

Otto-von-Guericke-Universität Magdeburg



Fakultät für Informatik  
Lehrstuhl für Simulation

## Diplomarbeit

**Proxel-basierte Simulation von stochastischen Reward-Netzen durch  
Betrachtung Reward-relevanter Zustände**

Autor:

Rebekka Brandt

16. Januar 2006

Betreuer:

Dipl. Inf. Fabian Wickborn

**Rebekka Brandt**

*Proxel-basierte Simulation von stochastischen Reward-Netzen  
durch Betrachtung Reward-relevanter Zustände*

Diplomarbeit, Otto-von-Guericke-Universität  
Magdeburg, 2006.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Erwarteter Nutzen . . . . .	3
1.3	Ziele dieser Arbeit . . . . .	3
1.4	Aufgabenstellung . . . . .	4
1.5	Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Grundlagen der Arbeit</b>	<b>6</b>
2.1	Stochastische Petrinetze . . . . .	6
2.2	Stochastische Reward-Netze . . . . .	8
2.3	Das Proxel-basierte Verfahren . . . . .	10
2.4	Zusammenfassung . . . . .	14
<b>3</b>	<b>Ein adaptiertes Proxelverfahren</b>	<b>15</b>
3.1	Anforderungen an ein angepasstes Proxelverfahren . . . . .	15
3.2	Relevanz von Systemzuständen . . . . .	16
3.3	Die zugrundeliegende Idee . . . . .	17
3.4	Anpassungen am Proxelalgorithmus . . . . .	17
3.5	Ein Beispiel . . . . .	24
3.6	Zusammenfassung . . . . .	27
<b>4</b>	<b>Experimente</b>	<b>29</b>
4.1	Annahmen und Erwartungen . . . . .	29
4.2	Experiment 1: Maschinenmodell 1 . . . . .	30
4.3	Experiment 2: Maschinenmodell 2 . . . . .	34
4.4	Experiment 3: Modell einer Warteschlange . . . . .	36
4.5	Experiment 4: Komplett verbundenes Modell . . . . .	40
4.6	Schlussfolgerungen aus den Experimenten . . . . .	41

<b>5 Zusammenfassung und Bewertung</b>	<b>43</b>
5.1 Zusammenfassung . . . . .	43
5.2 Ausblick . . . . .	44
5.3 Bewertung . . . . .	45

# Abbildungsverzeichnis

2.1	Ein stochastisches Petrinetz: Maschine arbeitet (OK), Maschine wird gewartet (M), Maschine ist ausgefallen(F)	7
2.2	Erreichbarkeitsgraph des SPNs	8
2.3	Struktur des Proxelbaums für das SPN	11
2.4	ein einfaches Petrinetz (a) und dessen Erreichbarkeitsgraph (b)	11
2.5	initialer Proxel	12
2.6	erste Schritte der Proxel-basierten Simulation	13
3.1	zugrundeliegende Proxelbaumstruktur	18
3.2	Gleiche R-Proxel	20
3.3	Datenstruktur einer Maske	20
3.4	Beispiel: aus einem I-Proxel entstehender R-Proxel im vierten Zeitschritt	23
3.5	Datenstruktur zur temporären Speicherung von R-Proxel	23
3.6	Datenstruktur eines R-Proxel	24
3.7	aSRN eines Wartungsmodells	25
3.8	erste Schritte beim adaptierten Verfahren	25
3.9	Durch die Suche nach R-Proxel erzeugte Maske für (Maint,0)	26
3.10	Ergebnis der Erzeugung relevanter R-Proxel aus dem I-Proxel (Maint,0)	26
3.11	Ergebnis bei erneutem Auftreten des I-Proxel (Maint,0)	27
4.1	aSRN eines Maschinenmodells	30
4.2	Transiente Wahrscheinlichkeiten a) Originalverfahren - alle Zustände relevant. b) Neues Verfahren - nur Zustand A relevant	31
4.3	Analysezeit pro Zeitschritt für das Maschinenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren	32
4.4	Analysezeit für das Maschinenmodell - Originalverfahren im Vergleich zum adaptieren Verfahren	33
4.5	Anzahl verarbeiteter Proxel a) Originalverfahren b) neues Verfahren (nur A relevant, A und C relevant, nur C relevant)	34

4.6	Analysezeit für das Maschinenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren (bei unterschiedlichen relevanten Zuständen)	35
4.7	a) aSRN einer Warteschlange b) Erreichbarkeitsgraph des aSRN . . . . .	36
4.8	Analysezeit für das Warteschlangenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren . . . . .	37
4.9	Analysezeit pro Zeitschritt für das Warteschlangenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren (bei unterschiedlicher Anzahl relevanter Zustände) . . . . .	38
4.10	Analysezeit für das Warteschlangenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren (bei unterschiedlicher Anzahl relevanter Zustände) . . . . .	39
4.11	Zustandsgraph eines komplett verbundenen Modells . . . . .	40
4.12	Analysezeit für das komplett verbundene Modell - Originalverfahren im Vergleich zum adaptierten Verfahren (bei relevantem Zustand C und vorheriger Erstellung der Maske für C) . . . . .	41

# Tabellenverzeichnis

4.1	Experiment 1 - Parameter des Maschinenmodells . . . . .	30
-----	---	----

# Verzeichnis der Abkürzungen

SPN	stochastisches Petrinetz
SRN	stochastisches Reward-Netz
aSRN	adaptiertes stochastisches Reward-Netz
DES	diskrete ereignisorientierte Simulation
IRF	Instantaneous Rate Function
SV	supplementary variables



# Zusammenfassung

Am Lehrstuhl für Simulation der Otto-von-Guericke-Universität Magdeburg interessieren sich die Forscher für das *Proxel-basierte Verfahren*, das zur Simulation von stochastischen Prozessen eingesetzt wird. Da das Verfahren den gesamten erreichbaren Zustandsraum eines Modells betrachtet, ist es bisher nur für Modelle mit einem kleinen Zustandsraum effizient einsetzbar.

Ziel dieser Arbeit war es zu untersuchen, ob sich bei der Proxel-basierten Simulation Effizienzvorteile erreichen lassen, wenn die Analyse auf Systemzustände beschränkt wird, die für eine bestimmte Fragestellung relevant sind.

Dazu wurde zunächst eine veränderte Modellspezifikation aus *stochastischen Petri-Netzen* und *stochastischen Reward-Netzen* entwickelt und definiert, wann ein diskreter Modellzustand relevant ist und damit durch die Proxel-basierte Analyse betrachtet werden soll. Anschließend wurde ein angepasstes Proxelverfahren entwickelt, mit dem es möglich ist während der Analyse weniger Proxel zu verarbeiten. In den durchgeführten Experimenten zeigte sich, dass durch diese Anpassungen für einige zu betrachtende Modelle ein Geschwindigkeitsvorteil erreicht werden kann.

Diese Arbeit kann dem Themengebiet der diskreten, zustandsraum-basierten Simulation zugeordnet werden.

# 1 Einleitung

Durch wissenschaftlichen und technischen Fortschritt werden immer komplexere Verfahren, Techniken und Maschinen entwickelt und eingesetzt. Eine Bewertung dieser zum Beispiel hinsichtlich der Effizienz, der Auslastung oder des Durchsatzes ist nicht oder nur sehr schwierig vorzunehmen. Dies trifft vor allem zu, wenn es sich um ein System handelt, das sich noch in der Planungsphase befindet oder für das aus Kostengründen kein realer Prototyp geschaffen werden kann.

Der Einsatz der Computersimulation hat sich deshalb in den letzten Jahrzehnten zunehmend verbreitet. Hierbei werden komplexe Sachverhalte abstrakt in einem Modell dargestellt und anschließend durch Verfahren am Computer analysiert. Mit Hilfe der entstehenden Ergebnisse sind dann Aussagen über das Verhalten des zu untersuchenden Systems möglich. Die Techniken der Simulation erlauben es, das Verhalten eines Systems zu verstehen, vorauszusagen und zu bewerten ohne in das System selbst einzugreifen.

Im Bereich der Simulation kann man im Allgemeinen zwischen der (zeit-) kontinuierlichen und der (zeit-) diskreten Simulation unterscheiden. Diese Arbeit lässt sich dem letztgenannten Gebiet, genauer gesagt der Analyse von diskreten stochastischen Modellen zuordnen.

Damit sind solche Systeme gemeint, die sich durch ein stochastisches Verhalten auszeichnen. Zum Beispiel unterliegt der der Ausfall eines Computers, der Zugriff auf das Internet oder die Ankunft von Kunden im Supermarkt keiner festen Zeit, sondern einer gewissen Wahrscheinlichkeitsverteilung.

Um zum Beispiel Aussagen über die Leistungsfähigkeit oder Zuverlässigkeit solcher Systeme treffen zu können, ist es wichtig, effiziente Methoden zur Analyse zu besitzen, mit denen mit möglichst geringen zeitlichen Aufwand, möglichst genaue Ergebnisse erzielt werden können.

Eine relativ neue Methode ist die Proxel-basierte Simulation [4]. Dieses Verfahren hat sich bei der Betrachtung einiger Modelle als ein viel versprechender Ansatz herausgestellt. Das Verfahren selbst ist ein aktuelles Forschungsthema am Lehrstuhl für Simula-

tion der Otto-von-Guericke-Universität Magdeburg. Es wird damit gerechnet, dass noch mehr Potential in dem Verfahren liegt als bisher erforscht wurde. Auch diese Arbeit soll einen Beitrag zur Erforschung des Proxel-basierten Verfahrens liefern.

### 1.1 Motivation

Die Analyse von stochastischen Modellen spielt insbesondere bei Leistungsbewertungen und Zuverlässigkeitsbetrachtungen eine große Rolle. Das gebräuchlichste Verfahren hierfür ist die diskrete ereignisorientierte Simulation [5]. Sind allerdings seltene Ereignisse involviert, oder bestehen hohe Anforderungen an die Genauigkeit der Ergebnisse, kann dieses Verfahren sehr ineffizient sein. Es ist eine hohe Anzahl von Simulationsläufen mit unterschiedlichen Zufallszahlenreihen notwendig, um statistisch aussagekräftige Ergebnisse zu erhalten.

Eine andere Möglichkeit der Analyse bieten zustandsraum-basierte Verfahren. Diese sind deterministisch und benötigen daher keine wiederholten Simulationsläufe. Problematisch ist allerdings, dass der Zustandsraum eines Modells exponentiell wachsen kann, was auch als Zustandsraumexplosion bezeichnet wird.

Ein zustandsraum-basiertes Verfahren ist die Proxel-basierte Simulation [4]. Das Verfahren verarbeitet während der Analyse sogenannte Proxel. Ein Proxel repräsentiert einen bestimmten Zustand des Modells zu einem Simulationszeitpunkt. Beginnend mit einem initialen Proxel werden iterativ die in einem diskreten Zeitschritt erreichbaren Folgezustände bestimmt, deren Wahrscheinlichkeiten ermittelt und dadurch der gesamte erreichbare Zustandsraum aufgebaut.

Im Vergleich zur diskreten ereignisorientierten Simulation hat sich das Verfahren bei einigen Modellen, insbesondere bei seltenen Ereignissen als viel effizienter herausgestellt [6]. Den Vorteilen steht aber das Problem der Zustandsraumexplosion gegenüber, da das Verfahren den gesamten erreichbaren Zustandsraum betrachtet. Der Aufwand des Proxelverfahrens ist exponentiell und bisherige Untersuchungen zeigten, dass es sich nur für Modelle eignet, die einen kleinen Zustandsraum besitzen. Von größtem Interesse ist daher die Komplexität des Verfahrens zu verringern, um es auch zur Analyse größerer Modelle einsetzen zu können.

Betrachtet man die Fragestellungen, die durch eine Simulation beantwortet werden sollen, so fällt auf, dass häufig nicht alle Zustände zur Beantwortung relevant sind. Eine solche Frage könnte zum Beispiel sein „Wie hoch ist die Wahrscheinlichkeit, dass die Warteschlange einer Bank leer ist?“.

Eine interessante Idee ist nun, die Anzahl der zu analysierenden Proxels zu reduzieren, indem nur Proxel betrachtet werden, die für eine Fragestellung relevant sind. Dadurch ließen sich möglicherweise der Speicheraufwand und die Rechenzeit beschränken und das Analyseverfahren insgesamt beschleunigen. Völlig unklar ist allerdings noch, ob dies funktioniert und wirklich zu Geschwindigkeitsvorteilen führt.

In dieser Arbeit wird untersucht, ob sich bei der Proxel-basierten Simulation Effizienzvorteile erreichen lassen, wenn die Analyse auf Proxel beschränkt wird, die für eine bestimmte Fragestellung relevant sind.

### 1.2 Erwarteter Nutzen

Welchen Nutzen könnte die neue Idee bieten?

Werden nur die für eine Fragestellung relevanten Zustände durch das Proxelverfahren verarbeitet und lassen sich dadurch Geschwindigkeitsvorteile erzielen, könnten einige der betrachteten Modelle viel effizienter analysiert werden. Damit könnten Aussagen und Entscheidungen schneller als bisher getroffen werden.

Weiterhin wäre es möglich auch größere Modelle, die bisher nur durch ereignisbasierte Algorithmen effizient analysierbar sind, mit dem Proxelverfahren zu verarbeiten.

Außerdem wird erwartet, dass sich anhand experimenteller Ergebnisse Aussagen über die Effizienz eines solchen Verfahrens treffen lassen. Vermutlich ergeben sich daraus weitere Themen für Forschungsarbeiten, mit denen das Potential des Proxelverfahrens weiter ausgeschöpft werden kann.

### 1.3 Ziele dieser Arbeit

Entsprechend der Motivation ergeben sich die folgenden Ziele für diese Diplomarbeit:

#### **Ziel 1** *Definition der Relevanz von Systemzuständen*

Die gegenwärtige Implementierung des Proxel-basierten Verfahrens betrachtet für eine Simulationsaufgabe den gesamten erreichbaren Zustandsraum. Oft sind aber für eine konkrete Fragestellung nicht alle Zustände relevant. Das Verfahren kann also aufgrund der bisherigen Modellbeschreibung nicht zwischen relevanten und nicht-relevanten Zuständen unterscheiden. Daraus ergibt sich das Ziel, ein Maß für die Relevanz von Zuständen zu definieren.

### **Ziel 2** *Prototypische Implementierung eines angepassten Proxelalgorithmus*

Hierfür wird ein angepasster Proxelalgorithmus entwickelt, der in der Lage ist, zwischen den als relevant und nicht-relevant definierten Zuständen zu unterscheiden und diese entsprechend unterschiedlich zu behandeln.

### **Ziel 3** *Aussage über die Verwendbarkeit*

Mit der Definition der Relevanz von Systemzuständen und der Klassifikation durch ein angepasstes Proxelverfahren, wird die Analyse auf Proxel beschränkt, die für die Beantwortung einer konkreten Fragestellung relevant sind. Deshalb wird untersucht, welche Bedingungen Modelle erfüllen müssen, damit diese effizient verarbeitet werden können. Dazu wird zum Vergleich der Originalalgorithmus implementiert.

