented as follows:

$$Proxel = \big((P0, P1, P2, P3), t, Pr\big)$$

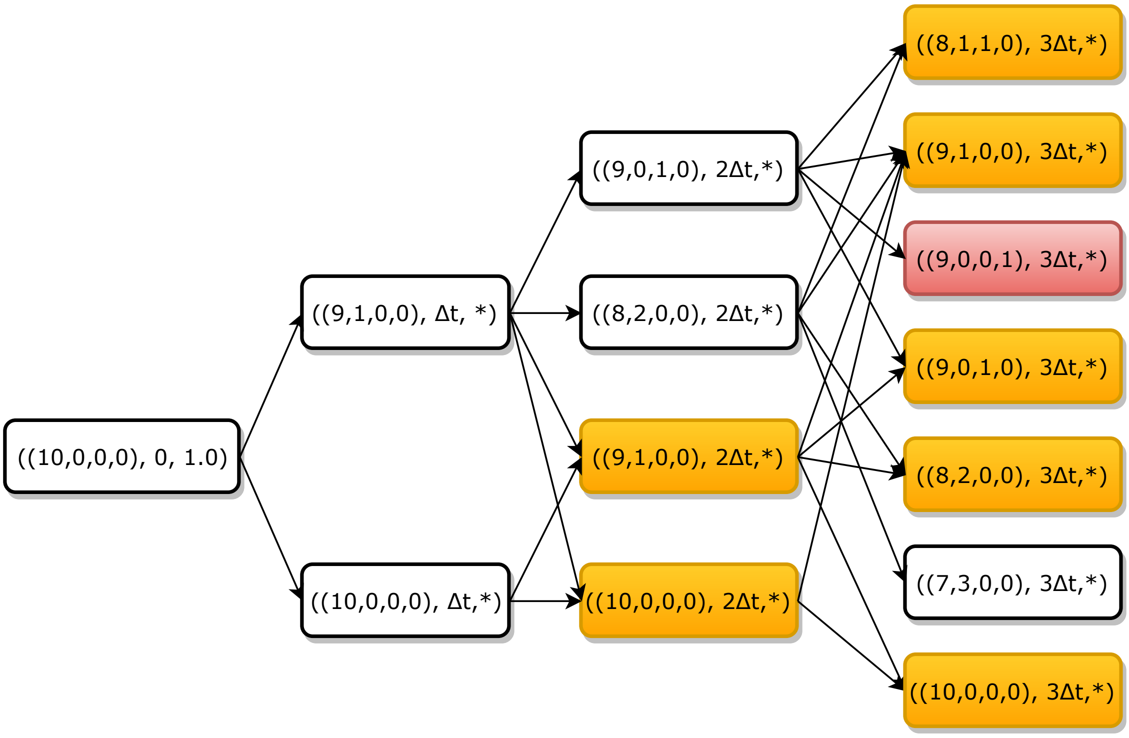

Figure 3.4: Proxel tree of the system failure model

Figure 3.4 shows the proxel generation process from the initial state until $t = 3\Delta t$. Probabilities are ignored for simplicity purpose. As already discussed in Chapter 2, proxel-based method starts always with an initial proxel with simulation time t=0 and probability equal to 1.0. At initial state, we have 10 tokens only at P0. Initial proxel is represented as follows:

$$Proxel_{initial} = \big((10, 0, 0, 0), 0, 1.0\big)$$

As seen in Figure 3.4, after one time step, at simulation t= Δt, either transition T01 fires, where one token move from P0 to P1, or no transitions fire. Yellow colored proxels represent the merged proxels at the current discrete time step.

The rare event probability is defined with the probability of failure of the system, which is represented with having at least one token in P3 (#P3>0). The rare event first occurs at simulation t = $3\Delta t$, which is highlighted in the figure with red color.

To illustrate the probability calculation, let's consider the rate of the transitions T01, T12, and T23 to be α, and the rate of the transitions T10, T12, and T32 to be ε. In this model, all timed transitions are set to be infinite server semantic, that is mean when a transition fires, the number of tokens from the place fired is multiplied in the probability.

Probabilities are calculated as follows:

$$Pr_{repair} = m \times \Delta t \times \alpha \times Pr_{parent}$$

$$Pr_{failure} = m \times \Delta t \times \varepsilon \times Pr_{parent}$$

$$Pr_N = Pr_{parent} - \sum_x Pr_x$$

- Pr_{repair} corresponds to the probability of a proxel created when repair transition fired.
- $Pr_{failure}$ corresponds to the probability of a proxel created when failure transition fired.
- Pr_{parent} corresponds to the parent's proxel probability, from which the proxel is generated.
- m is the current number of components in a place before the transition fired.
- Pr_N corresponds to the probability of a proxel created when no transitions fire at that given time.
- $\sum_x Pr_x$ is the sum of probabilities of all proxel created of a common proxel parent at a given time.

In this section, we describe both of the RESTART and proxel-based method implementation of a system failure model, where the RESTART implementation was done in TimeNet. The experiment results of the current implementations will be discussed in Chapter 4.

3.3 Queuing Systems Models

In this section, we present a further discussion of the proxel-based implementation in queuing systems. According to Kendall's Notation, single queue process can be represented with a notation of five elements A/B/X/Y/Z, which describes the queuing model system [KH07, Gro08].

In our work, we represent the single queue models with the first the three elements: A/B/C, where A is the inter-arrival time distribution, B the service time distribution, and C the number of servers in the system. A and B can be described with either M for Markovian or G for General.

This section consists of two main study cases as follows:

- Single server queuing models
- Tandem queue model

3.3.1 Single Server Queuing Models

A single server queuing system is another often studied model in the analysis of rare events. Figure 3.5 shows the conceptual representation of the queuing model, where the system model consists of the number of arrivals system and a single server which can serve only one customer at time [BGdMT06].

For explanation reasons, let's assume that the arrival and service are exponentially distributed. Customers arrive at the queue with an arrival rate of λ and leave the system with a service rate of μ. The system can process only one customer at a time. It has only two states "Ready" or "Busy [YV04, BGdMT06].

For queue stability, the service rate should be greater than the arrival rate $\lambda < \mu$. The traffic load on the queue is defined $\rho = \lambda/\mu$ [YV04].

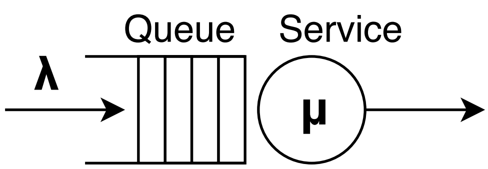

Figure 3.5: Single server queuing system model [YV04]

The rare event is defined as the system reach a level "L" of waiting customers at the queuing system during a busy interval of time.

3.3.1.1 M/M/1 Model

In this section, we consider a Markovian single server queuing system. Customers arrive with exponentially distributed arrival rate λ and leave with an exponentially distributed service rate μ.

As the transitions in this model are exponentially distributed, that's means the hazard rate function is constant. Therefore, proxels are represented as follows:

$$(S, t, Pr)$$

S is the number of customers at the system, t corresponds to the global simulation time, and Pr to the probability of being in the current state.

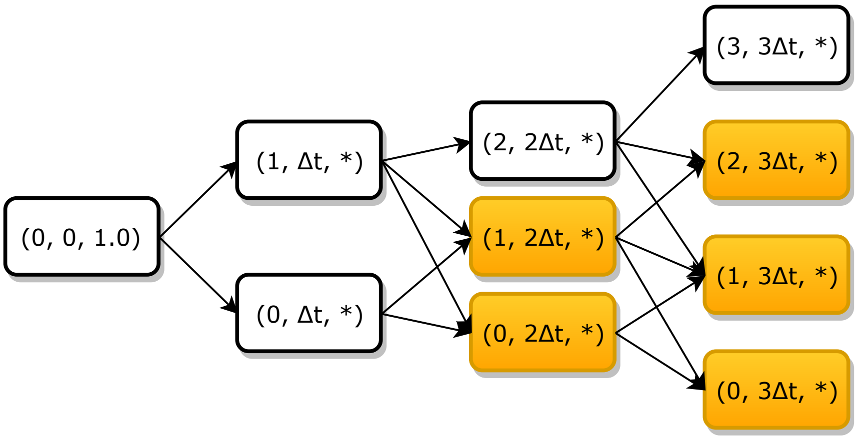

Figure 3.6: Proxel tree of the M/M/1 model

Figure 3.6 shows the proxel generation process starting from the initial state until simulation t=3Δt. Probabilities in the figure are ignored for simplicity reasons. At initial proxel, there are no customers at the system.

To illustrate the model process, let's consider one time step after the initial state. After the initial proxel, we have two possibilities: the arrival of a new customer or no arrival of a new customer. Yellow colored proxels corresponds to the merged proxels.

The rare event is defined with the probability of overflow of the system, which corresponds to the probability of having a number of customers at the system exceed a certain threshold "L".

