Università Cà Foscari di Venezia

DIPARTIMENTO DI SCIENZE AMBIENTALI, INFORMATICA E STATISTICA

Corso di Laurea Magistrale in Informatica

Simulazione di reti di Petri stocastiche in forma prodotto: applicazione e criteri di arresto

Relatrice	prof. Simonetta Balsamo
Correlatore	prof. Andrea Marin

A. A. 2012/2013

Panzetta Pasquale 800405 ppanzett@dsi.unive.it

Abstract

La valutazione e l'analisi delle prestazioni di sistemi basata su modelli analitici e di simulazione trova ampia applicabilità per lo studio del comportamento di sistemi di elaborazione e di comunicazione. I modelli analitici possono essere risolti con metodi esatti e la classe di modelli in forma prodotto può permettere l'applicazione di algoritmi efficienti, sotto particolari ipotesi e condizioni, mentre i modelli di simulazione sono caratterizzati da una più ampia generabilità di applicazione, anche se il costo può essere rilevante. Nel corso della valutazione di un sistema mediante un modello simulativo si è spesso interessati al comportamento a regime del sistema stesso, ovvero all'analisi dello stato stazionario e questo può comportare che per stimare una misura di prestazione possono essere necessarie prove molto lunghe della simulazione. A tal scopo, a partire da un modello di simulazione implementato per lo studio di reti di Petri in forma prodotto, l'elaborato dedica particolare attenzione al criterio di arresto dei singoli esperimenti di simulazione. In particolare, proponiamo due differenti metodi per interrompere la simulazione al termine di un valore temporale sufficientemente grande da garantire una buona accuratezza della stima della probabilità di stato stazionaria relativa ad un insieme finito di marcature prese in esame. Tale insieme di marcature è generato da un particolare algoritmo che tramite l'esecuzione di un run pilota, individua le marcature della rete che si suppone vengano raggiunte con una maggiore frequenza nel corso della simulazione. Inoltre, l'elaborato propone un confronto con il simulatore TimeNET al fine di effettuare una comparazione relativa alle metodologie utilizzate per la terminazione degli esperimenti di simulazione.

Indice

I Introduzione **8**

II Formalismi e definizioni **16**

1 Introduzione alle reti di Petri **17**
1.1 Definizione 17
1.2 Componente statica 19
1.3 Regole di Enablig e Firing 21
1.4 Comportamento dinamico 23
1.4.1 Concorrenza 23
1.4.2 Dipendenza casuale 23
1.4.3 Conflitti 24
1.5 Proprietà 25
1.5.1 Reachability e Reversibily 25
1.5.2 Liveness 28
1.5.3 Boundedness 29
1.5.4 Coverability 29
1.5.5 Mutua esclusione 29
1.6 Invarianti 30
1.6.1 T-invariante 30
1.6.2 P-invariante 31

2 Reti di Petri stocastiche **33**
2.1 Il tempo nelle reti di Petri 33
2.1.1 Rete di Petri con transizioni temporizzate 34
2.1.2 Comportamento, politiche di Firing e conflitti. 35
2.2 Reti SPN 37
2.2.1 Tempo di soggiorno e probabilità di firing 40
2.2.2 Esempi 40
2.3 Indici di prestazione 43
2.3.1 Probabilità di una particolare condizione. 44
2.3.2 Numero medio di token 44
2.3.3 Utilizzazione 45

2.3.4 Throughtput 46
2.3.5 Tempo medio di risposta 46
2.4 GSPN 46
2.4.1 Analisi GSPN 49
2.4.2 GSPN-SPN in forma prodotto 49
2.5 Teorema di Coleman, Henderson et al. (Forma Prodotto) . . 50

III Simulazione 61

3 Criteri di Arresto di simulazione 62
3.1 Introduzione 62
3.2 Primo Criterio: Metodo della matrice 63
3.2.1 Formalizzazione del criterio. 63
3.2.2 Classe Test Matrix 65
3.2.3 Algoritmo per il ritrovamento delle marcature maggiormente probabili 73
3.2.4 Convalida, esecuzione ed analisi dei dati 76
3.2.5 Complessità e miglioramenti 78
3.2.6 Convalida ed analisi dei risultati relative a SPN chiuse. 78
3.2.6.1 Primo esempio 79
3.2.6.2 Secondo esempio 83
3.2.7 Convalida ed analisi dei risultati relative a SPN aperte. 86
3.2.7.1 Primo esempio 88
3.2.7.2 Secondo esempio 90
3.2.8 Precisione 92
3.3 Secondo Criterio: Metodo del vettore 98
3.3.1 Formalizzazione del criterio. 98
3.3.2 Classe TestVector 99
3.3.3 Complessità e miglioramenti 103
3.3.4 Convalida ed analisi dei risultati relative a SPN chiuse. 104
3.3.4.1 Primo esempio 104
3.3.4.2 Secondo esempio 109
3.3.5 Convalida ed analisi dei risultati relative a SPN aperte. 112
3.3.5.1 Primo esempio 112
3.3.5.2 Secondo esempio 114
3.3.6 Precisione 116
3.4 Confronto tra i due criteri 121
3.4.1 Esempio 122

4 Confronto con TimeNET 125
4.1 Introduzione di TimeNET 125
4.2 Interfaccia 127
4.2.1 Validate 130

4.2.2 Evaluation 131
4.2.2.1 Stationary Analysis 132
4.2.2.2 Stationary Simulation Standard 134
4.2.2.3 Stationary Simulation RESTART 136
4.2.2.4 Transient Analysis 137
4.2.2.5 Transient Simulation 138
4.3 Valutazione dei risultati 139
4.4 Confronto 141
4.4.1 Classi di reti supportate 141
4.4.2 Modalità di arresto della simulazione 142
4.4.3 Funzionalità 143
4.4.4 Stima degli indici di prestazione 143
4.4.4.1 Throughtput 144
4.4.4.2 Numero medio di token 144
4.4.4.3 Utilizzazione 144
4.4.4.4 Probabilità di stato 145
4.4.5 Risultati di simulazione 145
4.4.5.1 Specifiche hardware 146
4.4.5.2 Esperimento 146

5 Conclusioni 152

Elenco delle figure

1.1 Componenti grafiche 18
1.2 Esempio PN 19
1.3 Esempio PN per la costruzione delle matrici I, O e C 20
1.4 Esempio PN utile per l'analisi di una sequenza di firing 22
1.5 Rete di Petri con transizioni non concorrenti 23
1.6 Rete di Petri con transizioni in dipendenza casuale 24
1.7 Due tipologie di reti di Petri. (a) PN con transizioni non in conflitto effettivo. (b) PN con transizioni in conflitto effettivo 24
1.8 Rete di Petri utilizzata per la costruzione del grafo di raggiungibilità 27
1.9 Grafo di raggiungibilità della rete in Figura 1.8 27
1.10 Reti di Petri caratterizzate da proprietà di boundedness, liveness e reversibily 30
1.11 Esempio PN per lo studio del vettore T-invariante. 31
1.12 Rete di Petri per il calcolo del vettore P-invariante 32

2.1 Comportamento di una TTPN 35
2.2 Politica race a fronte di due transizioni in conflitto 37
2.3 Esempio di SPN con transizioni a cui sono associati tassi di firing indipendenti dalla marcatura 40
2.4 (a) Grafo di raggiungibilità e (b) CTMC della rete in Figura 2.3 41
2.5 (a) Grafo di raggiungibilità e (b) CTMC della rete con tasso di firing dipendente dalla marcatura 42
2.6 Primo esempio di SPN esprimibile in forma prodotto 53
2.7 Rete di Petri stocastica con due possibili firing della transizione t_1 57
2.8 Rete di petri stocastica con vincolo sul rango delle matrici . 59

3.1 Classe TestMatrix 66
3.2 Algoritmo per il ritrovamento delle marcature maggiormente probabili 75
3.3 Output di simulazione 81

3.4 Resoconto prove ripetute 82
3.5 Corrispondenza modello $M/M/1$ e rete di Petri 87
3.6 SPN aperta con grafo di raggiungibilità infinito 88
3.7 Andamento della probabilità stimata relativa alla marcatura [0 4] 93
3.8 Andamento della probabilità stimata relativa alla marcatura [1 3] 94
3.9 Andamento della probabilità stimata relativa alla marcatura [3 1] 94
3.10 Andamento della probabilità stimata relativa alla marcatura [2 2] 95
3.11 Classe TestVector 100
3.12 Output simulazione 106
3.13 Resoconto prove ripetute 108
3.14 Andamento della probabilità stimata relativa alla marcatura [0 4] 117
3.15 Andamento della probabilità stimata relativa alla marcatura [1 3] 117
3.16 Andamento della probabilità stimata relativa alla marcatura [3 1] 118
3.17 Andamento della probabilità stimata relativa alla marcatura [2 2] 118
3.18 Andamento della probabilità stimata relativa alla marcatura [4 0] 119

4.1 Interfaccia TimeNET 128
4.2 Toolbar superiore 129
4.3 Toolbar inferiore 130
4.4 Menù Evaluation 132
4.5 Stationary Analysis 133
4.6 Stationary Simulation 135
4.7 Stationary Simulation RESTART 137
4.8 Transient Analysis 138
4.9 Transient Simulation 139

Parte I

Introduzione

Ambito ed inquadramento generale

Un modello di un sistema reale è definito come un'astrazione in grado di rappresentare particolari aspetti della sua struttura e del suo comportamento. In generale, un modello può essere:

- Analitico: le componenti sono rappresentate da variabili e parametri e le interazioni fra quest'ultime sono descritte mediante relazioni fra le quantità definite. La valutazione del sistema effettuata con un modello analitico necessita del calcolo della sua soluzione tramite metodi analitici o soluzioni numeriche.

- Simulativo: riproduce il comportamento dinamico del sistema nel tempo rappresentando le componenti e le interazioni in termini di relazioni funzionali. La valutazione di un sistema tramite un modello di simulazione richiede l'esecuzione di un programma di simulazione o di un simulatore che rappresenta l'evoluzione temporale del sistema e su cui si effettuano delle misure per stimare le grandezze di interesse.

Le tecniche di modellazione attualmente utilizzate sono molteplici ed in particolare si distinguono i reliability graph, le stochastic reward net, i reliability block diagram, i fault tree, le catene di Markov a tempo continuo o discreto, le reti di code ed infine le reti di Petri con tutte le loro estensioni (generalizzate, colorate, stocastiche, etc.).

Fra le tecniche di modellazione, che vengono particolarmente analizzate dai ricercatori per la loro potenza espressiva e su cui sono stati prodotti molti tool, emergono le reti di Petri con le loro estensioni.

Le reti di Petri rappresentano uno strumento per la modellizzazione di processi ed in particolare per la rappresentazione delle interazioni fra processi paralleli. Spesso, nella descrizione di un processo si ha la necessità di rappresentare attività che possono essere eseguite contemporaneamente, in parallelo, ma non indipendentemente l'una dall'altra. Potrebbe accadere infatti che una particolare fase del processo non si verifichi o non venga attivata finché altre fasi non vengono concluse o si verificano particolari condizioni. Situazioni di questo tipo, per esempio, non possono essere descritte mediante un diagramma di flusso come quelli utilizzati per la descrizione di un algoritmo nell'ambito della programmazione strutturata. In un diagramma di flusso le attività sono serializzate dato che due rami distinti non vengono mai percorsi contemporaneamente, ma rappresentano due cammini alternativi scelti sulla base di un criterio deterministico dipendente dall'esito della valutazione di una espressione booleana. L'utilizzo delle reti di Petri permette di superare tale limite, introducendo un formalismo grafico e semantico che consente di aggiungere una ulteriore funzionalità rispetto ai diagrammi di flusso.

Le reti di Petri non solo permettono di rappresentare e di descrivere globalmente un processo, ma consentono anche di seguirne l'evoluzione permettendo di visualizzare lo stato in cui si trova la rete in un determinato istante. Per tale motivo costituiscono uno strumento modellistico efficace e versatile per la rappresentazione e l'analisi di sistemi reali complessi.

Se si intende utilizzare le reti di Petri per studiare le prestazioni di un sistema reale, è necessario includere nel modello anche il concetto del tempo introducendo in tal modo le reti di Petri temporizzate (TPN). In particolare, l'elaborato dedica maggiore attenzione alle reti di Petri stocastiche (SPN) in forma prodotto le quali definiscono la variabile temporale nello svolgimento delle attività della rete come una variabile casuale con una particolare distribuzione di probabilità. La soluzione in forma prodotto di una rete di Petri stocastica permette di definire la probabilità congiunta degli stati raggiungibili della rete come il prodotto delle probabilità marginali di ciascuna delle risorse che compongono la stessa agevolandone in tal modo anche il calcolo degli indici di prestazione.

Le prestazioni dei sistemi reali possono essere effettuate sia tramite tecniche di misurazione, sia mediante tecniche modellistiche. Fra le tecniche modellistiche citate in precedenza, tale elaborato prende in considerazione i modelli di simulazione. La simulazione è particolarmente flessibile in quanto permette di rappresentare e di studiare solo attraverso analisi di tipo statistico sistemi molto complessi dei quali si conoscono solamente alcune caratteristiche.

Gli aspetti rilevanti che fanno della simulazione uno strumento largamente utilizzato sono legati al fatto che quest'ultima permette di rappresentare sistemi reali anche complessi ed inoltre consente di riprodurre il comportamento di un sistema in riferimento a situazioni che non sono sperimentabili direttamente (eventi rari, condizioni di rischio). Inoltre, è possibile, mediante simulazione, ottenere una ripetibilità dell'esperimento di analisi.

Tuttavia, durante la fase di progettazione di un simulatore, uno dei problemi maggiori consiste nella determinazione della lunghezza della simulazione dopo che il sistema ha raggiunto uno stato di equilibrio. Infatti, si deve sempre tener presente che la simulazione non produce valori esatti delle misure di prestazione di un sistema in quanto ogni singola simulazione può essere vista come un "esperimento statistico" che genera osservazioni statistiche sulle prestazioni del sistema. Queste osservazioni sono poi utilizzate per produrre stime delle misure di prestazione ed il loro livello di precisione dipende dalla scelta della lunghezza del run di simulazione.

La scelta di una lunghezza troppo breve del run di simulazione produce delle stime di misure di prestazione imprecise in quanto tali stime dipendono dalla fase transiente della simulazione e quindi ricavate a partire da uno stato in cui il sistema risente delle condizioni iniziali. Al contrario, prove di simulazione molto lunghe producono stime di misure di prestazione ricavate dalla fase stazionaria e quindi risultano corrette e precise ma spesso, tali

prove di simulazione sono caratterizzate da tempi di esecuzione onerosi e spesso eccessivi in quanto maggiori del tempo utile necessario per ottenere stime altrettanto precise ed affidabili. Per risolvere tale problema è quindi necessario introdurre e definire dei criteri di arresto di simulazione in grado di terminare, dopo che il sistema simulato ha raggiunto uno stato di equilibrio, il run di simulazione non appena il simulatore è in grado di fornire in output stime precise delle misure di interesse.

Scopo e motivazione della tesi

Lo scopo della tesi è quello di definire un modello di simulazione denominato PFNetS che permette la rappresentazione di una generica rete di Petri stocastica in forma prodotto e lo studio dell'andamento dei principali indici di prestazione e delle probabilità stazionarie di stato del sistema reale modellato da quest'ultima.

L'obiettivo principale è quello di sfruttare le informazioni ricavate dalla forma prodotto delle reti analizzate per la definizione di due differenti criteri di arresto di simulazione, denominati rispettivamente metodo della matrice e metodo del vettore. In particolare, tali criteri si basano sul rapporto tra le probabilità stazionarie stimate da PFNetS e le rispettive probabilità di stato stazionarie analitiche non normalizzate relativamente ad un numero finito di marcature. Tali marcature sono caratterizzate dai maggiori valori di probabilità stazionaria analitica. Data la difficoltà di calcolo, abbiamo scelto di non considerare nel rapporto la costante di normalizzazione ed inoltre di prendere in esame solo un numero arbitrario di marcature maggiormente probabili per evitare problemi legati al raggiungimento di eventi rari in fase di simulazione.

La motivazione che ha portato a sviluppare l'elaborato proposto è stata quella di stabilire come le informazioni offerte dalla forma prodotto per SPN, ricavabili mediante il teorema di Coleman Henderson et al. potessero essere sfruttate per capire dapprima il comportamento della rete e successivamente per restituire stime sufficientemente accurate al termine dei run di simulazione in accordo a quanto imposto dai vincoli definiti dai criteri di arresto implementati.

A tale scopo l'elaborato presenta una definizione formale dei due criteri di arresto proposti ed un algoritmo che, tramite un run di simulazione, restituisce le marcature caratterizzate dai maggiori valori di probabilità stazionaria analitica a meno della costante di normalizzazione, offrendo in tal modo una prima analisi del comportamento della rete. Le informazioni raccolte in tale fase vengono quindi sfruttate dai metodi implementati per la definizione dei criteri di arresto.

Metodo applicato e risultati ottenuti

Per il raggiungimento degli obiettivi sopra citati, abbiamo implementato, mediante il linguaggio orientato agli oggetti Java, il simulatore PFNetS che permette la rappresentazione di una generica SPN e lo studio dei principali indici di prestazione dei sistemi reali modellati da quest'ultima. Al simulatore prodotto, quindi, sono stati integrati gli algoritmi che implementano i criteri di arresto proposti.

Per verificare il corretto funzionamento del simulatore PFNetS e stabilire il livello di precisione dei risultati offerti da quest'ultimo, l'elaborato presenta ed analizza gli esiti ottenuti dalle fasi di validazione del simulatore ed analisi dei risultati forniti in output dallo stesso, entrambe a partire dalla simulazione sia di SPN chiuse, sia di SPN aperte. I risultati restituiti in output da PFNetS e presi in esame per le fasi sopra descritte sono relativi sia ai principali indici di prestazione analizzati sia alle probabilità stazionarie di stato di un numero finito di marcature caratterizzate dai maggiori valori di probabilità stazionaria analitica. Gli esiti delle fasi di convalida ed analisi dei risultati hanno confermato il corretto funzionamento del simulatore PFNetS.

Per determinare l'attendibilità dei risultati ottenuti si presenta un confronto tra i risultati forniti dal simulatore proposto ed i rispettivi risultati offerti dal tool di simulazione TimeNET [13] a partire dalla medesima rete oggetto di studio e dalle stesse condizioni iniziali di simulazione. Sulla base di quanto ottenuto dal confronto si cerca di stabilire se l'utilizzo dei metodi che implementano i criteri di arresto di simulazione possa migliorare la simulazione di SPN sia in termini di durata, sia in termini di precisione dei risultati.

Dal confronto con TimeNET emerge che il simulatore PFNetS fornisce valori di stima, di un particolare indice di prestazione, caratterizzati da ottimi livelli di precisione e nella maggior parte dei casi migliori rispetto a quelli ottenuti da TimeNET. Inoltre, sebbene le durate delle simulazioni, effettuate mediante lo strumento proposto, siano superiori, a parità di condizioni iniziali, a quelle prodotte da TimeNET, possiamo ritenerci soddisfatti dei risultati ottenuti in quanto, a nostro avviso, il simulatore PFNetS è funzionante, corretto e preciso nei risultati.

Inoltre, a partire da una rete di Petri stocastica che modella un sistema reale è stato effettuato uno studio sui risultati forniti da PFNetS, i quali sono stati interpretati in relazione al significato associato ai posti e alle transizioni che compongono la rete. In particolare, il sistema preso in esame descrive l'esecuzione di due differenti tipologie di attività quali operazioni batch ed operazioni interattive, entrambe eseguite successivamente all'allocazione di CPU e di processi clienti.

Problemi aperti e sviluppi futuri

La limitazione principale nell'utilizzo del simulatore PFNetS è legata principalmente al numero limitato di metodi che consentono di calcolare la forma prodotto di reti di Petri stocastiche. Infatti, sebbene il modello simulativo permetta la rappresentazione e l'analisi di una qualsiasi SPN, l'impossibilità di ottenere una forma prodotto per alcune di queste, non consente l'utilizzo dei metodi per l'arresto delle simulazione, ottenendo in tal modo analisi caratterizzare da simulazioni aventi tutte la medesima lunghezza. Possiamo quindi affermare che studi futuri che coinvolgono il ritrovamento della forma prodotto delle SPN potrebbero aumentare l'applicabilità del simulatore proposto.

Inoltre, essendo il simulatore PFNetS testato a partire da un numero limitato di modelli (circa pari a 20) vi è la possibilità che non siano emersi tutti i limiti riconducibili agli algoritmi implementati per la definizione dello stesso. Tuttavia, a partire da un'analisi relativa al costo computazionale e alle strutture dati utilizzate per la definizione del simulatore PFNetS, l'elaborato illustra i possibili miglioramenti da introdurre nell'implementazione degli algoritmi proposti al fine di garantire una maggiore affidabilità dello strumento.

Paragonando il simulatore PFNetS con i principali tool esistenti per l'analisi delle reti di Petri è facile intuire come quest'ultimo sia privo delle funzionalità che permettono di effettuare studi strutturali delle reti oggetto di analisi. A nostro avviso un interessante sviluppo di PFNetS può essere quello di implementare algoritmi per la generazione del grafo di raggiungibilità sfruttando l'algoritmo per il ritrovamento delle marcature con i più alti valori di probabilità analitica a meno della costante di normalizzazione. Infatti, dato che tale algoritmo prevede una visita caratterizzata da una successione di passi casuali dello stato della rete, è possibile sfruttare le informazioni (marcature raggiunte dalla rete) ricavate durante tale operazione per la costruzione del grafo di raggiungibilità.

Inoltre, sebbene l'utilizzo del simulatore presentato nell'elaborato sia limitato alla simulazione di SPN in forma prodotto, le metodologie di terminazione della simulazione ed in particolare le idee alla base dei criteri di arresto, possono, a nostro avviso, essere sfruttate o applicate in futuro per l'implementazione di modelli simulativi che studiano ulteriori tipologie di reti in forma prodotto (es. reti di code, GSPN) di cui si è interessati a studiarne il comportamento.

Struttura della tesi

La tesi illustra parte del lavoro descritto in precedenza e svolto con il collega Massimo Mariuzzo. Nel dettaglio, dedica particolare attenzione alla

definizione ed all'analisi dettagliata dei criteri di arresto sopra citati. Inoltre, presenta un confronto tra i risultati ottenuti dal simulatore PFNetS ed i rispettivi risultati forniti dal tool di simulazione TimeNET a partire dalla medesima rete oggetto di studio e dalle stesse condizioni iniziali di simulazione. Sulla base di quanto ottenuto dal confronto si stabilisce se l'utilizzo dei metodi che implementano i criteri di arresto di simulazione è in grado di migliorare la simulazione di SPN sia in termini di durata, sia in termini di costo computazionale.

La tesi è suddivisa in due parti. La prima parte, costituita dai Capitoli 1 e 2, introduce i formalismi e le definizioni riguardanti le differenti tipologie di reti di Petri. La seconda parte, articolata nei Capitoli 3 e 4, illustra i due differenti criteri di arresto implementati per la terminazione dei run di simulazione e presenta l'analisi dei risultati forniti in output dal simulatore PFNetS. Inoltre, propone un confronto tra i risultati ottenuti dal simulatore proposto ed i rispettivi risultati forniti da TimeNET.

Il Capitolo 1 presenta un'introduzione relativa alle reti di Petri descrivendone la componente statica, il comportamento dinamico e le principali proprietà. Nel dettaglio, riportiamo alcuni esempi per illustrare le regole di firing (scatto) ed enabling (abilitazione) che costituiscono la base del funzionamento di tale tipologia di rete.

Il Capitolo 2 presenta le reti di Petri stocastiche (SPN) e le reti di Petri stocastiche generalizzate (GSPN). Inizialmente introduciamo il concetto del tempo nelle reti di Petri e successivamente analizziamo nel dettaglio le SPN e le GSPN, descrivendone il funzionamento, le principali proprietà e la relazione che intercorre con le CTMC (Catene di Markov a tempo continuo). Si dedica maggiore attenzione all'analisi della distribuzione stazionaria delle probabilità di stato ed al calcolo degli indici di prestazione. Introduciamo il teorema di Coleman, Henderson et al. utilizzato per l'analisi delle reti reti in forma prodotto.

Il Capitolo 3 presenta due differenti criteri di arresto della simulazione implementati entrambi mediante due differenti algoritmi, chiamati metodo della matrice e metodo del vettore. Tali algoritmi permettono di terminare la simulazione della SPN analizzata dopo un numero finito di passi di simulazione, garantendo in tal modo un limitato costo computazionale relativo alla fase di analisi. Inoltre, il capitolo propone un algoritmo per il ritrovamento delle marcature caratterizzate dai maggiori valori di probabilità analitica non normalizzata. La parte conclusiva del capitolo prevede la definizione dei metodi utilizzati per le stime ed il calcolo degli intervalli di confidenza basati sulle singole probabilità di stato stazionarie stimate da PFNetS ed illustra la fase di convalida ed analisi dei risultati forniti da quest'ultimo a partire da differenti casi di studio.

Il Capitolo 4 presenta un confronto tra il simulatore PFNetS e lo strumento di analisi TimeNET. In particolare, il confronto è effettuato a partire

da differenti aspetti quali le tipologie di reti analizzabili, le tecniche utilizzate per limitare la lunghezza del run di simulazione, le funzionalità offerte ed infine le modalità utilizzate per la stima degli indici di prestazione. Il confronto termina con la presentazione dei risultati di simulazione restituiti dai differenti simulatori a partire dagli stessi modelli di rete analizzati.

Il Capitolo 5 presenta la parte conclusiva dell'elaborato in cui sono raccolte considerazioni personali in merito al lavoro effettuato per la tesi.

Parte II

Formalismi e definizioni

Capitolo 1

Introduzione alle reti di Petri

1.1 Definizione

Le reti di Petri sono dei modelli astratti sviluppati alla ricerca di metodi potenti semplici e naturali per la descrizione e l'analisi dei flussi di informazione e di controllo dei sistemi. Sono particolarmente indicate per sistemi che prevedono attvità concorrenti, sincronizzate, conflittuali ed inoltre permettono la descrizione modulare di sistemi complessi considerando solo stati locali e i loro cambiamenti [1].

Le reti di Petri sono state introdotte da Carl Adam Petri nel 1962 e da allora ne sono state definite molteplici varianti in grado di adattarsi a specifici obiettivi modellistici.

Una rete di Petri è un grafo bipartito orientato e pesato che consiste in due tipologie di nodi, posti e transizioni. Gli archi collegano posti a transizioni e viceversa. Nodi della stessa tipologia non possono essere collegati (posto con posto o transizione con transizione).

Lo stato della rete è descritto da una marcatura, la quale è una funzione $M : P \to \mathbb{N}$ che assegna ad ogni posto un numero intero non negativo di token, dove P indica l'insieme dei posti della rete. Se la marcatura assegna al posto p un intero k, il posto p è contrassegnato da k token.

Una marcatura è denotata da M, vettore di dimensione m, dove m indica il numero totale di posti della rete. La i-esima componente della marcatura M, rappresentata con $M(i)$, indica il numero di token presenti nel posto i, con $i \in P$.

In una rappresentazione grafica, i posti sono raffigurati mediante forme circolari, le transizioni sono rappresentate da segmenti o rettangoli, mentre gli archi sono rappresentati da frecce e contrassegnati dal loro peso (numero intero). Un arco k-pesato può essere interpretato come un insieme di k archi paralleli. Nel caso in cui il peso dell'arco sia unitario l'indicazione del peso viene omessa. I token associati ai posti vengono rappresentati con k punti neri nel posto p.

In Figura 1.1 vengono riassunte le componenti grafiche utilizzate per la rappresentazione di una rete di Petri.

Figura 1.1: Componenti grafiche

La Tabella 1.1 illustra le differenti interpretazioni che possono essere associate rispettivamente a posti e transizioni in una rete di Petri [1].

Input Place	Transition	Output Place
Precondizioni	Evento	Postcondizione
Risorse necessarie	Task	Risorse rilasciate
Dati di input	Passo computazionale	Dati in output

Tabella 1.1: Differenti interpretazioni

Come si può osservare dalla Tabella 1.1, i posti in input / output e le transizioni che compongono la rete possono assumere differenti significati. Per esempio, in un modello che rappresenta un sistema reale che descrive una catena di montaggio i posti rappresentano le risorse necessarie per l'avanzamento della produzione e le transizioni identificano particolari fasi di lavorazione. Ogni fase necessità di un numero di risorse minimo per essere effettuata e per produrre una nuova quantità di risorse in output.

Inoltre, le reti di Petri possono essere utilizzate per modellare l'esecuzione di un generico algoritmo. In tal caso i posti rappresentano i dati in input necessari per effetture i passi computazionali dell'algoritmo, descritti dalle transizioni della rete. L'esecuzione dei passi computazionali restituisce in output nuovi dati modellati dai posti in output alla transizione.

Segue una definizione formale delle reti di Petri.

Definizione 1. (Reti di Petri). Una rete di Petri è definita [1] dalla seguente 5-upla: $PN = (P, T, F, W, M_0)$ dove:

$P = \{p_1, p_2, \ldots, p_n\}$ rappresenta un insieme finito di posti.

$T = \{t_1, t_2, \ldots, t_m\}$ rappresenta un insieme finito di transizioni.

$F \subseteq (P \times T) \cup (T \times P)$ rappresenta un insieme di archi.

$W : F \to \mathbb{N}$ rappresenta una funzione che associa un determinato peso ad un arco.

$M_0 : P \to \mathbb{N}$ rappresenta la marcatura iniziale.
$P \cap T = \emptyset \wedge P \cup T \neq \emptyset$.

(N, M_0) definisce una rete di Petri con marcatura iniziale M_0. Mentre, PN denota una rete di Petri priva di marcatura iniziale M_0.

Esempio 2. In Figura 1.2 è rappresentata una rete di Petri $PN = (P, T, F, W, M_0)$ con $P = \{p_1, p_2, p_3\}$, $T = \{t_1\}$, $F = \{(p_1, t_1)\,(p_2, t_1)\,(t_1, p_3)\}$ e $M_0 = [2\,2\,0]$

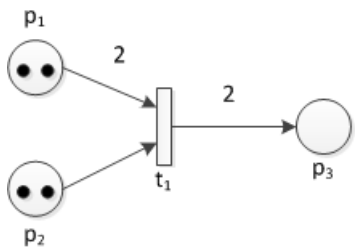

Figura 1.2: Esempio PN

1.2 Componente statica

La componente statica di una rete di Petri è composta dal grafo di rappresentazione e da relative matrici che ne descrivono il comportamento. Si definiscono le matrici di input ed output come segue:

- $I(t_j, p_i)$ Matrice di dimensione $m \times n$, che indica il numero di archi che vanno dal posto p_i alla transizione t_j, $1 \leq i \leq n$, $1 \leq j \leq m$.
- $O(t_j, p_i)$ Matrice di dimensione $m \times n$, che indica il numero di archi che vanno dalla transizione t_j al posto p_i, $1 \leq j \leq m$, $1 \leq i \leq n$.

Con la notazione $I(t_j, \cdot)$ si indica la riga (vettore input) della matrice I relativa alla transizione t_j. Con $I(\cdot, p_i)$ si denota la colonna della matrice I relativa al posto p_i. La stessa notazione vale per la matrice O.

L'informazione contenuta nelle matrici I ed O può essere composta ottenendo un'unica matrice definita matrice di incidenza.

Definizione 3. (Matrice di Incidenza). Data una rete PN, con m transizioni ed n posti, la matrice di incidenza $C : T \times P \to \mathbb{Z}$ è la matrice $m \times n$ calcolata come segue:

$$C = O - I$$

Dove ogni elemento di C vale: $C(t_j, p_i) = O(t_j, p_i) - I(t_j, p_i)$.

Utilizzando le matrici I e O è possibile risalire al grafo di rappresentazione. Ciò non è possibile esclusivamente con la sola matrice C.

Per ogni rete di Petri è possibile definire i seguenti insiemi:

- ${}^\bullet t = \{p \mid (p,t) \in F\}$ Insieme dei posti in ingresso alla transizione t.
- $t^\bullet = \{p \mid (t,p) \in F\}$ Insieme dei posti in uscita alla transizione t.
- ${}^\bullet p = \{t \mid (t,p) \in F\}$ Insieme delle transizioni in ingresso al posto p.
- $p^\bullet = \{t \mid (p,t) \in F\}$ Insieme delle transioni in uscita al posto p.

.

Esempio 4. In Figura 1.3 è rappresentata una rete di Petri $PN = (P, T, F, W, M_0)$ con $P = \{p_1, p_2, p_3, p_4\}$ e $T = \{t_1, t_2, t_3\}$. Le matrici I, O e C e gli insiemi sopra citati sono definiti come segue:

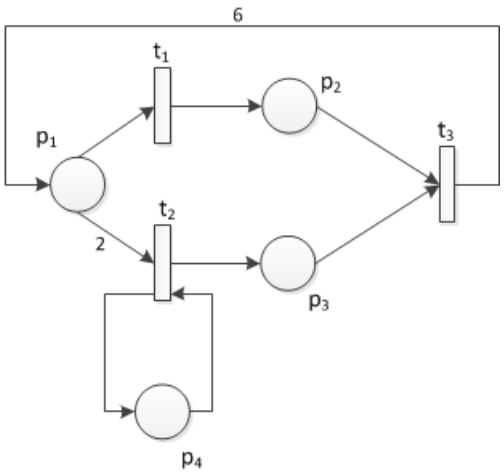

Figura 1.3: Esempio PN per la costruzione delle matrici I, O e C

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad O = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 6 & 0 & 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 6 & -1 & -1 & 0 \end{bmatrix}$$

$${}^\bullet t_1 = \{p_1\} \quad {}^\bullet t_2 = \{p_1, p_4\} \quad {}^\bullet t_3 = \{p_2, p_3\}$$
$$t_1^\bullet = \{p_2\} \quad t_2^\bullet = \{p_3, p_4\} \quad t_3^\bullet = \{p_1\}$$

$${}^\bullet p_1 = \{t_3\} \quad {}^\bullet p_2 = \{t_1\} \quad {}^\bullet p_3 = \{t_2\} \quad {}^\bullet p_4 = \{t_2\}$$
$$p_1^\bullet = \{t_1, t_2\} \quad p_2^\bullet = \{t_3\} \quad p_3^\bullet = \{t_3\} \quad p_4^\bullet = \{t_2\}$$

1.3 Regole di Enablig e Firing

Il comportamento dinamico di una rete di Petri è governato da particolari regole associate ad ogni transizione della rete. Quest'ultime sono rappresentate dalle regole di enabling (abilitazione) e firing (scatto). Tali regole gestiscono operazioni che creano e consumano token modificando la marcatura che descrive lo stato della rete.

In particolare, la regola di enabling descrive le condizioni sotto le quali alle transizioni è permesso eseguire un operazione di firing. La regola di firing definisce come varia la marcatura corrente successivamente allo scatto della transizione. Formalmente:

Definizione 5. (Enabling). Una transizione t è abilitata dalla marcatura M se e solo se:

$$\forall p \in {}^{\bullet}t : M\,(p) \geq I\,(t, p)$$

In particolare, una transizione t è abilitata se per ogni posto $p_i \in {}^{\bullet}t$ questo contiene un numero di token pari o superiore al peso $w\,(p_i, t)$ dell'arco che collega la transizione t al posto p_i.

La seguente notazione $M[t\rangle$ indica che la transizione t è abilitata da M. Mentre $M\neg[t\rangle$ indica che la transizione t non è abilitata da M.

L'insieme delle transizioni abilitate da una marcatura M è indicata con $E(M)$.

Data una marcatura M si definisce grado di abilitazione relativa della transizione t il numero di volte che una transizione t può essere abilitata. Formalmente:

$$e(M) = \max\{k \in \mathbb{N} : M \geq kI(t, \cdot)\}$$

Definizione 6. (Firing). Il firing di una transizione t, abilitata dalla marcatura M, produce una marcatura M' tale che:

$$M' = M + (O\,(t, \cdot) - I\,(t, \cdot))$$

$$M' = M + C\,(t, \cdot)$$

In particolare, il firing di t rimuove $I\,(t, p_i)$ token da ogni posto $p_i \in {}^{\bullet}t$ e aggiunge $O\,(t, p_k)$ token ad ogni posto $p_k \in t^{\bullet}$

La seguente notazione $M[t\rangle M'$ indica che il firing della transizione t, partendo dalla marcatura M, produce la marcatura M'.

Una sequenza di transizioni $\sigma = t_1 t_2 \dots t_n \in T$ è abilitata da una marcatura M se: t_1 abilitata da M dopo il suo firing porta ad una nuova marcatura $M_1 = M + C(t_1, \cdot)$; t_2 abilitata da M_1 dopo il suo firing porta ad una nuova marcatura $M_2 = M_1 + C(t_2, \cdot)$, etc. Tale procedimento viene ripetuto fino a raggiungere la marcatura finale M_n

$$M[t_1\rangle M_1\ [t_2\rangle M_2 \dots [t_n\rangle M_n$$

La sequenza può essere riassunta in $M[\sigma\rangle M_n$.

Dato un insieme di transizioni $T = \{t_1 t_2 \dots t_m\}$ ed una sequenza $\sigma \in T$ si definisce vettore di scatto (firing count vector) associato a σ il vettore $v\sigma$ la cui generica componente $v\sigma_i = \sigma(t_i)$ indica quante volte la transizione t_i compare in σ.

Data una rete (N, M_0), sia C la sua matrice di incidenza, se la marcatura M è raggiungibile da M_0 attraverso il firing delle transizioni $t \in \sigma$ allora vale:

$$M^T = M_0^T + C^T \cdot v\sigma$$

Esempio 7. In Figura 1.4 è rappresentata una rete di Petri $PN = (P, T, F, W, M_0)$ con $P = \{p_1, p_2, p_3, p_4\}$, $T = \{t_1, t_2, t_3\}$ e marcatura iniziale $M_0 = [5\,0\,0\,1]$.

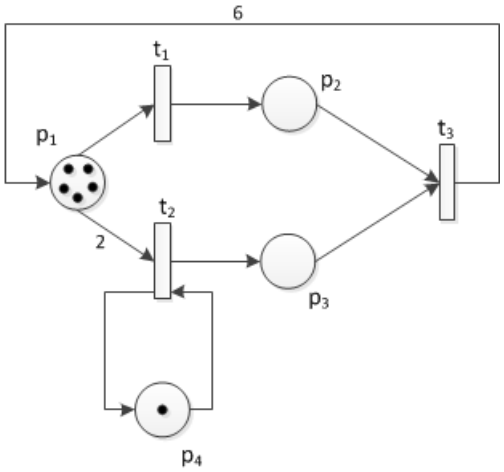

Figura 1.4: Esempio PN utile per l'analisi di una sequenza di firing

Consideriamo la seguente sequenza di firing $M_0[t_1\rangle M_1\ [t_2\rangle M_2\ [t_2\rangle M_3$. Analizzando i singoli passaggi:

$M_1 = M_0 + C(t_1, \cdot) = [5\,0\,0\,1] + [-1\,1\,0\,0] = [4\,1\,0\,1]$
$M_2 = M_1 + C(t_2, \cdot) = [4\,1\,0\,1] + [-2\,0\,1\,0] = [2\,1\,1\,1]$
$M_3 = M_2 + C(t_2, \cdot) = [2\,1\,1\,1] + [-2\,0\,1\,0] = [0\,1\,2\,1]$

Oppure:

$$M_3^T = M_0^T + C^T \cdot v\sigma = \begin{bmatrix} 5 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & -2 & 6 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = [0\,1\,2\,1]^T$$

1.4 Comportamento dinamico

Il comportamento dinamico di una rete di Petri è caratterizzato da tre fenomeni di base: concorrenza, dipendenza casuale e conflitto.

1.4.1 Concorrenza

Due transizioni t_i e t_j sono definite concorrenti a partire da una marcatura M (con notazione: $< t_i, t_j > \in CO(M)$) se $M[t_i\rangle M'$, $M[t_j\rangle M''$, $e_j(M') > 0$ e $e_i(M'') > 0$.

Nello specifico, $< t_i,t_j > \in CO(M)$ se $M > C(t_i, \cdot) + C(t_j, \cdot)$. La concorrenza di due (o più) transizioni t_i, t_j è rappresentata dalla possibilità di eseguire le transizioni simultaneamente o in qualsiasi ordine, prima t_i e poi t_j o viceversa.

Tuttavia la sola possibilità di effettuare i firing di due (o piu) transizioni in ordine differente non implica necessariamente che le due transizioni siano concorrenti.

Come si nota dall'esempio in Figura 1.5 le due transizioni t_1 e t_2 possono essere eseguite in qualsiasi ordine (t_1 e successivamente t_2 oppure t_2 e successivamente t_1) ma non possono essere eseguite simultaneamente. Le due transizioni quindi, non si possono definire concorrenti.

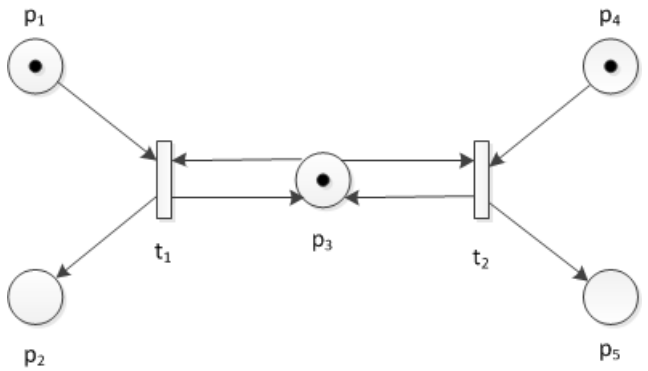

Figura 1.5: Rete di Petri con transizioni non concorrenti

1.4.2 Dipendenza casuale

La transizione t_i è in relazione casuale diretta con la transizione t_j a partire da una marcatura M (con notazione: $< t_i,t_j > \in DC(M)$) se $M[t_i\rangle M'$ e

$e_j(M') > e_j(M)$.

Nello specifico, la transazione t_j può effettuare il firing solamente dopo il firing della transizione t_i. Banalmente, se le transizioni sono connesse tra loro attraverso un singolo posto come in Figura 1.6, allora le due transizioni sono in dipendenza casuale strutturale. Formalmente $< t_i,t_j > \in SCC(M)$ se $t_i^\bullet \cap^\bullet t_j \neq \emptyset$.