Zusammenfassend lässt sich für dieses Ziel festhalten, dass experimentelle Untersuchungen und Schlussfolgerungen angestellt und Aussagen über die Effizienz eines angepassten Verfahrens getroffen werden.

Eine Erwartung ist, dass sich das Proxelverfahren insoweit anpassen lässt, dass es automatisch irrelevante Zustände „überspringen“ kann. Eine Vermutung ist jedoch, dass weitere Datenstrukturen benötigt werden, die die Wahrscheinlichkeiten von nicht-relevanten Zuständen berücksichtigen, weil diese wieder zu relevanten Zuständen führen könnten.

Vermutlich ist es auch von der Anzahl der relevanten Zustände abhängig, wie sich die Berechnungszeit und damit auch der Analyseaufwand ändert. Ist zum Beispiel die Betrachtung aller Zustände nötig, wird sich an der Komplexität des Verfahrens nichts ändern.

## **1.4 Aufgabenstellung**

Die Aufgabenstellung ergibt sich aus der Motivation und besteht unter anderem aus einer Einarbeitung in die Analyseverfahren für diskrete stochastische Modelle. Damit sind insbesondere die zustandsraum-basierten Verfahren und im speziellen das Proxel-basierte Verfahren gemeint.

Weiterhin wird ein Maß für die Beurteilung der Größe „Relevanz eines Zustandes“ benötigt, auf dessen Basis dann zwischen den für eine Fragestellung interessanten und nicht-interessanten Zuständen unterschieden werden kann. Anschließend kann das Proxelverfahren so angepasst werden, dass nur relevante Proxel verarbeitet und irrelevante „übersprungen“ werden.

Anhand einer prototypischen Implementierung eines angepassten Proxelverfahrens soll dann das Potential experimentell untersucht werden. Hier gilt es insbesondere Vor- und Nachteile der neuen Analysemethode zu betrachten und mögliche Spezialfälle herauszufinden. Weiterhin sollen Experimente aufzeigen, inwieweit sich Geschwindigkeitsvorteile erreichen lassen. Mit Schlussfolgerungen aus den Experimenten sollen dann Aussagen über die Verwendbarkeit des neuen Verfahrens getroffen werden.

### 1.5 Aufbau der Arbeit

In diesem Einführungskapitel wurden die Beweggründe für die Aufgabenstellung dargelegt. Weiterhin wurde auf den erwarteten Nutzen der weiteren Untersuchung des Proxelverfahrens eingegangen.

In Kapitel zwei werden die Grundlagen, die für das Verständnis der darauf folgenden Kapitel, notwendig sind gelegt. Dazu werden zunächst stochastische Petrinetze und stochastische Reward-Netze beschrieben und anschließend eine Einführung in die Proxelbasierte Simulation gegeben.

Kapitel drei beschäftigt sich mit der Implementierung eines angepassten Proxelverfahrens, das während der Analyse nur relevante Proxel verarbeitet und irrelevante Proxel „überspringt“. Dazu wird zunächst auf die Anforderungen, die an das Verfahren gestellt wurden, eingegangen. Weiterhin werden die einzelnen Schritte und Datenstrukturen des adaptierten Verfahrens beschrieben. Ein Beispiel verdeutlicht anschließend die Funktionsweise des entwickelten Verfahrens.

Anhand von Experimenten wird in Kapitel vier das Laufzeitverhalten des neuen Verfahrens im Vergleich zum Originalverfahren evaluiert.

Das letzte Kapitel fasst die Ergebnisse dieser Arbeit zusammen und weist auf offene Probleme hin. Weiterhin erfolgt eine persönliche Beurteilung der Arbeit.

## 2 Grundlagen der Arbeit

Dieses Kapitel soll die Grundlagen, die für das weitere Verständnis der folgenden Kapitel notwendig sind, bereitstellen. Hierzu wird zunächst auf stochastische Petrinetze (SPNs) und stochastische Reward-Netze (SRN) eingegangen. Aus diesen beiden Formalismen wird dann eine veränderte Modellspezifikation entwickelt, die die Grundlage für die Anpassung am Proxelverfahren bildet. Anschließend wird das Proxel-basierte Verfahren eingeführt und anhand eines Beispiels erläutert.

### 2.1 Stochastische Petrinetze

Der Ursprung der Petrinetz-Theorie geht zurück auf Carl Adam Petri. In seiner Dissertation wurden sie zur Modellierung nebenläufiger, diskreter Systeme vorgeschlagen. Eine Einführung in die Theorie der Petrinetze findet man in vielen Werken der Standardliteratur, siehe zum Beispiel [11] oder [1].

Stochastische Petrinetze [3] (SPNs) sind eine Methode zur Beschreibung und Analyse von stochastischen Modellen. Sie werden beispielsweise genutzt, um Verkehrssysteme, Computernetzwerke oder Fertigungssysteme zu modellieren. Der Vorteil des Konzeptes liegt in der intuitiven Modellierung und graphischen Anschaulichkeit. SPNs lassen sich als gerichteter Graph darstellen, bei dem Kanten die als Stellen und Transitionen bezeichneten Knoten verbinden. Stellen bezeichnen statische Elemente, wie zum Beispiel eine Warteschlange oder ein Lager, während Transitionen Aktivitäten, wie die Ankunft von Kunden oder den Ausfall einer Maschine modellieren.

Die diskreten Zustände des modellierten Systems werden durch die Markierung (der Belegung der Stellen mit Marken) beschrieben und Aktivitäten durch das Schalten bzw. Feuern der Transitionen modelliert.

Ein Beispiel für ein stochastisches Petrinetz einer Maschine zeigt die Abbildung 2.1.

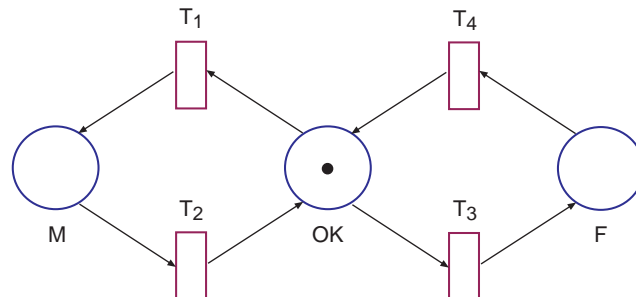


Abbildung 2.1: Ein stochastisches Petrinetz: Maschine arbeitet (OK), Maschine wird gewartet (M), Maschine ist ausgefallen(F)

Dabei repräsentieren die Stellen mögliche Zustände: „Maschine arbeitet“ (OK), „Maschine wird gewartet“ (M) oder „Maschine ist ausgefallen“ (F). Die Transitionen beschreiben die Zustandsübergänge, das heißt sie verändern den Zustand des Netzes.

Bei den in dieser Arbeit betrachteten stochastischen Petrinetzen (SPNs) lassen sich zwei Arten von Transitionen unterscheiden. Dies sind einerseits die zeitbehafteten Transitionen, denen zufällige Feuerzeit zugrunde liegt. Für das Beispielnetz kann das bedeuten, das die Zeit bis zum Ausfall der Maschine normalverteilt mit Mittelwert  $\mu$  und einer Standardabweichung  $\sigma$  ist. Die Aktivierungszeit der Transition ist dann die Zeit, von der Aktivierung bis zum Feuern der Transition. Weiterhin gibt es die zeitlosen Transitionen, diese schalten mit einer bestimmten Wahrscheinlichkeit, sobald sie aktiviert sind, also ohne Verzögerung in der Zeit.

Die Menge aller möglichen erreichbaren Markierungen wird als Erreichbarkeitsgraph des SPNs bezeichnet. Der Erreichbarkeitsgraph für das Petrinetzbeispiel ist in Abbildung 2.2 dargestellt. Jeder Kreis repräsentiert eine eindeutige Markierung des Netzes und die Elemente in den Kreisen beschreiben die Anzahl der Marken in den einzelnen Stellen (OK, M, F), die gerichteten Kanten bezeichnen die Zustandsübergänge, die von der entsprechenden Transition ausgelöst werden.

Es gibt viele Erweiterungen für stochastische Petrinetze, um verschiedene Analyseprobleme zu lösen. Eine dieser Erweiterungen sind die stochastischen Reward-Netze, die im folgenden Abschnitt kurz beschrieben werden.



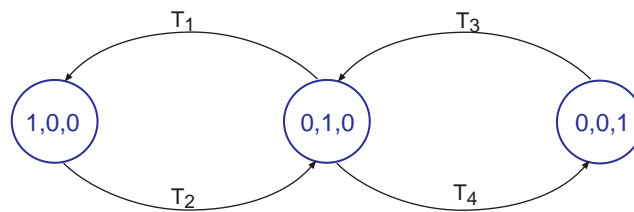


Abbildung 2.2: Erreichbarkeitsgraph des SPNs

## 2.2 Stochastische Reward-Netze

Stochastische Reward-Netze [2] stellen eine Erweiterung der generalisierten stochastischen Petrinetze [10](GSPN) dar. Sie wurden entworfen, um kombinierte Aussagen über die Leistung und Zuverlässigkeit von Systemen treffen zu können. Das GSPN wird dabei um die Definition von Bewertungsmaßen (Rewards) ergänzt. Dabei lassen sich zwei Arten von Rewards unterscheiden:

- Rate Rewards: diese werden einem Zustand zugeordnet
- Impulse Rewards: diese werden auf einem Zustandsübergang definiert

Die Rate Rewards können zum Beispiel als Kosten pro Zeiteinheit verstanden werden, die in einem Zustand fällig werden, oder aber als Gewinn, den das System in diesem Zustand erwirtschaften kann. Mit Impulse Rewards lassen sich Gewichtungen während des Überganges von einem Zustand in einen anderen angeben. Für die Analyse der Leistungsfähigkeit für das stochastische Petrinetz aus Abbildung 2.1 könnte das bedeuten, das bei jedem Ausfall der Maschine Kosten in Höhe von hundert Euro fällig werden, oder wenn die Maschine arbeitet ein Gewinn von dreihundert Euro erwirtschaftet wird.

Der wesentliche Unterschied zu den in Abschnitt 2.1 beschriebenen stochastischen Petrinetzen besteht nun darin, dass zum einen alle zeitbehafteten Transitionen eines SRNs exponentialverteilt sind und zum anderen zusätzlich Rewards in Abhängigkeit von der Markierung des Netzes angegeben werden können.

Viele reale Simulationsprobleme enthalten Zustandsübergänge, bei denen die Markov-eigenschaft nicht erfüllt ist. Gemeint sind Verteilungsfunktionen, wie die Weibull-, die Deterministische oder die Uniform Verteilung. Daher ist es sinnvoll den Formalismus zu erweitern. Dies ist auch sinnvoll, da eines der Vorteile des Proxelverfahrens ist allgemein verteilte Zustandsübergänge analysieren können.

Die in dieser Arbeit betrachteten adaptierten stochastischen Reward-Netze (aSRN) lassen sich abweichend zu der in [2] gegebenen Definition formal wie folgt beschreiben:

$$aSRN = (P, T, A, G, m_0, R)$$

mit:

1.  $P = \{p_1, p_2, \dots, p_n\}$ , Endliche Menge von Stellen
2.  $T = \{t_1, t_2, \dots, t_k\}$ , Endliche Menge von Transitionen. Jede Transition ist definiert als  $t_i = (DF, type)$ , wobei DF die Verteilungsfunktion einer zeitbehafteten Transition ist. Eine zeitlose Transition erhält einen Wert für die Wahrscheinlichkeit.  $type \in \{age, enable, immediate\}$  ist der Typ der Verdrängungsstrategie für zeitbehaftete Transitionen und *immediate* für zeitlose Transitionen.
3.  $A = A_I \cup A_O \cup A_H$ , Menge der Kanten, wobei  $A_O$  die Menge der Ausgangskanten (von  $t$  zu  $p$ ),  $A_I$  Menge der Eingangskanten (von  $p$  zu  $t$ ) und  $A_H$  die Menge der hemmenden Kanten (von  $p$  zu  $t$ ). Jede der Kanten besitzt eine gewisse Multiplizität.
4.  $G = \{g_1, g_2, \dots, g_l\}$ , Menge der Guardfunktionen (Wächterfunktionen), die zu Transitionen gehören.
5.  $m_0$ , ist die initiale Markierung des aSRN. Durch  $M = \{m_0, m_1, m_2, \dots\}$  wird die Menge aller erreichbaren Markierungen beschrieben. Jede Markierung enthält die Anzahl der Marken, die sich in den einzelnen Stellen befinden. Die Notation  $\#(p, m)$  wird benutzt, um die Anzahl der Marken in Stelle  $p$  in der Markierung  $m$  anzugeben. Ist die Markierung aus dem Kontext heraus bekannt, wird  $\#(p)$  benutzt.
6. Die letzte Komponente der aSRN Spezifikation definiert die Bewertungsmaße des aSRN.

$R = \{sr_0, sr_1, sr_2, \dots\}$ , ist eine abzählbare Menge von Bewertungsmaßen. Jedes Maß  $sr \in R$  ist ein Zustands-Reward.  $sr_i$  der ist der Zustands-Reward für die Markierung  $m_i$  des adaptierten stochastischen Petrinetzes.

□

1. - 5. definieren dabei stochastische Petrinetze, wie sie bisher zur Analyse durch das Proxelverfahren verwendet werden. Mit 6. ist es möglich alle Markierungen des Erreichbarkeitsgraphen des stochastischen Petrinetzes zu bewerten.

Bevor im folgenden Kapitel ein Proxelverfahren für adaptierte stochastische Reward-Netze entwickelt wird, soll zunächst eine Einführung in das Proxelverfahren gegeben werden.

## 2.3 Das Proxel-basierte Verfahren

Das Proxelverfahren ist ein algorithmisches Verfahren zur Analyse von diskreten stochastischen Modellen, wie zum Beispiel von stochastischen Petrinetzen [4]. Dabei bilden Proxel (probability elements) die Basisrecheneinheit des Verfahrens.

Ein Proxel  $P$  trägt die Informationen über den Systemzustand  $S$  und der Wahrscheinlichkeit  $p$  zu einem bestimmten Zeitpunkt in diesem Zustand zu sein.

Im Fall eines stochastischen Petrinetzes besteht der Systemzustand aus der Markierung  $m$ , den Aktivierungszeiten der Transitionen  $\tau_i$  und der Systemzeit  $t$ . Hierbei sind die Aktivierungszeiten der deaktivierten Transitionen mit der Verdrängungsstrategie „Race Age“ und alle aktivierten Transitionen von Interesse. Diese Elemente des Alterungsvektors sind beim Proxelverfahren die Hilfsvariablen.

Die Hilfsvariablen sind notwendig für eine komplette Definition des Zustandes des Petrinetzes, weil die Schaltzeit jeder zeitbehafteten Transition davon abhängig ist, wie lange die Transition bereits aktiviert war [9].

Formal lässt sich ein Proxel damit folgendermaßen beschreiben:

$$P = (S, p) \tag{2.1}$$

$$S = (m, \tau, t) \tag{2.2}$$

$$m = (m_0, m_1, \dots, m_n), n = \text{Anzahl der Stellen im SPN} \tag{2.3}$$

$$\vec{\tau} = (\tau_1, \tau_2, \dots, \tau_j), \text{ Alterungsvektor} \tag{2.4}$$

Die Idee des Verfahrens ist nun, diese Proxel zur Approximation des dynamischen Verhaltens des Modells zu nutzen, das heißt, den Wahrscheinlichkeitsfluss durch die erreichbaren Zustände des Systems zu bestimmen.

