



UNIVERSITÀ DI PISA

Dipartimento di Matematica

Laurea magistrale in Matematica

MARKOVIAN BINARY TREES: An algorithmic approach

Tesi di Laurea Magistrale

Candidato

Rolando F. Ichpas Tapia
ichpastapia@mail.dm.unipi.it

Relatore

Prof. Beatrice Meini
Università di Pisa

Anno Accademico 2016/2017

Indice

Introduzione Generale	i
1 Branching Processes	1
1.1 Introduzione	1
1.2 Il processo di Galton-Watson	1
1.2.1 Distribuzione di probabilità transitoria di Z_n	4
1.2.2 Momenti di Z_n	4
1.2.3 La probabilità di estinzione	5
1.3 The One-Dimensional Continuous-Time Markovian Branching Processes (ctMBP)	6
1.3.1 Momenti di $Z(t)$	11
1.3.2 La probabilità di estinzione	11
1.4 The Continuous-Time Markovian Multi-type Branching Processes (ctMMTBP)	12
1.4.1 Calcolo del primo momento di $Z(t)$	14
1.4.2 La probabilità di estinzione	16
2 Matrix Analytic Methods: Una introduzione	17
2.1 Dalla Exponential Distribution alla Phase-Type Distributions	17
2.1.1 La Exponential Distribution	18
2.1.2 Le Phase-Type Distributions	20
2.2 Dal processo di Poisson alle Markovian Arrival Processes	25
2.2.1 Costruzione del processo di Poisson	26
2.2.2 Markovian Arrival Processes (MAPs)	32
2.2.3 Transient Markov Arrival Processes	36
3 Quasi-Birth-and-Death Processes	43
3.1 Level Independent Quasi-Birth-and-Death Processes - (LIQBD)	43
3.1.1 LIQBD a tempo discreto	43
3.1.2 LIQBD a tempo continuo	48
3.2 Algoritmi per LIQBD	51
3.3 Level Dependent Quasi-Birth-and-Death Processes - (LDQBD)	62
3.3.1 LDQBD a tempo discreto	62
3.3.2 LDQBD a tempo continuo	66
3.4 Algoritmi per LDQBD	67

4	Markovian Binary Trees	71
4.1	Markovian Binary Trees and Branching Processes	75
4.2	Regolarità e Numero medio di Branches	76
4.3	La probabilità di estinzione	77
4.4	Approcci algoritmici per il MBT	77
4.4.1	Algoritmi con convergenza lineare	78
4.4.2	Sperimentazione numerica con gli algoritmi lineari	90
4.4.3	Algoritmi con convergenza quadratica	95
4.4.4	Sperimentazione numerica con gli algoritmi quadratici	101
4.4.5	Confronto tra gli algoritmi lineari e gli algoritmi quadratici	104
5	Analisi transitorio dei Markovian binary trees	109
5.1	Derivati di matrici	109
5.2	Dimensione della Popolazione al tempo \mathbf{T}	110
5.2.1	Momenti Fattoriali	112
5.2.2	Dimensione della Popolazione condizionata all'estinzione	113
5.3	Tempo fino all'estinzione: T_e	114
5.4	Dimensione totale della progenie	115
5.5	Un'applicazione alla demografia	118
6	Modellazione dell'estinzione nel processo di diffusione delle informazioni nei Wireless Sensor Networks	121
6.1	Introduzione	121
6.2	Descrizione del sistema	121
6.3	Modello analitico	122
6.4	Sperimentazione numerica	124
6.4.1	Probabilità di estinzione delle informazioni	125
6.4.2	Numero medio di Nodi con le informazioni	125
6.4.3	Numero medio totale di Nodi che avevano le informazioni	126
6.4.4	Tempo fino all'estinzione delle informazioni	126
6.4.5	Tempo medio per raggiungere " K " nodi prima dell'estinzione	129
6.5	Conclusioni	130
	Appendice	131
	A Funzione generatrice di probabilità e Convoluzione	133
	B Tree Topology	137
	Bibliografia	141

Introduzione Generale

È ben noto che qualsiasi modello realistico di un fenomeno del mondo reale deve prendere in considerazione la possibilità di aleatorietà, cioè, molto spesso le misure che ci interessano non sono prevedibili in anticipo, ma esibiscono una variazione intrinseca che deve essere presa in considerazione nel modello. Questo può essere ottenuto mediante l'uso dei processi stocastici, tra cui, i Processi di Ramificazione (*Branching Processes*), e più in generale, le catene di Markov che svolgono un ruolo molto importante a causa delle loro caratteristiche strutturali che permettono di bilanciare la dipendenza del modello con la manipolazione matematica.

La teoria dei modelli probabilistici di Ramificazione inizia ad essere elaborata nella seconda metà del XIX secolo, al fine di dare una risposta completa da un punto di vista scientifico, al problema dell'estinzione di alcune linee di famiglie dell'aristocrazia e della borghesia europea, essendo i suoi precursori I.J. Bienaymé, F. Galton e H. Watson. Il loro studio attualmente occupa un posto di rilievo all'interno della teoria della probabilità e della statistica matematica con numerosi articoli pubblicati sulla sua teoria e applicazioni. Tra i quali citeremo quelli di Harris (1963, 1989), Mode (1971), Sevast'yanov (1971), Athreya & Ney (1972), Jagers (1975), Asmussen & Hering (1983), Sankaranarayanan (1989), Guttorp (1991), Athreya & Jagers (1997) o la di Heyde (1995). I *Branching Processes* descrivono l'evoluzione delle popolazioni che si riproducono e muoiono indipendentemente l'uno dall'altro in base a determinate leggi di probabilità specifiche e svolgono un ruolo sempre più importante in diversi settori scientifici come: la biologia molecolare, ecologia, epidemiologia, teoria evolutiva, fisica delle particelle, chimica e informatica. Anche se molti modellisti matematici accettano con grande entusiasmo i *Branching Processes*, sono spesso frenati dalla difficoltà che sorgono nell'utilizzo della teoria generale. La maggior parte di queste difficoltà sono di tipo numerico, ad esempio, problemi di base come trovare la distribuzione completa del numero di particelle presenti per un tempo arbitrario sembra intrattabile e scoraggia l'applicazione.

D'altra parte, i "*Matrix analytic methods*" sono stati introdotti per prima volta da Marcel F. Neuts e furono sviluppati prima nel contesto dei "*queueing models*" negli anni 70 dando origine alla teoria dei *Quasi-Birth-and-Death processes (QBDs)* (Neuts 1981, 1989). Da allora, sono diventati uno strumento indispensabile nella modellazione stocastica e hanno trovato applicazione nell'analisi e design di sistemi di produzione, reti di telecomunicazione, modelli di rischio/assicurazione, modelli di affidabilità e sistemi di inventario, ecc. La potenza e la popolarità di questi metodi provengono dalla loro flessibilità nella modellazione stocastica, capacità per l'esplorazione analitica, dal pensiero algoritmico naturale e dalla trattabilità nel calcolo numerico. Sono state pubblicate numerose monografie sui "*Matrix analytic methods*". Tra i quali citeremo a Neuts (1981, 1989), Latouche & Ramaswami (1999), Ost (2001), Grassmann (2000), Bini-Latouche-

Meini (2005), Breuer & Baum (2005), Tian & Zhang (2006), Artalejo & Gomez-Corral (2008), Alfa (2010) e Li (2010).

L'intersezione di queste due teorie ha dato origine a una delle teorie più studiate attualmente nel campo della probabilità applicata, i *Markovian Trees*. I *Markovian Trees (MT)* formano una classe particolarmente trattabile di processi di ramificazione a tempo continuo che sono stati definiti per prima volta da Kontoleon (2005) e furono utilizzati per modellare processi evolutivi. In un *MT*, le epoche di vita e di riproduzione di ogni singolo individuo sono controllate da un *continuous-time Markov chain* con un numero finito di stati transitori chiamati anche *fasi* e inoltre, *una fase assorbente*. Queste fasi possono avere un'interpretazione fisica, come l'età, la condizione di salute degli individui o possono essere puramente fittizie. Durante la loro vita, gli individui si riproducono e subiscono transizioni tra le fasi transitorie finché entrano nella fase assorbente e muoiono. Così, la rappresentazione di un *Branching Process* come un *Markovian Tree* consente a un campo quasi privo di approcci algoritmici di essere sottoposto alle potenti tecniche della *Matrix Analytic theory*.

Quando gli individui in un (*MT*) possono generare al massimo un figlio alla volta, si parla di un "*Markovian Binary Tree*" (*MBT*). Questi formano una classe importante di processi di ramificazione e saranno l'oggetto principale di questa tesi. Pertanto, lo scopo di questa tesi è quello di studiare le "*Markovian Binary Tree*" dalla sua genesi e anche esporre i principali risultati algoritmici sviluppati fino ad oggi, e alla fine, mostrare con una applicazione concreta ai *Wireless Sensor Networks* la sua versatilità nella modellazione stocastica.

La tesi è stata strutturata in sei Capitoli e due appendici.

Nel Capitolo 1, forniamo una panoramica dei "*Branching Processes*", iniziamo con il processo di Galton-Watson, che è il più semplice di tutti, e poi studiamo al *One-Dimensional Continuous-Time Markovian Branching Process (ctMBP)* e finiamo con l'introduzione del *Continuous-Time Markovian Multi-type Branching Process (ctMMTBP)*, in ognuno di essi si studia: la probabilità di estinzione e il calcolo dei momenti.

Nei Capitolo 2 e 3, si studia la *Matrix Analytic theory*. Il capitolo 2 introduce le *Phase-Type Distributions* come una generalizzazione della *Exponential Distribution* e analogamente i *markovian Arrival Processes* vengono introdotti come una generalizzazione del *Poisson Process*, mentre, nel capitolo 3 si studiano i *Quasi-Birth-and-Death process* con la derivazione dei principali algoritmi esistenti nella letteratura matematica su questo argomento, che sono: l'algoritmo di Neuts, l'algoritmo $U - U^*$ e l'algoritmo *LR (Logarithmic Reduction)*, sia per il caso discreto che per il caso continuo e si fa anche un breve confronto tra di loro.

Nei Capitoli 4, 5 e 6 trattiamo nel dettaglio, anche negli aspetti computazionali; i *Markovian Binary Trees*. Nel capitolo 4 si definisce i *Markovian Binary Trees*, si studiano le sue caratteristiche principali e il legame con i *Continuous-Time Markovian Multi-type Branching Process (ctMMTBP)*, si determinano le condizioni per una eventuale estinzione del processo e si analizzano i principali algoritmi per approssimare la probabilità di estinzione. Nel capitolo 5, si studiano le *transient characteristics* più importanti dei *MBT* come: *Population size, time until extinction, total progeny size* e concludiamo con una breve applicazione alla dinamica di popolazione. Infine, nel capitolo 6, si modella l'estinzione del processo di diffusione delle informazioni in una "*Wireless Sensor Network*".

Ci teniamo a precisare che gli algoritmi presentati in questa tesi sono stati completamente sviluppati utilizzando il linguaggio *MatLab* e *C#*.

*Dedicated to my parents and brothers, whom with their achievements taught me...
that with perseverance, any dream is achievable.*

Capitolo 1

Branching Processes

1.1 Introduzione

I processi di Ramificazione (*Branching Processes*) costituiscono uno dei campi classici della probabilità applicata e sono ancora un'area attiva di ricerca. Dato che questo settore è cresciuto così ampio e diversificato, in questa tesi si fa solo un'introduzione ai concetti generali che ci aiuterà a comprendere la costruzione dei *Markovian Trees*. I teoremi e le proposizioni viste qui non verranno dimostrati, un trattamento più completo su questo argomento insieme alle dimostrazioni può essere trovato in [4] e [19]. In senso lato, un processo di ramificazione potrebbe essere descritto come un modello stocastico per lo sviluppo di una popolazione, i cui membri di solito si chiamano individui o particelle, e che si riproducono in modo indipendente. Il prototipo di un processo di Ramificazione modellando un fenomeno nella sua forma più pura è il semplice processo di Galton-Watson o, storicamente più corretto, il processo di Bienaymé-Galton-Watson.

1.2 Il processo di Galton-Watson

Vogliamo studiare degli oggetti che possono riprodursi generando oggetti dello stesso tipo. Può trattarsi in generale di esseri umani, batteri od anche neutroni nelle reazioni nucleari. Partiamo da un insieme iniziale di oggetti che corrisponde alla generazione 0. I figli di questi oggetti individuano la prima generazione e così via. Denoteremo con Z_0, Z_1, Z_2, \dots il numero d'individui alla zero, prima, seconda generazione, ecc... Facciamo le seguenti assunzioni:

1. Se supponiamo noto il numero di oggetti nella generazione n , la legge di probabilità per le generazioni seguenti dipenderà solo dal numero d'individui della $n - \text{esima}$ generazione e non da quelle precedenti. Ciò rende le Z_0, Z_1, Z_2, \dots una catena di Markov. Inoltre, assumeremo a meno di specificazioni diverse che le probabilità di transizione non varino nel tempo (omogeneità della catena).
2. Assumeremo che i diversi oggetti abbiano evoluzioni del tutto indipendenti e non interferiscano l'uno con l'altro, in particolare il numero di figli di un oggetto non dipenda da quanti oggetti sono presenti.

Possiamo quindi procedere con la formalizzazione matematica.

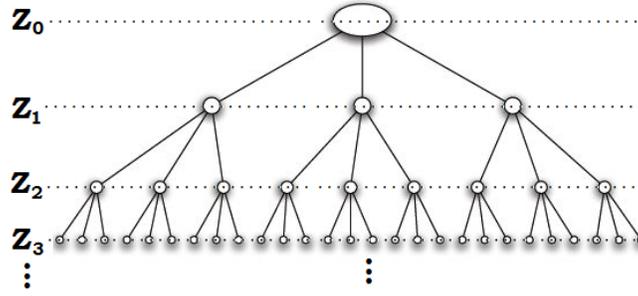


Figure 1.1: A schematic representation of a likely trajectory.

Descrizione matematica del processo di Galton-Watson

Sia Z_n il processo di Markov a tempo discreto con le condizioni di cui sopra. Salvo diversamente specificato, si assume che $Z_0 = 1$, quindi è chiaro che lo spazio degli stati $\mathcal{S} = \{0, 1, 2, \dots\}$. Dato che gli individui si riproducono indipendentemente ciascuno secondo la stessa *distribuzione* F (concentrato su \mathcal{S}), possiamo rappresentare il numero di figli del ℓ -esimo individuo della n -esima generazione con la v.a $X_{n,\ell}$, quindi

$$Z_{n+1} = \sum_{\ell=1}^{Z_n} X_{n,\ell} \quad \forall n \in \mathbb{N}_0, \quad (1.2.1)$$

ove gli $X_{n,\ell}$ è una famiglia di variabili aleatorie i.i.d, ognuno con distribuzione F . Il processo Z_n si chiama **Galton-Watson Process**, e le sue realizzazioni sono spesso raffigurati come *genealogical trees* (vedere Fig.(1.1)).

Notare che in questo modello gli individui Z_n della n -esima generazione “muoiono” al tempo $n + 1$ in quanto non sono considerati nella somma (1.2.1). Al fine di mantenere anche loro nella generazione successiva, avremmo dovuto modificare (1.2.1) in

$$Z_{n+1} = Z_n + \sum_{\ell=1}^{Z_n} X_{n,\ell} \quad \forall n \in \mathbb{N}_0,$$

tuttavia, non adatteremo questa convenzione e ci concentreremo soltanto in (1.2.1). Questo metodo di costruzione del processo si chiama *forward equation* giacché l’analisi è fatta dalla prima generazione in avanti; un approccio alternativo è basato su un’analisi indietro chiamata *backward equation* che a volte risulta di essere più adatto per processi più generali. (vedere [34]).

Sia \mathbb{P} la misura di probabilità per il nostro processo, allora possiamo definire una *funzione di densità di probabilità* $(p_k)_{k \in \mathbb{N}_0}$ per F come segue:

$$\mathbb{P}\{X_{n,\ell} = k\} = p_k \quad \forall k \in \mathbb{N}_0,$$

dove $0 \leq p_k \leq 1$ e $\sum_{k=0}^{\infty} p_k = 1$. In questo caso p_k rappresenta la probabilità che un qualsiasi individuo di qualsiasi generazione “ n ” abbia “ k ” figli, e stiamo anche assumendo che p_k non dipenda dalla generazione n .

Con tutta l’informazione di cui sopra è possibile ottenere una distribuzione di probabilità di tran-

sizione per il processo, il quale viene indicata con

$$P_{ij} = \mathbb{P}\{Z_{n+1} = j | Z_n = i\} = \mathbb{P}\left\{\sum_{\ell=1}^i X_{n,\ell} = j\right\} = \begin{cases} p_j^{*i} & \text{se } i \geq 1, \quad j \geq 0, \\ \delta_{0j} & \text{se } i = 0, \quad j \geq 0, \end{cases} \quad (1.2.2)$$

dove p_j^{*i} rappresenta la j -esima componente della i -esima convoluzione di $\{p_k : k = 0, 1, \dots\}$ (Vedere (A.0.1) dell'appendice(A)) e δ_{ij} rappresenta il delta di Kronecker. Quindi, la matrice di transizione del processo Z_n sarà:

$$P = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & \dots \\ p_0^{*1} & p_1^{*1} & p_2^{*1} & \dots & p_n^{*1} & \dots \\ p_0^{*2} & p_1^{*2} & p_2^{*2} & \dots & p_n^{*2} & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ p_0^{*n} & p_1^{*n} & p_2^{*n} & \dots & p_n^{*n} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (1.2.3)$$

Notare che $p_k^{*1} = p_k$ per ogni $k = 0, 1, 2, \dots$. Dalla matrice P si vede che lo stato 0 è uno *stato assorbente* poiché per costruzione abbiamo sempre che:

$$\mathbb{P}\{Z_{n+1} = 0 | Z_n = 0\} = 1, \quad \forall n \in \mathbb{N} = \{1, 2, \dots\}.$$

Avendo già definito il processo, ora siamo interessati a conoscere alcune delle sue proprietà:

- I.- La distribuzione di probabilità di Z_n per ogni $n \in \mathbb{N}$.
- II.- I momenti di Z_n per ogni $n \in \mathbb{N}$.
- III.- La probabilità che la successione aleatoria Z_0, Z_1, Z_2, \dots eventualmente converga a zero (estinzione).

Funzioni Generatrici di Probabilità

Una delle funzioni che ci aiuterà a fare l'analisi del comportamento del processo GW è nota come: **funzione generatrice di probabilità**. Vedere l'appendice (A).

Quindi, sia f la funzione generatrice di probabilità della v.a $X_{0,1} = Z_1$, data da:

$$f(t) = \mathbb{E}\{t^{Z_1}\} = \sum_{n=0}^{\infty} p_n t^n, \quad |t| \leq 1.$$

Si definisce **the iterates** di f come segue:

- $f_0(t) = I(t) = t$.
- $f_1(t) = f(t)$.
- $f_{n+1}(t) = f[f_n(t)] \quad n = 1, 2, \dots$

Si può facilmente verificare che ogni funzione iterata f_n è ancora una funzione generatrice di probabilità e che le seguenti relazioni sono una conseguenza immediata

$$f_{m+n}(t) = f_m[f_n(t)], \quad m, n = 0, 1, \dots,$$

Il seguente risultato permette di caratterizzare a Z_n per ogni $n \in \mathbb{N}_0$ attraverso l'iterate di f .

Teorema 1.1. *La funzione generatrice di probabilità di Z_n è la n -esima iterata di f , cioè f_n , per ogni $n \in \mathbb{N}$.*

□

Il teorema 1.1 permette di calcolare la funzione generatrice di probabilità e, pertanto la distribuzione di probabilità di Z_n per ogni $n = 1, 2, \dots$; inoltre, tutte le proprietà delle funzioni di transizione P_{ij}^n sono contenute nella sua funzione generatrice di probabilità $f_n(t)$. Nonostante ciò, solo raramente è possibile trovare la funzione f_n in una semplice forma esplicita; tuttavia, alcune proprietà come i momenti e il comportamento asintotico del processo sono calcolabili tramite le f_n .

Prima di iniziare l'analisi del comportamento di Z_n dobbiamo fare 2 “*basic assumptions*”:

Basic assumptions:

- a.- Nessuna delle probabilità p_0, p_1, \dots è uguale a 1, e $p_0 + p_1 < 1$. Quindi f è strettamente convessa sull'intervallo unità.
- b.- Il valore atteso $\mathbb{E}\{Z_1\} = \sum_{k=0}^{\infty} k p_k < \infty$ (è finito). Ciò implica che la derivata $f'(1)$ è finita. I simboli $f'(1), f''(1), \dots$ di solito si riferiscono alle derivate a sinistra di f in $t = 1$, perché abbiamo considerato che $|t| \leq 1$.

1.2.1 Distribuzione di probabilità transitoria di Z_n

Per calcolare la distribuzione di probabilità transitoria $\pi^{(n)}$ di Z_n possiamo usare le equazioni di Chapman-Kolmogorov. Quindi,

$$\pi^{(n)} = \pi^{(0)} P^n,$$

dove P è la matrice di transizione data in (1.2.3) e $\pi^{(0)}$ è la distribuzione di probabilità iniziale, che in questo caso sarebbe $\pi^{(0)} = (0, 1, 0, 0, 0, \dots)$ dato che abbiamo supposto $Z_0 = 1$. Un'altra forma per calcolare la distribuzione potrebbe essere quella di usare le funzioni generatrici f_n di Z_n .

1.2.2 Momenti di Z_n

In questa parte ci focalizzeremo solo sulla ricerca del valore atteso e la varianza di Z_n per ogni $n \in \mathbb{N}$. Per la seconda assunzione che abbiamo fatto in (1.2), $\mathbb{E}\{Z_1\} < \infty$, questo ci permette di definire:

Definizione 1.1. Denotiamo la media e la varianza di Z_1 come segue:

$$m = \mathbb{E}\{Z_1\}, \quad \sigma^2 = \mathbb{E}\{Z_1^2\} - m^2,$$

sempre che $\mathbb{E}\{Z_1^2\} < \infty$.

Osservazione:

1. Dalle proprietà delle funzioni generatrice di probabilità (A) abbiamo che $m = f'(1)$ e $\sigma^2 = f''(1) + m - m^2$, dove f è la funzione generatrice di probabilità di Z_1 .
2. Dobbiamo fare attenzione che data una v.a X in generale, l'esistenza di $\mathbb{E}\{X\}$ non garantisce l'esistenza di $\mathbb{E}\{X^2\}$, invece l'esistenza di $\mathbb{E}\{X^2\}$ garantisce l'esistenza di $\mathbb{E}\{X\}$ (vedere [30]).

Ora, usando la proposizione (1.1) possiamo dimostrare il seguente teorema.

Teorema 1.2. Sia Z_n un processo di GW, allora:

- **Media:** $\mathbb{E}\{Z_n\} = m^n, \quad n = 1, 2, 3, \dots$
- **Varianza:** $Var(Z_n) = \begin{cases} \frac{\sigma^2 m^n (m^n - 1)}{m^2 - m} & , \text{ se } m \neq 1 \\ n\sigma^2 & , \text{ se } m = 1 \end{cases}, \quad \text{sempre che } \sigma^2 < \infty$

1.2.3 La probabilità di estinzione

Definizione 1.2 (Estinzione). Sia Z_n un processo GW, definiamo come **estinzione** l'evento E per cui la sequenza $\{Z_n\}$ è diversa da 0 solo per un numero finito di termini, cioè, è definitivamente 0. Indicheremo con q la sua probabilità: $\mathbb{P}\{E\} = q$.

□

Poiché Z_n è un valore intero, l'evento che $Z_n \rightarrow 0$ è anche un'estinzione, inoltre, come $\mathbb{P}\{Z_{n+1} = 0 | Z_n = 0\} = 1$, abbiamo le seguenti uguaglianze:

$$\begin{aligned} \mathbb{P}\{Z_n \rightarrow 0\} &= \mathbb{P}\{Z_n = 0 \text{ per alcun "n"}\} \\ &= \mathbb{P}\{(Z_1 = 0) \cup (Z_2 = 0) \cup \dots\} \\ &= \lim_{n \rightarrow \infty} \mathbb{P}\{(Z_1 = 0) \cup \dots \cup (Z_n = 0)\} \\ &= \lim_{n \rightarrow \infty} \mathbb{P}\{(Z_n = 0)\} = \lim_{n \rightarrow \infty} f_n(0). \end{aligned}$$

Quindi abbiamo che $q = \mathbb{P}\{Z_n \rightarrow 0\} = \lim_{n \rightarrow \infty} f_n(0)$, e siccome $f_n(0)$ è una funzione non decrescente su n , abbiamo il seguente teorema.