Probabilities are calculated as follows:

$$Pr_{arrival} = Pr_{parent} \times \Delta t \times \lambda$$

$$Pr_{service} = Pr_{parent} \times \Delta t \times \mu$$

$$Pr_N = Pr_{parent} - (Pr_{arrival} + Pr_{service})$$

- $Pr_{arrival}$ corresponds to the probability of a proxel created when a customer arrives at the system, and λ is the arrival rate.
- $Pr_{service}$ corresponds to the probability of a proxel created when a customer leaves the system, and μ is the service rate.
- Pr_{parent} corresponds to the parent's proxel probability, from which the proxel is generated.
- Pr_N corresponds to the probability of a proxel created when no transition fire at that given time.

In this section, we describe the proxel-based method implementation of a Markovian single server queuing system. The purpose of the proxel-based implementation of the M/M/1 model is to validate our proxel-based code with the M/M/1 model analytic results.

3.3.1.2 G/G/1 Model

In this section, we describe the proxel-based implementation of a non-Markovian single server queuing system. As mentioned in Chapter 2, proxel-based method solve non-Markovian models by converting them to Markovian one. The probability of the transition to fire within a time step is determined by the hazard function.

The proxel state in our current model is determined with (S, τ_1, τ_2). S is the number of customers at the system, τ_1 and τ_2 represent the age vector of the arrival and service transitions respectively.

Proxels are represented as follows:

$$\big((S, \tau_1, \tau_2), t, Pr\big)$$

Figure 3.7 show the proxel generation process starting from the initial state until t=2Δt with time step of Δt. Proxel starts with initial proxel with no customers at the system with a probability of 1.0. Customers arrive and leave the system according to defined distributed functions. Probabilities in the figure are ignored for simplicity reasons.

Once there is at least one customer in the system, in the next time step three proxels are generated as follows:

- A Customer arrives to the system, arrival age vector is reset, while service age vector is incremented by the time step.

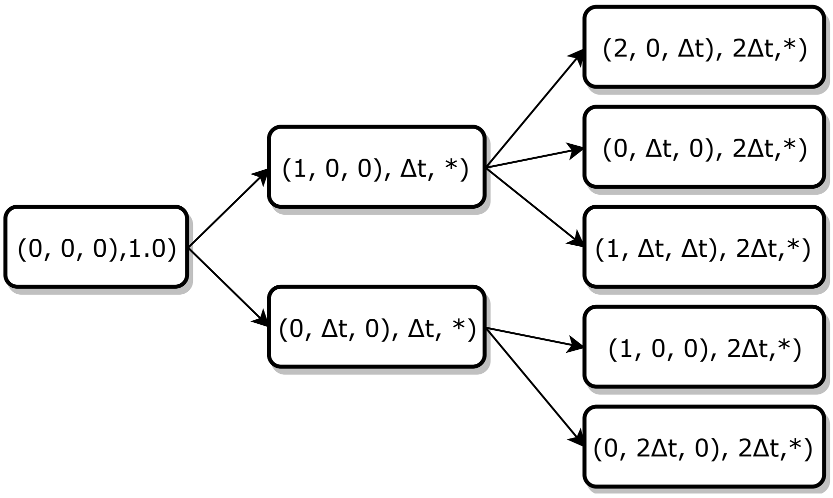

Figure 3.7: Proxel tree of the G/G/1 model

- A customer leaves the system, and service age vector is reset, while arrival age vector is incremented by the time step.
- no state change, both age vectors are incremented by the time step.

In this model, more proxels are generated on every simulation time step compared to the M/M/1 model (Figure 3.6). For instance, at simulation time t=2Δt in M/M/1 model only three proxels are generated on that time while in G/G/1 model, five proxels are generated. That is due to the discrete state in the M/M/1 model consists of the number of customers in the system only. By contrast in the G/G/1, the discrete state of the model consists of the number of customers in the systems associated with the age vector of the arrival and service respectively.

3.3.2 Tandem Queue Model

Tandem queue systems are often studied queuing model in rare event literature, that's due to the possibility of estimating various rare events in the model [GK98, VAVA99]. The rare event of interest in this model is defined with the overflow of the queue.

Figure 3.8 represents the conceptual representation of the tandem queue model. Customers arrive at the first queuing system with an arrival rate of λ, and after being served with service rate μ_1 they enter another queuing system with service rate μ_2.

The traffic load on the each queue is defined $\rho_i = \lambda/\mu_i$.

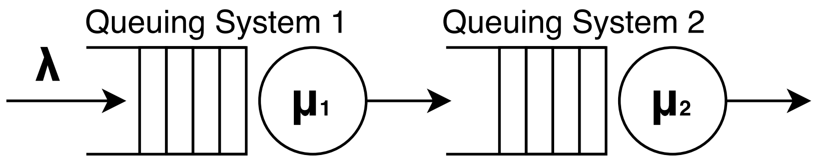

Figure 3.8: Tandem queue model [VAVA99]

In tandem queue systems, the discrete state consists of the number of customers at the first queuing system q_1 and in the second queuing system q_2:

$$(q_1, q_2)$$

Three definitions of the rare event will be evaluated using the proxel-based method in this section as follows:

- $q_2 \geq L$

  First case, the rare event is defined as the number of customers at the second queue system is greater or equal to a threshold "L".

- $(q_1 + q_2) \geq L$

  Second case, the rare event is defined as the number of customers in both of the queuing systems is greater or equal to a threshold "L"

- $\min(q_1, q_2) \geq L$

  Last case, the rare event is defined as the minimum number of customers at the first or second queuing systems is greater or equal to a threshold "L". In other words, the rare event is represented as ($q_1 \geq$ L) $\cap(q_2 \geq L)$.

The arrival rate and service rate in both queuing systems are exponentially distributed. Therefore, proxels are represented as follows:

$$\big((q_1, q_2), t, Pr\big)$$

(q_1, q_2) corresponds to the number of customers at the first and second queuing systems respectively, t corresponds to the simulation time, and Pr probability of being in the current state.

Figure 3.9 represents the proxel tree generation starting from the initial proxel until simulation time $2\Delta t$. Probabilities are ignored in the model for simplicity reasons.

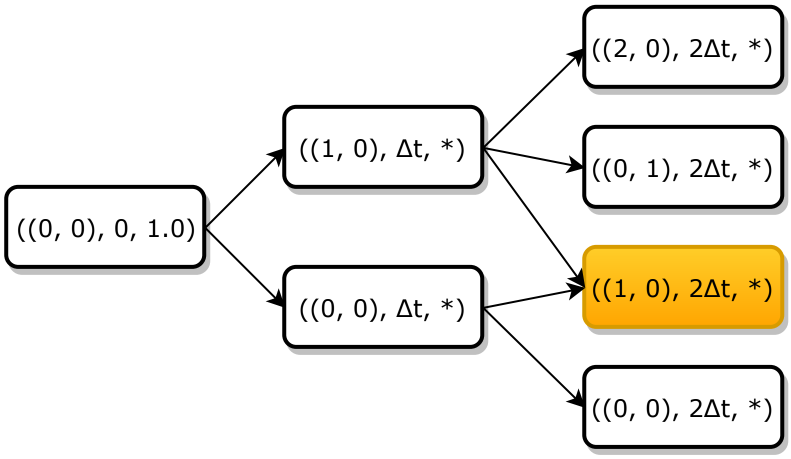

Figure 3.9: Proxel tree of the tandem queue model

Starting from the initial proxel, there are no customers at both of the queuing systems, that's means at the next time step t=Δt either one customer arrives at the first queuing system or no customer arrives.

Let's consider the state with one customer at each of the queuing systems (1,1). After one time step, we have 4 proxel possibilities:

- One customer arrives to the first queue (2,1).
- One customer is served by μ_1 and arrives to second queuing system (0,2).
- One customer is served by μ_2 and leave the second queuing system (1,0).
- No events was fired (1,1).

Probabilities in the model are calculated as follows:

$$Pr_{arrive} = Pr_{parent} \times \Delta t \times \lambda$$

$$Pr_{Service1} = Pr_{parent} \times \Delta t \times \mu_1$$

$$Pr_{Service2} = Pr_{parent} \times \Delta t \times \mu_2$$

$$Pr_N = Pr_{parent} - (Pr_{arrive} + Pr_{Service1} + Pr_{Service2)}$$

- Pr_{arrive} corresponds to the probability of a proxel created when a customer arrives at the first queuing system, with an arrival rate of λ.
- $Pr_{Service1}$ corresponds to the probability of a proxel created when a customer is served by the first queuing system with a service rate of μ_1 and then enter the second queuing system.

- $Pr_{Service2}$ corresponds to the probability of a proxel created when a customer is served by the second queuing system with a service rate of μ_2 and then leave the system.
- Pr_{parent} corresponds to the parent's proxel probability, from which the proxel is generated.
- Pr_N corresponds to the probability a proxel created when no transitions fire at that given time.