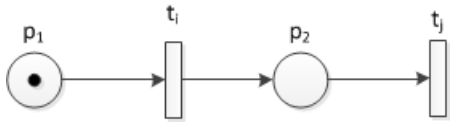

Figura 1.6: Rete di Petri con transizioni in dipendenza casuale

1.4.3 Conflitti

Due transizioni t_i e t_j sono in **conflitto strutturale** (con notazione: $< t_i,t_j > \in SC(M)$) se esiste un posto p_i dal quale escono almeno due transizioni, e in tale insieme sono presenti t_i e t_j. Formalemente $\exists p_i \in P : |p_i^\bullet| \geq 2$ e $t_i, t_j \in p_i^\bullet$.

Una transizione t_i è in **conflitto effettivo** con una transizione t_j a partire da una marcatura M con notazione: $< t_i,t_j > \in EFC(M)$ se e solo se $t_i,t_j \in E(M)$ e se $M[t_i\rangle M'$ e $e_j(M') < e_j(M)$.

In Figura 1.7 (a) le transizioni t_i e t_j non sono in conflitto effettivo. Infatti, il numero di token della marcatura corrente è sufficiente a far scattare la sola transizione t_i.

In Figura 1.7 (b) la transizione t_i è in conflitto effettivo con la transizione t_j in quanto, entrambe sono tranzioni abilitate ma il firing della transizione t_i comporta il decremento del grado di abilitazione della transizione t_j. Tuttavia non è detto il contrario. Infatti la transizione t_j non è in conflitto con la transizione t_i (il suo firing non decrementa il grado di abilitazione dell'altro).

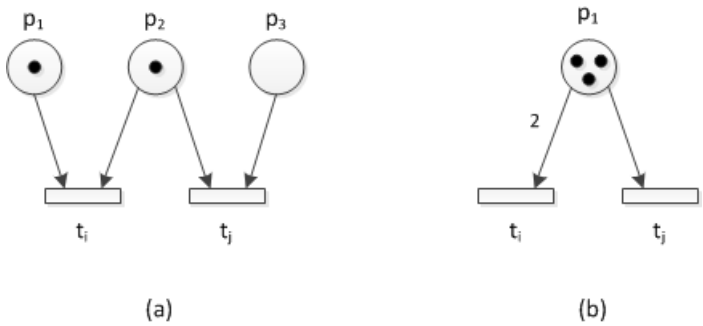

Figura 1.7: Due tipologie di reti di Petri. (a) PN con transizioni non in conflitto effettivo. (b) PN con transizioni in conflitto effettivo

1.5 Proprietà

Le proprietà delle reti di Petri rappresentano delle caratteristiche tali da permettere la valutazione in modo oggettivo delle qualità del sistema stesso. In seguito sono riportate le proprietà basilari per l'analisi comportamentale di una rete di Petri.

1.5.1 Reachability e Reversibily

La proprietà di raggiungibilità viene utilizzata per capire se è possibile raggiungere una particolare marcatura a partire da una marcatura data. Una marcatura M_n è definita raggiungibile (reachable) da una marcatura M_0 se esiste una sequenza di firing σ che partendo da M_0 raggiunge la marcatura M_n.

In una rete di Petri (N, M_0), l'insieme di tutte le possibili marcature raggiungibili a partire da M_0 viene indicata con $RS\,(N, M_0)$ o semplicemente con $RS\,(M_0)$. Formalmente l'insieme $RS\,(M_0)$ è definito ricorsivamente come il più piccolo insieme di marcature tale che:

- $M_0 \in RS\,(M_0)$
- $M_i \in RS\,(M_0) \wedge \exists t \in T : M_i[t\,\rangle M_j \Rightarrow M_j \in RS\,(M_0)$

In generale l'insieme $RS\,(M)$ indica l'insieme delle marcature raggiungibili da un generica marcatura M, tuttavia non contiene informazioni relative alla sequenza delle transizioni che vengono impiegate per raggiungere le marcature contenute nell'insieme. Tale informazione è specificata dal grafo di raggiungibilità, dove ogni nodo rappresenta uno stato raggiungibile ed è presente un arco che collega M_i ad M_j solamente nel caso in cui M_j è direttamente raggiungibile da M_i. Se $M_i[t\,\rangle M_j$ l'arco viene etichettato con t. Due nodi possono essere collegati da due archi differenti (una marcatura puo abilitare due differenti transizioni le quali dopo la fase di firing raggiungono la stessa marcatura).

Definizione 8. (Grafo di Raggiungibilità). Data una rete di Petri (N, M_0) e il relativo insieme di raggiungibilità $RS\,(M_0)$, viene definito grafo di raggiungibilità $RG\,(M_0)$ un grafo orientato ed etichettato dove ogni vertice (nodo) corrisponde ad una marcatura raggiungibile da M_0 (nodo iniziale del grafo) e l'insieme degli archi A viene definito come segue:
$A \subseteq RS(M_0) \times RS(M_0) \times T$
$\langle M_i, M_j, t\rangle \in A \Longleftrightarrow M_i[t\,\rangle M_j$

Data una qualsiasi rete di Petri (N, M_0), l'algoritmo per la costruzione del grafo di raggiungibilità si sviluppa nei seguenti passi:

Passo 1: Creare il nodo corrispondente alla marcatura iniziale M_0 e impostare questo come nodo corrente N_{corr}.

Passo 2: Definire M_{corr} la marcatura corrispondente al nodo corrente. Se non esistono transizioni t attivabili a partire da M_{corr} allora M_{corr-1} diventa il nodo corrente e si calcolano le transizioni attivabili da M_{corr-1}. L'algoritmo termina se $corr == 0$.

Passo 3: Se esistono transizioni ancora attivabili a partire da M_{corr} eseguire la prima delle transizioni attivabili, non considerate in precedenza e calcolare la nuova marcatura.

- Se tale marcatura non è presente nell'insieme delle marcature precedenti, allora creare una nuova marcatura M_{corr+1} e il relativo nodo che diventerà il nodo corrente N_{corr}. Creare un arco tra i nodi M_{corr} e M_{corr+1} ed etichettarlo con il nome della transizione ad esso associata.
- Se invece la marcatura già appartiene all'insieme delle marcature esistenti, creare un arco tra M_{corr} e la marcatura trovata (qualora esso non esista). Il nodo attivo rimane M_{corr}.

Passo 4: Ripetere l'algoritmo dal Passo 2.

Nel caso di reti di Petri il cui insieme di raggiungibilità ha cardinalità infinita, si utilizza il grafo di copertura per rappresentare graficamente $RS\,(M_0)$.

In questo caso ci sono delle transizioni il cui firing porta ad incrementare sempre almeno una componente del vettore della marcatura corrente. Per poter quindi rappresentare l'insieme di raggiungibilità a cardinalità infinita si introduce il simbolo ω associandolo al valore di $M(i)$ con crescita infinita.

Esempio 9. La Figura 1.8 illustra una rete di Petri $PN = (P, T, F, W, M_0)$ con $M_0 = [2\,0\,0\,0]$.

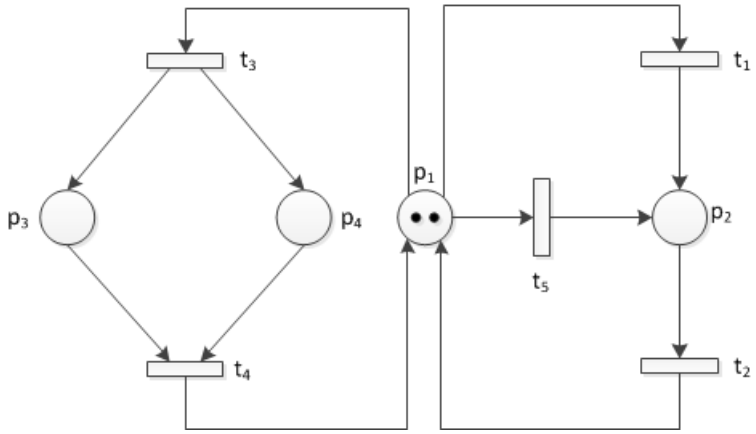

Figura 1.8: Rete di Petri utilizzata per la costruzione del grafo di raggiungibilità

A partire dalla rete in Figura 1.8 ed utilizzando l'algoritmo descritto in precedenza, si riporta in Figura 1.9 il grafo di raggiungibilità della rete oggetto di studio:

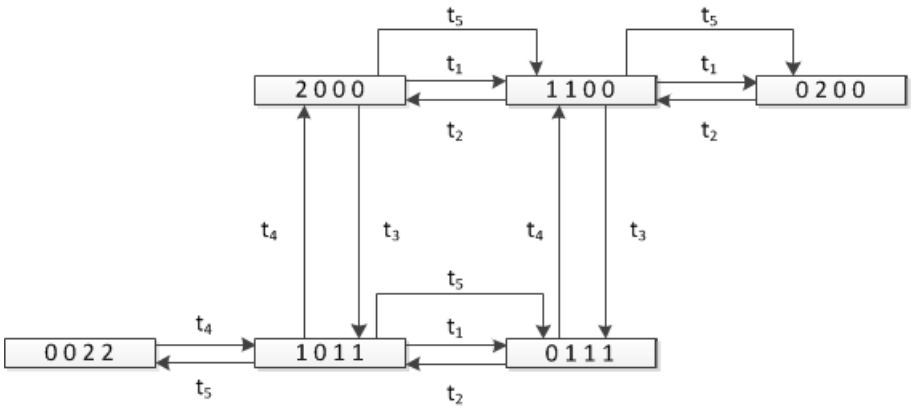

Figura 1.9: Grafo di raggiungibilità della rete in Figura 1.8

Definizione 10. (Rete reversible). Una rete di Petri è definita reversible se e solo se per ogni marcatura M appartenente agli stati (marcature) raggiungibili da M_0, la marcatura M_0 appartiene agli stati raggiungibili da M. Vi è quindi la possibilita di ritornare alla marcatura iniziale. Formalmente:

$$\forall M \in RS(M_0), M_0 \in RS(M)$$

In generale definiamo *home-state* la marcatura $M \in RS(M_0)$ se e solo se $\forall M' \in RS(M_0), M \in RS(M')$. Una rete di Petri è reversibile ogni volta che lo stato iniziale M_0 è home-state.

1.5.2 Liveness

Definizione 11. (Liveness). Una transizione t è definita *live* se e solo se, per ogni marcatura M raggiungibile da M_0, esiste una marcatura M', raggiungibile da M tale che t appartiene all'insieme delle transazioni abilitate dalla marcatura M'. Formalmente una transizione t è live se [7]:

$$\forall M \in RS(M_0), \exists M' : \big(M \to M' \wedge t \in E(M')\big)$$

Una rete di Petri è definita *live* se e solo se per ogni $t \in T : t$ è live.

La proprietà di liveness definisce la possibilità per una transizione di essere abilitata (e successivamente effettuare un firing) infinite volte, inoltre dipende da una marcatura iniziale. Una transizione non live viene definita dead. Per ogni transizione t dead è possibile trovare una marcatura M tale che nessuna delle marcature contenute in $RS(M)$ abilita t.

Un importante conseguenza della proprietà di liveness consiste nel fatto che se almeno una delle transizioni è live, la rete di Petri non può presentare deadlock (situazione in cui la rete raggiunge una marcatura nella quale nessuna transizione può essere abilitate ad effettuare un firing).

Liveness è una proprietà ideale per molti sistemi, tuttavia verificare tale proprietà risulta essere poco pratico e troppo costoso. Dunque la condizione di liveness viene indebolita definendo i seguenti livelli di liveness:

Una transizione t in una rete di Petri (N, M_0) è definita:

- dead ($L_0 - live$): Se non vi è mai la possibilità di eseguire il firing di t in qualsiasi sequenza di scatto in $L(M_0)$. $L(M_0)$ indica l'insieme di tutte le possibile sequenze di firing a partire da M_0.
- $L_1 - live$: Se è possibile effettuare il firing della transizione t almeno una volta in alcune sequenze di scatto in $L(M_0)$.
- $L_2 - live$: Se dato un qualsiasi intero positivo k è possibile effettuare il firing di t almeno k volte in alcune sequenze in $L(M_0)$.
- $L_3 - live$: Se il firing della transizione t può essere eseguito un numero di volte indefinito in alcune sequenze in $L(M_0)$.
- $L_4 - live$: Se t è $L_1 - live$ per ogni marcatura.

Una rete di Petri (N, M_0) è definita $L_k - live$ se ogni transizione della rete è $L_k - live$ con $k = 0, 1, 2, 3, 4$. Inoltre:

$$L_4 - liveness \Rightarrow L_3 - liveness \Rightarrow L_2 - liveness \Rightarrow L_1 - liveness$$

1.5.3 Boundedness

Definizione 12. (Boundedness). Un posto p di una rete di Petri è definito $k-$bounded se e solo se per ogni marcatura M raggiungibile da M_0, il numero di token in p è minore o uguale a k. Formalmente un posto p_i è $k-$bounded se e solo se [7]:

$$\forall M \in RS(M_0), \exists k : M(i) \leq k$$

Una rete di Petri è definita $k-$bounded se e solo se tutti i posti $p \in P$ sono $k-$bounded. Una rete di Petri $1-$bounded è definita *safe*.

Una rete di Petri $k-$bounded con $|P| = N$ presenta un numero di marcature raggiungibili minori o uguali a $(k+1)^N$.

1.5.4 Coverability

Definizione 13. (Coverability). Una marcatura M in una rete di Petri è definita *coverable* se esiste una marcatura M' appartenete a $RS(M_0)$ tale che $M'(i)$ è maggiore o uguale di $M(i)$ per ogni i appartenete a P. Formalmente M è coverable se:

$$\exists M' \in RS(M_0) | M'(i) \geq M(i) \, \forall i \in P$$

1.5.5 Mutua esclusione

La proprietà di mutua esclusione è relativa sia ai posti che alle transizioni.

Definizione 14. (Mutua Esclusione). In una rete di Petri due posti p e q sono in mutua esclusione se per ogni marcatura raggiungibile da M_0 il numero dei token relativi a p e q non è maggiore di zero per entrambi. Formalmente p e q sono in mutua esclusione se [7]:

$$\forall M \in RS(M_0) \quad M(p) \cdot M(q) = 0$$

Due transizioni t_i e t_j si dicono in mutua esclusione tra loro se non sono abilitate entrambe a partire da ogni marcatura M raggiungibile da una marcatura iniziale M_0. Formalmente t_i e t_j sono in mutua esclusione se:

$$\forall M \in RS(M_0) \quad (M[t_i\rangle \wedge M\neg[t_j\rangle) \vee (M\neg[t_i\rangle \wedge M[t_j\rangle)$$

Esempio 15. In Figura 1.10 sono riportati esempi di rete di Petri che soddisfano differenti proprietà:

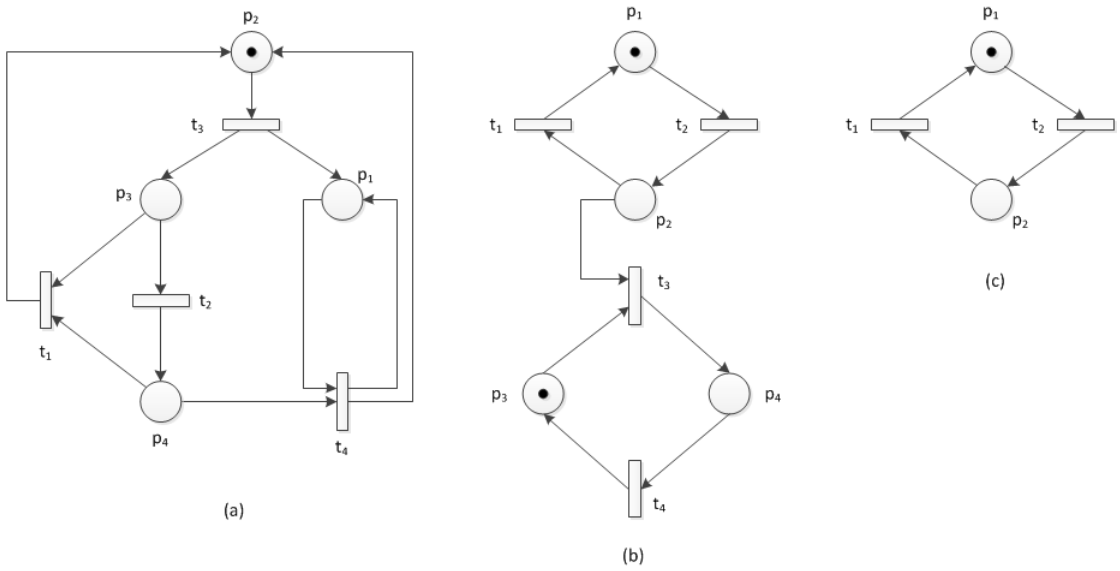

Figura 1.10: Reti di Petri caratterizzate da proprietà di boundedness, liveness e reversibily

La rete di Petri in Figura 1.10 (a) non soddisfa le proprietà di boundedness, liveness e reversibily (la transizione t_1 è dead ed il posto p_1 è unbounded).

La rete di Petri in Figura 1.10 (b) soddisfa solamente la proprietà di boundedness (t_1, t_2, t_3, t_4 non sono $L_4 - live$) .

La rete di Petri in Figura 1.10 (c) soddisfa le proprietà di boundedness liveness e reversibily.

1.6 Invarianti

Gli invarianti rappresentano generalmente proprietà che si conservano durante l'evoluzione della rete.

1.6.1 T-invariante

Data una rete di Petri, un T-invariante (invariante di transizione) è un vettore $m-$dimensionale le cui componenti rappresentano il numero di occorrenze di firing che devono essere effettuate da ogni transizione per ottenere una variazione complessiva nulla della marcatura, ovvero per riportare la rete nella marcatura iniziale. In altre parole data una rete di Petri (N, M_0), il vettore colonna $X = (x_1, \ldots, x_m)$ con $m = |T|$ è un T-invariante se:

$$M^T = M_0^T + C^T X = M_0^T$$

Definizione 16. (T-invariant). Data una rete di Petri (N, M_0) con matrice di incidenza C, un T-invariant di PN è un vettore $X = (x_1, \ldots, x_m)$ che si ottiene dalla soluzione del seguente sistema [4]:

$$C^T X = 0$$

dove C^T è la matrice di incidenza trasposta corrispondente alla rete di Petri data.

L'esistenza di un T-invariante non implica necessariamente l'esistenza di una sequenza ammissibile tale che il vettore delle occorrenze associato coincida con il T-invariante. Infatti la marcatura iniziale M_0 potrebbe non soddisfare la condizione di abilitazione per nessuna transizione, ovvero potrebbe non esistere nessuna sequenza di enabling e firing tale che il relativo vettore delle occorrenze sia proprio il T-invariante.

Se la rete di Petri (con un insieme di raggiungibilità che non contiene la sola marcatura M_0) non ammette alcun T-invariante allora la rete è sicuramente non reversibile.

Un vettore T-invariante X si definisce minimo se non esiste un vettore X' tale che $x_i^{'} \leq x_i$ per ogni $i = 1, \ldots, m$.

Esempio 17. Data la seguente rete di Petri riportata in Figura 1.11.

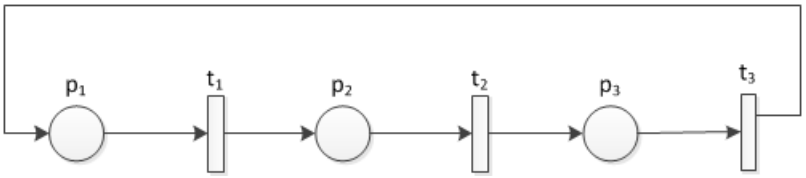

Figura 1.11: Esempio PN per lo studio del vettore T-invariante.

Il vettore $X = [1\,1\,1]$ è un T-invariante, infatti, una qualunque sequenza che prevede un singolo firing di ciascuna delle tre transizione che costituiscono la rete riporta la rete alla marcatura di partenza.

1.6.2 P-invariante

Ogni P-invariante corrisponde all'insieme dei posti tali per cui la somma pesata dei token che presentano al loro interno rimane costante per tutte le marcature raggiungibili dalla rete.

Un P-invariante è definito come un vettore colonna avente la stessa dimensione dell'insieme dei posti P.

Data una rete di Petri (N, M_0), il vettore colonna $Y = (y_1, \ldots, y_n)$ con $n = |P|$ è un P-invariante se:

$$M \in RS\,(M_0) \Rightarrow \sum_{i=1}^{n} y_i M_i = k$$

dove k è una costante reale. Se tutte le componenti del vettore Y sono positive allora la rete è definita *conservativa*, ciò significa che il numero di token all'interno della rete rimane costante.

Definizione 18. (P-invariant). Data una rete di Petri (N, M_0) con matrice di incidenza C, un P-invariant di PN è un vettore $Y = (x_1, \ldots, x_n)$ che si ottiene dalla soluzione del seguente sistema:

$$CY = 0$$

Un vettore Y P-invariante è detto minimo se non esiste nessun'altro Y' tale che $y_i^{'} < y_i$ per ogni $i = 1, \ldots, n$.

Esempio 19. Consideriamo la rete di Petri illustrata in Figura 1.12.

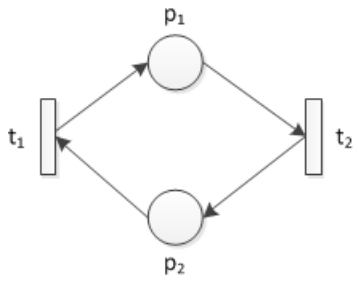

Figura 1.12: Rete di Petri per il calcolo del vettore P-invariante

Osserviamo che, a prescindere dalla marcatura iniziale, il firing della transizione t_1 elimina un token dal posto p_2 e lo ripone in p_1. Analogamente il firing di t_2 preleva un token da p_1 e lo ripone in p_2. Inoltre, dato che il numero di token non può variare a meno del firing di una delle due transizioni t_1 e t_2 allora possiamo dire che qualunque sia la marcatura iniziale la somma dei token nei posti p_1 e p_2 rimane costante ed è pari al valore che assume inizialmente per ogni stato raggiungibile. Possiamo concludere che $Y = [1\,1]$ è P-invariante per questo esempio di rete di Petri.

Capitolo 2

Reti di Petri stocastiche

2.1 Il tempo nelle reti di Petri

Nel capitolo precedente abbiamo introdotto i principali concetti delle reti di Petri; tali reti rappresentano uno strumento affidabile e potente per lo studio delle iterazioni in sistemi ad eventi discreti (DEDS).

Tuttavia, per uno studio adeguato delle prestazioni di tali sistemi è necessario introdurre una metrica temporale al fine di garantire un corretta e dettagliata rappresentazione del loro comportamento. Da tale accorgimento vengono definite le reti di Petri temporizzate (TPN).

Le TPN garantiscono, a differenza di altri modelli, una riproduzione fedele delle caratteristiche principali dei DEDS in quanto offrono la possibilità di descrivere proprietà quali priorità, sincronizzazione ed uso di risorse condivise. Permettono inoltre di definire modelli logici e prestazionali per lo studio di veri e propri sistemi concorrenti, in grado di descrivere tutti gli aspetti di parallelismo e di conflitto nelle attività asincrone che coinvolgono molteplici attori.

In particolare, il concetto del tempo non risulta essere importante nel momento in cui si considerano solo le relazioni logiche tra le entità che fanno parte del sistema reale, diventa invece di fondamentale importanza quando lo studio viene rivolto all'efficienza e alle prestazioni dello stesso. Tali reti quindi trovano una loro naturale applicazione in campi come: architettura hardware, computer design, protocolli di comunicazione ed analisi dei sistemi software.

Con l'introduzione del concetto del tempo si è quindi stati in grado di fornire metodi per il calcolo degli indici di prestazione. Esistono differenti metodi per incorporare il concetto del tempo in una rete di Petri. I diversi modi sono influenzati dai campi di applicazione e possono essere riassunti nei seguenti punti [3]:

- Posti: I token in output da un posto diventano disponibili per effettuare il firing in una transizione solo dopo aver trascorso nel posto un

determinato quanto di tempo.

- Token: Ad ogni token viene associato un timestamp. Tale valore indica quanto tempo deve trascorrere prima che lo stesso token venga inviato alla transizione per successivamente eseguirne il firing. Il valore del timestamp può essere incrementato ad ogni firing di transizione.

- Archi: Ad ogni arco è associato un quanto temporale per l'attraversamento dell'arco stesso. I token sono disponibili al firing di una transizione non appena raggiungono la transizione stessa.

- Transizioni: Il ritardo associato ad una transizione corrisponde al tempo che intercorre tra la sua abilitazione e il termine dell'operazione di firing (quando i token sono riposti nei posti in uscita).

2.1.1 Rete di Petri con transizioni temporizzate

Formalmente una rete di Petri con transizioni temporizzate (TTPN) è definita dalla coppia $< R, Tempo >$ dove:

- R è una rete di Petri con marcatura iniziale M_0.

- $Tempo$: è una funzione $T \rightarrow \mathbb{R}_0^+$ che restituisce un valore temporale (d_j) per ogni transizioni $t_j \in T$.

Una transizione t_j di una TTPN può assumere differenti forme in base al valore d_j ad essa associato:

- Immediata: Il tempo associato a t_j è nullo. Ciò comporta che la transizione esegue il proprio firing nel medesimo istante in cui viene abilitata.

- Deterministica: Ad ogni transizione viene associato un ritardo d_j scelto in modo deterministico [5]. Le reti cosi ottenute sono chiamate DTPN (Deterministic timed transitions Petri net), definite per la prima volta da Ramamoorthy e Ho nel 1980. Tali modelli trovano una loro applicazione all'interno di una classe ristretta di sistemi denominati decision-free nets. In termini strutturali, ogni posto è collegato sia in ingresso che in uscita, rispettivamente con non più di una transizione cosi da garantire un aspetto deterministico del comportamento dinamico.

- Stocastica: Indicate con SPN (Molloy 1982; Bause and Kritzinger 2002). Il tempo associato a t_j viene calcolato tramite una funzione che segue una distribuzione di probabilità nota. In particolare ad ogni transizione viene associato un parametro λ_j definito *velocità* o *rate* della transizione t_j, il quale è descritto da una variabile aleatoria che

segue una distribuzione esponenziale negativa corrispondente all'inverso del ritardo medio relativo al firing. Maggiori dettagli sono riportati nel paragrafo 2.2.

2.1.2 Comportamento, politiche di Firing e conflitti.

Per descrivere il comportamento di una rete temporizzata, possiamo immaginare che ad ogni transizione t_j sia associato un timer inizializzato con il valore d_j. Supponiamo che T sia la particolare istanza del tempo associata ad una transizione e calcolata a partire da una distribuzione di probabiltà. Supponiamo inoltre che la transizione esegua il firing all'istante t_0 e successivamente venga nuovamente abilitata all'istante $t_1 \geq t_0$. Se dopo un tempo $T_c < T$ e quindi all'istante $t_1 + T_c$ la transizione viene disabilitata, il timer si arresta. Nel momento in cui al tempo t_2, la stessa transizione viene di nuovo abilitata il ritardo residuo prima del firing sarà pari a $T - T_c$. Se all'istante $t_2 + T - T_c$ la transizione è ancora abilitata, esegue il firing (Figura 2.1).

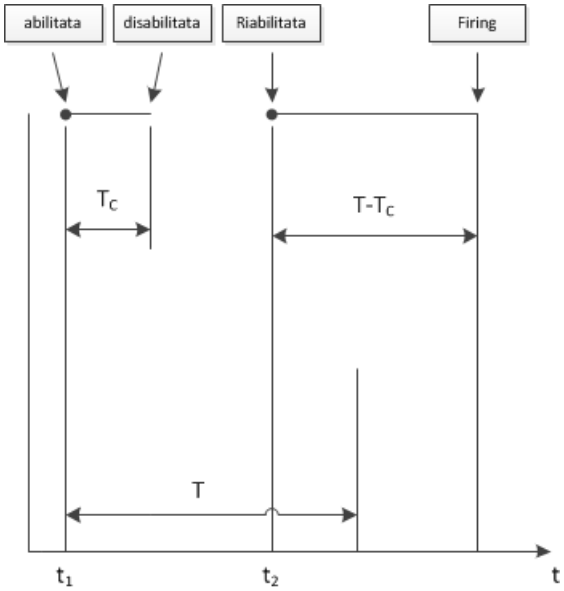

Figura 2.1: Comportamento di una TTPN

Per l'assenza di memoria delle variabili casuali con densità di probabilità esponenziale, il tempo residuo prima del firing ha la stessa distribuzione del tempo che viene campionato dalla distribuzione del ritardo la prima volta che la transizione è abilitata dopo il firing. Il comportamento delle rete è dunque equivalente a quello che si avrebbe se tutte le volte che si entra in una nuova marcatura ogni transizione abilitata ricampionasse dalla propria densità di probabilità del ritardo una nuova istanza.

Esistono due differenti politiche di firing associate alle transizioni temporizzate:

- **Three-phase Firing** (scatto in tre fasi):
  - ▷ Nella prima fase si prelavano i token dai posti in input alla transizione. Tale fase viene denominata *inizio del firing* ed il tempo associato al suo completamento è nullo.
  - ▷ Nella seconda fase denominata *firing in corso*, viene simulato il tempo relativo all'attività modellata dalla transizione. Tale valore è quindi una grandezza reale e positiva.
  - ▷ Nella terza fase vengono depositati i token nei posti in output della transizione. Tale fase viene denominata *fine del firing* ed il tempo associato al suo completamento è nullo.
- **Atomic Firing** (scatto atomico): Il firing della transizione viene considerata un'operazione atomica. La temporizzazione viene ottenuta associando ad ogni transizione un ritardo corrispondente al tempo che deve trascorrere tra l'abilitazione e il firing della transizione stessa. I token vengono mantenuti nel posto di input fino all'esecuzione del firing, a meno che una differente transizione utilizzi gli stessi token per effettuare il proprio firing. La transizione in attesa esegue finalmente il suo firing solamente se al termine del quanto temporale continua a sussistere la condizione di abilitazione.

In presenza di più transizioni temporizzate e abilitate contemporaneamente (in conflitto o meno) è necessario definire un insieme di regole per stabilire quale delle transizioni possa effettuare il firing per prima.

Principalmente vengono definite due politiche [3]:

1. **Preselection**: La transizione abilitata che esegue il firing per prima viene scelta sulla base di alcune metriche (es. priorità).
2. **Race**: La risoluzione dei conflitti dipende dai valori temporali associati alle transizioni. Quando più transizioni temporizzate sono abilitate al firing a partire da una marcatura M, la transizione candidata ad eseguire il firing è quella alla quale è stato associato il quanto di tempo minore (disabilitando cosi le possibili transizioni in conflitto). Questa politica viene utilizzata nei modelli GSPN (Generalized Stochastic Petri Nets) e SPN (Stochastic Petri Nets), in cui il tempo di firing delle transizioni segue una variabile aleatoria esponenziale negativa.

Esempio 20. Consideriamo un esempio che illustra la politica di race a fronte di due transizioni in conflitto, mostrato in Figura 2.2.

Supponendo che τ_1 e τ_2 siano rispettivamente i quanti temporali associati alle transizioni t_1e t_2 con $\tau_1 < \tau_2$ e la marcatura iniziale sia $[1\,0\,0]$, l'insieme

delle transizioni abilitate comprende sia la transizione t_1 che la transizione t_2. In base alla politica race la transizione candidata ad eseguire il firing risulta essere t_1 dato che $\tau_1 < \tau_2$.

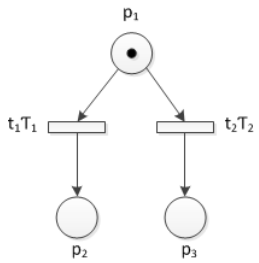

Figura 2.2: Politica race a fronte di due transizioni in conflitto

Quando due o più transizioni immediate sono abilitate al firing nel medesimo istante è necessario definire delle regole per specificare quale di queste deve eseguire il firing per prima. Anche in questo caso vengono utilizzate metriche opportune quali per esempio la priorità, oppure l'associazione di una funzione di distribuzione di probabilità discreta all'insieme delle transizioni conflittuali. In questo caso i conflitti tra transizioni immediate sono risolti in modo casuale.

In tale elaborato tratteremo con maggior attenzione le SPN e le GSPN. Entrambe sono tipologie di reti le cui transizioni sono associate a quanti di tempo stocastici. Tale caratteristica permette di trascurare il caso in cui vi sia la possibilità di firing contemporaneo di due o più transizioni (infatti la probabilità dell'evento è zero in una rete SPN o in una GSPN).

2.2 Reti SPN

Le reti di Petri stocastiche (SPN) sono reti di Petri a cui viene associato un valore di delay (quanto temporale) ad ogni transizione t_i appartenente alla rete. Tale valore indica il tempo che intercorre dall'abilitazione della transizione t_i (fase di enabling) allo scatto (fase di firing) della stessa.

Nel caso in cui più transizioni sono contemporaneamente abilitate la transizione candidata ad eseguire il firing è quella a cui è associato il tempo minore.

Il valore di delay associato alle transizioni segue una variabile casuale continua X con funzione di distribuzione di probabilità esponenziale negativa

$$F_X(x) = Prob\{X \leq x\} = 1 - e^{-\lambda/x}$$

(o funzione di densità di probabilità $f_X(x) = \lambda_i e^{-\lambda_i/x}$).

Il valore di delay medio si ricava mediante la seguente formula

$$d_i = \int_0^\infty (1 - F_X(x))\, dx = \int_0^\infty e^{-\lambda/x} dx = \frac{1}{\lambda_i}$$

dove λ_i rappresenta il tasso di firing (rate) della transizione. Tale concetto verrà analizzato successivamente nel dettaglio.

Definizione 21. (Rete di Petri Stocastica) Formalmente, una rete di Petri stocastica è rappresentata da un 6-tupla definita nel seguente modo [1] [3] [5]:

$$SPN = (P, T, I, O, W, M_0)$$

$P = \{p_1, p_2, \ldots, p_n\}$ rappresenta un insieme finito di posti.

$T = \{t_1, t_2, \ldots, t_m\}$ rappresenta un insieme finito di transizioni.

$I(t_j, p_i)$ Matrice di dimensione $m \times n$, che indica il numero di archi che vanno dal posto p_i alla transazione t_j, $1 \leq i \leq n$, $1 \leq j \leq m$.

$O(t_j, p_i)$ Matrice di dimensione $m \times n$, che indica il numero di archi che vanno dalla transizione t_j al posto p_i, $1 \leq j \leq m$,$1 \leq i \leq n$.

$W(t_j, M) : T \times \mathbb{N}^n \to \mathbb{R}_0^+$ è una funzione che specifica un tasso di firing dipendente dallo marcatura.

$M_0 : P \to \mathbb{N}$ rappresenta la marcatura iniziale.

$P \cap T = \emptyset \wedge P \cup T \neq \emptyset$.

La funzione W permette la definizione della componente stocastica della SPN e mappa le transizioni in funzioni reali e positive relative alle marcature della rete stessa. Per ogni transizione t_j, si definisce la funzione $W(t_j, M)$. Nel caso di indipendenza dalla marcatura si utilizza la notazione w_k per indicare $W(t_k)$ per ogni transizione $t_k \in T$. La quantità w_k è definita *tasso di firing* della transizione t_k.

In questo capitolo, vengono analizzate solamente SPN con uno spazio degli stati finito rendendo le catene di Markov risolvibili con tecniche numeriche standard. Data la proprietà di assenza di memoria relativa alla distribuzione esponenziale, infatti, il grafo di raggiungibilità di una rete di Petri "bounded" e stocastica risulta isomorfo ad una catena di Markov continua e finita (CTMC).

In generale, la catena di Markov si ottiene applicando le seguenti regole:

1. Lo spazio degli stati della catena di Markov S corrisponde all'insieme dei nodi relativo al grafo di raggiungibilità $RS(M_0)$. Percui ad ogni marcatura $M_i \in RS(M_0)$ corrisponde uno stato s_i, con $s_i \in S$.

2. Il tasso di firing di transizione dallo stato s_i, corrispondente alla marcatura M_i, allo stato s_j, corrispondente alla marcatura M_j, si ottiene mediante la somma dei tassi di firing relativi alle transizioni abilitate della marcatura M_i e i cui firing generano la marcatura M_j.

Basandosi su queste due semplici regole, è possibile concepire algoritmi per la costruzione automatica di un generatore infinitesimale denotato come *matrice dei rate di transizione tra stati* di una CTMC a partire dalla descrizione di una SPN.

Assumiamo che tutte le transizioni della rete operino con semantiche single-server e con velocità indipendenti dalle marcature e denotiamo con $\mathbf{Q}$ la matrice dei rate di transizione, con w_k il tasso di firing della transizione t_k ed infine con:

$$E_j\left(M_i\right) = \{h : t_h \in E\left(M_i\right) \wedge M_i[t_h\rangle M_j]\}$$

l'insieme delle transizioni i cui firing portano la rete dalla marcatura M_i alla marcatura M_j.

Allora, le componenti della matrice $\mathbf{Q}$ sono le seguenti:

$$q_{ij} = \begin{cases} \sum_{t_k \in E_j(M_i)} w_k & se \;\; i \neq j \\ -q_i & se \;\; i = j \end{cases}$$

dove $q_i = \sum_{t_k \in E(M_i)} w_k$.

Vengono considerate solamente SPN che danno origine a CTMC omogenee ed ergodiche. Una SPN k-bounded è ergodica se genera una CTMC ergodica; è possibile dimostrare che una SPN è ergodica se M_0 è homestate.

Se la rete SPN è ergodica, esiste la distribuzione stazionaria di probabilità di stato relativa alle marcature ed è definita mediante il seguente limite:

$$\boldsymbol{\pi} = \lim_{\tau \to \infty} \boldsymbol{\pi}\left(\tau\right)$$

Tale valore può essere calcolato risolvendo il seguente sistema lineare di equazioni:

$$\boldsymbol{\pi Q} = \mathbf{0} \wedge \boldsymbol{\pi}\mathbf{1}^T = \mathbf{1} \tag{2.1}$$

dove $\mathbf{0}$ rappresenta un vettore riga avente la stessa dimensione del vettore $\boldsymbol{\pi}$ con componenti pari a zero. Il vettore $\mathbf{1}^T$ rappresenta una vettore colonna con dimensione pari a quella di $\boldsymbol{\pi}$ e componenti di valore unitario.

L'equazione $\boldsymbol{\pi}\mathbf{1}^T = \mathbf{1}$ del sistema viene utilizzata per garantire la condizione di normalizzazione relativa alle probabilità.

2.2.1 Tempo di soggiorno e probabilità di firing

Il tempo di soggiorno indica il tempo speso dalla rete di Petri in una data marcatura M_i. Tale valore segue una distribuzione esponenziale negativa di tasso q_i (relativa alla marcatura M_i).

Il tempo medio di soggiorno nella marcatura M_i è dato dalla seguente espressione.

$$SJ_i = \frac{1}{\sum_{t_j \in E(M_i)} w_j} = \frac{1}{q_i}$$

Per una rete di Petri stocastica le regole di firing sono identiche a quelle di una rete di PN definita nel Capitolo 1 con la differenza che nel caso di una SPN la scelta della transizione candidata ad eseguire il firing è di tipo probabilistico. La probabilità che una data transizione abilitata $t_k \in E(M_i)$ esegua il firing (per prima) a partire dalla marcatura M_i viene calcolata come segue [3]:

$$P\{t_k|M_i\} = \frac{w_k}{\sum_{t_j \in E(M_i)} w_j} = \frac{w_k}{q_i}$$

2.2.2 Esempi

Seguono due esempi di SPN di cui vengono illustrati grafo di raggiungibilità, CTMC associata, matrice $\boldsymbol{Q}$ e distribuzione stazionaria di probabilità di stato:

Esempio 22. Come primo esempio è analizzata una rete di Petri stocastica con tassi di firing associate alle transizioni indipendenti dalla marcatura. In Figura 2.3 è rappresentata una SPN con $P = \{p_1, p_2, p_3, p_4\}$, $T = \{t_1, t_2, t_3\}$ e marcatura iniziale $M_0 = [1\,0\,0\,1]$.

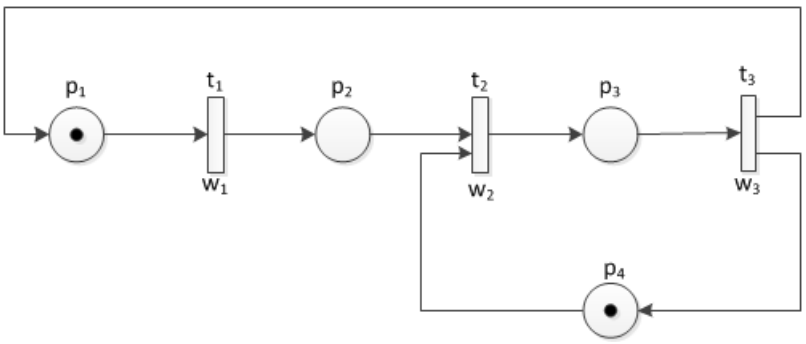

Figura 2.3: Esempio di SPN con transizioni a cui sono associati tassi di firing indipendenti dalla marcatura

Le Figure 2.4 (a) ed 2.4 (b) mostrano rispettivamente il corrispondente grafo di raggiungibilità e la CTMC.