Die Analyse startet dazu mit einem initialen Proxel, dieser gibt an, dass mit einer Wahrscheinlichkeit von 1.0 in der initialen Markierung des SPN die Aktivierungszeiten der Transitionen und die Simulationszeit null sind.

Ausgehend von diesem Proxel werden dann iterativ die in einem diskreten Zeitschritt  $dt$  erreichbaren Folgeproxel bestimmt und der Wahrscheinlichkeitsfluss zu diesen ermit-

telt. Dadurch entsteht eine Baumstruktur, die den erreichbaren Zustandsraum in Form von Proxels repräsentiert. Für das stochastische Petrinetz aus Abschnitt 2.1 würde sich dabei die folgende Struktur des Proxelbaumes ergeben.

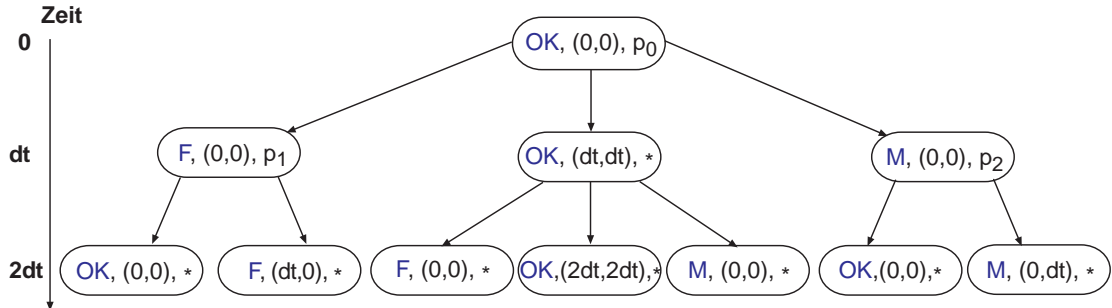


Abbildung 2.3: Struktur des Proxelbaums für das SPN

### Ein Beispiel

Anhand des folgenden Beispiels soll die Funktionsweise der Proxelmethode verdeutlicht werden. Zur Erläuterung wird das in Abbildung 2.4a) dargestellte Petrinetz verwendet. Abbildung 2.4b) zeigt zusätzlich den Erreichbarkeitsgraphen des Modells. Das Petrinetz stellt ein Wartungsmodell einer Maschine dar. Es gibt zwei mögliche Zustände zum einen „Maschine arbeitet“ (Work) und zum anderen „Maschine wird gewartet“ (Maint).

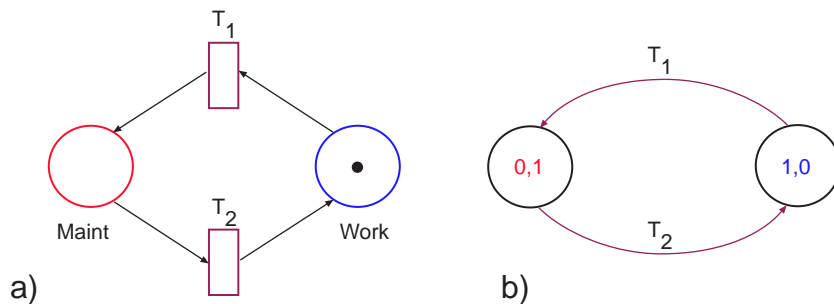


Abbildung 2.4: ein einfaches Petrinetz (a) und dessen Erreichbarkeitsgraph (b)

Die initiale Markierung (Work) repräsentiert den initialen Proxel zum Zeitpunkt  $t = 0$  mit einer Wahrscheinlichkeit von  $p_0 = 1.0$ . Der Alterungsvektor besteht aus den Aktivierungszeiten der zwei Transitionen  $T_1$  und  $T_2$ . Die Transition  $T_1$  ist in der initialen Markierung aktiviert, aber es ist noch keine Zeit vergangen, d.h. diese Transition ist noch

nicht gealtert. Daher sind die Einträge des Alterungsvektors  $\tau = (0, 0)$ . Die Abbildung 2.5 zeigt den soeben beschriebenen initialen Proxel.

$$\text{Work} \\ (\tau_1 = 0, \tau_2 = 0), t = 0, p_0 = 1.0$$

Abbildung 2.5: initialer Proxel

Aus diesem werden nun für den diskreten Zeitschritt die möglichen Entwicklungen betrachtet:

- $T_1$  ist in der aktuellen Markierung aktiviert und feuert. Danach befindet sich eine Marke in der Stelle Maint
- $T_1$  ist aktiviert, feuert aber nicht. Die Markierung ändert sich nicht, d.h. es findet kein Zustandsübergang statt.

Findet kein Zustandsübergang statt, so wird das Alter der aktivierten Transition um die Zeit der Aktivierung erhöht (d.h.  $\tau_1$  ist jetzt für die Zeit  $dt$  aktiviert). Beim Zustandswechsel zu Maint wird die Transition  $T_1$  mit dem Alter 0 neu aktiviert.

Die Wahrscheinlichkeit für den Zustandsübergang errechnet sich aus der Wahrscheinlichkeit des initialen Proxel und der Instantaneous Rate Function [6]  $\mu$  der Verteilungsfunktion der Transition  $T_1$ .

$$p_1 = 1.0 * \mu(\text{bisherige Aktivierungszeit der Transition } T_1) * dt \\ p_2 = 1 - p_1$$

Die Instantaneous Rate Function wird benutzt, um die Wahrscheinlichkeit zu bestimmen, dass die Transition im nächsten Zeitintervall  $(\tau, \tau + dt)$  schalten wird, falls sie für die Zeit  $\tau$  aktiviert war. Diese Wahrscheinlichkeit wird daher durch  $\mu(\tau) \cdot dt$  approximiert (vgl. [9]).

Für jedes der neu erzeugten Proxel wird dann das gleiche Verfahren erneut angewendet, bis eine vom Benutzer vorgegebene Endzeit erreicht wird. Die Abbildung 2.6 veranschaulicht den erzeugten Proxelbaum für die ersten drei Zeitschritte.

Bevor im kommenden Kapitel ein angepasstes Proxelverfahren entwickelt wird, bei dem nicht alle Proxel analysiert werden, ist es wichtig zum Abschluss noch einige wichtige Eigenschaften des Proxelverfahrens zu betrachten.

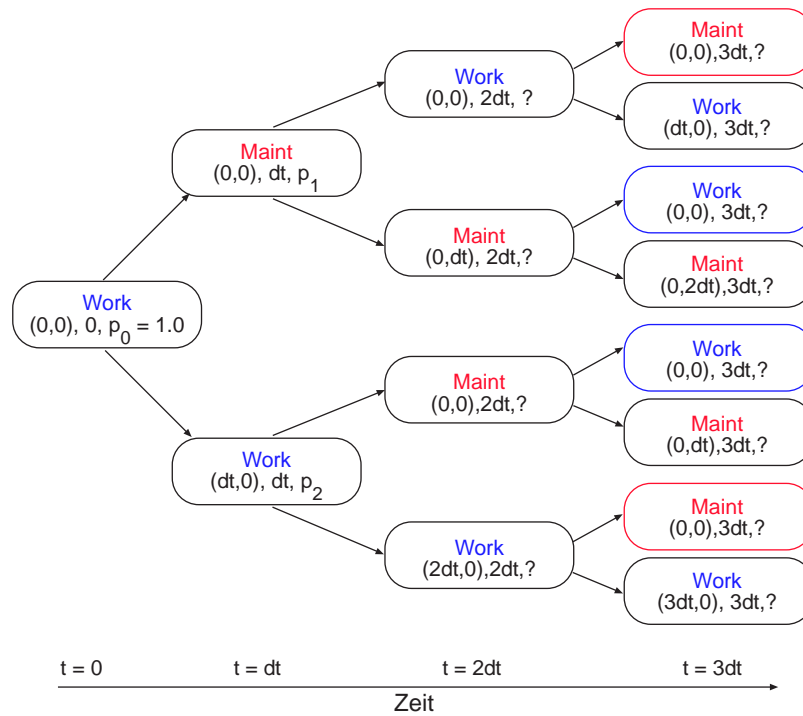


Abbildung 2.6: erste Schritte der Proxel-basierten Simulation

### Eigenschaften des Proxelverfahrens

Der Proxel-Algorithmus approximiert den kontinuierlichen stochastischen Prozess durch Benutzung einer Zeitschrittweite  $dt$ . Aus diesem Grund ist die Genauigkeit der Approximation von der Zeitschrittweite abhängig. Das bedeutet, je kleiner der diskrete Zeitschritt gewählt ist, desto genauer ist auch das Ergebnis der Proxel-basierten Analyse.

Eine kleine Schrittweite führt allerdings dazu, dass insgesamt mehr Proxel erzeugt werden. Dies wiederum beeinflusst die Laufzeit, die dadurch exponentiell ansteigt (vgl. [7]).

Eine Möglichkeit das exponentielle Wachstum des Proxelbaumes zu beschränken besteht darin, Proxel mit gleicher Markierung und gleichem Alterungsvektor zusammenzufassen. Hierbei lassen sich die Wahrscheinlichkeiten dieser Proxel aufsummieren. Dadurch wird für die weitere Analyse nur ein Proxel verwendet. In Abbildung 2.6 sind derartige Proxel durch eine farbliche Umrandung (rot bzw. blau) dargestellt.

Das Wachstum des Proxelbaumes kann zusätzlich durch Einführung einer Schwelle für die Wahrscheinlichkeit verringert werden. Dabei wird davon ausgegangen, dass neue Proxel mit einer Wahrscheinlichkeit unter dieser Schwelle nur einen sehr geringen Anteil an der Lösung haben und deshalb nicht weiter betrachtet werden müssen (vgl. [4], [8]).

Werden nun Proxel, die unter dieser Schwelle liegen nicht betrachtet, werden auch die aus ihnen resultierenden Proxel nicht erzeugt und der Zustandsraum kann dadurch „gestutzt“ werden.

Verteilungsfunktionen mit beschränktem Träger, wie zum Beispiel die Uniform Verteilung oder die Deterministische Verteilung sind für das Verfahren besonders vorteilhaft [8]. Diese Verteilungen sind nur für einen gewissen Zeitraum aktiviert. Deshalb erzeugen sie auch nur in dieser Aktivierungszeit Proxel mit einer Wahrscheinlichkeit, die größer als null ist. Das heißt, die Anzahl Proxel, die durch eine solche Transition pro Zeitschritt entstehen ist deutlich geringer.

### 2.4 Zusammenfassung

In diesem Kapitel wurden die Grundlagen beschrieben, die zum Verständnis der weiteren Kapitel notwendig sind. Dabei wurde zunächst aus stochastischen Petrinetzen und stochastischen Reward-Netzen eine veränderte Modellspezifikation formalisiert, die es ermöglicht, für jeden diskreten Zustand des Modells ein Bewertungsmaß, in Form eines Zustands-Rewards anzugeben. Anschließend wurde das Proxel-basierte Verfahren beschrieben. Im folgenden Kapitel wird nun ein angepasstes Proxelverfahren entwickelt, das irrelevante Proxel „überspringt“.

## 3 Ein adaptiertes Proxelverfahren

Im letzten Kapitel wurde die Netzklasse der adaptierten stochastischen Reward-Netze beschrieben. Diese unterscheiden sich in der bisher durch das Proxelverfahren betrachteten Klasse von stochastischen Petrinetzen nur darin, dass zusätzlich Zustand-Rewards für Markierungen des stochastischen Petrinetzes angegeben werden können. Auf diese veränderte Modellspezifikation wird in diesem Kapitel das Proxelverfahren angepasst. Dazu werden zunächst die Anforderungen, die an das adaptierte Proxelverfahren gestellt werden, beschrieben.

Anschließend wird definiert, welches Maß zur Klassifikation von Proxel-Zuständen verwendet wird, das heißt, wie die Menge der diskreten Zustände in ergebnisrelevante und -nichtrelevante Zustände eingeteilt wird. Darauf aufbauend wird die Idee für ein neues Proxelverfahren beschrieben und anschließend die dazu notwendigen Anpassungen am Proxelalgorithmus vorgenommen.

Anhand eines kleinen aSRN Beispiels wird dann die Vorgehensweise des angepassten Proxelverfahrens demonstriert.

### 3.1 Anforderungen an ein angepasstes Proxelverfahren

Ein für adaptierte stochastische Reward-Netze angepasster Proxelalgorithmus, soll die folgenden Anforderungen erfüllen:

1. Relevanz von Systemzuständen

Das neue Verfahren soll zwischen relevanten und nicht-relevanten Proxels unterscheiden. Damit dies möglich ist, muss zunächst definiert werden, wann ein diskreter Zustand des Modells relevant ist.

2. Effizienz

Es wird gefordert, dass sich für einige Modelle Geschwindigkeitsvorteile im Vergleich zum Originalalgorithmus erzielen lassen. Das adaptierte Proxelverfahren



sollte möglichst wenig zusätzlichen Rechenaufwand benötigen.

### 3. Güte der Ergebnisse

Es wird gefordert, dass das neue Verfahren die Ergebnisse mit vergleichbarer Genauigkeit approximiert, wie der Originalalgorithmus.

## 3.2 Relevanz von Systemzuständen

In diesem Abschnitt wird der Klassifikator eingeführt, mit dem das angepasste Verfahren Zustände hinsichtlich ihrer Relevanz unterscheiden soll.

### Reward-relevanter Zustand bzw. R-Zustand

*Eine diskrete Markierung (diskreter Zustand) ist für die Berechnung eines Ergebniswertes relevant, wenn für diese ein Zustands-Reward angegeben wurde, der ungleich Null ist.*

$$sr_i \neq 0$$

### Irrelevanter Zustand bzw. I-Zustand

*Eine diskrete Markierung ist ein I-Zustand, wenn sie zur Berechnung eines Ergebniswertes nicht relevant ist. Ein solcher Zustand hat einen Zustands-Reward von Null.*

$$sr_i = 0$$

Durch diese beiden Definitionen lässt sich die Menge der diskreten Zustände des Modells in zwei Klassen einteilen, zum einen in relevante diskrete Zustände und zum anderen in nicht-relevante Zustände.

Da, wie in Kapitel 2 beschrieben, die Definition eines Proxels den diskreten Zustand beinhaltet, lassen sich somit auch Proxel in zwei Klassen einteilen, was zu relevanten und nicht-relevanten Proxel führt.

### R-Proxel und I-Proxel

*Ein relevantes Proxel oder R-Proxel ist ein Proxel, dessen diskreter Zustand ein R-Zustand ist.*

*Ein I-Proxel ist ein Proxel, dessen diskreter Zustand ein I-Zustand ist.*

Durch diese Definitionen ist es nun möglich ein adaptiertes Proxelverfahren zu entwickeln, das zwischen den Zuständen auf Basis der für einen Zustand angegebenen Rewards unterscheidet.

Im folgenden Abschnitt wird nun die Idee für ein adaptiertes Proxelverfahren beschrieben.