In this section, we describe the proxel-based implementation of a tandem queue model with three different rare event study cases. The experiment results of the current implementation will be presented in Chapter 4

3.4 Rare Event Probability Calculation

In all of the above models, the rare event probability is calculated on each of the simulation time steps as the sum of specific proxel's probability. For instance, in system failure model proxels with (#P3>0) are added to form the rare event probability for the current discrete time state. While in the queuing models, proxels having customers at the system higher than a specified threshold forms the rare event probability for the current discrete time state.

During the proxel simulation, the rare event probability is added together on each time step to form the rare event transient probability. All of the transient probabilities are added into a one-dimensional array. At the end of the simulation, the one-dimensional array contains the transient probabilities and the steady-state probability.

3.5 Implementation Challenges

Due to that, all models of interest in this thesis are rare event models, where rare event probability can be extremely small. Estimating very small probabilities would require more memory and longer computational runtime.

To achieve better accuracy, it is necessary to use the threshold-based pruning bounding technique explained in Chapter 2. The threshold-based pruning technique will be used in all the discussed above models. Proxels are generated if the maximum simulation time is not reached and the generated proxel's probability is higher than the selected probability cut-off value.

In models with no rare event, choosing a probability threshold can be easy. On the contrary, when the model of interest contains rare event, selecting such bounding threshold can be tricky and complicated. Different probabilities threshold were tested in the above models to achieve better accuracy.

Time step as well plays an essential role in the simulation runtime and the accuracy of the simulation. It is recommended to choose a smaller time step then the globally acceptable time step explained in Section 2.5.1. We tested several time step sizes for the above models to examine how time step can affect the accuracy and the simulation runtime.

As a summary, a combination of time step size and probability threshold should be selected accurately to avoid longer runtime and in the same manner to achieve high accuracy.

4. Evaluation

In this chapter, we analyze the results obtained from the implementation done in the previous chapter.

4.1 Overview

In this chapter, we compare the results obtained using the proxel-based method with other results obtained using the RESTART method. For the comparison, the experimental error will be used to evaluate/calculate the accuracy of our implementation.

The proxel-based code is written in Java language. All experiments are done with a personal laptop with setup settings as follows:

* Intel i7-7500U CPU 2.7GHz
* 8GB RAM
* Microsoft Windows 10
* Java 1.8.0

This chapter is organized as follows:

- **Section 4.2** compares the results of both RESTART and proxel-based methods in the system failure model.
- **Section 4.3** represents the results of the proxel-based implementation in three different single queue models.
- **Section 4.4** compares the results of the proxel-based implementation in a tandem queue model.

- **Section 4.5** discusses the effect of time step size in the proxel-based method.
- **Section 4.6** discusses the effect of probability cut-off in the proxel-based method.
- **Section 4.7** represents the summary and conclusion of this thesis, followed by the future work.

4.2 System Failure Model

In this section, we compare the results of the system failure model using both of the RESTART and proxel-based methods. However, both simulation methods were run on a personal PC with the mentioned above setup.

Table 4.1 and Table 4.2 shows the computational runtime and accuracy for both RESTART and proxel-based methods. Several failure rates ε values were tested, where ε values and the analytic results were obtained from [ZRWL16].

ε	Analytic result	RESTART result	Time (sec)	% error
1.00E-01	8.9645E-03	9.0229E-03	1	0.65 %
2.00E-02	7.8397E-05	7.70516E-5	2	1.72 %
1.00E-02	9.9000E-06	9.8349E-6	4	0.66 %
2.00E-03	7.9840E-08	7.9164E-8	10	0.8 %
1.00E-03	9.9900E-09	9.3630E-9	43	6.27 %

Table 4.1: RESTART results for the system failure model

ε	Analytic result	Proxel result	Time (sec)	% error
1.00E-01	8.9645E-03	8.9645E-3	0.1	0 %
2.00E-02	7.8397E-05	7.8397E-5	0.04	0 %
1.00E-02	9.9000E-06	9.8999E-6	0.04	0.001 %
2.00E-03	7.9840E-08	7.9839E-8	0.04	0.001 %
1.00E-03	9.9900E-09	9.989E-09	0.04	0.01 %

Table 4.2: Proxel-based results for the system failure model

To obtain accurate results, we perform several runs in TimeNet. The splitting factor was modified according to ε values, where it is important to increase the splitting factor to achieve higher accuracy with smaller ε values.

As overall results, RESTART method shows high accuracy with a short computational runtime. While the proxel-based method took milliseconds to achieve results with almost 0 % error.

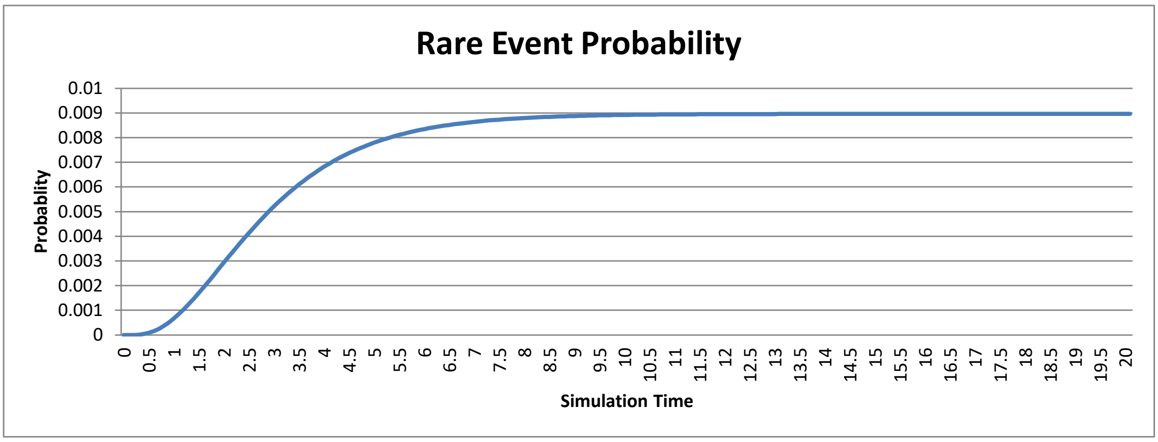

Figure 4.1: Transient probabilities for ε= 1.00E-01 in the system failure model

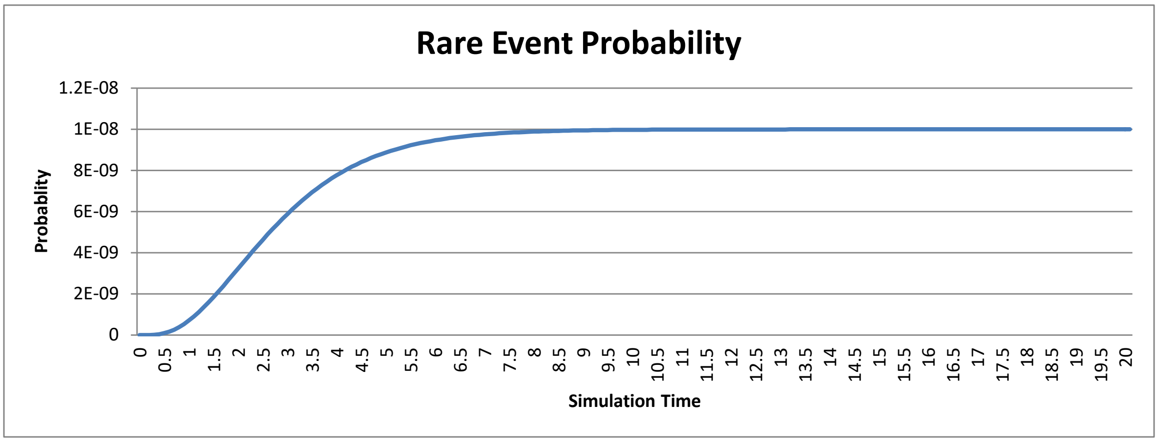

Figure 4.2: Transient probabilities for ε= 1.00E-03 in the system failure model

The computational runtime in the proxel-based method was calculated as the time needed until the steady-state was reached. Figure 4.1 and Figure 4.2 show the rare event transient probability until reaching the steady-state using the proxel-based method for ε= 1.00E-01 and ε= 1.00E-03 respectively.

In all the experiment cases the time step size Δt =0.1 and a probability cut-off of 1.00E-15 were used. All proxels with probability less than 1.00E-15 were discarded from the simulation.

As shown in Figure 4.1, the simulation converges after simulation time t=10. While in Figure 4.2, the simulation converges after simulation time t=9. It has been noticed that in all the ε values the system start converging after short simulation time. That's because that proxel-based method reaches the rare event in the same simulation time in all ε values.

4.3 Single Queue Models

In this section, we discuss the results of three different proxel-based implementations of single server queuing system models. The purpose of this section is to estimate a rare event probability for a non-Markovian model using the proxel-based method.

In the first model, we validate our proxel-based code, where we estimate the rare event probability in a Markovian single queue system, which has analytic results.

After the validation of our proxel code, we estimate using our proxel-based code the rare event probability in two other models, which don't have analytic results.