Teorema 1.3. Sia $m = \mathbb{E}\{Z_1\}$, allora

- Se $m \leq 1 \Rightarrow q = 1$.
- Se $m > 1 \Rightarrow q$ è l'unica soluzione non negativa minore di 1 dell'equazione $t = f(t)$.

Il teorema appena visto garantisce l'estinzione con probabilità 1 quando $m \leq 1$, ma quando $m > 1$ potrebbe non avere estinzione, vediamo il seguente teorema:

Teorema 1.4. *Sia $m = \mathbb{E}\{Z_1\} < \infty$, allora:*

$$\lim_{n \rightarrow \infty} \mathbb{P}\{Z_n = i\} = 0, \quad i = 1, 2, \dots$$

Inoltre, $Z_n \rightarrow \infty$ con probabilità “ $1 - q$ ” (esplosione) e $Z_n \rightarrow 0$ con probabilità “ q ” (estinzione).

□

Osservazione:

Val la pena di notare che questo teorema fornisce un risultato per il valore asintotico della distribuzione di Z_n indipendente dal valore finito di m , cioè dal numero medio di oggetti che un oggetto può produrre. Esso asserisce quindi che, al limite, o la popolazione si estingue (con la probabilità q già vista) oppure diviene infinita (con la probabilità complementare $1 - q$). Ovviamente se $m \leq 1$, allora $q = 1$ e quindi la popolazione s'estingue certamente. Se invece $m > 1$ la popolazione può estinguersi o diventare infinita, però mai rimanere in uno stato transitorio $i = 1, 2, \dots$. Ci riferiremo al processo di GW come:

- *Sub-critical*, quando $m < 1$.
- *Critical*, quando $m = 1$.
- *Super-critical*, quando $m > 1$.

1.3 The One-Dimensional Continuous-Time Markovian Branching Processes (ctMBP)

Nel capitolo precedente è stato studiato il processo di ramificazione di Galton-Watson (il più semplice di tutti), dove il periodo di vita di ogni individuo è uguale ad una unità di tempo. Come possiamo vedere, questo processo non è sufficiente per modellare la crescita di molte popolazioni fisiche o biologiche, motivo per cui è necessario un'estensione della teoria del processo di ramificazione di Galton-Watson per il tempo continuo, noto come: *The One-Dimensional Continuous-Time Markovian Branching Process-ctMBP*.

Il processo può essere descritto come segue. Nasce una sola particella progenitrice al tempo $t = 0$ (chiamata generazione zero). Questa particella vive per un tempo τ , questo periodo di tempo di vita ha una distribuzione esponenziale di parametro “ a ”. Al momento della sua morte, la particella produce un numero aleatorio di prole secondo una distribuzione di probabilità con pgf $f(s)$. Ognuna delle prole della prima generazione si comporta indipendentemente l'una dall'altra, allo stesso modo della particella iniziale e vive per un tempo distribuito esponenzialmente e produce un numero aleatorio di prole. La prole di ciascuna delle generazioni successive si comporta allo stesso modo. Se indichiamo con $Z(t)$ la quantità di particelle nel tempo t , otteniamo il processo stocastico $\{Z(t) : t \geq 0\}$.

Definizione 1.3. Un processo stocastico $\{Z(t) : t \geq 0\}$ definito su uno spazio di probabilità $(\Omega, \mathcal{F}, \mathbb{P})$ è detto *One-Dimensional Continuous-Time Markovian Branching Process-ctMBP* se:

- i)* Il suo spazio di stati è l'insieme $S = \{0, 1, 2, \dots\}$.
- ii)* È una catena di Markov stazionaria rispetto ai campi $\mathcal{F}_t = \sigma\{Z(s, \omega) : s \leq t\}$, per qualsiasi collezione “ D ” di variabili aleatorie a valori reali e misurabili secondo Borel definite su $(\Omega, \mathcal{F}, \mathbb{P})$ ($\sigma(D)$ indica il sub- σ -algebra di \mathcal{F} generata per “ D ”).
- iii)* Le probabilità di transizione $P_{ij}(t)$ soddisfano

$$\sum_{j=0}^{\infty} P_{ij}(t)s^j = \left[\sum_{j=0}^{\infty} P_{1j}(t)s^j \right]^i. \quad (1.3.1)$$

Per ogni $i \geq 0$ e $|s| \leq 1$

Le proprietà (i) e (ii) semplicemente dicono che $Z(t)$ è un processo di Markov a tempo continuo con spazi di stati numerabile, quindi una catena di Markov a tempo continuo (vedere [5]), mentre che (iii) caratterizza la proprietà di base della ramificazione. Vediamo di seguito come si fa questa caratterizzazione.

Costruzione del ctMBP

Per la costruzione del ctMBP è fondamentale studiare le probabilità di transizione, che sono date da

$$P_{ij}(\tau, \tau + t) = \mathbb{P}\{Z(\tau + t) = j | Z(\tau) = i\} \quad \tau \geq 0, t \geq 0,$$

il quale nel caso omogeneo (non dipende del tempo attuale) diventa

$$P_{ij}(\tau, \tau + t) = P_{ij}(t) \quad t \geq 0, \quad (1.3.2)$$

La *probabilità* che si verifichi una transizione da uno stato attuale dipende non solo dello stato attuale, ma anche della lunghezza dell'intervallo di osservazione. Consideriamo un periodo di osservazione Δt e un numero iniziale “ i ” di particelle. Se una di queste particelle è viva per un certo tempo, la sua durata di vita supplementare sarà una variabile aleatoria distribuita esponenzialmente con parametro “ $a > 0$ ”. Dopo la sua morte questa particella lascia “ k ” discendenti con probabilità p_k , $k = 0, 1, 2, \dots$. Siccome le particelle si comportano indipendentemente l'una rispetto all'altra, quindi

1. $P_{ij}(\Delta t) = ia p_{j-i+1} \Delta t + o(\Delta t)$ quando $\Delta t \rightarrow 0$, se $j \geq i - 1$, $j \neq i$,
2. $P_{ij}(\Delta t) = 1 - ia \Delta t + o(\Delta t)$ quando $\Delta t \rightarrow 0$,
3. $P_{ij} = o(\Delta t)$ quando $\Delta t \rightarrow 0$, se $j < i - 1$,

Dobbiamo notare che l'intervallo di osservazione è $[t, t+\Delta t)$, p_k è una distribuzione di probabilità e che $o(\Delta t)$ è il “*little oh*”, quest'ultima notazione è definito come una quantità per cui

$$\lim_{\Delta t \rightarrow 0} \frac{o(\Delta t)}{\Delta t} = 0.$$

Il processo appena costruito si chiama *il processo minimal* e si può dimostrare mediante metodi standard che verificano le seguenti equazioni di Kolmogorov:

$$\frac{d}{dt} P_{ij}(t) = -jaP_{ij}(t) + a \sum_{k=1}^{j+1} kP_{ik}(t)p_{j-k+1} \quad (\text{forward}), \quad (1.3.3)$$

e

$$\frac{d}{dt} P_{ij}(t) = -iaP_{ij}(t) + ia \sum_{k=i-1}^{\infty} p_{k-i+1}P_{kj}(t) \quad (\text{backward}), \quad (1.3.4)$$

con condizioni al contorno

$$P_{ij}(0+) = \delta_{ij} = \begin{cases} 1 & \text{per } i = j \\ 0 & \text{per } i \neq j \end{cases} \quad (1.3.5)$$

e come $Z(t)$ è una catena di Markov a tempo continuo, allora $\{P_{ij}(t)\}$ verifica anche l'equazione di Chapman-Kolmogorov

$$P_{ij}(t) = \sum_{k=0}^{\infty} P_{ik}(t_1)P_{kj}(t), \quad i, j = 0, 1, 2, \dots; 0 \leq t_1 \leq t. \quad (1.3.6)$$

Abbiamo la seguente definizione:

Definizione 1.4. *Chiameremo all'insieme delle funzioni $\{P_{ij}(t)\}$ una **soluzione delle equazioni forward e backward di Kolmogorov**, se loro sono non negativi, assolutamente continui in “0” e “t”, soddisfano [(1.3.3),(1.3.5)] o [(1.3.4),(1.3.5)], soddisfano (1.3.6) e la disuguaglianza*

$$\sum_{j=0}^{\infty} P_{ij}(t) \leq 1 \quad i = 0, 1, 2, \dots$$

□

Sfortunatamente, sebbene esistono soluzioni delle equazioni forward e backward di Kolmogorov, esse possono non essere unici. Questi problemi sorgono ad esempio quando esiste la possibilità che il processo produca infinite particelle in un tempo finito (che significherebbe avere infinite transizioni in $\langle 0, t \rangle$), i.e., esplodere (vedere [48] cap. 2), situazione che si presenta quando $\sum_{j=0}^{\infty} P_{ij}(t) < 1$. Tuttavia, sotto un'ipotesi addizionale (chiamata *non-explosion hypothesis*) avremo che la soluzione delle equazioni di Kolmogorov soddisfano $\sum_{j=0}^{\infty} P_{ij}(t) = 1$ per ogni $t \geq 0$, il quale assicura che detta soluzione sarà unica.

Sia la funzione:

$$f(s) = \sum_{j=0}^{\infty} p_j s^j, \quad |s| \leq 1, \quad (1.3.7)$$

questa ipotesi afferma che:

Non-Explosion Hypothesis

Per ogni $\epsilon > 0$

$$\int_{1-\epsilon}^1 \frac{ds}{f(s) - s} = \infty.$$

Allora abbiamo il seguente risultato:

Teorema 1.5. *L'ipotesi di Non-esplosione è una condizione necessaria e sufficiente per avere $\mathbb{P}\{Z(t) < \infty\} \equiv 1$, dove $Z(t)$ è il processo minimal.*

□

La dimostrazione di questo teorema può essere trovata in [19]. Abbiamo un risultato che permette garantire che un processo non sia esplosivo.

Proposizione 1.1. *La condizione*

$$f'(1) < \infty$$

è sufficiente per garantire che il processo non sia esplosivo.

□

D'ora in poi si considererà solo i processi non esplosivi, salvo dove diversamente indicato.

Funzioni Generatrici di Probabilità

La funzione generatrice svolge un ruolo importante nell'analisi dei processi continui. Per definire la funzione generatrice del processo useremo la funzione (1.3.7), che viene indicato come

$$f(s) = \sum_{j=0}^{\infty} p_j s^j, \quad |s| \leq 1,$$

e un'altra funzione $u(s)$ definita come

$$u(s) = a[f(s) - s] \tag{1.3.8}$$

Il numero “ a ” che appartiene all'intervallo $\langle 0, \infty \rangle$, e la funzione di probabilità $\{p_j : j = 0, 1, 2, \dots\}$ sono il totale dei dati del processo $\{Z(t) : t \geq 0\}$.

Si definisce ora la **generating function** del processo $\{Z(t) : t \geq 0\}$ come

$$\begin{aligned} F_k(s, t) &= \sum_{j=0}^{\infty} \mathbb{P}\{Z(t) = j | Z(0) = k\} s^j \\ &= \sum_{j=0}^{\infty} P_{kj}(t) s^j \\ &= \mathbb{E}\{s^{Z(t)}\}. \end{aligned}$$

Notare che nella definizione appena data inizialmente avremo k particelle, se $k = 1$ allora la funzione generatrice viene indicato con $F(s, t)$.

Dall'equazioni di Kolmogorov (1.3.3), (1.3.4) e (1.3.5) possiamo vedere che $F(s, t)$ soddisfa le seguenti relazioni

$$\frac{\partial}{\partial t} F(s, t) = u(s) \frac{\partial}{\partial s} F(s, t) \quad (\text{forward equation}), \quad (1.3.9)$$

e

$$\frac{\partial}{\partial t} F(s, t) = u(F(s, t)) \quad (\text{backward equation}), \quad (1.3.10)$$

con condizione al contorno

$$F(s, 0) = s. \quad (1.3.11)$$

Usando il teorema (1.5) si può dimostrare che (1.3.9), (1.3.10) e (1.3.11) hanno una unica soluzione, il quale è una funzione generatrice di probabilità, inoltre, esiste un unico processo di Markov con questa funzione generatrice.

Un punto importante da considerare è che l'equazione (1.3.10) può essere ottenuto da

$$\int_s^{F(s,t)} \frac{dx}{u(x)} = t, \quad (1.3.12)$$

purché $u(x) \neq 0$ per $s \leq x \leq F(s, t)$, e così, F può anche essere ottenuto dalla soluzione di (1.3.12). Questa relazione è stato usato per Harris (1963) per ricavare la condizione di non-esplosione visto sopra.

Per ottenere l'iterata della funzione generatrice è necessario utilizzare la funzione (1.3.7) e l'equazione di Chapman-Kolmogorov, in questo modo abbiamo

$$\begin{aligned} F(s, t + u) &= \sum_{k=0}^{\infty} P_{1k}(t + u) s^k \\ &= \sum_{k=0}^{\infty} \sum_{j=0}^{\infty} P_{1j}(u) P_{jk}(t) s^k \end{aligned}$$

scambiando l'ordine della somma

$$\begin{aligned} F(s, t + u) &= \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} P_{1j}(u) P_{jk}(t) s^k \\ &= \sum_{j=0}^{\infty} P_{1j}(u) \sum_{k=0}^{\infty} P_{jk}(t) s^k \\ &= \sum_{j=0}^{\infty} P_{1j}(u) [F(s, t)]^j. \end{aligned}$$

Così

$$F(s, t + u) = F(F(s, t), u), \quad |s| \leq 1, \text{ per } u \geq 0, t \geq 0.$$

Questa relazione è l'analogo alla formula di iterazione funzionale per $f_n(s)$ nel processo di ramificazione di Galton-Watson.

1.3.1 Momenti di $Z(t)$

Definiamo il k -esimo momento fattoriale di $Z(t)$, dove $k = 1, 2, 3, \dots$, come

$$m_k(t) = \mathbb{E}\{Z(t)[Z(t) - 1] \dots [Z(t) - k + 1] | Z(0) = 1\}.$$

e consideriamo un'altra volta le funzioni (1.3.7), (1.3.8) e facciamo:

$$m = f'(1) \quad e \quad \lambda = u'(1) = a(m - 1).$$

Se la serie $\sum_{j=0}^{\infty} j^k p_k$ converge fortemente in ogni intervallo finito $\langle 0, t \rangle$, allora $m_k(t)$ è finito e può essere ottenuto differenziando l'equazioni di Kolmogorov per le funzioni generatrici k volte con rispetto a "s" e valutando in $s = 1$.

Teorema 1.6. *Sia $Z(0) = 1$, allora*

- $\mathbb{E}\{Z(t)\} = e^{\lambda t}$.
- $Var[Z(t)] = \begin{cases} \frac{f''(1) - f'(1) + 1}{f'(1) - 1} e^{\lambda t} (e^{\lambda t} - 1) & \text{se } \lambda \neq 0 \\ f''(1)at & \text{se } \lambda = 0 \end{cases}$

Osservazione:

Del teorema appena visto, sia $M(t) = \mathbb{E}\{Z(t)\} = e^{\lambda t}$,

- Se $\lambda > 0$, allora $\lim_{t \rightarrow \infty} M(t) = \infty$ (il processo è detto *super-critical*).
- Se $\lambda = 0$, allora $\lim_{t \rightarrow \infty} M(t) = 1$ (il processo è detto *critical*).
- Se $\lambda < 0$, allora $\lim_{t \rightarrow \infty} M(t) = 0$ (il processo è detto *sub-critical*).

1.3.2 La probabilità di estinzione

In questa sezione risolveremo il problema dell'estinzione delle famiglie per il processo Markoviano di Ramificazione a Tempo Continuo. In primo luogo supponiamo che $p_0 > 0$, poiché altrimenti l'estinzione sarebbe impossibile. Dobbiamo anche prendere in considerazione il caso in cui il processo inizi al tempo zero con un singolo individuo.

Per la proprietà della ramificazione del processo, avremo che

$$P_{i0}(t) = [P_{10}(t)]^i.$$

Intuitivamente possiamo considerare che $P_{i0}(t)$ è una funzione non decrescente al tempo t . Siamo in grado di dimostrarlo formalmente visto che per $h > 0$ abbiamo

$$P_{i0}(t * h) = F_i(0, t + h) = [F(0, t + h)]^i = [F(F(0, h), t)]^i \geq [P_{10}(t)]^i.$$

Definizione 1.5. *La probabilità di estinzione può essere definita come la probabilità che la famiglia generata da un singolo individuo sparisca, matematicamente*

$$q = \lim_{t \rightarrow \infty} P_{10}(t).$$

ove q è la probabilità dell'estinzione. □

Utilizzando la teoria di ramificazione del processo di Galton-Watson, è facile determinare la probabilità dell'estinzione nel caso continuo.

Teorema 1.7. *La probabilità dell'estinzione “ q ” del ctMBP $Z(t)$ è la radice non negativa più piccola dell'equazione $F(s, t_0) = s$, dove t_0 è un numero positivo arbitrario.*

□

Siccome “ q ” è la radice di $F(s, t_0) = s$ per qualsiasi t_0 , allora è possibile calcolarlo da una equazione che non dipenda del tempo. Vediamo il seguente teorema.

Teorema 1.8. *La probabilità dell'estinzione “ q ” è la radice non negativa più piccola dell'equazione $u(s) = 0$. Dove $q = 1$ se e soltanto se $u'(1) \leq 0$.*

□

1.4 The Continuous-Time Markovian Multi-type Branching Processes (ctMMTBP)

Dopo aver discusso il processo di Galton-Watson e il processo ctMBP, siamo ora in grado di introdurre *the Continuous-Time Markovian Multi-type Branching Process* (ctMMTBP). La differenza principale tra questo processo e il processo ctMBP, è che ora ci sono “ n ” diversi tipi di particelle. Supponiamo di avere un processo con “ n ” tipi di particelle, ciascuna particella del tipo $i \in \{1, 2, \dots, n\}$ ha un periodo di vita distribuito esponenzialmente con media $\frac{1}{a_i}$, e al momento della sua morte, produrrà n tipi di discendenti con distribuzione $p^{(i)}(j_1, j_2, \dots, j_n)$, dove $j_k \in \{0\} \cup \mathbb{Z}^+$ rappresenta il numero di particelle del tipo $k \in \{1, 2, \dots, n\}$ che verrà generato. Le particelle dopo esser nate evolvono indipendentemente l'una dall'altra e anche del passato.

La i -esima funzione generatrice di probabilità $f^{(i)}$, determinerà la distribuzione del numero di discendenti dei diversi tipi che produrrà un genitore di tipo “ i ”, quindi abbiamo

$$f^{(i)}(s_1, s_2, \dots, s_n) = \sum_{j_1, j_2, \dots, j_n \geq 0} p^{(i)}(j_1, j_2, \dots, j_n) s_1^{j_1} s_2^{j_2} \dots s_n^{j_n} \quad (1.4.1)$$

$$0 \leq s_\alpha \leq 1, \quad \alpha = 1, 2, \dots, n,$$

ove è chiaro $p^{(i)}(j_1, j_2, \dots, j_n) =$ la probabilità che un genitore del tipo “ i ” produca j_1 particelle del tipo 1, j_2 particelle del tipo 2, ..., j_n particelle del tipo n .

Osservazione:

- Si dice che il processo è **singolare** se le funzioni generatrici (1.4.1) sono costituite soltanto da termini lineari in “ s ” per tutti $i \in \{1, 2, \dots, n\}$.
- Un punto di ramificazione è detto: **punto di ramificazione singolare**, se una particella di tipo “ i ” si trasforma in una particella di tipo “ j ”, per $i \neq j$.

- Un **punto di ramificazione binario** si verifica quando una particella di tipo “ i ” termina e genera due particelle figlie di tipi “ j ” e “ k ”, per ogni $j, k = 1, 2, \dots, n$.

Sia $Z_i(t)$ = il numero di particelle del tipo “ i ” che esistono al tempo “ t ”, allora

$$\mathbf{Z}(t) = (Z_1(t), Z_2(t), \dots, Z_n(t)), \quad t \geq 0,$$

rappresenta il vettore corrispondente alla quantità di particelle degli “ n ” tipi esistenti al tempo “ t ”.

Definizione 1.6. Un processo stocastico $\{\mathbf{Z}(t, \omega) : t \geq 0\}$ su uno spazio di probabilità $(\Omega, \mathcal{F}, \mathbb{P})$ è chiamato un **Continuous-Time Markovian Multi-type Branching Process**, se

- i)* Il suo spazio di stati è l'insieme $\mathcal{R}_n^+ = \{\mathbf{x} \in \mathcal{R}_n : x_i \geq 0, i = 1, 2, \dots, n\}^a$.
- ii)* È un processo di Markov fortemente stazionario rispetto ai campi $\mathcal{F}_t = \sigma\{\mathbf{Z}(s, \omega) : s \leq t\}$.
- iii)* Le probabilità di transizione $P(\mathbf{i}, \mathbf{j}; t)$ soddisfano

$$\sum_{\mathbf{j} \in \mathcal{R}_n^+} P(\mathbf{i}, \mathbf{j}; t) \mathbf{s}^{\mathbf{j}} = \prod_{k=1}^n \left[\sum_{j \in \mathcal{R}_n^+} P(e_k, \mathbf{j}; t) \mathbf{s}^{\mathbf{j}} \right]^{i_k};$$

per ogni $\mathbf{i} \in \mathcal{R}_n^+$ e $\mathbf{s} \in \mathcal{C}_n = \{\mathbf{x} : 0 \leq x_i < 1\}$ (il cubo unitario in \mathbb{R}^n).

^a \mathcal{R}_n = l'insieme di tutti i punti di \mathbb{R}^n con coordinate intere

Dobbiamo fare attenzione che in generale le lettere in **grassetto** rappresentano vettori e non numeri scalari.

Siano $\mathbf{j} = (j_1, \dots, j_n)$ e $\mathbf{i} = (i_1, \dots, i_n)$ due vettori tale che $j_k, i_k \in \{0\} \cup \mathbb{Z}^+$ per ogni $k \in \{1, \dots, n\}$ e $\mathbf{Z}^{\mathbf{i}}(t) = (Z_1^{\mathbf{i}}(t), \dots, Z_n^{\mathbf{i}}(t))$ il numero di particelle di ogni tipo al tempo “ t ” dato che il processo è iniziato con “ \mathbf{i} ” particelle al tempo “0”. Se $P(\mathbf{i}, \mathbf{j}; t)$ denota la probabilità che il processo iniziando con “ \mathbf{i} ” particelle al tempo 0 abbia “ \mathbf{j} ” particelle al tempo “ t ”, allora la funzione generatrice è data da

$$F(\mathbf{i}, \mathbf{s}; t) = \mathbb{E}\{s_1^{Z_1^{\mathbf{i}}(t)} \dots s_n^{Z_n^{\mathbf{i}}(t)}\} = \sum_{\mathbf{j} \in (\{0\} \cup \mathbb{Z}^+)^n} P(\mathbf{i}, \mathbf{j}; t) s_1^{j_1} \dots s_n^{j_n}, \quad (1.4.2)$$

ove $\mathbf{s} = (s_1, \dots, s_n)$. Per comodità denoteremo $s_1^{Z_1^{\mathbf{i}}(t)} \dots s_n^{Z_n^{\mathbf{i}}(t)}$ con $\mathbf{s}^{\mathbf{Z}^{\mathbf{i}}(t)}$.