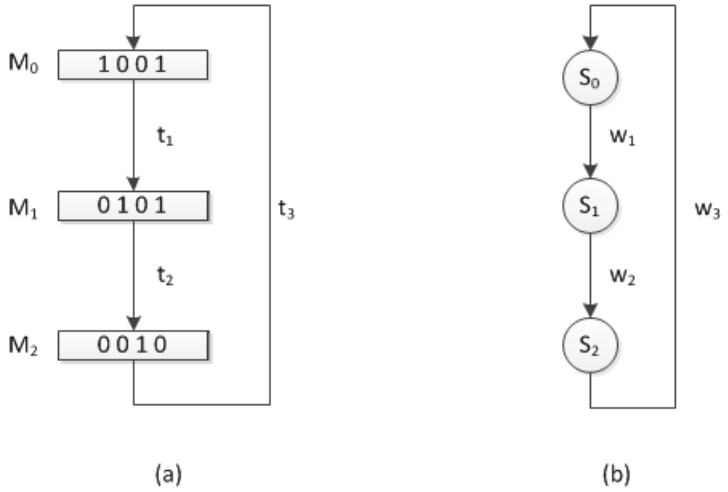

Figura 2.4: (a) Grafo di raggiungibilità e (b) CTMC della rete in Figura 2.3

La matrice $\boldsymbol{Q}$ contenete i rate di transizione è la seguente:

$$\boldsymbol{Q} = \begin{bmatrix} -w_1 & w_1 & 0 \\ 0 & -w_2 & w_2 \\ w_3 & 0 & -w_3 \end{bmatrix}$$

Risolvendo il sistema lineare (2.1) si ottengono le seguenti quattro equazioni.

$$\begin{cases} -w_1\pi_0 + \pi_2 w_3 = 0 \\ w_1\pi_0 - \pi_1 w_2 = 0 \\ w_2\pi_1 - \pi_2 w_3 = 0 \\ \pi_0 + \pi_2 + \pi_3 = 1 \end{cases}$$

Le cui soluzioni della distribuzione stazionaria di probabilità di stato sono:

$$\begin{cases} \pi_0 = \frac{w_2 w_3}{W} \\ \pi_1 = \frac{w_1 w_3}{W} \\ \pi_2 = \frac{w_1 w_2}{W} \end{cases}$$

dove $W = w_2 w_3 + w_3 w_1 + w_2 w_1$.

Esempio 23. In questo esempio consideriamo nuovamente la rete di Petri illustrata in Figura 1.8 [1] a cui vengono associati i seguenti tassi di firing alle transizioni:

$t_1 \rightarrow w_1$
$t_2 \rightarrow m_2 w_2$
$t_3 \rightarrow w_3$
$t_4 \rightarrow w_4$

$t_5 \rightarrow w_5$

Ottenendo in tal modo una rete di Petri stocastica. Tale SPN presenta una transizione (t_2) avente un tasso di firing dipendente dalla marcatura e pari a $m_2 w_2$ dove m_2 indica il numero di token presenti nel posto p_2. Il tasso di firing delle transizioni t_1, t_3, t_4 sono tutte indipendenti dalla marcatura ed i loro valori sono rispettivamente w_1, w_3, w_4.

Le Figura 2.5 (a) ed 2.5 (b) illustrano rispettivamente il grafo di raggiungibilità e la CTMC della rete di Petri stocastica analizzata.

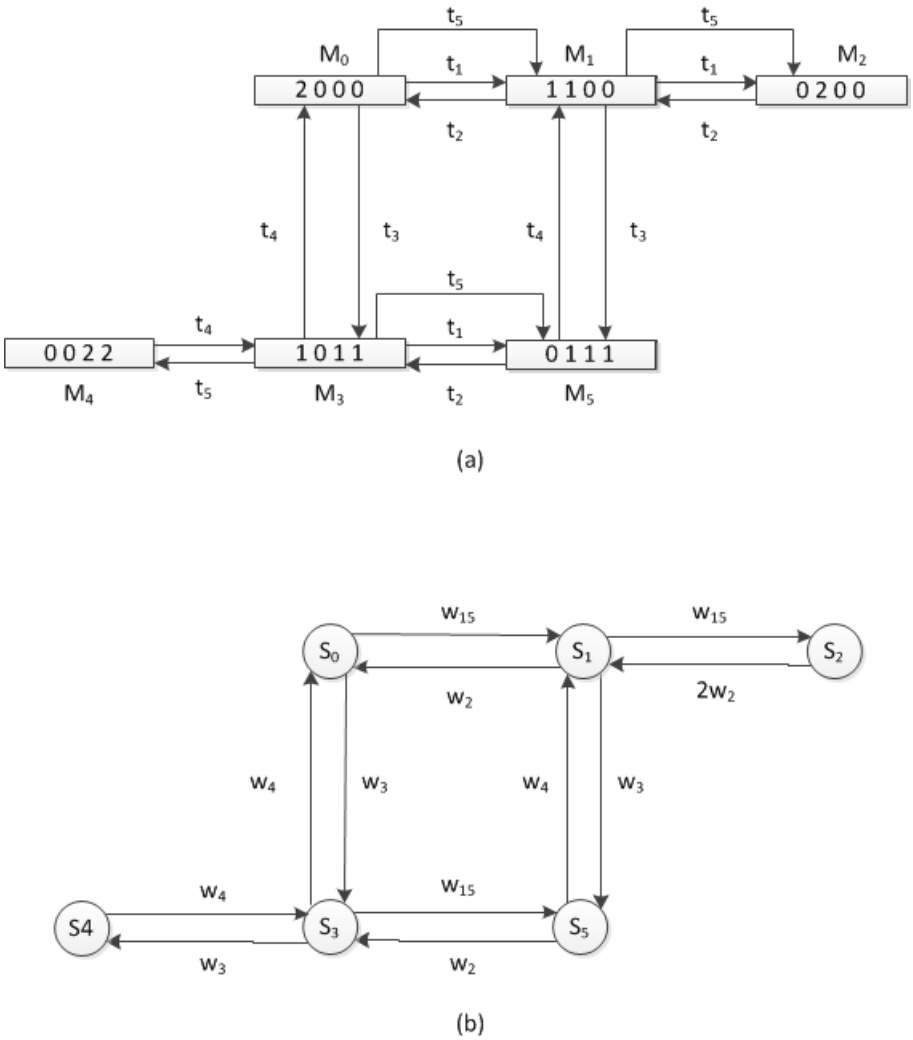

Figura 2.5: (a) Grafo di raggiungibilità e (b) CTMC della rete con tasso di firing dipendente dalla marcatura

La matrice $\boldsymbol{Q}$ contenete i rate di transizione è la seguente:

$$\boldsymbol{Q} = \begin{bmatrix} -w_{15} - w_3 & w_{15} & 0 & w_3 & 0 & 0 \\ w_2 & -w_{15} - w_2 - w_3 & w_{15} & 0 & 0 & w_3 \\ 0 & 2w_2 & -2w_2 & 0 & 0 & 0 \\ w_4 & 0 & 0 & -w_{15} - w_3 - w_4 & w_3 & w_{15} \\ 0 & 0 & 0 & w_4 & -w_4 & 0 \\ 0 & w_4 & 0 & w_2 & 0 & -w_2 - w_4 \end{bmatrix}$$

dove w_{15} è uguale alla somma tra w_1 e w_5.

Dato $w_1 = w_5 = 1/2$ e $w_2 = w_3 = w_4 = 1$, si risolve il seguente sistema lineare (2.1) ottenendo le seguenti sette equazioni:

$$\begin{cases} -2\pi_0 + \pi_1 + \pi_3 = 0 \\ \pi_0 - 3\pi_1 + 2\pi_2 + \pi_5 = 0 \\ \pi_1 - 2\pi_2 = 0 \\ \pi_0 - 3\pi_3 + \pi_4 + \pi_5 = 0 \\ \pi_3 - \pi_4 = 0 \\ \pi_1 + \pi_3 - 2\pi_5 = 0 \\ \pi_0 + \pi_1 + \pi_2 + \pi_3 + \pi_4 + \pi_5 = 1 \end{cases}$$

Le cui soluzioni della distribuzione stazionaria di probabilità di stato sono:

$$\begin{cases} \pi_0 = \pi_1 = \pi_3 = \pi_4 = \pi_5 = \frac{2}{11} \\ \pi_2 = \frac{1}{11} \end{cases}$$

2.3 Indici di prestazione

L'analisi dei sistemi modellati da reti di Petri stocastiche può essere spesso ricondotta all'analisi dei processi stocastici associati che rappresentano l'evoluzione nel tempo del sistema, utilizzando, in particolare, i processi stocastici di Markov (CTMC).

Per i processi Markoviani sono possibili due differenti analisi dell'evoluzione: l'analisi transiente relativa ad un limite temporale finito e l'analisi stazionaria che si riferisce al comportamento del sistema in condizioni di equilibrio. L'analisi transiente fornisce la distribuzione di probabilità di stato del processo dipendente dal tempo e dallo stato iniziale, mentre quella stazionaria fornisce la distribuzione di probabilità di stato indipendente dal tempo.

In generale, si effettua un'analisi stazionaria per una successiva valutazione degli indici di prestazione del sistema reale modellato. Tali indici rappresentano una caratterizzazione quantitativa delle prestazioni del sistema di interesse o di una sua componente.

Per ricavare tali indici è quindi basilare conoscere il concetto di distribuzione stazionaria di probabilità di stato π. Tale nozione è stata definita nel paragrafo 2.2.

La maggior parte di questi indici di prestazione può essere ricavata a partire dalla seguente formula [3]:

$$E[R] = \sum_{M_i \in RS(M_0)} r(M_i)\pi_i \tag{2.2}$$

dove π_i è la probabilità dello stato corrispondente alla marcatura M_i (marcatura raggiungibile da M_0) e $r(M_i)$ è una funzione definita *reward function* che associa una quantità intera alla probabilità di stato π_i.

2.3.1 Probabilità di una particolare condizione.

Data la condizione $\Upsilon(M)$ definiamo la funzione $r(M)$ che verifica se la condizione è soddisfatta a partire da una marcatura M.

$$r(M) = \begin{cases} 1 & \Upsilon(M) = true \\ 0 & altrimenti \end{cases}$$

Con la notazione $P\{\Upsilon\}$ indichiamo la probabilità che la condizione Υ sia soddisfatta, ovvero:

$$P\{\Upsilon\} = \sum_{M_i \in A} \pi_i \tag{2.3}$$

dove A è l'insieme di tutte le marcature che soddisfano la condizione Υ, ovvero $A = \{m_i \in RS(M_0) : \Upsilon(M_i) = true\}$.

Notiamo che tale definizione deriva dalla osservazione che le singole marcature sono mutuamente esclusive e quindi nella definizione (2.3) si considera la probabilità che la condizione Υ sia soddisfatta come la somma delle probabilità delle marcature singole.

2.3.2 Numero medio di token

Definiamo il sottoinsieme $A_{j,k} \in RS(M_0)$ come $A_{j,k} \triangleq \{M_i \in RS(M_0) : M_i(p_j) = k\}$. Tale insieme identifica tutte le marcature M_i raggiungibili da M_0 le quali presentano k token all'interno del posto p_j. Con la notazione $P\{A_{j,k}\}$ indichiamo la probabilità che venga soddisfatta la condizione $\Upsilon = A_{j,k}$.

Riformuliamo $r(M)$ come:

$$r(M) = k \;\; iff \;\; M(p_j) = k$$

dove $r(M)$ identifica il numero di token all'interno del posto p_j.

Ne consegue che il numero medio di token nel posto j (posto denotato da N_j) è quindi definito come segue [3]:

$$E[N_j] = \sum_k kP\{A_{j,k}\} \tag{2.4}$$

Il numero medio di token presenti nell'intera rete si può ottenere come segue:

Definiamo il sottoinsieme $A_k \in RS(M_0)$ come $A_k \triangleq \{M_i \in RS(M_0) : \sum_j M_i(p_j) = k\}$ con $0 \leq j \leq |P|$, che include tutte le marcature M_i raggiungibili da M_0 le quali presentano un numero totale di token pari a k. Con la notazione $P\{A_k\}$ indichiamo la probabilità che venga soddisfatta la condizione $\Upsilon = A_k$.

Riformuliamo $r(M)$ come:

$$r(M) = k \;\; iff \;\; \sum_j M(p_j) = k$$

dove $r(M)$ identifica il numero di token totale all'interno della marcatura M.

Il numero medio di token nella rete è quindi definito come segue:

$$E[N] = \sum_k kP\{A_k\} \tag{2.5}$$

2.3.3 Utilizzazione

Sia $\pi(0)$ la probabilità dell'evento rete vuota, ovvero la probabilità di essere nello stato in cui $\forall p_i \in P$ il numero di token prensenti è pari a zero, allora l'utilizzazione totale può essere calcolata considerando l'evento complementare:

$$U_{TOT}(PN) = 1 - \pi(0) \tag{2.6}$$

Di conseguenza l'utilizzazione del singolo posto j (posto denotato da N_j) è calcolata come segue:

$$U(N_j) = 1 - P\{\Upsilon\} \tag{2.7}$$

dove $P\{\Upsilon\}$ rappresenta la probabilità relativa all'insieme delle marcature M_i (raggiungibili da M_0) che non presentano token all'interno del posto $j - esimo$:

$$P\{\Upsilon\} = \sum_{M_i \in A} \pi_i \tag{2.8}$$

formalmente $A = \{M_i \in RS(M_0) : M_i(p_j) = 0\}$.

2.3.4 Throughtput

Il throughtput relativo ad una singola transizione t_j è definito come il numero medio di firing di quest'ultima nell'unità di tempo.

Definiamo $A_j = \{M_i \in RS(M_0) : t_j \in E(M_i)\}$ l'insieme delle marcature per le quali la transizione t_j è abilitata.

La funzione $r(M)$ è definita come segue:

$$r(M) = \begin{cases} w_j & t_j \in E(M) \\ 0 & altrimenti \end{cases}$$

Il throughtput della transizione t_j può essere calcolato dalla seguente formula [3]:

$$X_j = \sum_{M_i \in A_j} w_j \pi_i \tag{2.9}$$

Il throughtput totale della rete può essere calcolato come la somma dei throughtput delle transizioni che compongono la rete. Nel dettaglio tale indice è calcolato mediante la seguente formula:

$$X_{TOT} = \sum_{t_j \in T} X_j \tag{2.10}$$

dove T identifica l'insieme delle transizioni che compongono la rete.

2.3.5 Tempo medio di risposta

Il tempo medio impiegato da un token per attraversare una porzione di sottorete in condizione di regime, può essere calcolato usando la Legge di Little [6]:

$$E[T] = \frac{E[N]}{E[\lambda]} \tag{2.11}$$

dove $E[T]$ è il tempo medio impiegato dal token per attraversare la sottorete, $E[N]$ indica il numero medio di token che attraversano la sottorete ed infine $E[\lambda]$ denota la velocità media di ingresso dei token nella sottorete.

2.4 GSPN

Le reti di Petri stocastiche generalizzate rappresentano una generalizzazione delle reti di Petri stocastiche SPN ed ammettono transizioni immediate, archi inibitori e priorità relative alle transizioni.

Il firing di una transizione, potrebbe descrivere sia un'attività a cui si associa un determinato valore temporale, sia il verificarsi di una condizione logica. Per quel che concerne il primo caso risulta naturale utilizzare transizioni

temporizzate mentre, per il secondo caso risulta più appropriato l'utilizzo di transizioni immediate.

Definizione 24. (Reti di Petri Stocastiche Generalizzate) Una GSPN è una tupla definita nel seguente modo [18]:

$$GSPN = (P, T, I(\cdot,\cdot), O(\cdot,\cdot), H(\cdot,\cdot), \Pi(\cdot,\cdot), W(\cdot,\cdot), M_0)$$

dove:
$P = \{p_1, p_2, \ldots, p_n\}$ è un insieme finito di posti.
$T = \{t_1, t_2, \ldots, t_m\}$ è un insieme finito di transizioni.
$I(t_i, p_j) : T \times P \to \mathbb{N}$ è la funzione di input $1 \leq i \leq m$, $1 \leq j \leq n$.
$O(t_i, p_j) : T \times P \to \mathbb{N}$ è la funzione di output $1 \leq i \leq m$, $1 \leq j \leq n$.
$H(t_i, p_j) : T \times P \to \mathbb{N}$ è la funzione relativa agli archi inibitori $1 \leq i \leq m$, $1 \leq j \leq n$.
$\Pi(t_i) : T \to \mathbb{N}$ è una funzione che specifica la priorità della transizione t_i con $1 \leq i \leq m$.
$W(t_i, M) : T \times \mathbb{N}^n \to \mathbb{R}$ è una funzione che specifica: un tasso di firing dipendente dallo stato (per ogni transizione temporizzata e per ogni marcatura M) oppure un peso dipendente dallo stato (per ogni transizione immediata).
$M \in \mathbb{N}^n$ denota la marcatura (o stato) della rete, in particolare M_i rappresenta il numero di token nel posto p_i con $1 \leq i \leq n$.
M_0 rappresenta la marcatura iniziale.

Analogamente alle reti di Petri stocastiche, si possono definire il vettore di input, il vettore di output ed in aggiunta il vettore di inibizione:

- $I(t_i) = (i_1, \ldots, i_n)$ rappresenta il vettore di input e $i_j = I(t_i, p_j)$.
- $O(t_i) = (o_1, \ldots, o_n)$ rappresenta il vettore di output e $o_j = O(t_i, p_j)$.
- $H(t_i) = (h_1, \ldots, h_n)$ rappresenta il vettore di inibizione e $h_j = H(t_i, p_j)$.

La funzione $\Pi(t_i)$ associa una priorità alla transizione t_i. Le transizione immediate hanno priorità su tutte le transizioni temporizzate. A quest'ultime è associato il livello più basso di priorità (zero). Per le transizioni immediate si possono definire priorità maggiori o uguali a uno. In tal modo è possibile costruire gruppi di transizioni immediate con differenti livelli di priorità. Formalmente:

- Se $\Pi(t_i) = 0$ allora t_i è una transizione temporizzata, quindi, esegue il firing dopo un valore temporale che si distribuisce come una variabile casuale esponenziale negativa di media $1/w(t_i, M)$, dove M è la marcatura della rete.

- Se $\Pi(t_i) > 0$ allora t_i è una transizione immediata ed il suo tempo di firing è nullo.

Una transizione t_a è abilitata dalla marcatura M se:

- $M_i \geq I(t_a, p_i)$ per ogni $1 \leq i \leq n$. Ovvero, se tutti i posti in input di t_a contengono un numero di token almeno pari alla molteplicità dell'arco che collega il posto alla transizione.

- $M_i < H(t_a, p_i)$ per ogni $1 \leq i \leq n$. Ovvero, se tutti i posti collegati a t_a con arco inibitore contengono un numero di token inferiore al peso dell'arco che collega il posto alla transizione.

Quando una transizione immediata è abilitata dalla marcatura, tutte le transizioni temporizzate vengono disabilitate.

Come visto un precedenza per le reti PN nel Capitolo 1, il firing della transizione t_i cambia lo stato della rete (o marcatura) da M a $M + (O(t_i) - I(t_i))$.

La marcatura M viene definita *tangibile* se abilita solamente transizioni temporizzate, *evanescente* altrimenti. Il termine tangibile infatti indica che il tempo di soggiorno nelle marcatura è maggiore di zero, mentre il termine evenescente denota che il tempo di soggiorno nella marcatura è nullo.

Nel caso di marcatura tangibile, il meccanismo di scelta tra le transizioni abilitate dalla stessa, viene regolato a partire dai valori di delay associati alle transizioni come per una SPN. Invece, nel caso di marcature evanescenti, definiamo con T_α l'insieme delle transizioni immediate che sono abilitate dalla stessa marcatura M e con $p(t_i, M)$ la probabilità di firing relativa ad ogni transizione $t_i \in T_\alpha$:

$$p(t_i, M) = \frac{w(t_i, M)}{\sum_{t_j \in T_\alpha} w(t_j M)}$$

dove $w(t_i, M)$ indica una peso dipendente dallo stato.

Nel caso di marcature tangibili, può essere necessario asoociare alle varie transizioni più vettori di output, allo scopo di rappresentare, a partire da una singola transizione, differenti politiche di firing. Nello specifico, data una transizione t_i un possibile vettore di output può essere definito con $O_j(t_i)$. Ad ogni vettore di output si associa una probabilità d_{ij} ed inoltre $\sum_j d_{i,j} = 1$.

La rappresentazione grafica di una GSPN che rispetta le seguenti convenzioni:

- Le transizioni temporali sono rappresentate da rettangoli bianchi.

- Le transizioni immediate sono rappresentate da sottili rettangoli neri.

- I posti sono rappresentati da forme circolari.

Se $I(t_i, p_j) > 0$ allora si disegna una freccia da p_j a t_i etichettata dal valore $I(t_i, p_j)$. Se $O(t_i, p_j) > 0$ allora si disegna una freccia da t_i a p_j etichettata con il valore $O(t_i, p_j)$. Infine, se $H(t_i, p_j) > 0$ si disegna una linea continua con un cerchio all'estremità che collega p_j a t_j.

Una specifica marcatura M viene rappresentata graficamente disegnando m_j punti neri che rappresentano i token presenti nel posto p_j per ogni $j = 1, \ldots n$. Vengono omesse le etichette sugli archi nel caso in cui i loro pesi siano unitari.

2.4.1 Analisi GSPN

L'analisi delle reti GSPN consiste nel calcolare la probabilità stazionaria per ogni marcatura (stato) tangibile appartenente all'insieme di raggiungibilità a partire da una marcatura iniziale M_0.

Il processo stocastico associato ad un modello GSPN è un processo semi-Markoviano. Infatti, il tempo di soggiorno in un generico stato può essere sia distribuito esponenzialmente nel caso di marcature tangibili sia nullo nel caso di marcature evanescenti. Tuttavia, è possibile eliminare gli stati evanescenti con algoritmi efficenti, analizzando cosi una catena markoviana continua.

In una CTMC associata ad una GSPN i tempi di soggiorno relativi agli stati sono calcolati a partire dai ritardi medi delle transizioni.

2.4.2 GSPN-SPN in forma prodotto

L'analisi di una CTMC associata ad una GSPN o ad una SPN può risultare estremamente complessa a causa del problema dell'esplosione dello spazio degli stati, ovvero la possibilità per un rete di avere un insieme di raggiungibilità con cardinalità infinita. In tal caso è impossibile calcolare la probabilità stazionaria di stato utilizzando il solo sistema lineare (2.1) descritto nel paragrafo 2.2.

Per far fronte a tale problema, è possibile evitare il calcolo della soluzione esatta della distribuzione ed accontentarsi di approssimazioni o di risultati forniti da simulazioni. Un'altra possibilità, consiste nel focalizzarsi su un'insieme di sottoclassi per le quali è possibile ottenere una soluzione in forma prodotto. Nel dettaglio, tale approccio consiste nell'esprimere la probabilità stazionaria di uno stato globale come un prodotto di probabilità dipendenti solamente da stati locali e divisi da una costante di normalizzazione. Ogni stato locale dipende quindi dal numero di token presenti in ogni singolo posto. La costante di normalizzazione è calcolata in modo da garantire che le probabilità siano consistenti con la definizione di densità di probabilità, ovvero che la somma di tutte le probabilità di stato sia pari ad uno.

Storicamente sono stati intrapresi più studi dedicati alla ricerca delle condizioni per ottenere reti in forma prodotto [15], [16]. Un importante

risultato è rappresentato dalla classe di modelli in forma prodotto Coleman, Henderson et al. [2].

Le condizioni per ottenere quanto descritto da questa tipologia di classe dipendono dalla configurazione complessiva della rete (struttura e tassi delle transizioni), percui non permettono un approccio gerarchico alle definizioni del modello.

Ci baseremo sul teorema di Coleman, Henderson et al. per costruire il simulatore per reti di Petri stocastiche in forma prodotto.

2.5 Teorema di Coleman, Henderson et al. (Forma Prodotto)

Il teorema viene introdotto in due fasi [2]. Dapprima vengono analizzate le condizioni medianti le quali può essere applicato e successivamente viene illustrato il teorema principale.

Condizioni:

- **C1:** Il teorema viene definito per modelli SPN (senza transizioni immediate o archi inibitori), ed utilizza vettori di probabilità relativi all'output delle transizioni e tassi di firing dipendenti dallo stato. Le transizioni sono tutte temporizzate quindi formalmente: $\prod(t_i) = 0$ per ogni $t_i \in T$. Non vi sono archi inibitori e quindi $H(t_i, p_j) = 0$ per ogni $t_i \in T, p_j \in P$ dove T è l'insieme delle transizioni e P l'insieme dei posti.

- **C2:** Il tasso di firing di una transizione deve essere nella seguente forma:
$$w(t_i, M) = \frac{\psi(M - I(t_i))\chi_i}{\phi(M)} \tag{2.12}$$
dove χ_i è una costante non negativa che dipende solo dalla transizione t_i, ψ è una funzione non negativa e ϕ è una funzione positiva. Si può modellare una larga classe di tassi di firing (load-dependent). La transizione $t_i \in T$ è abilitata se e solo se $\psi(M - I(t_i)) > 0$.

- **C3:** Non ci possono essere due transizioni con lo stesso vettore di input. Spesso si può trasformare una SPN che non soddisfa tale condizione in una nuova SPN che la soddisfa, fondendo le transizioni con lo stesso vettore di input in un'unica transizione ed introducendo vettori di probabilità di output. Tuttavia, tale procedura non può essere applicata se le transizioni con lo stesso vettore di input dipendono dallo stato della rete in modo differente (ψ e ϕ dipendono solo dallo stato della rete e dal vettore di input della transizione).

- **C4:** Sia B_i il numero di vettori di output relativi alla transizione t_i, allora abbiamo che $\sum_{j=1}^{B_i} d_{ij} = 1$.
  Per ogni transizione $t_i \in T$ e per ogni $O_j(t_i)$ con $1 \leq j \leq B_i$ deve esistere esattamente una transizione $t_s \in T$ tale che:

  $$O_j(t_i) = I\left(t_s\right)$$

  Indichiamo con $E_j(t_i)$ la transizione t_s, e con $p\left(t_i, t_s\right)$ la probabilità relativa al vettore di output, pari a d_{ij}. Viceversa per ogni transizione t_s vi è almeno un vettore di output j relativo alla transizione $t_i \in T$ tale che $I\left(t_s\right) = O_j\left(t_i\right)$ per $1 \leq j \leq B_i$.

- **C5:** Il seguente sistema di equazioni di traffico con $1 \leq i, s \leq N$, $N = |T|$:

  $$\chi_i f_i = \sum_s \chi_s f_s p(t_s, t_i) \tag{2.13}$$

  deve avere una soluzione $\boldsymbol{f} = (f_1, \ldots, f_N)$ sotto la condizione $f_i > 0$ per ogni $t_i \in T$.
  Si può dimostrare che se f e f' sono soluzioni del sistema allora esiste $k > 0$ tale che $f = kf'$, quindi le soluzioni differiscono di una costante.
  Definiamo il vettore C come segue:

  $$C = \begin{bmatrix} log\left(\frac{f_1}{f_{E_1}(t_i)}\right) \\ \cdots \\ log\left(\frac{f_1}{f_{E_{B_1}}(t_i)}\right) \\ \cdots \\ log\left(\frac{f_N}{f_{E_1}(t_N)}\right) \\ \cdots \\ log\left(\frac{f_N}{f_{E_{B_N}}(t_N)}\right) \end{bmatrix} \tag{2.14}$$

  tale vettore è indipendente dalla particolare soluzione f del sistema dato che tutte le soluzioni differiscono di una costante positiva.

Teorema. ***(Coleman, Henderson et al.).*** *Sia S una rete di Petri stocastica che soddisfa le condizioni C1, C2, C3, C4 e C5. Indichiamo con $P = |P|$, $N = |T|$, A la matrice di incidenza della rete S, C il vettore colonna definito in (2.14) e con ϕ la funzione che appare in (2.1). Allora S ha la seguente soluzione in forma prodotto* [2]*:*

$$\pi\left(M\right) = \phi\left(M\right) g\left(M\right)$$

dove $g\left(M\right) = \prod_{i=1}^{P} y_i^{M_i}$.

se e solo se

$$rango\,(A) = rango\,([A;C])$$

dove $[A;C]$ *indica la matrice ottenuta aggiungendo alla matrice* A *la colonna* C*. In questo caso possiamo ottenere le soluzioni* y_i *per* $i = 1, \ldots, M$ *risolvendo il sistema:*

$$-A \begin{bmatrix} log\,(y_1) \\ \cdots \\ \cdots \\ log\,(y_M) \end{bmatrix} = C \tag{2.15}$$

Ricordiamo che il rango di una matrice A equivale al massimo ordine di una matrice quadrata estraibile da A avente determinante diverso da zero.

In altre parole il rango di A è r se esiste una sottomatrice di ordine r con determinante non nullo ed inoltre tutte le possibili sottomatrici quadrate avente ordine maggiore hanno determinante nullo.

Il rango $r\,(A)$ può anche essere definito come il massimo numero di righe (o colonne) linearmente indipendenti della matrice A.

Esempio 25. Consideriamo una rete di Petri stocastica $SPN = (P, T, I, O, W)$ con $n = 5$ e $m = 6$ illustrata in Figura 2.6.

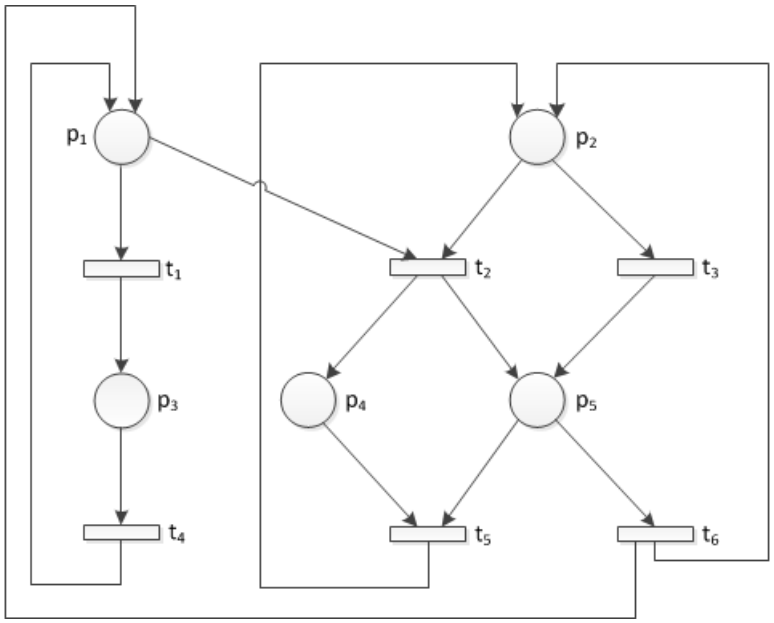

Figura 2.6: Primo esempio di SPN esprimibile in forma prodotto

Le matrici I ed O sono definite come segue:

$$I = \begin{bmatrix} & p_1 & p_2 & p_3 & p_4 & p_5 \\ t_1 & 1 & 0 & 0 & 0 & 0 \\ t_2 & 1 & 1 & 0 & 0 & 0 \\ t_3 & 0 & 1 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 1 & 0 & 0 \\ t_5 & 0 & 0 & 0 & 1 & 1 \\ t_6 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad O = \begin{bmatrix} & p_1 & p_2 & p_3 & p_4 & p_5 \\ t_1 & 0 & 0 & 1 & 0 & 0 \\ t_2 & 0 & 0 & 0 & 1 & 1 \\ t_3 & 0 & 0 & 0 & 0 & 1 \\ t_4 & 1 & 0 & 0 & 0 & 0 \\ t_5 & 0 & 1 & 0 & 0 & 0 \\ t_6 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Da cui si ottiene la seguente matrice di incidenza A:

$$A = \begin{bmatrix} & p_1 & p_2 & p_3 & p_4 & p_5 \\ t_1 & -1 & 0 & 1 & 0 & 0 \\ t_2 & -1 & -1 & 0 & 1 & 1 \\ t_3 & 0 & -1 & 0 & 0 & 1 \\ t_4 & 1 & 0 & -1 & 0 & 0 \\ t_5 & 0 & 1 & 0 & -1 & -1 \\ t_6 & 1 & 1 & 0 & 0 & -1 \end{bmatrix}$$

La rete non presenta transizioni immediate nè archi inibitori. Tutte le transizioni sono temporizzate e non vi sono due transizioni con vettore di input uguali. Per ogni transizione $t_i \in T$ esiste esattamente una transizione

$t_s \in T$ tale che il vettore di output di t_i è uguale a t_s. Sono quindi soddisfatte le condizioni C1, C2, C3, C4 e C5.

Data la notazione $t_s = E_j(t_i)$:

dato $O_1(t_1) = I(t_4)$ allora $E_1(t_1) = t_4$ con $p(t_1, t_4) = 1$.
dato $O_1(t_2) = I(t_5)$ allora $E_1(t_2) = t_5$ con $p(t_2, t_5) = 1$.
dato $O_1(t_3) = I(t_6)$ allora $E_1(t_3) = t_6$ con $p(t_3, t_6) = 1$.
dato $O_1(t_4) = I(t_1)$ allora $E_1(t_4) = t_1$ con $p(t_4, t_1) = 1$.
dato $O_1(t_5) = I(t_3)$ allora $E_1(t_5) = t_3$ con $p(t_5, t_3) = 1$.
dato $O_1(t_6) = I(t_2)$ allora $E_1(t_6) = t_2$ con $p(t_6, t_2) = 1$.

Risolviamo il sistema di equazioni di traffico (2.13), ponendo $x(i) = \chi_i f_i$.

$x(1) = x(1)\,p(t_1,t_1) + x(2)\,p(t_2,t_1) + x(3)\,p(t_3,t_1) + x(4)\,p(t_4,t_1) + x(5)\,p(t_5,t_1) + x(6)\,p(t_6,t_1) = x(4)\,p(t_4,t_1)$

$x(2) = x(1)\,p(t_1,t_2) + x(2)\,p(t_2,t_2) + x(3)\,p(t_3,t_2) + x(4)\,p(t_4,t_2) + x(5)\,p(t_5,t_2) + x(6)\,p(t_6,t_2) = x(6)\,p(t_6,t_2)$

$x(3) = x(1)\,p(t_1,t_3) + x(2)\,p(t_2,t_3) + x(3)\,p(t_3,t_3) + x(4)\,p(t_4,t_3) + x(5)\,p(t_5,t_3) + x(6)\,p(t_6,t_3) = x(5)\,p(t_5,t_3)$

$x(4) = x(1)\,p(t_1,t_4) + x(2)\,p(t_2,t_4) + x(3)\,p(t_3,t_4) + x(4)\,p(t_4,t_4) + x(5)\,p(t_5,t_4) + x(6)\,p(t_6,t_4) = x(1)\,p(t_1,t_4)$

$x(5) = x(1)\,p(t_1,t_5) + x(2)\,p(t_2,t_5) + x(3)\,p(t_3,t_5) + x(4)\,p(t_4,t_5) + x(5)\,p(t_5,t_4) = x(6)\,p(t_6,t_5) = x(2)\,p(t_2,t_5)$

$x(6) = x(1)\,p(t_1,t_6) + x(2)\,p(t_2,t_6) + x(3)\,p(t_3,t_6) + x(4)\,p(t_4,t_6) + x(5)\,p(t_5,t_6) = x(6)\,p(t_6,t_6) = x(3)\,p(t_3,t_6)$

Quindi:

$\chi_1 f_1 = \chi_4 f_4$
$\chi_2 f_2 = \chi_6 f_6$
$\chi_3 f_3 = \chi_5 f_5$
$\chi_4 f_4 = \chi_1 f_1$
$\chi_5 f_5 = \chi_2 f_2$
$\chi_6 f_6 = \chi_3 f_3$

Da cui:

$$\chi_1 f_1 = \chi_4 f_4$$

$$\chi_2 f_2 = \chi_3 f_3 = \chi_5 f_5 = \chi_6 f_6$$

Definiamo il vettore C come indicato in (2.14):

$$C = \begin{bmatrix} \log\left(\frac{f_1}{f_{E_1(t_1)}}\right) \\ \log\left(\frac{f_2}{f_{E_1(t_2)}}\right) \\ \log\left(\frac{f_3}{f_{E_1(t_3)}}\right) \\ \log\left(\frac{f_4}{f_{E_1(t_4)}}\right) \\ \log\left(\frac{f_5}{f_{E_1(t_5)}}\right) \\ \log\left(\frac{f_6}{f_{E_1(t_6)}}\right) \end{bmatrix} = \begin{bmatrix} \log\left(\frac{f_1}{f_4}\right) \\ \log\left(\frac{f_2}{f_5}\right) \\ \log\left(\frac{f_3}{f_6}\right) \\ \log\left(\frac{f_4}{f_1}\right) \\ \log\left(\frac{f_5}{f_3}\right) \\ \log\left(\frac{f_6}{f_2}\right) \end{bmatrix} = \begin{bmatrix} \log\left(\frac{\chi_4}{\chi_1}\right) \\ \log\left(\frac{\chi_5}{\chi_2}\right) \\ \log\left(\frac{\chi_6}{\chi_3}\right) \\ \log\left(\frac{\chi_1}{\chi_4}\right) \\ \log\left(\frac{\chi_3}{\chi_5}\right) \\ \log\left(\frac{\chi_2}{\chi_6}\right) \end{bmatrix}$$

Calcoliamo il rango della matrice $[A]$ e della matrice $[A|C]$:

$$rango \begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & -1 \end{bmatrix} = 4$$

$$rango \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & \log\left(\frac{\chi_4}{\chi_1}\right) \\ -1 & -1 & 0 & 1 & 1 & \log\left(\frac{\chi_5}{\chi_2}\right) \\ 0 & -1 & 0 & 0 & 1 & \log\left(\frac{\chi_6}{\chi_3}\right) \\ 1 & 0 & -1 & 0 & 0 & \log\left(\frac{\chi_1}{\chi_4}\right) \\ 0 & 1 & 0 & -1 & -1 & \log\left(\frac{\chi_3}{\chi_5}\right) \\ 1 & 1 & 0 & 0 & -1 & \log\left(\frac{\chi_2}{\chi_6}\right) \end{bmatrix}$$

Le condizioni affinchè il $rango\,[A|C]$ sia uguale a 4 e quindi equivalente al $rango\,[A]$ sono:

$$C_1 + C_4 = 0$$

$$C_2 + C_3 + C_5 + C_6 = 0$$

Dato che tali condizioni sono sempre soddisfatte allora calcoliamo le soluzioni y_i per $i = 1, \ldots, M$ mediante la risoluzione del sistema (2.15):

$$-\begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \log(y_1) \\ \log(y_2) \\ \log(y_3) \\ \log(y_4) \\ \log(y_5) \end{bmatrix} = \begin{bmatrix} \log\left(\frac{\chi_4}{\chi_1}\right) \\ \log\left(\frac{\chi_5}{\chi_2}\right) \\ \log\left(\frac{\chi_6}{\chi_3}\right) \\ \log\left(\frac{\chi_1}{\chi_4}\right) \\ \log\left(\frac{\chi_3}{\chi_5}\right) \\ \log\left(\frac{\chi_2}{\chi_6}\right) \end{bmatrix}$$

$$\begin{cases} \log(y_1) - \log(y_3) = \log\left(\frac{\chi_4}{\chi_1}\right) \\ \log(y_1) + \log(y_2) - \log(y_4) - \log(y_5) = \log\left(\frac{\chi_5}{\chi_2}\right) \\ \log(y_2) - \log(y_5) = \log\left(\frac{\chi_6}{\chi_3}\right) \\ -\log(y_1) + \log(y_3) = \log\left(\frac{\chi_1}{\chi_4}\right) \\ -\log(y_2) + \log(y_4) + \log(y_5) = \log\left(\frac{\chi_3}{\chi_5}\right) \\ -\log(y_1) - \log(y_2) + \log(y_5) = \log\left(\frac{\chi_2}{\chi_6}\right) \end{cases}$$

Preso $y_2 = 1$, la soluzione del sistema è la seguente:
$y_1 = \frac{\chi_3}{\chi_2}$
$y_2 = 1$
$y_3 = \frac{\chi_1\chi_3}{\chi_2\chi_4}$
$y_4 = \frac{\chi_6}{\chi_5}$
$y_5 = \frac{\chi_3}{\chi_6}$

Allora la rete di Petri stocastica in Figura 2.6 ha la seguente soluzione in forma prodotto:

$$g(M) = \left[\frac{\chi_3}{\chi_2}\right]^{M_1} \left[\frac{\chi_1\chi_3}{\chi_2\chi_4}\right]^{M_3} \left[\frac{\chi_6}{\chi_5}\right]^{M_4} \left[\frac{\chi_3}{\chi_6}\right]^{M_5}$$

Esempio 26. Consideriamo la rete di Petri stocastica $SPN = (P, T, I, O, W)$ con $n = 5$ e $m = 6$ illustrata in Figura 2.7.

Tale rete presenta due vettori di output per quel che concerne la transizione t_1. Ciascun vettore identifica un possibile firing della transizione stessa. Il primo viene rappresentato graficamente mediante archi con linea continua e permette un passaggio di token ai posti p_2 e p_3. Il secondo identificato da archi con linee tratteggiate permette un passaggio di token verso i posti p_3 e p_4. Le probabilità associate alle due modalità di firing sono rispettivamente $1 - b$ e b. Con $0 \leq b \leq 1$.

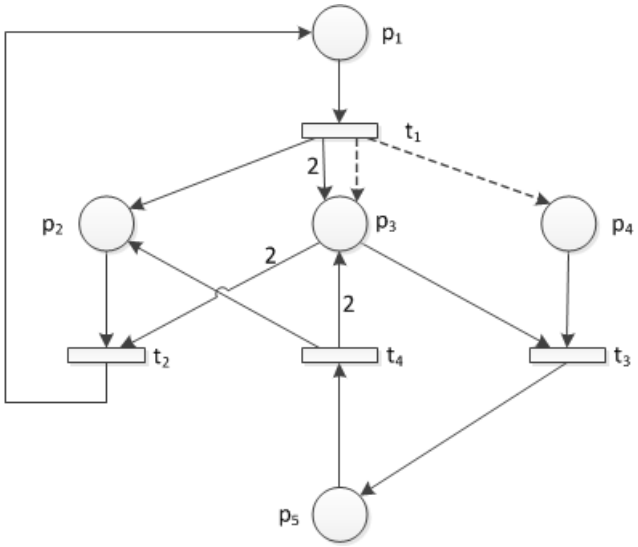

Figura 2.7: Rete di Petri stocastica con due possibili firing della transizione t_1

Le matrici I ed O sono definite come segue:

$$I = \begin{array}{c} \\ t_{11} \\ t_{12} \\ t_2 \\ t_3 \\ t_4 \end{array} \begin{array}{c} \begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \qquad O = \begin{array}{c} \\ t_{11} \\ t_{12} \\ t_2 \\ t_3 \\ t_4 \end{array} \begin{array}{c} \begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 \end{bmatrix} \end{array}$$

Da cui si calcola la seguente matrice di incidenza A:

$$A = \begin{array}{c} \\ t_{11} \\ t_{12} \\ t_2 \\ t_3 \\ t_4 \end{array} \begin{array}{c} \begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \end{array} \\ \begin{bmatrix} -1 & 1 & 2 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 1 & -1 & -2 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 \\ 0 & 1 & 2 & 0 & -1 \end{bmatrix} \end{array}$$

Tutte le condizioni nel paragrafo 2.5 vengono soddisfatte. Inoltre:
dato $O_1(t_1) = I(t_2)$ allora $E_1(t_1) = t_2$ con $p(t_1, t_2) = 1 - b$.
dato $O_2(t_1) = I(t_3)$ allora $E_2(t_1) = t_3$ con $p(t_1, t_3) = b$.
dato $O_1(t_2) = I(t_1)$ allora $E_1(t_2) = t_1$ con $p(t_2, t_1) = 1$.
dato $O_1(t_3) = I(t_4)$ allora $E_1(t_3) = t_4$ con $p(t_3, t_4) = 1$.
dato $O_1(t_4) = I(t_2)$ allora $E_1(t_4) = t_2$ con $p(t_4, t_2) = 1$.