4.3.1 M/M/1 Model

In this section, we discuss the results obtained in the Markovian single server queuing system model Figure 3.5. A simple single queue proxel-based implementation was done to validate our proxel-based code, where analytic results were calculated using Pr=$\rho^{\frac{L}{2}}$ [VARLE16].

We assume that customers arrive at the system with arrival rate λ=1 and leave the system with service rate μ= 2. The traffic load $\rho = 0.5$. We calculated the rare event probability for different thresholds L (20, 40, 60, 100) using the proxel-based code, then we compared our result to the analytic results.

Table 4.3 shows the simulation results using the proxel-based method in the M/M/1 model. As seen in the table, we achieved an accuracy with 0 % error in computational runtime less than 1 second for all of the different thresholds. Time step size used equal to 0.1 in all the L cases.

Figure 4.3 shows the transient probability of having 20 customers at the system until reaching the steady-state. As seen in the graph, the probability converges after simulation time t= 55 with probability equal to 9.536E-07 with a computational runtime of 0.06 seconds.

Different probability cut-off values were used in all of the above thresholds experiments varying from 1.00E-15 to 1.00E-50. However, using a probability cut-off of 1.00E-50 for all of the threshold experiment will slightly affect the simulation computational runtime. For instance, using a probability cut-off of 1.0E-50 for estimating

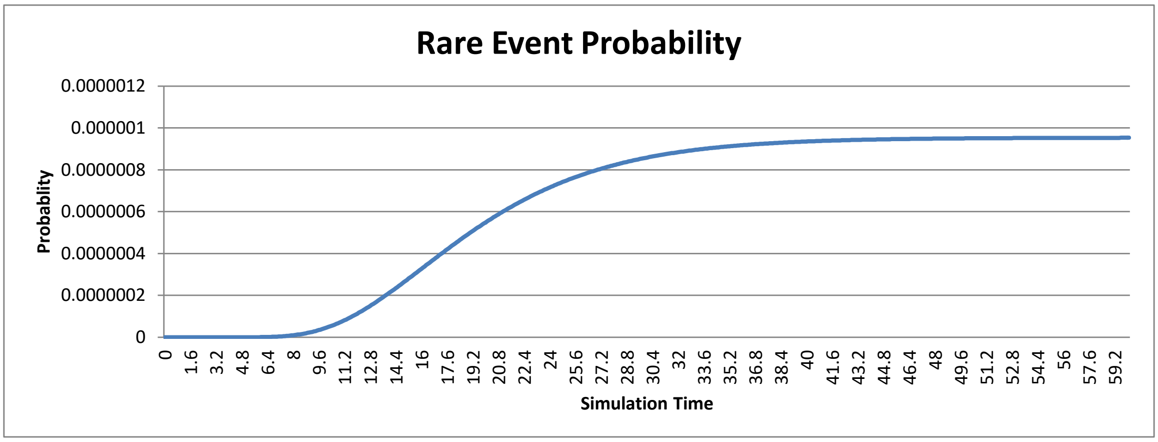

Figure 4.3: Transient probabilities for L=20 in the M/M/1 model

L	Analytic result	Proxel result	Time (sec)	% error
20	9.536E-7	9.536E-07	0.06	0 %
40	9.094E-13	9.094E-13	0.1	0 %
60	8.673E-19	8.673E-19	0.15	0 %
100	7.888E-31	7.888E-31	0.2	0 %

Table 4.3: Proxel-based results for M/M/1 model

that 20 customers in the system achieve the same accuracy with a computational runtime of 0.12.

4.3.2 G/M/1 Model

In this model, we use proxel-based method to estimate the rare event probability in a model, that doesn't have any analytic results.

Assuming that customers arrive with normally distributed intervals time [1;0.4] and the service is exponentially distributed with a rate of 3. In all the experiment cases the time step size Δt=0.05. The rare event is defined as the number of the customers in the system are equal or greater than a certain threshold.

L	Proxel result	Time (sec)
5	2.66E-05	8
10	2.12E-10	34
15	1.68E-15	107
20	1.34E-20	227

Table 4.4: Proxel-based results for G/M/1 model

Table 4.4 shows the simulation results for the proxel-method with different L thresholds (5, 10, 15, 20). The table shows longer computational runtime compared to the M/M/1 model, that's due to the non-Markovian service in the model.

Figure 4.4 shows the transient probability of having 10 customers in the systems until reaching the steady-state. As seen in the graph, the probability converges after simulation time t= 15 with a probability of 2.66E-05.

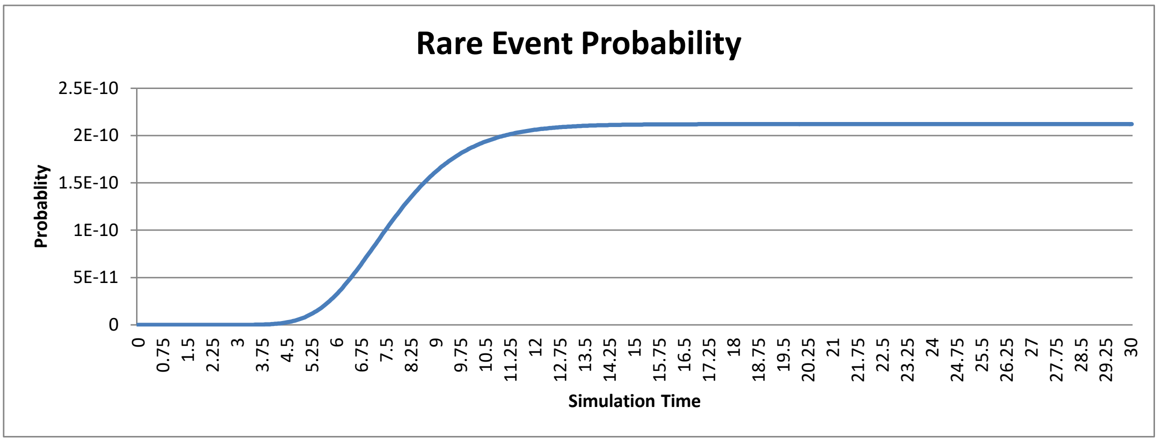

Figure 4.4: Transient probabilities for L=10 in G/M/1 model

The computational runtime in this experiment relies mainly on the probability cut-off values. This means, using a small probability cut-off would increase the computational runtime of the simulation. Several probability cut-off values were tested. For instance, estimating a probability of having 10 customers in the system using a probability cut-off of 1.00E-30 would increase the computational runtime from 34 seconds to 175 seconds.

4.3.3 G/G/1 Model

This is another example, where we use our proxel-based method to estimate the rare event probability in a model that doesn't have any analytic results.

Let's assume that customers arrive and leaves with normally distributed intervals time [2;0.4] and [1.5;0.2] respectively. The rare event probability is defined as the number of customers in the system are equal to or greater than a certain threshold L (5, 10, 15, 20). In all the experiment cases, the time step size was Δt=0.05.

Similar to the previous G/M/1 model, Table 4.5 shows the simulation results for the proxel-method with different thresholds L values. As seen in the table, the

L	Proxel result	Time (sec)
5	2.09E-13	11
10	5.84E-32	135
15	1.63E-50	360
20	4.55E-69	688

Table 4.5: Proxel-based results for G/G/1 model

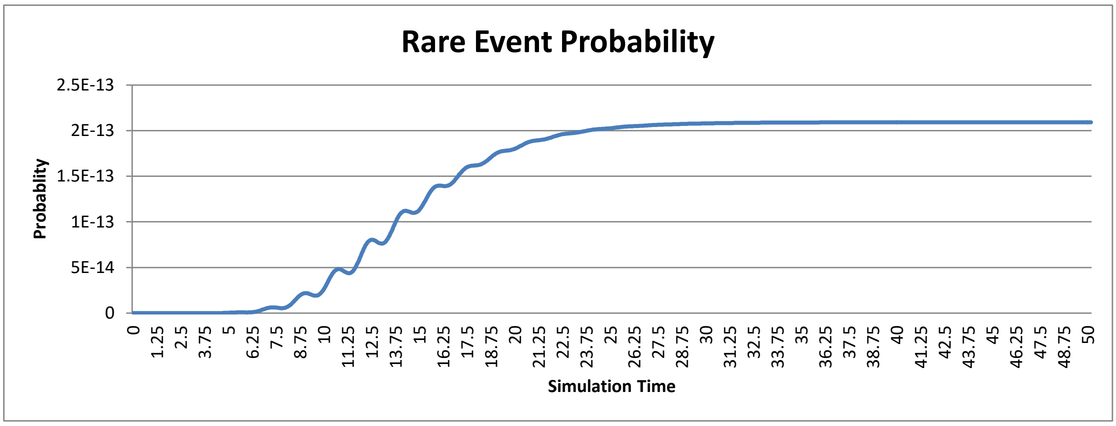

Figure 4.5: Transient probabilities for L=5 in the G/G/1 model

computational runtime is longer compared to the previous single queue models due to the non-Markovian arrival and service distribution.