Grazie all'indipendenza dell'evoluzione di ognuna di queste particelle, ciascuna particella inizia un'altro processo di ramificazione multitypo. Pertanto, se $Z_j^{k,i}(\tau)$ è il numero di particelle di tipo j che sono generati per la k -esima particella del tipo i in un intervallo di tempo τ , allora il numero totale di particelle del tipo j che esistono al tempo $t + \tau$ sarà la somma delle particelle del tipo j generati per le $Z_i(t)$ particelle al tempo t , per ogni $i = 1, 2, \dots, n$. Quindi, il numero totale di

particelle del tipo j sono

$$Z_j(t + \tau) = \sum_{i=1}^n \sum_{k=1}^{Z_i(t)} Z_j^{k,i}(\tau). \quad (1.4.3)$$

In termini delle funzioni generatrici, l'ultima equazione può essere scritta come

$$\mathbf{F}(\mathbf{s}; t + \tau) = \mathbf{F}(\mathbf{F}(\mathbf{s}; \tau); t); \quad (1.4.4)$$

ove $\mathbf{F}(\mathbf{s}; t) = (F(\mathbf{e}_1, \mathbf{s}; t), \dots, F(\mathbf{e}_n, \mathbf{s}; t))$ e

$$F(\mathbf{e}_i, \mathbf{s}; t) = \mathbb{E}\{\mathbf{s}^{\mathbf{Z}^{\mathbf{e}_i}(t)}\}.$$

Analogamente ai casi precedenti è possibile derivare l'equazioni di Kolmogorov per il *ctMMTBP*, loro hanno la forma seguente:

$$\frac{\partial}{\partial t} F(\mathbf{e}_i, \mathbf{s}; t) = \sum_{k=1}^n u^{(k)}(\mathbf{s}) \frac{\partial}{\partial s_k} F(\mathbf{e}_i, \mathbf{s}; t), \quad (\text{forward equation}). \quad (1.4.5)$$

$$\frac{\partial}{\partial t} F(\mathbf{e}_k, \mathbf{s}; t) = u^{(k)}[\mathbf{F}(\mathbf{s}; t)], \quad (\text{backward equation}). \quad (1.4.6)$$

Dove

$$u^{(i)}(\mathbf{s}) = a_i [f^{(i)}(s_1, \dots, s_n) - s_i], \quad (1.4.7)$$

per ogni $i \in \{1, 2, \dots, n\}$ e ogni $k \in \{1, 2, \dots, n\}$.

In modo simile alla sezione precedente, possiamo descrivere il processo minimal come segue: Se una particella del tipo “ i ” è viva per un certo tempo, allora la sua durata di vita supplementare sarà una v.a distribuita esponenzialmente con parametro $a_i > 0$, dopo la sua morte questa particella lascerà discendenti di n tipi secondo una distribuzione di probabilità $p^{(i)}(\mathbf{j})$. Le particelle dopo esser nate evolvono indipendentemente l'una dall'altra e anche del passato.

Di seguito calcoleremo il valore atteso del processo (primo momento).

1.4.1 Calcolo del primo momento di $\mathbf{Z}(t)$

Dobbiamo prima garantire che il processo non sia esplosivo, cioè, che non si producono infinite particelle in un tempo finito. In [4] è data una condizione sufficiente ma non necessaria per garantire la non esplosione del processo.

Proposizione 1.2. *Sia $\{\mathbf{Z}(t) : t \geq 0\}$ un *ctMMTBP*, allora una condizione sufficiente ma non necessaria per garantire la non esplosione è:*

$$\left. \frac{\partial f^{(i)}(\mathbf{s})}{\partial s_j} \right|_{\mathbf{s}=\mathbf{e}} < \infty \quad \forall i, j. \quad (1.4.8)$$

□

Una condizione necessaria e sufficiente per garantire la non esplosione la possiamo trovare in [54]. Sotto questa ipotesi, è possibile anche garantire che l'equazioni *forward e backward di Kolmogorov*, con condizioni di contorno

$$P(\mathbf{i}, \mathbf{j}; 0^+) = \delta(\mathbf{i}, \mathbf{j}) = \begin{cases} 1 & \text{se } \mathbf{i} = \mathbf{j} \\ 0 & \text{se } \mathbf{i} \neq \mathbf{j} \end{cases}$$

abbia una unica soluzione, in questo caso la funzione generatrice di $P(\mathbf{i}, \mathbf{j}; t)$, con la condizione $\sum_{\mathbf{j}} P(\mathbf{i}, \mathbf{j}; t) = 1$, che corrisponde al processo minimal.

Usando (1.4.8) si può anche dimostrare che

$$m_{ij}(t) = \mathbb{E}\{Z_j(t) | Z(0) = \mathbf{e}_i\} < \infty. \quad (1.4.9)$$

Sia ora la matrice $\mathbf{M}(t) = \{m_{ij}(t) : i, j = 1, 2, \dots, n\}$ dei valori attesi dei tipi di particelle al tempo "t". Dell'equazione backward di Kolmogorov può essere derivato la proprietà di semigrupp per la matrice \mathbf{M}

$$\mathbf{M}(t + u) = \mathbf{M}(t)\mathbf{M}(u), \quad t, u \geq 0$$

e anche la condizione di continuità

$$\lim_{t \rightarrow 0} \mathbf{M}(t) = \mathbf{I}$$

ove \mathbf{M} e \mathbf{I} sono matrici di dimensione $n \times n$.

Questi due ultimi risultati implica l'esistenza di una matrice \mathbf{A} , chiamata il *generatore infinitesimale del semigrupp* $\{\mathbf{M}(t); t \geq 0\}$, tale che

$$\mathbf{M}(t) = \exp(\mathbf{A}t) = \sum_0^{\infty} \frac{t^r \mathbf{A}^r}{r!}, \quad (1.4.10)$$

Ciascun elemento della matrice \mathbf{A} , denotato con A_{ij} , può essere interpretato come il tasso medio alla quale una particella di tipo "i" dà origine a particelle di tipo "j". Quindi, possiamo scrivere $A_{ij} = a_i b_{ij}$, dove

$$b_{ij} = \left. \frac{\partial f^{(i)}(\mathbf{s})}{\partial s_j} \right|_{\mathbf{s}=\mathbf{e}} - \delta_{ij}. \quad (1.4.11)$$

Il processo sarà detto *positiva regolare* se esiste qualche $t = t_0 > 0$, tale che

$$m_{ij}(t) > 0 \quad \forall i, j.$$

Dalla teoria di matrici positive (vedere [55]) esiste un autovalore $\rho_1(t_0)$ strettamente positivo di $\mathbf{M}(t_0)$, chiamata *l'autovalore di Perron-Frobenius*, tale che

- 1.- Dato un'altro autovalore $\rho(t_0)$ di $\mathbf{M}(t_0)$, allora $|\rho(t_0)| < \rho_1(t_0)$.
- 2.- La molteplicità algebrica e geometrica di $\rho_1(t_0)$ sono entrambi uguali a uno.

Dal momento che $\mathbf{M}(t) = \exp(\mathbf{A}t)$ gli autovalori di $\mathbf{M}(t)$ hanno la forma $e^{\lambda_i t}$ per $i = 1, 2, \dots, n$, dove $\lambda_1, \dots, \lambda_n$ sono gli autovalori di \mathbf{A} , e $\mathbf{M}(t)$ e \mathbf{A} hanno gli stessi autovalori. Questo implica che è possibile ordinare gli autovalori di \mathbf{A} come

$$\lambda_1 > \text{Re}(\lambda_2) \geq \text{Re}(\lambda_3) \geq \dots \geq \text{Re}(\lambda_n),$$

e così determinare gli autovettori a sinistra e a destra \mathbf{v} e \mathbf{u} rispp. di λ_1 con tutte le sue coordinate strettamente positive, tale che

$$\mathbf{u} \cdot \mathbf{v} = 1, \quad \mathbf{u} \cdot \mathbf{e} = 1$$

Osservazione:

- Se $\lambda_1 > 0$, allora il processo è detto *super-critical*.
- Se $\lambda_1 = 0$, allora il processo è detto *critical*.
- Se $\lambda_1 < 0$, allora il processo è detto *sub-critical*.

1.4.2 La probabilità di estinzione

Le dimostrazioni della transitorietà degli stati finiti diversi da zero e la probabilità di estinzione sono relativamente semplici per il processo di ramificazione multitypo a tempo discreto (noto anche come *the Galton-Watson multi-type branching process*) ([19]). Con piccole modifiche, queste prove possono trasferirsi al *ctMMTBP*. Così, se il *ctMMTBP* è *positivo regolare* e *non singolare*, allora tutti gli stati con un numero finito di particelle sono transitori (vedere [4],[19]). Quindi, con probabilità *uno*, tutte le realizzazioni del processo o eventualmente si estingueranno o il numero totale di rami tenderà all'infinito.

Sia $q^{(i)}$ la probabilità che il processo iniziando con una particella di tipo “ i ” eventualmente si estingua. Sia $\mathbf{q} = (q^{(1)}, \dots, q^{(n)})$. Si può dimostrare usando l'equazione backward di Kolmogorov che \mathbf{q} è la soluzione minimal non negativa di

$$\mathbf{u}(\mathbf{s}) = (u^{(1)}(\mathbf{s}), \dots, u^{(n)}(\mathbf{s})) = \mathbf{0}, \tag{1.4.12}$$

ove $u^{(i)}(\mathbf{s})$ è data in (1.4.7). Questo è equivalente alla condizione

$$\mathbf{q} = \mathbf{f}(\mathbf{q}), \tag{1.4.13}$$

ove \mathbf{f} è data in (1.4.1) per il caso in tempo discreto. Infatti, si può dimostrare, come per il caso a tempo discreto (vedere Harris [19]), che la soluzione \mathbf{q} è o uguale a \mathbf{e} , o tutti i suoi componenti devono essere strettamente minori di *uno*. Argomentazioni simili possono essere utilizzate per il caso a tempo continuo. Di conseguenza, se $\lambda_1 > 0$ allora $\mathbf{q} < \mathbf{e}$ componente per componente, e se $\lambda_1 \leq 0$ allora $\mathbf{q} = \mathbf{e}$.

Osservazione:

Nel caso *supercritical* a tempo discreto, in [19] si dimostra che se il processo è positivo regolare e non singolare, allora

$$\lim_{n \rightarrow \infty} \mathbf{f}_n(\mathbf{q}_0) = \mathbf{q}, \tag{1.4.14}$$

ove \mathbf{q}_0 è qualsiasi vettore iniziale nel cubo unitario con una dimensione appropriata e $\mathbf{f}_n(\mathbf{s}) = \mathbf{f}(\mathbf{f}_{n-1}(\mathbf{s}))$. Ciò fornisce un algoritmo per calcolare la probabilità di una eventuale estinzione del processo. Più avanti deriveremo un algoritmo che utilizza un'equazione molto simile all'equazione (1.4.14) e quindi svilupperemo un'altro algoritmo che converga a \mathbf{q} in un modo molto più efficiente.

Capitolo 2

Matrix Analytic Methods: Una introduzione

La teoria di *Matrix-Analytic Methods* costituisce la base di gran parte del lavoro in questa tesi. Questi metodi sono un evento storico nella matematica della probabilità poiché loro illustrano l'arricchimento di una scienza: *“la Probabilità Applicata”* tramite l'uso della tecnologia, in questo caso l'informatica digitale. Marcel Neuts ha giocato un ruolo molto importante in questo eccitante sviluppo attraverso la promozione della indagine numerica come parte essenziale nella soluzione dei modelli probabilistici. Lui ha scritto nel 1973,

“Il lavoro in matematica computazionale èun impegno con una definizione molto più esigente di quello che costituisce la soluzione di un problema matematico. Quando è fatto correttamente, questo soddisfa i più elevati standard della ricerca scientifica.” (Vedere [45], pag. 20)

Questo capitolo presenta le idee matematiche e algoritmiche sviluppati nei libri [39] e [47]. I metodi sono presentati all'interno del semplice quadro dei *“Quasi-Birth-and-Death processes”* (QBDs). Questi sono processi di Markov in due dimensioni, il *level* e la *fase*, in modo che il processo non consenta i salti attraverso diversi livelli in una transizione. Limitarci ai QBDs, tuttavia, non è eccessivamente restrittivo, in quanto si mostrerà nel processo, che la teoria per le classe più generali dei modelli possono essere dedotti dall'analisi di un QBD.

Siccome il tema è molto ampio, questa verrà sviluppata in questo capitolo e il successivo. Questo primo capitolo è diviso in due sezioni: *i) From the Exponential Distribution to Phase-Type Distributions* e *ii) From the Poisson Process to Markovian Arrival Processes*. Nel seguente capitolo studieremo *iii) Quasi-Birth-and-Death Processes*. L'obiettivo di questo approccio è di mostrare in un certo senso che *“the phase-type Distributions”* e *“the Markovian arrival processes”* sono una generalizzazione della *“Exponential Distribution”* ed il *“Poisson Process”* rispettivamente. Le dimostrazioni dei teoremi, proposizioni, ecc., discussi in questo e il prossimo capitolo, possono essere trovati in [8], [11], [27], [39] e [47].

2.1 Dalla Exponential Distribution alla Phase-Type Distributions

Le distribuzioni di probabilità sono stati ampiamente studiati e usate estesamente nel campo della scienza e dell'ingegneria. Tra queste distribuzioni abbiamo, la normal, lognormal, uniforme, exponential, Erlang, Poisson, ecc. Queste distribuzioni di probabilità possiedono alcune proprietà

vantaggiose che appellano ad alcune applicazioni specifiche. Questa sezione si concentra sulle *Exponential and Phase-type distributions*, che godono di grande popolarità in modellazione stocastica dovuto soprattutto alla loro proprietà di “*assenza di memoria parziale*” (o *memoryless*).

2.1.1 La Exponential Distribution

Definizione 2.1. Una variabile aleatoria continua X si dice che ha una **Exponential Distribution** di parametro $\lambda > 0$, se la sua funzione di distribuzione di probabilità è data da:

$$F(t) = \mathbb{P}\{X \leq t\} = 1 - e^{-\lambda t} \quad \text{per ogni } t \geq 0$$

Nel contesto della definizione (2.1), chiameremo X una *variabile aleatoria esponenziale* di parametro λ . Derivando la funzione di distribuzione $F(t)$ rispetto a “ t ”, la sua funzione di densità risulta

$$f(x) = \frac{dF(t)}{dt} = \begin{cases} \lambda e^{-\lambda t} & ; t \geq 0 \\ 0 & ; t < 0 \end{cases}$$

Di questo’ultimo risultato possiamo trovare facilmente il valore atteso, la varianza e il coefficiente di variazione di X :

Proposizione 2.1. Sia X una variabile aleatoria esponenziale di parametro λ , allora abbiamo

- Il *valore atteso*

$$\mathbb{E}\{X\} = \frac{1}{\lambda};$$

- La *varianza*

$$\text{Var}(X) = \sigma^2(X) = \sigma_X^2 = \frac{1}{\lambda^2};$$

- Il *coefficiente di variazione*

$$cv(X) = \frac{\sqrt{\text{Var}(X)}}{\mathbb{E}\{X\}} = \sqrt{\text{Var}\left(\frac{X}{\mathbb{E}\{X\}}\right)} = 1.$$

The Memoryless property

Sia X una v.a. con distribuzione esponenziale, da (2.1) insieme alla definizione di probabilità condizionale, è facile dimostrare che

1. $\mathbb{P}\{X > t + s\} = \mathbb{P}\{X > t\}\mathbb{P}\{X > s\} \quad \forall t, s \geq 0$
2. $\mathbb{P}\{X > t + s | X > s\} = \mathbb{P}\{X > t\} \quad \forall t, s \geq 0$

Nota: La proprietà (2), che può essere dimostrata di essere equivalente a (1) è conosciuto come “*lack of memory*” (assenza di memoria) chiamata anche “*memoryless*”.

Teorema 2.1. *Le variabili aleatorie esponenziali sono l'uniche variabili aleatorie continue non negative e finite che possiedono la proprietà di assenza di memoria.*

Una dimostrazione di questo teorema possiamo trovarlo in [27] ed alcuni esercitazioni su questo argomento sono disponibili in [10] e [53]. Di seguito diamo alcuni risultati molto importanti che sono ampiamente utilizzati nelle applicazioni.

Proposizione 2.2. *Sia $\mathbf{X} = (X_1, \dots, X_n)$ una successione finita di variabili aleatorie ove X_i ha una distribuzione esponenziale di parametro $\lambda_i > 0$ per ogni $i \in \{1, \dots, n\}$.*

1. *Se $U = \min\{X_1, \dots, X_n\}$, allora $U \sim \text{Exp}(\sum_{i=1}^n \lambda_i)$.*
2. *Se $V = \max\{X_1, \dots, X_n\}$, allora V ha una funzione di distribuzione F data da:*

$$F(x) = \prod_{i=1}^n (1 - e^{-\lambda_i x}) \quad \forall x \geq 0$$

Abbiamo anche bisogno di estendere il risultato (1) di cui sopra per un numero infinito numerabile di variabili aleatorie.

Proposizione 2.3. *Sia $U = \inf\{X_i : i \in I\}$ una successione numerabile di variabili aleatorie ove X_i ha una distribuzione esponenziale di parametro $\lambda_i > 0$ per ogni $i \in I$, allora $U \sim \text{Exp}(\sum_{i \in I} \lambda_i)$.*

La proposizione (2.2) mostra che l'insieme delle variabili aleatorie esponenziali è chiusa sotto l'operazione "min", mentre che sotto l'operazione "max" non è chiusa.

Proposizione 2.4. *Supponiamo che le v.a indipendenti X_1 e X_2 hanno distribuzioni esponenziali con parametri λ_1 e λ_2 rispettivamente, allora*

$$\mathbb{P}\{X_1 < X_2\} = \frac{\lambda_1}{\lambda_1 + \lambda_2}$$

La proposizione (2.4) può essere generalizzata come segue

Proposizione 2.5. *Supponiamo che le v.a indipendenti $\{X_j : j = 1, 2, \dots, n\}$ hanno distribuzioni esponenziali con parametri $\{\lambda_j : j = 1, 2, \dots, n\}$ rispettivamente, allora*

$$\mathbb{P}\{X_1 = \min\{X_1, \dots, X_n\}\} = \frac{\lambda_1}{(\lambda_1 + \dots + \lambda_n)}$$

La *trasformata di Laplace-Stieltjes (LST)* è uno strumento utile nello studio e applicazioni delle distribuzioni di probabilità. Esiste una relazione uno-a-uno tra **LSTs** di funzioni di distribuzione di probabilità e le variabili aleatorie. Per le variabile aleatoria esponenziale X con parametro λ , abbiamo, per $s \geq 0$,

$$\begin{aligned}
f^*(s) &= \mathbb{E}\{e^{-sX}\} &= \int_0^\infty e^{-st} d\mathbb{P}\{X < t\} \\
&= \int_0^\infty e^{-st} \lambda e^{-\lambda t} dt &= \lambda \int_0^\infty e^{-(\lambda+s)t} dt \\
&= \frac{\lambda}{\lambda + s}.
\end{aligned}$$

Dunque ogni v.a. esponenziale X con parametro λ ha una **LST** data da

$$f^*(s) = \frac{\lambda}{\lambda + s} \quad (2.1.1)$$

La funzione $f^*(s)$ è chiamata la **LST** di X . Si può dimostrare che ogni v.a. con **LST** data per l'equazione (2.1.1) ha una distribuzione esponenziale con parametro λ . Così, la distribuzione esponenziale può essere utilizzata nella costruzione di distribuzioni di probabilità più complicati. Prima di dare l'ultima proposizione di questa sezione, dobbiamo ricordare com'è definito una *distribuzione Erlang*, il qual è un caso particolare della *distribuzione Gamma*.

Definizione 2.2. Una distribuzione Erlang con parametri (n, λ) è definita come una distribuzione di probabilità continua con funzione densità

$$f_{(n,\lambda)}(t) = \frac{t^{n-1} \lambda^n}{(n-1)!} e^{-\lambda t}, \quad t \geq 0, n = 1, 2, \dots$$

Vediamo ora l'ultima proposizione,

Proposizione 2.6. Sia $\mathbf{X} = (X_1, \dots, X_n)$ una successione finita di variabili aleatorie i.i.d. esponenzialmente di parametro $\lambda > 0$, allora la suma $S_n = X_1 + \dots + X_n$ ha una distribuzione Erlang di parametri (n, λ) , cioè $S_n \sim \text{Erlang}(n, \lambda)$. □

Quest'ultimo risultato può essere dimostrato per induzione o usando la **LST** di S_n e la definizione della distribuzione Erlang.

2.1.2 Le Phase-Type Distributions

Le (*Phase-Type Distributions*) sono state introdotte da Neuts [47] come una generalizzazione della *Exponential distribution*. In questa sezione, si da una definizione algebrica e una definizione probabilistica delle distribuzioni tipo-fase, e si presentano alcune delle loro proprietà di base.

Definizione 2.3. Una variabile aleatoria non negativa X ha una **phase-type Distribution** (*PH-distribution*) se la sua funzione di distribuzione è data da

$$F(t) = \mathbb{P}\{X \leq t\} = 1 - \boldsymbol{\tau} e^{\mathbf{B}t} \mathbf{e}, \quad t \geq 0, \quad (2.1.2)$$

ove:

- (i) \mathbf{e} è il vettore colonna di dimensione $1 \times m$ con tutti gli elementi uguali a uno,
- (ii) $\boldsymbol{\tau}$ è un vettore riga *substochastic* di ordine “ m ”, cioè, tutti i suoi elementi sono non negativi, e $\boldsymbol{\tau} \mathbf{e} \leq 1$; e

(iii) \mathbf{B} è un *subgenerator* di ordine “ m ”, i.e., \mathbf{B} è una matrice di dimensione $m \times m$ tale che:

- Tutti gli elementi diagonali sono negativi (< 0),
- Tutti gli elementi non diagonali sono non negativi (≥ 0),
- La suma di gli elementi di ogni riga è non positivo (≤ 0),
- \mathbf{B} è invertibile.

Chiameremo a \mathbf{B} un *PH-generator*. La 2-tuple $(\boldsymbol{\tau}, \mathbf{B})$ è detto una *phase-type representation* (*PH-representation*) di ordine “ m ” della *PH-distribution*.

Proposizione 2.7. *Sia $(\boldsymbol{\tau}, \mathbf{B})$ una PH-representation data nella definizione (2.3), la funzione $F(t)$ definita in (2.1.2) è una distribuzione di probabilità e la sua funzione densità è data da*

$$f(t) = \frac{dF(t)}{dt} = \boldsymbol{\tau} e^{(\mathbf{B}t)} \mathbf{b}, \quad t \geq 0. \quad (2.1.3)$$

Dove $\mathbf{b} = -\mathbf{B}\mathbf{e}$.

Quattro PH-distributions con PH-representations	
$\boldsymbol{\tau}_1 = (0.2, 0.8)$	$\mathbf{B}_1 = \begin{pmatrix} -2 & 1 \\ 0.5 & -10 \end{pmatrix}$
$\boldsymbol{\tau}_2 = (0.1, 0, 0.9)$	$\mathbf{B}_2 = \begin{pmatrix} -15 & 0 & 10 \\ 0 & -2 & 0 \\ 0 & 2 & -2 \end{pmatrix}$
$\boldsymbol{\tau}_3 = (0, 0.1, 0.8, 0.1)$	$\mathbf{B}_3 = \begin{pmatrix} -20 & 0 & 0 & 15 \\ 20 & -20 & 0 & 0 \\ 0 & 0.5 & -1 & 0.5 \\ 0 & 0 & 0 & -1 \end{pmatrix}$
$\boldsymbol{\tau}_4 = (0, 0.2, 0, 0, 0.8)$	$\mathbf{B}_4 = \begin{pmatrix} -15 & 0 & 0 & 0 & 0 \\ 15 & -15 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}$

Tabella 2.1

La figura (2.1) mostra tre funzioni di densità corrispondenti ai tre *PH-representations* fornite nella tabella (2.1). Come si può vedere, a differenza delle funzioni di densità esponenziali che hanno un comportamento simile al variare il suo parametro, il comportamento delle *PH-distributions* sono molto diversi al variare i suoi parametri. Infatti, variando i parametri delle *PH-representations* di ordini piccoli (ad esempio, $m = 2, 3, 4$ o 5), si può generare una vasta gamma di distribuzioni di probabilità con caratteristiche diverse. Quindi, l’insieme delle *PH-distributions* può essere grande e versatile, il seguente teorema conferma formalmente quello menzionato.