Risolviamo il sistema di equazioni di traffico (2.13) ottenendo le seguenti soluzioni:

$$\chi_1 f_1 = \chi_2 f_2$$

$$\chi_4 f_4 = \chi_3 f_3$$

Definiamo il vettore C come indicato in (2.14):

$$C = \begin{bmatrix} \log\left(\frac{f_1}{f_{E_1(t_1)}}\right) \\ \log\left(\frac{f_1}{f_{E_2(t_2)}}\right) \\ \log\left(\frac{f_2}{f_{E_1(t_2)}}\right) \\ \log\left(\frac{f_3}{f_{E_1(t_3)}}\right) \\ \log\left(\frac{f_4}{f_{E_1(t_4)}}\right) \end{bmatrix} = \begin{bmatrix} \log\left(\frac{f_1}{f_2}\right) \\ \log\left(\frac{f_1}{f_3}\right) \\ \log\left(\frac{f_2}{f_1}\right) \\ \log\left(\frac{f_3}{f_4}\right) \\ \log\left(\frac{f_4}{f_2}\right) \end{bmatrix} = \begin{bmatrix} \log\left(\frac{\chi_2}{\chi_1}\right) \\ \log\left(\frac{\chi_3}{b\chi_1}\right) \\ \log\left(\frac{\chi_1}{\chi_2}\right) \\ \log\left(\frac{\chi_4}{\chi_3}\right) \\ \log\left(\frac{b\chi_2}{\chi_4}\right) \end{bmatrix}$$

Il rango della matrice $[A]$ è pari a 3. Le condizioni affinchè il $rango\,[A|C]$ sia equivalente al $rango\,[A]$ sono:

$$C_1 + C_3 = 0$$

$$C_2 + C_4 + C_5 - C_1 = 0$$

Dato che tali condizioni sono sempre soddisfatte allora calcoliamo le soluzioni y_i per $i = 1, \ldots, M$ mediante la risoluzione del sistema (2.15). Presi $y_1 = 1$ e $y_3 = 1$, la soluzione del sistema è la seguente:
$y_1 = 1$
$y_2 = \frac{\chi_1}{\chi_2}$
$y_3 = 1$
$y_4 = \frac{b\chi_1}{\chi_3}$
$y_5 = \frac{b\chi_1}{\chi_4}$

Allora la rete di Petri stocastica in Figura 2.7 ha la seguente soluzione in forma prodotto:

$$g(M) = \left[\frac{\chi_1}{\chi_2}\right]^{M_2} \left[\frac{b\chi_1}{\chi_3}\right]^{M_4} \left[\frac{b\chi_1}{\chi_4}\right]^{M_5}$$

Esempio 27. Consideriamo una rete di Petri stocastica $SPN = (P, T, I, O, W)$ con $n = 2$ e $m = 4$ illustrata in Figura 2.8.

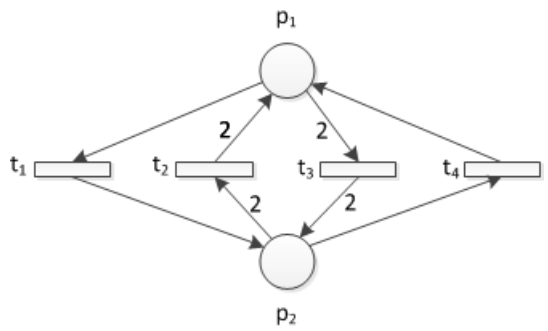

Figura 2.8: Rete di petri stocastica con vincolo sul rango delle matrici

Le matrici I, O ed A sono definite come segue:

$$I = \begin{bmatrix} & p_1 & p_2 \\ t_1 & 1 & 0 \\ t_2 & 0 & 2 \\ t_3 & 2 & 0 \\ t_4 & 0 & 1 \end{bmatrix} \quad O = \begin{bmatrix} & p_1 & p_2 \\ t_1 & 0 & 1 \\ t_2 & 2 & 0 \\ t_3 & 0 & 2 \\ t_4 & 1 & 0 \end{bmatrix} \quad A = \begin{bmatrix} & p_1 & p_2 \\ t_1 & -1 & 1 \\ t_2 & 2 & -2 \\ t_3 & -2 & 2 \\ t_4 & 1 & -1 \end{bmatrix}$$

Tutte le condizioni nel paragrafo 2.5 vengono soddisfatte. Inoltre:
dato $O_1(t_1) = I(t_4)$ allora scriviamo $E_1(t_1) = t_4$ con $p(t_1, t_4) = 1$.
dato $O_1(t_2) = I(t_3)$ allora scriviamo $E_1(t_2) = t_3$ con $p(t_2, t_3) = 1$.
dato $O_1(t_3) = I(t_2)$ allora scriviamo $E_1(t_3) = t_2$ con $p(t_3, t_2) = 1$.
dato $O_1(t_4) = I(t_1)$ allora scriviamo $E_1(t_4) = t_1$ con $p(t_4, t_1) = 1$.

Risolviamo il sistema di equazioni di traffico (2.13), ottenendo:

$$\chi_1 f_1 = \chi_4 f_4$$

$$\chi_2 f_2 = \chi_3 f_3$$

Definiamo il vettore C come indicato in (2.14):

$$C = \begin{bmatrix} \log\left(\frac{f_1}{f_{E_1(t_1)}}\right) \\ \log\left(\frac{f_2}{f_{E_1(t_2)}}\right) \\ \log\left(\frac{f_3}{f_{E_1(t_3)}}\right) \\ \log\left(\frac{f_4}{f_{E_1(t_4)}}\right) \end{bmatrix} = \begin{bmatrix} \log\left(\frac{f_1}{f_4}\right) \\ \log\left(\frac{f_2}{f_3}\right) \\ \log\left(\frac{f_3}{f_2}\right) \\ \log\left(\frac{f_4}{f_1}\right) \end{bmatrix} = \begin{bmatrix} \log\left(\frac{\chi_4}{\chi_1}\right) \\ \log\left(\frac{\chi_3}{\chi_2}\right) \\ \log\left(\frac{\chi_2}{\chi_3}\right) \\ \log\left(\frac{\chi_1}{\chi_4}\right) \end{bmatrix}$$

Calcoliamo il rango della matrice $[A]$ e della matrice $[A|C]$:

$$rango \begin{bmatrix} -1 & 1 \\ 2 & -2 \\ -2 & 2 \\ 1 & -1 \end{bmatrix} = 1$$

$$rango \begin{bmatrix} -1 & 1 & \log\left(\frac{\chi_4}{\chi_1}\right) \\ 2 & -2 & \log\left(\frac{\chi_3}{\chi_2}\right) \\ -2 & 2 & \log\left(\frac{\chi_2}{\chi_3}\right) \\ 1 & -1 & \log\left(\frac{\chi_1}{\chi_4}\right) \end{bmatrix}$$

Le condizioni affinchè il $rango\,[A|C]$ sia uguale a 1 e quindi equivalente al $rango\,[A]$ sono:

$$C_1 + C_4 = 0$$

$$C_2 + 2C_1 = 0$$

$$C_3 - 2C_1 = 0$$

La prima condizione è sempre soddisfatta. La seconda e la terza condizione sono equivalenti e possono essere riscritte nel seguente modo:

$$\frac{\chi_2}{\chi_3} = \left(\frac{\chi_4}{\chi_1}\right)^2$$

Calcoliamo le soluzioni y_i per $i = 1, \ldots, M$ mediante la risoluzione del sistema (2.15). Preso $y_2 = 1$, la soluzione del sistema è la seguente:
$y_1 = \frac{\chi_4}{\chi_1}$ oppure $y_1 = \left(\frac{\chi_2}{\chi_3}\right)^{1/2}$
$y_2 = 1$

Allora la rete di Petri stocastica in Figura 2.8 ha la seguente soluzione in forma prodotto:

$$g\,(M) = \left[\frac{\chi_4}{\chi_1}\right]^{M_1}$$

oppure:

$$g\,(M) = \left[\frac{\chi_2}{\chi_3}\right]^{M_1/2}$$

Parte III

Simulazione

Capitolo 3

Criteri di Arresto di simulazione

3.1 Introduzione

Consideriamo una rete di Petri stocastica che soddisfi particolari condizioni imposte dal teorema di Coleman Henderson et al. come descritto nel paragrafo 2.5 e la cui soluzione può essere quindi espressa in forma prodotto. Uno dei problemi riscontrabili nella soluzione dei modelli in forma prodotto Coleman Henderson et al. consiste nel calcolo della costante di normalizzazione G. Esistono due differenti algoritmi per il calcolo delle probabilità di stato di una SPN in forma prodotto e dei relativi indici di prestazione: algoritmi di convoluzione e MVA (Mean Value Analysis). Tuttavia tali algoritmi possono essere applicati solamente ad un limitato sottoinsieme di reti di Petri stocastiche in forma prodotto.

Tali classi di reti SPN devono soddisfare i due seguenti vincoli. Il primo vincolo impone che la funzione $\phi(M)$ (introdotta nel paragrafo 2.5) sia definita nella seguente forma:

$$\phi(M) = \prod_{i=1}^{M} \phi_i(M_i) \tag{3.1}$$

dove $M \in RS(M_0)$.

In tal caso si deduce che le probabilità stazionarie di stato possono essere calcolate mediante la seguente formula:

$$\pi(M) = G \prod_{i=1}^{M} \left[\phi_i(M_i) \rho_i^{M_i} \right] \tag{3.2}$$

dove G rappresenta la costante di normalizzazione e $\phi_i(0) = 1$.

Verificare il vincolo (3.1) è un'operazione semplice.

La verifica del secondo vincolo (3.2), invece, risulta essere piuttosto costosa perchè richiede l'analisi dell'insieme di raggiungibilità della SPN. Dato uno stato iniziale M_0 e detto Y un vettore S-invariante, allora uno stato M è raggiungibile se e solo se $YM = YM_0$. In generale, per verificare tale condizione è necessaria la generazione dell'insieme di raggiungibilità relativo alla SPN. Tale procedimento si ottiene banalmente per alcune tipologie di SPN quali le reti *marked graphs* (in cui ogni posto della rete è caratterizzato al massimo da un arco in input e da un arco in output) e reti *state machine* (SPN in cui ogni transizione ha al massimo un arco entrante ed un arco uscente). In altre tipologie di SPN la generazione dell'insieme di raggiungibilità può risultare complesso e talvolta impossibile [1].

Data la difficoltà nel calcolo della costante G, in questa sezione vengono proposte alcune osservazioni che si basano inizialmente sull'analisi e successivamente sulla definizione di alcune grandezze generate calcolando differenti rapporti tra le probabilità stazionarie analitiche e le probabilità stazionarie stimate relative ad un insieme finito di stati raggiungibili dalla SPN oggetto di studio. Tali rapporti risultano essere interessanti in quanto permettono di ottenere un buon livello di correttezza relativo alla stima delle probabilità offerte da PFNetS, ignorarndo nel calcolo delle grandezze stesse la complessa determinazione della costante di normalizzazione G.

Inoltre, utilizzando i rapporti, definiamo in seguito due differenti criteri di arresto della simulazione implementati entrambi mediante due differenti algoritmi, chiamati *metodo della matrice* e *metodo del vettore*. Tali algoritmi permettono di terminare la simulazione della SPN analizzata dopo un numero finito di passi di simulazione, garantendo in tal modo un limitato costo computazionale relativo alla fase di analisi.

Il simulatore PFNetS definito nell'elaborato del collega Massimo Mariuzzo applica il metodo delle prove ripetute, ovvero esegue una serie di run i.i.d., ciascuna di durata sensibilmente diversa per l'analisi dei risultati ottenuti.

3.2 Primo Criterio: Metodo della matrice

3.2.1 Formalizzazione del criterio.

Data una qualsiasi rete SPN, l'algoritmo proposto per l'implementazione del metodo della matrice descritto in questa sezione si sviluppa nei seguenti passi:

Passo 1: Costruzione di un insieme di un numero arbitrario di marcature della SPN caratterizzate dai più alti valori di probabilità stazionaria analitica (a meno di una costante di normalizzazione G) raggiungibili dalla marcatura iniziale M_0. Tale insieme è denotato con W. Il procedimento utilizzato per la costruzione dell'insieme W è descritto nel dettaglio nel paragrafo 3.2.3. Definiamo con $\boldsymbol{g}$ il vettore delle probabilità analitiche a meno della costante di normalizzazione G relativo alle marcature appartenenti al-

l'insieme W. Formalmente $\boldsymbol{g}(i)$ indica la $i-esima$ componente del vettore con $1 \leq i \leq |W|$.

Osservazione. Per ogni coppia di marcature appartenenti all'insieme W, e quindi $\forall i, j$ con $1 \leq i, j \leq |W|$ indichiamo con:

- $\widetilde{\pi}_i$ e $\widetilde{\pi}_j$ le probabilità stimate dal simulatore PFNetS e relative alle marcature M_i ed M_j appartenenti all'insieme W.
- $\dot{\pi}_i$ e $\dot{\pi}_j$le probabilità analitiche relative alle marcature M_i ed M_j appartenenti all'insieme W.

Se la stima delle probabilità delle marcature ottenute dalla simulazione è accurata, possiamo assumere che:

$$\frac{\widetilde{\pi}_i}{\widetilde{\pi}_j} \backsimeq \frac{\dot{\pi}_i}{\dot{\pi}_j}$$

$\forall i, j$ con $1 \leq i, j \leq |W|$

ovvero per una qualsiasi coppia di marcature dell'insieme W, il rapporto tra le loro probabilià stimate è circa uguale, a meno di un errore di stima, al rapporto tra le corrispondenti probabilità analitiche.

Sulla base di quanto descritto nel paragrafo 2.5 ogni tipologia di SPN che rispetta le condizioni imposte dal teorema di Coleman Henderson et al. ha una soluzione in forma prodotto. In tali reti si ottiene quindi che per ogni marcatura i appartenente a W la rispettiva probabilità analitica $\dot{\pi}_i$ può essere approssimata ad un valore $g_i = \prod_{i=1}^{P} \rho_i^{M_i}$ a meno della costante di normalizzazione G.

$$\dot{\pi}_i \propto g_i$$

Si ottiene quindi:

$$\frac{\widetilde{\pi}_i}{\widetilde{\pi}_j} \backsimeq \frac{G \cdot g_i}{G \cdot g_j}$$

Esplicitando il valore $\dot{\pi}_i$ con $G \cdot g_i$ ed il valore $\dot{\pi}_j$ con $G \cdot g_j$ evitiamo di considerare nel calcolo del rapporto la costante di normalizazzione G.

Si ottiene quindi:

$$\frac{\widetilde{\pi}_i}{\widetilde{\pi}_j} \backsimeq \frac{g_i}{g_j}$$

che equivale a:

$$\frac{\widetilde{\pi}_i \cdot g_j}{\widetilde{\pi}_j \cdot g_i} \backsimeq 1$$

Passo 2: Sulla base di quanto ottenuto dall'osservazione, generiamo ad intervalli regolari di passi di simulazione, dapprima un vettore di stime di probabilità di marcature denotato con $\widetilde{\boldsymbol{\pi}}$ ed utilizzando tale vettore, costruiamo successivamente una matrice indicata con Z.

Il vettore $\widetilde{\boldsymbol{\pi}}$ rappresenta il vettore delle probabilità stimate dal simulatore PFNetS nel periodo di osservazione simulato, dove $\widetilde{\boldsymbol{\pi}}(i)$ indica la $i-esima$ componente del vettore con $1 \leq i \leq |W|$.

Definiamo la matrice Z, quadrata di ordine $|W|$ i cui elementi z_{ij} sono calcolati come segue:

$$z_{ij} = \begin{cases} 1 & se\, i = j \\ \frac{\widetilde{\boldsymbol{\pi}}(i) \cdot \boldsymbol{g}(j)}{\widetilde{\boldsymbol{\pi}}(j) \cdot \boldsymbol{g}(i)} & altrimenti \end{cases}$$

Passo 3: Il criterio di arresto proposto consiste nel terminare il run di simulazione al passo di simulazione quando si verifica la condizione:

$$\forall z_{ij} \in Z : |1 - z_{ij}| \leq \xi \tag{3.3}$$

dove ξ indica un valore di tolleranza scelta in modo approssimato ed arbitrario.

Se la condizione (3.3) non è soddisfatta si prosegue la simulazione per un dato intervallo di tempo predefinito e si ritorna al Passo 2.

3.2.2 Classe Test Matrix

La classe TestMatrix è utilizzata per l'implementazione dell'algoritmo (metodo della matrice) descritto nel paragrafo precedente e per l'esecuzione della fase di analisi dei risultati mediante il metodo delle prove ripetute. La Figura 3.1 illustra un'astrazione del funzionamento della classe.

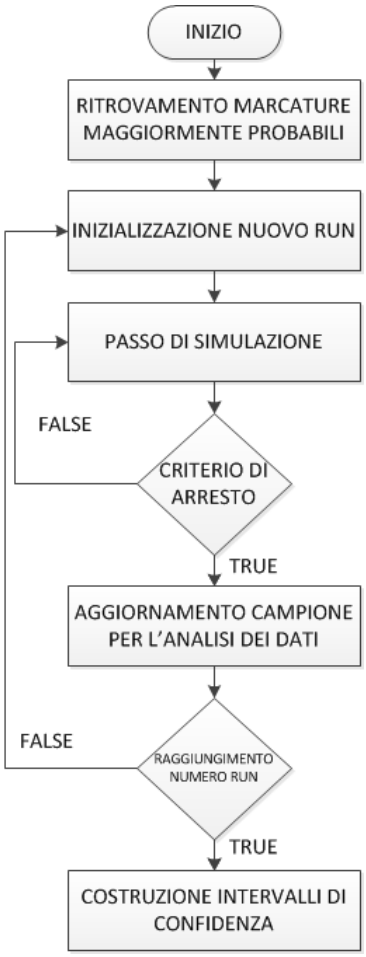

Figura 3.1: Classe TestMatrix

La fase denotata con "RITROVAMENTO MARCATURE MAGGIORMENTE PROBABILI" prevede l'esecuzione di un run pilota per la ricerca di un insieme di marcature caratterizzate dai più alti valori di probabilità analitica non normalizzata. Maggiori dettagli sono presenti nel paragrafo 3.2.3.

La fase di "INIZIALIZZAZIONE NUOVO RUN" comprende la costruzione della struttura di una rete di Petri stocastica e l'eliminazione del transitorio.

E' indicata con "PASSO DI SIMULAZIONE" la fase che descrive il comportamento dinamico della rete. In particolare vengono effettuate le fasi che rappresentano l'astrazione delle regole di enablig e di firing per un intervallo di tempo predefinito.

Il test denotato con "CRITERIO DI ARRESTO" implementa il metodo descritto nel paragrafo 3.2.1.

La fase di "AGGIORNAMENTO CAMPIONE PER L'ANALISI DEI

DATI" prevede l'aggiornamento delle strutture dati dedicate alla costruzione ed alla gestione dei campioni utilizzati per l'analisi dei dati.

Il test "RAGGIUNGIMENTO NUMERO RUN" rappresenta un contatore di iterazioni e verifica il raggiungimento di un numero di prove ripetute scelto inizialmente in modo arbitrario e sulla base di quanto descritto nel paragrafo 3.2.4.

La fase "COSTRUZIONE INTERVALLI DI CONFIDENZA" effettua al campione ottenuto mediante l'esecuzione delle prove ripetute un'analisi statistica classica per la stima della media e della varianza di un particolare indice.

La classe implementata utilizza le seguenti strutture dati:

- Un valore intero (int) denotato con *NumMark*. In riferimento a quanto descritto nel paragrafo 3.2.1, tale valore rappresenta il numero di marcature appartenenti all'insieme W.
- Un valore intero (int) denotato con *NumSimulation*, che indica il numero degli n run i.i.d. (in riferimento al paragrafo 3.2.4) effettuati in fase di analisi dei risultati del simulatore.
- Un numero reale positivo (double) denotato con *tolerance*, che identifica un valore di tolleranza ξ scelto in modo arbitrario e descritto nel Passo 4 dell'algoritmo in paragrafo 3.2.1.
- Una matrice di valori reali (double) denotata con *matrixrap*, che identifica la matrice Z descritta nel Passo 2 dell'algoritmo. Nel dettaglio tale matrice è caratterizzata da un numero *NumMark* di righe e di colonne.
- Una matrice di valori reali (double) denotata con *matrixprob*, caratterizzata da un numero *NumMark* di righe ed un numero di colonne pari a *NumSimulation*. Ogni singola colonna della matrice contiene i valori delle probabilità di stato stimate da PFNetS e relative alle marcature dell'insieme W. Tali valori di probabilità sono ottenuti al termine di ciascun run di simulazione (termine dettato dal verificarsi del criterio descritto nel Passo 3 dell'algoritmo). La matrice così costruita viene utilizzata per la fase di convalida ed analisi dei risultati offerti da PFNetS.
- Un vettore di valori reali (double) denotato con *vetanalitic*, che identifica il vettore $\boldsymbol{g}$ descritto nel Passo 1 dell'algortimo. La dimensione del vettore è pari a *NumMark*.
- Un vettore di valori reali (double) denotato con *vetstim*, che rappresenta il vettore $\widetilde{\boldsymbol{\pi}}$ descritto nel Passo 2 dell'algortimo. La dimensione del vettore è pari a *NumMark*.

- Un vettore di stringhe (double) denotato con *mk*, che identifica l'insieme W descritto nel Passo 1 dell'algoritmo. La dimensione del vettore è pari a *NumMark*.
- Una hashtable denotata con *mark*. Tale struttura contiene lo storico dell'insieme W durante l'esecuzione dell'algoritmo per il ritrovamento delle marcature caratterizzate dai più alti valori di probabilità analitica a meno di una costante di normalizzazione G.
- Un oggetto *pn* che estende la classe MarkovianPetriNet e rappresenta una generica rete SPN di cui si vuole analizzare il comportamento mediante la simulazione.

La classe TestMatrix implementa i seguenti metodi:

```
private static double[][] builtmatrix (double tolerance, int
dim, double[] probstim, double[] probmat)
```

Il metodo prende in ingresso un valore reale positivo *tolerance* che rappresenta la tolleranza ξ scelta in modo arbitrario, un intero positivo *dim* che identifica la dimensione della matrice quadrata da restituire in output, un vettore *probstim* che identifica il vettore $\widetilde{\boldsymbol{\pi}}$ ed infine un vettore *probmat* che rappresenta il vettore $\boldsymbol{g}$. In particolare, si applica il metodo per l'implementazione del Passo 2 e per la verifica della condizione posta dal criterio di arresto descritto al Passo 3 dell'algoritmo. Tale controllo avviene mediante la chiamata al metodo `ControlMatrix(matrix, tolerance)`, dove matrix rappresenta la matrice quadrata Z. Il metodo ritorna il valore `null` se il controllo restituisce un esito negativo, restituisce la matrice *matrix* altrimenti.

Algoritmo 3.1 Metodo builtmatrix

```
private static double[][] buildmatrix(double tolerance, int
dim, double[] probstim, double[] probmat){
double matrix[][] = new double[dim][dim];
boolean flag;
    for (int i = 0; i < dim; i++) {
      for (int j = 0; j < dim; j++) {
        if (i == j) {
          matrix[i][j] = 1;
        } else {
              if (probstim[i] != 0) {
                matrix[i][j] = (probstim[j] / probstim[i]) *
(probmat[i] / probmat[j]);
              }
          }
      }
    }
    flag = ControlMatrix(matrix, tolerance);
    if (flag == true) {
        return matrix;
    } else {
        return null;
    }
 }
```

`private static boolean ControlMatrix (double[][] matrix, double tolerance)`

Il metodo implementa il Passo 3 dell'algoritmo. Si occupa quindi di verificare che ogni elemento della matrice Z soddisfi quanto imposto dal criterio di arresto proposto. In tal caso il metodo restituisce in output il valore booleano `true`. Restituisce il valore booleano `false` altrimenti.

Algoritmo 3.2 Metodo ControlMatrix

```
private static boolean ControlMatrix(double[][] matrix, double
tolerance) {
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix.length; j++) {
            if ((Math.abs(1 - matrix[i][j])) > tolerance) {
            return false;
            }
        }
    }
return true;
}
```

`private static double[][] CompileMatrixProb (double[][] matrixprob, double[] vetstim, int j)`

Il metodo è invocato al termine di ogni singolo $j - esima$ run di simulazione ed inizializza la $j - esima$ colonna della matrice *matrixprob* (passata come parametro) con i valori relativi al vettore *vetstim* le cui componenti rappresentano le probabilità stimate relative alle marcature dell'insieme W.

La costruzione di tale matrice permette di ottenere dei campioni con i quali analizzare i risultati del simulatore PFNetS mediante il metodo delle prove ripetute.

Algoritmo 3.3 Metodo CompileMatrixProb

```
private static double[][] CompileMatrixProb(double[][]
matrixprob, double[] vetstim, int j) {
    for (int i = 0; i < NumMark; i++) {
        matrixprob[i][j] = vetstim[i];
    }
  return matrixprob;
}
```

`private static double computeMeans (double[][] matrixprob, int i)`

Tale metodo prende in input la matrice *matrixprob* (restituita in output dal metodo `CompileMatrixProb`) e restituisce in output la media campionaria (indicata con $\overline{X}$) calcolata sui valori presenti nella riga $i-esima$ della matrice

stessa. In particolare, viene calcolata la media relativa ai valori assunti dalla $i-esima$ marcatura dell'insieme W in ogni singolo run di simulazione.

Dato che, la fase di analisi dei dati offerti da PFNetS prevede l'utilizzo di un campione avente dimensione $n \geq 35$, il metodo controlla che il numero di osservazioni (run terminati rispettando il criterio di arresto) sia maggiore di un valore `minrun` preso arbitrariamente in modo tale da soddisfare le condizioni imposte sulla dimensione del campione.

Algoritmo 3.4 Metodo computeMeans

```
private static double computeMeans(double[][] matrixprob, int
i) {
int num = 0;
double sum = 0.0;
    for (int j = 0; j < NumSimulation; j++) {
        if (matrixprob[i][j] != 0.0) {
          num++;
          sum = sum + matrixprob[i][j];
        }
    } if (num >= minrun) {
        return sum / num;
      } else {
        return -1.0;
      }
  }
```

`private static double computeVariance (double[][] matrixprob, int i, double mean)`

Il metodo prende in input la matrice *matrixprob* (restituita in output dal metodo `CompileMatrixProb`) ed un valore `mean` (restituito dal metodo `computeMeans`). Restituisce in output la varianza calcolata sui valori presenti nella riga $i-esima$ della matrice *matrixprob* utilizzando la media relativa alla stessa riga.

La stima della varianza (indicata con $s^2(x)$) rappresenta una funzione necessaria per il calcolo degli intervalli di confidenza utilizzati in fase di analisi dei risultati. In particolare per la stima della varianza è utilizzata la seguente formula:

$$s^2(x) = \frac{\sum_{j=1}^{n} \left(x_j - \overline{X}\right)^2}{n-1} \tag{3.4}$$

dove n indica la dimensione del campione ed x_i corrisponde alla $j-esima$ osservazione del campione.

Algoritmo 3.5 Metodo computeVariance

```
private static double computeVariance(double[][] matrixprob,
int i, double means) {
int num = 0;
double sum = 0.0;
    for (int j = 0; j < NumSimulation; j++) {
        if (matrixprob[i][j] != 0.0) {
          num++;
          sum = sum + Math.pow(matrixprob[i][j] - means, 2);
        }
    }
  return sum / (num - 1);
}
```

`private static double computeAmpIC (double[][] matrixprob, double mean, double variance, int i)`

Tale metodo prende in input la matrice *matrixprob* (restituita in output dal metodo `CompileMatrixProb`), il valore `mean` (restituito dal metodo `computeMeans`) ed il valore `variance` (restituito dal metodo `computeVariance`).

Ritorna in output l'ampiezza dell'intervallo di confidenza calcolato a partire dai valori presenti nella riga $i - esima$ della matrice *matrixprob* utilizzando la media e la varianza relative alla stessa riga.

Per la costruzione degli intervalli di confidenza si adottano livelli di confidenza pari al 95% ed al 99%.

Algoritmo 3.6 Metodo computeAmpIC

```
private static double computeAmpIC(double[][] matrixprob,
double means, double variance, int i) {
double perc = 0.0;
int num = 0;
    if (i == 95) {
      perc = 1.96;
    } else {
      perc = 2.576;
    }
    for (int j = 0; j < NumSimulation; j++) {
        if (matrixprob[1][j] != 0.0) {
          num++;
        }
    }
double inf = means - perc * (Math.sqrt(variance / num));
double sup = means + perc * (Math.sqrt(variance / num));
return (sup - inf) / means;
}
```

3.2.3 Algoritmo per il ritrovamento delle marcature maggiormente probabili

Data una generica SPN, un possibile evento raro può essere rappresentato dal raggiungimento di una marcatura caratterizzata da una probabilità analitica con grandezza 10^{-10}. Questo significa che mediamente, in fase di simulazione, devono essere eseguiti 10^{10} run indipendenti per ottenere una sola volta un evento raro. Dato che il simulatore PFNetS fornisce due metodi di arresto di simulazione che si basano su un preciso numero di marcature raggiungibili, è utile considerare marcature con i più alti valori di probabilità analitica. Tale accorgimento permette di ottenere una convergenza delle probabilità stimate alle probabilità attese in un minor numero di passi di simulazione rispetto ad un'analisi di marcature meno probabili o rare. Inoltre, è più probabile che in ogni simulazione eseguita, si riscontri l'evento maggiormente probabile a discapito di un evento raro o poco probabile.

Date due marcature M_i e M_j entrambe raggiungibili a partire da M_0 ed aventi probabilità analitica denotata da $\dot{\pi}_i$ ed $\dot{\pi}_j$ rispettivamente, allora la relazione d'ordine ($\leq$) che intercorre tra quest'ultime si mantiene anche per le loro rispettive g_i e g_j. In generale infatti, come illustrato nel paragrafo 3.2.1, i due valori risultano essere approssimativamente a meno di una costante G. Sulla base di ciò e, data la difficoltà riscontrabile nel calcolo delle

probabilità analitiche $\dot{\pi}_i$ ed $\dot{\pi}_j$, l'algoritmo proposto utilizza le probabilità non normalizzate g_i e g_j.

Tuttavia, non è sempre vero che una marcatura avente un alto valore di g_i (calcolato mediante le formule di Coleman Henderson et. al.) corrisponda ad una marcatura con un'alta probabilità stimata. Infatti, è necessario assicurarsi che la stessa marcatura sia raggiunta a partire da una marcatura iniziale M_0. A tal fine è quindi opportuno eseguire un run pilota per assicurarsi che la marcatura appartenga all'insieme di raggiungibilità $RS\,(M_0)$.

Data una rete di Petri stocastica (SPN), a partire dalla marcatura iniziale M_0 (equivalente alla marcatura iniziale utilizzata nella successiva fase di simulazione della stessa rete) viene eseguito un run pilota caratterizzato da un numero arbitrario n di passi di simulazione. Tale esecuzione non prevede l'eliminazione del transitorio ed il suo comportamento può essere descritto mediante un modello di *random walk*.

In generale, una random walk consiste in una formalizzazione matematica di un cammino caratterizzato da una successione di passi casuali.

Definiamo con $\{p_1, \ldots, p_n\}$ i passi di simulazione relativi al run pilota e con $\{M_1, \ldots, M_n\}$ gli stati della rete raggiunti nei successivi passi di simulazione.

Descriviamo qui di seguito l'algoritmo proposto per il ritrovamento delle marcature caratterizzate dai più alti valori di probabilità analitica non normalizzata.

Passo 1: Definiamo con W il vettore delle marcature caratterizzate dai più alti valori di probabilità analitica. Tale vettore rappresenta quanto si vuole ottenere in output dall'algoritmo ed ha dimensione arbitraria k. Il vettore è inizializzato a zero in tutte le sue componenti. Denotiamo con $W\,[i]$ la generica componente del vettore.

Passo 2: Per ogni passo di simulazione p_i con $1 \leq i \leq n$:

- Se $M_i \notin W$
  - ▷ Si calcola g_i che rappresenta la probabilità analitica della marcatura M_i a meno della costante di normalizzazione, come descritto nel paragrafo 2.5.
  - ▷ Si individua l'elemento minimo del vettore W. In presenza di più occorrenze dell'elemento minimo, si prende in considerazione la prima occorrenza. Indichiamo tale valore con $\min(W)$ e con min l'indice relativo alla componente contenente $\min(W)$.
    - ∗ Se $g_i > \min(W)$ allora si sostituisce l'elemento $\min(W)$ con g_i nel vettore W. Formalmente $W\,[min] = g_i$.
    - ∗ Se $g_i \leq \min(W)$ la marcatura non è presa in considerazione in quanto la probabilità g_i ad essa associata non risulta essere una tra le k probabilità maggiori.

- Se $M_i \in W$ allora M_i rappresenta una marcatura candidata ad essere tra le più probabili a meno di successivi aggiornamenti del vettore W.

La Figura 3.2 mostra l'algoritmo implementato nel simulatore PFNetS per l'individuazione delle marcature con i più alti valori di probabilità analitica a meno di una costante di normalizzazione G.

```
...
for (int i = 0; i < 10000000; i++) {
pn.executeFiring();
String s = pn.getMarking().toString();
 if (!mark.containsKey(s)) {
      double probanalitic = pn.calcolaprobanalitica();
      double min = vetanalitic[0];
      int indice = 0;
      for (int j = 0; j < NumMark; j++) {
          if (vetanalitic[j] < min) {
            min = vetanalitic[j];
            indice = j;
          }
      }
  if (vetanalitic[indice] < probanalitic) {
    vetanalitic[indice] = probanalitic;
    mk[indice] = new Marking(pn.getMarking());
    mark.put(new String(pn.getMarking().toString()), new
Integer(1));
  }
 }
}
...
```

Figura 3.2: Algoritmo per il ritrovamento delle marcature maggiormente probabili

L'algoritmo, dopo aver aggiornato la marcatura corrente M_i, verifica che tale marcatura non sia presente nella struttura dati denominata *mark*. Tale struttura rappresenta lo storico dell'insieme W. Tale scelta implementativa è motivata al fine di evitare la memorizzazione di tutte le marcature raggiunte durante l'esecuzione del run pilota, in quanto in tal caso, risulterebbero elevati sia lo spazio per la memorizzazione dei dati, sia il costo computazionale relativo alla ricerca dell'elemento corrente (M_i) effettuata all'interno della struttura (*mark*). E' inoltre necessario avere un riferimento delle marcature appartenenti all'insieme W per evitare l'inserimento di più occorrenze relative alla stessa marcatura.

Una volta individuato il valore minimo del vettore *vetanalitic*, l'algoritmo controlla se quest'ultimo è minore della probabilità analitica, a meno di una costante G, associata alla marcatura corrente M_i. In caso positivo si aggiornano le strutture dati utilizzate per la memorizzazione delle marcature maggiormente probabili.

3.2.4 Convalida, esecuzione ed analisi dei dati

La fase di convalida consiste nel confrontare i risultati forniti dal modello di simulazione con i risultati ottenuti dal sistema reale oggetto di studio. In particolare la convalida si occupa di determinare se il modello operazionale rappresenta in modo corretto il funzionamento del sistema reale.

L'analisi dei risultati della simulazione, invece, si basa sul metodo delle prove ripetute. In generale, supponiamo di voler misurare un indice di prestazione o una particolare misura di interesse x. Quest'ultima è una variabile aleatoria con media μ e varianza σ^2. L'obiettivo della simulazione consiste nello stimare il valore di μ. A tal scopo, si eseguono n esperimenti (con $n \geq 35$) e si eliminano per ognuno di questi le informazioni riguardanti la fase transiente. L'eliminazione del transiente è ottenuta utilizzando il Test di Welch [8].

In particolare, ogni run di simulazione produce un'osservazione che coicide con la stima delle probabilità di stato stazionarie associate alle marcature dell'insieme W.

E' possibile eseguire tale procedimento in quanto le osservazioni derivando da run diverse possono essere considerati i.i.d.

Ogni run prevede l'esecuzione di un numero m di passi di simulazione che cambia in base al comportamento della rete ed è stabilito sulla base del criterio di arresto scelto.

Al termine degli n esperimenti si ottengono le seguenti osservazioni:

$$x_1, x_2, \ldots, x_n$$

A partire dalle osservazioni ottenute si calcola il seguente stimatore per la media:

$$\overline{X} = \frac{\sum_{j=1}^{n} x_j}{n} \tag{3.5}$$

dove x_j corrisponde all'osservazione ottenuta al termine del $j - esimo$ run di simulazione.

L'intervallo di confidenza per la media fornisce un campo di variazione (centrato sulla media campionaria) all'interno del quale ci si aspetta di trovare il parametro incognito μ. Nel dettaglio, l'intervallo ci dice con quale probabilità μ cade in un intervallo attorno ad $\overline{X}$. Ricordiamo che lo stimatore della media è una variabile casuale con media pari a:

$$E\left[\overline{X}\right] = \mu$$

e varianza pari a:

$$\sigma^2\left(\overline{X}\right) = \frac{\sigma^2}{n}$$

Notiamo che all'aumentare del numero di prove di simulazione la varianza decresce, ovvero $\overline{X}$ diventa molto vicino a μ.

Consideriamo ora la seguente variabile aleatoria:

$$Z = \frac{\overline{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Tale variabile ha una distribuzione gaussiana con media 0 e varianza 1 (distribuzione normale). Sia u_α un valore tale che:

$$Prob\left\{-u_\alpha < Z < u_\alpha\right\} = 1 - \alpha$$

quindi:

$$Prob\left\{-u_\alpha < \frac{\overline{X} - \mu}{\frac{\sigma}{\sqrt{n}}} < u_\alpha\right\} = 1 - \alpha$$

ovvero:

$$Prob\left\{\overline{X} - u_\alpha\frac{\sigma}{\sqrt{n}} < \mu < \overline{X} + u_\alpha\frac{\sigma}{\sqrt{n}}\right\}$$

la costante $1-\alpha$ è di solito espressa in percentuale ed è definita livello di confidenza. Tale costante rappresenta il grado di attendibilità dell'intervallo. L'intervallo $\overline{X} \pm u_\alpha\frac{\sigma}{\sqrt{n}}$ è definito intervallo di confidenza [9].

I valori utilizzati nel simulatore PFNetS relativi al valore $1-\alpha$ sono 0.95 e 0.99. Per un grado di confidenza del 95% il valore di u_α è pari a 1.96, mentre per un grado di fiduci del 99% il valore di u_α è pari a 2.576.

Ciò significa che il valore teorico μ cade nell'intervallo costruito con una probabilità del 95% o del 99% rispettivamente.

Dato che σ^2 non è nota, il simulatore sostituisce quest'ultima con lo stimatore di varianza (3.6).

L'ampiezza relativa dell'intervallo di confidenza viene calcolata come segue:

$$ampiezza\,relativa = \frac{\left(\overline{X} + u_\alpha\frac{\sigma}{\sqrt{n}}\right) - \left(\overline{X} - u_\alpha\frac{\sigma}{\sqrt{n}}\right)}{\overline{X}}$$

L'analisi dei risultati della simulazione restituisce un esito positivo se il valore teorico μ cade nell'intervallo di confidenza stimato.

3.2.5 Complessità e miglioramenti

L'algoritmo descritto nel paragrafo 3.2.3 e relativo al ritrovamento delle marcature raggiungibili dallo stato iniziale M_0 caratterizzate dai maggiori valori di probabilità non normalizzata prevede, per quel che concerne l'operazione di ricerca, un costo computazionale pari a $O(n \cdot \log(m))$ dove n è la cardinalità del vettore W ed m è la dimensione della hashtable. Tale costo risulta quindi essere logaritmico per la ricerca dello stato corrente effettuata all'interno della hashtable ed inoltre, al costo ottenuto viene moltiplicata una quantità lineare che identifica il costo computazionale relativo alla ricerca del valore minimo nel vettore che identifica l'insieme W.

Per quel che concerne l'algoritmo descritto nel paragrafo 3.2.1, sia la fase relativa alla costruzione della matrice *matrix* (mediante il metodo ***builtmatrix***), sia la fase relativa al controllo imposto dal primo criterio di arresto (tramite il metodo ***ControlMatrix***), prevedono un costo computazionale pari a $O(NumMark^2)$ dove ***NumMark*** rappresenta la cardinalità dell'insieme W. In tal modo, si ottiene un costo computazionale complessivo e relativo ad un singolo run pari a $NumStep * 2 * O(NumMark^2)$ dove *NumStep* indica il numero totale di passi di valutazione necessari per terminare ogni singola run in rispetto del criterio di arresto proposto.

E' possibile introdurre dei miglioramenti per diminuire il costo computazionale degli algoritmi sopracitati. In particolare, per quel che riguarda l'algoritmo di costruzione e di controllo della matrice ***matrix***, è possibile effettuare il controllo relativo alle componenti della matrice stessa successivamente ad ogni singolo passo di costruzione ottenendo in tal modo un costo computazionale inferiore e pari a $O(NumMark^2)$ per il singolo passo di valutazione e $NumStep * O(NumMark^2)$ per i singoli run. Tuttavia, non abbiamo adottato tale modifica nella costruzione, dell'algoritmo presentato nel nostro elaborato per poter analizzare e rendere visibili ad ogni singolo passo di analisi i cambiamenti della matrice Z.