Moreover, the proxel-based method shows the ability to estimate a very small rare event probability up to 5.55E-69 in a computational runtime of 688 seconds. Several probability cut-off values were tested to obtain better accuracy and short runtime in each of the L experiments.

Figure 4.5 shows the transient probability of having 5 customers in the systems until reaching the steady-state. As seen in the graph, the probability converges after simulation time t= 30 with a probability of 2.09E-13.

4.4 Tandem Queue Model

This section presents the result of the proxel-based implementation for a tandem queue model. For comparison reasons, we adapted RESTART results from [VAVA99]. To be able to compare the RESTART results to our proxel-based method, we used the same given numerical values in [VAVA99].

In this experiment, we focus our comparison on the accuracy of both simulation approaches. That's due to RESTART results were obtained using a Sun Ultra 5 workstation and our proxel-based implementation was run on a local PC.

Let's assume that customers arrive at the first queuing system with arrival rate λ= 1 and served with a service rate μ_1= 2, after being served from the first queuing system they enter another queuing system with service rate of μ_2= 3.

The traffic load on the each queue ρ_1 and ρ_2 is 0.5 and 0.33 respectively. Several probability cut-off values were tested to obtain better accuracy and short runtime in each of the L experiments. Time step size used in all the discussed cases Δt=0.1. The analytic results below were obtained from [VAVA99].

1. $q_2 \geq L$

   The rare event is defined as the number of the customers at the second queuing system are greater or equal to a certain threshold. The thresholds values L varies are 20, 40, and 60.

   Table 4.6 and Table 4.7 show the rare event probability with both simulation methods. Proxel-based method achieves a higher accuracy compared to the RESTART method with short computational runtime.

L	Analytic result	RESTART result	Time (sec)	% error
20	2.87E-10	2.92E-10	34,2	1.7 %
40	8.22E-20	8.10E-20	900	1.4 %
60	2.36E-29	2.39E-29	18600	1.25 %

Table 4.6: RESTART results for the tandem queue model for $q_2 \geq L$ [VAVA99]

L	Analytic result	Proxel result	Time (sec)	% error
20	2.87E-10	2.86E-10	2	0.3 %
40	8.22E-20	8.22E-20	35	0 %
60	2.36E-29	2.358E-29	231	0,08 %

Table 4.7: Proxel-based results for the tandem queue model for $q_2 \geq L$

   Figure 4.6 shows the transient probability of having 20 customers at the second queuing system until reaching the steady-state. The probability converges after simulation t =50 with a probability of 2.87E-10.

2. $(q_1 + q_2) \geq L$

   The rare event is defined as the number of customers in both queuing systems is greater than or equal to a certain threshold.

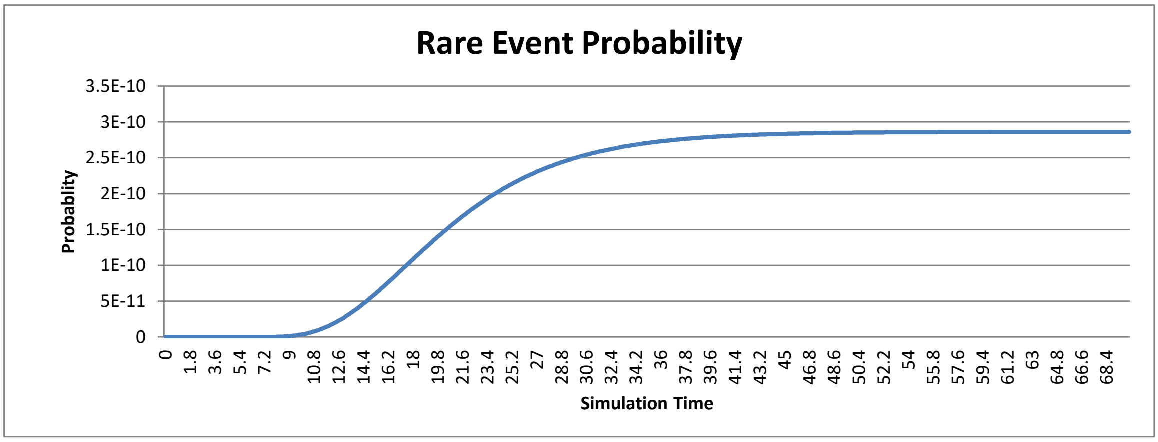

Figure 4.6: Transient probabilities for L=20 in the tandem queue model

Table 4.8 and Table 4.9 shows the rare event probability with both simulation methods, where the proxel-based method achieve accuracy of 0 error %.

L	Analytic result	RESTART result	Time (sec)	% error
20	1.90E-06	1.94E-06	4.2	2.1 %
60	1.73E-18	1.63E-18	52	6.13 %
100	1.58E-30	1.51E-30	161	4.4 %

Table 4.8: RESTART results for the tandem queue model for $(q_1+q_2) \geq L$ [VAVA99]

L	Analytic result	Proxel result	Time (sec)	% error
20	1.90E-06	1.90E-6	1	0 %
60	1.73E-18	1.73E-18	35	0 %
100	1.58E-30	1.57E-30	270	0.63 %

Table 4.9: Proxel-based results for the tandem queue model for $(q_1 + q_2) \geq L$

Figure 4.7 shows the transient probability of having 20 customers in both queuing systems until reaching the steady-state. The probability converges after simulation t=50 with probability of 1.90E-06.

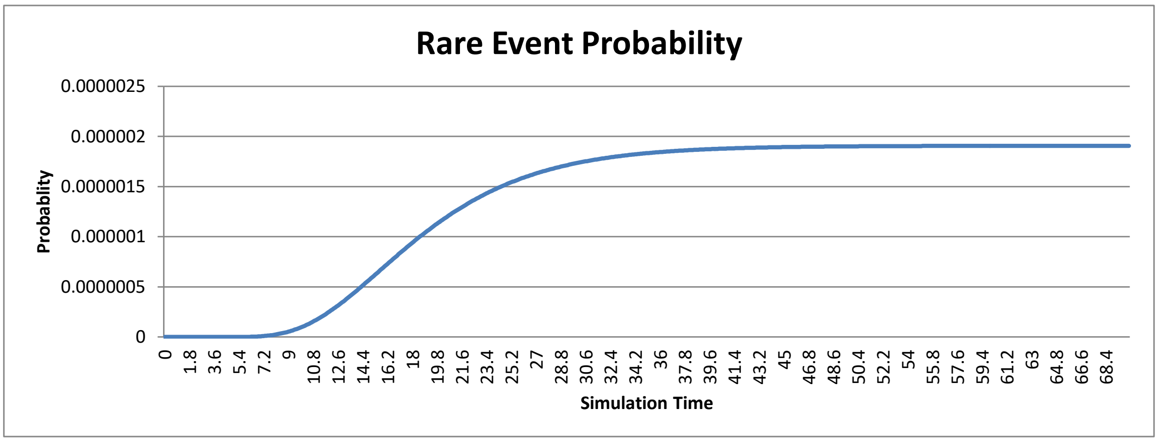

Figure 4.7: Transient probabilities for L=20 in the tandem queue model

3. $\min(q_1, q_2) \geq L$

   The rare event is defined as the minimum number of customers in one of the queuing systems is greater than or equal to a certain threshold.

   Table 4.10 and Table 4.11 shows the rare event probability using both simulation methods. Again, the proxel-based method shows higher accuracy compared to the RESTART method.

L	Analytic result	RESTART result	Time (sec)	% error
20	2.74E-16	2.92E-16	108	6.16 %
30	4.52E-24	4.56E-24	381	0.87 %
40	7.48E-32	7.10E-32	1050	5.35 %

Table 4.10: RESTART results for the tandem queue model for $\min(q_1, q_2) \geq L$ [VAVA99]

L	Analytic result	Proxel result	Time (sec)	% error
20	2.74E-16	2.73E-16	18	0.365 %
30	4.52E-24	4.52E-24	75	0 %
40	7.48E-32	7.48E-32	437	0 %

Table 4.11: Proxel-based results for the tandem queue model for $\min(q_1, q_2) \geq L$

Figure 4.8 shows the transient probability for threshold L= 20 customers until reaching the steady-state. The probability converges after simulation t=78 with probability of 2.74E-16.