### 3.3 Die zugrundeliegende Idee

In dem bisher existierenden Proxelverfahren für SPN wird davon ausgegangen, dass alle erreichbaren Zustände zur Bestimmung der Zustandswahrscheinlichkeiten analysiert werden müssen. Dadurch werden auch Proxel verarbeitet, die zur Berechnung eines Ergebniswertes nicht relevant sind. Weiterhin führt dies dazu, dass sowohl für den Speicherbedarf als auch für die Laufzeit des Algorithmus im schlimmsten Fall exponentieller Aufwand besteht. Wird die Anzahl der Proxel, die bei der Analyse entstehen also verringert, so könnten damit Effizienzvorteile erzielt werden.

Die Idee für einen angepassten Algorithmus ergibt sich aus dem gerade genannten Problem und besteht darin, nur die R-Proxel, also Proxel, deren diskreter Zustand einen Zustands-Reward ungleich Null hat, wie bisher durch den Originalalgorithmus zu verarbeiten. Die möglicherweise aus einem R-Proxel entstehenden I-Proxel werden „übersprungen“, indem über einen Zwischenschritt die zukünftigen relevanten Nachfolger dieses I-Proxels ermittelt werden.

Um diesen Zwischenschritt für ein bereits untersuchtes I-Proxel nicht mehrfach zu wiederholen, werden dabei sogenannte Masken erstellt. Eine solche Maske kann dann bei erneutem Auftreten des gleichen I-Proxels benutzt werden, um sofort alle zukünftigen R-Proxel zu erzeugen (vgl. Abschnitt 3.4). Das bedeutet, dass I-Proxel viel schneller verarbeitet werden als R-Proxel.

Für die zugrundeliegende Proxelbaumstruktur für das stochastische Petrinetz aus Abschnitt 2.1, würde sich aus dieser Idee die in Abbildung 3.1 dargestellte Struktur ergeben, wenn nur der diskrete Zustand „OK“ relevant ist. Die gestrichelten Pfeile führen dabei zu I-Proxel, die durch diese Vorgehensweise nicht erzeugt werden.

### 3.4 Anpassungen am Proxelalgorithmus

Um die gerade beschriebene Idee für ein angepasstes Proxelverfahren umzusetzen sind keine grundlegenden Veränderungen am Proxelverfahren selbst notwendig. Die in Kapitel 2 beschriebene Grunddatenstruktur eines Proxels kann unverändert übernommen werden. Der veränderte Algorithmus setzt allerdings nunmehr nicht, wie in [8] beschrie-

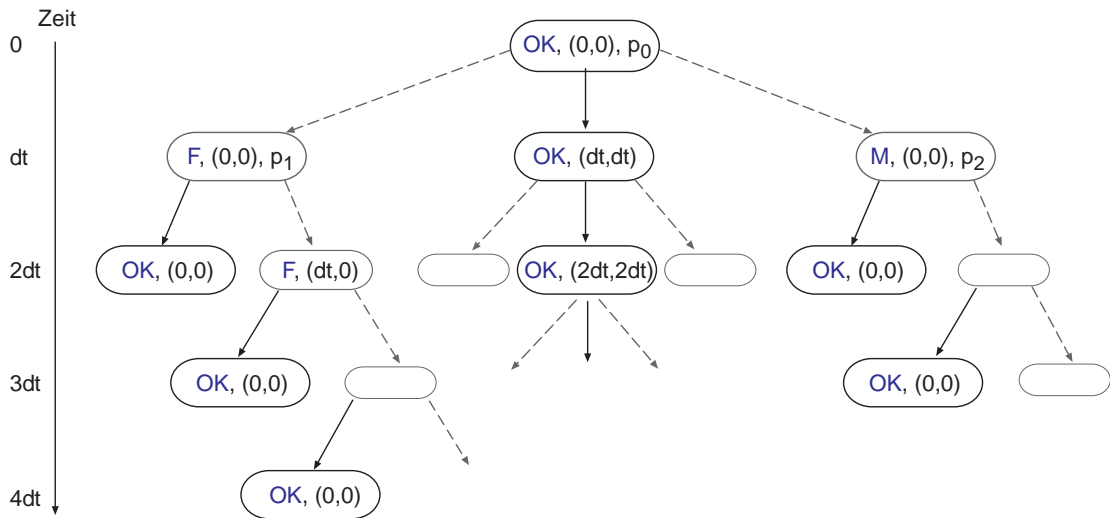


Abbildung 3.1: zugrundeliegende Proxelbaumstruktur

ben, den Erreichbarkeitsgraphen des SPNs voraus, sondern den eines aSRNs, der zusätzlich die Rewards der einzelnen Zustände beinhaltet.

Im Folgenden soll nun der Ablauf des adaptierten Verfahrens und die vorgenommenen Änderungen beschrieben werden. Dazu wird zunächst der allgemeine Ablauf im Pseudocode präsentiert. Anschließend wird auf die Erstellung und Verwendung der Masken eingegangen.

---

**Eingabe** : Zeitschrittweite  $dt$ , maximale Simulationszeit  $T_{max}$

---

- 1 füge initiales Proxel  $P_0$  in Datenstruktur ( $t = 0$ ) ein
  - 2 **für**  $t = 0$  **bis**  $\lceil T_{max}/dt \rceil$  **tue**
  - 3     **für jedes** Proxel  $P$  aus Datenstruktur ( $t$ ) **tue**
  - 4         **für jedes** Folgeproxel  $F_p$  von  $P$  **tue**
  - 5             **wenn** Zustands-Reward( $F_p$ )  $\neq 0$  **dann**
  - 6                 füge R-Proxel  $F_p$  in Datenstruktur ( $t + 1$ ) ein
  - 7             **sonst**
  - 8                 **wenn** Maske für  $F_p$  unbekannt **dann**
  - 9                     suche zukünftige R-Proxel und erzeuge Maske
  - 10                 erzeuge zukünftige R-Proxel und füge in Datenstruktur ( $t + *$ ) ein
-

Um die Wahrscheinlichkeit eines relevanten Zustandes zu bestimmen, startet der angepasste Algorithmus genau wie der Originalalgorithmus in einem oder mehreren Anfangszuständen. Es wird das entsprechende Proxel einer Initialmarkierung gewählt und in die Datenstruktur des ersten Zeitschrittes eingefügt. Dieses initiale Proxel kann entweder ein R-Proxel oder ein I-Proxel sein.

Für jede in der Markierung des Proxels aktivierte Transition wird der Folgezustand bestimmt, dessen Wahrscheinlichkeit berechnet und der Alterungsvektor aktualisiert. Dann wird die Relevanz dieses Folgezustandes überprüft. Hat der Folgezustand einen Zustands-Reward ungleich 0, wird er in die Datenstruktur des folgenden Zeitschrittes eingetragen. Andernfalls werden über den bereits erwähnten Zwischenschritt die zukünftigen relevanten Nachfolger ermittelt.

Zur Bestimmung der relevanten Nachfolger kamen zwei Suchmöglichkeiten in Frage: zum einen eine Tiefensuche und zum anderen eine Breitensuche. Die ursprüngliche Idee war eine Tiefensuche zu verwenden, da diese speichereffizienter ist. In Experimenten stellte sich jedoch heraus, dass sich eine Breitensuche besser eignet, da während der Suche nach R-Proxels irrelevante Proxel mit gleichem Zustand und gleichen Alterungsvariablen, genau wie beim Originalverfahren, zusammengefasst werden können. Dadurch müssen insgesamt weniger irrelevante Proxel während der Suche betrachtet werden. Weiterhin zeigte sich, dass dies auch eher der in Abschnitt 3.1 genannten Anforderung zwei, das möglichst wenig zusätzlicher Rechenaufwand benötigt werden soll, entspricht.

#### **Erstellen von Masken**

Würde der Zwischenschritt der Suche jedes Mal durchgeführt, wenn das I-Proxel im Verlauf der Analyse erneut auftritt, hätte dies einen hohen zusätzlichen Rechenaufwand zur Folge, was der zweiten Anforderung (vgl. Abschnitt 3.1) widersprechen würde. Dieser Aufwand kann eingeschränkt werden, denn für jedes I-Proxel mit den gleichen Alterungsvariablen entstehen immer die gleichen relevanten Proxel in der Zukunft (vgl. Abbildung 3.2). Deswegen werden diese beim ersten Auftreten eines I-Proxels, nur einmal gesucht und die relativen Daten in einer sogenannten Maske (vgl. Abbildung 3.3) gespeichert. Beim nächsten Auftreten des gleichen I-Proxels kann die Maske dann benutzt werden, um alle R-Proxel zu erzeugen.

Diese Masken sind Listen, die für jeden irrelevanten Zustand in einem Binärbaum gehalten werden. Als Schlüssel dient ein Hashwert, der aus den Alterungsvariablen und dem Zustand des aufgetretenen I-Proxels berechnet wird.

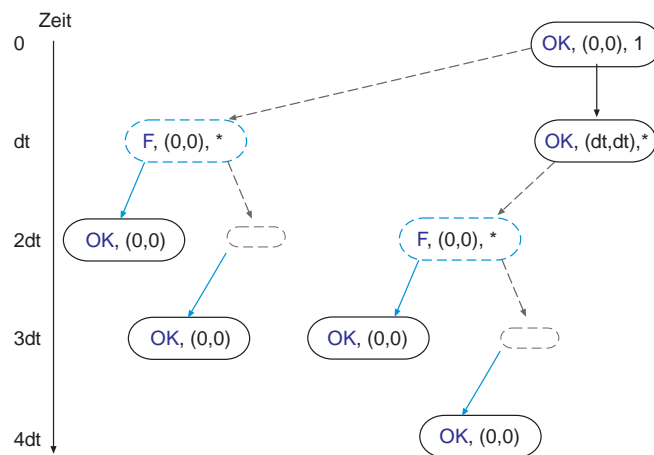


Abbildung 3.2: Gleiche R-Proxel

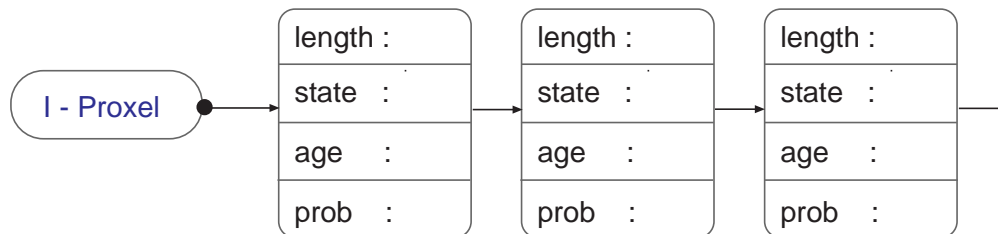


Abbildung 3.3: Datenstruktur einer Maske

Erstellt wird eine Maske, wie bereits erwähnt, während der Suche. Diese beginnt mit dem aktuellen Zeitschritt des aufgetretenen I-Proxels. Es wird allerdings von diesem I-Proxel ein dummy-Proxel abgeleitet, welches eine Wahrscheinlichkeit von 1.0 und einen relativen Zeitschritt von null hat. Dadurch ist es während der Suche möglich die relativen Daten eines R-Proxel zu bestimmen, das heißt:

- Die relative Wahrscheinlichkeit (normierten Wahrscheinlichkeiten), mit der relevante Proxel über einen bestimmten Pfad des I-Proxel auftreten. Diese werden unter „prob“ in die Maske eingetragen.
- Der relative Zeitschritt, in dem relevante Proxel aus dem I-Proxel entstehen würden. Dieser wird unter „length“ in die Maske eingetragen.
- Neben diesen relativen Daten, muss die Maske auch den Zustand und die Alterungsvariablen enthalten, da unterschiedliche Zustände des Modells relevant sein könnten.

Diese Daten für die relevanten Proxel werden während der Suche nach dem relativen Zeitschritt sortiert in die Maske eingetragen. Das hat den Vorteil, dass beim Verwenden der Masken die Erzeugung von R-Proxels abgebrochen werden kann, wenn die maximale Simulationszeit  $T_{max}$  erreicht wurde.

Während der Suche nach relevanten Nachfolgern eines I-Proxels können auch irrelevante Proxel entstehen, für die bereits eine Maske vorliegt. Um den Pfad eines solchen irrelevanten Proxels nicht nochmals nach relevanten Proxels zu durchsuchen, wird die bereits vorhandene Maske benutzt.

Die Suche nach relevanten Proxel wird, abgebrochen, wenn der Wert der Wahrscheinlichkeit eines Suchpfades unter einem Schwellenwert  $\epsilon$  liegt. Hierbei kann der gleiche Schwellenwert gewählt werden, wie das Originalverfahren verwendet, da während der Suche nur die relative Wahrscheinlichkeit eines Proxels betrachtet wird.

Der folgende Pseudocode verdeutlicht die Erstellung der Masken.

---

**Eingabe** : Zeitschrittweite  $dt$ , maximale Simulationszeit  $T_{max}$ , Schwellenwert  $\epsilon$

**Eingabe** : I-Proxel  $P_i$  mit  $P_i.t = step$

```

1 erzeuge Maske für  $P_i$ 
2 füge dummyProxel  $P_d$  in Suchdatenstruktur ein
3 für  $t = step$  bis  $[T_{max}/dt]$  tue
4   für jedes Proxel  $P$  aus Suchdatenstruktur tue
5     für jedes Folgeproxel  $Fp$  von  $P$  tue
6       wenn  $Fp.prob \geq \epsilon$  dann
7         wenn  $Zustands-Reward(Fp) \neq 0$  dann
8           füge R-Proxel  $Fp$  sortiert in Maske für  $P_i$  ein
9         sonst
10          wenn Maske für  $Fp$  unbekannt dann
11            füge  $Fp$  in Suchdatenstruktur ein
12          sonst
13            füge Daten der Maske von  $Fp$  sortiert in Maske für  $P_i$  ein

```

---

#### Verwendung der Masken

Jedes Mal, wenn aus einem R-Proxel ein I-Proxel entstehen würde, wird im Binärbaum des irrelevanten Zustandes nach einer Maske mit den entsprechenden Koordinaten des I-Proxel gesucht. Wenn diese Maske bereits existiert, dann können sofort alle zukünftigen relevanten Proxel durch lineares Durchlaufen der Maske erzeugt werden.

Der Zeitpunkt  $t$ , zu dem ein R-Proxel gültig ist, berechnet sich aus dem in der Maske eingetragenen relativen Zeitschritt „length“ und dem Zeitpunkt des aufgetretenen I-Proxels. Die Wahrscheinlichkeit  $p$  für ein R-Proxel zu diesem Zeitpunkt ergibt sich durch Multiplikation der Wahrscheinlichkeit des I-Proxels mit der in der Maske eingetragenen relativen Wahrscheinlichkeit „prob“. Da der Alterungsvektor bereits während der Erstellung der Maske aktualisiert wurde, kann er genau wie in der Maske eingetragen, übernommen werden. Gleiches gilt für den relevanten diskreten Zustand.

Da es sich bei der Maske um eine nach dem zeitlichen Abstand sortierte Liste handelt, kann die Erzeugung der R-Proxel abgebrochen werden, wenn die maximale Simulationszeit  $T_{max}$  eines R-Proxels erreicht wurde. Das hat den Vorteil, dass die Maske mit fortschreitender Simulationszeit bei erneutem Auftreten eines I-Proxel nicht bis zum Ende durchlaufen werden muss.