In fase di costruzione della matrice *matrix*, è possibile diminuire il numero di operazioni effettuate dall'algoritmo ***builtmatrix*** calcolando gli elementi relativi alla sola sotto-matrice triangolare superiore della matrice stessa. Ogni elemento della sottomatrice triangolare inferiore, infatti, corrisponde al reciproco dell'elemento (avente indici i e j invertiti) della sottomatrice triangolare superiore.

3.2.6 Convalida ed analisi dei risultati relative a SPN chiuse.

L'analisi dei risultati del simulatore PFNetS relative a reti di Petri stocastiche chiuse consiste, come descritto nel paragrafo 3.2.4, di effettuare una stima delle grandezze prese in considerazione mediante la generazione di un intervallo di confidenza relativo al campione prodotto dagli n (pari a 50) run di simulazione. Le fasi restituiscono un esito positivo se il valore analitico

della grandezza presa in esame cade all'interno dell'intervallo di confidenza stimato. In particolare, negli esempi che seguono, si prendono in considerazione come grandezza di studio le singole probabilità di stato stazionarie stimate da PFNetS e relative alle rispettive marcature dell'insieme W.

Trattandosi di reti di Petri stocastiche chiuse con insieme di raggiungibilità finito, il valore analitico relativo alle singole probabilità di stato stazionarie viene calcolato utilizzando il sistema lineare (2.1) presente in paragrafo 2.2.

3.2.6.1 Primo esempio

Riportiamo i risultati ottenuti da PFNetS relativamente ad una rete di Petri stocastica chiusa con insieme di raggiungibilità finito. In particolare, analizziamo la SPN illustrata in Figura 2.8, presentata nel paragrafo 2.5, con i seguenti valori relativi ai tassi di firing delle transizioni: $\chi_1 = 5$, $\chi_2 = 4$, $\chi_3 = 25$, $\chi_4 = 2$. Sulla base di tali valori e mediante l'applicazione del teorema di Coleman Henderson et. al. si ottengono le seguenti intensità di traffico:

$$\begin{cases} \rho_1 = \chi_4/\chi_1 = 0.4 \\ \rho_2 = 1 \end{cases}$$

Consideriamo i seguenti parametri di simulazione:

- Il valore di tolleranza ξ è 0.0015.
- La marcatura iniziale $M_0 = [6\,4]$.
- Il numero di marcature appartenenti all'insieme W, $|W| = 4$.
- Il numero di prove n di simulazione $n = 50$.

L'algoritmo, descritto nel paragrafo 3.2.3 per il ritrovamento delle marcature raggiungibili a partire da M_0 aventi i maggiori valori di probabilità analitica stazionaria non normalizzata restituisce l'insieme W con le marcature illustrate in Tabella 3.1.

Marcatura	Prob. analitica a meno di G
[1 9]	0.4
[3 7]	0.064
[0 10]	1.0
[2 8]	0.16

Tabella 3.1: Marcature dell'insieme W

Il simulatore, sulla base dell'insieme W restituisce il seguente output:

```
Arresto Simulazione 1 al passo:  497000
Tempo di simulazione 41481.85824928762
Matrice Z
        1.0          0.9987315347183288  0.9999599610796267  0.9990277079695631
 1.0012700763294002          1.0          1.001229986556542   1.0002965494138703
 1.0000400405235526  0.9987715244518672          1.0          0.9990677095620336
  1.000973238302282  0.9997035385016134  1.000933160414498           1.0

Risultati:
[ 1 9 ] con probabilità stimata:  0.24010139627208008
[ 3 7 ] con probabilità stimata:  0.03836749375789251
[ 0 10 ] con probabilità stimata:  0.6002294571784831
[ 2 8 ] con probabilità stimata:  0.0959471790391952

Arresto Simulazione 2 al passo:  175000
Tempo di simulazione14578.012836330496
Matrice Z
        1.0          1.000275865999885   1.0000075717253594  0.999056236984579
 0.9997242100811768          1.0          0.9997317797183308  0.9987807073460813
 0.9999924283319712  1.0002682922430903          1.0          0.9990486724624106
 1.0009446545454432  1.0012207811433989  1.0009522334234675          1.0

Risultati:
[ 1 9 ] con probabilità stimata:  0.24002496671444798
[ 3 7 ] con probabilità stimata:  0.038414589030702086
[ 0 10 ] con probabilità stimata:  0.6000669602939384
[ 2 8 ] con probabilità stimata:  0.09591937601123411
                                  .
                                  .
                                  .
Arresto Simulazione 49 al passo:  2856000
Tempo di simulazione 237867.8463530521
Matrice Z
        1.0          0.9991428695293697  0.9979581060087899  0.9970463714172982
 1.0008578657735243          1.0          0.998814220111346   0.9979017033739539
 1.0020460718530324  1.0011871876318719          1.0          0.9990863999340232
 1.0029623783480635  1.0021027087326855  1.0009144354943047          1.0

Risultati:
[ 1 9 ] con probabilità stimata:  0.24019978137724232
[ 3 7 ] con probabilità stimata:  0.03842739293584089
[ 0 10 ] con probabilità stimata:  0.5998894494461401
[ 2 8 ] con probabilità stimata:  0.09593661824874102

Arresto Simulazione 50 al passo:  2856000
Tempo di simulazione 237867.8463530521
Matrice Z
        1.0          1.00004874053921    0.9987086645920956  1.0001950553177523
 0.9999512618363144          1.0          0.9986599893657264  1.000146307647429
 1.0012930051111872  1.0013418086721633          1.0          1.0014883126364624
 0.9998049827214048  0.9998537137553675  0.9985138991462174          1.0

Risultati:
[ 1 9 ] con probabilità stimata:  0.24017810301029238
[ 3 7 ] con probabilità stimata:  0.03843036950728633
[ 0 10 ] con probabilità stimata:  0.5996698813041796
[ 2 8 ] con probabilità stimata:  0.09608998041059691
```

Figura 3.3: Output di simulazione

In particolare, l'output di simulazione restituisce, per ogni run, il tempo di simulazione ed il passo per il quale vengono verificate le condizioni imposte dal criterio utilizzato. Inoltre, si stampa a video l'ultima matrice Z dei rapporti prodotta dalla simulazione e le probabilità di stato stazionarie stimate relative alle marcature dell'insieme W.

Al termine dell'esecuzione delle 50 prove di simulazione, per ogni marcatura dell'insieme W, si ottengono le seguenti informazioni relative agli intervalli di confidenza:

```
Resoconto Fine Prove Ripetute

Marcatura [ 1 9 ] media95:  0.24003718091305914
Marcatura [ 1 9 ] varianza95:  2.3786479927793925E-8
Marcatura [ 1 9 ] ampiezzaIC95:  3.5619484988174397E-4

Marcatura [ 3 7 ] media95:  0.038401184182459816
Marcatura [ 3 7 ] varianza95:  1.0622730203750585E-9
Marcatura [ 3 7 ] ampiezzaIC95:  4.705161379040675E-4

Marcatura [ 0 10 ] media95:  0.6000290926887053
Marcatura [ 0 10 ] varianza95:  6.624508575418225E-8
Marcatura [ 0 10 ] ampiezzaIC95:  2.3779657234206512E-4

Marcatura [ 2 8 ] media95:  0.09600753777135879
Marcatura [ 2 8 ] varianza95:  4.685304930091305E-9
Marcatura [ 2 8 ] ampiezzaIC95:  3.952434336295675E-4

Marcatura [ 1 9 ] media99:  0.24003718091305914
Marcatura [ 1 9 ] varianza99:  2.3786479927793925E-8
Marcatura [ 1 9 ] ampiezzaIC99:  4.681418027016215E-4

Marcatura [ 3 7 ] media99:  0.038401184182459816
Marcatura [ 3 7 ] varianza99:  1.0622730203750585E-9
Marcatura [ 3 7 ] ampiezzaIC99:  6.183926383881926E-4

Marcatura [ 0 10 ] media99:  0.6000290926887053
Marcatura [ 0 10 ] varianza99:  6.624508575418225E-8
Marcatura [ 0 10 ] ampiezzaIC99:  3.1253263793545476E-4

Marcatura [ 2 8 ] media99:  0.09600753777135879
Marcatura [ 2 8 ] varianza99:  4.685304930091305E-9
Marcatura [ 2 8 ] ampiezzaIC99:  5.194627984843762E-4
```

Figura 3.4: Resoconto prove ripetute

Sulla base delle informazioni raccolte in Figura 3.4 riportiamo l'esito relativo alla fase di analisi dei risultati effettuata a partire da differenti gradi di confidenza:

Confidenza 95%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[1 9]	0.240010	0.240037	0.239994	0.240079	3.561948E-4	SI
[3 7]	0.038401	0.038401	0.038392	0.038410	4.70516E-4	SI
[0 10]	0.600025	0.600029	0.599957	0.600100	2.377965E-4	SI
[2 8]	0.096004	0.096007	0.095988	0.096026	3.952434E-4	SI

Tabella 3.2: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[1 9]	0.240010	0.240037	0.239980	0.240093	4.681418E-4	SI
[3 7]	0.038401	0.038401	0.038389	0.038413	6.183926E-4	SI
[0 10]	0.600025	0.600029	0.599935	0.600122	3.125326E-4	SI
[2 8]	0.096004	0.096007	0.095982	0.096032	5.194627E-4	SI

Tabella 3.3: Risultati di simulazione. Intervalli con grado di confidenza del 99%

Come si può notare dalle Tabelle 3.2 e 3.3, le fasi di convalida ed analisi dei dati offerti da PFNetS restituiscono un esito positivo in quanto i valori analitici relativi alle probabilità stazionarie di stato cadono all'interno dei rispettivi intervalli di confidenza centrati sulla media del campione.

3.2.6.2 Secondo esempio

Al fine di studiare il comportamento della SPN in Figura 2.8 con valori di intensità di traffico maggiori ai rispettivi valori definiti nell'esempio 3.2.6.1, modifichiamo i valori associati ai tassi di firing delle transizioni, garantendo comunque la condizione di stazionarietà $\rho_i < 1$ per ogni $i-esimo$ posto della rete.

Si scelgono i seguenti tassi di firing delle transizioni: $\chi_1 = 6$, $\chi_2 = 25$, $\chi_3 = 36$, $\chi_4 = 5$.

Sulla base di tali valori e mediante l'applicazione del teorema di Coleman Henderson et. al. otteniamo le seguenti intensità di traffico:

$$\begin{cases} \rho_1 = \chi_4/\chi_1 = 0.83 \\ \rho_2 = 1 \end{cases}$$

Consideriamo i seguenti parametri di simulazione:

- Il valore di tolleranza $\xi = 0.0015$.
- La marcatura iniziale $M_0 = [6\,4]$.
- Il numero di marcature appartenenti all'insieme W, $W = 4$.
- Il numero di prove di simulazione $n = 50$.

L'algoritmo per il ritrovamento delle marcature con i maggiori valori di probabilità analitica non normalizzara, raggiungibili a partire da M_0, restituisce l'insieme W illustrato in Tabella 3.4. A differenza dell'esempio precedente, le marcature assumono differenti valori di probabilità analitica non normalizzata.

Marcatura	Prob. analitica a meno di G
[1 9]	0.833333
[3 7]	0.578703
[0 10]	1.0
[2 8]	0.694444

Tabella 3.4: Marcature dell'insieme W

Riportiamo gli esiti relativi alle fasi di convalida ed analisi dei risultati ottenuti a partire da differenti gradi di confidenza:

Confidenza 95%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[1 9]	0.160489	0.160481	0.160416	0.160546	8.096296E-4	SI
[3 7]	0.111451	0.111441	0.111396	0.111485	8.011628E-4	SI
[0 10]	0.192586	0.192569	0.192489	0.192650	8.360303E-4	SI
[2 8]	0.133741	0.133712	0.133660	0.133764	7.811021E-4	SI

Tabella 3.5: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[1 9]	0.160489	0.160481	0.160395	0.160566	0.001064	SI
[3 7]	0.111451	0.111441	0.111382	0.111499	0.001052	SI
[0 10]	0.192586	0.192569	0.192463	0.192675	0.001098	SI
[2 8]	0.133741	0.133712	0.133643	0.133781	0.001026	SI

Tabella 3.6: Risultati di simulazione. Intervalli con grado di confidenza del 99%

Come si può notare dalle Tabelle 3.5 e 3.6, anche considerando valori di intensità di traffico vicini ad 1 le fasi restituiscono un esito positivo. Infatti, i valori analitici relativi alle probabilità stazionarie di stato cadono all'interno dei rispettivi intervalli di confidenza centrati sulla media del campione.

3.2.7 Convalida ed analisi dei risultati relative a SPN aperte.

La fase di analisi dei risultati del simulatore PFNetS per reti di Petri stocastiche aperte segue la stessa procedura descritta per le reti chiuse nel paragrafo 3.2.6. Essendo le reti di Petri stocastiche aperte caratterizzate da uno spazio degli stati avente cardinalità infinita, il sistema lineare (2.1) descritto nel paragrafo 2.2 è applicabile solamente in pochi casi pratici perchè difficile dal punto di vista computazionale. Per tale motivo, in questa sezione prendiamo in esame per le fasi di analisi e convalida dei risultati solamente SPN riconducibili a reti di code aperte di Jackson. Tale espediente permette di calcolare i valori di probabilità di stato stazionaria analitica delle marcature prese in esame in forma chiusa e quindi come prodotto di funzioni degli stati dei singoli posti.

Osservazione. In generale una rete di code aperta si dice rete di Jackson aperta se soddisfa le seguenti condizioni:

- Gli arrivi dall'esterno della rete ad un nodo i della rete con $i = 1, ..., m$ sono Poissoniani di parametro γ_i con $\gamma_i > 0$ per almeno un nodo i.
- I tempi di servizio di ciascuno degli s_i serventi presenti in ogni nodo i sono indipendenti e distribuiti esponenzialmente di paramento μ_i.
- Le probabilità che un utente dopo aver completato il servizio nel nodo i si rechi presso il nodo j con $i, j = 1, ..., n$, indicata con p_{ij}, è indipendente dallo stato del sistema.
- Servizi e arrivi sono indipendenti e gli utenti indistinguibili.

Il teorema di Jackson [19] fornisce la soluzione delle probabilità stazionarie di stato in forma prodotto, ovvero la probabilità che la rete si trovi allo stato $n = (n_1, \ldots, n_m)$ è uguale al prodotto delle probabilità marginali, ovvero al prodotto delle probabilità $\pi_i(n_i)$ che il nodo i si trovi nello stato n_i per ogni $i = 1, \ldots, n$.

La condizione sotto la quale una rete di code ammette distribuzione stazionaria è che la capacità di servizio di ogni singolo nodo sia strettamente maggiore della frequenza media effettiva degli arrivi, ovvero l'intensità di traffico ρ di ogni nodo j, denotato da ρ_j sia strettamente minore di uno, dove nel caso di nodi a servente singolo $\rho_j = \lambda_j/\mu_j < 1$ con $j = 1, \ldots, m$

Teorema. ***(Jackson)*** *Data una rete aperta di Jackson composta da m nodi allora la distribuzione stazionaria di probabilità congiunta che la rete si trovi allo stato $(n_1, \ldots, n_m)$ si fattorizza nel prodotto delle distribuzioni di probabilità marginali, ovvero:*

$$\pi(n_1, \ldots, n_m) = \prod_{i=1}^{m} \pi_i(n_i) \tag{3.6}$$

dove

$$\pi_i(n) = \rho_i^n (1 - \rho_i) \tag{3.7}$$

se il numero dei serventi è unitario.

In virtù del teorema di Jackson particolari rete di Petri stocastiche possono essere considerate come m sistemi a coda $M/M/1$ indipendenti, ovvero la rete si comporta come se ciascun posto che la compone (e annesse transizioni collegate in input ed output) fosse un nodo isolato indipendente $M/M/1$ con frequenza media di arrivo λ corrispondente al tasso di firing delle transizioni in input, e tempo medio di servizio pari a μ, corrispondente al tasso di firing delle transizioni in output. Inoltre, $\pi_i(n)$ rappresenta la probabilità stazionaria di osservare n token nel posto i considerato come sistema isolato di tipo $M/M/1$.

In Figura 3.5 viene schematizzata la corrispondenza tra i due modelli.

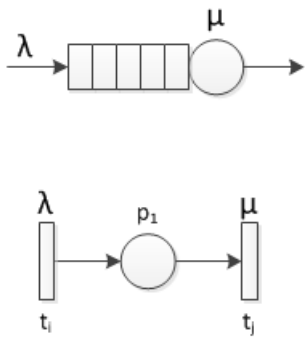

Figura 3.5: Corrispondenza modello $M/M/1$ e rete di Petri

Un modello a singola coda $M/M/1$ che soddisfa le condizioni di stazionarietà ha la seguente distribuzione di probabilità del numero di utenti nel sistema in condizioni stazionarie:

$$\pi_k = \rho^k (1 - \rho) \tag{3.8}$$

con $k \geq 0$.

π_k indica quindi la probabilità di avere k utenti nel sistema, per$k \geq 0$.

Consideriamo ora la convalida e l'analisi dei risultati del simulatore PFNetS nel caso di SPN aperte con uno spazio degli stati avente cardinalità pari a $\mathbb{N}^m$ dove m corrisponde al numero dei posti della rete. Assumiamo che i valori di intensità di traffico relativi ai posti della rete siano tutti minori di

1 garantendo in tal modo la condizione di stazionarietà della rete. Di conseguenza, la costante di normalizzazione può essere calcolata facilmente come il prodotto delle costanti di normalizzazione relative ai singoli posti. Ogni posto i viene considerato come una coda $M/M/1$ con intensità di traffico pari a ρ_i. Nel dettaglio:

$$G = \prod_{i=1}^{m} (1 - \rho_i) \tag{3.9}$$

Le probabilità stazionarie di stato relative alle marcature oggetto di studio hanno la seguente soluzione in forma prodotto:

$$\pi(t_1, .., t_m) = G \prod_{i=1}^{m} \rho_i^{t_i} \tag{3.10}$$

dove t_i rappresenta il numero di token presente nel posto i.

3.2.7.1 Primo esempio

In questo paragrafo analizziamo una $SPN = (P, T, I, O, W)$ aperta con grafo di raggiungibilità infinito e con $n = 5$ e $m = 8$. La rete è illustrata in Figura 3.6.

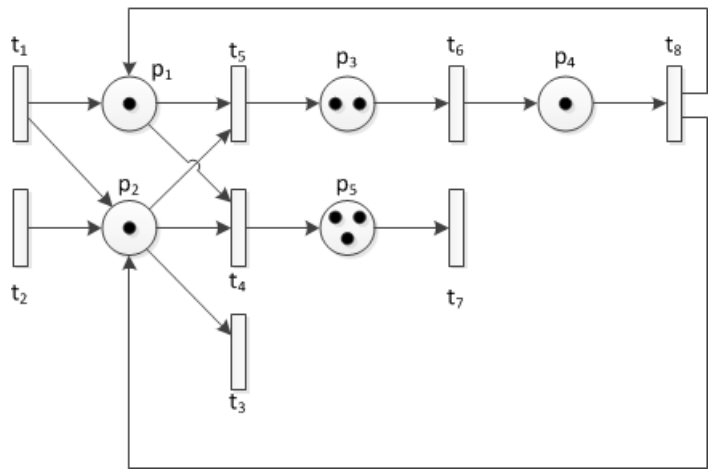

Figura 3.6: SPN aperta con grafo di raggiungibilità infinito

Applicando il teorema di Coleman Henderson et. al. otteniamo la seguente soluzione in forma prodotto:

$$\pi(M) = G \left[\frac{\chi_1\chi_3}{\chi_4\chi_2}\right]^{M_1} \left[\frac{\chi_2}{\chi_3}\right]^{M_2} \left[\frac{\chi_5\chi_1}{\chi_4\chi_6}\right]^{M_3} \left[\frac{\chi_1\chi_5}{\chi_4\chi_8}\right]^{M_4} \left[\frac{\chi_1}{\chi_7}\right]^{M_5}$$

In particolare, impostatiamo i seguenti valori dei tassi di firing delle transizioni: $\chi_1 = 1.5$, $\chi_2 = 3.0$, $\chi_3 = 11.0$, $\chi_4 = 19.0$, $\chi_5 = 9.0$, $\chi_6 = 2.0$, $\chi_7 = 10.0$, $\chi_8 = 3.0$.

Di conseguenza si ricavano i seguenti valori relativi alle intensità di traffico associate ai posti della rete.

$$\begin{cases} \rho_1 = \frac{\chi_1\chi_3}{\chi_4\chi_2} = 0.289 \\ \rho_2 = \frac{\chi_2}{\chi_3} = 0.272 \\ \rho_3 = \frac{\chi_5\chi_1}{\chi_4\chi_6} = 0.355 \\ \rho_4 = \frac{\chi_1\chi_5}{\chi_4\chi_8} = 0.236 \\ \rho_5 = \frac{\chi_1}{\chi_7} = 0.15 \end{cases}$$

Assumiamo i seguenti parametri di simulazione:

- Il valore di tolleranza $\xi = 0.0015$.
- La marcatura iniziale $M_0 = [1\,1\,0\,1\,0]$.
- Il numero di marcature appartenenti all'insieme $|W| = 4$.
- Il numero di prove di simulazione pari a 50.

L'algoritmo utilizzato per l'identificazione delle marcature raggiungibili a partire da M_0 ed aventi i più alti valori di probabilità analitica non normalizzata restituisce l'insieme W con le marcature illustrate in Tabella 3.7:

Marcatura	Prob. analitica a meno di G
[0 0 0 0 0]	1.0
[0 1 0 0 0]	0.272727
[1 0 0 0 0]	0.289473
[0 0 1 0 0]	0.355263

Tabella 3.7: Marcature dell'insieme W

I valori analitici delle probabilità stazionarie di stato delle marcature appartenenti all'insieme W sono calcolate utilizzando le formule (3.9) e (3.10).

Riportiamo i risultati ottenuti da PFNetS relativamente alle fase di convalida ed analisi dei risultati effettuate a partire da differenti gradi di confidenza.

Confidenza 95%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[0 0 0 0 0]	0.216119	0.216119	0.215989	0.216271	0.001306	SI
[0 1 0 0 0]	0.058941	0.058944	0.058905	0.058983	0.001325	SI
[1 0 0 0 0]	0.062560	0.062562	0.062521	0.062604	0.001323	SI
[0 0 1 0 0]	0.076779	0.07678	0.076733	0.076826	0.001213	SI

Tabella 3.8: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[0 0 0 0 0]	0.216119	0.216130	0.215945	0.216316	0.001716	SI
[0 1 0 0 0]	0.058941	0.058944	0.058893	0.058995	0.001741	SI
[1 0 0 0 0]	0.062560	0.062562	0.062508	0.062617	0.0017395	SI
[0 0 1 0 0]	0.076779	0.076780	0.076718	0.076841	0.001595	SI

Tabella 3.9: Risultati di simulazione. Intervalli con grado di confidenza del 99%

Come riportato dalle Tabelle 3.8 e 3.9, considerando bassi valori di intensità di traffico relativi ai posti della rete, la fase di convalida e la successiva analisi dei dati restituiscono un esito positivo. Infatti, i valori analitici relativi alle probabilità stazionarie di stato associate alle marcature appartenenti all'insieme W cadono all'interno dei rispettivi intervalli di confidenza centrati sulla media del campione. Gli intervalli di confidenza sono costruiti con grado di confidenza del 95% e del 99%.

3.2.7.2 Secondo esempio

Lo scopo di questo esempio consiste nel convalidare ed effettuare l'analisi dei risultati offerti da PFNetS analizzando la rete proposta nell'esempio 3.2.7.1 con valori di intensità di traffico maggiori. A tal scopo modifichiamo i valori associati ai tassi di firing delle transizioni, continuando a garantire la condizione di stazionarietà $\rho_i < 1$ relativa ad ogni $i-esimo$ posto della rete.

In particolare, analizziamo la rete in Figura 3.6 con i seguenti valori di tassi di firing delle transizioni: $\chi_1 = 1.0$, $\chi_2 = 5.0$, $\chi_3 = 9.0$, $\chi_4 = 3.0$, $\chi_5 = 7.0$, $\chi_6 = 4.0$, $\chi_7 = 1.7$, $\chi_8 = 3.8$.

Da questi parametri si ottengono le seguenti intensità di traffico:

$$\begin{cases} \rho_1 = \frac{\chi_1\chi_3}{\chi_4\chi_2} = 0.6 \\ \rho_2 = \frac{\chi_2}{\chi_3} = 0.555 \\ \rho_3 = \frac{\chi_5\chi_1}{\chi_4\chi_6} = 0.583 \\ \rho_4 = \frac{\chi_1\chi_5}{\chi_4\chi_8} = 0.615 \\ \rho_5 = \frac{\chi_1}{\chi_7} = 0.588 \end{cases}$$

Impostiamo i seguenti parametri di simulazione:

- Il valore di tolleranza pari a $\xi = 0.003$.
- La marcatura iniziale $M_0 = [1\,1\,0\,1\,0]$.
- Il numero di marcature appartenenti all'insieme $|W| = 4$.
- Il numero di prove di simulazione pari a 50.

L'algoritmo per l'identificazione delle marcature appartenti all'insieme W restituisce le marcature riportate in Tabella 3.10

Marcatura	Prob. analitica a meno di G
[0 0 0 0 0]	1.0
[0 0 0 0 1]	0.588235
[0 0 0 1 0]	0.614035
[1 0 0 0 0]	0.6

Tabella 3.10: Marcature dell'insieme W

Le Tabella 3.11 e 3.12 riportano i risultati ottenuti da PFNetS al fine di effettuarne l'analisi.

Confidenza 95%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[0 0 0 0 0]	0.011772	0.011779	0.011758	0.011801	0.003684	SI
[0 0 0 0 1]	0.006924	0.006931	0.006919	0.006942	0.003417	SI
[0 0 0 1 0]	0.007228	0.007235	0.007222	0.007249	0.003679	SI
[1 0 0 0 0]	0.007063	0.007070	0.007058	0.007083	0.003592	SI

Tabella 3.11: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[0 0 0 0 0]	0.011772	0.011779	0.011751	0.011808	0.004842	SI
[0 0 0 0 1]	0.006924	0.006931	0.006915	0.006946	0.004491	SI
[0 0 0 1 0]	0.007228	0.007235	0.007218	0.007253	0.004836	SI
[1 0 0 0 0]	0.007063	0.007070	0.007054	0.007087	0.004721	SI

Tabella 3.12: Risultati di simulazione. Intervalli con grado di confidenza del 99%

Come si evince dalle Tabelle 3.11 e 3.12, le fasi di convalida e di analisi dei risultati del simulatore restituiscono esito positivo per intervalli di confidenza costruiti con grado di confidenza del 95% e del 99%.

3.2.8 Precisione

Lo scopo di questo paragrafo consiste nell'analizzare come la scelta di differenti valori di tolleranza e di dimensione dell'insieme W possano influenzare l'accuratezza delle stime prodotte da PFNetS utilizzando il criterio di arresto per terminare le singole run.

Il primo esperimento consiste in un'analisi dei risultati offerti dal simulatore, relativamente alle stime di probabilità di stato stazionarie delle marcature dell'insieme W, al variare del valore di tolleranza scelto. In particolare, a partire dalla rete di Petri stocastica presente in Figura 2.8 eseguiamo differenti simulazioni con un valore ξ di tolleranza variabile tra il valore 0.0007 ed il valore 0.01.

Ogni simulazione è caratterizzata dai seguenti parametri:

- La marcatura iniziale $M_0 = [2\,2]$.
- 50 prove ripetute per la creazione del campione.
- Il numero di marcature appartenenti all'insieme $|W| = 4$.
- I seguenti valori relativi ai tassi di firing delle transizioni: $\chi_1 = 5$, $\chi_2 = 4$, $\chi_3 = 25$, $\chi_4 = 2$.

Si ricavano le seguenti intensità di traffico relative ai posti della rete:

$$\begin{cases} \rho_1 = \chi_4/\chi_1 = 0.4 \\ \rho_2 = 1 \end{cases}$$

L'algoritmo utilizzato per il ritrovamento delle marcature raggiungibili a partire da M_0 ed aventi i più alti valori di probabilità analitica non normalizzata restituisce un insieme W con le seguenti marcature:

Marcatura	Prob. analitica a meno di G	Prob. Analitica
[3 1]	0.064	0.038796
[1 3]	0.4	0.242483
[0 4]	1.0	0.606208
[2 2]	0.16	0.096993

Tabella 3.13: Marcature dell'insieme W

Le Figure 3.7, 3.8, 3.9 e 3.10 illustrano in forma grafica i risultati ottenuti in output dal simulatore PFNetS per le probabilità stazionarie di stato stimate per le marcature apparteneti all'insieme W, al variare del parametro di tolleranza ξ. Inoltre, in ogni grafico si rappresentata la retta di equazione $y = PA_{M_i}$ dove PA_{M_i} indica il valore di probabilità stazionaria di stato analitica relativa alla marcatura M_i appartente all'insieme W.

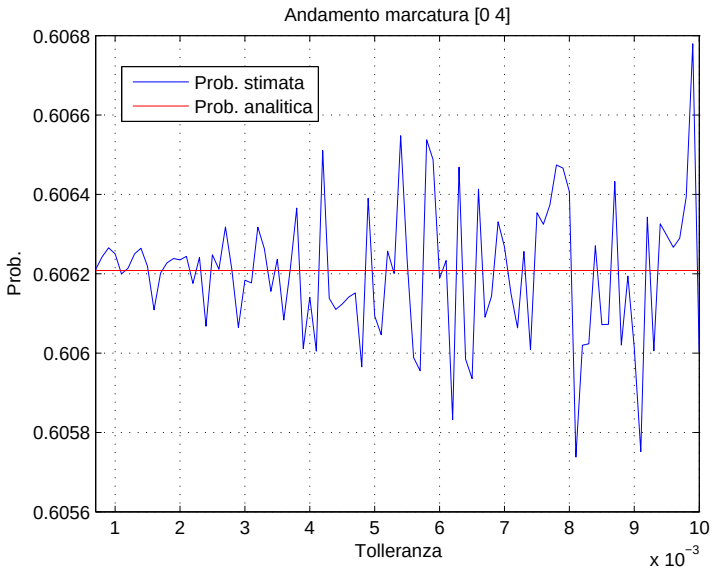

Figura 3.7: Andamento della probabilità stimata relativa alla marcatura [0 4]

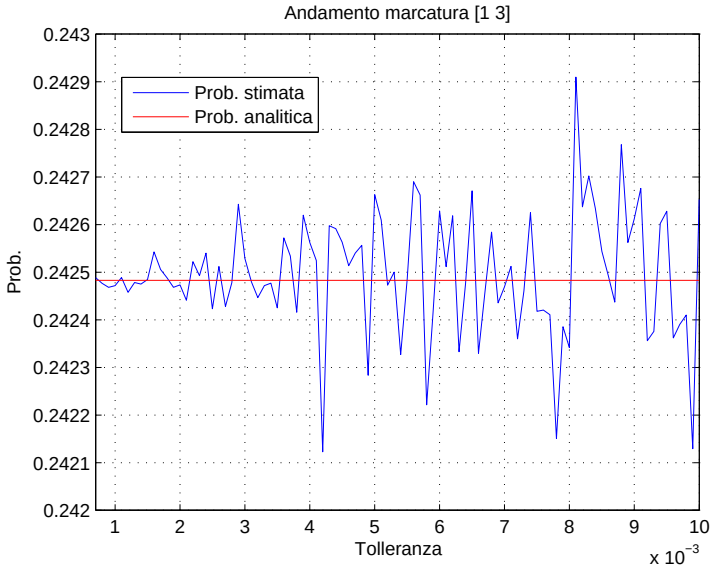

Figura 3.8: Andamento della probabilità stimata relativa alla marcatura [1 3]

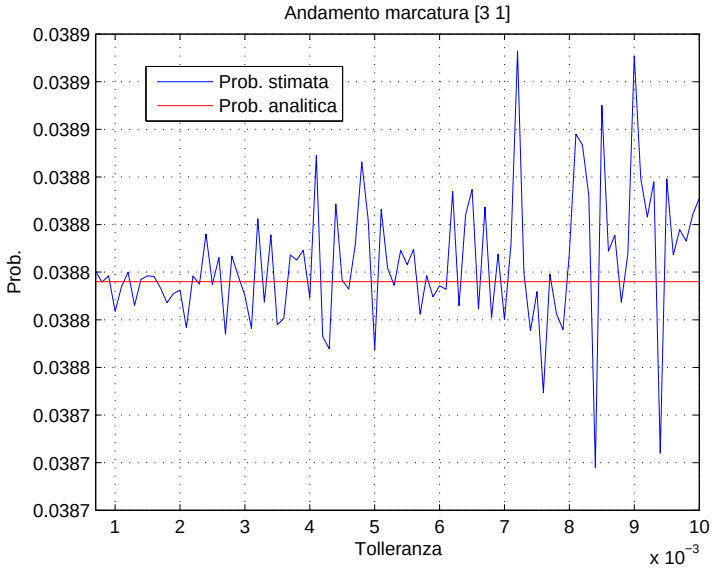

Figura 3.9: Andamento della probabilità stimata relativa alla marcatura [3 1]

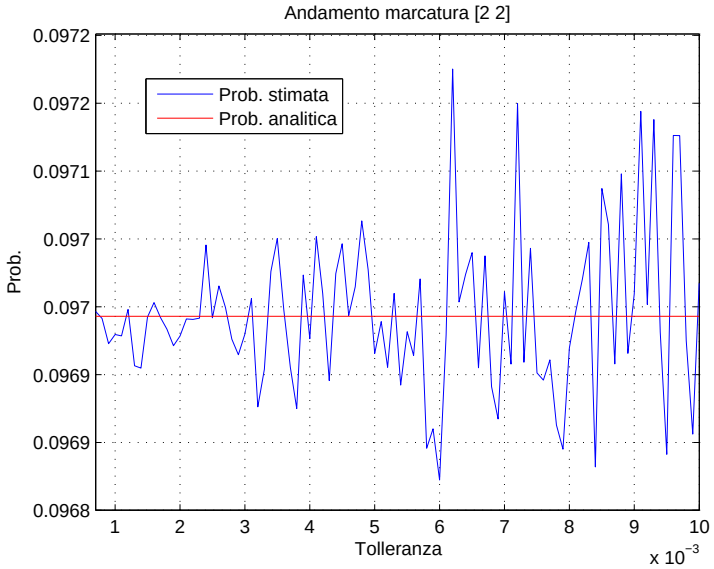

Figura 3.10: Andamento della probabilità stimata relativa alla marcatura [2 2]

Come possiamo notare dalle Figure 3.7÷3.10 al crescere del valore della tolleranza ξ, i valori delle probabilità stimate da PFNetS e relative alle marcature dell'insieme W sono caratterizzate da una maggiore divergenza rispetto ai corrispettivi valori ottenuti dal modello analitico. Concludiamo quindi che è possibile ottenere stime più accurate delle probabilità stazionarie di stato diminuendo il valore di tolleranza a discapito di un elevato tempo di simulazione.

La Tabella 3.14 riporta le ampiezze relative degli intervalli di confidenza calcolati con alcuni dei valori di tolleranza utilizzati e con grado di confidenza del 95%.

Marcatura	AmpiezzaIC $\xi = 0.01$	AmpiezzaIC $\xi = 0.005$	AmpiezzaIC $\xi = 0.001$	AmpiezzaIC $\xi = 0.0007$
[3 1]	0.003116	0.001380	3.030949E-4	1.830949E-4
[1 3]	0.002145	0.001208	2.487770E-4	9.865780E-5
[0 4]	9.192340E-4	6.268347E-4	1.343427E-4	8.993521E-5
[2 2]	0.002291	0.001373	2.592512E-4	1.090212E-4

Tabella 3.14: Ampiezze relative degli intervalli di confidenza calcolati con differenti valori di tolleranza e con grado di confidenza del 95%

In particolare, il decrescere del valore di tolleranza ξ porta ad una diminuzione delle ampiezze degli intervalli di confidenza costruiti sulla base dei valori di probabilità stazionaria di stato stimata fornita in output al termine

delle 50 prove ripetute.

Il secondo esperimento consiste nello studio delle probabilità stazionarie di stato offerte da PFNetS al variare della dimensione dell'insieme W. A tal scopo analizziamo la rete in Figura 2.8 alla quale associamo i seguenti valori relativi ai tassi di firing delle transizioni: $\chi_1 = 3$, $\chi_2 = 4$, $\chi_3 = 9$, $\chi_4 = 2$. Si ottengono le seguenti intensità di traffico relative ai posti della rete:

$$\begin{cases} y_1 = \chi_4/\chi_1 = 0.66 \\ y_2 = 1 \end{cases}$$

Impostiamo i seguenti parametri di simulazione:

- Il valore di tolleranza pari a $\xi = 0.003$.
- La marcatura iniziale $M_0 = [6\,4]$.
- Il numero di prove di simulazione pari a 50.

L'esperimento consiste nell'effettuare differenti prove di simulazione a partire da una cardinalità dell'insieme W variabile da 3 ad 11.

Prob. stazionarie di stato stimate da PFNetS						
Marcatura	Prob. Analitica	$\|W\| = 3$	$\|W\| = 5$	$\|W\| = 7$	$\|W\| = 9$	$\|W\| = 11$
[0 10]	0.337232	0.336915	0.337187	0.337189	0.337228	0.337212
[1 9]	0.224821	0.224635	0.22476	0.224848	0.224792	0.224816
[2 8]	0.149881	0.149728	0.149886	0.149883	0.149887	0.149880
[3 7]	0.099920	-	0.099886	0.099918	0.099909	0.099922
[4 6]	0.066613	-	0.066602	0.066623	0.066619	0.066617
[5 5]	0.044409	-	-	0.044423	0.044411	0.044413
[6 4]	0.029606	-	-	0.029605	0.029612	0.029608
[7 3]	0.019737	-	-	-	0.019743	0.019734
[8 2]	0.013158	-	-	-	0.013160	0.013161
[9 1]	0.008772	-	-	-	-	0.008770
[10 0]	0.005848	-	-	-	-	0.005849

Tabella 3.15: Prob. stazionarie di stato stimate dal simulatore PFNetS al variare di W

La Tabella 3.15 illustra le stime di probabilità offerte dal simulatore PFNetS. Come ci si può aspettare all'aumentare della cardinalità dell'insieme W le stime di probabilità delle corrispondenti marcature convergono con una maggiore precisione al valore di probabilità stazionaria analitica. Si ottiene tale risultato in quanto una maggiore dimensione dell'insieme W corrisponde ad un maggior numero di elementi che costituiscono la matrice

Z. L'aumento della dimensione di tale matrice comporta una terminazione di simulazione caratterizzata da un numero crescente di elementi z_{ij} da valutare. In particolare, il numero di elementi z_{ij} è pari a $|W|^2 - W$ in quanto si escludono i valori appartenenti alla diagonale principale della matrice Z. Il crescere del numero di tali operazioni di controllo garantisce una maggiore precisione dei risultati offerti a discapito di un elevato costo computazionale e tempo di simulazione.

3.3 Secondo Criterio: Metodo del vettore

3.3.1 Formalizzazione del criterio.

Data una qualsiasi rete SPN, l'algoritmo proposto per il secondo criterio di arresto e quindi del metodo del vettore si sviluppa nei seguenti passi:

Passo 1: Costruzione dell'insieme W. L'insieme di marcature W è costituito da un numero arbitrario di marcature caratterizzate dai più alti valori di probabilità analitica (a meno di un costante di normalizzazione G) raggiungibili dalla marcatura iniziale M_0. Il procedimento utilizzato per la costruzione dell'insieme W è descritto nel paragrafo 3.2.3.

Definiamo con $\boldsymbol{g}$ il vettore delle probabilità di stato analitiche a meno della costante di normalizzazione G relativo alle marcature appartenenti all'insieme W. Formalmente $\boldsymbol{g}(i)$ indica la $i-esima$ componente del vettore con $1 \leq i \leq |W|$.

Osservazione. Per ogni marcatura appartenente all'insieme W indichiamo con:

- $\widetilde{\pi}_i$ la probabilità stimata dal simulatore per la marcatura $M_i \in W$, $\forall i$ con $1 \leq i \leq W$.

Se la stima delle probabilità delle marcature ottenute dalla simulazione è accurata, possiamo assumere che:

$$\frac{\widetilde{\pi_1}}{\boldsymbol{g}(1)} \backsimeq \frac{\widetilde{\pi_2}}{\boldsymbol{g}(2)} \backsimeq \ldots \backsimeq \frac{\widetilde{\pi_h}}{\boldsymbol{g}(h)}$$

dove $h = |W|$, ovvero per ogni marcatura M_i in W il rapporto tra la probabilità stazionaria stimata dal simulatore PFNetS e la analitica (a meno della costante G), corrispondenti alla marcatura M_i, è costante a meno di errori di stima.

Inoltre, il rapporto $\frac{\widetilde{\pi}_i}{\boldsymbol{g}(i)}$ può essere considerato un'approssimazione della costante di normalizzazione G.

Passo 2: Sulla base dell'osservazione, generiamo ad intervalli regolari di passi di simulazione, dapprima un vettore denotato con $\widetilde{\boldsymbol{\pi}}$ ed utilizzando tale vettore, costruiamo successivamente un vettore denominato Z.

Il vettore $\widetilde{\boldsymbol{\pi}}$ rappresenta il vettore delle probabilità stazionarie stimate da PFNetS e $\widetilde{\boldsymbol{\pi}}(i)$ indica la $i-esima$ componente del vettore stesso con $1 \leq i \leq h$.

Z è definito come un vettore di dimensione h la cui componente z_i è calcolata come segue:

$$z_{ij} = \frac{\widetilde{\boldsymbol{\pi}}(i)}{\boldsymbol{g}(i)} \quad con\, 1 \leq i \leq h$$

Passo 3: Se ogni elemento del vettore Z è diverso da zero, calcoliamo la media degli elementi del vettore come segue:

$$m = \frac{\sum_{i=1}^{h} z_i}{h}$$

In caso contrario si ritorna al Passo 2.

Passo 4: Il criterio di arresto proposto consiste nel terminare il run di simulazione quando si verifica la condizione:

$$\sum_{i=1}^{h} (z_i - m)^2 \leq \xi \tag{3.11}$$

dove ξ indica un valore di tolleranza.

In caso contrario si ritorna al Passo 2.