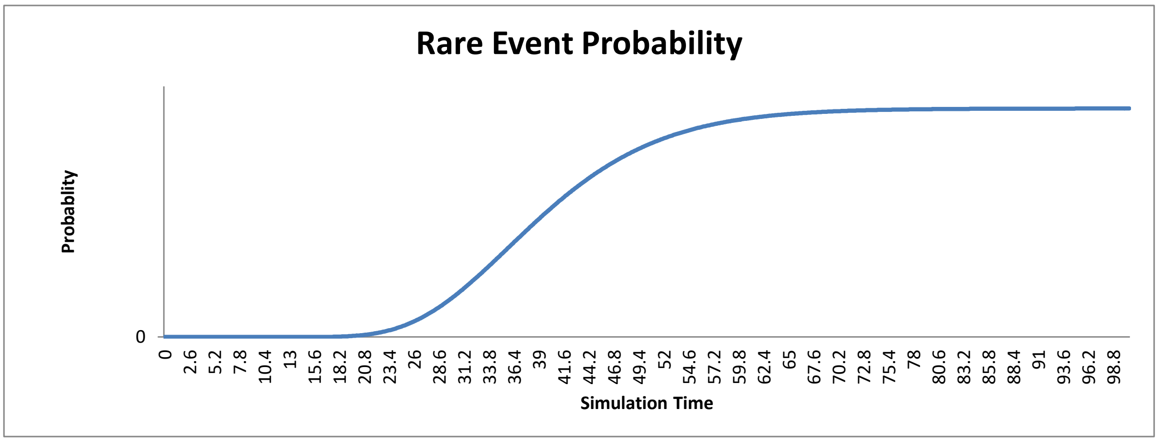

Figure 4.8: Transient probabilities for L=20 in the tandem queue model

As an overall summary of all of the above results, proxel-based simulation has higher accuracy compared to the RESTART method. Several probability cut-off values were used in each of the different thresholds values, that's to achieve short computational runtime and higher accuracy.

4.5 Effect of Time step

Time step size has an important role in the proxel-based method. The smaller the time step size the more accurate the simulation results. However, smaller time step would as well increase the simulation computational runtime. Therefore, time step size Δt should be selected carefully, where it is recommended to select a time step size smaller than the globally acceptable time step of the system as explained in Chapter 2.

Several time step sizes were tested in the evaluation of all the above models to achieve higher accuracy and short simulation runtime.

For explanation purpose, we use the system failure model with failure rate ε= 1.00E-02 in this section to illustrate the effect of the time step size.

Figure 4.9 shows the rare event probability using the time step Δt = 0.4, where the probability converge only after several time steps with 1% error with simulation runtime of 0.01 seconds. While, using a smaller time step such as Δt = 0.1 (see Figure 4.10) has 0 % error and simulation runtime of 0.04 seconds with a longer time to converge simulation time to converge.

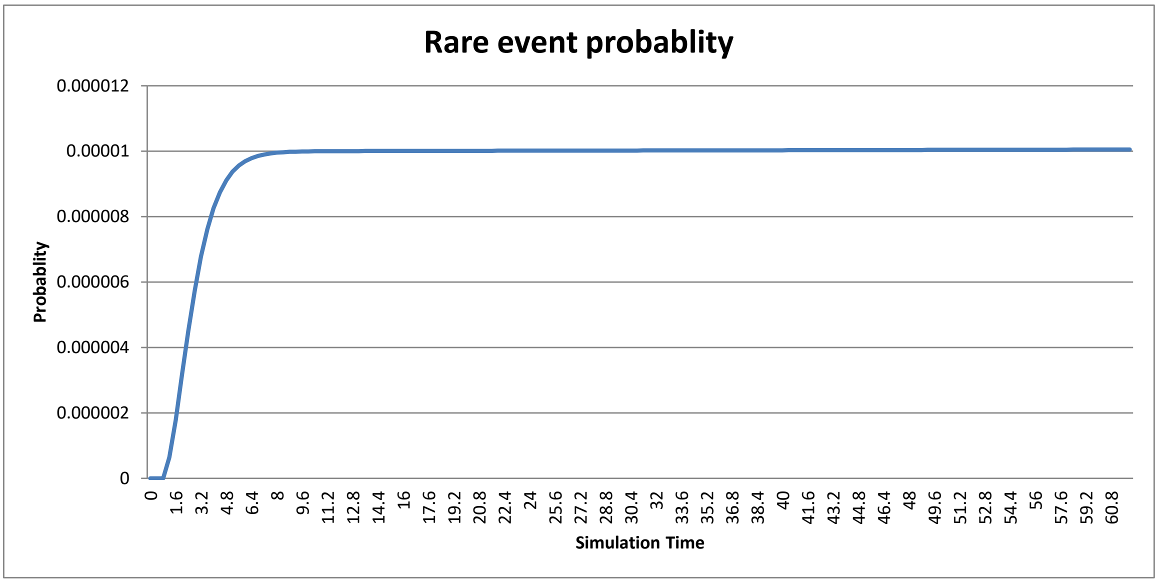

Figure 4.9: Transient probabilities for ε= 1.00E-02 using Δt=0.4 in the system failure model

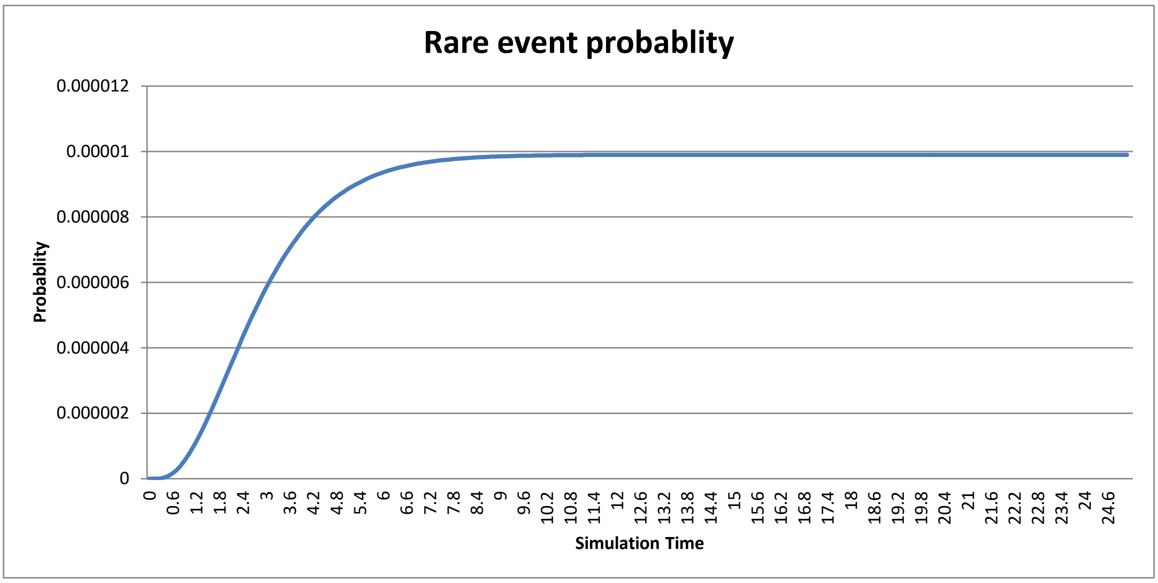

Figure 4.10: Transient probabilities for ε= 1.00E-02 using Δt=0.1 in the system failure model

4.6 Effect of Probability Cut-off

It has been noticed that probability pruning in the rare event is an important factor to avoid long simulation runtime and to achieve better accuracy.

To illustrate the importance of the probability cut-off let's consider the system failure and tandem queue models.

- System failure model:

probability cut-off =1.00E-10				
ε	Analytic result	Proxel result	Time (sec)	% error
1.00E-01	8.9645E-03	8.9644E-3	0.07	0.001116%
2.00E-02	7.8397E-05	7.8393E-5	0.03	0.0051%
1.00E-02	9.9000E-06	9.89702E-6	0.03	0.030%
2.00E-03	7.9840E-08	7.9800E-8	0.03	0.05013 %
1.00E-03	9.9900E-09	9.8116E-09	0.03	1.818 %
probability cut-off =1.00E-15				
ε	Analytic result	Proxel result	Time (sec)	% error
1.00E-01	8.9645E-03	8.9645E-3	0.1	0 %
2.00E-02	7.8397E-05	7.8397E-5	0.04	0 %
1.00E-02	9.9000E-06	9.8999E-6	0.04	0.001 %
2.00E-03	7.9840E-08	7.9839E-8	0.04	0.001 %
1.00E-03	9.9900E-09	9.989E-09	0.04	0.01 %

Table 4.12: Proxel-based results for the system failure model using probability cut-off 1.00E-10 and 1.00E-15 respectively

We perform one more run for the proxel-based code of the system failure model, but this time we used a probability cut-off equal to 1.00E-10. As seen in Table 4.12 using probability cut-off close to the desired probability did not make the accuracy too much worse. Another point to realize, the computational time was slightly decreased. For instance, estimating the rare event probability with failure rate ε=1.00E-03 with a probability cut-off of 1.00E-10 increased the % error to 1.818% and reduced the simulation runtime to 0.03 seconds.

- Tandem queue:

  We decerased the probability cut-off value from 1.0e-15 to 1.0e-35 to estimate the rare event probability of 20 customers at the second queuing system ($q_2 \geq 20$), and as a result, we achieved 0 % error, but a longer computational runtime of 90 seconds instead of 2 seconds.