#### Datenstruktur zur temporären Speicherung von R-Proxel

Durch die Anpassung des Proxelverfahrens auf die Analyse von nur relevanten Zuständen ändert sich auch die Anforderung, die an die Datenstruktur zur Verarbeitung der R-Proxel gerichtet ist.

Werden beim Originalalgorithmus zwei Binärbäume (einer für den aktuellen und einer für den folgenden Zeitschritt) zur temporären Speicherung der Proxel benutzt, so muss die Datenstruktur nun in der Lage sein auch R-Proxel aufzunehmen, die zu zukünftigen Zeitschritten gültig sind.

Dazu ein Beispiel: aus einem R-Proxel im Zeitschritt eins würde ein I-Proxel im Zeitschritt zwei entstehen, da dieses übersprungen wird, daraus aber ein R-Proxel im Zeitschritt vier entsteht, sollte das R-Proxel auch erst im vierten Zeitschritt durch das Verfahren weiterverarbeitet werden (vgl. Abbildung 3.4).

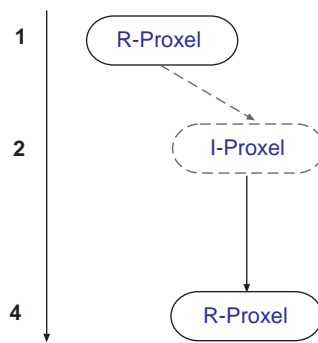


Abbildung 3.4: Beispiel: aus einem I-Proxel entstehender R-Proxel im vierten Zeitschritt

Um diese Anforderung zu erfüllen, wird, wie in Abbildung 3.5 beispielhaft dargestellt, für jeden möglichen Zeitschritt ein Binärbaum erzeugt. Jedes Mal, wenn dann ein R-Proxel für einen Zeitschritt  $t+x$  erzeugt wird, wird er in den entsprechenden Binärbaum ( $t+x$ ) eingefügt und erst wenn der Zeitschritt  $t+x$  erreicht wurde, weiterverarbeitet.

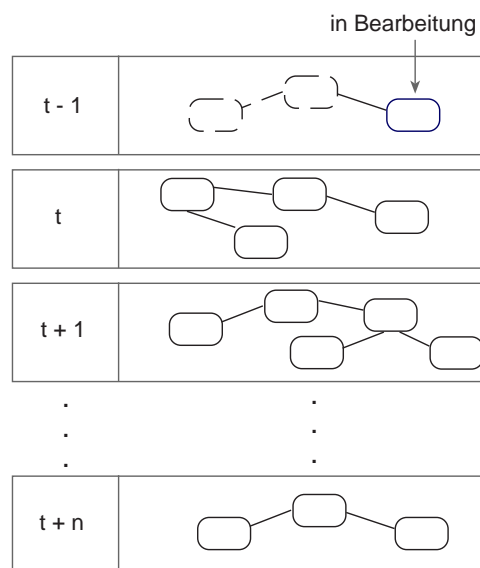


Abbildung 3.5: Datenstruktur zur temporären Speicherung von R-Proxel



### Masken für R-Proxel

Neben der Erstellung von Masken für I-Proxel, wurde zusätzlich ein Caching von R-Proxel Übergängen, wie unter [12] beschrieben, implementiert. Dieses Caching funktioniert ähnlich wie die Erstellung der Masken für I-Proxel. Dabei erhält jedes R-Proxel für jede gefeuerte Transition eine Referenz auf seine Nachfolger (vgl. Abbildung 3.6). Beim nächsten Auftreten des R-Proxels können dadurch die relevanten Nachfolger schneller berechnet und erzeugt werden.

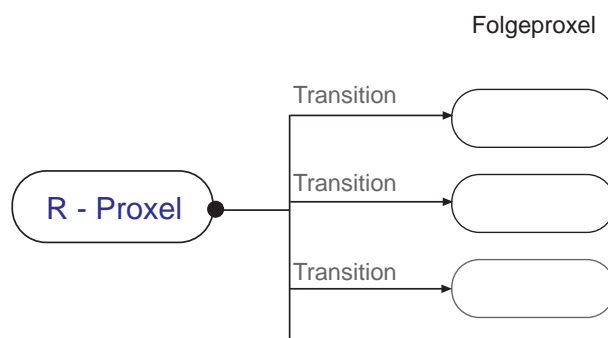


Abbildung 3.6: Datenstruktur eines R-Proxel

### 3.5 Ein Beispiel

Anhand des folgenden Beispiels soll die Funktionsweise des auf aSRNs angepassten Proxelverfahren verdeutlicht werden. Zur Erläuterung wird das in Abbildung 3.7 dargestellte Modell verwendet. Dieses stellt ein Wartungsmodell einer Maschine dar. Es existieren zwei mögliche Zustände zum einen kann die Maschine funktionstüchtig sein (Work) und zum anderen gewartet werden (Maint). Die zu beantwortende Fragestellung lautet: „Mit welcher Wahrscheinlichkeit ist die Maschine funktionstüchtig?“. Daher wird für „Work“ ein Zustands-Reward ungleich null gewählt.

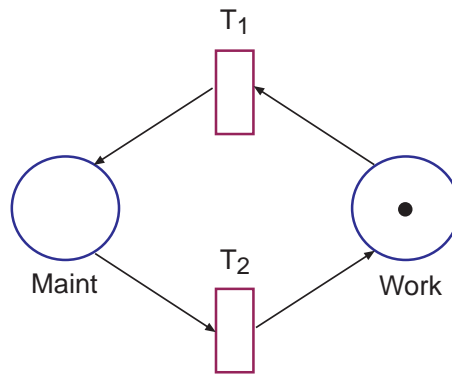


Abbildung 3.7: aSRN eines Wartungsmodells

Als Verteilungsfunktionen sei für  $T_1 = \text{Weibull}(1.0, 2.0)$  und für  $T_2 = \text{Uniform}(0.0, 1.0)$  angenommen. Wählt man eine Schrittweite von  $\Delta t = 0.2$ , so entsteht aus dem initialen Proxel  $((\text{Work}, 0), 0, 1.0)$  nur ein relevanter Nachfolger, was durch die Wahl der Uniform-Verteilung bedingt ist.

Aus diesem ergeben sich die in Abbildung 3.8 dargestellten Entwicklungsmöglichkeiten für den zweiten Zeitschritt:

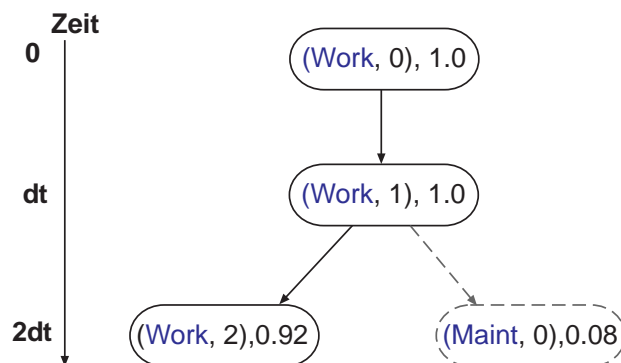


Abbildung 3.8: erste Schritte beim adaptierten Verfahren

- I-Proxel  $((\text{Maint}, 0), t = 2, p = 0.08)$ .

Da dieser einen Reward von null hat, wird nun eine Suche nach zukünftigen R-Proxel durchgeführt. Richtig wäre hier, dass während der Suche eine Maske mit genau fünf Einträgen erzeugt wird, da die Transition  $T_1$  aufgrund der Uniform Verteilung und der gewählten Schrittweite nur in dieser Zeitspanne aktiviert ist und daher auch nur fünf R-Proxel aus diesem I-Proxel entstehen können. Die Abbildung 3.9 zeigt die bei der Suche entstandene Maske. Durch das Durchlaufen der

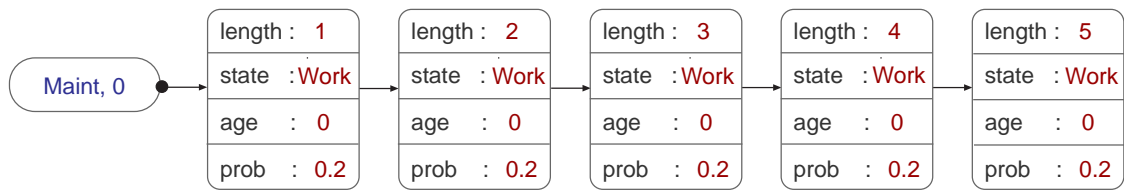


Abbildung 3.9: Durch die Suche nach R-Proxel erzeugte Maske für (Maint,0)

Maske können nun alle relevanten Proxel dieses irrelevanten Pfads erzeugt werden. Die Abbildung 3.10 zeigt den daraus resultierenden Proxelbaum.

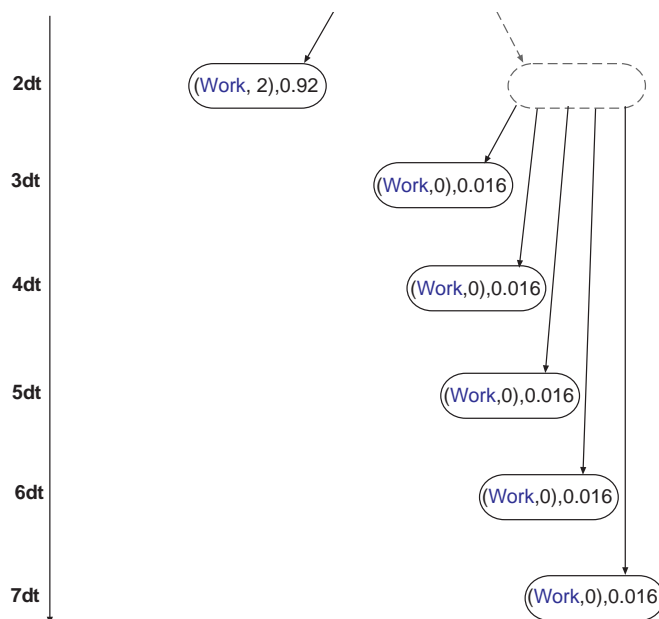


Abbildung 3.10: Ergebnis der Erzeugung relevanter R-Proxel aus dem I-Proxel (Maint,0)

- Proxel ((Work, 2), t = 2, p = 0.92).

Aus diesem würde als irrelevanter Nachfolger ((Maint, 0), t = 3, p = 0.15) entstehen. Da die Maske für (Maint,0) bereits existiert, können aus dieser wiederum alle relevanten Proxel erzeugt werden (vgl. Abbildung 3.11). Zusätzlich werden die in der Abbildung farblich umrandeten Proxel einer Proxelbaumebene ( $k * dt$ ) dabei zusammengefasst.

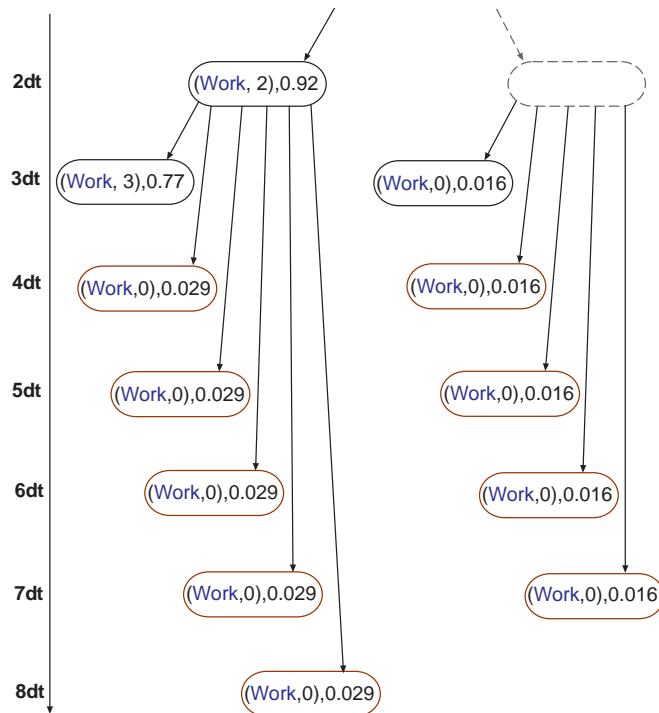


Abbildung 3.11: Ergebnis bei erneutem Auftreten des I-Proxel (Maint,0)

Für dieses Beispielmmodell wird bis zum Ende der Simulationszeit keine weitere Suche nach relevanten Proxel mehr nötig sein, da aus jedem Proxel mit dem Zustand „Work“, das I-Proxel mit dem Systemzustand (Maint,0) entsteht, und deshalb immer die gleiche Maske durchlaufen wird.

### 3.6 Zusammenfassung

In diesem Kapitel, wurde das Proxelverfahren auf die in Kapitel 2 eingeführten adaptierten stochastischen Reward-Netze angepasst. Dazu wurden zunächst die Anforderungen, die an das adaptierte Verfahren gestellt werden, beschrieben. Anschließend wurde das Maß zur Klassifizierung von relevanten und nicht-relevanten Zuständen definiert und gezeigt, dass sich dadurch auch Proxel in zwei Klassen einteilen lassen. Anschließend wurde die Idee für ein angepasstes Verfahren präsentiert, das während der Analyse nur die R-Proxel verarbeitet. Da I-Proxel während einer Analyse aber berücksichtigt werden müssen, um die korrekten Wahrscheinlichkeiten zu bestimmen, wurde ein Zwischenschritt implementiert, der auf einem Maskenkonzept beruht. Dabei werden während einer Suche nach zukünftigen Nachfolgern eines I-Proxels Masken erstellt, und bei erneu-

tem Auftreten des gleichen I-Proxels verwendet. Anhand eines Beispiels wurde anschließend die korrekte Funktionsweise der vorgenommenen Implementierung demonstriert.

Um nun herauszufinden, ob sich mit dem veränderten Proxelverfahren Vorteile gegenüber dem bisherigen Ansatz erzielen lassen, werden im nächsten Kapitel Experimente durchgeführt, mit dem das Laufzeitverhalten des neuen Verfahrens im Vergleich zum Originalverfahren evaluiert wird.

## 4 Experimente

Im vergangenen Kapitel wurde die Idee für ein adaptiertes Proxelverfahren beschrieben, das irrelevante Proxel überspringen kann. Um nun herauszufinden, ob sich durch diese Vorgehensweise Effizienzvorteile im Vergleich zum Originalalgorithmus erreichen lassen, wird in diesem Kapitel das Laufzeitverhalten des angepassten Verfahrens mit dem des Originalalgorithmus verglichen. Dazu wurden lediglich kleine Modelle gewählt, da diese geringe Analysezeiten aufweisen. Im ersten und zweiten Experiment wird das aSRN, das im Verlauf dieser Arbeit häufig zu Illustrationszwecken benutzt wurde, unter verschiedenen Parametrisierungen getestet. Experiment drei evaluiert das Laufzeitverhalten bei einem Warteschlangenmodell und Experiment vier bei einem Modell, bei dem jeder Zustand durch jeden erreicht werden kann. Alle Experimente wurden auf einem Pentium M Notebook mit 1.7 GHz und einem GB Arbeitsspeicher durchgeführt.