3.3.2 Classe TestVector

La classe TestVector, illustrata in Figura 3.11, implementa l'algoritmo sopra descritto (metodo del vettore) e si applica per le prove ripetute.

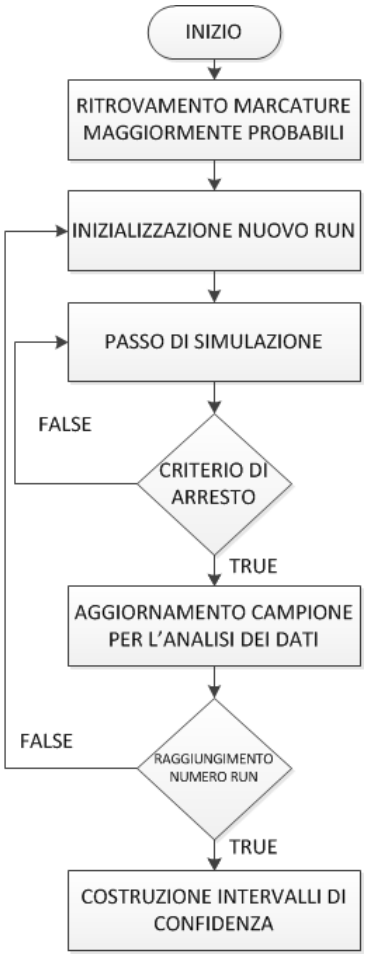

Figura 3.11: Classe TestVector

La descrizione delle fasi riportate in Figura 3.11 è identica a quanto analizzato per la classe TestMatrix in paragrafo 3.2.2.

La classe TestVector utilizza le seguenti strutture dati:

- Un valore intero (int) denotato con *NumMark*. In riferimento a quanto descritto nel paragrafo 3.3.1, tale valore rappresenta il numero di marcature appartenenti all'insieme W, denotato con h.
- Un valore intero (int) denotato con *NumSimulation*, che indica il numero degli n run i.i.d. (in riferimento al paragrafo 3.2.4) effettuati in fase di analisi dei risultati del simulatore PFNetS.
- Un valore reale positivo (double) denotato con *tolerance*, che identifica un valore di tolleranza ξ scelto in modo arbitrario e descritto nel Passo 4 dell'algoritmo descritto nel paragrafo 3.3.1.

- Una vettore di numeri reali (double) denotato con *vetrap*, che identifica il vettore Z definito nel Passo 2 dell'algoritmo. Nel dettaglio tale vettore è caratterizzato da un numero *NumMark* di elementi.

- Una matrice di valori reali (double) denotata con *matrixprob*, caratterizzata da un numero *NumMark* di righe ed un numero di colonne pari a *NumSimulation*. Ogni singola colonna della matrice contiene i valori delle probabilità di stato stimate da PFNetS e relative alle marcature dell'insieme W. Tali valori di probabilità sono ottenuti al termine di ciascun run di simulazione, termine dettato dal verificarsi del criterio (3.11). La matrice cosi definita viene utilizzata per la fase di convalida del simulatore.

- Un vettore di valori reali (double) denotato con *vetanalitic*, che identifica il vettore $\boldsymbol{g}$ descritto nel Passo 1 dell'algortimo. La dimensione del vettore è pari a *NumMark*.

- Un vettore di valori reali (double) denotato con *vetstim*, che rappresenta il vettore $\widetilde{\boldsymbol{\pi}}$ descritto nel Passo 2 dell'algortimo. La dimensione del vettore è pari a *NumMark*.

- Un vettore di stringhe (double) denotato con *mk*, che identifica l'insieme W descritto nel Passo 1 dell'algoritmo. La dimensione del vettore è pari a *NumMark*.

- Una hashtable denotata con *mark*. Tale struttura contiene lo storico dell'insieme W durante l'esecuzione dell'algoritmo per il ritrovamento delle marcature caratterizzate dai maggiori valori di probabilità analitica a meno della costante di normalizzazione G.

- Un oggetto *pn* che estende la classe MarkovianPetriNet e rappresenta una generica rete SPN di cui si vuole analizzare il comportamento mediante la simulazione.

La classe TestVector implementa i seguenti metodi:

```
private static double[] builtvect (int NumMark, double[] vetstim,
double[] vetanalitic)
```

Il metodo prende in ingresso un intero positivo *NumMark* che identifica la dimensione del vettore da restituire in output, un vettore *vetstim* che identifica il vettore $\widetilde{\boldsymbol{\pi}}$ ed infine un vettore *vetanalitic* che rappresenta il vettore $\boldsymbol{g}$. In particolare, il metodo è utilizzato per l'implementazione del Passo 2 dell'algoritmo. Il vettore restituito in output corrisponde al vettore Z.

Algoritmo 3.7 Metodo builtvect

```
private static double[] buildvect(int NumMark, double[]
vetstim, double[] vetanalitic) {
double vet[] = new double[NumMark];
    for (int i = 0; i < NumMark; i++) {
        vet[i] = vetstim[i] / vetanalitic[i];
    }
return vet;
}
```

`private static boolean ControlVect (double[][] vet, double tolerance)`

Tale metodo prende in ingresso un vettore *vet* il quale corrisponde al vettore *vetrap* restituito dalla chiamata al metodo `builtvect(NumMark,vetstim,vetanalitic)` ed un valore reale positivo *tolerance* che rappresenta la tolleranza ξ scelta arbitrariamente.

Tale metodo è utilizzato per l'implementazione del Passo 3 e del Passo 4 dell'algoritmo. Verifica che ogni elemento del vettore Z soddisfi le condizioni imposte dal criterio di arresto proposto. In tal caso, il metodo restituisce il valore booleano `true` ed in caso contrario il valore boolenano `false`.

Algoritmo 3.8 Metodo ControlVect

```
private static boolean ControlVect(double[] vet, double
tolerance) {
double tot = 0.0;
double sum = 0.0;
    for (int i = 0; i < NumMark; i++) {
        sum = sum + vet[i];
        if (vet[i] == 0) {
            return false;
        }
    }
double mean = sum / NumMark;
    for (int j = 0; j < NumMark; j++) {
        tot = tot + Math.pow(vet[j] - mean, 2);
    }
if (tot < tolerance) {
    return true;
} else {
    return false;
    }
}
```

Come di può notare dall'algoritmo 4.8 è stato necessario introdurre un controllo al fine di verificare che tutte le componenti del vettore Z siano non nulle. Tale accorgimento viene introdotto in quanto, in fase di simulazione (nello specifico nel corso dei passi iniziali di simulazione), potrebbe verificarsi una situazione del seguente tipo: i valori del vettore Z restituiscono un esito positivo alla condizione imposta dal criterio di arresto (Passo 4) sebbene una o più componenti dello stesso vettore sia pari a zero. Tale situazione si manifesta nei casi in cui alcune delle marcature appartenenti all'insieme W non sono ancora state raggiunte nei primi passi di simulazione e di conseguenza le rispettive probabilità stimate dal simulatore PFNetS ($\widetilde{\boldsymbol{\pi}}(i)$) risultano essere nulle e quindi non corrette ma comunque in grado di soddisfare la condizione $\sum_{i=1}^{|W|}(z_i - m)^2 \leq \xi$.

In particolare, la quantità $\sum_{i=1}^{|W|}(z_i - m)^2$ rappresenta una valore di devianza (o somma dei quadrati della media) ovvero, un indice di dispersione dei dati caratterizzato dalla seguente espressione generale $\sum_{i=1}^{n}(x_i - \mu)^2$ dove μ indica la media dei dati.

I metodi che seguono corrispondono sia nella firma che nell'implementazione ai relativi metodi descritti in paragrafo 3.2.2.

```
private static double[][] CompileMatrixProb (double[][] matrixprob,
double[] vetstim, int j)
```

```
private static double computeMeans (double[][] matrixprob, int
i)
```

```
private static double computeVariance (double[][] matrixprob,
int i, double mean)
```

```
private static double computeAmpIC (double[][] matrixprob, double
mean, double variance, int i)
```

3.3.3 Complessità e miglioramenti

Come descritto nel paragrafo 3.2.5, il costo computazionale relativo all'algoritmo per il ritrovamento delle marcature raggiungibili dallo stato iniziale M_0 caratterizzate dai maggiori valori di probabilità non normalizzata è pari a $O(n \cdot \log(m))$ dove n è la cardinalità del vettore W ed m è la dimensione della hashtable.

Per quel che riguarda l'algoritmo descritto nel paragrafo 3.3.1, la fase relativa alla costruzione del vettore Z (mediante il metodo ***builtvector***) prevede un costo computazionale pari a $O(NumMark)$ dove $NumMark$ identifica la cardinalità dell'insieme W. La fase relativa al controllo imposto

dal secondo criterio di arresto (tramite il metodo `ControlVector`) prevede un costo computazionale pari a $2 \cdot O(NumMark)$ dato che il vettore viene visitato due volte. In tal modo, si ottiene un costo computazionale complessivo relativo ad ogni singolo run pari a $NumStep * 3 * O(NumMark)$ dove *NumStep* indica il numero totale di passi di valutazione necessari per terminare ogni singolo run in rispetto del criterio di arresto proposto.

E' possibile diminuire il costo computazionale relativamente agli algoritmi sopracitati. In particolare, per quel che riguarda l'algoritmo di costruzione e di controllo del vettore `vetrap`, è possibile effettuare il controllo relativo alle componenti del vettore stesso successivamente ad ogni singolo passo di costruzione ottenendo in tal modo un costo computazionale inferiore e pari a $2 \cdot O(NumMark)$ (in quanto è necessaria una prima visita per la costruzione di Z con il controllo che ogni componente sia non nulla ed il calcolo della media, ed una seconda visita per il calcolo del valore di devianza) per un singolo passo di valutazione e quindi per la singola run si ottiene un costo computazionale pari a $NumStep \cdot 2 \cdot O(NumMark)$. Tuttavia, tale modifica, non viene presa in considerazione nella costruzione dell'algoritmo presentato nel nostro elaborato al fine di fornire la possibilita di analizzare e rendere visibili ad ogni singolo passo di analisi i cambiamenti del vettore Z.

3.3.4 Convalida ed analisi dei risultati relative a SPN chiuse.

In questo paragrafo riportiamo i risultati ottenuti per le fasi di convalida ed analisi dei risultati forniti da PFNetS per SPN chiuse utilizzando per l'arresto di ogni singola run di simulazione il metodo del vettore. In particolare, la procedura utilizzata è la stessa per il procedimento definito nel paragrafo 3.2.6. Gli esempi che seguono sono relativi a simulazioni che utilizzano la rete rappresentata in Figura 2.8 nel paragrafo 2.5. L'analisi viene effettuata studiando il modello al variare delle intensità di traffico relativi ai posti.

3.3.4.1 Primo esempio

In questo esempio analizziamo la rete in Figura 2.8 con i seguenti valori relativi ai tassi di firing delle transizioni: $\chi_1 = 10$, $\chi_2 = 1$, $\chi_3 = 25$, $\chi_4 = 2$. Successivamente all'applicazione del teorema di Coleman Henderson et. al. si ottiene una soluzione in forma prodotto con i seguenti valori di intensità di traffico:

$$\begin{cases} \rho_1 = \chi_4/\chi_1 = 0.2 \\ \rho_2 = 1 \end{cases}$$

Effettuiamo una simulazione a partire da un valore di tolleranza $\xi = 1.0E-6$, da una marcatura iniziale $M_0 = [5\,3]$, da un numero di marcature appartenenti all'insieme $|W| = 4$ ed infine da un numero di prove di simulazione pari a 50.

L'algoritmo utilizzato per il ritrovamento delle marcature raggiungibili a partire da M_0 ed aventi i più alti valori di probabilità analitica non normalizzata restituisce un insieme W con le marcature illustrate in Tabella 3.16.

Marcatura	Prob. analitica a meno di G
[3 5]	0.008
[1 7]	0.2
[0 8]	1.0
[2 6]	0.04

Tabella 3.16: Marcature dell'insieme W

Il simulatore PFNetS restituisce il seguente output.

```
Arresto Simulazione 1 al passo:  10687000
Tempo di simulazione 1781346.2806370836
Vettore Z
0.7999656010823337 0.8009795159801693 0.7998185799725862
0.7997451560626172

Risultati:
[ 3 5 ] con probabilità stimata:  0.006399724808658671
[ 1 7 ] con probabilità stimata:  0.16019590319603386
[ 0 8 ] con probabilità stimata:  0.7998185799725862
[ 2 6 ] con probabilità stimata:  0.03198980624250469

Arresto Simulazione 2 al passo:  6471000
Tempo di simulazione 1078571.78544874
Vettore Z
0.7987679442018489 0.799892087531305 0.8000381125759688
0.7995318415121688

Risultati:
[ 3 5 ] con probabilità stimata:  0.006390143553614793
[ 1 7 ] con probabilità stimata:  0.15997841750626102
[ 0 8 ] con probabilità stimata:  0.8000381125759688
[ 2 6 ] con probabilità stimata:  0.03198127366048676
                                  .
                                  .
                                  .
Arresto Simulazione 49 al passo:  222000
Tempo di simulazione 37003.1991109097
Vettore Z
0.8000491754137335 0.8008177909527703 0.7998305512396712
0.8003343428016859

Risultati:
[ 3 5 ] con probabilità stimata:  0.00640732797456684
[ 1 7 ] con probabilità stimata:  0.1599152909322307
[ 0 8 ] con probabilità stimata:  0.8000399007479516
[ 2 6 ] con probabilità stimata:  0.03201852196508391

Arresto Simulazione 50 al passo:  873000
Tempo di simulazione 145475.88325013212
Vettore Z
0.7990796169351256 0.8001628196142337 0.7999855477694572
0.799145773035935

Risultati:
[ 3 5 ] con probabilità stimata:  0.006392636935481007
[ 1 7 ] con probabilità stimata:  0.16003256392284676
[ 0 8 ] con probabilità stimata:  0.7999855477694572
[ 2 6 ] con probabilità stimata:  0.031965830921437406
```

Figura 3.12: Output simulazione

Sebbene l'output fornito da PFNetS sia relativo all'esecuzione di 50 run ripetute, la Figura 3.12 riporta i risultati di un numero limitato di run. Per ogni run, riportiamo il passo ed il corrispettivo tempo di simulazione relativi alla prima occorenza del vettore Z che rispetta il vincolo richiesto dal criterio di arresto. Inoltre, mostriamo le stime delle probabilità stazionarie di stato relative alle marcature appartenenti all'insieme W.

Al termine dell'esecuzione delle 50 prove, il campione prodotto a partire dalle osservazioni effettuate, è utilizzato per la costruzione degli intervalli di confidenza. In particolare, per ogni marcatura oggetto di studio il simulatore PFNetS restituisce le informazioni relative alla media, varianza ed ampiezza dell'intervallo costruito con differenti gradi di confidenza. Tali risultati sono illustrati in Figura 3.13.

```
Resoconto Fine Prove Ripetute

Marcatura [ 3 5 ] media95:  0.006399736947266655
Marcatura [ 3 5 ] varianza95:  4.123367824076298E-11
Marcatura [ 3 5 ] ampiezzaIC95:  5.562435401908342E-4

Marcatura [ 1 7 ] media95:  0.16000542149759
Marcatura [ 1 7 ] varianza95:  8.937401831726997E-9
Marcatura [ 1 7 ] ampiezzaIC95:  3.275457803356257E-4

Marcatura [ 0 8 ] media95:  0.7999972216641367
Marcatura [ 0 8 ] varianza95:  9.936150876295084E-9
Marcatura [ 0 8 ] ampiezzaIC95:  6.90751242627813E-5

Marcatura [ 2 6 ] media95:  0.03199600673956829
Marcatura [ 2 6 ] varianza95:  5.921594757602258E-10
Marcatura [ 2 6 ] ampiezzaIC95:  4.216233230880776E-4

Marcatura [ 3 5 ] media99:  0.006399736947266655
Marcatura [ 3 5 ] varianza99:  4.123367824076298E-11
Marcatura [ 3 5 ] ampiezzaIC99:  7.310629385365173E-4

Marcatura [ 1 7 ] media99:  0.16000542149759
Marcatura [ 1 7 ] varianza99:  8.937401831726997E-9
Marcatura [ 1 7 ] ampiezzaIC99:  4.3048873986977865E-4

Marcatura [ 0 8 ] media99:  0.7999972216641367
Marcatura [ 0 8 ] varianza99:  9.936150876295084E-9
Marcatura [ 0 8 ] ampiezzaIC99:  9.078444903125051E-5

Marcatura [ 2 6 ] media99:  0.03199600673956829
Marcatura [ 2 6 ] varianza99:  5.921594757602258E-10
Marcatura [ 2 6 ] ampiezzaIC99:  5.541335103442562E-4
```

Figura 3.13: Resoconto prove ripetute

Utilizzando il procedimento descritto nel paragrafo 3.2.4, si ottengono i seguenti risultati:

Confidenza 95%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[3 5]	0.0064	0.006399	0.006397	0.006401	5.562435E-4	SI
[1 7]	0.16	0.160005	0.159970	0.160031	3.2754578E-4	SI
[0 8]	0.8	0.799997	0.799969	0.800024	6.907512E-5	SI
[2 6]	0.032	0.031996	0.031989	0.032002	4.216233E-4	SI

Tabella 3.17: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[3 5]	0.0064	0.006399	0.006397	0.006402	7.310629E-4	SI
[1 7]	0.16	0.160005	0.159968	0.160039	4.304887E-4	SI
[0 8]	0.8	0.799997	0.799960	0.800033	9.078444E-5	SI
[2 6]	0.032	0.031996	0.031987	0.032004	5.541335E-4	SI

Tabella 3.18: Risultati di simulazione. Intervalli con grado di confidenza del 99%

Come si può notare dalle Tabella 3.17 e 3.18 i valori analitici relativi alle probabilità stazionarie di stato cadono all'interno dei rispettivi intervalli di confidenza centrati sulla media.

3.3.4.2 Secondo esempio

Consideriamo l'esempio di SPN illustrato in Figura 2.8 modificando i valori dei tassi di firing delle transizioni come segue: $\chi_1 = 7$, $\chi_2 = 25$, $\chi_3 = 49$, $\chi_4 = 5$ al fine di analizzare la stessa rete con valori di intensità di traffico maggiori rispetto all'esempio 3.3.4.1.

$$\begin{cases} \rho_1 = \chi_4/\chi_1 = 0.71428 \\ \rho_2 = 1 \end{cases}$$

Assumiamo i seguenti parametri di simulazione:

- Il valore di tolleranza $\xi = 1.0E - 6$.
- La marcatura iniziale $M_0 = [5\,3]$.

- Il numero di marcature appartenenti all'insieme $|W| = 7$.
- Il numero di prove di simulazione pari a 50.

La Tabella 3.19 illustra le marcature maggiormente probabili raggiungibili a partire da M_0.

Marcatura	Prob. analitica a meno di G
[4 4]	0.260308
[2 6]	0.510204
[0 8]	1.0
[1 7]	0.714285
[3 5]	0.364431
[5 3]	0.185934
[6 2]	0.132810

Tabella 3.19: Marcature dell'insieme W

La Tabella 3.20 e 3.21 illustrano i risultati ottenuti successivamente all'esecuzione delle prove ripetute e riportano l'esito relativo alla fase di analisi dei risultati a partire da differenti gradi di confidenza.

Confidenza 95%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[4 4]	0.078156	0.078165	0.078136	0.078194	7.389418E-4	SI
[2 6]	0.153187	0.153184	0.153142	0.153226	5.505970E-4	SI
[0 8]	0.300246	0.300201	0.300120	0.300283	5.452063E-4	SI
[1 7]	0.214462	0.214535	0.214471	0.214599	5.946602E-4	NO
[3 5]	0.109419	0.109431	0.109395	0.109467	6.532111E-4	SI
[5 3]	0.055826	0.055812	0.055789	0.055834	7.963532E-4	SI
[6 2]	0.039875	0.039863	0.039844	0.039883	9.850449E-4	SI

Tabella 3.20: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[4 4]	0.078156	0.078165	0.078127	0.078203	9.711807E-4	SI
[2 6]	0.153187	0.153184	0.153128	0.153239	7.236417E-4	SI
[0 8]	0.300246	0.300201	0.300094	0.300309	7.1655692E-4	SI
[1 7]	0.214462	0.214535	0.214451	0.214619	7.8155348E-4	SI
[3 5]	0.109419	0.109431	0.109384	0.109478	8.585060E-4	SI
[5 3]	0.055826	0.055812	0.055783	0.055841	0.001046	SI
[6 2]	0.039875	0.039863	0.039837	0.039889	0.001294	SI

Tabella 3.21: Risultati di simulazione. Intervalli con grado di confidenza del 99%

A differenza dell'esempio riportato nel paragrafo 3.3.4.1, osserviamo dalla Tabella 3.20 che il valore di probabilità analitica stazionaria relativa alla marcatura [1 7] non cade all'interno dell'intervallo di confidenza del 95%. Ma come osserviamo in Tabella 3.21 per un livello di confidenza del 99% tutti i valori di probabilità analitica cadono nei rispettivi intervalli. Quindi le fasi di convalida ed analisi dei risultati per tale esempio è verificata.

3.3.5 Convalida ed analisi dei risultati relative a SPN aperte.

La fase di convalida del simulatore PFNetS relativa a reti di Petri stocastiche aperte segue la stessa procedura descritta nel paragrafo 3.2.7. In questo paragrafo, quindi, vengono prese in esame SPN riconducibili a reti aperte di Jackson. Tale accorgimento, permette di calcolare i valori di probabilità di stato stazionaria analitica relativi alle marcature dell'insieme W come prodotto di funzioni degli stati relativi ai singoli posti della rete.

In particolare, il primo esempio prevede l'analisi di una rete SPN aperta con posti a cui sono associate basse intensità di traffico ed il secondo studia il comportamento della stessa rete con posti a cui si assegna un' intensità di traffico maggiore.

3.3.5.1 Primo esempio

Analizziamo la rete aperta in Figura 3.6 con i seguenti valori relativi ai tassi di firing delle transizioni: $\chi_1 = 1.5$, $\chi_2 = 3.0$, $\chi_3 = 11.0$, $\chi_4 = 19.0$, $\chi_5 = 9.0$, $\chi_6 = 2.0$, $\chi_7 = 10.0$, $\chi_8 = 3.0$. I valori relativi alle intensità di traffico associate ai posti della rete sono:

$$\begin{cases} \rho_1 = \frac{\chi_1\chi_3}{\chi_4\chi_2} = 0.28947 \\ \rho_2 = \frac{\chi_2}{\chi_3} = 0.2727 \\ \rho_3 = \frac{\chi_5\chi_1}{\chi_4\chi_6} = 0.3552 \\ \rho_4 = \frac{\chi_1\chi_5}{\chi_4\chi_8} = 0.236842 \\ \rho_5 = \frac{\chi_1}{\chi_7} = 0.15 \end{cases}$$

Effettuiamo una simulazione a partire da un valore di tolleranza $\xi = 1.0E-6$, da una marcatura iniziale $M_0 = [1\,1\,0\,1\,0]$, da un numero di marcature appartenenti all'insieme W pari a 9 e da un numero di prove di simulazione pari a 50.

Marcatura	Prob. analitica a meno di G
[0 1 0 0 0]	0.272727
[0 0 1 0 0]	0.355263
[0 1 1 0 0]	0.096889
[0 0 2 0 0]	0.126211
[1 0 0 0 0]	0.289473
[1 0 1 0 0]	0.102839
[0 0 0 0 1]	0.15
[0 0 0 1 0]	0.236842
[0 0 0 0 0]	1.0

Tabella 3.22: Marcature dell'insieme W

Confidenza 95%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[0 1 0 0 0]	0.058941	0.058918	0.058890	0.058946	9.540080E-4	SI
[0 0 1 0 0]	0.076779	0.076785	0.076754	0.076817	8.169170E-4	SI
[0 1 1 0 0]	0.020939	0.020941	0.020932	0.020950	8.6938837E-4	SI
[0 0 2 0 0]	0.027276	0.027284	0.027268	0.027299	0.001161	SI
[1 0 0 0 0]	0.062560	0.06256	0.062528	0.062598	0.001123	SI
[1 0 1 0 0]	0.022225	0.022233	0.022222	0.022243	9.593958E-4	SI
[0 0 0 0 1]	0.032417	0.032416	0.032399	0.032433	0.001040	SI
[0 0 0 1 0]	0.051186	0.051187	0.051162	0.051211	9.544521E-4	SI
[0 0 0 0 0]	0.216119	0.216055	0.215973	0.216138	7.652613E-4	SI

Tabella 3.23: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[0 1 0 0 0]	0.058941	0.058918	0.058881	0.058955	0.001250	SI
[0 0 1 0 0]	0.076779	0.076785	0.076744	0.076827	0.001073	SI
[0 1 1 0 0]	0.020939	0.020941	0.020929	0.020953	0.001142	SI
[0 0 2 0 0]	0.027276	0.027284	0.027263	0.027304	0.001526	SI
[1 0 0 0 0]	0.062560	0.062563	0.062517	0.062609	0.001476	SI
[1 0 1 0 0]	0.022225	0.022233	0.022219	0.022247	0.001260	SI
[0 0 0 0 1]	0.032417	0.032416	0.032394	0.032438	0.001360	SI
[0 0 0 1 0]	0.051186	0.051187	0.051155	0.051219	0.001254	SI
[0 0 0 0 0]	0.216119	0.216055	0.215947	0.216164	0.001005	SI

Tabella 3.24: Risultati di simulazione. Intervalli con grado di confidenza del 99%

Similmente ai risultati ottenuti a partire dal metodo della matrice nel paragrafo 3.2.7.1, considerando bassi valori di intensità di traffico relativi ai posti della rete oggetto di studio, notiamo dalle Tabelle 3.23 e 3.24 che valori analitici relativi alle probabilità stazionarie di stato della marcature dell'insieme W cadono all'interno dei rispettivi intervalli di confidenza. Tali intervalli sono centrati sulla media del campione e vengono costruiti con grado di confidenza del 95% e 99%.

3.3.5.2 Secondo esempio

A partire dalla rete proposta nell'esempio 3.3.5.1, impostiamo i seguenti valori relativi ai tassi di firing delle transizioni: $\chi_1 = 1.0$, $\chi_2 = 5.0$, $\chi_3 = 9.0$, $\chi_4 = 3.0$, $\chi_5 = 7.0$, $\chi_6 = 4.0$, $\chi_7 = 1.7$, $\chi_8 = 3.8$ ottenendo in tal modo le seguenti intensità di traffico associate ai posti della rete:

$$\begin{cases} \rho_1 = \frac{\chi_1\chi_3}{\chi_4\chi_2} = 0.6 \\ \rho_2 = \frac{\chi_2}{\chi_3} = 0.555 \\ \rho_3 = \frac{\chi_5\chi_1}{\chi_4\chi_6} = 0.583 \\ \rho_4 = \frac{\chi_1\chi_5}{\chi_4\chi_8} = 0.614 \\ \rho_5 = \frac{\chi_1}{\chi_7} = 0.588 \end{cases}$$

Inpostiamo i seguenti parametri di simulazione:

- Il valore di tolleranza $\xi = 2.0E - 8$.
- La marcatura iniziale $M_0 = [1\,1\,0\,1\,0]$.
- Il numero di marcature appartenenti all'insieme $|W| = 11$.
- Il numero di prove di simulazione pari a 50.

Marcatura	Prob. analitica a meno di G
[0 0 1 1 0]	0.358187
[0 0 0 0 0]	1.0
[0 0 0 1 0]	0.614035
[1 0 0 1 0]	0.368421
[2 0 0 0 0]	0.36
[0 0 0 2 0]	0.377039
[0 0 1 0 0]	0.583333
[0 0 0 1 1]	0.361197
[0 1 0 0 0]	0.555555
[1 0 0 0 0]	0.6
[0 0 0 0 1]	0.588235

Tabella 3.25: Marcature dell'insieme W

Confidenza 95%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[0 0 1 1 0]	0.004216	0.004216	0.004209	0.00422	0.003246	SI
[0 0 0 0 0]	0.011772	0.011756	0.011737	0.011776	0.003339	SI
[0 0 0 1 0]	0.007228	0.007238	0.007226	0.007250	0.003297	SI
[1 0 0 1 0]	0.004337	0.004339	0.004333	0.004345	0.002938	SI
[2 0 0 0 0]	0.004238	0.004237	0.004231	0.004243	0.003001	SI
[0 0 0 2 0]	0.004438	0.004439	0.004432	0.004445	0.003105	SI
[0 0 1 0 0]	0.006867	0.006871	0.006859	0.006882	0.003244	SI
[0 0 0 1 1]	0.004252	0.004254	0.004248	0.004260	0.002879	SI
[0 1 0 0 0]	0.006540	0.006534	0.006522	0.006546	0.003776	SI
[1 0 0 0 0]	0.007063	0.007067	0.007055	0.007079	0.003302	SI
[0 0 0 0 1]	0.006924	0.006914	0.006904	0.006925	0.003103	SI

Tabella 3.26: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
Marcatura	Prob. Analitica	Prob. Stimata	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
[0 0 1 1 0]	0.004216	0.004216	0.004207	0.004225	0.004266	SI
[0 0 0 0 0]	0.011772	0.011756	0.011730	0.011782	0.004389	SI
[0 0 0 1 0]	0.007228	0.007238	0.007222	0.007253	0.004334	SI
[1 0 0 1 0]	0.004337	0.004339	0.004331	0.004347	0.003862	SI
[2 0 0 0 0]	0.004238	0.004237	0.004229	0.004245	0.003944	SI
[0 0 0 2 0]	0.004438	0.004439	0.004429	0.004448	0.004081	SI
[0 0 1 0 0]	0.006867	0.006871	0.006856	0.006885	0.004263	SI
[0 0 0 1 1]	0.004252	0.004254	0.004246	0.004262	0.003785	SI
[0 1 0 0 0]	0.006540	0.006534	0.006518	0.006550	0.004963	SI
[1 0 0 0 0]	0.007063	0.007067	0.007052	0.007082	0.004340	SI
[0 0 0 0 1]	0.006924	0.006914	0.006900	0.006928	0.004079	SI

Tabella 3.27: Risultati di simulazione. Intervalli con grado di confidenza del 99%

Come si può notare dalle Tabelle 3.26 e 3.27, anche considerando valori di intensità di traffico elevati, le fase di convalida ed analisi dei risultati del simulatore restituiscono un esito positivo. Infatti i valori analitici relativi alle probabilità stazionarie di stato cadono all'interno dei rispettivi intervalli

di confidenza centrati sulla media del campione e con livelli di confidenza del 95% e del 99%.

3.3.6 Precisione

Lo scopo di questo paragrafo, similmente a quanto descritto nel paragrafo 3.2.8, consiste nell'analizzare come il variare del valore di tolleranza ξ e della dimensione dell'insieme W possa influenzare la precisione relativa alle stime di probabilità fornite in output da PFNetS.
In particolare, a partire dalla SPN chiusa con insieme di raggiungibilità finito illustrata in Figura 2.8, effettuiamo diverse simulazioni ciascuna composta da 50 run. Ogni simulazione viene eseguita a partire da un valore di tolleranza variabile tra il valore $1.0E-7$ ed il valore $2.0E-5$.

Ogni simulazione è caratterizzata dai seguenti parametri:

- La marcatura iniziale $M_0 = [2\,2]$.
- Il numero di marcature appartenenti all'insieme $|W| = 5$.
- I seguenti valori relativi ai tassi di firing delle transizioni $\chi_1 = 6$, $\chi_2 = 25$, $\chi_3 = 36$, $\chi_4 = 5$.

Otteniamo le seguenti intensità di traffico relativi ai posti della rete:

$$\begin{cases} \rho_1 = \chi_4/\chi_1 = 0.83 \\ \rho_2 = 1 \end{cases}$$

L'algoritmo utilizzato per il ritrovamento delle marcature raggiungibili a partire da M_0 ed aventi i più alti valori di probabilità analitica non normalizzata restituisce un insieme W con le seguenti marcature:

Marcatura	Prob. analitica a meno di G	Prob. Analitica
[3 1]	0.578703	0.161256
[1 3]	0.833333	0.232208
[0 4]	1.0	0.278650
[4 0]	0.482253	0.134380
[2 2]	0.694444	0.193500

Tabella 3.28: Marcature dell'insieme W

Le figure 3.14÷3.18 illustarno in forma grafica i risultati ottenuti in output da PFNetS per le probabilità stazionarie di stato stimate per le marcature dell'insieme W, al variare del parametro di tolleranza ξ. Inoltre, in ogni grafico, è rappresentata la retta di equazione $y = PA_{M_i}$ descritta nel paragrafo 3.2.8 .

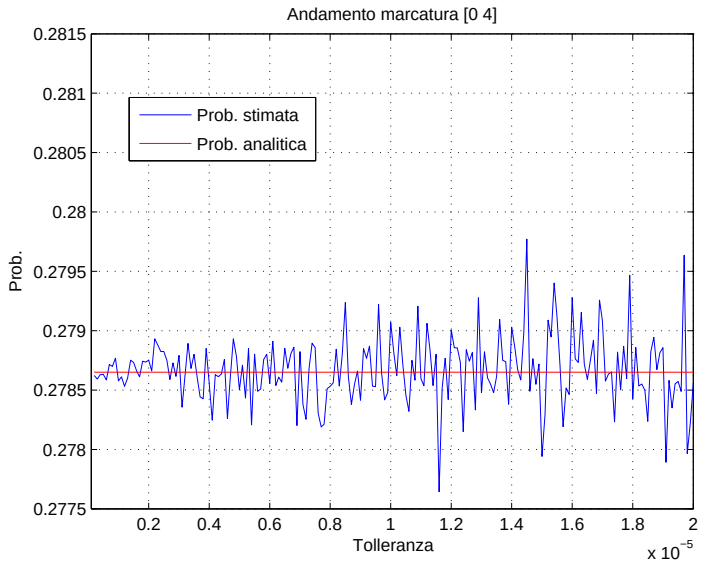

Figura 3.14: Andamento della probabilità stimata relativa alla marcatura [0 4]

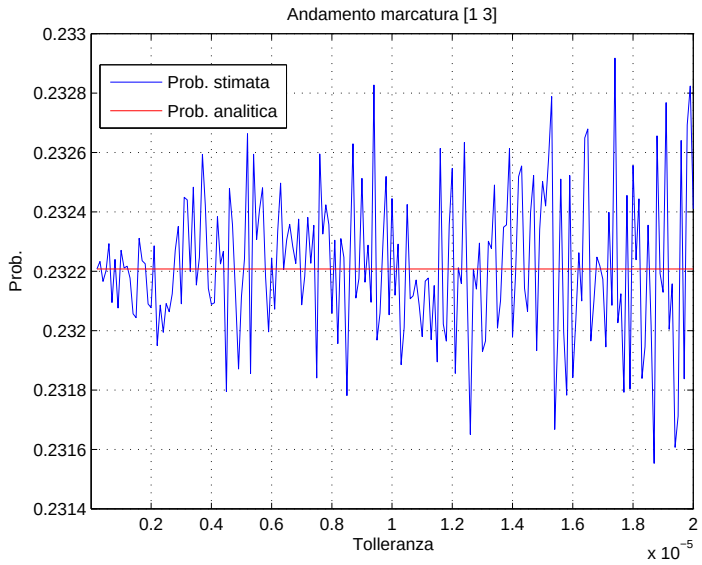

Figura 3.15: Andamento della probabilità stimata relativa alla marcatura [1 3]

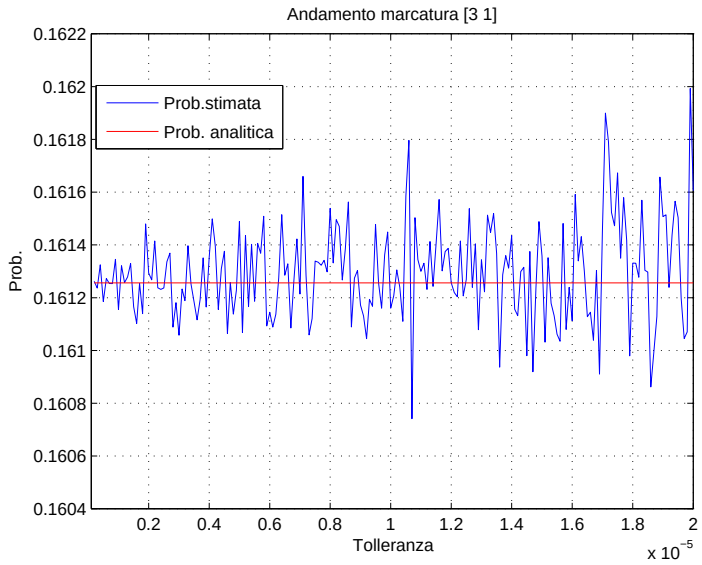

Figura 3.16: Andamento della probabilità stimata relativa alla marcatura [3 1]

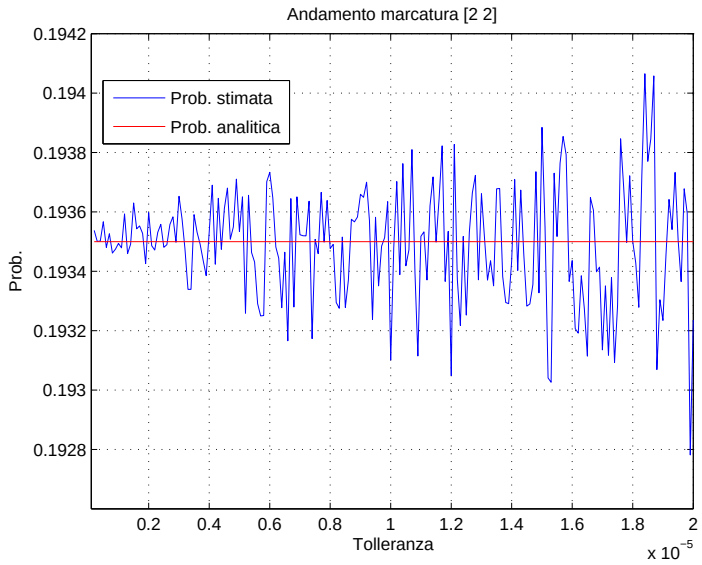

Figura 3.17: Andamento della probabilità stimata relativa alla marcatura [2 2]

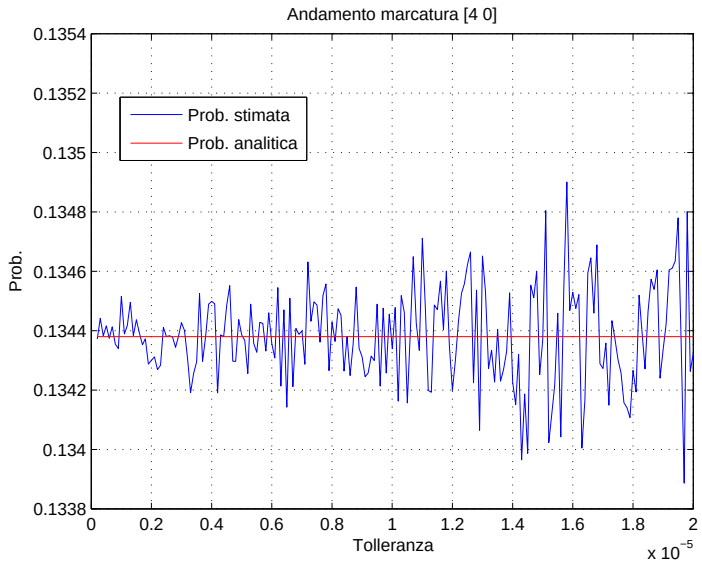

Figura 3.18: Andamento della probabilità stimata relativa alla marcatura [4 0]

Similmente ai risultati ottenuti nel paragrafo 3.2.8, come possiamo notare dai grafici in Figura 3.13÷3.18, al crescere del valore della tolleranza ξ, i valori delle probabilità stimate da PFNetS e relative alle marcature dell'insieme W sono caratterizzate da una maggiore divergenza rispetto ai corrispettivi valori che descrivono la probabilità calcolata a partire dal modello analitico. Possiamo quindi concludere che è possibile avere stime più accurate di probabilità stazionaria di stato diminuendo il valore di tolleranza ξ a discapito di un elevate tempo di simulazione.

Marcatura	AmpiezzaIC $\xi = 2.0E-5$	AmpiezzaIC $\xi = 8.0E-6$	AmpiezzaIC $\xi = 2.0E-6$	AmpiezzaIC $\xi = 1.0E-7$
[3 1]	0.003394	0.002681	0.001166	3.200132E-4
[1 3]	0.003787	0.002294	0.001157	2.607323E-4
[0 4]	0.003744	0.002062	0.001045	2.474915E-4
[2 2]	0.003564	0.002266	0.001240	2.607323E-4
[4 0]	0.004200	0.002680	0.001516	3.135515E-4

Tabella 3.29: Ampiezze relative degli intervalli di confidenza calcolati con differenti valori di tolleranza e con grado di confidenza del 95%

Come possiamo notare dalla Tabella 3.29, il descrescere del valore di ξ porta ad una diminuzione delle ampiezze degli intervalli di confidenza costruiti sulla base dei valori di probabilità stazionaria di stato stimata fornita in

output al termine delle 50 prove ripetute

Per quel che concerne il secondo esperimento, analizziamo il comportamento della SPN in Figura 2.8 con i seguenti valori relativi ai tassi di firing delle transizioni: $\chi_1 = 3$, $\chi_2 = 4$, $\chi_3 = 9$, $\chi_4 = 2$. Si ottengono le seguenti intensità di traffico relativi ai posti della rete:

$$\begin{cases} \rho_1 = \chi_4/\chi_1 = 0.66 \\ \rho_2 = 1 \end{cases}$$

Impostiamo i seguenti parametri di simulazione:

- Il valore di tolleranza $\xi = 1.0E - 6$.
- La marcatura iniziale $M_0 = [6\,4]$.
- Il numero di prove di simulazione pari a 50.

Il test prevede l'analisi dei risultati ottenuti a partire da differenti simulazioni, ognuna delle quale caratterizzata da un numero di marcature appartenenti all'insieme W variabile da 3 ad 11.