Teorema 2.2. *L’insieme delle PH-distributions è denso nell’insieme delle distribuzioni di probabilità definite sulla semiretta non negativa.*

□

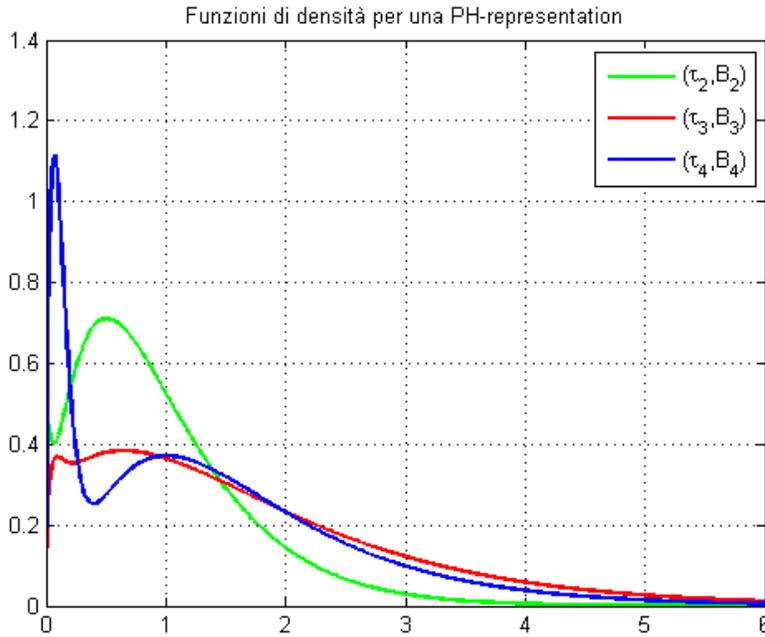


Figure 2.1: Tre funzioni di densità per una PH-representation

Il Teorema (2.2) implica che le *PH-distributions* possono approssimare ogni distribuzione di probabilità definita sulla semiretta non negativa, questo è uno dei motivi principali del perché le *PH-distributions* sono utilizzati così ampiamente nella modellazione stocastica.

L'interpretazione probabilistica associata con le *PH-distributions* presenta un altro importante motivo, daremo una definizione probabilistica delle PH-distributions che è stata la prima definizione riportata da Neuts (1975).

Per introdurre l'idea, sia $\{Y(t) : t \geq 0\}$ una catena di Markov a tempo continuo con $m + 1$ stati $\{0, 1, 2, \dots, m\}$ e generatore infinitesimale

$$Q = \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{b} & \mathbf{B} \end{pmatrix}, \quad (2.1.4)$$

dove \mathbf{B} è un *PH-generator* (vedere def.(2.3)). Poiché le somme delle righe di un generatore infinitesimale sono pari a zero, abbiamo $\mathbf{b} = -\mathbf{B}\mathbf{e}$. Si osserva inoltre nella matrice Q , che il tasso di transizione dello stato "0" è zero, quindi "0" è uno stato assorbente¹. Definiamo

$$X = \min\{t : Y(t) = 0, t \geq 0\}, \quad (2.1.5)$$

che è chiaramente il tempo di assorbimento dello stato "0". Allora, $\mathbb{P}\{X < t\}$ può essere interpretata come la probabilità che la catena di Markov è stato assorbito nello stato "0" prima del tempo " t ".

Data una distribuzione iniziale $\pi(0) = (1 - \boldsymbol{\tau}\mathbf{e}, \boldsymbol{\tau})$, dove abbiamo usato il vettore *substochastic* $\boldsymbol{\tau}$ di ordine " m " della def. (2.3), allora la distribuzione di probabilità della catena di Markov $Y(t)$ è

¹È possibile avere come stato assorbente a " m " cambiando soltanto la posizione della matrice B in Q , ma l'analisi generale non cambia.

data da $\pi(t) = \pi(0)\exp(Qt)$, che può essere calcolato come

$$\begin{aligned}
\pi(t) &= (1 - \tau \mathbf{e}, \tau) \exp \left(\begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{b} & \mathbf{B} \end{pmatrix} t \right) \\
&= (1 - \tau \mathbf{e}, \tau) \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{e} - \exp(\mathbf{B}t)\mathbf{e} & \exp(\mathbf{B}t) \end{pmatrix} \\
&= (1 - \tau \exp(\mathbf{B}t)\mathbf{e}, \tau \exp(\mathbf{B}t)).
\end{aligned} \tag{2.1.6}$$

Il primo termine nell'ultima riga in (2.1.6) dà la probabilità che la catena di Markov sarà nello stato “0” al tempo “ t ”, quindi, la probabilità $\mathbb{P}\{X \leq t\}$. Così, abbiamo ottenuto

$$\mathbb{P}\{X \leq t\} = \mathbb{P}\{Y(t) = 0\} = 1 - \tau \exp(\mathbf{B}t)\mathbf{e}, \quad t \geq 0. \tag{2.1.7}$$

Quest'ultimo risultato porta alla seguente definizione di PH-distribuzioni.

Definizione 2.4. *Supponiamo che una catena di Markov a tempo continuo $\{Y(t) : t \geq 0\}$ con generatore infinitesimale data in (2.1.4) e con $m + 1$ stati $\{0, 1, 2, \dots, m\}$, sarà assorbita nello stato “0” con probabilità 1. Una **phase-type Random Variable** X , è definita come il tempo di assorbimento dello stato “0” della catena di Markov $Y(t)$ con distribuzione iniziale $\pi(0) = (1 - \tau \mathbf{e}, \tau)$. La 2-tupla (τ, \mathbf{B}) è una PH-representation di X .*

Osservazione:

1. Le definizioni (2.3) e (2.4) sono chiaramente equivalenti, ma la def. (2.4) è utile in modellazione stocastica poiché collega una variabile aleatoria tipo-fase ad una catena di Markov a tempo continuo, che di solito è chiamato la *underlying Markov chain*.
2. Prima dell'assorbimento, la *underlying Markov chain* è controllata dai tassi di transizione della matrice \mathbf{B} .
3. I tassi di assorbimento degli stati transitori allo stato di assorbimento sono riportati in \mathbf{b} . Di conseguenza, seguendo la traccia dello stato $Y(t)$, possiamo trovare la distribuzione del tempo residual di assorbimento dato che il tempo residual ha una PH-distribuzione. Ad esempio, se $Y(t) = 2$, allora il tempo residual ha una PH-distribuzione con una PH-representation $((0, 1, 0, \dots, 0), \mathbf{B})$. Questa proprietà delle PH-representations è chiamata la proprietà “*partial memoryless*” basato sullo stato della *underlying Markov chain*.
4. Tradizionalmente, uno stato della *underlying Markov chain* di una PH-distribution è chiamata “**fase**”. Quindi, d'ora in avanti useremo “*fase*” e “*stato*” in modo intercambiabile.

Di seguito diamo una proposizione che permette calcolare i momenti di una qualsiasi variabile aleatoria X con una PH-representation (τ, \mathbf{B}) .

Proposizione 2.8. *Data una PH-random variable X con PH-representation (τ, \mathbf{B}) , allora*

(i) *La LST di X è: $f^*(s) = 1 - \tau \mathbf{e} + \tau(s\mathbf{I} - \mathbf{B})^{-1}\mathbf{b}$, per ogni $s \geq 0$, dove \mathbf{I} è la matrice identità.*

(ii) $\mathbb{E}\{X^n\} = (-1)^n n! \boldsymbol{\tau} \mathbf{B}^{-n} \mathbf{e}$, per ogni $n = 1, 2, \dots$

□

Usando quest'ultima proposizione possiamo calcolare il valore atteso e la varianza del tempo di assorbimento di X , questo sarà riportato in una tabella al finale di questa sezione.

Osservazione:

Dobbiamo sottolineare che una *PH-representation* $(\boldsymbol{\tau}, \mathbf{B})$ per una *PH-distribution* non è unica, infatti, nella tabella (2.2) ci sono tre *PH-representations* per una distribuzione esponenziale con parametro $\lambda = 1$. In generale si può dimostrare che per una *PH-distribution* esistono infinite rappresentazioni, dato che una *PH-distribution* di ordine m è determinato per al massimo $2m-1$ parametri indipendenti però la rappresentazione matriciale $(\boldsymbol{\tau}, \mathbf{B})$ ha $m^2 + m - 1$ parametri, quindi questa rappresentazione è altamente ridondante. Inoltre, esistono *PH-representations* $(\tilde{\boldsymbol{\tau}}, \tilde{\mathbf{B}})$ di ordine $n \neq m$ che descrivono la stessa distribuzione di $(\boldsymbol{\tau}, \mathbf{B})$ (vedere [11] e [27]).

3 PH-rappresentazioni per una dist. esponenziale di par. $\lambda = 1$	
$\boldsymbol{\tau}_1 = (2, 0)$	$\mathbf{B}_1 = \begin{pmatrix} -1 & 0 \\ 1 & -5 \end{pmatrix}$
$\boldsymbol{\tau}_2 = (0.2, 0.8)$	$\mathbf{B}_2 = \begin{pmatrix} -2 & 1 \\ 2 & -3 \end{pmatrix}$
$\boldsymbol{\tau}_3 = (0.5, 0.5)$	$\mathbf{B}_3 = \begin{pmatrix} -1 & 1 \\ 0 & -2 \end{pmatrix}$

Tabella 2.2

Nell'osservazione precedente abbiamo implicitamente affermato che una distribuzione esponenziale è una *PH-distribution*, che segue direttamente dalla definizione sopra come un caso particolare,... ma, tutta distribuzione di probabilità sarà una *PH-distributions*?

Il seguente teorema risponde a questa domanda.

Teorema 2.3. *Una distribuzione di probabilità sulla semiretta non negativo è una PH-distribution, se e solo se:*

- (a) *ha una densità continua e strettamente positivo sui numeri reali positivi, e*
- (b) *ha una LST razionale con un unico polo di parte reale maximal.*

□

Ora che sappiamo quando una distribuzione di probabilità è una *PH-distribution*, segue trovare una *PH-representation*, bisogna prima dare la seguente definizione.

Definizione 2.5. *Sia F una funzione di distribuzione tale che*

$$F(\infty) := \lim_{t \rightarrow \infty} F(t) = 1,$$

*allora si dice che F è **propria** o non difettoso, altrimenti si chiama **difettosa**[43].*

Sulla base di questa definizione diciamo che la *PH-distribution* $(\boldsymbol{\tau}, \mathbf{B})$ è non difettosa “*non defective*” se e soltanto se l’assorbimento avviene in un tempo finito *quasi certamente* (q.c.; cioè, con probabilità 1). Per quanto riguarda questa definizione abbiamo il seguente teorema.

Teorema 2.4. *Per ogni PH-distribution non defective, esiste una rappresentazione $(\boldsymbol{\tau}, \mathbf{B})$ che soddisfa la seguente condizione: partendo da qualsiasi fase $\mathbf{i} \in \{1, \dots, m\}$, l’assorbimento nello stato $\mathbf{0}$ si verifica con probabilità 1.*

□

Si potrebbe verificare direttamente dal diagramma di transizione se una *PH-distribution* è non difettosa, sarebbe necessario e sufficiente che per ogni fase i tale che fase $\tau_i > 0$, tutte le possibili traiettorie conducono alla fase assorbente 0. Il seguente teorema da una semplice condizione algebrica per cui l’assorbimento nella fase 0 si verifica q.c in un tempo finito.

Teorema 2.5. *Sia una PH-distribution $(\boldsymbol{\tau}, \mathbf{B})$. L’assorbimento nello stato 0 si verifica con probabilità 1 partendo di una fase \mathbf{i} in $\{1, \dots, m\}$ se e soltanto se la matrice \mathbf{B} è non singolare. Inoltre, $(-\mathbf{B}^{-1})_{ij}$ è il valore atteso del tempo totale trascorso nella fase \mathbf{j} durante il tempo fino all’assorbimento, dato che la fase iniziale è \mathbf{i} .*

□

Osservazione:

Una conseguenza diretta del teorema sopra è che $\mathbf{B}^{-1} \geq 0$

Per finire resta dire che le *PH-distributions* sono chiusi sotto un numero di operazioni, ad esempio, “min”, “max”, “+”. Riportiamo di seguito una tabella di confronto tra la *Exponential distribution* e la *PH-distribution*.

	Esponenziale	Tipo-Fase
Parametri	λ	$(m, \boldsymbol{\tau}, \mathbf{B})$
CDF	$1 - e^{-\lambda t}$	$1 - \boldsymbol{\tau} e^{\mathbf{B}t} \mathbf{e}$
PDF	$\lambda e^{-\lambda t}$	$-\boldsymbol{\tau} e^{\mathbf{B}t} \mathbf{T} \mathbf{e}$
Valore atteso	$1/\lambda$	$-\boldsymbol{\tau} \mathbf{B}^{-1} \mathbf{e}$
Varianza	$1/\lambda^2$	$2\boldsymbol{\tau} \mathbf{B}^{-2} \mathbf{e} - (\boldsymbol{\tau} \mathbf{B}^{-1} \mathbf{e})^2$
cv	1	$\sqrt{2\boldsymbol{\tau} \mathbf{B}^{-2} \mathbf{e} / (\boldsymbol{\tau} \mathbf{B}^{-1} \mathbf{e})^2} - 1$
Proprietà	<i>Assenza di memoria</i>	(1) <i>Assenza di memoria parziale</i> (2) <i>Approssima qualsiasi distribuzione su $[0, \infty)$</i>
Proprietà di Chiusura	(1) <i>min</i> (2) <i>suma geometrica</i>	(1) <i>min</i> (2) <i>max</i> (3) <i>sum</i> (4) <i>suma geometrica</i>

2.2 Dal processo di Poisson alle Markovian Arrival Processes

Il *Poisson processes* è uno dei processi aleatori più importanti nella teoria della probabilità giacché è uno dei processi di conteggio più semplice e possiede la proprietà di assenza di memoria che spesso porta a modelli Markoviani per sistemi stocastici, ha una bella struttura matematica

e viene utilizzato come base per la costruzione di una serie di altri processi aleatori più complicati. I *Markovian arrival processes* (MAPs) sono una generalizzazione dei processi di Poisson che mantiene molte delle loro proprietà, ad esempio, la proprietà di assenza di memoria del processo di Poisson è parzialmente preservata per il MAP quando sono condizionate sulle fasi di una *underlying Markov chain*. L'insieme dei *Markovian arrival processes* è versatile, cioè, qualsiasi "counting process" può essere approssimata arbitrariamente da una sequenza di MAPs.

Ci sono tre metodi di come affrontare la definizione del *Poisson process*: la costruttiva, l'infinitesimale e l'assiomatica. Si può dimostrare che queste tre definizioni sono equivalenti, quindi qui studieremo il metodo costruttivo, visto che questo metodo è adatto alle esigenze della tesi.

2.2.1 Costruzione del processo di Poisson

Consideriamo un processo in cui i punti si verificano in modo casuale nel tempo. La frase *punti nel tempo* è generico e potrebbe rappresentare, per esempio:

- I tempi in cui un campione di materiale radioattivo emette particelle.
- I tempi in cui i clienti arrivano in una stazione di servizio.
- I tempi in cui le richieste di file arrivano a un computer server, ecc.

Ci sono tre collezioni di variabili aleatori fortemente interconnessi che possono essere utilizzati per descrivere il processo, la successione di *inter-arrival times*, la successione di *arrival times* e il *counting process*. Di seguito definiremo e studieremo le proprietà di base di ognuno di questi processi.

Interarrival Times Process

Per avere un'idea grafica, supponiamo che lo stesso evento si verifica ripetutamente in modo casuale nel tempo come mostra la Figura 2.2. Sia X_1 il tempo del primo arrivo, e X_i il tempo tra il $(i - 1)$ -esimo e il i -esimo arrivo per $i \in \{2, 3, \dots\}$. La nostra ipotesi di base è che la successione di *interarrival times* $\mathbf{X} = (X_1, X_2, \dots)$ sia una successione di variabili aleatori indipendente e identicamente distribuite (i.i.d.). Supponiamo che X_i assume valori in $[0, \infty)$ e $\mathbb{P}\{X_i > 0\} > 0$, questo garantisce che *the interarrival times* siano non negativi e non identicamente 0. Allora, se $\mu = \mathbb{E}\{X_i\}$ denota la media comune dei *interarrival times*, abbiamo i seguenti risultati:

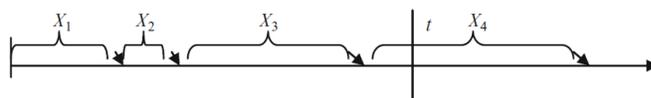


Figure 2.2: Linea temporale del processo.

1. $\mu > 0$ (con la possibilità di $\mu = \infty$).
2. Se $\mu < \infty$ abbiamo che $\sigma^2 = \text{Var}(X_i)$ denota la varianza comune dei *interarrival times*.
3. La funzione di distribuzione comune F in generale può essere qualsiasi ma per la costruzione del processo di Poisson, *the interarrival times* hanno la distribuzione esponenziale con parametro $\lambda > 0$, quindi $\mu = \frac{1}{\lambda}$. Il caso generale se usa nella costruzione della teoria del *Renewal process*.

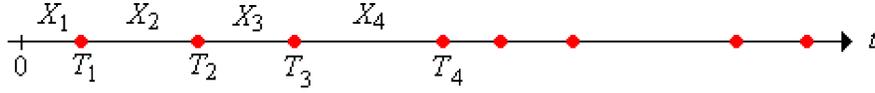


Figure 2.3: The interarrival times and arrival times

Arrival Times Process

Sia

$$T_n = \sum_{i=1}^n X_i, \quad n \in \mathbb{N}$$

Se seguiamo la solita convenzione che la somma su l'insieme di indici vuoto è 0 ($\sum_{i \in \emptyset} X_i = 0$ quindi $T_0 = 0$), la successione $\mathcal{T} = (T_0, T_1, \dots)$ è chiamato il *Arrival Times Process* (anche se T_0 non sia considerato un arrivo).

La successione \mathcal{T} è il processo di *somma parziale* associato alla successione \mathcal{X} di tempi di interarrivo i.i.d. La Figura 2.3 mostra la relazione tra questi due processi, quindi siamo in grado di recuperare i tempi di interarrivo (*interarrival times*) dei tempi di arrivo (*arrival times*) della seguente equazione:

$$X_i = T_i - T_{i-1}, \quad i \in \mathbb{N}_+$$

Raccoglieremo alcuni dei risultati più importanti su questo processo, ma prima diamo due definizioni su processi stocastici.

Definizione 2.6 (Independent increments). *Si dice che un processo stocastico ξ_t ove $t \in T$ ha incrementi indipendenti se*

$$\xi_{t_1} - \xi_{t_0}, \dots, \xi_{t_n} - \xi_{t_{n-1}}$$

sono indipendenti per ogni $t_0 < t_1 < \dots < t_{n-1} < t_n$ tale che $t_0, t_1, \dots, t_n \in T$.

Definizione 2.7 (Stationary increments). *Si dice che un processo stocastico ξ_t ove $t \in T$ ha incrementi stazionari se per ogni $s, t \in T$ la distribuzione di probabilità di $\xi_{t+h} - \xi_{s+h}$ è lo steso per ogni h tale che $s+h, t+h \in T$*

Se F_n denota la funzione di distribuzione di T_n , cioè

$$F_n(t) = \mathbb{P}\{T_n \leq t\}, \quad t \in [0, \infty)$$

allora, abbiamo i seguenti risultati:

Proposizione 2.9. *La successione \mathcal{T} ha incrementi stazionari e indipendenti, cioè*

a. Se $m \leq n$ allora $T_n - T_m$ ha la stessa distribuzione di T_{n-m} e ha funzione di distribuzione F_{n-m} .

b. Se $n_1 \leq n_2 \leq \dots$ allora $(T_{n_1}, T_{n_2} - T_{n_1}, T_{n_3} - T_{n_2}, \dots)$ è una successione di v.a indipendenti.

□

Proposizione 2.10. *Se $n, m \in \mathbb{N}$, allora*

a. $\mathbb{E}\{T_n\} = n\mu$.

b. $Var(T_n) = n\sigma^2$

c. $Cov(T_n, T_m) = \min\{m, n\}\sigma^2$

Ove μ e σ^2 sono la media e la varianza comune del processo dei tempi di interarrivo X_i .

□

Proposizione 2.11. Ricordando la legge dei grandi numeri abbiamo che:

$$\frac{T_n}{n} \rightarrow \mu \quad \text{quando } n \rightarrow \infty$$

a. Con probabilità 1 (la legge forte)

b. In probabilità (la legge debole)

□

Per finire, dato che $T_n \leq T_{n+1}$ per $n \in \mathbb{N}$ e che $\mathbb{P}\{T_n = T_{n-1}\} = \mathbb{P}\{X_n = 0\} = F(0)$, abbiamo la possibilità che allo stesso tempo più di un arrivo può verificarsi. D'altra parte, i tempi di arrivo sono illimitati:

Proposizione 2.12. $T_n \rightarrow \infty$ quando $n \rightarrow \infty$ con probabilità 1.

□

Counting Process

Per ogni $t \geq 0$, il processo N_t denota il numero di arrivi sull'intervallo $[0, t]$:

$$N_t = \sum_{n=1}^{\infty} \mathbf{1}_{\{T_n \leq t\}}, \quad t \in [0, \infty)$$

Si farà riferimento al processo aleatorio $\mathcal{N} = (N_t : t \geq 0)$ come il processo di conteggio (*Counting Process*). Ricordiamo che T_0 non è proprio un arrivo, ma è possibile avere $T_n = 0$ per $n \in \mathbb{N}_+$, potrebbe darsi più di un arrivo nel tempo 0. Abbiamo i seguenti risultati:

1. $N_t = \max\{n \in \mathbb{N} : T_n \leq t\}$ per $t \geq 0$
2. $T_n = \min\{t \geq 0 : N_t = n\}$ per $n \in \mathbb{N}$
3. Se $s, t \in [0, \infty)$ e $s \leq t$, allora $N_t - N_s$ è il numero di arrivi in $(s, t]$.

La Figura 2.4 mostra il processo di conteggio, note che come funzione di t , N_t è una funzione (aleatoria) a gradini con salti nei distinti valori di (T_1, T_2, \dots) , la dimensione del salto in un tempo di arrivo è il numero di arrivi in quel momento. In particolare, \mathcal{N} è una funzione crescente di t .

Proposizione 2.13. Per ogni $t \geq 0$ e $n \in \mathbb{N}$,

a. $T_n \leq t$ se e solo se $N_t \geq n$.

b. $N_t = n$ se e solo se $T_n \leq t \leq T_{n+1}$

□

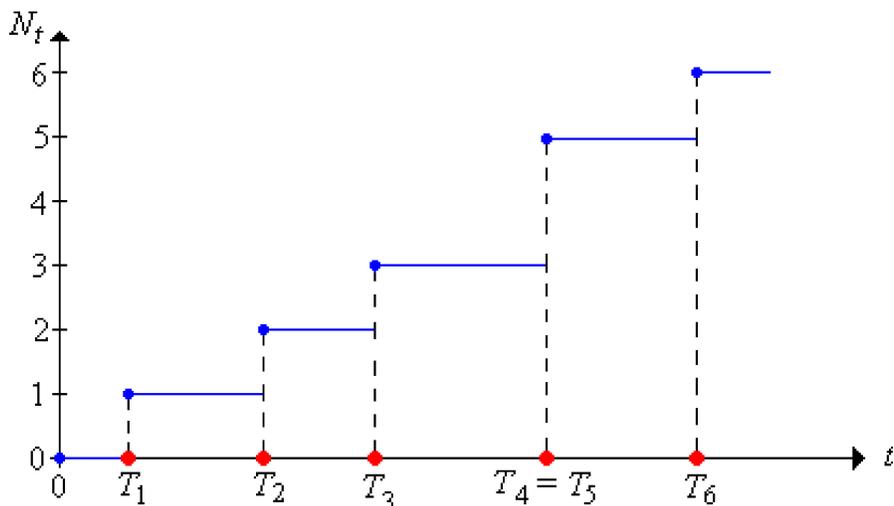


Figure 2.4: The counting process

Proposizione 2.14. *Con probabilità 1,*

- a. $N_t < \infty$ per ogni $t \in [0, \infty)$.
- b. $N_t \rightarrow \infty$ quando $t \rightarrow \infty$.