### 4.1 Annahmen und Erwartungen

Das neue Verfahren wird bei allen Experimenten mit dem Originalalgorithmus verglichen. Die Maße für den Vergleich sind dabei die Analysezeit pro Zeitschritt und die akkumulierte Analysezeit.

Insbesondere zu Beginn einer Proxel-basierten Analyse wird ein erhöhter Aufwand für das neue Verfahren erwartet, da hier häufig nach relevanten Zuständen gesucht werden muss. Dieser zusätzliche Rechenaufwand ist durch die Erstellung der Masken bedingt. Ist allerdings ein Großteil der Masken erstellt, sollte sich der Analyseaufwand verringern, so dass in folgenden Zeitschritten, im Vergleich zum Originalverfahren weniger Zeit benötigt wird. Durch das neue Verfahren werden nur die relevanten Proxel verarbeitet, deshalb wird erwartet, dass auch die Anzahl der letztendlich verarbeiteten Proxel geringer ausfallen wird als beim Originalverfahren.

## 4.2 Experiment 1: Maschinenmodell 1

Das erste analysierte Modell ist das in Abbildung 4.1 dargestellte adaptierte stochastische Reward-Netz. Dieses kleine Modell eignet sich gut, um das neue Verfahren mit dem Originalalgorithmus zu vergleichen, da es aufgrund der geringen Entwicklungsmöglichkeiten (wenige gleichzeitig aktivierte Transitionen und eine geringe Anzahl von Zuständen) effizient analysiert werden kann. Die Maße für den Vergleich der beiden Verfahren in

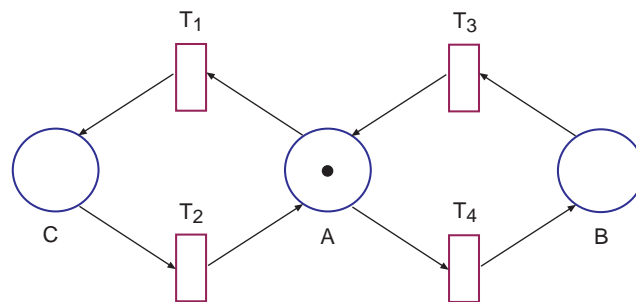


Abbildung 4.1: aSRN eines Maschinenmodells

diesem Experiment sind die akkumulierte Rechenzeit und die Rechenzeit pro Zeitschritt. Dazu wurde das Modell mit den in Tabelle 4.1 aufgeführten Verteilungsfunktionen für die einzelnen Transitionen parametrisiert. Für das neue Verfahren wurden die Zustände

Transition	Typ	Verteilungsfunktion
T1	Age	Deterministisch (3.0)
T2	Enable	Uniform (4.0, 9.0)
T3	Enable	Weibull (8.0, 2.0)
T4	Enable	Weibull (8.0, 2.0)

Tabelle 4.1: Experiment 1 - Parameter des Maschinenmodells

in Abhängigkeit von der Markierung mit der folgenden Reward-Funktion bewertet:

$$sr_i = \begin{cases} 1, & \text{falls } \#(A) = 1 \text{ in Markierung } i \\ 0, & \text{sonst} \end{cases}$$

Diese besagt, dass Zustand A einen Reward von 1 erhält und damit relevant ist und alle anderen Zustände nicht relevant sind. Die Analyse wurde bis zu einer maximalen Simulationszeit von  $T_{max} = 50$  durchgeführt, wobei eine diskrete Schrittweite von  $\Delta t = 0.05$  verwendet wurde.

Das Modell wurde validiert, indem die Analyseergebnisse für den diskreten Zustand A bei Verwendung des adaptierten Verfahrens mit den Ergebnissen des Originalverfahrens verglichen wurden. Wie die Abbildung 4.2 zeigt, approximiert der veränderte Algorithmus die gleichen Wahrscheinlichkeiten, wie der Basisalgorithmus. Das lässt darauf schließen, dass die vorgenommene Implementierung korrekt ist und der veränderte Algorithmus die Güte der Ergebnisse nicht wesentlich beeinflusst.

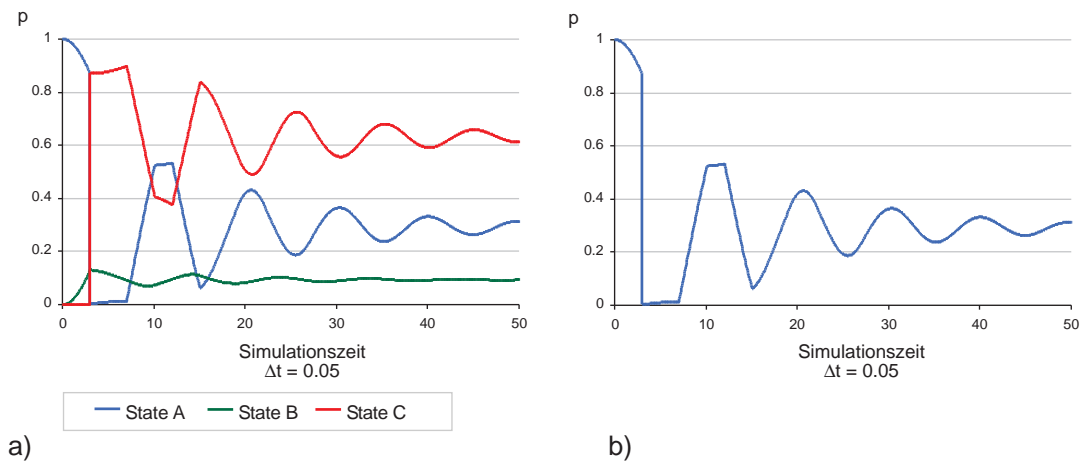


Abbildung 4.2: Transiente Wahrscheinlichkeiten a) Originalverfahren - alle Zustände relevant. b) Neues Verfahren - nur Zustand A relevant

Die Abbildung 4.3 zeigt die Analysezeit pro Zeitschritt zum einen für das Originalverfahren (roter Graph), bei dem alle Zustände relevant sind und zum anderen den Analyseaufwand bei dem nur der Zustand A (grüner Graph) relevant ist. Auffällig ist, dass die Rechenzeit für das adaptierte Verfahren in den ersten Zeitschritten höher ist als beim Originalverfahren und nach ca. 350 Zeitschritten stark abfällt. Beim Originalverfahren hingegen ist die Rechenzeit pro Zeitschritt anfangs zwar geringer, steigt aber mit zunehmender Simulationszeit an.

Dies begründet sich daraus, dass beim neuen Verfahren in den ersten Zeitschritten häufig eine Suche nach dem relevanten Zustand A für unterschiedliche Kombinationen im Alterungsvektor von B und C durchgeführt werden muss, um die Masken zu erstellen. Ab einem bestimmten Zeitpunkt liegen dann alle Masken vor und die zukünftigen relevanten Proxel können sofort durch das Durchlaufen der Masken erzeugt werden. Dies wirkt sich positiv auf die Rechenzeit, die in einem Zeitschritt benötigt wird, aus, was genau den anfänglich beschriebenen Erwartungen entspricht.



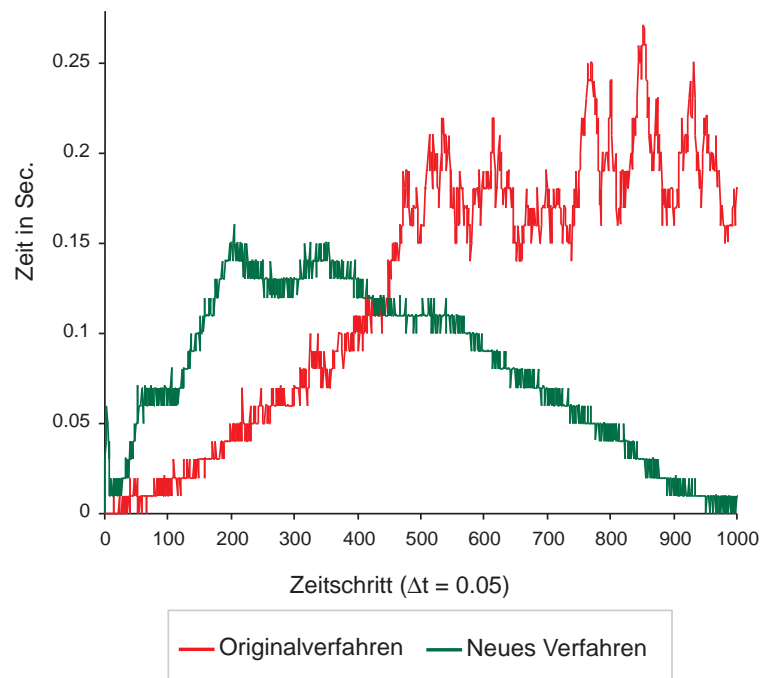


Abbildung 4.3: Analysezeit pro Zeitschritt für das Maschinenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren

In Abbildung 4.4 ist die akkumulierte Analysezeit für das Originalverfahren und das neu entwickelte Verfahren bis  $T_{max} = 50$  dargestellt.

Der rote Graph (Originalverfahren) steigt für dieses Modell linear mit der Zeit. Auffällig ist, dass bei dem neuen Verfahren die Analysezeit bis zur Simulationszeit 37 viel höher ist, ab diesem Zeitpunkt aber nicht mehr stark ansteigt. Das liegt daran, dass der Analyseaufwand pro Zeitschritt bei diesem Modell mit zunehmender Simulationszeit sinkt.

Daher lässt sich für dieses Modell schlussfolgern, dass je länger die Simulationszeit, desto effizienter ist das neue Verfahren im Vergleich zum Originalverfahren. Für den Gesamtanalyseaufwand bis  $T_{max} = 50$  bedeutet das, dass sich ein Geschwindigkeitsvorteil von circa 35 Prozent erzielen lässt. Die Anwendung des adaptierten Verfahrens ist daher bei diesem Modell sehr vorteilhaft.

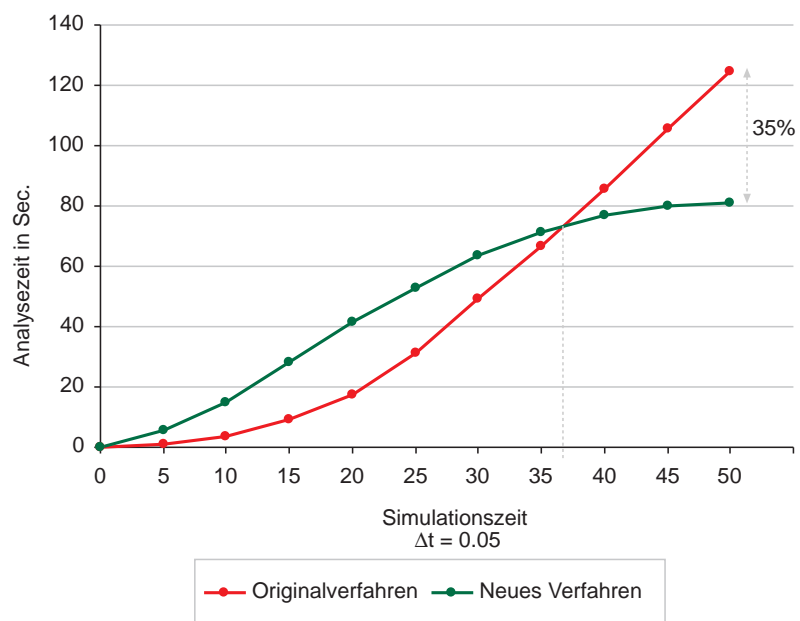


Abbildung 4.4: Analysezeit für das Maschinenmodell - Originalverfahren im Vergleich zum adaptieren Verfahren

### 4.3 Experiment 2: Maschinenmodell 2

Das zweite gestestete Modell ist von der gleichen Struktur, wie das in Experiment 1. Hier wurden lediglich die Verteilungsfunktionen der Transitionen  $T_1$  und  $T_2$  verändert, die nun der Weibullverteilung mit  $\alpha = 1.0$  und  $\beta = 2.0$  unterliegen. Bei diesem Experiment wurden drei Proxel-basierte Simulationen durchgeführt:

- Versuch 1: Zustand A relevant
- Versuch 2: Zustand C relevant
- Versuch 3: Zustände A und C relevant

Die Abbildung 4.5 zeigt die Anzahl der verarbeiteten Proxel bis zur Simualtionszeit  $T_{max} = 50$  für das Originalverfahren (a) und die drei Versuche mit dem adaptierten Verfahren (b). Wie anfänglich erwartet, werden mit dem neuen Verfahren bei allen Versuchen viel weniger Proxel verarbeitet als mit dem Originalverfahren.

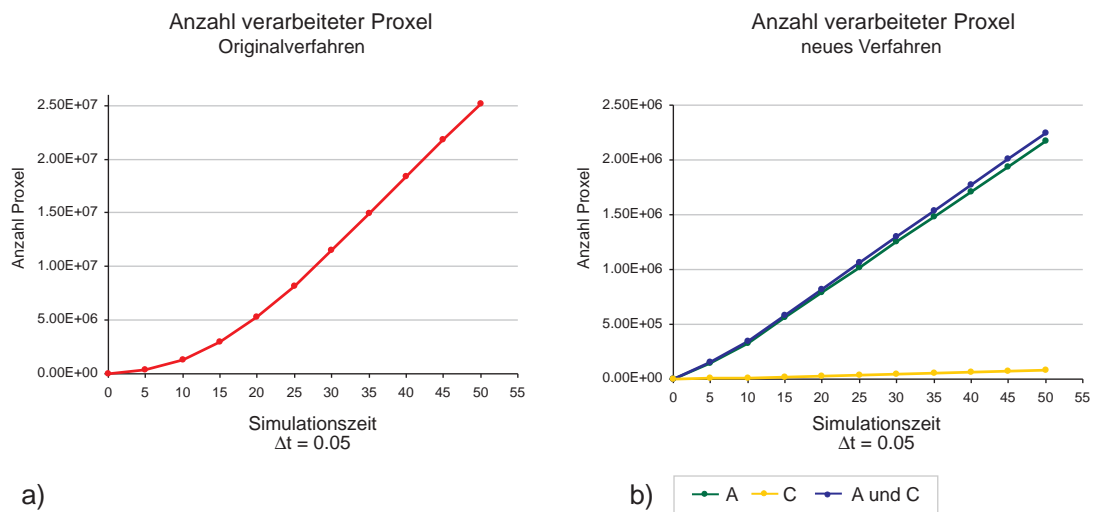


Abbildung 4.5: Anzahl verarbeiteter Proxel a) Originalverfahren b) neues Verfahren (nur A relevant, A und C relevant, nur C relevant)

Die Abbildung 4.6 zeigt die akkumulierte Analysezeit der drei durchgeführten Versuche für das adaptierte Verfahren im Vergleich zum Originalverfahren. Erkennbar ist, dass der Rechenaufwand für das adaptierte Verfahren bei allen drei durchgeführten Versuchen bis  $T \approx 30$  höher ist als beim Originalverfahren. Anschließend steigen die Graphen im Gegensatz zum Originalverfahren nicht so stark an. Dies ist wieder durch die Analy-

seizeit pro Zeitschritt bedingt, die für das neue Verfahren zu Beginn der Analyse höher als beim Originalverfahren ist.