Prob. stazionarie di stato stimate da PFNetS						
Marcatura	Prob. Analitica	$\|W\| = 3$	$\|W\| = 5$	$\|W\| = 7$	$W\| = 9$	$W\| = 11$
[0 10]	0.337232	0.337416	0.337242	0.337280	0.337221	0.337240
[1 9]	0.224821	0.224958	0.224871	0.224814	0.224824	0.224807
[2 8]	0.149881	0.149929	0.149871	0.149886	0.149884	0.149885
[3 7]	0.099920	-	0.099888	0.099918	0.099933	0.099917
[4 6]	0.066613	-	0.066610	0.066610	0.066615	0.066620
[5 5]	0.044409	-	-	0.044407	0.044406	0.044409
[6 4]	0.029606	-	-	0.029606	0.029612	0.029607
[7 3]	0.019737	-	-	-	0.019739	0.019737
[8 2]	0.013158	-	-	-	0.013161	0.013156
[9 1]	0.008772	-	-	-	-	0.008771
[10 0]	0.005848	-	-	-	-	0.005847

Tabella 3.30: Prob. stazionarie di stato stimate dal simulatore PFNetS al variare di W

A partire dalla Tabella 3.30, contenente le stime di probabilità stazionarie di stato offerte da PFNetS, è possibile comprendere come al crescere della cardinalità dell'insieme W i valori delle grandezze considerate convergano con una maggiore precisione al rispettivo valore di probabilità stazionaria analitica. Si raggiunge tale risultato in quanto un aumento della dimensione dell'insieme W corrisponde ad un maggior numero di elementi che costituiscono

il vettore Z. La crescita lineare della dimensione di tale vettore comporta una terminazione di simulazione caratterizzata da un costante aumento del numero di elementi z_{ij} da valutare e di conseguenza da una crescita lineare del numero di operazioni di controllo da effettuare a partire da quest'ultimi. Ciò garantisce, a discapito di un elevato costo computazionale e tempo di simulazione, una miglior precisione dei risultati offerti dal simulatore PFNetS.

3.4 Confronto tra i due criteri

In questo paragrafo effettuiamo un confronto tra i metodi di arresto proposti sulla base dei costi computazionali e riportiamo infine un esempio di due differenti simulazioni, effettuate a partire dalla stessa SPN, che utilizzano rispettivamente i due metodi proposti per la loro terminazione.

La Tabella 3.31 illustra un riepilogo dei costi computazionali per ogni run di simulazione dei metodi che implementano i criteri di arresto proposti:

	Metodo della matrice
Ritrovamento insieme W	$O(NumMark \cdot \log m)$
Costruzione	$O\left(NumMark^2\right)$
Verifica criterio	$O\left(NumMark^2\right)$
Costo analisi totale	$NumStep \cdot 2 \cdot O\left(NumMark^2\right)$
	Metodo del vettore
Ritrovamento insieme W	$O(NumMark \cdot \log m)$
Costruzione	$O\left(NumMark\right)$
Verifica criterio	$2 \cdot O\left(NumMark\right)$
Costo analisi totale	$NumStep \cdot 3 \cdot O\left(NumMark\right)$

Tabella 3.31: Confronto costo computazionale

dove m è la dimensione della hashtable utilizzata, *NumMark* denota il numero di marcature dell'insieme W ed infine *NumStep* identifica il numero di passi necessari al fine di terminare una simulazione.

Osserviamo dalla Tabella 3.31 che il costo computazionale relativo al metodo del vettore che implementa il secondo criterio di arresto (metodo del vettore) è inferiore e di conseguenza meno oneroso rispetto al primo criterio di arresto (metodo della matrice).

Non è possibile effettuare un'analisi comparativa in quanto *NumStep* rappresenta una grandezza che varia sulla base della tolleranza presa in considerazione e sulla base della velocità di convergenza della rete verso una buona

stima delle probabilità di stato prese in esame (insieme W). Tali fattori dipendono esclusivamente dalla sequenza di firing nel corso della simulazione.

Sulla base dei risultati ottenuti, possiamo derivare un'indicazione della relazione tra gli ordini di grandezza del costo computazionale dei due metodi in relazione al grado di tolleranza utilizzata nei due metodi di arresto. In generale, al fine di ottenere circa gli stessi risultati (relativi agli output forniti dai due differenti criteri di arresto) a partire dalla medesima struttura di rete SPN analizzata, è necessario definire le tolleranze ξ_1 (tolleranza scelta per il metodo della matrice) ed ξ_2 (tolleranza scelta per il metodo del vettore) nel seguente modo:

$$\xi_2 \simeq (\xi_1)^2$$

in quanto il metodo del vettore, a differenza del metodo della matrice utilizza per l'arresto della simulazione la condizione (3.11) che prevede di elevare al quadrato le differenze utilizzate per l'arresto dei singoli run di simulazione.

Inoltre, per quel che concerne l'utilizzo di alti valori di tolleranza, il metodo della matrice è da considerarsi migliore in quanto, sebbene il numero di elementi da controllare nella matrice Z (pari ad $NumMark^2 - NumMark$) sia superiore al numero di elementi da controllare nel vettore Z ($NumMark$ nello specifico), il primo metodo prevede che gli elementi convergano verso un valore pari a 1 (il quale identifica matematicamente per il rapporto analizzato un indice di convergenza ideale) garantendo in tal modo la correttezza delle grandezze analizzate. Il secondo criterio, invece, effettua un calcolo di un indice di devianza ottenuto a partire dai valori del vettore Z. Come analizzato nel paragrafo 3.3.2, l'utilizzo di tale indice può risultare problematico in quanto i valori utilizzati nel calcolo, possono divergere dal valore analitco ma restituire comunque un esito positivo.

Riassumendo, si predilige l'utilizzo del metodo della matrice nel caso in cui si vuole ottenere una migliore accuratezza nei risultati a discapito di un elevato costo computazionale, mentre, è preferibile utilizzare il metodo del vettore nel caso in cui si preferisca ottenere un costo computazionale limitato a discapito di risultati meno precisi.

3.4.1 Esempio

Al fine di effettuare un confronto relativo ai risultati ottenuti sulla base dei due differenti criteri di arresto utilizzati, prendiamo in considerazione la rete SPN in Figura 2.8 con i seguenti valori relativi ai tassi di firing delle transizioni: $\chi_1 = 6$, $\chi_2 = 25$, $\chi_3 = 36$, $\chi_4 = 5$. A partire da una marcatura iniziale $M_0 = [1\,2]$ effetuiamo due differenti prove, ognuna delle quali prevede l'utilizzo di un diverso criterio di arresto. La cardinalità dell'insieme W, è scelta con un numero di marcature pari a 4. Si sceglie un valore di tolleranza

$\xi = 3.0E-3$ per il metodo della matrice ed un valore di tolleranza $\xi = 1.0E-6$ per il metodo del vettore. Per ciascuna prova vengono effettuati 50 run di simulazione.

L'algoritmo utilizzato per il ritrovamento delle marcature raggiungibili a partire da M_0 ed aventi i maggiori valori di probabilità analitica non normalizzata restituisce un insieme W con le marcature mostrate in Tabella 3.32:

Marcatura	Prob. analitica a meno di G
[3 0]	0.578703
[1 2]	0.833333
[0 3]	1.0
[2 1]	0.694444

Tabella 3.32: Marcature dell'insieme W

Riportiamo nelle Tabelle 3.33 e 3.34 i risultati offerti dal simulatore PFNetS corrispondenti alle stime delle probabilità stazionarie relative alle marcature dell'insieme W. Riportiamo inoltre le ampiezze dei campioni ottenuti al termine delle 50 prove ripetute.

Confidenza 95%						
		Metodo della matrice			Metodo del vettore	
Marcatura	Prob. Analitica	Prob. Stimata	Ampiezza relativa IC		Prob. Stimata	Ampiezza relativa IC
[3 0]	0.186289	0.186276	5.97896E-4		0.186224	8.950942E-4
[1 2]	0.268256	0.268236	6.210898E-4		0.268416	7.343892E-4
[0 3]	0.321908	0.321954	6.06407E-4		0.321835	6.502751E-4
[2 1]	0.223547	0.223548	6.906702E-4		0.223537	9.29812E-4

Tabella 3.33: Risultati di simulazione. Intervalli con grado di confidenza del 95%

Confidenza 99%						
		Metodo della matrice			Metodo del vettore	
Marcatura	Prob. Analitica	Prob. Stimata	Ampiezza relativa IC		Prob. Stimata	Ampiezza relativa IC
[3 0]	0.186289	0.186276	7.858074E-4		0.186224	9.89896E-4
[1 2]	0.268256	0.268236	8.162894E-4		0.268416	9.651972E-4
[0 3]	0.321908	0.321954	7.969922E-4		0.321835	8.546474E-4
[2 1]	0.223547	0.223548	9.077379E-4		0.223537	0.001222

Tabella 3.34: Risultati di simulazione. Intervalli con grado di confidenza del 99%

E' possibile notare dai dati in tabella che l'ordine di grandezza delle ampiezze relative viene mantenuto anche se si utilizzano due differenti criteri.

Riportiamo in Tabella 3.35 i dati del costo computazionale di entrambi i metodi relativo all'esecuzione di un singola run di simulazione.

	Metodo della matrice	Metodo del vettore
Ritrovamento insieme W	$O(4 \cdot \log 4)$	$O(4 \cdot \log 4)$
Costruzione	$O\left(4^2\right)$	$O\left(4\right)$
Verifica criterio	$O\left(4^2\right)$	$2 \cdot O\left(4\right)$
Costo analisi totale	$NumStep \cdot 2 \cdot O\left(4^2\right)$	$NumStep \cdot 3 \cdot O\left(4\right)$

Tabella 3.35: Confronto costo computazionale

Per quel che concerne il costo dell'analisi totale del singolo run di simulazione non è possibile specificare un valore numerico esatto per $NumStep$ in quanto tale valore varia in modo indipendente per ogni singola run.

I tempi impiegati per effettuare i 50 run di simulazione sono i seguenti:

- 1 minuto e 7 secondi, utilizzando il metodo della matrice.
- 50 secondi, utilizzando il metodo del vettore.

I risultati ottenuti confermano quanto descritto nel paragrafo 3.4.

Capitolo 4

Confronto con TimeNET

4.1 Introduzione di TimeNET

TimeNET è uno strumento commerciale non open source, distribuito gratuitamente per uso accademico in grado di modellare ed analizzare reti di Petri di varie forme e tipologia.

Lo sviluppo di TimeNET ebbe inizio nel 1991, sulla base della precedente implementazione denominata DSPNexpress. Entrambi i tool sono stati sviluppati inizialmente dalla Tecnische Universitat di Berlino per poi essere gestiti dalle università di Dortmund e Leipzig. Rispetto all'implementazione precedente, il nuovo tool prevede la possibilità di modellare e di analizzare e risolvere reti di Petri stocastiche e deterministiche estese (eDSPN). L'interfaccia grafica utilizzata era originariamente basata sul widget toolkit XAthena.

La versione 3.0 di TimeNET si distingue dalla precedente versione in quanto utilizza una nuova interfaccia motif-based in grado di permettere l'utilizzo di ambienti di modellazione differenti così da includere l'analisi di reti di Petri colorate variable-free, reti di Petri stocastiche fluide e reti di Petri stocastiche a tempo discreto [13].

Dal 2007 la versione 4.0 del tool include la possibilità di modellare e simulare reti di Petri stocastiche colorate, rendendo inoltre il suo utilizzo disponibile per ambienti Linux e Windows.

Le descrizioni della tipologia dei modelli creati da TimeNET4.0, sono memorizzate in formato XML. Il tool ad oggi supporta la modellazione e la valutazione delle seguenti classi di reti:

- eDSPN: Reti di Petri stocastiche e deterministiche estese. Le reti di Petri stocastiche generalizzate sono un caso speciale di tale classe.
- SCPNs: Reti di Petri stocastiche colorate.

Nel dettaglio, la classe eDSPN è caratterizzata da token indistinguibili e dalle seguenti tipologie di transizioni della rete:

- Immediate: il tempo di firing associato a tali transizioni è pari a zero. Le transizioni sono descritte dalle seguenti proprietà: *nome*, *priorità*, *peso* e *funzione di attivazione*. I valori associati agli attributi *priorità* e *peso* permettono di identificare la transizione candidata al firing nel caso in cui vi sia una concorrenza nell'accesso alle risorse. La *funzione di attivazione* identifica un'espressione dipendente dalla marcatura e determina se la transizione è abilitata o meno ad eseguire il firing.

- Con tempi di firing distribuiti esponenzialmente. Le transizioni caratterizzate da tali tempi sono definite dall'attributo *delay* che identifica il tempo di firing associato alle stesse.

- Con tempi deterministici. Il tempo di firing ad esse associato è esprimibile da un valore costante.

- Con tempi appartenenti ad una particolare classe di distribuzioni definita *expolinomial*. Tale distribuzione di probabilità prevede la presenza di componenti discrete e continue. A questa classe appartengono distribuzioni note quali: distribuzione uniforme, triangolare e discreta finita.

Nel caso in cui in un modello tutte le transizioni con tempi di firing non-esponenzialmente distribuite sono mutuamente esclusive è possibile effettuare un'analisi stazionaria numerica del comportamento del modello.

Inoltre, se le transizioni sono caratterizzate da tempi di firing deterministici è possibile affettuare un analisi transiente.

Il modello è definito GSPN se prevede solamente transizioni immediate o transizioni con tempi di firing esponenzialmente distribuiti. In tal caso, è possibile applicare algoritmi standard per la valutazione steady-state e transiente a partire dalla catena di Markov sottostante alla rete oggetto di studio.

Consideriamo la versione di TimeNET 4.0 che fornisce la valutazione di alcune proprietà strutturali della eDSPN analizzata. La proprietà più rilevante è la possibile ricerca degli invarianti e degli insiemi di conflitto relativi alla rete stessa. Inoltre, a partire da una qualsiasi rete di Petri stocastica è possibile generare il relativo grafo di raggiungibilità per l'analisi transiente e steady-state. Per ottenere una generazione efficente del grafo di raggiungibilità si sfruttano particolari proprietà strutturali.

TimeNET 4.0 dispone di uno strumento grafico che permette, durante un'analisi transiente, di mostrare graficamente l'evoluzione delle misure di prestazione. Il tool inoltre, prevede algoritmi di simulazione steady-state e transiente. I risultati offerti da tali algoritmi possono essere ottenuti più velocemente mediante replicazioni parallele (architetture master/slave).

TimeNET 4.0 propone una variante dell'algoritmo per la simulazione stady-state utile per la valutazione di modelli con eventi rari (stati con

probabilità minori di 10^{-6}) basato sul metodo e l'algoritmo di RESTART. Tale funzionalità risulta fondamentale per l'analisi di sistemi safety-critical dove la valutazione delle occorrenze di eventi rari necessitano di simulazioni inaccettabili in termini di tempo.

4.2 Interfaccia

L'interfaccia grafica di TimeNET4.0, a differenza delle sue versioni precedenti, è stata completamente riscritta in Java permettendo in tal modo una possibile integrazione di qualsiasi formalismo grafico di modellazione.

Le classi del modello di rete sono descritte da un file XML che definisce gli elementi appartenenti al modello stesso, infatti, mediante la creazione di un file Schema XML è possibile definire oggetti di tipo posto, archi, transizioni ed altri elementi necessari per creare la struttura della rete. Inoltre, gli oggetti di tipo *posto* possono essere definiti gerarchicamente dai corrispondenti sottomodelli, che possono appartenere ad una classe di reti differente.

Per ciascun posto, arco e transizione del modello è possibile specificare particolari attributi ed informazioni. La forma di ognuno di questi elementi, viene definita mediante un insieme di primitive quali ad esempio 'polyline', 'ellipse' e 'text'.

Infine la creazione dei token dipende dai valori assegnati agli attributi dell'oggetto posto, rendendo possibile, la loro visualizzazione mediante puntini neri interni al posto stesso. Il modello creato viene memorizzato in un file XML. Inoltre vengono messe a disposizione operazioni di modifica e di memorizzazione dei modelli creati.

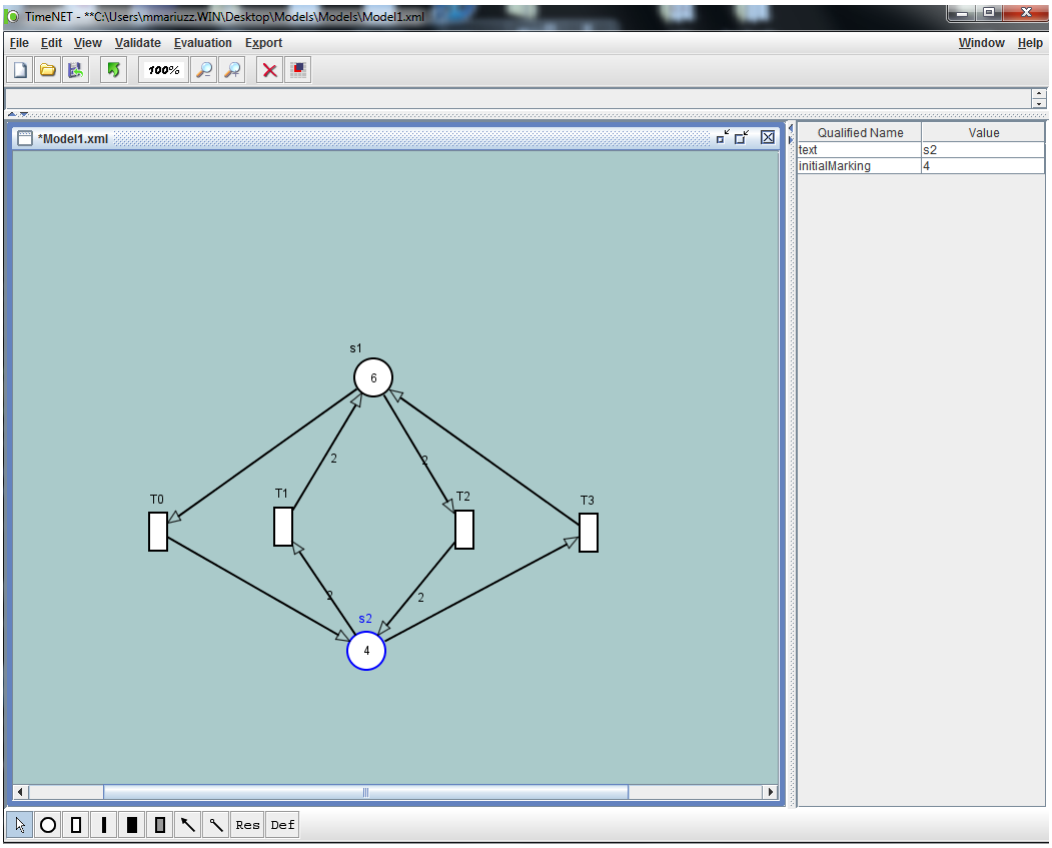

Figura 4.1: Interfaccia TimeNET

La Figura 4.1 mostra una schermata relativa alla GUI. L'interfaccia grafica è suddivisa in cinque parti:

1. Barra del menù: Posizionata nella parte alta della finestra, permette all'utente di effettuare le principali operazioni offerte dal tool. In particolare:

   (a) **File**. Raggruppa i comandi per la gestione dei file (creazione, apertura e salvataggio). Inoltre, permette il salvataggio del modello in formato SVG (Scalable Vector Graphics, il quale permette di rappresentare delle immagini in formato vettoriale, quindi con la possibilità di gestire il ridimensionamento senza perdere nulla in termini qualitativi) per la stampa. Infine con l'opzione *Setting* è possibile configurare lo strumento.

   (b) **Edit**. Raccoglie i comandi principali di editing: taglia, copia, incolla.

(c) **View**. Racchiude i comandi per la gestione delle varie viste, includendo la possibilità di effetture operazioni di zoom.

(d) **Validate**. Voce relativa esclusivamente al modello eDSPN. Raggruppa i comandi relativi all'analisi statica della rete di Petri oggetto di studio ed include l'opzione tokengame.

(e) **Evaluation**. Voce relativa esclusivamente al modello eDSPN. Raccoglie comandi per effetture l'analisi dinamica della rete.

(f) **Export**. Voce relativa esclusivamente al modello eDSPN. Permette tramite opportuni comandi di esportare la rete nei formati PNML e TN (vecchi formati proprietari di TimeNET).

(g) **Windows**. Insieme di comandi per la gestione della visualizzazione di più finestre nell'area di disegno.

(h) **Help**. Riporta alla consultazione del manuale relativo al tool.

2. Riquadro delle proprietà: Posizionato sulla destra dell'area di disegno, cambia aspetto sulla base dell'oggetto selezionato dall'utente, fornendo in tal modo un accesso diretto agli attributi dello stesso.

3. Area di disegno: Occupa la maggior parte della finestra. Il tool permette di aprire molteplici finestre di lavoro contenenti anche reti appartenenti a differenti modelli.

4. Prima toolbar: Posizionata sotto la barra del menù principale. Racchiude alcuni comandi presenti all'interno della barra del menù. In particolare permette la creazione di un nuovo foglio di lavoro, l'apertura di una rete già esistente ed il salvataggio del lavoro eseguito dall'utente nell'area di disegno. Permette una gestione dello zoom della finestra e l'eliminazione di oggetti selezionati nell'area di progettazione. Segue in Figura 4.2 la rappresentazione grafica della toolbar.

Figura 4.2: Toolbar superiore

5. Seconda Toolbar: Posizionata sotto l'area di disegno, permette tramite icone intuitive di costruire la rete di Petri oggetto di studio. Nel dettaglio, sono presenti tutte le componenti disponibili del modello. Il contenuto della barra, è automaticamente derivato dalla descrizione della classe del modello d'interesse (eDSPN o SCPNs). Cliccando su uno di questi, è possibile aggiungere il corrispondente elemento nell'area di disegno. In particolare la toolbar comprende le seguenti icone:

(a) **Select**: Permette di selezionare e spostare gli oggetti all'interno dell'area di progettazione.

(b) **Place**: Consente di aggiungere un posto alla rete.

(c) **ExponentialTransition**: Permette di creare una transizione con tempo di firing esponenzialmente distribuito.

(d) **ImmediateTransition**: Consente di aggiungere alla rete una transizione immediata.

(e) **GeneralTransition**: Permette di creare una transizione specificando un tempo di firing esprimibile da una funzione probabilitstica.

(f) **Arc**: Permette di inserire un generico arco tra un posto e una transizione o viceversa.

(g) **Inhibitor**: Permette di inserire tra le componenti posti e transizioni della rete un arco inibitore.

(h) **Measure**: Consente l'inserimento di funzioni di valutazione che descrivono gli indici di prestazione che si vogliono analizzare durante la simulazione.

(i) **Definition**: Consente l'inserimento di costanti che possono essere richiamate nelle definizioni di attributi e measure.

Figura 4.3: Toolbar inferiore

4.2.1 Validate

Per la classe di reti eDSPN, sono disponibili, nel menù principale, alcune funzioni ausiliarie. La prima voce del menù **Validate** contiene le funzioni relative alla valutazione della struttura del modello eDSPN. In particolare, la funzione *Estimate Statespace* produce una stima del numero di stati raggiungibili a partire dalla struttura del modello.

Le opzioni *Trap* e *Siphons* calcolano l'insieme delle trappole e dei sifoni minimali della rete. In paricolare, un sifone rappresenta un insieme di posti che tende a perdere token durante l'evoluzione di una rete, e che una volta persi tutti i token non è più in grado di riacquisirne. Una trappola identifica un'insieme di posti che complessivamente tende ad acquistare token durante l'evoluzione di una rete, e che una volta preso almeno un token non è più in grado di liberarsene.

La funzione *check structure* recupera l'insieme di P-invarianti e l'extended conflit set, ossia l'insieme delle transizioni immediate in conflitto.

Infine, la funzione *token game* avvia una simulazione della rete con la possibilità di scegliere l'ordine delle transizioni da far scattare.

4.2.2 Evaluation

Le funzioni messe a disposizioni da TimeNET per la valutazione delle prestazioni sono accessibili tramite il menù indicato dalla voce **Evaluation**.

Le differenti possibilità di valutazione del modello prevedono metodi per l'analisi, l'approssimazione e la simulazione steady-state e transiente della rete oggetto di studio.

Con il termine "analisi" si indica una valutazione analitica diretta e precisa delle prestazioni a partire da un'esplorazione completa del grafo di raggiungibilità relativo alla rete. Gli algoritmi di "approssimazione" utilizzano tecniche numeriche dirette ma evitano parti costose di valutazione producendo un livello minimo di imprecisione relativo ai risultati forniti. Infine, gli algoritmi di "simulazione" non prevedono la costruzione del grafo di raggiungibilità e forniscono stime attraverso tecniche di simulazione.

Una valutazione del transiente analizza il comportamento del modello a partire dalla marcatura iniziale, stato della rete a tempo zero, fino ad un tempo fissato dall'utente. La valutazione dello stato stazionario invece stima la prestazione media del sistema al termine della fase transiente, nel momento in cui viene raggiunto uno stato di equilibrio della rete.

Per permettere agli algoritmi di valutazione di stimare alcuni indici di prestazione è necessario, definire quest'ultimi mediante l'utilizzo di una particolare grammatica. La grammatica permette di generare le seguenti espressioni:

- $P\,\{\langle condizione_logica\rangle\}$ corrisponde alla probabilità relativa al verificarsi della $\langle condizione_logica\rangle$.
- $P\,\{\langle condizione_logica1\rangle\, IF\, \langle condizione_logica2\rangle\}$ calcola la probabilità condizionata che la $\langle condizione_logica1\rangle$ si verifichi sapendo che $\langle condizione_logica2\rangle$ si è verificata.
- $E\,\{\langle marc_func\rangle\}$ si riferisce al valore atteso di un espressione marking-dependent $\langle marc_func\rangle$. Una funzione marking-dependent è nella forma $\#Pn$ ed indica il numero di token presenti nel posto Pn.
- $E\,\{\langle marc_func\rangle\, IF\, \langle condizione_logica\rangle\}$ si riferisce al valore atteso dell'espressione $\langle marc_func\rangle$ marking-dependent solo per marcature che rispettano la condizione $\langle condizione_logica\rangle$.

In Figura 4.4 si riportano le opzioni del menù Evaluation.

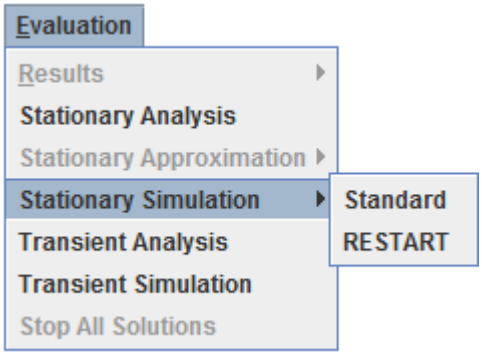

Figura 4.4: Menù Evaluation

4.2.2.1 Stationary Analysis

Calcola la soluzione stazionaria di stato della rete con tempo continuo. La Figura 4.5 illustra la finestra contenente i parametri da impostare per effetture l'analisi stazionaria del modello.

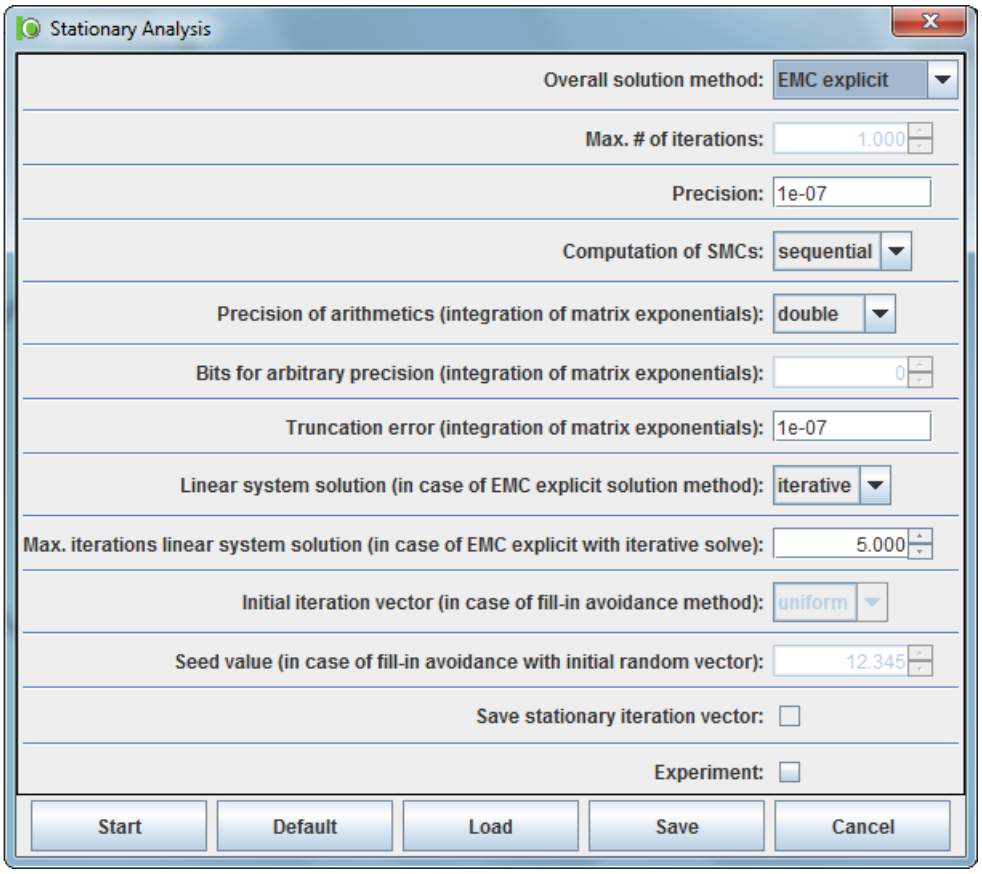

Figura 4.5: Stationary Analysis

- **Overall solution method**: Descrive la modalità con cui viene gestita la EMC (Embedded Markovian Chain) associata alla rete. Mediante l'opzione *EMC explicit* la catena di Markov viene calcolata e memorizzata utilizzando parti indipendenti (SMC) della stessa.
  E' possibile evitare il fenomeno di fill-in durante il calcolo della matrice EMC mediante l'opzione *fill-in avoidance*. In tal caso, è possibile definire mediante la scelta di un valore intero il numero di iterazioni dell'algoritmo di analisi.

- La scelta delle modalità di calcolo delle SMC (per l'opzione *EMC explicit*) può essere effettuata tramite l'opzione **Computation of SMCs**: le possibili modalità sono sequenziale e parallela.

- Per il calcolo della matrice Q (generatore infinitesimale) è possibile definire tramite l'opzione **Precision of arithmetics** il tipo di precisione da utilizzare nel calcolo (double o arbitrary). Nel caso di pre-

cisione arbitraria, è necessario definire il numero di bit per la memorizzazione dei valori con l'opzione **Bits for arbitrary precision** e l'errore di troncamento mediante l'opzione **Truncation Error.**

- Il sistema di equazioni lineari (in caso di calcolo esplicito della EMC) può essere risolto in modo diretto o in modo iterativo. Per quel che concerne il metodo *fill-in avoidance*, il vettore iniziale di iterazione può essere distribuito in modo uniforme, casuale o caricato da un file.

- L'opzione *Experiment* permette di eseguire in modo automatico analisi multiple a partire da una determinata gamma di parametri. L'avvio di analisi mediante l'opzione experiment apre una finestra di dialogo utile per la definizione dei parametri e di ulteriori possibili opzioni.

4.2.2.2 Stationary Simulation Standard

Tale opzione permette di simulare il comportamento steady-state di una rete di Petri Stocastica arbitraria e di ottenere le stime relative a particolari misure di prestazione definite.

Per eseguire la simulazione, è necessario definire almeno una misura di prestazione. Poichè le osservazioni della fase transiente non rappresentano il comportamento steady-state del modello, la durata di tale fase viene rilevata automaticamente in fase di simulazione e vengono scartate le osservazioni relative ad essa. La Figura 4.6 illustra la finestra con i parametri da impostare per effetture la simulazione stazionaria del modello.

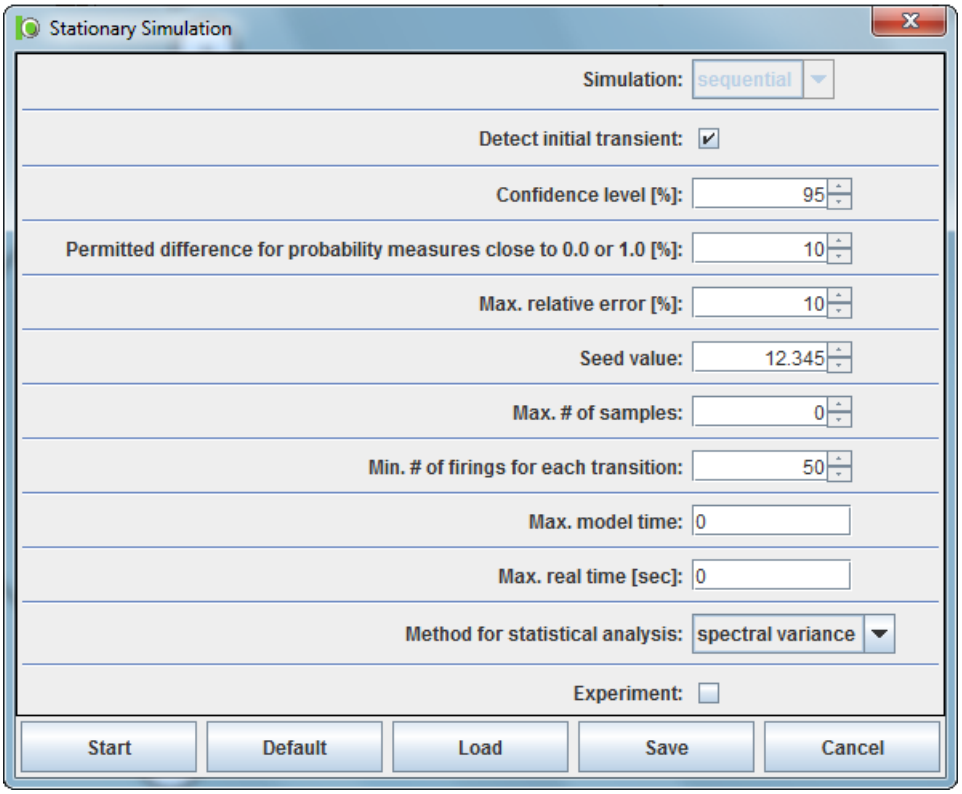

Figura 4.6: Stationary Simulation

E' possibile scegliere se attivare / disattivare il rilevamento della fase transiente selezionando o meno l'opzione *detect initial transient*. Solitamente, un run di simulazione di TimeNET 4.0 si arresta al raggiungimento di una precisione dei risultati definita dall'utente. Il raggiungimento della precisione richiesta avviene in modo statistico e mediante l'utilizzo degli intervalli di confidenza. In paricolare, il livello di confidenza *(Confidence level[%])* definisce la probabilità (in percentuale) che il valore reale della misura di prestazione cade nell'intervallo di confidenza calcolato durante la simulazione e relativo alla misura stessa.

Il valore di semi-ampiezza relativa dell'intervallo di confidenza *(Max. relative error[%])* definisce la massima semi-ampiezza tollerata per l'intervallo di confidenza calcolato a partire da un campione di osservazioni. Ciascuna delle osservazioni del campione identifica la stima dell'indice di prestazione oggetto di studio restituita al termine di un singolo run. TimeNET prevede l'esecuzione di sei differenti run di simulazione.

Per le misure di probabilità si utilizzata una stima di varianza dato che le osservazioni relative alle probabilità non possono essere assunte come nor-

malmente distribuite. La precisione delle stime vicine al valore 0.0 o 1.0 può essere migliorata specificando la più piccola differenza permessa per tali misure *(permitted difference for probability measures close to 0.0 or 1.0 [%])*.

E' possibile definire il seme iniziale *Seed value* del generatore di numeri pseudocasuali per eseguire le differenti prove di simulazione.

La lunghezza di un run di simulazione dipende dal modello definito, dalle misure di prestazione e dalla precisione richiesta dall'utente. Per limitare la lunghezza del run è possibile utilizzare le seguenti quattro opzioni:

La voce *Max. # of samples* specifica il numero di osservazioni relative ad una particolare misura di prestazione da generare in fase di simulazione. Al raggiungimento di tale numero la simulazione si arresta. Il valore di default 0 indica l'assenza di un limite preciso per la durata complessiva della simulazione.

In fase di simulazione, inoltre, è possibile definire un limite relativo al numero minimo di operazioni di firing di ogni singola transizione del modello mediante l'opzione *Min. # of firing for each transition.* Tale opzione risulta essere utile in quanto garantisce la copertura di ogni attività del modello nei casi in cui le frequenze di firing delle transizioni differiscono notevolmente tra loro.

L'opzione *Max. model time* specifica un limite superiore relativo al tempo simulato.

La voce *Max real time[sec]* indica il massimo valore di tempo reale richiesto per eseguire la simulazione. Il valore è espresso in secondi.

Nel momento in cui la simulazione termina sulla base di uno dei vincoli imposti dalle opzioni sopra indicate, viene visualizzato in output un valore che rappresenta la precisione raggiunta dalla simulazione stessa.

La simulazione standard definisce due tipologie di stima della varianza, necessaria per il rilevamento della precisione raggiunta dalla simulazione tramite due tecniche (*spectral variance* [10] o *control variates* [11]) presenti nell'opzione *Method for statistical analysis.*

L'opzione *Experiment* permette di eseguire più simulazioni a partire da una gamma di parametri. L'avvio di analisi mediante l'opzione experiment apre una finestra di dialogo utile per la definizione dei parametri e di ulteriori possibili opzioni.

4.2.2.3 Stationary Simulation RESTART

Rappresenta una variante dell'algoritmo di simulazione steady-state particolarmente utile per la valutazione di modelli analizzati per lo studio di eventi rari, eventi con probabilità inferiori al valore 10^{-6} [17]. La stima di eventi di questo tipo rappresenta un problema noto in quanto richiede run di simulazione estremamente lunghi. Per utilizzare il metodo RESTART è sufficiente

definire una singola misura che identifica l'evento raro. Tale misura deve essere nella forma $P\{\#Pi \geq n\}$ oppure $P\{\#Pi \leq n\}$ al fine di misurare la probabilità di avere almeno n (oppure non più di n, rispettivamente) token nel posto Pi.

La Figura 4.7 illustra la finesta contenente i parametri da impostare per effetture la simulazione stazionaria RESTART del modello.

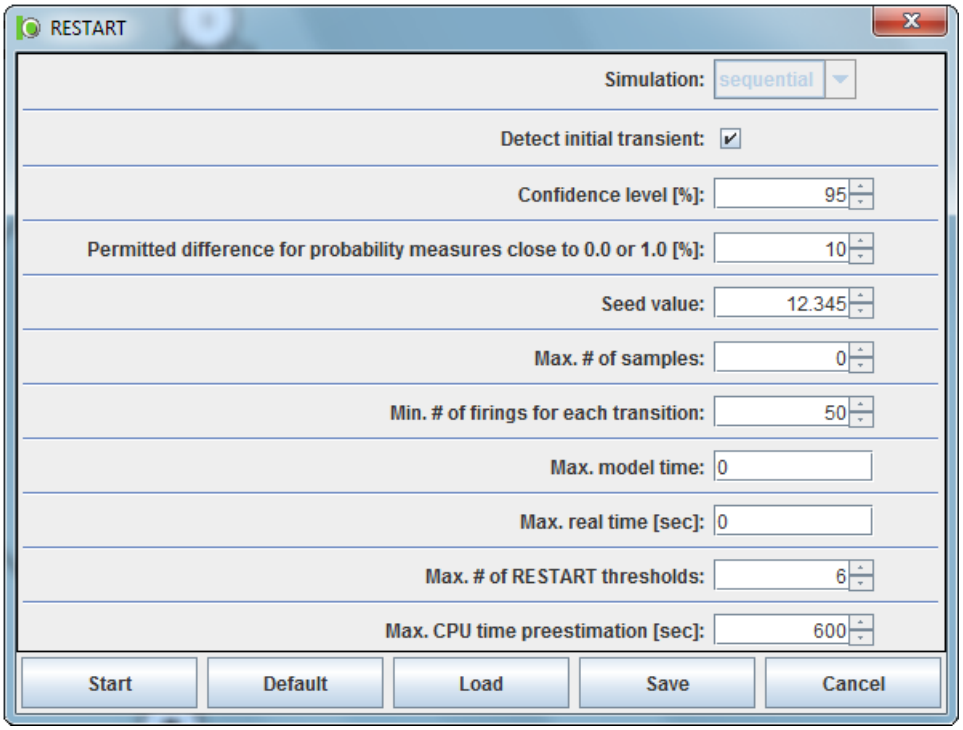

Figura 4.7: Stationary Simulation RESTART

L'opzione *Max. # of RESTART thresholds* rappresenta un importante parametro della tecnica RESTART che può essere imposto sulla base dell'ordine di grandezza di probabilità prevista relativo alla stima di probabilità dell'evento raro. Una regola pratica consiste nell'utilizzare come parametro un valore positivo relativo all'esponente della probabilità attesa. Per esempio si sceglie una soglia pari a 8 nel caso in cui la probabilità dell'evento raro sia nell'ordine di 10^{-8}.

4.2.2.4 Transient Analysis

Calcola la soluzione transiente della rete, ovvero il comportamento della stessa dato uno stato iniziale e per un tempo fissato dall'utente. Per eseguire

l'analisi transiente è necessario definire almeno un indice di prestazione. La Figura 4.8 illustra la finestra contenente i parametri da impostare per effetture un'analisi transiente del modello.