Tutti i risultati visti finora mostrano che il processo dei tempi di arrivo \mathcal{T} e il processo di conteggio \mathcal{N} sono inverse l'una dall'altra. Le importanti equivalenze di cui sopra possono essere usati per ottenere la distribuzione di probabilità delle variabili di conteggio in termini della funzione di distribuzione dei tempi di arrivo F .

Proposizione 2.15. *Per ogni $t \geq 0$ e $n \in \mathbb{N}$, abbiamo*

- a. $\mathbb{P}\{N_t \geq n\} = F_n(t)$.
- b. $\mathbb{P}\{N_t = n\} = F_n(t) - F_{n+1}(t)$.

Ove F_n è la funzione di distribuzione di T_n .

□

Un'altro risultato molto importante è il seguente,

Proposizione 2.16. *Con probabilità 1,*

$$\frac{N_t}{t} \rightarrow \frac{1}{\mu} \quad \text{quando } t \rightarrow \infty$$

Osservazione:

1. μ è la media comune dei *interarrival times*.

2. Le proposizioni precedenti sono vere anche quando μ è infinito. In questo caso interpretiamo $\frac{1}{\mu}$ come 0.
3. Il numero $\frac{1}{\mu}$ si chiama *il tasso del processo di rinnovamento*.

Ora siamo pronto a definire il processo di Poisson.

Definizione 2.8 (Processo di Poisson). *Un processo di Poisson di parametro $\lambda > 0$ è un **counting process** $\{N_t : t \geq 0\}$ a tempo continuo dove the interarrival times X_i hanno distribuzione esponenziale con lo stesso parametro λ .*

Siamo in grado ora di trovare la distribuzione di probabilità di N_t della relazione inversa tra \mathcal{N} e \mathcal{T} . In particolare, ricordare che

$$N_t \geq n \Leftrightarrow T_n \leq t, \quad t \in (0, \infty), \quad n \in \mathbb{N}_+ \quad (2.2.1)$$

poiché per entrambi eventi significa che ci sono almeno n arrivi in $(0, t]$.

D'altra parte, come

$$\mathbb{P}\{N_t = n\} = \mathbb{P}\{N_t \geq n\} - \mathbb{P}\{N_t \geq n + 1\}$$

e usando la relazione 2.2.1, risulta

$$\mathbb{P}\{N_t = n\} = \mathbb{P}\{T_n \leq t\} - \mathbb{P}\{T_{n+1} \leq t\}$$

quindi, sarebbe sufficiente conoscere la distribuzione di probabilità di T_n .

Siccome i processi di interarrivo X_i hanno distribuzione esponenziale di parametro λ , risulta che per la proposizione 2.6 T_n ha una funzione di densità tipo *Erlang* (caso particolare della distribuzione Gamma), cioè

$$f_{T_n}(t) = \lambda^n \frac{(\lambda t)^{n-1}}{(n-1)!} e^{-\lambda t}, \quad t \geq 0$$

La funzione di distribuzione e la funzione quantile della distribuzione Gamma non hanno espressioni semplici. Tuttavia, è possibile scrivere la funzione di distribuzione come somma

$$F_{T_n}(t) = 1 - \sum_{k=0}^{n-1} e^{-\lambda t} \frac{(\lambda t)^k}{k!}, \quad t \geq 0$$

quindi, possiamo facilmente dimostrare la seguente proposizione,

Proposizione 2.17. *Sia $\{N_t : t \geq 0\}$ un processo di Poisson di parametro $\lambda > 0$, allora $N_t \sim \text{Poisson}(\lambda t)$, con $t > 0$, cioè*

$$\mathbb{P}\{N_t = n\} = e^{-\lambda t} \frac{(\lambda t)^n}{n!}.$$

In particolare, $\mathbb{E}\{N_t\} = \lambda t$.

□

Osservazione:

1. La notazione $Poisson(\lambda t)$ vuole dire *distribuzione tipo Poisson* di parametro λt , il caso particolare noto come **Distribuzione di Poisson** è quando $t = 1$.
2. Il parametro λ ha un significato interessante, questo si capisce come il numero medio di eventi che si verificano per unità di tempo.

La proprietà di *assenza di memoria* della distribuzione esponenziale ha tre interessanti conseguenze per il processo di Poisson:

Proposizione 2.18. *Sia $\{N_t : t \geq 0\}$ un processo di Poisson di parametro $\lambda > 0$, e sia $s > 0$ un numero fisso, allora il processo $\{N_t - N_s : t \geq s\}$ è un processo di Poisson anche di parametro $\lambda > 0$. In particolare*

$$N_t - N_s \sim N_{t-s} \sim Poisson(\lambda(t-s))$$

□

Proposizione 2.19. *Il processo di Poisson N_t ha incrementi stazionari e indipendenti, cioè:*

1. *Per qualsiasi $s, t > 0$, la v.a. $N_{t+s} - N_s$ non dipende di s .*
2. *Per qualsiasi $0 \leq t_1 < t_2 < \dots < t_n$, le variabili aleatorie $N_{t_1}, N_{t_2} - N_{t_1}, \dots, N_{t_n} - N_{t_{n-1}}$ sono indipendenti.*

□

Proposizione 2.20. *Siano N_t^1 e N_t^2 due processi di Poisson indipendenti con parametri $\lambda_1 > 0$ e $\lambda_2 > 0$ risp. Allora, il processo $\{N_t^1 + N_t^2 : t \geq 0\}$ è un processo di Poisson con parametro $\lambda_1 + \lambda_2$.*

□

Un altro risultato importante del processo di Poisson è la sua relazione con le catene di Markov, ci sono due risultati importanti:

Proposizione 2.21. *Il processo di Poisson N_t è una catena di Markov a tempo continuo con lo spazio di stati $S = \{0, 1, 2, \dots\}$.*

□

Proposizione 2.22. *Le probabilità di transizioni sono stazionari e sono date da:*

$$p_{ij}(t) = \mathbb{P}\{N_{t+s} = j | N_s = i\} = \begin{cases} e^{-\lambda t} \frac{(\lambda t)^{j-i}}{(j-i)!} & ; \text{se } i \leq j \\ 0 & ; \text{altrimenti.} \end{cases}$$

□

Per approfondire su questo tema e vedere alcune simulazioni del processo di Poisson, visitare [29].

2.2.2 Markovian Arrival Processes (MAPs)

The PH-Renewal Process

Abbiamo visto che il processo di Poisson è un “*Renewal process*” con intervalli di rinnovo distribuiti esponenzialmente e inoltre, nell’ultima sezione abbiamo introdotto una potente generalizzazione della distribuzione esponenziale, allora, come primo passo per andare oltre il processo di Poisson, sembra naturale sostituire la distribuzione esponenziale degli intervalli di rinnovo da una *PH-distribution*. Questo ci porta immediatamente alla classe di “*PH Renewal Processes*”.

Mentre la determinazione concreta della funzione di rinnovo di un processo di rinnovamento arbitrario era difficile di ricavare, sappiamo che nel caso dove l’intervalli di rinnovo hanno una distribuzione esponenziale, i.e. per il processo di Poisson, si possono derivare espressioni semplici. Poiché una distribuzione esponenziale è una distribuzione di tipo-fase di ordine $m = 1$, un *PH-Renewal process* generalizza chiaramente il processo di Poisson. Si vedrà più avanti che il comportamento dei *PH-Renewal processes* può anche essere descritto da espressioni semplici. A tal fine, come nel caso delle *PH-distributions*, possiamo associare un processo di Markov con un *renewal process*.

Dato un processo di Markov con spazi di stati $\{0, 1, \dots, m\}$, distribuzione iniziale $(0, \boldsymbol{\tau})$ e generatore infinitesimale

$$\tilde{Q} = \begin{pmatrix} 0 & \mathbf{0} \\ \boldsymbol{\beta} & \mathbf{B} \end{pmatrix}$$

ove \tilde{Q} è la *underlying Markov chain* associato con la *PH-distribution* $F(\cdot)$. Il processo inizia al tempo 0 e dopo un periodo di tempo X (v.a) questa viene assorbita nello stato *zero*. Immediatamente dopo l’assorbimento, il processo di Markov si riavvia scegliendo un nuovo stato con una distribuzione di probabilità $\boldsymbol{\tau}$ e si evolve di nuovo fino un’altro assorbimento, e così via.

Sia l’insieme $\{0 = t_1 < t_2 < \dots\}$ dei punti nel tempo dove il processo di Markov è stata reinizializzata con distribuzione $\boldsymbol{\tau}$. È chiaro che il processo appena descritto è un *renewal process* con una *PH-distribution*.

Per questa *PH-distribution*, $PH(\boldsymbol{\tau}, \mathbf{B})$, il tempo restante X fino all’assorbimento, dato che la fase attuale è “*i*”, può essere espressa

$$\mathbb{P}\{X \leq t | Y_0 = i\} = 1 - \mathbf{e}_i e^{\mathbf{B}t} \mathbf{e},$$

ove $Y(t)$ ² è la *underlying Markov chain* associato alla *PH-distribution*. Si vede che X ha la stessa forma di una *PH-distribution* con *PH-representation* $(\mathbf{e}_i, \mathbf{B})$, dove \mathbf{e}_i denota il i -esimo vettore canonico. Questa è la proprietà condizionale di assenza di memoria (“*conditional memoryless property*”) della *PH-distribution*.

Se si segue la traccia della fase attuale della *PH-distribution* durante gli intervalli di rinnovo, si può ottenere una descrizione Markoviana $\Psi(t) = (N(t), \phi(t))$ del *PH-Renewal process* $N(t)$, dove $\{\phi(t) : t \geq 0\}$ è una catena di Markov a tempo continuo con spazio di stati $\{1, 2, \dots, m\}$ e $N(t)$ indica il numero di arrivi fino al tempo t . Notare che questa nuova catena di Markov $\phi(t)$ risulta di rimuovere lo stato assorbente 0 degli spazi di stati $\{1, \dots, m\}$ e quindi di ignorare le visite istantanei allo stato 0, questo è il “*processo fase*” associato con il *PH-Renewal process*. Il suo generatore infinitesimale è la matrice

²Notare che qui stiamo usando in modo intercambiabile il processo $Y(t)$ con la sua matrice generatrice \tilde{Q} .

$$D = \mathbf{B} + \boldsymbol{\beta}\boldsymbol{\tau}$$

È facile verificare che D soddisfa tutte le condizioni per un generatore infinitesimale, tuttavia, è più interessante capire il significato di ciascun addendo che compongono D .

Il processo $\Psi(t) = (N(t), \phi(t))$ è in realtà un processo di Markov con spazio di stati $E = \mathbb{N}_0 \times \{1, 2, \dots, m\}$. I tempi di permanenza in ogni stato, dipende solo della fase attuale della *PH-distribution*, i.e, $\lambda_{n,i} = \lambda_i \forall n \in \mathbb{N}_0$. Poiché $N(t)$ è *renewal process*, le transizioni degli stati della *embedded Markov chain* di $Y(t)$ sono limitati dalle $p_{n,i;m,j} = 0$ per $m < n$ o $m > n + 1$, quindi solo possono avere transizioni dello stato

$$(m, i) \quad a \quad (m + 1, j) \quad o,$$

$$(m, i) \quad a \quad (m, j).$$

che sono chiamati transizioni con o senza arrivi, rispettivamente.

- Per **transizioni senza arrivi**, non esiste evento di rinnovo, quindi nessun assorbimento per la *PH-distribution*, questo significa che le transizioni sono descritti dalla matrice \mathbf{B} della *PH-distribution* negli intervalli di rinnovo.

- Per **transizioni con arrivi**, si osserva lo seguente:

Se il processo si trova nella fase “ i ”, e si verifica un assorbimento (quindi un evento di rinnovo) con un tasso $\boldsymbol{\beta}_i$, dopo di che un nuovo intervallo di rinnovo PH-distribuito inizia, una fase viene scelta in base alla distribuzione di fase iniziale $\boldsymbol{\tau}$. Quindi le transizioni con arrivi sono determinati dalla matrice $A := \boldsymbol{\beta} \cdot \boldsymbol{\tau}$ con $\boldsymbol{\beta} = -\mathbf{B}\mathbf{e}$.

Dopo aver ordinato lo spazio di stati $E = \mathbb{N}_0 \times \{1, 2, \dots, m\}$ lessicograficamente, possiamo scrivere la matrice generatrice infinitesimale G del processo di Markov $\Psi(t) = (N(t), \phi(t))$ come una matrice a blocchi

$$G = \begin{pmatrix} B & A & & & \\ & B & A & & \\ & & B & A & \\ & & & \ddots & \ddots \\ & & & & \ddots & \ddots \end{pmatrix} \quad (2.2.2)$$

Vediamo di seguito alcune proprietà del *PH renewal process*.

Lemma 2.1. *Se la PH-representation $(\boldsymbol{\tau}, \mathbf{B})$ è irreducibile, allora la matrice $D = \mathbf{B} + \boldsymbol{\beta} \cdot \boldsymbol{\tau}$ è anche irreducibile.*

□

Il lemma appena descritto serve per garantire che tutte le fasi siano utili, cioè, per ogni $j \in \{1, 2, \dots, m\}$ esiste una traiettoria di alcuna fase i verso j con $\tau_i > 0$, quindi è garantita la visita alla fase j .

Con rispetto ai processi di rinnovamento esistono due quantità che risultano di notevole interesse, questi sono:

- La **funzione di rinnovo** $R(t)$, che rappresenta il numero atteso di rinnovi nell'intervallo $[0, t]$,

- La **densità di rinnovo** $r(t) = R'(t)$, dove $r(t)dt$ rappresenta la probabilità di avere un rinnovo nell'intervallo $\langle t, t + dt \rangle$.

Il seguente teorema che viene di seguito permette calcolare la densità di rinnovo di un *PH-Renewal process*.

Teorema 2.6. *Sia un renewal process con PH-representation $(\boldsymbol{\tau}, \mathbf{B})$. Allora, la sua funzione di densità di rinnovo è data da*

$$r(t) = \boldsymbol{\tau} e^{D t} \boldsymbol{\beta}, \quad t \geq 0.$$

inoltre,

$$R(t) = 1 + \int_0^t r(u) du.$$

□

Al fine di valutare $R(t)$, abbiamo bisogno della seguente proprietà algebrica.

Lemma 2.2. *La matrice $D - \Pi$ è non singolare, dove*

$$\Pi = \mathbf{e} \cdot \boldsymbol{\pi}$$

e

$$\boldsymbol{\pi} = (\boldsymbol{\tau} \mathbf{B}^{-1} \mathbf{e})^{-1} \boldsymbol{\tau} \mathbf{B}^{-1}.$$

Il vettore $\boldsymbol{\pi}$ è il vettore di probabilità stazionario del processo fase, quindi la soluzione del sistema $\boldsymbol{\pi} D = \mathbf{0}$ e $\boldsymbol{\pi} \mathbf{e} = 1$. Inoltre, abbiamo che $\boldsymbol{\pi} (D - \Pi)^{-1} = -\boldsymbol{\pi}$.

□

Con quest'ultimo risultato si può facilmente determinare una espressione per la funzione di rinnovo.

Teorema 2.7. *La funzione di rinnovo per il PH-Renewal process (iniziando con un rinnovo al tempo 0) con rappresentazione $(\boldsymbol{\tau}, \mathbf{B})$ è data da*

$$R(t) = 1 + t M^{-1} + \boldsymbol{\tau} (e^{D t} - I) (D - \Pi)^{-1} \boldsymbol{\beta},$$

ove $M = \boldsymbol{\tau} (-\mathbf{B})^{-1} \mathbf{e}$ è "the mean interrenewal time" e $\Pi = \mathbf{e} \cdot \boldsymbol{\pi}$.

□

Possiamo anche calcolare la funzione generatrice di probabilità del numero di eventi fino a tempo "t" in un processo di rinnovo tipo fase, che sarebbe

$$P(z, t) = e^{(\mathbf{B} + z \boldsymbol{\beta} \cdot \boldsymbol{\tau}) t}.$$

Si vede che questa equazione è un analogo del corrispondente risultato per il processo di Poisson (è sufficiente fare $m = 1$, $\mathbf{B} = -\lambda$ e $\boldsymbol{\tau} = 1$).

From PH-Renewal processes to MAPs

In questa sezione generalizziamo i *PH-Renewal processes*. Consideriamo di nuovo il processo $\{\Psi(t) = (N(t), \phi(t)) : t \geq 0\}$ dove $N(t)$ è il numero di rinnovi nell'intervallo $\langle 0, t \rangle$ e $\phi(t)$ è la fase al tempo "t". Il suo generatore infinitesimale è

$$G = \begin{pmatrix} \mathbf{B} & A & & & \\ & \mathbf{B} & A & & \\ & & \mathbf{B} & A & \\ & & & \ddots & \ddots \\ & & & & \ddots & \ddots \end{pmatrix} \quad (2.2.3)$$

Dove $A = \boldsymbol{\beta} \cdot \boldsymbol{\tau}$. Se ci dividiamo lo spazio degli stati $\{(n, j) : n \geq 0 \text{ e } 1 \leq j \leq m\}$ in sottoinsiemi

$$\ell(k) = \{(k, 1), \dots, (k, m)\} \quad \forall k \geq 0,$$

chiamati *livelli*, grazie all'ordine lessicografico si vede che la diagonale a blocchi in (2.2.3) sono formati dai tassi in cui si verificano cambiamenti di fase, mentre il livello rimane lo stesso. I blocchi sopra la diagonale sono formati dai tassi a cui si verifica un cambiamento di livello: il processo si muove di un livello al seguente livello successivo superiore e può verificarsi anche un cambio di fase. Abbiamo visto che in un *PH-Renewal process* lo sfasamento avviene in modo molto ristretto poiché la nuova fase dopo assorbimento è scelta con la stessa distribuzione $\boldsymbol{\tau}$. Possiamo generalizzare questo processo in due modi diversi:

Primo modo:

Scrivendo i vettori riga della matrice A , osserviamo

$$A = \begin{pmatrix} \beta_1 \boldsymbol{\tau} \\ \vdots \\ \beta_m \boldsymbol{\tau} \end{pmatrix}$$

che gli ingressi di ogni riga di A differiscono solo per un scalare β_i e quindi la nuova fase dopo un arrivo è scelta indipendentemente dalla fase immediatamente prima dell'arrivo. Se ci rilassiamo questa limitazione, si arriva a una nuova matrice

$$\tilde{A} = \begin{pmatrix} \beta_1 \boldsymbol{\tau}_1 \\ \vdots \\ \beta_m \boldsymbol{\tau}_m \end{pmatrix}$$

con l'unico requisito che $\boldsymbol{\tau}_i \mathbf{e} = 1$ per ogni $i = 1, \dots, m$. Qui la *PH-distribution* dopo un arrivo dipende dalla fase immediatamente prima dell'arrivo. Inoltre, il requisito $\boldsymbol{\tau}_i \mathbf{e} = 1$ in collegamento al fatto che $\boldsymbol{\beta} = -\mathbf{B}\mathbf{e}$ semplicemente riafferma l'osservazione che gli ingressi di ogni riga di una matrice generatore sommano a zero.

Quindi non c'è alcuna restrizione effettiva nella scelta di un \tilde{A} . Se indichiamo $D_0 := \mathbf{B}$ e $D_1 := \tilde{A}$ come di solito si trova nella letteratura matematica, si arriva ad un generatore del tipo

$$Q = \begin{pmatrix} D_0 & D_1 & & & \\ & D_0 & D_1 & & \\ & & D_0 & D_1 & \\ & & & \ddots & \ddots \\ & & & & \ddots & \ddots \end{pmatrix} \quad (2.2.4)$$

Con $D_1 \geq 0$, $D_0(i, j) \geq 0$ per $1 \leq i \neq j \leq m$, $D_0(i, i) < 0$ per $1 \leq i \leq m$, e $(D_0 + D_1)\mathbf{e} = \mathbf{0}$. Un processo di Markov con un tale generatore è chiamato **Markovian Arrival Process (MAP)**. Questi processi di arrivo svolgono un ruolo importante nei modelli di teoria di code.

Secondo modo:

Possiamo permettere la transizione in un passo del livello $\ell(k)$ ad un certo livello $\ell(k + \nu)$ con $\nu \geq 1$, quindi si arriva ad un generatore del tipo

$$Q = \begin{pmatrix} D_0 & D_1 & D_2 & D_3 & \cdots \\ 0 & D_0 & D_1 & D_2 & \cdots \\ 0 & 0 & D_0 & D_1 & \cdots \\ 0 & 0 & 0 & D_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (2.2.5)$$

Dove D_0, D_1, D_2, \dots sono matrici di ordine “ m ” con $D_k \geq 0$ per ogni $k \geq 1$, $D_0(i, j) \geq 0$ per $1 \leq i \neq j \leq m$, $D_0(i, i) < 0$ per $1 \leq i \leq m$, e $\sum_{k \geq 0} D_k \mathbf{e} = \mathbf{0}$. Un processo di Markov con un tale generatore è chiamato *Batch Markovian Arrival Process (BMAP)*.

2.2.3 Transient Markov Arrival Processes

Studieremo ora un tipo di MAP chiamata “*Transient Markov Arrival Processes*” ([35],[40]). Un *transient-MAP*, a differenza di un MAP ordinario, termina dopo un numero finito di arrivi, quasi sicuramente. Per il resto di questo capitolo e della tesi tratteremo con *transient-MAPs* e ci riferiremo a loro come semplicemente *MAPs*. Il processo di base che governa la crescita di un *Markovian Binary Tree* (MBT) è precisamente il MAP.

La matrice dei tassi di transizione per il MAP è

$$Q = \begin{pmatrix} D_0^* & D_1^* & & & \\ & D_0^* & D_1^* & & \\ & & D_0^* & D_1^* & \\ & & & \ddots & \ddots \end{pmatrix} \quad (2.2.6)$$

Supponiamo che esista una fase assorbente “0” tale che, per ogni “ j ”, $(D_1^*)_{0j} = (D_0^*)_{0j} = 0$ e che nella matrice $D^* = D_0^* + D_1^*$ esista una traiettoria di tasso positiva da tutte le altre fasi alla fase “0”. Pertanto, se $\phi(t_0) = 0$ per alcun “ t_0 ”, allora $\phi(t) = 0$ e $N(t) = N(t_0)$ per ogni $t \geq t_0$, quindi, il processo cessa di evolvere una volta che la fase “0” viene raggiunta. Diremo che si è verificato *una catastrofe* quando il processo si sposta in uno stato della forma $(n, 0)$.

Definizione 2.9. Un MAP transitorio è un processo di conteggio generata da un processo di Markov bidimensionale $\{\Psi(t) = (N(t), \phi(t)) : t \geq 0\}$ sullo spazio degli stati $\{(n, i) : n \in \mathbb{N}_0, i \in \{0, 1, \dots, m\}\}$ con matrice di transizione della forma (2.2.6), dove

$$D_0^* = \begin{pmatrix} 0 & \mathbf{0} \\ d_0 & D_0 \end{pmatrix} \quad e \quad D_1^* = \begin{pmatrix} 0 & \mathbf{0} \\ d_1 & D_1 \end{pmatrix}$$

con $D_1 \geq 0$, $d_0, d_1 \geq \mathbf{0}$, $D_0(i, j) \geq 0$ per $1 \leq i \neq j \leq m$, $D_0(i, i) < 0$ per $1 \leq i \leq m$ e $(D_0^* + D_1^*)\mathbf{e} = \mathbf{0}$, cioè

$$D_0\mathbf{e} + D_1\mathbf{e} + d_0 + d_1 = \mathbf{0}. \quad (2.2.7)$$

Così, oltre ai tradizionali blocchi D_0 e D_1 , introduciamo i tassi d_0 e d_1 in cui si verificano la catastrofe. Chiaramente, ogni stato della forma $(n, 0)$ è assorbente.