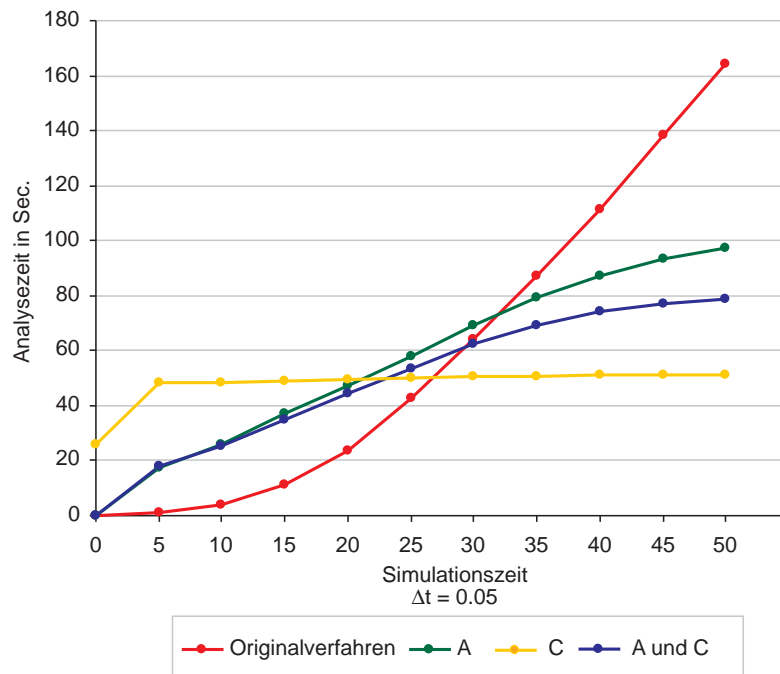


Abbildung 4.6: Analysezeit für das Maschinenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren (bei unterschiedlichen relevanten Zuständen)

Vergleicht man den Graph von Versuch 1 (A relevant) mit dem von Versuch 2 (C relevant), so ist auffällig, dass, obwohl es sich jeweils um die Analyse von einem Zustand handelt, der Rechenaufwand für Versuch 2 bis zur Simulationszeit 20 höher ist als für Versuch 1. Das liegt daran, dass bei Versuch 2 die Analyse mit dem irrelevanten Zustand A startet. Deshalb wird bereits im ersten Zeitschritt eine Suche im irrelevanten Pfad von A mit aktualisierten Alterungsvariablen durchgeführt. Da die Wahrscheinlichkeit eines Auftretens von A höher ist, fallen weniger irrelevante Proxel während der Suche unter die Wahrscheinlichkeitsschwelle. Dadurch ist der Rechenaufwand zu Beginn höher. Auffällig ist allerdings auch, dass der Graph für Versuch 2 ab  $T \approx 5$  fast nicht mehr ansteigt. Der Grund hierfür ist, dass für C ab diesem Zeitpunkt nahezu die gleiche Anzahl an Proxel verarbeitet wird und die Maske für den Übergang zu A bereits vorliegt.

Bei einem Vergleich von Versuch 1 und Versuch 3 fällt auf, dass der Analyseaufwand für Versuch 3 geringer ist, obwohl hier zwei Zustände relevant sind. Dies begründet sich wie folgt: für Versuch 1 müssen Masken für B und C, wohingegen für Versuch 3 nur Masken für B erstellt und gesucht werden.

Als Schlussfolgerung kann aus diesem Experiment festgehalten werden, dass das adaptierte Verfahren, wie in Experiment 1 mit zunehmender Simulationszeit effizienter im Vergleich zum Originalverfahren wird. Bei allen drei durchgeführten Versuchen lässt sich ein Geschwindigkeitsvorteil erzielen. Der erzielte Vorteil hängt von den für die Fragestellung relevanten Zuständen ab.

#### 4.4 Experiment 3: Modell einer Warteschlange

In diesem Experiment wird das in Abbildung 4.7a) dargestellte Modell einer Warteschlange, mit den folgenden Verteilungen für die Transitionen:  $T_1 = \text{Exponential}(1.0)$  und  $T_2 = \text{Deterministisch}(1.0)$ , untersucht. Auch wenn es für die Berechnung nicht notwendig ist, wurden für  $T_1$  Alterungsinformationen gespeichert.

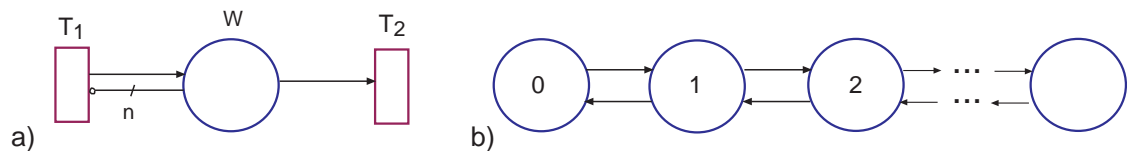


Abbildung 4.7: a) aSRN einer Warteschlange b) Erreichbarkeitsgraph des aSRN

Die Länge der Warteschlange wird in diesem Experiment auf  $n = 10$  begrenzt. Damit existieren 11 mögliche Zustände: Warteschlange ist leer, Warteschlange enthält eine Person, zwei usw. Die Analyse wird bis zu einer maximalen Simulationszeit von  $T_{max} = 50$  mit einer Schrittweite von  $\Delta t = 0.1$  durchgeführt. Untersucht wurde neben den Analyseergebnissen wiederum die Analysezeit pro Zeitschritt und die akkumulierte Analysezeit. Weiterhin wurde getestet, wie sich der Analyseaufwand ändert, wenn mehrere Zustände mit einem Reward ungleich null bewertet werden.

Die akkumulierte Analysezeit für das neue Verfahren, wenn nur der Zustand: „Warteschlange ist leer“ relevant ist, zeigt der grüne Graph in Abbildung 4.8. Hier zeigt sich, dass das neue Verfahren bereits bei  $T \approx 20$  effizienter als das Originalverfahren ist. Erkennbar ist aber wieder die erhöhte Analysedauer bis zu diesem Zeitpunkt.

Man erkennt in der Abbildung auch, dass sich der Analyseaufwand nach dem ersten Zeitschritt nicht wesentlich erhöht. Dieser Fall stellt einen Spezialfall dar, weil hier nur eine einzige Maske erzeugt werden muss. Das liegt daran, dass der irrelevante Zustand „Eine Person in  $W$ “, der aus dem relevanten Zustand „Warteschlange ist leer“ durch Feuern von  $T_1$  auftritt, immer die gleichen Alterungsvariablen hat. Das lässt darauf schlie-

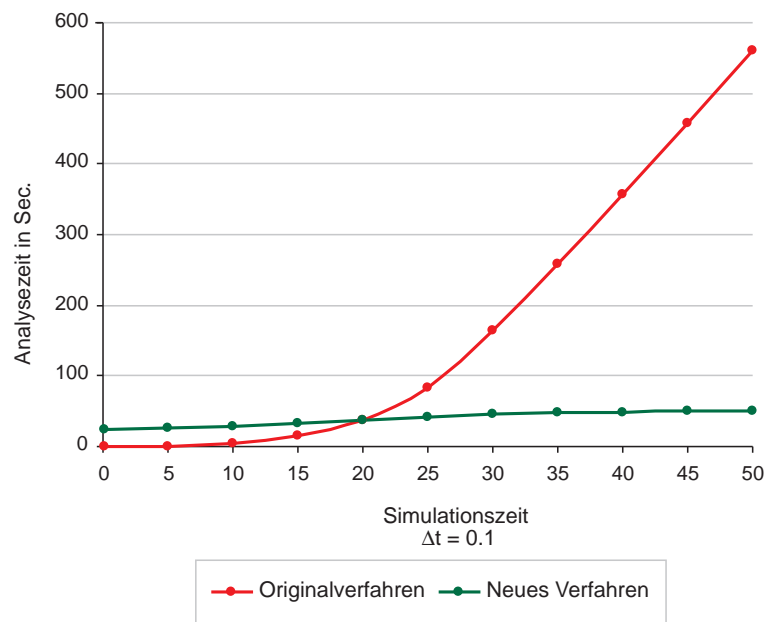


Abbildung 4.8: Analysezeit für das Warteschlangenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren

ßen, dass je weniger Masken erzeugt werden und je weniger relevante Zustände in dieser eingetragen sind, desto weniger Rechenzeit ist nötig.

Betrachtet man die Gesamtanalysezeit bei  $T_{max} = 50$ , so wird deutlich, dass durch das neue Verfahren, bei der ausschließlichen Betrachtung des relevanten Zustandes ein besonders hoher Geschwindigkeitsvorteil erzielt werden kann.

Als nächstes wurde untersucht, welchen Einfluss die Anzahl der relevanten Zustände auf die Analysezeit hat. Dazu wurde die Anzahl der relevanten diskreten Zustände jeweils um eins erhöht. Die Abbildung 4.9 zeigt die Analysezeit pro Zeitschritt für vier verschiedene Fälle im Vergleich zum Originalverfahren.

Auch hier ist erkennbar, dass beim neuen Verfahren besonders zu Beginn einer Analyse ein hoher Rechenaufwand besteht. Dieser aber zum Beispiel bei sieben relevanten Zuständen geringer ausfällt als bei der Analyse von vier relevanten Zuständen. Bei vier relevanten Zuständen sinkt die Analysedauer mit der Zeit, wohingegen sie bei sieben relevanten Zuständen ansteigt. Das liegt daran, dass bei vier relevanten Zuständen zur Erstellung der Masken mehr irrelevante Zustände betrachtet werden müssen als bei sieben relevanten Zuständen. Im weiteren Verlauf der Analyse werden dann bei vier relevanten Zuständen weniger Proxel pro Zeitschritt verarbeitet, als bei sieben relevanten

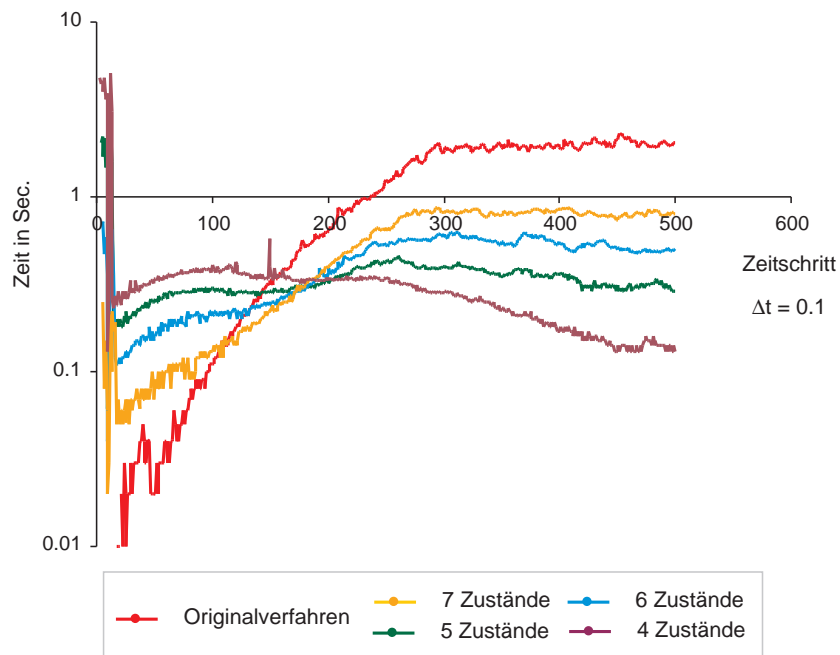


Abbildung 4.9: Analysezeit pro Zeitschritt für das Warteschlangenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren (bei unterschiedlicher Anzahl relevanter Zustände)

Zuständen.

Die Abbildung 4.10 zeigt für die vier verschiedenen Fälle die akkumulierte Analysezeit im Vergleich zum Originalverfahren. Erkennbar ist in der Abbildung, dass je weniger Zustände relevant sind, desto höher ist die anfängliche Analysezeit. Dies ist, wie bereits erwähnt, durch die Analysezeit pro Zeitschritt, bedingt. Deutlich wird auch, dass bei steigender Anzahl relevanter Zustände die Analysezeit ähnlich wie beim Originalverfahren ansteigt. Im Vergleich zum Originalverfahren kann aber dennoch ein Vorteil erzielt werden.

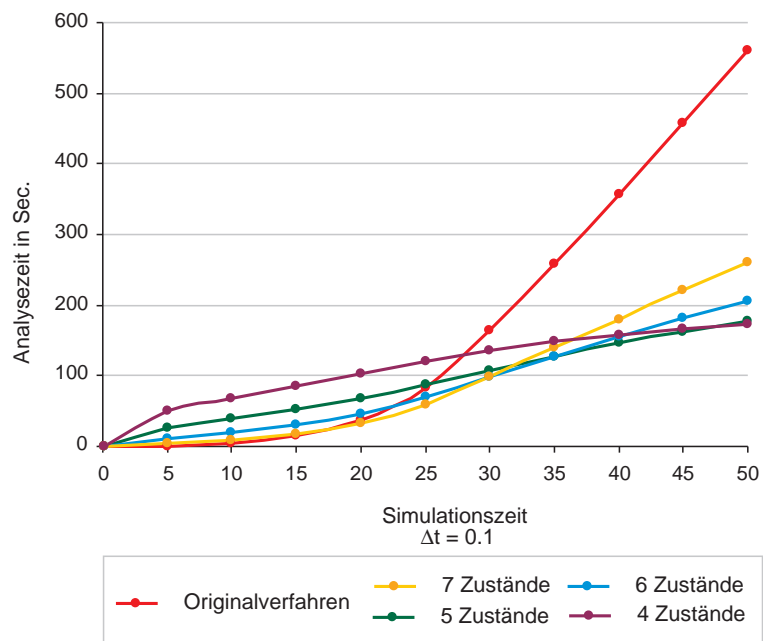


Abbildung 4.10: Analysezeit für das Warteschlangenmodell - Originalverfahren im Vergleich zum adaptierten Verfahren (bei unterschiedlicher Anzahl relevanter Zustände)



## 4.5 Experiment 4: Komplett verbundenes Modell

In diesem Experiment wird ein Modell mit neun diskreten Zuständen untersucht, bei dem jeder der Zustände durch jeden erreicht werden kann. Die Abbildung 4.11 zeigt den Zustandsgraphen für dieses Modell.