  Another example is to increase the probability cut-off value. We increased the probability cut-off from 1.00E-27 to 1.00E-23 to estimate the probability for $q_2 \geq 40$, and as a result, we achieved 3.28% error instead of 0 %, with a shorter computational runtime of 18 seconds instead of 35 seconds. Figure 4.11 shows the transient probabilities for $q_2 \geq 40$ using different probability cut-off values, whereas seen in the figure that using a probability cut-off from 1.00E-27 has a higher accuracy.

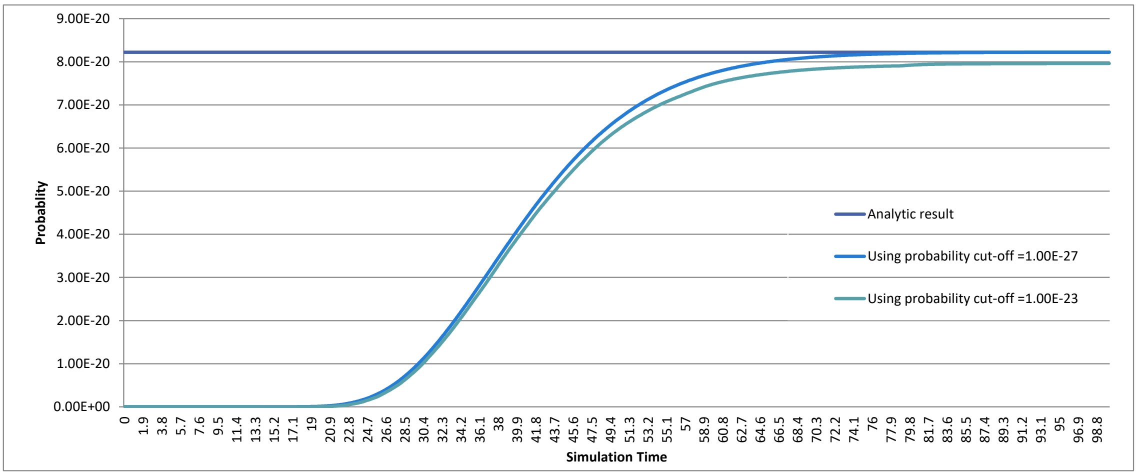

Figure 4.11: Transient probabilities for $q_2 \geq 40$ in the tandem queue model using different probability cut-off values

In this section, we show the effect of the probability cut-off value in both of the system failure and tandem queue model. The more accurate the selection of the probability cut-off value the higher the accuracy and faster simulation results.

4.7 Summary and Discussion

In this chapter, we compared the proxel-based method results with other results obtained using the RESTART method in reliability and queuing systems models. In addition to the comparison, we validate our proxel-based code with M/M/1 model analytic results. Then, we showed that proxel-based method is efficient to analyze the rare event in non-Markovian single server queuing system models. Furthermore, we showed the effect of the time step size and the probability cut-off in the estimating a rare event probability.

It turned out that some models such as non-Markovian models or models with larger state space require more careful decision in the selection of the probability cut-off value. For instance, in the system failure model, the probability cut-off value was the same for all the different ε values while in the queuing systems different probability cut-off values were selected for each of the threshold cases. The smaller the probability cut-off values, the more memory required while, the higher probability cut-off values, the less accurate results.

5. Conclusion and Future Work

Analyzing rare event models is still considered as a simulation limitation for discrete event simulation. The common goal of many of the rare event simulation approaches is to increase the rare event occurrence during the simulation. In this chapter, we provide a brief summary of the thesis, followed by the conclusion, and the future work.

5.1 Summary

In the first place, we reviewed several simulation methods used to analyze models containing rare events. Among the reviewed methods: Importance Sampling and Importance Splitting/RESTART. We discussed both of their advantages and disadvantages. Then, we used the proxel-based method as a rare event simulation method and suggested that the proxel-based approach is competitive approach to the existing rare events methods due to that proxel-based method gives all states in the model the same importance.

Also, we discussed the existing rare event methods mentioned in the thesis and showed that RESTART method has more advantages over the other existing rare event simulation methods. Then, we presented several commonly studied rare event models. The studied models are categorized into two groups: Reliability model and queuing systems models. During the implementation, we faced some challenges to achieve higher accuracy and to keep a short computational runtime.

Lastly, we compared all the results obtained with both methods. Besides that, we showed the effects of both of time step size and probability cut-off on the computational runtime and the accuracy of the simulation.

5.2 Conclusion

The primary goal of the thesis is to show whether the proxel-based method is a competitive approach to RESTART method. RESTART method has often been used for rare event simulation, where it has many successful publications.

In this work, we use the proxel-based method as a rare event simulation approach. Several rare event models were studied, and the results obtained using the proxel-based method were compared to similar results obtained using RESTART method.

As a summary of this work, the proxel-based method showed to be a competitive approach to the RESTART method. The results of this work can be listed as follows:

- Proxel-based method achieved higher accuracy and shorter computational runtime compared to the RESTART method in the system failure model. Moreover, the proxel-based method achieved higher accuracy as well in the in the tandem queue model compared to the RESTART.

- Proxel-based method overcomes non-Markovian rare event models and can estimate rare event probability in such models in a short computational runtime, which depends on the conditional probability distribution in a model.

- Probability cut-off and time step size play an essential factor in the simulation of the rare event. In that matter, a more precise selection of both probability cut-off and time step size are required to achieve accurate estimation in a short simulation runtime.

The reason behind the efficiency of the proxel-based method is that proxel-based method gives all events in the systems the same importance. However, the proxel-based method has one known disadvantage due to its deterministic nature is the state space explosions and the memory complexity.

In this thesis, we discussed the effect of time step size and the probability cut-off value in the proxel-based method. Both time step size and the probability cut-off value have an impact on the accuracy and the simulation runtime especially when a probability of interest is rare event probability. This means the performance of this method depends on the value selection of those parameters.

5.3 Future Work

This thesis can be extended by developing a software modeling tool with a graphical interface. This software modeling tool can make the proxel-based easier to be applied to various models, where no coding would be required.

Moreover, the proxel-based method is still needed to be tested with a real-world problems. In this thesis, the proxel-based method was tested with often used research academic models. However, we hope that the proxel-based approach is efficient to estimate rare event probability in various industry fields.

Bibliography

[AHO95] Sigrún Andradóttir, Daniel P. Heyman, and Teunis J. Ott. On the choice of alternative measures in importance sampling with markov chains. *Operations Research*, 43(3):509–519, 1995. (cited on Page 8)

[AMD+] Soren Asmussen, Ny Munkegade, Paul Dupuis, Reuven Rubinstein, and Hui Wang. Importance sampling for rare events. (cited on Page 8)

[Ban05] J. Banks. *Discrete-event System Simulation*. Prentice-Hall international series in industrial and systems engineering. Pearson Prentice Hall, 2005. (cited on Page 1, 2, and 7)

[Bay70] A.J Bayes. Statistical techniques for simulation models. *Australian Computer Journal*, pages 180 – 184, 1970. (cited on Page 9)

[BBC+09] Souheib Baarir, Marco Beccuti, davide Cerotti, Marco De Pierro, Susanna Donatelli, and Giuliana Franceschinis. The GreatSPN Tool: Recent Enhancements. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):4–9, March 2009. (cited on Page 18)

[BGdMT06] Gunter Bolch, Stefan Greiner, Hermann de Meer, and Kishor S. Trivedi. *Single Station Queueing Systems*, pages 241–319. John Wiley Sons, Inc., 2006. (cited on Page 27)

[CFJ+91] J Couvillion, Roberto Freire, Ron Johnson, W Douglas Obal, Muhammad A Qureshi, Manish Rai, William H Sanders, and Janet E Tvedt. Performability modeling with ultrasan. In *Petri Nets and Performance Models, 1991W, Proceedings of the Fourth International Workshop on*, pages 290–299. IEEE, 1991. (cited on Page 18)

[CGK+09] T. Courtney, S. Gaonkar, K. Keefe, E. W. D. Rozier, and W. H. Sanders. M x00f6;bius 2.3: An extensible tool for dependability, security, and performance evaluation of large and complex system models. In *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, pages 353–358, June 2009. (cited on Page 18)

[Gar00] Marnix Joseph Johann Garvels. The splitting method in rare event simulation. 2000. (cited on Page 9 and 12)

[GG02] Carmelita Görg and Stefano Giordano. Rare event simulation. *European Transactions on Telecommunications*, 13:299–301, 2002. (cited on Page 9)

[GHSZ99] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. Multilevel splitting for estimating rare event probabilities. *Operations Research*, 47(4):585–600, 1999. (cited on Page 19)

[GI89] Peter Glynn and Don Iglehart. Importance sampling for stochastic simulations. 35:1367–1392, 11 1989. (cited on Page 8)

[GK98] Marnix J. J. Garvels and Dirk P. Kroese. A comparison of restart implementations. In *Proceedings of the 30th Conference on Winter Simulation*, WSC '98, pages 601–608, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press. (cited on Page 19, 20, and 30)