Precisiamo un po' come si comporta il processo, supponga per un momento che lo stato attuale di un MAP è (n, i) , cioè, ci sono stati “ n ” transizioni osservabili e il processo si trova nella fase “ i ”. Il processo rimarrà in questo stato per un periodo di tempo distribuito esponenzialmente con media $-1/(D_0)_{ii}$. Nella transizione successiva ci sono quattro possibilità:

- Per $j \neq i$, con probabilità $-(D_0)_{ij}/(D_0)_{ii}$ si verificherà una **transizione nascosta** (*hidden transition*) alla fase “ j ”, e così il nuovo stato sarà (n, j) .
- Con probabilità $-(D_1)_{ij}/(D_0)_{ii}$ si verificherà una **transizione osservabile** alla fase “ j ”, e così il nuovo stato sarà $(n + 1, j)$.
- Con probabilità $-(d_0)_i/(D_0)_{ii}$ si verificherà una **transizione nascosta** alla fase “0”, e così il nuovo stato sarà $(n, 0)$. A questo evento si le chiama **catastrofe**.
- Con probabilità $-(d_1)_i/(D_0)_{ii}$ si verificherà una **transizione osservabile** alla fase “0”, e così il nuovo stato sarà $(n + 1, 0)$. A questo evento si le chiama anche **catastrofe**.

Quindi, la variabile aleatoria $N(t)$ conta il numero di *transizioni osservabili* che si verificano fino al tempo “ t ” dove i loro tassi di transizioni sono controllate dalla matrice D_1^* , mentre i tassi delle *transizioni nascoste* sono controllate dalla matrice D_0^* .

La MAP transitoria è caratterizzata dalla 4-tuple $(\boldsymbol{\tau}, D_0, D_1, d_0)$, dove $\boldsymbol{\tau} = (\tau_i)$ è un vettore di dimensione $1 \times m$, con τ_i che denota la probabilità che $\phi(0) = i$. La probabilità τ_0 che $\phi(0) = 0$ è implicitamente definito da $\tau_0 = 1 - \boldsymbol{\tau}\mathbf{e}$ e il vettore d_1 è definito da (2.2.7).

Possiamo anche definire un processo di *tempi di arrivo* $\{T_n : n \in \mathbb{N}_0\}$ a cui gli eventi della MAP si verificano, quindi $T_0 = 0$ e $T_n = \inf\{t \geq 0 : N(t) = n\}$ se esiste un tale “ t ”, altrimenti $T_n = \infty$.

Supponga ora che le seguenti condizioni sono soddisfatte:

Condizione 2.1. La matrice $D = D_0 + D_1$ è non singolare.

Questa condizione implica che gli stati della forma (n, i) con $i \neq 0$ sono transitorie e che, quasi sicuramente, il processo entrerà in uno degli stati assorbenti $(n, 0)$ in un tempo finito. Infatti, la matrice dei tassi di transizione del processo $\{\phi(t)\}$ è

$$D^* = \begin{pmatrix} 0 & \mathbf{0} \\ d_0 + d_1 & D \end{pmatrix}$$

La fase 0 è chiaramente assorbente e le fasi da 1 a m sono tutte transitorie se e solo se D è non singolare, (Teorema (2.5)).

Condizione 2.2. Per ogni fase $i \neq 0$ esiste al meno un “ j ” tale che c’è una traiettoria di (n, i) verso $(n + 1, j)$.

Se questa condizione non è soddisfata, allora esisteranno fasi $i \neq 0$ tale che una volta che il processo entra in uno stato della forma (n, i) mai uscirà (*quindi (n, i) sarebbe uno stato assorbente, situazione che vogliamo evitare*).

Per una MAP transitoria ci sono tre variabili aleatorie di molto interesse, questi sono:

1.- La durata L del processo, definita come

$$L = T_K = \sup\{T_n : T_n < \infty\},$$

2.- Il tempo V finché la catastrofe si verifica, definita come

$$V = \inf\{t \geq 0 : \phi(t) = 0\},$$

3.- Il numero totale di eventi K , definita come

$$K = \lim_{t \rightarrow \infty} N(t).$$

Il calcolo delle distribuzioni di probabilità per ognuno di queste variabili aleatorie con le loro dimostrazioni possiamo trovarlo in [40].

Qui invece siamo interessati nelle proprietà di correlazioni che può generare il processo fase di una MAP, tra il tempo trascorso tra due eventi osservabili e la fase del processo immediatamente dopo il secondo di questi due eventi osservabili quando $d_1 = 0$. Tali correlazioni sorgono perché i tassi di transizione dipendono dalla fase sottostante. Quello che ci interessa qui, è la distribuzione della fase immediatamente dopo un evento osservabile. Come primo passo determineremo questa distribuzione e poi illustreremo le correlazioni attraverso un esempio concreto.

Vediamo tre elementi importanti:

- La probabilità $(P_h(t))_{ij}$ che, al tempo “ t ”, la fase del processo è “ j ” dato che il processo è iniziato nella fase “ i ” con la condizione che tutte le transizioni di fasi intermedie sono *nascosti*, in forma matriciale è data da

$$P_h(t) = e^{D_0 t}. \quad (2.2.8)$$

- La densità $(R_0(t))_{ij}$, che il primo evento osservabile accada al tempo “ t ” e poi cambia allo stato “ j ” dato che il processo è iniziato nella fase “ i ”, in forma matriciale è data da

$$R_0(t) = e^{D_0 t} D_1. \quad (2.2.9)$$

- La probabilità $(P_{os}(t))_{ij}$ che, la fase del processo è “ j ” immediatamente dopo il primo evento osservabile, dato che il processo è iniziato nella fase “ i ” con la condizione che il primo evento osservabile è successo al tempo “ t ”, è data da

$$(P_{os}(t))_{ij} = \frac{(R_0(t))_{ij}}{(R_0(t) \cdot \mathbf{e})_i}. \quad (2.2.10)$$

L’ultima equazione illustra il fatto che la distribuzione della fase immediatamente dopo il primo evento osservabile dipende dal tempo “ t ” di tale primo evento osservabile.

Esempio 2.1. Consideriamo una MAP di quattro fasi:

Si considera il MAP sulle fasi $\{1, 2, 3, 4\}$ definita da

$$D_0 = \begin{pmatrix} -20.00 & 0.00 & 0.00 & 0.10 \\ 0.10 & -20.00 & 19.40 & 0.10 \\ 0.10 & 19.00 & -20.00 & 0.10 \\ 0.10 & 0.10 & 0.10 & -20.00 \end{pmatrix}$$

e

$$D_1 = \begin{pmatrix} 0.00 & 0.10 & 0.01 & 19.30 \\ 0.00 & 0.01 & 0.10 & 0.00 \\ 0.10 & 0.20 & 0.10 & 0.20 \\ 15.10 & 0.10 & 0.10 & 0.10 \end{pmatrix}$$

Un calcolo facile ci mostra che

$$P_{os}(0.01) = \begin{pmatrix} 0.0008 & 0.0052 & 0.0045 & 0.9936 \\ 0.1249 & 0.1849 & 0.4542 & 0.2260 \\ 0.1766 & 0.3070 & 0.1806 & 0.3357 \\ 0.9791 & 0.0065 & 0.0065 & 0.0078 \end{pmatrix}$$

$$P_{os}(0.1) = \begin{pmatrix} 0.0077 & 0.0052 & 0.0006 & 0.9866 \\ 0.2295 & 0.2063 & 0.1987 & 0.3655 \\ 0.2290 & 0.2099 & 0.1946 & 0.3665 \\ 0.9667 & 0.0069 & 0.0068 & 0.0195 \end{pmatrix}$$

L’indicazione più evidente della dipendenza della distribuzione della fase immediatamente dopo l’evento osservabili rispetto al tempo dell’evento viene quando si guarda la seguente tabella:

Cambio di fase	$t = 0.01$	$t = 0.1$
$2 \rightarrow 3$	$(P_{os}(0.01))_{23} = 0.4542$	$(P_{os}(0.1))_{23} = 0.1987$
$2 \rightarrow 4$	$(P_{os}(0.01))_{24} = 0.2260$	$(P_{os}(0.1))_{24} = 0.3655$
$1 \rightarrow 1$	$(P_{os}(0.01))_{11} = 0.0008$	$(P_{os}(0.1))_{11} = 0.0077$

Nella prima riga della tabella si vede che se il processo inizia nella fase “2”, quanto più presto accada il primo evento osservabile è più probabile che il processo entrerà nella fase “3”. Al contrario, nella seconda riga si vede che quanto più lungo è la durata del tempo che accada il primo evento osservabile è più probabile che il processo entrerà nella fase “4”. Nel caso della transizione della fase “1” alla fase “1” (terza riga) la probabilità di un tale evento aumenta quasi 10 volte

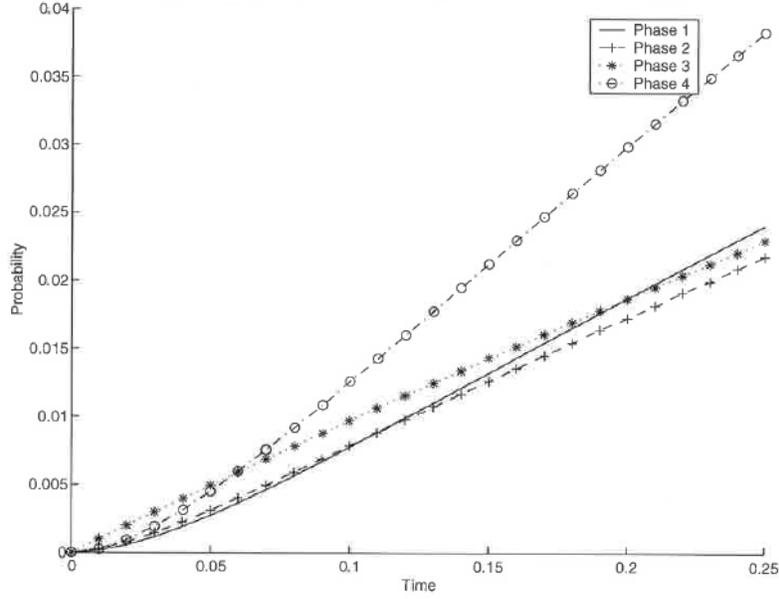


Figure 2.5: Distribuzione di probabilità per un Evento Osservabile rispetto al Tempo della Fase “2”

in quello spazio di tempo. La ragione di questo è che dalla fase “1” non ci può essere nessuna transizione osservabile direttamente che torni alla fase “1”, cioè, bisogna prima passare per alcuna transizione nascosta, ad esempio “4” seguita da una transizione osservabile dalla fase “4” alla fase “1” e questo aumenterà la probabilità in funzione del tempo.

Così, si vede che il livello di correlazione dipenderà dalla natura delle matrici D_0 e D_1 . Più avanti applicheremo questo criterio quando costruiamo la MBT usando la MAP.

Sia ora $P_{ij}(t)$ la probabilità che ci sarà una transizione osservabile alla fase “ j ” nell’intervallo $[0, t]$ dato che il processo è iniziato nella fase “ i ”, e $P(t) = (P_{ij}(t))$, allora

$$\begin{aligned} P(t) &= \int_0^t e^{D_0 x} D_1 dx \\ &= (D_0)^{-1} (I - e^{D_0 t}) D_1. \end{aligned} \quad (2.2.11)$$

La figura (2.5) rappresenta graficamente le variazioni rispetto del tempo, della probabilità di una transizione osservabile nelle quattro fasi dato che il processo è iniziato nella fase “2” per il MAP dell’esempio (4.1).

Questa figura dimostra abbastanza chiaramente come la probabilità che un evento osservabile si verifica dipende dal tempo. Ad esempio, una transizione osservabile dalla fase “2” alla fase “3” è la più probabile entro il tempo $t = 0.06$ dopo il quale la fase “4” diventa la più probabile. Inoltre, fino a circa $t = 0.11$ una transizione osservabile alla fase “1” è la transizione meno probabile, perché la transizione $2 \rightarrow 1$ non può avvenire direttamente, ma richiede un numero di transizioni nascoste, diciamo, una transizione $2 \rightarrow 4$ seguito da una transizione di $4 \rightarrow 1$. La probabilità di queste transizioni aumenta col tempo, aumentando così la probabilità di una transizione osservabile $2 \rightarrow 1$; la transizione osservabile $2 \rightarrow 1$ diventa la seconda transizione osservabile più probabile quando aumenta il tempo. Tali comportamenti osservati sono dovuti alle complesse transizioni interne e nascoste che sono generati dalla matrice D_0 .

È chiaro dall’equazione (2.2.11) che il tempo fino alla prossima transizione osservabile non è distribuita esponenzialmente. Sia T il tempo medio fino a quando un evento osservabile si è verificato,

dato che il processo è iniziato nella fase “ i ”, allora

$$T = \int_0^{\infty} t e^{D_0 t} D_1 e dt \quad (2.2.12)$$

Dopo l'integrazione per parti abbiamo

$$T = (-D_0)^{-2} D_1 e. \quad (2.2.13)$$

Nuovamente facendo riferimento all'esempio (4.1), il tempo medio tra le transizioni osservabili subordinata alla fase iniziale è,

$$T = \begin{pmatrix} 0.0492 \\ 0.8451 \\ 0.8363 \\ 0.0475 \end{pmatrix} \quad (2.2.14)$$

Questo è quello che ci saremmo aspettati, dato che le fasi “1” e “4” hanno un'alta probabilità di innescare transizioni osservabili in tempi precedenti, mentre le fasi “2” e “3” sono più propensi a sottoporsi a transizioni nascosti prima di uno osservabile.

Capitolo 3

Quasi-Birth-and-Death Processes

Questo capitolo introduce *The Quasi-Birth-and-Death Process QBDs* che sono una generalizzazione dei *Birth-and-Death processes*. Le QBDs sono una classe di catene di Markov strutturate di tipo bivariata con lo spazio degli stati $S = \{(n, i) : n \in \mathbb{N}_0, i = 1, 2, \dots, m_n\}$, dove la prima coordinata “ n ” è chiamata il *livello*, la seconda coordinata “ i ” è chiamata la *fase* dello stato (n, i) e “ m_n ” è un intero positivo che può essere finito o infinito e anche dipendere del livello. Il processo è restretto rispetto ai salti di livelli, cioè, i salti sono ammessi solo ai suoi vicini più prossimi, ma è senza restrizioni rispetto alla sua fase. Quando le transizioni di un processo QBD sono indipendenti rispetto al livello, stiamo parlando di un *level-independent QBD (LIQBD)*, altrimenti, si parla di un *level-dependent QBD (LDQBD)*. Vediamo di seguito ognuno di questi tipi.

3.1 Level Independent Quasi-Birth-and-Death Processes - (LIQBD)

La versione *LIQBD* è stato ampiamente studiato da un grande numero di ricercatori. Un trattamento molto lucida su questo argomento possiamo trovarlo nell’eccellenti testi di Marcel Neuts [43] e Latouche-Ramaswami [39]. Qui, studieremo entrambe le versioni discrete e continue del processo LIQBD.

3.1.1 LIQBD a tempo discreto

Definizione 3.1. *Un level-independent QBD a tempo discreto, è un processo di Markov $\{\Phi_t = (N_t, \phi_t) : t \in \mathbb{N}_0\}$ sullo spazio degli stati bidimensionale $S = \{(n, j) : n \in \mathbb{N}_0, j = 1, 2, \dots, m\}$ e con matrice di transizione della forma:*

$$P = \begin{pmatrix} B & A_0 & 0 & 0 & \cdots \\ A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (3.1.1)$$

Nella definizione appena citata, A_0 , A_1 , A_2 e B sono matrici quadrate di dimensione m e non negative, inoltre, la somma degli elementi nelle righe di P devono essere uguale all'unità, cioè,

$$(A_2 + A_1 + A_0)\mathbf{e} = \mathbf{e},$$

e

$$(B + A_0)\mathbf{e} = \mathbf{e}.$$

Osservazione:

- Le uniche possibili transizioni di stato sono:
 - ▷ Dallo stato (n, i) allo stato $(n + 1, j)$, con probabilità $(A_0)_{ij}$, $\forall n \geq 0, \forall i, j = 1, 2, \dots, m$.
 - ▷ Dallo stato (n, i) allo stato (n, j) , con probabilità $(A_1)_{ij}$, $\forall n \geq 0, \forall i, j = 1, 2, \dots, m$.
 - ▷ Dallo stato (n, i) allo stato $(n - 1, j)$, con probabilità $(A_2)_{ij}$, $\forall n \geq 1, \forall i, j = 1, 2, \dots, m$.
- Le probabilità di transizioni sono indipendenti rispetto ai livelli.
- Supporremo in tutta questa tesi che P sia irriducibile ,i.e, esiste una traiettoria tra qualsiasi due stati.
- Ogni livello è rappresentato come $\ell(n) = \{(n, 1), (n, 2), \dots, (n, m)\}$ per ogni $n \geq 0$, quindi $S = \bigcup_{n \geq 0} \ell(n)$.

Una domanda naturale è se il processo di Markov con matrice di transizione P possiede una distribuzione limite, e in caso affermativo, in quali condizioni esiste tale distribuzione e qual è la sua forma. Sappiamo che la distribuzione limite è strettamente legata alle proprietà di ricorrenza e periodicità, quindi se assumiamo momentaneamente che il processo è aperiodico e positivo ricorrente potremo caratterizzare la distribuzione limite π .

Sia $\pi > 0$ la distribuzione limite di $\{\Phi_t : t \in \mathbb{N}_0\}$ che risolve univocamente ¹ il sistema:

$$\begin{cases} \pi P &= \pi \\ \pi \mathbf{e} &= 1 \end{cases} \quad (3.1.2)$$

Ora, se partizionamo il vettore π in sottovettori in modo che

$$\pi = (\pi_0, \pi_1, \dots),$$

ove π_n contiene le probabilità limite per gli stati nel livello “ n ”, i.e., negli stati dell’insieme $\ell(n) = \{(n, 1), (n, 2), \dots, (n, m)\}$, allora il sistema (3.1.2) può essere decomposto come

$$\begin{aligned} \pi_0(B - I) + \pi_1 A_2 &= \mathbf{0}, \\ \pi_{n-1} A_0 + \pi_n(A_1 - I) + \pi_{n+1} A_2 &= \mathbf{0} \quad \text{per } n \geq 1, \\ \sum_{n \geq 0} \pi_n \mathbf{e} &= 1. \end{aligned}$$

In questi condizioni abbiamo il seguente teorema:

¹Dato che abbiamo supposto che il processo è irriducibile, aperiodico e positivo ricorrente.

Teorema 3.1. *Se il processo LIQBD a tempo discreto $\{\Phi_t : t \in \mathbb{N}_0\}$ è positivo ricorrente, allora esiste una matrice non negativa N di ordine “ m ” tale che:*

$$\pi_{n+1} = \pi_n A_0 N \quad \forall n \geq 0,$$

dove la matrice N è tale che N_{ij} ($1 \leq i, j \leq m$) è uguale al numero atteso di visite allo stato (n, j) , iniziando dallo stato (n, i) , prima della prima visita a qualsiasi degli stati di $\ell(n-1)$ ed è indipendente di $n \geq 1$.

Possiamo anche scrivere

$$\pi_n = \pi_0 R^n \quad \forall n \geq 0, \quad (3.1.3)$$

dove $R = A_0 N$ è tale che, per ogni $n \geq 0$, R_{ij} ($1 \leq i, j \leq m$) è uguale al numero atteso di visite allo stato $(n+1, j)$ prima di un ritorno all'insieme $\ell(0) \cup \dots \cup \ell(n)$, dato che il processo è iniziato dallo stato (n, i) . □

La matrice R può anche essere interpretata come la matrice che rappresenta il tasso medio di visite agli stati di $\ell(n+1)$ per unità del tempo trascorso negli stati di $\ell(n)$ misurata nel tempo locale dell'intero sottoinsieme $L = \ell(0) \cup \dots \cup \ell(n)$ (vedere Oss. 5.3.2 in [39]). A causa della struttura del QBD, la catena di Markov è obbligato a passare attraverso il livello “ n ” quando lascia l'insieme complementare L^c . Perciò, possiamo limitarci solo al tempo locale del livello “ n ” e interpretare R_{ij} come il tasso medio di visita a $(n+1, j)$ per unità del tempo locale di $\ell(n)$ trascorso in (n, i) .

Corollario 3.1. *Se il QBD è positivo ricorrente e “ m ” è finito, allora il raggio spettrale $sp(R)$ della matrice R è strettamente minore di 1, cioè $sp(R) < 1$.* □

Nel caso quando “ m ” è infinito, si può dimostrare che esiste $\theta > 1$ tale che $\sum_{n \geq 0} \theta^n (R^n)_{ij}$ diverge per alcuni “ i ” e “ j ”, e $\sum_{n \geq 0} r^n R^n$ converge quando $r < \theta$. La quantità $\theta^{-1} < 1$ è quindi l'analogo naturale del raggio spettrale per matrici finite.

Il teorema (3.1) fornisce un'interpretazione della matrice R , ma è possibile estendere questa idea e anche dare un'interpretazione delle sue potenze successive R^n . Per ogni $n \geq 0$, $k \geq 1$ e $1 \leq i, j \leq m$; $(R^k)_{ij}$ è il numero atteso di visite allo stato $(n+k, j)$ prima del primo ritorno all'insieme $\ell(0) \cup \dots \cup \ell(n)$ dato che il processo è iniziato nello stato (n, i) . La quantità $(R^k)_{ij}$ può anche essere interpretato come il tasso medio di visite allo stato $(n+k, j)$ per unità del tempo locale di $\ell(n)$ trascorso in (n, i) ; quest'ultima interpretazione, a differenza di quella precedente, ci sarà utile per l'analisi nei casi di tempo continuo.

Vediamo ora altre caratterizzazioni della distribuzione stazionaria π che sono più adatti per calcoli numerici, per questo introduciamo altre due matrici U e G . Sottolineiamo che la matrice R , e le altre nuove matrici U e G non sono solo artefatti matematici, ma sono legati in modo fondamentale alla dinamica dei QBDs.

Supponiamo che $\Phi_0 \in \ell(n)$. Siano τ la prima epoca di visita al livello $\ell(n-1)$ e θ la prima epoca di ritorno al livello $\ell(n)$. Allora, le matrici U e G sono definite come segue,

$$U_{ij} = \mathbb{P}\{\theta < \tau \ \& \ \Phi_\theta = (n, j) | \Phi_0 = (n, i)\}; \quad (3.1.4)$$

la matrice U rappresenta la probabilità che, iniziando da $\ell(n)$, ritornare a $\ell(n)$ prima di visitare $\ell(n-1)$.

$$G_{ij} = \mathbb{P}\{\tau < \infty \ \& \ \Phi_\tau = (n-1, j) | \Phi_0 = (n, i)\}; \quad (3.1.5)$$

la matrice G rappresenta la probabilità che, iniziando in $\ell(n)$, visitare $\ell(n-1)$ in un tempo finito.

Siccome la matrice P è omogenea, allora i valori di U e G non dipendono di $n \geq 1$. Il teorema che segue mostra la relazione fra le tre matrici appena descritti.

Teorema 3.2. *Se qualsiasi delle matrici U , G o R è conosciuta, allora è possibile determinare le altre due tramite le seguenti equazioni:*

$$\begin{aligned} R &= A_0(I - U)^{-1}, \\ G &= (I - U)^{-1}A_2, \\ U &= A_1 + A_0G, \\ U &= A_1 + RA_2. \end{aligned} \quad (3.1.6)$$

□

Notare che il teorema (3.2) dà uno strumento per conoscere una delle matrici R , G o U in funzione di una qualsiasi delle altre, tuttavia, ci piacerebbe avere qualche strumento per conoscere (separatamente) ognuno di questi matrici soltanto in termini delle matrici dati A_0 , A_1 e A_2 . Il seguente teorema risponde a questa esigenza.