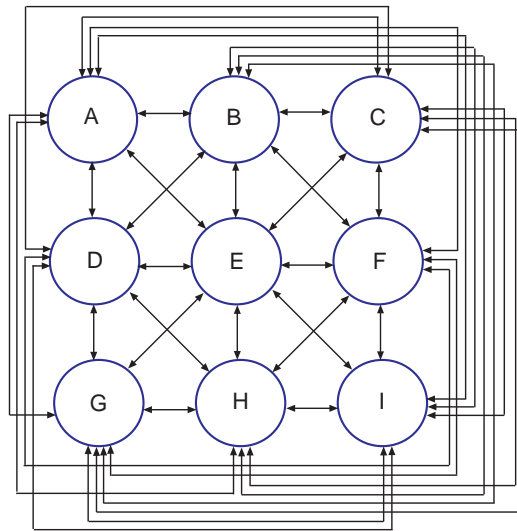


Abbildung 4.11: Zustandsgraph eines komplett verbundenen Modells

Das Modell wurde gewählt, um die in den vorherigen Experimenten getroffenen Aussagen zu bestätigen. Erwartet wird, dass sich mit dem adaptierten Verfahren kein Vorteil im Vergleich zum Originalverfahren erzielen lässt, da hier viele Masken mit vielen Einträgen erzeugt werden müssten. Aus einem Proxel können neun Folgeproxel entstehen. Wählt man einen relevanten Zustand, so müssen im ersten Zeitschritt acht irrelevante Pfade durchsucht werden, was einen erheblichen Aufwand zur Folge hat. Der Originalalgorithmus kann im Verlauf der Analyse hier sehr viele Proxel zusammenfassen, die beim adaptierten Verfahren durch die einzelne Betrachtung von entstehenden I-Proxel nicht zusammengefasst werden.

Um dies zu testen, wurde die Analyse bis  $T_{max} = 30$  durchgeführt, wobei eine diskrete Schrittweite von  $\Delta t = 0.02$  verwendet wurde. Alle Transitionen wurden mit der Weibullverteilung mit  $\alpha = 2.0$  und  $\beta = 2.0$  parametrisiert.

Die Abbildung 4.12 zeigt die akkumulierte Analysezeit (in logarithmischer Skalierung) für das adaptierte Verfahren, wenn nur der diskrete Zustand C (grüner Graph) relevant ist, im Vergleich zum Originalverfahren (roter Graph). Erkennbar ist, dass sich bis zur

Simulationszeit 30, wie erwartet, kein Geschwindigkeitsvorteil für das angepasste Verfahren erzielen lässt.

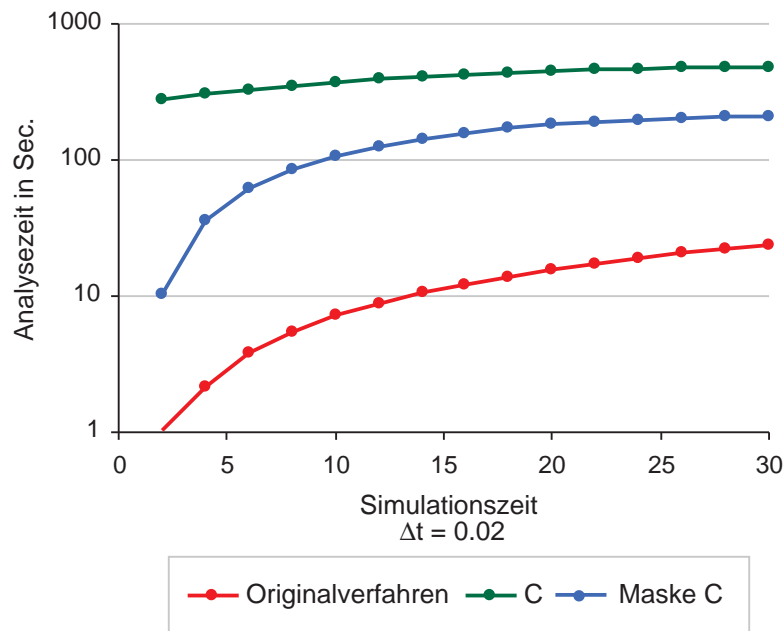


Abbildung 4.12: Analysezeit für das komplett verbundene Modell - Originalverfahren im Vergleich zum adaptierten Verfahren (bei relevantem Zustand C und vorheriger Erstellung der Maske für C)

Weiterhin wurde getestet, wie sich der Aufwand ändert, wenn die Masken bereits vor der Analyse vorliegen. Die akkumulierte Analysezeit, wenn der Zustand C relevant ist und die Masken bereits vor der Analyse vorhanden sind, zeigt der blaue Graph in der Abbildung 4.12. Erkennbar ist hier, dass der Analyseaufwand zwar geringer als ohne die vorherige Erstellung der Masken ist, aber dennoch höher als für das Originalverfahren ist. Das liegt daran, dass die Masken zum einen sehr viele Einträge enthalten und zum anderen in jedem Zeitschritt für jeden entstehenden I-Proxel nach der entsprechenden Maske gesucht werden muss.

## 4.6 Schlussfolgerungen aus den Experimenten

Aus den durchgeführten Experimenten können die folgenden Schlussfolgerungen gezogen werden. Bei der Proxel-basierten Simulation mit dem adaptierten Verfahren lassen sich Effizienzvorteile erzielen. Dabei ist zu Beginn einer Proxel-basierten Analyse durch das neue Verfahren die Analysezeit viel höher als beim Originalverfahren. Dies ist durch

die Suche nach relevanten Nachfolgern eines irrelevanten Zustandes bedingt. Der Aufwand ist dabei umso höher, je mehr I-Proxel aus einem R-Proxel entstehen können, das heißt je mehr Masken erstellt werden müssen.

Die Experimente legen den Schluss nahe, dass sich insbesondere bei längeren Simulationszeiten dadurch Effizienzvorteile im Vergleich zum Originalalgorithmus erzielen lassen. Dieser erzielte Vorteil ist von der Anzahl und der Größe der erzeugten Masken und damit von den möglichen Kombinationen im Alterungsvektor der I-Proxel abhängig.

Besonders vorteilhaft erweisen sich auch beim angepassten Verfahren Verteilungsfunktionen mit beschränktem Träger, wie die Deterministische oder die Uniform Verteilung. Diese verringern die Größe der Masken und die Anzahl der Suchdurchläufe.

Besonders große Vorteile im Vergleich zum Originalverfahren lassen sich erzielen, wenn nur eine Maske erstellt werden muss, diese wenige Einträge enthält und immer wieder benutzt werden kann.

# 5 Zusammenfassung und Bewertung

## 5.1 Zusammenfassung

Im ersten Kapitel wurde zunächst dargelegt, welche Beweggründe zur Aufgabenstellung und Motivation dieser Arbeit führten. Das zweite Kapitel erarbeitete die Grundlagen, die für die weitere Arbeit benötigt wurden. Dabei wurde aus stochastischen Petrinetzen [3], wie sie am Lehrstuhl für Simulation der Otto-von-Guericke-Universität Magdeburg verwendet werden, und stochastischen Reward-Netzen [2] die veränderte Modellspezifikation der adaptierten stochastischen Reward-Netze entwickelt. Diese veränderte Modellspezifikation diente als Grundlage für die Anpassungen am Proxelverfahren.

Im dritten Kapitel wurden dann Anpassungen am Proxelalgorithmus vorgenommen. Dazu wurden zunächst die Anforderungen, die an das adaptierte Verfahren gestellt wurden, beschrieben. Anschließend wurde das Maß zur Klassifizierung von relevanten und nicht-relevanten Zuständen definiert und gezeigt, dass sich dadurch auch Proxel in zwei Klassen einteilen lassen. Dann wurde die Idee für ein angepasstes Verfahren präsentiert, das während der Analyse nur für eine bestimmte Fragestellung relevante Proxel verarbeitet. Da irrelevante Proxel während einer Analyse berücksichtigt werden müssen, um korrekte Wahrscheinlichkeiten zu bestimmen, wurde ein Zwischenschritt implementiert, der auf einem Maskenkonzept beruht. Anhand eines kleinen Beispiels wurde anschließend die Funktionsweise der vorgenommenen Implementierung demonstriert.

In Kapitel vier wurde das Laufzeitverhalten des angepassten Verfahrens im Vergleich zum Originalverfahren evaluiert. Experiment eins zeigte, dass bei einem relevanten Modellzustand die Analysezeit pro Zeitschritt zu Beginn einer Analyse höher ist, dann aber mit zunehmender Simulationszeit sinkt. Dies war durch die Suche und Erstellung der Masken bedingt, die einen Großteil der Gesamtlaufzeit einnahm. Beim zweiten getesteten Modell konnte diese Aussage, auch bei der Wahl von anderen relevanten Zuständen bestätigt werden. Ein besonders großer Effizienzvorteil ließ sich im dritten Experiment, einem Warteschlangenmodell, erzielen. Hier war auffällig, dass nach der Erstellung einer Maske kaum Rechenaufwand nötig war und damit die Gesamtanalysezeit im Vergleich

zum Originalverfahren sehr viel geringer ausfiel. Es zeigte sich allerdings auch, dass mit zunehmender Anzahl relevanter Zustände, ein ähnliches Laufzeitverhalten auftritt, wie beim Originalverfahren.

Im vierten Experiment zeigte sich, dass bei einem Modell, bei dem jeder Zustand durch jeden erreicht werden kann, der Analyseaufwand durch das neue Verfahren sehr viel höher als für das Originalverfahren ist. Da für jeden irrelevanten Folgezustand eine Suche nötig ist und die Masken dabei sehr groß werden, konnte bei diesem Modell, wie erwartet, kein Vorteil erzielt werden. Das lag auch daran, dass das Originalverfahren während der Analyse nahezu alle Proxel zusammenfassen konnte.

Zusammenfassend lässt sich für die Experimente festhalten, dass sich Effizienzvorteile erzielen lassen, wenn nur für eine Fragestellung relevante Zustände analysiert werden. Der maximal erreichbare Vorteil ist aber nicht nur von der Struktur des Modells und den gewählten Verteilungen abhängig ist, sondern auch von der Wahl der für die Fragestellung relevanten Zustände.

## 5.2 Ausblick

In dieser Arbeit wurde ein Proxelverfahren für adaptierte stochastische Reward-Netze entwickelt, das während der Analyse nur reward-relevante Zustände verarbeitet und irrelevante Zustände „überspringt“. Im Folgenden werden einige Punkte genannt, die näher bei Fortführung dieser Arbeit näher untersucht werden sollten:

Da mit der erstellten Implementierung bisher nur kleine Modelle und auch nur das Laufzeitverhalten evaluiert werden konnten, sind weitere Untersuchungen notwendig. Insbesondere sollte dabei getestet werden, ob sich wirklich immer die gleiche Güte der Ergebnisse erzielen lässt, wie mit dem Originalverfahren.

Weiterhin sollte vor allem auch untersucht werden, welchen Einfluss der gewählte Schwellenwert auf den Abbruch der Suche nach relevanten Nachfolgern hat.

Es wurde bereits angemerkt, dass die Implementierung der Masken einer Verbesserung bedarf. Eine Möglichkeit wäre auch hier gleiche Proxel, das heißt Proxel mit gleichen Alterungsvariablen und gleichem Zustand, zusammenzufassen.

Weiterhin besteht Bedarf nach einer optimalen Datenstruktur, in der mit minimalem Aufwand geprüft werden kann, ob eine Maske für ein bereits untersuchtes I-Proxel vorhanden ist. Dadurch könnte der Suchaufwand, der bei jedem Auftreten eines I-Proxels notwendig ist, verringert werden.

### 5.3 Bewertung

Im Rückblick auf die in Kapitel eins formulierte Aufgabenstellung lässt sich festhalten, dass diese vollständig bearbeitet wurde. Die im selben Kapitel gesetzten Ziele wurden ebenfalls alle erreicht.

Durch die vorgenommenen Anpassungen am Proxelalgorithmus werden jetzt nur noch Proxel verarbeitet, die relevant für eine bestimmte Fragestellung sind. Gleichzeitig wurde dadurch erreicht, dass einige Modelle viel effizienter analysiert werden können. Die mit dieser Arbeit gewonnen Erkenntnisse stellen daher einen wichtigen Beitrag zur weiteren Erforschung des Proxelverfahrens am Lehrstuhl für Simulation dar.

Zurückblickend auf die Erwartungen, die im ersten Kapitel beschrieben wurden, hat sich bestätigt, dass weitere Datenstrukturen benötigt werden. Die größte Schwierigkeit bei der Implementierung bereitete dabei der Entwurf von Datenstrukturen zur Suche und zur Verwaltung der Zustände. Nachträglich gilt es dabei festzuhalten, dass die Implementierung der Masken als eine Liste, wie in den Experimenten bereits angemerkt, einer Verbesserung bedarf. Dennoch übertrafen die erzielten Geschwindigkeitsvorteile für einige getestete Modelle, insbesondere bei der Verwendung von Verteilungsfunktionen mit beschränktem Träger, die persönlichen Erwartungen sogar.

Im Rahmen dieser Arbeit wurde ein Proxelverfahren geschaffen, das einige Modelle viel effizienter mit vergleichbarer Güte der Ergebnisse analysieren kann. Dadurch, dass nicht mehr der gesamte erreichbare Zustandsraum betrachtet werden muss, wird das Problem der Zustandsraumexplosion eingedämmt. Dies stellt einen wertvollen Schritt auf dem Weg zu einem Proxelalgorithmus, mit dem auch größere Modelle, die bisher nur durch ereignisbasierte Algorithmen effizient analysiert werden können, dar.

# Literaturverzeichnis

- [1] B. Baumgarten. *Petri Netze: Grundlagen und Anwendungen*. B.I. Wissensverlag, 1990.
- [2] G. Ciardo, A. Blakemore, P. F. Chimento, J. K. Muppala, and K. S. Trivedi. *Automated Generation and Analysis of Markov Reward Models using Stochastic Reward Nets*. In C. Meyer und R.J. Plemmons (Herausgeber): *Linear Algebra, Markov Chains and Queuing Models*, Band 48 der Reihe *IMA Volumes in Mathematics and its Applications*. Springer-Verlag, 1993.
- [3] G. Horton. Introduction to Simulation Vorlesungsmaterialien. Fakultät für Informatik, Lehrstuhl für Simulation. Otto-von-Guericke Universität Magdeburg - <http://www.sim-md.de/its>.
- [4] G. Horton. A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times. In *Proceedings of the European Simulation Symposium 2002*, Dresden: SCS European Publishing House, Oktober 2002.
- [5] A. M. Law and D. M. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 1999.
- [6] S. Lazarova-Molnar. *The Proxel-Based Method: Formalisation, Analysis and Applications*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2005.
- [7] S. Lazarova-Molnar and G. Horton. An Experimental Study of the Behaviour of the Proxel-Based Simulation Algorithm. In *Simulation und Visualisierung 2003*, SCS European Publishing House, 2003.
- [8] S. Lazarova-Molnar and G. Horton. Proxel-Based Simulation of Stochastic Petri Nets Containing Immediate Transitions. In *On-Side Proceedings of the Satelite Workshop of ICALP 2003 in Eindhoven, Netherlands*, Forschungsbericht Universität Dortmund, 2003.
- [9] S. Lazarova-Molnar and G. Horton. Proxel-Based Simulation of Stochastic Petri Nets. In *Simulation und Visualisierung 2004*, SCS European Publishing House, 2004.

- [10] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley, Chichester, 1995.
- [11] W. Reisig. *Petri Nets: An Introduction*. Springer Verlag, 1985.
- [12] F. Wickborn, G. Horton, S. Heller, and F. Engelhardt. A General-Purpose Proxel Simulator for an Industrial Software Tool. In *18. Symposium Simulationstechnik (ASIM 2005)*, Erlangen: SCS European Publishing House, 2005.