[GKZH95] Reinhard German, Christian Kelling, Armin Zimmermann, and Günter Hommel. Timenet: a toolkit for evaluating non-markovian stochastic petri nets. *Performance Evaluation*, 24(1):69 – 87, 1995. Performance Modeling Tools. (cited on Page 18 and 19)

[Gro08] Donald Gross. *Fundamentals of queueing theory*. John Wiley & Sons, 2008. (cited on Page 26)

[GUD15] THORBJÖRN GUDMUNDSSON. *Rare-event simulation with Markov chain Monte Carlo*. PhD thesis, Royal Institute of Technology Stockholm, 2015. (cited on Page 3)

[GW97] Paul Glasserman and Yashan Wang. Counter examples in importance sampling for large deviations probabilities. *The Annals of Applied Probability*, 7(3):731–746, 1997. (cited on Page 8)

[Hei95] P. Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, 1(5):43–85, 1995. (cited on Page 21 and 22)

[Hor02] Graham Horton. A new paradigm for the numerical simulation of stochastic petri nets with general firing times. *Proceedings of the European Simulation Symposium*, pages 129–136, 10 2002. (cited on Page 3, 12, and 17)

[HTT00] Christophe Hirel, Bruno Tuffin, and Kishor S. Trivedi. Spnp: Stochastic petri nets. version 6.0. In Boudewijn R. Haverkort, Henrik C. Bohnenkamp, and Connie U. Smith, editors, *Computer Performance Evaluation.Modelling Techniques and Tools*, pages 354–357, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. (cited on Page 18)

[JKW07] Kurt Jensen, Lars Michael Kristensen, and Lisa Wells. Coloured petri nets and cpn tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3):213–254, Jun 2007. (cited on Page 18)

[KAAJB13] Ali Khalili, Mohammad Abdollahi Azgomi, and Amir Jalaly Bidgoly. Simgine: A simulation engine for stochastic discrete-event systems based on sdes description. 89:539–555, 04 2013. (cited on Page 18)

[KBH09] Claudia Krull, Robert Buchholz, and Graham Horton. Improving the efficiency of the proxel method by using individual time steps. In *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, pages 116–130. Springer, 2009. (cited on Page 16 and 17)

[Kel96] C. Kelling. A framework for rare event simulation of stochastic petri nets using ”restart”. In *Proceedings Winter Simulation Conference*, pages 317–324, 1996. (cited on Page 18)

[KH51] Herman Kahn and T. E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Applied Mathematics Series*, 12:27 – 30, 1951. (cited on Page 9)

[KH07] Claudia Krull and Graham Horton. Application of proxels to queuing simulation. In *SimVis*, pages 299–310, 2007. (cited on Page 26)

[Lam08] E. Lamers. *Contributions to Simulation Speed-Up: Rare Event Simulation and Short-Term Dynamic Simulation for Mobile Net-*

work Planning. Advanced Studies Mobile Research Center Bremen. Vieweg+Teubner Verlag, 2008. (cited on Page 1)

[L'E90] Pierre L'Ecuyer. Random numbers for simulation. *Commun. ACM*, 33(10):85–97, October 1990. (cited on Page 8)

[LM05] Sanja Lazarova-Molnar. The proxel-based method: formalisation, analysis and applications. 11 2005. (cited on Page 3, 5, 6, 7, 8, 12, 13, 15, 16, 17, 18, 19, and 20)

[LMH03a] Sanja Lazarova-Molnar and Graham Horton. An experimental study of the behaviour of the proxel-based simulation algorithm. *Conference: Simulation und Visualisierung*, pages 419–430, 01 2003. (cited on Page 12 and 20)

[LMH03b] Sanja Lazarova-Molnar and Graham Horton. Proxel-based simulation of stochastic petri nets containing immediate transitions. 01 2003. (cited on Page 3 and 12)

[LMT09] Pierre L'Ecuyer, Michel Mandjes, and Bruno Tuffin. *Rare Event Simulation using Monte Carlo Methods.* 03 2009. (cited on Page 8 and 9)

[LSP16] Francesco Longo, Marco Scarpa, and Antonio Puliafito. *WebSPN: A Flexible Tool for the Analysis of Non-Markovian Stochastic Petri Nets*, pages 255–285. Springer International Publishing, Cham, 2016. (cited on Page 18)

[McH09] R. McHaney. *Understanding Computer Simulation.* Ventus Publishing, 2009. (cited on Page 2 and 5)

[MSM09] Denis Miretskiy, Werner Scheinhardt, and Michel Mandjes. Rare-event simulation for tandem queues: A simple and efficient importance sampling scheme. In Rudesindo Núñez-Queija and Jacques Resing, editors, *Network Control and Optimization.* Springer Berlin Heidelberg, 2009. (cited on Page 9)

[OS94] W Douglas Obal and William H Sanders. Importance sampling simulation in ultrasan. *Simulation*, 62(2):98–111, 1994. (cited on Page 18)

[Pet62] Carl Adam Petri. *Kommunikation mit Automaten.* PhD thesis, Universität Hamburg, 1962. (cited on Page 6)

[PSW05] Pagano, Sandmann, and Werner. Efficient rare event simulation: A tutorial on importance sampling. In *Proceedings of the 3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks*, 2005. (cited on Page 2 and 3)

[RN16] M.S. Rao and V Naikan. Review of simulation approaches in reliability and availability modeling. 12:369–388, 07 2016. (cited on Page 1 and 21)

[RR08] Elena Rashkova and Dimitar Radev. Simulation of rare event in queueing systems. *International Scientific Conference Computer Science*, 2008. (cited on Page 19)

[RT09] G. Rubino and B. Tuffin. An introduction to monte carlo methods and rare event simulation. In *2009 Sixth International Conference on the Quantitative Evaluation of Systems*, pages 6–6, Sept 2009. (cited on Page 2 and 3)

[SSG97] Peter J. Smith, Mansoor Shafi, and Hongsheng Gao. Quick simulation: A review of importance sampling techniques in communications systems. *IEEE Journal on Selected Areas in Communications*, 15(4):597–613, 1997. (cited on Page 8 and 9)

[Tim] Timenet this is the official page of the timenet tool project. Available at `https://timenet.tu-ilmenau.de` (visited on 05/5/2018). (cited on Page 18 and 19)

[TT00] Bruno Tuffin and Kishor S Trivedi. Implementation of importance splitting techniques in stochastic petri net package. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 216–229. Springer, 2000. (cited on Page 18)

[VA07] Jose Villen-Altamirano. Rare event restart simulation of two-stage networks. *European Journal of Operational Research*, 179(1):148–159, 2007. (cited on Page 10 and 11)

[VA09] J. Villén-Altamirano. Restart simulation of networks of queues with erlang service times. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1146–1154, Dec 2009. (cited on Page 10 and 11)

[VARLE16] Manuel Villén-Altamirano, Arcadio Reyes-Lecuona, and Casilari Eduardo. Optimal importance function for restart simulation of a two-queue tandem jackson network. 03 2016. (cited on Page 38)

[VAVA91] Manuel Villén-Altamirano and José Villén-Altamirano. Restart: A method for accelerating rare event simulations. 3, 01 1991. (cited on Page 9, 10, and 11)

[VAVA99] Manuel Villén-Altamirano and José Villén-Altamirano. On the efficiency of restart. 10 1999. (cited on Page ix, xi, 9, 10, 11, 30, 31, 41, 42, 43, and 44)

[VAVA06] Manuel Villén-Altamirano and José Villén-Altamirano. On the efficiency of restart for multidimensional state systems. *ACM Trans. Model. Comput. Simul.*, 16(3):251–279, July 2006. (cited on Page 10 and 11)

[VAVA11] Manuel Villén-Altamirano and José Villén-Altamirano. The rare event simulation method restart: Efficiency analysis and guidelines for its application. pages 509–547, 2011. (cited on Page 10, 11, and 19)

[YV04] Ting Yan and Malathi Veeraraghavan. Networks of queues, April 2004. (cited on Page ix and 27)

[Zim06] Armin Zimmermann. Applied restart estimation of general reward measures. 01 2006. (cited on Page 19)

[Zim07] Armin Zimmermann. *Stochastic Discrete Event Systems: Modeling, Evaluation, Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. (cited on Page 6 and 8)

[Zim10] Armin Zimmermann. Dependability evaluation of complex systems with timenet. pages 33–34, 01 2010. (cited on Page 19)

[Zim18] Armin Zimmermann. Restart simulation of colored stochastic petri nets. 04 2018. (cited on Page 19)

[ZK] Armin Zimmermann and Michael Knoke. A software tool the performability evaluation with stochastic and colored petri nets. (cited on Page 23)

[ZRWL16] Armin Zimmermann, Daniel Reijsbergen, Alexander Wichmann, and Andres Canabal Lavista. Numerical results for the automated rare event simulation of stochastic petri nets. 2016. (cited on Page 19, 22, and 36)

Declaration

I hereby declare that I have written the present work myself and did not use any sources or tools other than the ones indicated.

Magdeburg, 18. 06 2018

Amine Mouzannar