Teorema 3.3. *Le tre matrici U , G e R , rispettivamente, soddisfano le seguenti equazioni:*

$$G = A_2 + A_1G + A_0G^2, \quad (3.1.7)$$

$$R = A_0 + RA_1 + R^2A_2, \quad (3.1.8)$$

$$U = A_1 + A_0(I - U)^{-1}A_2. \quad (3.1.9)$$

□

Queste equazioni hanno le seguenti interpretazioni probabilistiche:

- Nel caso di (3.1.7). Il lato sinistro dell'uguaglianza rappresenta la distribuzione del primo stato visitato in $\ell(n-1)$, condizionata allo stato iniziale in $\ell(n)$; mentre il lato destro mostra una decomposizione secondo la prima transizione: il primo termine corrisponde al caso in cui il QBD si muove direttamente da $\ell(n)$ a $\ell(n-1)$ in una transizione con probabilità registrati in A_2 ; il secondo termine corrisponde al caso in cui con le probabilità registrati in A_1 il QBD rimanga in $\ell(n)$, dove ancora eventualmente si sposterà a $\ell(n-1)$ con probabilità registrati in G ; infine, il terzo termine corrisponde al caso in cui con le probabilità registrati in A_0 il QBD si muove a $\ell(n+1)$ da dove eventualmente con probabilità registrati in G ritornerà a $\ell(n)$, e poi un'altra volta con probabilità registrati in G si muoverà a $\ell(n-1)$.

- L'interpretazione di (3.1.8) è semplice. Il lato sinistro dell'uguaglianza, rappresenta il numero atteso di visite a $\ell(n)$, iniziando in $\ell(n-1)$ e prima di un primo ritorno a $\ell(n-1)$; mentre nel lato destro le visite a $\ell(n)$ sono decomposti in tre gruppi a seconda del livello di dove si fa la visita a $\ell(n)$: il primo termine corrisponde al caso in cui si verifica una prima visita a $\ell(n)$ immediatamente dopo iniziare da $\ell(n-1)$ con probabilità registrata in A_0 ; il secondo termine corrisponde al caso in cui per ogni visita a $\ell(n)$ il numero atteso è registrato in R e può essere immediatamente seguito da un'altra visita a $\ell(n)$ con probabilità registrata in A_1 ; infine, il terzo termine corrisponde al caso in cui ogni visita a $\ell(n+1)$ con numero atteso registrato in R^2 può essere immediatamente seguito da un'altra visita a $\ell(n)$ con probabilità registrata in A_2 .
- L'interpretazione di (3.1.9) è immediata se scriviamo $U = A_1 + \sum_{i \geq 0} A_0 U^i A_2$. Infatti, il termine A_1 registra la probabilità di un immediato ritorno a $\ell(n)$; il termine $A_0 U^i A_2$ registra la probabilità di ritornare a $\ell(n)$ dopo esattamente " $i+1$ " visite a $\ell(n+1)$.

Per specificare completamente la distribuzione stazionaria π , abbiamo bisogno di determinare il subvector π_0 , questo lo facciamo con l'aiuto del seguente lemma:

Lemma 3.1 (Boundary distribution). *La distribuzione stazionaria π_0 degli stati limite in $\ell(0)$ è l'unica soluzione del sistema*

$$\begin{aligned}\pi_0(B + A_0G) &= \pi_0 \\ \pi_0(I - R)^{-1}e &= 1,\end{aligned}$$

ove la prima equazione può anche essere scritta come

$$\pi_0(B + RA_2) = \pi_0$$

□

Quando si verifica che $B = A_1 + A_2$ è possibile semplificare la caratterizzazione di π_0 . Ora si definiamo la matrice stocastica $A = A_0 + A_1 + A_2$, abbiamo il seguente lemma,

Lemma 3.2. *Supponiamo che il QBD è irriducibile, positivo ricorrente e che $B = A_1 + A_2$, allora*

$$\pi_n = \alpha(I - R)R^n \quad \forall n \geq 0,$$

ove α è l'unica soluzione del sistema $\alpha = \alpha A$ e $\alpha e = 1$.

□

3.1.2 LIQBD a tempo continuo

Definizione 3.2. *Un level-independent QBD a tempo continuo, è un processo di Markov $\{\Phi(t) = (N(t), \phi(t)) : t \geq 0\}$ sullo spazio degli stati bidimensionale $S = \{(n, i) : n \in \mathbb{N}_0, i = 1, 2, \dots, m\}$ con matrice di transizione della forma:*

$$Q = \begin{pmatrix} B & A_0 & 0 & 0 & \cdots \\ A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Dove $A_0, A_2 \geq 0$, $B(i, j), A_1(i, j) \geq 0$ per $1 \leq i \neq j \leq m$, $B(i, i), A_1(i, i) < 0$ per $1 \leq i \leq m$, e la somma degli elementi nelle righe di Q devono essere uguale a zero, cioè

$$(B + A_0)e = \mathbf{0}, \quad e$$

$$(A_0 + A_1 + A_2)e = \mathbf{0}.$$

Osservazione:

- Le uniche possibili transizioni di stato sono:
 - Dallo stato (n, i) allo stato $(n + 1, j)$, con tasso di transizione $(A_0)_{ij}$, $\forall n \geq 0, \forall i, j = 1, 2, \dots, m$.
 - Dallo stato (n, i) allo stato (n, j) , con tasso di transizione $(A_1)_{ij}$, $\forall n \geq 0, \forall i, j = 1, 2, \dots, m$.
 - Dallo stato (n, i) allo stato $(n - 1, j)$, con tasso di transizione $(A_2)_{ij}$, $\forall n \geq 1, \forall i, j = 1, 2, \dots, m$.
- Le probabilità di transizioni sono indipendenti rispetto ai livelli.
- Le matrici B, A_0, A_1 e A_2 sono matrici quadrate di ordine “ m ”, inoltre, analogamente al caso discreto supporremo in tutta questa tesi che Q è irriducibile.
- Ogni livello è rappresentato come $\ell(n) = \{(n, 1), (n, 2), \dots, (n, m)\}$ per ogni $n \geq 0$, quindi $S = \bigcup_{n \geq 0} \ell(n)$.

L’analisi del LIQBD a tempo continuo può essere effettuata seguendo la stessa idea del caso discreto, iniziamo quindi dando la prima proprietà strutturale del processo.

Teorema 3.4. *Se il processo LIQBD a tempo continuo $\{\Phi_t : t \geq 0\}$ è positivo ricorrente, allora la distribuzione di probabilità stazionaria è tale che:*

$$\pi_n = \pi_0 R^n \quad \forall n \geq 0, \tag{3.1.10}$$

Dove la matrice R registra il tasso di soggiorno negli stati di $\ell(n + 1)$ per unità del tempo locale di $\ell(n)$. Inoltre, abbiamo che $R = A_0 N$, dove la matrice N registra il tempo di soggiorno atteso negli stati di $\ell(n)$, iniziando da $\ell(n)$ e prima della prima visita a $\ell(n - 1)$. \square

Sia $\tau(s) = \inf\{t > |\Phi(t) \in \ell(s)\}$ il primo tempo di passaggio in $\ell(n)$. Come nel caso a tempo discreto, possiamo definire due matrici G e U come segue:

$$G_{ij} = \mathbb{P}\{\tau(n-1) < \infty \ \& \ \Phi(\tau(n-1)) = (n-1, j) | \Phi(0) = (n, i)\}, \quad (3.1.11)$$

ove τ in questo caso è il primo tempo di passaggio del livello “ n ” al livello “ $n-1$ ” e G è la probabilità di un eventuale movimento a partire da $\ell(n)$ verso a $\ell(n-1)$, cioè, la probabilità che il processo eventualmente raggiunga $\ell(n-1)$ per la prima volta nella fase “ j ”, dato che è iniziato nello stato (n, i) .

Dobbiamo notare che, poiché il processo è indipendente dal livello, G_{ij} non dipende da “ n ”, quindi adotteremo un leggero abuso di notazione e scriveremo,

$$G = \mathbb{P}\{\tau < \infty \ \& \ \Phi(\tau) | \Phi(0) \in \ell(n)\}, \quad (3.1.12)$$

per indicare l’equazione (3.1.11) ed evitare complicazioni future, rendendo così le equazioni più facile da leggere. Poiché questo processo è omogeneo rispetto al livello, abbiamo che G è indipendente dal livello $\ell(n)$ da cui inizia il processo, con la condizione che $n > 0$.

Si può dimostrare che G è la soluzione minima non negativa dell’equazione matriciale quadratica

$$F = (-A_1)^{-1}A_2 + (-A_1)^{-1}A_0F^2. \quad (3.1.13)$$

e come G è la soluzione minima non negativa dell’equazione (3.1.13) possiamo scrivere

$$G = (-A_1)^{-1}A_2 + (-A_1)^{-1}A_0G^2. \quad (3.1.14)$$

L’equazione (3.1.14) ha la seguente interpretazione fisica. Se il processo inizia a $\ell(n)$ allora ci sono due modi diversi di raggiungere $\ell(n-1)$.

1. Il primo modo è una transizione diretta a $\ell(n-1)$ che si verifica con probabilità $(-A_1)^{-1}A_2$, terminando così il primo tempo di passaggio “ τ ”.
2. Nel secondo modo, il processo fa una transizione a destra e si muove fino a $\ell(n+1)$ con probabilità $(-A_1)^{-1}A_0$. Da qui, il processo eventualmente si muove da $\ell(n+1)$ verso $\ell(n-1)$, con una transizione intermedia tramite $\ell(n)$; ciascuna di queste transizioni ha probabilità G . Siccome ciascuno di questi tre eventi sono indipendenti, allora la probabilità di questo secondo modo è uguale a $(-A_1)^{-1}A_0G^2$.

Sostituendo ripetutamente il lato sinistro nel lato destro, l’equazione (3.1.14) può anche essere espresso come

$$G = \sum_{k=0}^{\infty} U^k (-A_1)^{-1}A_2 = (I - U)^{-1}(-A_1)^{-1}A_2, \quad (3.1.15)$$

dove $U = (-A_1)^{-1}A_0G$. L’equazione (3.1.15) ha la seguente interpretazione probabilistica: la matrice “ U ” è la probabilità di primo ritorno a $\ell(n)$ sotto il *taboo* che il processo non vada sotto di $\ell(n)$ dato che ha iniziato a $\ell(n)$. Matematicamente,

$$U = \mathbb{P}\{\tau(n) < \tau(n-1) \ \& \ \tau(n) < \infty \ \& \ \Phi(\tau(n)) | \Phi(0) \in \ell(n)\}, \quad (3.1.16)$$

Analogamente a G , la matrice U non dipende dal livello iniziale. Ogni termine individuale, $U^k(-A_1)^{-1}A_2$, dell’equazione (3.1.15) fornisce la probabilità che il processo ritornerà a $\ell(n)$ “ k ”

volte prima che eventualmente subisca una transizione a sinistra e poi entri a $\ell(n-1)$, dato che ha iniziato in $\ell(n)$. La probabilità totale di eventualmente entrare in $\ell(n-1)$ è dato dalla somma di questi termini per ogni “ k ”.

Inoltre, essendo G la minima soluzione non negativa dell’equazione (3.1.13), può anche essere dimostrato che G è la minima soluzione non negativa per

$$F = \sum_{k=0}^{\infty} ((-A_1)^{-1} A_0 F)^k (-A_1)^{-1} A_2. \quad (3.1.17)$$

Si può ottenere nel caso continuo un insieme di equazioni simile al caso a tempo discreto che collega le tre matrici R , G e U .

Teorema 3.5. *Se qualsiasi delle matrici U , G o R è conosciuta, allora è possibile determinare le altre due tramite le seguenti equazioni:*

$$\begin{aligned} U &= A_1 + A_0 G, \\ R &= A_0 (-U)^{-1}, \\ G &= (-U)^{-1} A_2, \end{aligned} \quad (3.1.18)$$

e che soddisfano le seguenti equazioni:

$$U = A_1 + A_0 (-U)^{-1} A_2, \quad (3.1.19)$$

$$A_0 + R A_1 + R^2 A_2 = 0, \quad (3.1.20)$$

$$A_2 + A_1 G + A_0 G^2 = 0. \quad (3.1.21)$$

□

Il subvector π_0 (*Boundary distribution*) è determinato come segue:

Lemma 3.3 (Boundary distribution). *La distribuzione stazionaria π_0 degli stati limite in $\ell(0)$ è l’unica soluzione del sistema*

$$\begin{aligned} \pi_0(B + A_0 G) &= \mathbf{0} \\ \pi_0(I - R)^{-1} e &= 1, \end{aligned}$$

ove la prima equazione può anche essere scritta come

$$\pi_0(B + R A_2) = \mathbf{0}$$

□

Sono state sviluppate alcune importanti schemi numerici per trovare la matrice R . Due approcci principali sono stati considerati da diversi gruppi di ricercatori. Il primo approccio è basato sulla *decomposizione spettrale*, su cui ci sono molte pubblicazioni di Lucantoni e Daigli, Miltra, Stern, ecc. Il secondo approccio, è un processo di riduzione logaritmico che è stato sviluppato da *Latouche e Ramaswami* ([38]).

In questo lavoro ci concentreremo solo sui metodi che ci serviranno nello sviluppo delle MBTs, questi sono: *l’algoritmo di Neuts* (chiamato anche il metodo di sostituzioni modificati) che non appartiene a nessuno dei due approcci e *l’algoritmo “U”*, dove il metodo di riduzione logaritmico è la sua versione più efficiente.

3.2 Algoritmi per LIQBD

Dal punto di vista numerico, l'analisi per l'implementazione degli algoritmi per il LIQBD a tempo continuo e a tempo discreto non sono molto diversi, quindi i metodi per trovare R , G e U sia per il caso continuo, come per il caso discreto si svilupperanno in parallelo.

Come abbiamo già visto nelle sezioni (3.1.1) e (3.1.2), se il LIQBD è aperiodico e positivo ricorrente allora abbiamo $\pi_n = \pi_0 R^n$ per il caso discreto e continuo. Neuts ha dimostrato in [47] che R può essere calcolato con una procedura iterativa in base alle seguenti equazioni:

- Caso Discreto:

$$R = A_0 + RA_1 + R^2 A_2$$

- Caso Continuo:

$$0 = A_0 + RA_1 + R^2 A_2.$$

Risolvere queste due equazioni è lo stesso che risolvere l'equazione matriciale

$$A + XB + X^2 C = 0 \tag{3.2.1}$$

dove chiaramente si osserva che $X = R$. La procedura iterativa sarebbe:

- Caso Discreto:

$$R_{n+1} = A_0(I - A_1)^{-1} + R_n^2 A_2(I - A_1)^{-1}, \quad n = 0, 1, 2, \dots,$$

- Caso Continuo:

$$R_{n+1} = A_0(-A_1)^{-1} + R_n^2 A_2(-A_1)^{-1}, \quad n = 0, 1, 2, \dots,$$

usando $R_0 = 0$ per inizializzare il processo. Queste due successioni matriciali sono monotone, non decrescenti e convergono alla soluzione minima non negativa R dell'equazione (3.2.1). Un algoritmo sulla base di questo approccio, che chiameremo l'algoritmo “*Classico*” è descritto di seguito.

Algoritmo: Classico

1. Inizializzare:

- Verificare se il *LIQBD* è a tempo discreto o continuo.

- Caso discreto:

$$M := A_0 * (I - A_1)^{-1};$$

$$N := A_2 * (I - A_1)^{-1};$$

- Caso continuo:

$$M := A_0 * (-A_1)^{-1};$$

$$N := A_2 * (-A_1)^{-1};$$

- $R_0 := 0$.

2. Fino alla convergenza:

Do:

- $R = M + R_0^2 * N$

- Verificare se: $\|R - R_0\|_\infty \leq \epsilon$

- SI: *break*

- NO: $R_0 = R$

EndDo

Benché R può essere calcolato direttamente con il metodo appena visto, in [36] si dimostra che ciò è il passo più costoso nella determinazione del vettore stazionario π .

I seguenti algoritmi che vedremo di seguito, che risultano di essere più efficiente che l'algoritmo classico usano le matrici G e U , studiati nelle sezioni precedenti. Siccome le transizioni per il processo *LIQBD* sono indipendenti dai livelli, allora gli elementi delle matrici R , G e U possono essere interpretati solo in termini dei livelli "0" e "1".

Vediamo cosa rappresentano gli elementi di queste matrici:

- R_{ij} : rappresenta il numero atteso di visite allo stato $(1, j)$, iniziando da $(0, i)$, fino a un primo ritorno al livello "0".
- G_{ij} : rappresenta la probabilità che iniziando dallo stato $(1, i)$ la catena eventualmente visiti il livello "0", e lo fa visitando lo stato $(0, j)$.
- U_{ij} : rappresenta la probabilità *taboo* che iniziando dallo stato $(1, i)$, la catena eventualmente ritorni al livello "1", e lo fa visitando lo stato $(1, j)$ sotto il *taboo* del livello "0".

Osservazione:

Ricordare che:

- 1.- La matrice R ha raggio spettrale $\rho(R) < 1$ se il *LIQBD* è positivo ricorrente e $\rho(R) = 1$ altrimenti.

2.- La matrice G è *stocastica* se il *LIQBD* è positivo ricorrente e *sub-stocastica* altrimenti.

3.- La matrice U è sempre *sub-stocastica*, quindi per il caso discreto avremo $U\mathbf{e} \leq \mathbf{e}$ e per il caso continuo $U\mathbf{e} \leq \mathbf{0}$.

Un teorema che sarà di grande aiuto è il seguente:

Teorema 3.6. *Siano la matrice stocastica $A = A_0 + A_1 + A_2$ irriducibile con distribuzione di probabilità stazionaria α e $\mu = \alpha A_0 \mathbf{e} - \alpha A_1 \mathbf{e}$. Se il *LIQBD* è irriducibile e “ m ” è finito, allora*

- *Il *LIQBD* è positivo ricorrente $\Leftrightarrow \mu < 0$.*
- *Il *LIQBD* è nullo ricorrente se $\mu = 0$.*
- *Il *LIQBD* è transitorio se $\mu > 0$.*

L’algoritmo di Neuts

L’algoritmo di Neuts, è un schema iterativo per calcolare G basato nelle equazioni (3.1.7) per il caso discreto e (3.1.13) per il caso continuo, quest’ultimo è uguale all’equazione (3.1.21) e in generale queste equazioni possono essere espresse come un’equazione matriciale della forma

$$A + BX + CX^2 = 0 \tag{3.2.2}$$

dove è chiaro che $X = G$. Si noti che l’equazione (3.2.2) è molto simile a l’equazione (3.2.1), da cui il metodo stesso prende il nome di *metodo di sostituzioni modificati*. Un algoritmo sulla base di questo approccio, che chiameremo l’algoritmo di “*Neuts*” o algoritmo N è descritto di seguito.

Algoritmo: Neuts (N)

1. Inizializzare:
 - Verificare se il *LIQBD* è a tempo discreto o continuo.
 - Caso discreto:

$$M := (I - A_1)^{-1} * A_2;$$

$$N := (I - A_1)^{-1} * A_0;$$

$$G_0 = M;$$
 - Caso continuo:

$$M := (-A_1)^{-1} A_2;$$

$$N := (-A_1)^{-1} A_0;$$

$$G_0 = M;$$
2. Fino alla convergenza:

Do:

 - $G = M + N * G_0^2$
 - Verificare se: $\|G - G_0\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: $G_0 = G$

EndDo

Neuts ha mostrato in [[47], [46]] che la sequenza di matrici definite da,

$$\begin{aligned} G_0 &= (-A_1)^{-1} A_2, \\ G_n &= (-A_1)^{-1} A_2 + (-A_1)^{-1} A_0 G_{n-1}^2, \end{aligned} \tag{3.2.3}$$

per ogni $n \geq 1$, sono non decrescenti e convergono alla soluzione minima non negativa G dell'equazione (3.2.2). Discuteremo un po' di più questo metodo nel capitolo di Markovian Binary Trees.

Algoritmo U

L'algoritmo U è uno schema iterativo basato in due equazioni date nei teoremi (3.2) per il caso discreto e (3.5) per il caso continuo, queste sono:

- Caso Discreto:

$$\begin{aligned} U &= A_1 + A_0 G, \\ G &= (I - U)^{-1} A_2, \end{aligned}$$

- Caso Continuo:

$$\begin{aligned} U &= A_1 + A_0 G, \\ G &= (-U)^{-1} A_2, \end{aligned}$$

È naturale cercare di risolvere queste equazioni per sostituzioni successive: partendo in qualche modo arbitrario, ad esempio con $G_0 = 0$, possiamo calcolare iterativamente due sequenze $\{U_k : k \geq 1\}$ e $\{G_k : k \geq 1\}$ come segue:

- Caso Discreto:

$$\begin{aligned} U_1 &= A_1, \\ G_k &= (I - U_k)^{-1} A_2, \\ U_{k+1} &= A_1 + A_0 G_k \quad k \geq 1. \end{aligned} \tag{3.2.4}$$

- Caso Continuo:

$$\begin{aligned} U_1 &= A_1, \\ G_k &= (-U_k)^{-1} A_2, \\ U_{k+1} &= A_1 + A_0 G_k \quad k \geq 1. \end{aligned} \tag{3.2.5}$$

Ora dobbiamo esaminare due situazioni: in primo luogo, se queste sequenze sono convergenti; e se lo sono vorremmo sapere se i loro limiti sono U e G .

Per il caso discreto abbiamo il seguente teorema,

Teorema 3.7. *Le successioni $\{U_k : k \geq 1\}$ e $\{G_k : k \geq 1\}$ definite da:*

$$U_k = \mathbb{P}\{\gamma(1) < \gamma(0) \ \& \ \gamma(1) < \gamma(k+1) \ \& \ \Phi_{\gamma(1)} | \Phi_0 \in \ell(1)\};$$

e

$$G_k = \mathbb{P}\{\gamma(0) < \gamma(k+1) \ \& \ \Phi_{\gamma(0)} | \Phi_0 \in \ell(1)\};$$

per $k \geq 1$ soddisfano (3.2.4), inoltre, le due successioni sono monotonicamente crescenti e convergono alle matrici U e G rispettivamente. □

E per il caso continuo abbiamo il seguente teorema,

Teorema 3.8. *Le successioni $\{U_k : k \geq 1\}$ e $\{G_k : k \geq 1\}$ definite da:*

$$\begin{aligned} U_1 &= A_1, \\ G_k &= (-U_k)^{-1} A_2, \\ U_{k+1} &= A_1 + A_0 G_k, \end{aligned}$$

per $k \geq 1$, convergono alle matrici U e G rispettivamente, inoltre, la successione $\{G_k : k \geq 1\}$ è monotona. □

Si può anche dimostrare che le successioni convergenti appena descritte (3.2.4) e (3.2.5) soddisfano le equazioni (3.1.7) e (3.1.9) per il caso discreto e le equazioni (3.1.21) e (3.1.19) per il caso continuo rispettivamente come soluzioni minimali, i dettagli possono essere trovati in [39].

L'algoritmo “ U ” è descritto di seguito.

Algoritmo: U

1. Inizializzare:
 - $U := A_1$;
 - Verificare se il *LIQBD* è a tempo discreto o continuo.
 - Caso discreto:
 $G := (I - U)^{-1} * A_2$;
 - Caso continuo:
 $G := (-U)^{-1} * A_2$;
2. Fino alla convergenza:
Do:
 - $U := A_1 + A_0 * G$;
 - Verificare se il *LIQBD* è a tempo discreto o continuo.
 - Caso discreto:
 $G := (I - U)^{-1} * A_2$;
 - Caso continuo:
 $G := (-U)^{-1} * A_2$;
 - Verificare se: $\|e - Ge\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: *continue*EndDo

Osservazione:

- Il criterio di arresto viene scelto con l'ipotesi che il *LIQBD* sia *ricorrente*, in questo caso G è stocastico, l'algoritmo funziona benissimo quando il *LIQBD* è positivo ricorrente.
- Se il *LIQBD* è transitorio, non esiste un criterio di arresto soddisfacente. Una pratica standard è quella di seguire la traccia di due matrici successive in ogni iterazione, G_{old} , ad esempio, oltre a G_* per fermare le iterazioni quando $\|G_{old} - G_*\|_\infty \leq \epsilon$. Il problema è che la convergenza può essere molto lenta, ed è possibile che uno ha $\|G_{old} - G_*\|_\infty \leq \epsilon$ mentre $\|G - G_*\|_\infty$ sia in realtà maggiore di ϵ .

Esiste una versione migliorata dell'algoritmo U , che chiameremo l'algoritmo U^* . Invece di iniziare con $G_0 = 0$, potremmo fare che il processo converga più velocemente scegliendo un valore iniziale migliore per G . Nel caso in cui il *LIQBD* è ricorrente, sappiamo che G è stocastico, possiamo quindi scegliere un'altra matrice stocastica per G_0 . Questo accelera certamente il processo di convergenza. Diamo di seguito il secondo algoritmo per calcolare G quando è stocastica.