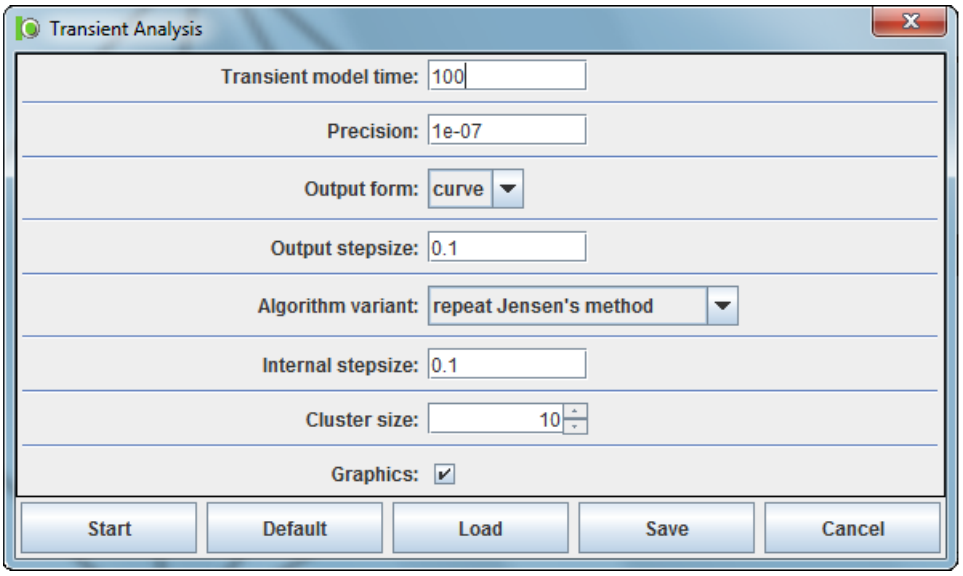

Figura 4.8: Transient Analysis

Il primo ed il più importante parametro di scelta è rappresentato dal valore relativo al tempo associato alla durata della fase transiente. Tale valore temporale è espresso in numero di operazioni di firing delle transizioni che compongono la rete ed è identificato da *Transient model time*.

L'opzione *Precision* indica il valore di precisione numerica desiderata dall'utente e relativa ai risultati forniti in output.

L'opzione *Output form* rappresenta la modalità grafica da utilizzare per visualizzare l'output del comportamento transiente. La voce *Output stepsize* definisce il numero dei passi dopo i quali vengono calcolati e visualizzati i risultati relativi all'analisi. Il risultato può essere ottenuto mediante l'utilizzo di due differenti algoritmi (*repeating Jensen's method* [12] e *computing the matrix exponentials*).

4.2.2.5 Transient Simulation

Stima il comportamento transiente della rete dato uno stato iniziale e fino ad un determinato tempo. La simulazione può essere effettuata a partire da una qualsiasi tipologia di rete ma prende in considerazione solo le misure definite dall'utente.

La Figura 4.9 illustra la finestra contenente i parametri da impostare per effetture una simulazione transiente del modello.

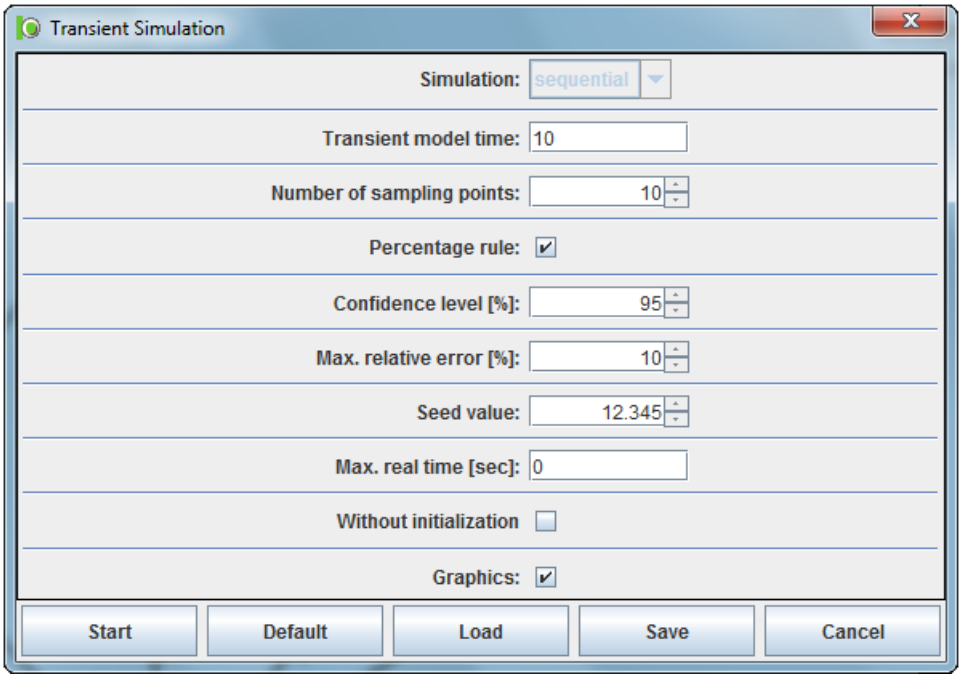

Figura 4.9: Transient Simulation

Mediante l'opzione *Number of sampling points* è possibile ottenere un miglioramento della risoluzione delle curve generate in fase di simulazione transiente. Tali curve identificano l'andamento delle stime delle misure di prestazione indicate dall'utente.

Nel caso in cui l'opzione *Percentage rule* sia selezionata, solamente una percentuale di punti di campionamento deve raggiungere la precisione predefinita in fase di simulazione transiente. In caso contrario questo deve valere per tutti i punti di campionamento.

Le rimanenti opzioni sono caratterizzate da significati equivalenti alle voci indicate per la simulazione steady-state (Stationary Simulation Standard / RESTART).

4.3 Valutazione dei risultati

Successivamente ad una valutazione del modello terminata con esito positivo, le definizioni delle misure di prestazione relative al modello stesso sono

aggiornate in modo automatico ed i rispettivi risultati di valutazione sono proposti in output nell'area di disegno della GUI.

La valutazione delle misure di prestazione genera alcuni file di testo contenenti sia risultati intermedi (per esempio relativamente all'extended conflict set) sia l'output dettagliato relativo ai risultati finali.

I file generati sono memorizzati in una directory denotata con `<nome del modello>.dir` contenuta nella stessa cartella in cui è stato caricato il modello oggetto di studio. Alcuni di questi file di testo possono essere utilizzati per rappresentare graficamente i risultati ottenuti mediante un programma per la realizzazione di grafici di funzioni matematiche.

La valutazione delle prestazioni del modello crea solamente un sottoinsieme dei possibili file di testo generabili ed in particolare i file:

- `<nomedelmodello>.RESULTS`
- `<nomedelmodello>.STAT_OUT`
- `<nomedelmodello>.curves`.

Segue la lista completa dei file di testo generati sulla base della loro estensione:

- AUX: Contiene alcune informazioni ausiliare relative alle valutazioni effettuate. Nello specifico il file contiene il nome del modello, il metodo risolutivo utilizzato ed il tipo di valutazione effettuata.
- curves: Contiene i dati per la visualizzazione dei risultati relativi alla misure di prestazione definite nel caso di un esperimento (insieme di piu analisi). Tale file può essere utilizzato da un tool esterno per la rappresentazione grafica dei risultati ottenuti.
- DEFINFO: Contiene informazioni relative alle proprietà strutturali del modello quali ad esempio le funzioni di enabling delle transizioni ed il peso delle transizioni dipendenti dalle marcatura.
- ECS: Descrive gli insiemi extended conflict del modello.
- ese: Contiene una stima della cardinalità relativa allo spazio degli stati.
- EXPRESULT: Contiene i risultati ottenuti, in forma tabellare, successivamente all'esecuzione di un esperimento. Nel dettaglio, il file contiene il risultato finale relativo ad ogni indice di prestazione di ogni singola fase dell'esperimento.
- INV: Contiente i P-invarianti del modello.
- pid: Contiene la lista dei processi software utilizzati per la valutazione del modello.

- RESULTS: Rappresenta la lista dettagliata dei risultati relativi alle misure di prestazione definite dall'utente. Inoltre contiene di default i valori di throughput delle transizioni temporizzate che compongono la rete.
- RRG: Contiene il grafo di raggiungibilità ridotto (RRG) del modello espresso in forma binaria.
- siphons: Include i possibili sifoni del modello.
- STAT_OUT: Contiene i risultati relativi alle differenti analisi statistiche in caso di simulazione stazionaria oppure include i risultati ottenuti da una simulazione transiente.
- STRUCT: Contiene informazioni relative alla struttura della rete.
- tmark: Include la numerazione delle marcature tangible del modello.
- TN: Contiene il modello oggetto di studio in formato .TN. Tale formato è stato utilizzato da TimeNET 2.0 e viene tuttora preso come input da alcuni metodi di valutazione.
- traps: Contiene le possibili trappole del modello.

4.4 Confronto

In questo paragrafo si presenta un confronto tra il simulatore PFNetS (descritto nella tesi del collega Massimo Mariuzzo) e lo strumento di analisi TimeNET 4.0 analizzato nel paragrafo precedente. In particolare, presentiamo il confronto a partire da differenti aspetti quali le tipologie di reti analizzabili, le tecniche utilizzate per limitare la lunghezza del run di simulazione, le funzionalità offerte ed infine le modalità utilizzate per la stima degli indici di prestazione.

Il confronto termina con la presentazione dei risultati di simulazione restituiti dai differenti simulatori a partire dagli stessi modelli di rete analizzati. A tal scopo impostiamo i parametri di entrambi i simulatori al fine di ottenere simulazioni tra loro confrontabili.

4.4.1 Classi di reti supportate

Come descritto nel paragrafo 4.1 il simulatore TimeNET 4.0 è in grado di analizzare reti di Petri stocastiche, deterministiche estese (eDSPN) e reti di Petri stocastiche colorate (SCPNs). A differenza di quest'ultimo, il simulatore da noi proposto prevede la sola analisi di reti di Petri stocastiche esprimibili in forma prodotto. Tuttavia, la rappresentazione utilizzata per la definizione delle reti prevede strutture dati che consentono di costruire GSPN

in grado quindi di supportare reti con archi inibitori e transizioni immediate. Al fine di effettuare un'analisi di tale classe di reti è necessario modificare il codice implementato relativamente ai metodi che identificano le regole di enabling e firing in modo da garantire eventuali abilitazioni di transizioni caratterizzate da archi inibitori in ingresso. Inoltre per quel che riguarda reti GSPN, il simulatore PFNetS non prevede la terminazione dei run sulla base dei criteri di arresto descritti nel Capitolo 3, in quanto quest'ultimi utilizzano grandezze dipendenti esclusivamente dall'applicazione del teorema di Coleman Henderson et al. Tali valori sono rappresentati dalle probabilità di stato stazionario non normalizzata ottenibili dalla forma prodotto. Tuttavia, per reti GSPN è possibile ottenere stime degli indici di prestazione utilizzando i metodi presentati nella tesi del collega Massimo Mariuzzo.

4.4.2 Modalità di arresto della simulazione

Entrambi i simulatori definiscono differenti criteri di terminazione che prevedono l'arresto dei run di simulazione sulla base di un valore di precisione specificato in modo arbitrario dall'utente. Tuttavia, tale misura assume differenti significati sulla base del simulatore utilizzato.

In particolare, TimeNET 4.0 utilizza una misura di precisione che corrisponde ad un limite massimo, definito dall'utente, di semi-ampiezza relativa dell'intervallo di confidenza costruito a partire da osservazioni di stime degli indici di prestazione. In particolare la simulazione si arresta dopo un numero di osservazioni tali da garantire la costruzione di un intervallo di confidenza con una semi-ampiezza relativa minore di quella definita dall'utente.

La misura di precisione utilizzata nel simulatore PFNetS, invece, è rappresentata da un valore massimo di tolleranza definito dall'utente ed applicabile a particolari grandezze costruite ad intervalli regolari di passi di simulazione sulla base delle probabilità di stato stazionarie analitiche e stimate. Maggiori dettagli nella definizione e nell'applicabilità del valore di tolleranza sono presentati nel Capitolo 3.

Tuttavia, sebbene in entrambi i casi il valore di precisione identifichi la modalità principale utilizzata per terminare le simulazioni, i due simulatori nel caso in cui non si raggiunga la precisione definita dall'utente, garantiscono comunque un arresto di run sulla base della sua lunghezza massima tollerabile. In particolare, TimeNET identifica un limite massimo relativo al numero di firing delle transizioni, un numero minimo di firing di ogni singola transizione ed infine una durata massima di tempo di simulazione espressa in secondi. Il simulatore PFNetS, invece, definisce un limite massimo relativo alla lunghezza di un run di simulazione basato sul numero di firing effettuati complessivamente dalle transizioni (pari a 3×10^8). Inoltre, a differenza di TimeNET, il simulatore PFNetS prevede un arresto immediato di simulazione nel caso in cui la rete oggetto di analisi raggiunga uno stato di

deadlock. Tale stato identifica il raggiungimento di una marcatura che non consente l'abilitazione di alcuna transizione della rete.

4.4.3 Funzionalità

Il simulatore TimeNET 4.0 offre un numero maggiore di funzionalità rispetto al simulatore per SPN proposto. In particolare TimeNET, a differenza del simulatore PFNetS, dispone di funzionalità che prevedono, a partire da un modello di rete, il calcolo dell'insieme delle trappole e dei sifoni minimali, dei P-invarianti, dell'extended conflit set della rete e la stima del numero di stati raggiungibili a partire da una marcatura iniziale. Inoltre, TimeNET prevede l'utilizzo di un'interfaccia grafica per la creazione e la modifica del modello analizzato.

A differenza di TimeNET, il simulatore PFNetS prevede l'implementazione di un algoritmo per il ritrovamento delle marcature raggiungibili con i maggiori valori di probabilità di stato stazionaria non normalizzata. Tale algoritmo risulta di fondamentale importanza per l'utilizzo dei due metodi proposti per stabilire un criterio di terminazione dei run di simulazione.

TimeNET permette di effettuare un'analisi ed una simulazione sia transiente che stazionaria della rete. In particolare, l'analisi stazionaria viene effettuata solamente nei casi in cui la rete oggetto di studio sia caratterizzata da un insieme di raggiungibilità finito. Contrariamente, PFNetS prevede di eseguire solamente l'analisi e la simulazione stazionaria del modello preso in esame. In particolare, per quel che riguarda l'analisi stazionaria, è in grado di restituire sia i valori analitici riferiti ai principali indici di prestazione sia le probabilità di stato analitiche stazionarie delle marcature oggetto di studio. Tale procedimento è proposto esclusivamente nel caso in cui la rete oggetto di studio sia aperta e riconducibile ad una rete di Jackson in forma prodotto.

Entrambi i simulatori offrono la possibilità di rilevare la fase transiente della simulazione e di scartare successivamente le osservazioni ottenute in tale fase. Nel dettaglio, il simulatore PFNetS stima la lunghezza della fase transiente mediante il test di Welch.

4.4.4 Stima degli indici di prestazione

In TimeNET, la definizione dei principali indici di prestazione si basa sull'utilizzo di una particolare grammatica descritta in appendice nel manuale del tool [13]. Nel dettaglio, la definizione e la successiva stima degli indici di prestazione sono legate, nella maggior parte dei casi alle singole probabilità di avere, in fase di simulazione, un determinato numero di token all'interno dei posti dalla rete.

A differenza di TimeNET, il simulatore PFNetS, permette, per determinati indici di prestazione, di ottenere una stima degli stessi mediante osser-

vazioni dirette del comportamento della rete. In tal modo si sfrutta la sola definizione degli indici senza far uso di alcuna formula analitica.

4.4.4.1 Throughtput

In TimeNET per stimare il throughtput di una generica transizione t della rete è necessario utilizzare la sintassi $P\{\wedge(\#P_i > n)\}*\lambda_t$ dove $P\{\wedge(\#P_i > 0)\}$ indica la probabilità che ogni posto P_i, in input alla transizione t, contenga un numero minimo di token (indicato con $\#P_i$) maggiore di n, sufficiente cosi ad abilitare la transizione t. Il valore n indica il peso dell'arco che collega il posto P_i alla transizione t e λ_t denota il rate della transizione t.

Al fine di ottenere una stima dell'indice di throughtput di una generica transizione t, il simulatore PFNetS calcola esclusivamente il rapporto tra il numero totale di operazioni di firing effettuate dalla transizione stessa ed il tempo complessivo di simulazione effettuando in tal modo un'osservazione diretta del comportamento della transizione durante la simulazione.

4.4.4.2 Numero medio di token

Per la stima del numero medio di token relativo ai posti delle rete, TimeNET mette a disposizione una particolare funzione che permette di stimare l'indice utilizzando la sintassi $E\{\#P_i\}$ dove P_i indentifica un generico posto della rete.

Il simulatore PFNetS stima tale indice di prestazione mediante un particolare rapporto descritto nella tesi del collega Massimo Mariuzzo (paragrafo 4.3).

Non conoscendo le procedure utilizzate dalla funzione $E\{\#P_i\}$ non si può effettuare un confronto tra le differenti modalità utilizzate dai due simulatori per la stima dell'indice di prestazione.

4.4.4.3 Utilizzazione

TimeNET, similmente al simulatore proposto, stima l'indice di utilizzazione mediante le singole probabilità di avere, in fase di simulazione, un determinato numero di token all'interno dei posti dalla rete. Nel dettaglio, TimeNET tramite l'utilizzo della sintassi $P\{\#P_i > 0\}$ restituisce una stima dell'utilizzazione del generico posto P_i. Il simulatore PFNetS sfrutta la medesima idea in quanto ricava una stima dell'indice di utilizzazione mediante la formula $1 - PE_i$ dove PE_i identifica la probabilità di avere 0 token nel posto P_i.

In entrambi i casi quindi si sfrutta la stima di probabilità di stato riferita al posto $i - esimo$.

4.4.4.4 Probabilità di stato

Data una generica marcatura di interesse $M = [M(0)\, M(1) \ldots M(n)]$ dove $M(i)$ identifica il numero di token nel posto $i-esimo$ ed $n = |P|$, TimeNET stima la probabilità stazionaria di stato relativa ad M utilizzando la sintassi $P\{\#P_0 = M(0)\ AND\, \#P_1 = M(1)\ AND \ldots AND\, \#P_n = M(n)\}$ oppure il prodotto $P\{\#P_0 = M(0)\} * P\{\#P_1 = M(1)\} * \ldots * P\{\#P_n = M(n)\}$.

Il simulatore PFNetS stima la probabilità stazionaria di stato relativa alla marcatura M mediante il rapporto tra il tempo in cui la rete si trova nello stato M ed il tempo totale di simulazione. In tal modo si evita di ottenere la stima di interesse come risultato di formule analitiche preferendo un'osservazione diretta del comportamento della rete, analizzando ad ogni passo di simulazione la marcatura raggiunta.

A differenza di TimeNET il simulatore PFNetS prevede la definizione di un indice di prestazione legato al tempo. Infatti, la grammatica fornita dal tool non consente di definire indici che restituiscano una metrica temporale.

4.4.5 Risultati di simulazione

Lo scopo di questo paragrafo consiste nel confrontare il simulatore PFNetS con TimeNET 4.0 sulla base della precisione dei risultati ottenuti al termine delle simulazioni e sul tempo impiegato dai differenti simulatori per ottenere le stima delle misure scelte come oggetto di analisi.

Per effettuare un corretto confronto è necessario eseguire le differenti simulazioni a partire dalle medesime condizioni. E' quindi necessario impostare in modo opportuno particolari parametri in entrambi i simulatori utilizzati. Dato che, in entrambi i casi, il tempo di simulazione varia sulla base del numero di indici di prestazione da stimare è necessario che i simulatori si concentrino esclusivamente sulla stima dello stesso numero di misure. Per tale motivazione, il confronto proposto analizza una particolare SPN esclusivamente sulla base dell'indice di prestazione riferito al numero medio dei token presenti nei posti e nell'intera rete. Si eliminano in tal modo le operazioni per la raccolta di informazioni utilizzate per la stima di ulteriori indici quali utilizzazione e throughput.

Si effettua il confronto sulla base del numero medio di token in quanto, a differenza degli altri indici di prestazione, quest'ultimo non è calcolato, in entrambi i simulatori, mediante l'utilizzo di formule analitiche che coinvolgono le probabilità di stato stazionarie.

L'operazione di confronto è articolata nelle seguenti fasi:

1. A partire dal modello di rete stocastica di Petri illustrata in Figura 3.6, il simulatore PFNetS esegue un numero arbitrario di run. Tale operazione permette di costruire gli intervalli di confidenza a partire

da campioni di osservazioni riferite al numero medio di token dei singoli posti e dell'intera rete. Il livello di confidenza scelto è del 95%. Ogni singolo elemento di ogni singolo campione identifica la stima, dell'indice preso in esame, ottenuta al termine di un run di simulazione. Per la costruzione degli intervalli di confidenza sopracitati il simulatore TimeNET prevede un numero di prove pari a 6. Nel simulatore proposto, tale valore risulta essere insufficiente per la costruzione degli intervalli di confidenza. Per tale motivo è necessario eseguire almeno 15 run di simulazione differenti.

2. Al termine dell'esecuzione dei run, si analizzano le ampiezze relative agli intervalli di confidenza ottenuti. In base al valore di quest'ultime si settano i parametri di TimeNET 4.0 ed in particolare i valori riferiti alle voci *Max. relative error[%]* e *permitted difference for probability measures close to 0.0 or 1.0 [%]* analizzate nel paragrafo 4.2.2.2. Tale operazione garantisce che i due differenti simulatori forniscano stime degli indici di prestazione analizzati con intervalli di confidenza dello stesso ordine di grandezza.

3. Infine, effetuiamo la simulazione stazionaria di TimeNET e procediamo al confronto tra i due simulatori sulla base del tempo impiegato per le rispettive simulazioni e relativamente alla precisione delle stime ottenute.

4.4.5.1 Specifiche hardware

La macchina su cui si effettuano le differenti simulazioni ha le seguenti specifiche tecniche:

- Processore AMD Athlon(tm) 64 X2 Dual Core Processor 5600+ 2.90 GHz
- Memoria RAM 4.00GB
- Sistema operativo Windows7 64-bit

4.4.5.2 Esperimento

Lo scopo di questa prova consiste nell'effettuare un confronto tra i due simulatori a partire dalla rete in Figura 3.6 i cui posti sono caratterizzati da bassi valori di intensità di traffico. A tal scopo si associano i seguenti tassi di firing alle transizioni: $\chi_1 = 1.5$, $\chi_2 = 3.0$, $\chi_3 = 11.0$, $\chi_4 = 19.0$, $\chi_5 = 9.0$, $\chi_6 = 2.0$, $\chi_7 = 10.0$, $\chi_8 = 3.0$. Si ottengono i seguenti valori di intensità di traffico:

$$\begin{cases} \rho_1 = \frac{\chi_1\chi_3}{\chi_4\chi_2} = 0.289473 \\ \rho_2 = \frac{\chi_2}{\chi_3} = 0.272727 \\ \rho_3 = \frac{\chi_5\chi_1}{\chi_4\chi_6} = 0.355263 \\ \rho_4 = \frac{\chi_1\chi_5}{\chi_4\chi_8} = 0.23668 \\ \rho_5 = \frac{\chi_1}{\chi_7} = 0.15 \end{cases}$$

Tali valori garantiscono la condizione di stazionarietà $\rho_i < 1$ relativamente ad ogni $i - esimo$ posto della rete.

Il criterio di arresto di simulazione scelto si basa sul metodo della matrice e prevede una cardinalità dell'insieme W pari a 4 ed un valore di tolleranza pari a 0.008. La scelta di tale valore permette di ottenere ampiezze relative agli intervalli di confidenza caratterizzati da un ordine di grandezza massimo pari a $1 \cdot 10^{-2}$. Tale decisione è dettata dal fatto che in TimeNET, non è possibile impostare come criterio di arresto un valore di ampiezza dell'intervallo di confidenza, associato alle stime degli indici oggetti di studio, con un ordine di grandezza inferiore.

Scelta una marcatura iniziale $M_0 = [1\,1\,0\,1\,0]$, l'esecuzione delle 15 prove di simulazioni indipendenti restituisce i seguenti risultati:

Indice	Valore analitico	Valore stimato Simulatore	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
T_1	0.407407	0.407210	0.405086	0.409334	0.010431	SI
T_2	0.374999	0.375316	0.374593	0.376039	0.003852	SI
T_3	0.551020	0.548861	0.542886	0.554837	0.021773	SI
T_4	0.310344	0.309832	0.307930	0.311733	0.012273	SI
T_5	0.176470	0.175951	0.175281	0.176621	0.007618	SI
T_{TOT}	1.820243	1.817172	1.807923	1.826421	0.010179	SI

Tabella 4.1: Analisi dati offerti dal simulatore PFNetS

Come si può osservare dalla Tabella 4.1, il valore di ampiezza massima relativa, restituita da PFNetS, è pari a 0.021773. Tale valore è indicativo al fine di impostare in modo opportuno i parametri di simulazione di TimeNET. I risultati ottenuti dalla simulazione effettuata da TimeNET sono riassunti in Tabella 4.2.

Indice	Valore analitico	Valore stimato TimeNET4.0	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Esito
T_1	0.407407	0.408200	0.404633	0.411767	0.01747	SI
T_2	0.374999	0.376030	0.372469	0.379592	0.018941	SI
T_3	0.551020	0.551093	0.546367	0.555819	0.017152	SI
T_4	0.310344	0.305912	0.303561	0.308263	0.015368	NO
T_5	0.176470	0.177611	0.175861	0.179361	0.019707	SI
T_{TOT}	1.820243	1.815862	1.799935	1.831789	0.017542	SI

Tabella 4.2: Analisi dati offerti da TimeNET

Il tempo di simulazione impiegato da PFNetS per l'esecuzione delle 15 run è di 35 secondi mentre il tool TimeNET termina la simulazione in 15 secondi. Tuttavia, dato che l'esperimento è stato eseguito utilizzando una macchina con processore Dual Core è necessario raddoppiare il tempo di simulazione ottenuto da TimeNET al fine di effettuare un confronto dei due simulatori sotto le stesse condizioni. Si ottiene in tal modo un valore temporale di simulazione circa pari a 30 secondi e di conseguenza inferiore al tempo impiegato dal simulatore PFNetS.

A partire da un'analisi dei risultati presenti in Tabella 4.1 e 4.2 si nota come il simulatore proposto restituisca risultati relativi alla stima dell'indice di interesse più vicini ai corrispettivi valori analitici rispetto a quelli offerti da TimeNET.

Inoltre, sebbene le ampiezze relative agli intervalli di confidenza restituiti dai due simulatori, riferite all'indice T_4, siano circa uguali, la stima restituita da TimeNET non cade nel rispettivo intervallo di confidenza.

Come si può osservare dalla Tabella 4.2 i valori delle ampiezze degli intervalli di confidenza offerti da TimeNET sono tutti dello stesso ordine di grandezza e precisamente pari all'1%. Ciò accade in quanto, in fase di simulazione, TimeNET termina di stimare un indice oggetto di studio nel momento in cui l'intervallo di confidenza calcolato a partire da quest'ultimo nel corso delle prove di simulazione, raggiunge il valore di semi-ampiezza relativa espresso in percentuale e fornita dall'utente come possibile criterio di simulazione. Nel momento in cui ogni intervallo di confidenza relativo ad ogni indice di prestazione oggetto di studio raggiunge, nel corso della simulazione, l'ampiezza desiderata, TimeNET arresta l'intera simulazione. A causa di tale limitazione, un confronto tra i due simulatori sulla base di molteplici misure oggetto di studio non è attendibile in quanto nel simulatore PFNetS la terminazione non si basa sull'analisi delle ampiezze degli intervalli di confidenza relativi agli indici di prestazione. Infatti, il simulatore PFNetS fornisce stime caratterizzate da intervalli di confidenza con differenti ordini di grandezza.

Per tal motivo si procede effettuando un confronto tra i due simulatori a partire da un'analisi relativa esclusivamente ad una singola misura di prestazione. Scelto in modo arbitrario un posto della rete, aumentiamo drasticamente il valore relativo all'intensità di traffico di quest'ultimo portandolo quasi al limite della condizione di stazionarietà. Tale espediente ci permette di osservare la rete sotto condizioni critiche e quindi di effettuare il confronto tra i simulatori a partire dalle medesime condizioni.

In riferimento alla rete in Figura 3.6, si scelgono i seguenti valori di tassi di firing delle transizioni: $\chi_1 = 1.5$, $\chi_2 = 3.0$, $\chi_3 = 11.0$, $\chi_4 = 19.0$, $\chi_5 = 9.0$, $\chi_6 = 2.0$, $\chi_7 = 1.6$, $\chi_8 = 3.0$. A partire da quast'ultimi, ai posti della rete si associano le seguenti intensità di traffico:

$$\begin{cases} \rho_1 = \frac{\chi_1\chi_3}{\chi_4\chi_2} = 0.289473 \\ \rho_2 = \frac{\chi_2}{\chi_3} = 0.272727 \\ \rho_3 = \frac{\chi_5\chi_1}{\chi_4\chi_6} = 0.355263 \\ \rho_4 = \frac{\chi_1\chi_5}{\chi_4\chi_8} = 0.236842 \\ \rho_5 = \frac{\chi_1}{\chi_7} = 0.9375 \end{cases}$$

Si sceglie come criterio di arresto della simulazione il metodo della matrice e si impostano dei valori ξ di tolleranza in modo da ottenere come output di simulazione differenti valori corrispondenti alle ampiezze degli intervalli di confidenza costruiti a partire dalle osservazioni relative al numero medio di token del posto p_5 con valore analitico pari a 15. La cardinalità dell'insieme W è pari a 4.

I risultati ottenuti al termine dei differenti run di simulazione sono riportati in Tabella 4.3.

Valore stimato Simulatore	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Errore Assoluto	Errore Relativo %
15.127683	14.726972	15.528394	0.05297	0.127683	0.85
14.893279	14.328521	15.019036	0.046364	0.106721	0.71
14.930198	14.677542	15.182858	0.033844	0.069802	0.46
14.997131	14.807178	15.191084	0.025598	0.002869	0.02
15.002821	14.881288	15.124354	0.016201	0.002821	0.018

Tabella 4.3: Analisi dati offerti dal simulatore PFNetS

A partire da ogni valore di ampiezza relativa dell'intervallo di confidenza ottenuto dal simulatore PFNetS, si effettua una simulazione con TimeNET mediante l'opportuna configurazione dei parametri di simulazione. L'esecuzione delle differenti simulazioni restituisce i risultati riportati in Tabella 4.4.

Valore stimato TimeNET4.0	Estremo Inferiore	Estremo Superiore	Ampiezza relativa	Errore Assoluto	Errore Relativo %
15.188676	14.806513	15.570820	0.050320	0.188676	1.25
15.156485	14.778160	15.416775	0.042134	0.156485	1.04
15.097450	14.754395	15.265108	0.033827	0.097450	0.65
15.006833	14.863317	15.273354	0.027323	0.006833	0.04
15.002978	14.940702	15.118790	0.011870	0.002978	0.019

Tabella 4.4: Analisi dati offerti da TimeNET4.0

Come si può notare dalle Tabelle 4.3 e 4.4, a parità di ampiezza relativa, i risultati offerti da PFNetS e riferiti alla stima dell'indice di prestazione oggetto di studio sono caratterizzati da una maggiore precisione. Tale osservazione si evince nelle due tabelle analizzando i valori presenti nelle colonne denotate da *Errore Assoluto* ed *Errore Relativo %*. Si comprende, infatti, come il simulatore proposto fornisca a parità di condizioni di simulazione stime caratterizzate da errori (in relazione al corrispettivo valore analitico) inferiori rispetto a quelle fornite dalle simulazioni di TimeNET.

La Tabella 4.5 riporta i valori relativi ai tempi impiegati dalle varie simulazioni effettuate da entrambi i simulatori, ognuna delle quali contraddistinta da una differente ampiezza relativa dell'intervallo di confidenza associato alla stima dell'indice oggetto di studio. In particolare, i dati riferiti ai tempi di simulazione sono rappresentati come un'intervallo temporale. L'estremo inferiore (superiore) di tale range indica il minor (maggior) tempo impiegato dalle simulazioni per ottenere le ampiezza descritte. Ogni range temporale è costruito successivamente all'esecuzione di 20 differenti simulazioni.

Dimensione Ampiezza	Durata TimeNET	Durata Simulatore PFNetS
0.05	1 min 21 sec - 1 min 40 sec	2 min 10 sec - 3 min 25 sec
0.04	1 min 49 sec - 2 min 21	3 min 4 sec - 4 min 5 sec
0.03	2 min 50 sec - 3 min 20 sec	4 min 47 sec - 6 min 23 sec
0.02	4 min 23 sec - 5 minuti 51 sec	8 min 25 sec - 14 min 34 sec
0.01	18 min 16 sec - 20 min 13 sec	23 min 53 sec - 38 min 16 sec

Tabella 4.5: Analisi delle durate delle simulazioni

Dato che il processore presente nella macchina utilizzata per eseguire le simulazioni è di tipo dual core, è necessario raddoppiare i valori temporali

riferiti alle simulazioni effettuate da TimeNET. A partire dai risultati riportati in Tabella 4.5, si evince come i due simulatori impieghino pressoché lo stesso tempo necessario per il completamento della simulazione.

Tuttavia, i valori che descrivono le durate di simulazione eseguite da TimeNET, sono caratterizzati da range meno ampi rispetto a quelli ricavati a partire dai risultati ottenuti dal simulatore PFNetS. Ciò, a nostro avviso, è dovuto al fatto che il criterio di arresto di simulazione del simulatore PFNetS si basa esclusivamente su osservazioni relative alle probabilità stazionarie riferite ad un insieme fnito di marcature raggiungibili durante la simulazione senza effettuare alcuna analisi del comporatmento dell' indice di prestazione preso in esame.

In conclusione, è possibile affermare che PFNetS è in grado di fornire valori di stima caratterizzati da una maggiore precisione rispetto ai risultati offerti da TimeNET a discapito di una durata di simulazione superiore nei casi in cui entrambi i simulatori vengano eseguiti su macchine non single core.

Tuttavia la capacità di TimeNET di suddividere la simualazione in processi eseguibili in parallelo rappresenta una notevole potenzialità dell'applicazione in quanto con lo sviluppo e la diffusione di tecnologie multicore il rapporto tra il tempo impiegato e la bontà dei risultati offerti può notevolmente migliorare a discapito dei simulatori che non prevedono tale vantaggio.

Capitolo 5

Conclusioni

L'obiettivo del lavoro svolto con il collega Massimo Mariuzzo è stato quello di implementare un simulatore per reti di Petri stocastiche in forma prodotto e di sfruttare le informazioni ricavabili da tali tipologie di reti per la definizione di due differenti criteri di arresto di simulazione. In particolare, tali informazioni sono rappresentate dalle probabilità stazionarie di stato analitiche a meno della costante di normalizzazione ottenibili mediante l'applicazione del teorema di Coleman Henderson et al.

Per raggiungere tale obiettivo è stato quindi effettuato uno studio dettagliato del comportamento e delle proprietà delle reti di Petri dedicando particolare attenzione alle reti di Petri stocastiche esprimibili in forma prodotto.

Mediante l'utilizzo del linguaggio ad oggetti Java, è stato implementato il simulatore PFNetS in grado di rappresentare una generica SPN e di studiare l'andamento dei principali indici di prestazione del sistema reale modellato da quest'ultima successivamente al raggiungimento di uno stato di equilibrio.

Al fine di ottenere lunghezze di prove di simulazione sufficentemente grandi da garantire una stima affidabile dei risultati offerti in output, il simulatore PFNetS è stato integrato con il metodo della matrice ed il metodo del vettore che implementano rispettivamente i due criteri di arresto proposti. Il funzionamento di tali metodi è basato su un numero arbitrario e finito di marcature raggiungibili dalla SPN caratterizzate dai maggiori valori di probabilità analitica stazionaria a meno della costante di normalizzazione. L'insieme di tali marcature è restituito in output da un particolare algoritmo che prevede un'analisi dei risultati ottenuti da un run pilota di simulazione.

I metodi per l'arresto della simulazione sono stati confrontati al fine di analizzarne il costo computazionale, i tempi di esecuzione e la precisione dei risultati offerti dal simulatore che ne prevede l'utilizzo. Dalle analisi dei risultati offerti dal simulatore PFNetS, emerge che l'utilizzo del metodo della matrice è preferibile nel caso in cui si vuole ottenere una migliore accuratezza nei risultati, mentre, si predilige il metodo del vettore nel caso in cui si

vuole ottenere un costo computazionale limitato a discapito di risultati meno precisi.

Le fasi di convalida e di analisi dei risultati offerti dal simulatore PFNetS, entrambe effettuate a partire da differenti tipologie di SPN, hanno permesso di affermare che il simulatore implementato rappresenta adeguatamente il comportamento del sistema reale oggetto di analisi e fornisce un elevato grado di accuratezza dei risultati forniti in output.

Al fine di comprendere la potenzialità e l'applicabilità del simulatore PFNetS basato sui due differenti criteri di arresto implementati, è stato effettuato un confronto con il tool TimeNET il quale ad oggi è considerato lo strumento di simulazione più affidabile e maggiormente utilizzato nel campo delle reti di Petri. Sebbene il tool sia uno strumento caratterizzato da un maggior numero di funzionalità rispetto al simulatore presentato nella tesi, possiamo concludere che il modello di simulazione da noi sviluppato rappresenta a nostro avviso uno strumento più preciso nella stima degli indici di prestazione in relazione esclusivamente ad un'analisi di reti di Petri stocastiche in forma prodotto. A partire da tale risultato, possiamo sostenere di aver quindi sfruttato nel migliore dei modi le informazioni ricavabili dalla soluzione in forma prodotto ottenibile mediante l'applicazione del teorema di Coleman Henderson et al. Infatti, utilizzando le probabilità di stato stazionarie analitiche non normalizzate in relazione con le probabilità di stato stazionarie stimate da PFNetS abbiamo ottenuto risultati soddisfacenti che ci portano a concludere che i due criteri di arresto proposti nell'elaborato rappresentano metodi affidabili per garantire simulazioni di lunghezza limitata che forniscono in output stime accurate sia delle probabilità stazionarie di stato stazionarie sia dei principali indici di prestazione.

A partire da un'analisi delle metodologie utilizzate dai tool di simulazione (TimeNET, PIPE [14], ecc.) per la stima dei principali indici di prestazione, possiamo concludere che i metodi proposti nell'elaborato per il raggiungimento dello stesso obiettivo risultano essere a nostro avviso più affidabili. Infatti, tali metodi basano il loro funzionamento sull'osservazione diretta del comportamento della rete a differenza dei metodi utilizzati nei tool che prevedono la stima degli indici di prestazione tramite l'uso di formule analitiche.

Concludendo, possiamo affermare di aver raggiunto l'obiettivo prefissato, riconoscendo tuttavia di aver realizzato uno strumento utilizzabile esclusivamente per un limitato sottoinsieme di reti di Petri stocastiche quali SPN in forma prodotto.

Analizzando i tool di simulazione esistenti emerge che le funzionalità e le potenzialità offerte da quest'ultimi sono nettamente superiori rispetto al simulatore proposto in quanto forniscono interfacce grafiche usabili e metodi che permettono di analizzare molteplici proprietà strutturali della rete di interesse (Invarianti, grafo di raggiungibilità, ecc.). Tale mancanza può essere facilmente colmata mediante future implemetazioni di metodi da integrare al simulatore PFNetS. Possono essere inoltre apportati miglioramenti relativi

al costo computazionale dei metodi proposti sia per la terminazione del run di simulazione sia per la stima degli indici di prestazione della rete.

Tali miglioramenti sono stati introdotti nell'elaborato e coinvolgono le strutture dati utilizzate dai metodi di arresto nonché possibili ottimizzazioni del codice.

Bibliografia

[1] Murata Tadao. (April 1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE, Vol. 77, No. 4*, pages 541-580.

[2] J. L. Coleman, W. Henderson, and P. G. Taylor. (1996). Product form equilibrium distributions and a convolution algorithm for Stochastic Petri nets. *Perform. Eval., Elsevier,* 26:159–180.

[3] G. Balbo. (May 2001). Introduction to Stochastic Petri Nets, *volume LNCS-2090, Springer-Verlag,* pages 84-155. Berlino.

[4] G. Memmi G. Roucairol, (1980) Linear algebra in net theory, *Net Theory and Applications, Ed: W. Brauer, Leacture Notes in Computer Science 84.*

[5] Jiacun Wang (2007) Petri Nets for Dynamic Event-Driven System Modeling *Handbook of Dynamic System Modeling, Ed: Paul Fishwick, CRC Press.*

[6] Marco Ajmone Marsan (1990) Stochastic Petri Nets: an Elementary Introduction *G.Rozenberg (Editor), Advances in Petri Nets 1989, Springer Verlag, 'Lecture Notes in Computer Science' n. 424.*

[7] Michel Diaz (2007) Petri Nets: Fundamental Models, Verification and Applications *Ed: John Wiley & Sons Incorporated.*

[8] Averill M. Law, W. David Kelton (2000) Simulation Modeling and Analysis. *Mc Graw Hill* Capitolo 9.

[9] David M. Levine, Timothy C. Krehbiel, Mark L. Berenson (2006) Statistica *Ed: Apogeo*

[10] P. Heidelberger, P. D. Welch (1981) A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM, vol. 24(4)*, pp. 233–245.

[11] C. Kelling, (1994) Control variates selection strategies for timed Petri nets, *Proc. European Simulation Symposium, Istanbul*, pp. 73–77.

[12] R. German and J. Mitzlaff, (1995) Transient analysis of deterministic and stochastic Petri nets with TimeNET, *Proc. Joint Conf. 8th Int. Conf. on Modelling Techniques and Tools for Performance Evaluation, Lecture Notes in Computer Science. Springer Verlag,* vol. 977, pp. 209–223.

[13] A. Zimmerman, M. Knoke (2007) TimeNET 4.0: A software Tool for the Performability Evaluation with Stochastic and Coloured Petri Nets; user manual Tecnische Universitat Berlin.

[14] N.J. Dingle, W.J. Knottenbelt and T. Suto. (2009) PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets. *ACM SIGMETRICS Performance Evaluation Review*, Vol. 36(4), pp. 34-39.

[15] J. Mairesse, H-T. Nguyen, (2009) Deficiency Zero Petri Nets and Product Form. Petri Nets, *LNCS 5606*, pp. 103-122.

[16] W. Henderson, D. Lucic, P.G. Taylor. (1989) A net level performance analysis of stochastic Petri nets. *Journal of Australian Mathematical*, pp. 176–187.

[17] James Bucklew (2010) Introduction to Rare Event Simulation *Ed: Springer.*

[18] M .A. Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis (1995) Modelling with generalized stochastic Petri nets. *Ed. Wiley Series*

[19] G. Bolch, S. Greiner, H. de Meer and K.S. Trivedi. (1998) Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications *John Wiley*, New York, NY.