Algoritmo: U^*

1. Inizializzare:
 - $U := A_1 + A_0$;
 - Verificare se il *LIQBD* è a tempo discreto o continuo.
 - Caso discreto:
 $G := (I - U)^{-1} * A_2$;
 - Caso continuo:
 $G := (-U)^{-1} * A_2$;
2. Fino alla convergenza:

Do:

 - $U := A_1 + A_0 * G$;
 - $G_{old} := G$;
 - Verificare se il *LIQBD* è a tempo discreto o continuo.
 - Caso discreto:
 $G := (I - U)^{-1} * A_2$;
 - Caso continuo:
 $G := (-U)^{-1} * A_2$;
 - Verificare se: $\|G - G_{old}\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: *continue*

EndDo

Notare che nell'algoritmo appena descritto abbiamo usato $G = I$. I seguenti teoremi garantiscono la convergenza di questa procedura.

Per il caso discreto,

Teorema 3.9. *Se “m” è finito e il LIQBD è ricorrente, allora le successioni $\{\tilde{U}_k : k \geq 1\}$ e $\{\tilde{G}_k : k \geq 1\}$ definite da:*

$$\begin{aligned} \tilde{G}_0 &= I, \\ \tilde{U}_k &= A_1 + A_0 \tilde{G}_{k-1}, \\ \tilde{G}_k &= (I - \tilde{U}_k)^{-1} A_2 \quad \text{per } k \geq 1, \end{aligned}$$

sono tale che \tilde{U}_k è sub-stocastica, \tilde{G}_k è stocastica per ogni “k” e $\lim_{k \rightarrow \infty} \tilde{U}_k = U$, $\lim_{k \rightarrow \infty} \tilde{G}_k = G$, inoltre, abbiamo che $\tilde{G}_k \geq G_k$ e $\tilde{U}_k \geq U_k$ per ogni k.

□

E per il caso continuo abbiamo il seguente teorema,

Teorema 3.10. *Se “m” è finito e il LIQBD è ricorrente, allora le successioni $\{\tilde{U}_k : k \geq 1\}$ e*

$\{\tilde{G}_k : k \geq 1\}$ definite da:

$$\begin{aligned}\tilde{U}_1 &= A_1 + A_0, \\ \tilde{G}_k &= (-\tilde{U}_k)^{-1}A_2, \\ \tilde{U}_{k+1} &= A_1 + A_0\tilde{G}_k \quad \text{per } k \geq 1,\end{aligned}$$

sono tale che \tilde{U}_k è sub-stocastica, \tilde{G}_k è stocastica per ogni “ k ” e $\lim_{k \rightarrow \infty} \tilde{U}_k = U$, $\lim_{k \rightarrow \infty} \tilde{G}_k = G$. \square

Osservazione:

- Il criterio di arresto si basa sul confronto tra due approssimazioni successive, ciò non è molto buono come avevamo discusso previamente, ma non possiamo usare lo stesso criterio di arresto dell’algoritmo U dato che ogni calcolo approssimato di G è stocastico.
- Le matrici G_k hanno una significativa interpretazione probabilistica: gli elementi di $(G_k)_{ij}$ rappresentano la probabilità che, partendo dallo stato $(i, 1)$, la catena eventualmente visiterà il livello “0”, e lo farà, visitando lo stato $(0, j)$ sotto il *taboo* dei livelli $k + 1$ ed in avanti. Così, quando si avvia l’algoritmo, la catena non può superare il livello “1”; e con ogni nuova iterazione, la catena può visitare un livello in più rispetto al precedente.
- La complessità degli algoritmi U e U^* , misurati dal numero di moltiplicazioni e divisioni in *floating point*, è uguale a $\frac{7}{3}I_U m^3 + O(m)$ e $\frac{7}{3}I_{U^*} m^3 + O(m)$, rispettivamente, dove I_U e I_{U^*} rappresentano il numero di iterazioni per ogni algoritmo. Esperimenti numerici riportati finora indicano che $I_U \geq I_{U^*}$, quindi l’algoritmo U^* generalmente è preferibile.

Logarithmic Reduction Algorithm

Una pratica standard per migliorare la velocità di convergenza è considerare la successione di Newton. Infatti, in [37] Latouche ha dimostrato che in pochissimi iterazioni si può raggiungere $\|G - \tilde{G}_k\|_\infty \leq \epsilon$ con una convergenza quadratica e monotona, purtroppo la complessità di ogni iterazione è $O(m^6)$ invece di $O(m^3)$, in modo che la successione di Newton è più lento rispetto ai due algoritmi già discussi. L’algoritmo che vedremo ora, proposto da *Guy Latouche* e *V. Ramaswami* in 1993 ([38]) noto come *Logarithmic Reduction Algorithm (RL)* converge quadraticamente con complessità $O(m^3)$ in ogni iterazione e ciò è basato su una rappresentazione diretta della matrice G .

Per calcolare G , si deve valutare la massa di probabilità di tutte le traiettorie che la catena deve seguire prima di visitare il livello “0” per prima volta. Se ciò non può essere fatto esattamente, si può scegliere un numero arbitrario $\epsilon > 0$ e definire $L(\epsilon) = \inf\{k : \|\mathbf{e} - G_k \mathbf{e}\|_\infty < \epsilon\}$, con l’obiettivo di semplicemente limitarsi alle traiettorie che rimangono tra i livelli “1” e $L(\epsilon)$. L’algoritmo U è lineare in L : ad ogni iterazione si aumenta in “1” il numero di livelli che la catena può visitare, e così $I_U = L(\epsilon) - 1$.

L’algoritmo di *(RL)*, procede moltiplicando per due il numero di livelli che la catena può visitare in ogni iterazione. Quindi, il numero di iterazioni sarà approssimativamente $\log_2 L(\epsilon)$, cioè, una riduzione logaritmica del numero di iterazioni sull’algoritmo U . Si dimostra che questa riduzione è ottenuta a un modesto costo aggiuntivo per ogni iterazione. L’algoritmo è presentato di seguito:

Algoritmo: RL

1. Inizializzare:

▶ Verificare se il *LIQBD* è a tempo discreto o continuo.

○ Caso discreto:

$$B_0 := (I - A_1)^{-1} * A_0;$$

$$B_2 := (I - A_1)^{-1} * A_2;$$

○ Caso continuo:

$$B_0 := (-A_1)^{-1} * A_0;$$

$$B_2 := (-A_1)^{-1} * A_2;$$

▶ $S := B_2$;

▶ $H := B_0$;

2. Fino alla convergenza:

Do:

▶ $A_1^\diamond := B_0 * B_2 + B_2 * B_0$;

▶ $A_0^\diamond := B_0^2$;

▶ $A_2^\diamond := B_2^2$;

▶ $B_0 := (I - A_1^\diamond)^{-1} * A_0^\diamond$;

▶ $B_2 := (I - A_1^\diamond)^{-1} * A_2^\diamond$;

▶ $S := S + H * B_2$;

▶ $H := H * B_0$;

▶ Verificare se: $\|e - Se\|_\infty \leq \epsilon$

○ SI:

$$G := S;$$

$$U := A_1 + A_0 * G;$$

break

○ NO: *continue*

EndDo

Osservazione:

- Il criterio di arresto viene scelto in base al supposto che il *LIQBD* è ricorrente, cioè, quando G è stocastico.
- La complessità dell'algoritmo *RL* è uguale a $\frac{25}{3}I_{RL}m^3 + O(m)$, dove I_{RL} rappresenta il numero di iterazioni.
- Numerosi esempi e risultati teorici indicano che l'algoritmo *RL* è numericamente più stabile e molto più veloce affrontando i problemi difficili.

Per vedere l'efficienza di questi algoritmi, esaminiamo due esempi:

Esempio 3.1. Si considera un processo LIQBD a tempo discreto definito per le matrici:

$$A_2 = \begin{pmatrix} 0.5 & 0 \\ 0.3 & 0.3 \end{pmatrix} \quad A_1 = \begin{pmatrix} 0.1 & 0.1 \\ 0 & 0.1 \end{pmatrix} \quad A_1 = \begin{pmatrix} 0.2 & 0.1 \\ 0.3 & 0 \end{pmatrix}$$

con matrice $R = \begin{pmatrix} 0.38162976 & 0.18197520 \\ 0.49491846 & 0.08756795 \end{pmatrix}$

Applicando gli algoritmi descritti in questa sezione per calcolare R , otteniamo la tabella (3.1), che mostra gli errori $E = \|R_{esatto} - R_{iter}\|_2$ nella soluzione approssimata, R_{iter} , calcolati in ogni iterazione per ciascuno di questi cinque metodi.

Iter.	$E = \ R_{esatto} - R_{iter}\ _2$				
	Alg. Classico	Alg. N	Alg. U	Alg. U*	Alg. RL
1	2.3706×10^{-1}	7.2450×10^{-2}	1.1323×10^{-1}	4.5181×10^{-3}	3.1903×10^{-2}
2	1.4204×10^{-1}	4.8855×10^{-2}	5.8877×10^{-2}	7.4442×10^{-4}	3.1604×10^{-3}
3	9.2186×10^{-2}	3.3818×10^{-2}	3.1903×10^{-2}	1.1943×10^{-4}	3.4406×10^{-5}
4	6.2328×10^{-2}	2.3773×10^{-2}	1.7667×10^{-2}	1.9252×10^{-5}	4.1285×10^{-9}
5	4.3149×10^{-2}	1.6877×10^{-2}	9.8990×10^{-3}	3.1010×10^{-6}	0
6	3.0322×10^{-2}	1.2061×10^{-2}	5.5830×10^{-3}	4.9956×10^{-7}	0
7	2.1519×10^{-2}	8.6585×10^{-3}	3.1604×10^{-3}	8.0475×10^{-8}	0
8	1.5375×10^{-2}	6.2354×10^{-3}	1.7927×10^{-3}	1.2964×10^{-8}	0
9	1.1037×10^{-2}	4.5004×10^{-3}	1.0181×10^{-3}	2.0884×10^{-9}	0
10	7.9485×10^{-2}	3.2534×10^{-3}	5.7858×10^{-4}	3.3643×10^{-10}	0

Tabella 3.1: Errori calcolando R - Caso discreto.

Esempio 3.2. Si considera un processo LIQBD a tempo continuo definito per le matrici:

$$A_2 = \begin{pmatrix} 0 & 9 & 9 \\ 3 & 2 & 7 \\ 0 & 8 & 0 \end{pmatrix} \quad A_1 = \begin{pmatrix} -38 & 9 & 8 \\ 6 & -37 & 6 \\ 8 & 4 & -35 \end{pmatrix} \quad A_1 = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 10 & 3 \\ 9 & 4 & 2 \end{pmatrix}$$

con matrice $R = \begin{pmatrix} 0.1042582691 & 0.0558990230 & 0.0565703584 \\ 0.1992329111 & 0.5290439542 & 0.3831600188 \\ 0.4061312811 & 0.3986441106 & 0.3632384517 \end{pmatrix}$

I risultati degli errori sono riportati nella tabella (3.2).

Le figure (3.1) e (3.2) mostrano la velocità di convergenza per le cinque algoritmi, nel caso discreto e nel caso continuo rispettivamente.

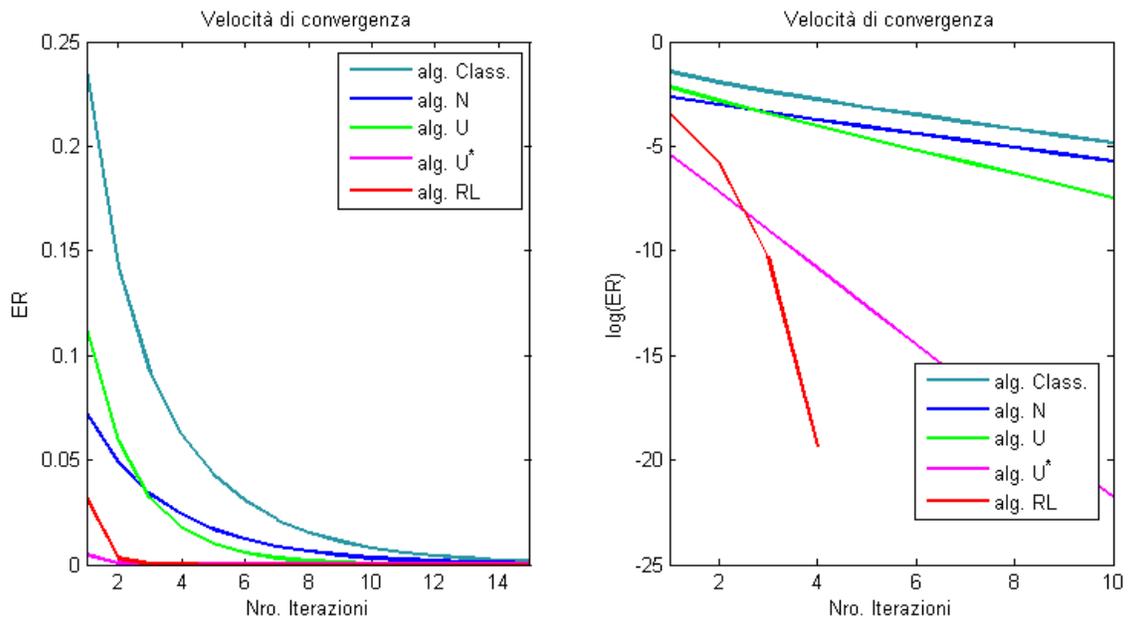


Figure 3.1: Residui ottenuti nel calcolo di R - Caso discreto.

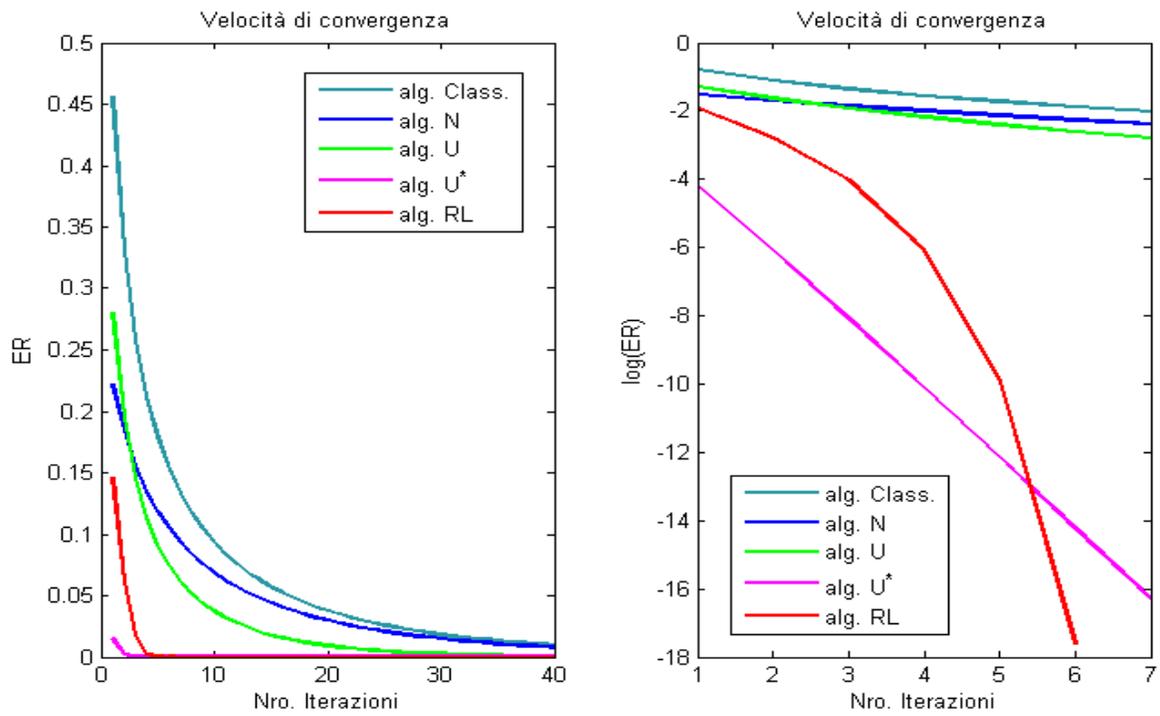


Figure 3.2: Residui ottenuti nel calcolo di R - Caso continuo.

$E = \ R_{esatto} - R_{iter}\ _2$					
<i>Iter.</i>	<i>Alg. Classico</i>	<i>Alg. N</i>	<i>Alg. U</i>	<i>Alg. U*</i>	<i>Alg. RL</i>
1	4.5668×10^{-1}	2.2203×10^{-1}	2.8057×10^{-1}	1.5057×10^{-2}	1.4606×10^{-1}
2	3.3057×10^{-1}	1.8377×10^{-1}	1.9580×10^{-1}	2.2350×10^{-3}	6.1283×10^{-2}
3	2.5927×10^{-1}	1.5611×10^{-1}	1.4606×10^{-1}	3.1193×10^{-4}	1.7758×10^{-2}
4	2.1227×10^{-1}	1.3497×10^{-1}	1.1355×10^{-1}	4.1188×10^{-5}	2.3114×10^{-3}
5	1.7860×10^{-1}	1.1821×10^{-1}	9.0766×10^{-2}	5.2816×10^{-6}	4.9024×10^{-5}
6	1.5316×10^{-1}	1.0456×10^{-1}	7.4025×10^{-2}	6.6740×10^{-7}	2.2904×10^{-8}
7	1.3322×10^{-1}	9.3228×10^{-2}	6.1283×10^{-2}	8.3717×10^{-8}	0
8	1.1715×10^{-1}	8.3660×10^{-2}	5.1324×10^{-2}	1.0463×10^{-8}	0
9	1.0393×10^{-1}	7.5480×10^{-2}	4.3377×10^{-2}	1.3052×10^{-9}	0
10	9.2868×10^{-2}	6.8410×10^{-2}	3.6931×10^{-2}	1.6266×10^{-10}	0

Tabella 3.2: Errori calcolando R - Caso continuo.

3.3 Level Dependent Quasi-Birth-and-Death Processes - (LDQBD)

Come abbiamo già accennato, il processo $LDQBD$ è un processo di Markov bivariato dove la sua matrice di probabilità di transizioni presenta una struttura tridiagonale a blocchi. Nonostante ciò, a differenza del LIQBD, le sue transizioni sono esplicitamente dipendenti dal livello.

3.3.1 LDQBD a tempo discreto

Definizione 3.3. *Un level-dependent QBD a tempo discreto, è un processo di Markov $\{\Phi_t = (N_t, \phi_t) : t \in \mathbb{N}_0\}$ sullo spazio degli stati bidimensionale $S = \{(n, j) : n \in \mathbb{N}_0, j = 1, 2, \dots, m_n\}$ con matrice di transizione della forma:*

$$P = \begin{pmatrix} A_1^{(0)} & A_0^{(0)} & 0 & 0 & \cdots \\ A_2^{(1)} & A_1^{(1)} & A_0^{(1)} & 0 & \cdots \\ 0 & A_2^{(2)} & A_1^{(2)} & A_0^{(2)} & \cdots \\ 0 & 0 & A_2^{(3)} & A_1^{(3)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (3.3.1)$$

Dove la matrice $A_2^{(n)}$ è di dimensione $m_n \times m_{n-1}$, $A_1^{(n)}$ è di dimensione $m_n \times m_n$, $A_0^{(n)}$ è di dimensione $m_n \times m_{n+1}$ e $A_0^{(0)}$, $A_1^{(0)}$, $A_0^{(1)}$, $A_1^{(1)}$, $A_2^{(n)}$ sono non negative, inoltre, la somma degli elementi nelle righe di P devono essere uguale all'unità, cioè,

$$A_0^{(n)} \mathbf{e} + A_1^{(n)} \mathbf{e} + A_2^{(n)} \mathbf{e} = \mathbf{e}, \quad \forall n \geq 1$$

e

$$A_0^{(0)} \mathbf{e} + A_1^{(0)} \mathbf{e} = \mathbf{e}.$$

ove ogni \mathbf{e} ha una dimensione appropriata.

Osservazione:

- Si osserva nella definizione che il numero di fasi in ogni livello dipende di quel livello, quindi le matrici $A_1^{(n)}$ sono quadrate mentre che l'altre possono essere rettangolari.
- Ogni livello è rappresentato come $\ell(n) = \{(n, 1), (n, 2), \dots, (n, m_n)\}$ per ogni $n \geq 0$, quindi $S = \bigcup_{n \geq 0} \ell(n)$.
- Ai fini di questa tesi, si assume che “ m_n ” è finito e che P è irriducibile.

Per interpretare gli elementi della matrice P , notare che:

Quando $n \geq 1$, le uniche possibili transizioni di stato sono:

- ▷ Dallo stato (n, j) allo stato $(n - 1, k)$, con probabilità $(A_2^{(n)})_{jk}$, $\forall j, k = 1, 2, \dots, m_n$.
- ▷ Dallo stato (n, j) allo stato (n, k) , con probabilità $(A_1^{(n)})_{jk}$, $\forall j, k = 1, 2, \dots, m_n$.
- ▷ Dallo stato (n, j) allo stato $(n + 1, k)$, con probabilità $(A_0^{(n)})_{jk}$, $\forall j, k = 1, 2, \dots, m_n$.

Quando $n = 0$, le uniche possibili transizioni di stato sono:

- ▷ Dallo stato $(0, j)$ allo stato $(0, k)$, con probabilità $(A_1^{(0)})_{jk}$, $\forall j, k = 1, 2, \dots, m_0$.
- ▷ Dallo stato $(0, j)$ allo stato $(1, k)$, con probabilità $(A_0^{(0)})_{jk}$, $\forall j, k = 1, 2, \dots, m_0$.

Analogamente al caso LIQBD, vogliamo sapere se il processo di Markov con matrice di transizione P possiede una distribuzione limite, e in caso affermativo, in quali condizioni esiste tale distribuzione e qual è la sua forma. Purtroppo, l'estrema generalità del processo LDQBD impedisce lo sviluppo di strumenti generici che possono essere utilizzati per stabilire proprietà come la irriducibilità, ricorrenza e positiva ricorrenza; tuttavia, nella pratica, è spesso possibile stabilire queste proprietà in applicazioni specifiche utilizzando gli attributi del modello. Gli argomenti di dominanza stocastica sono spesso impiegati per dimostrare che il processo stocastico LDQBD è dominato da un processo più semplice (per esempio, un processo QBD di tipo omogenea) in cui le proprietà di irriducibilità e ricorrenza possono essere più facile di stabilire. L'obiettivo qui è quello di rivedere le condizioni teoriche per la positiva ricorrenza dei processi LDQBD irriducibili.

Sia il vettore $\pi > 0$ la distribuzione limite di $\{\Phi_t : t \in \mathbb{N}_0\}$ che risolve univocamente il sistema

$$\begin{cases} \pi P &= \pi \\ \pi \mathbf{e} &= 1 \end{cases}$$

Il vettore π è partizionato in sottovettori in modo che

$$\pi = (\pi_0, \pi_1, \dots),$$

ove π_n contiene le probabilità limite per gli stati nel livello “ n ”, i.e., negli stati del insieme $\ell(n) = \{(n, 1), (n, 2), \dots, (n, m_n)\}$.

Nel capitolo 6 di [39] si dimostra che il vettore π_n assume una forma di tipo *matrice-geometrica* quando il processo è level-independent, cioè, quando per $i = 0, 1, 2$; le matrici $A_i^{(n)} = A_i$ per ogni $n \geq 0$. Più specificamente, per $n \geq 0$, il vettore π_n soddisfa

$$\pi_n = \pi_0 R^n,$$

dove R è una matrice fissa (indipendente dal livello) tale che per ogni coppia di fasi j e k , $(R)_{jk}$ è il numero atteso di visite allo stato $(1, k)$ prima di ritornare all'insieme $\ell(0)$, a condizione che il processo inizia nello stato $(0, j)$.

Prima di dare le condizioni necessarie e sufficienti per la positiva ricorrenza di un processo LDQBD, un primo passo sarà assumere che sia irriducibile, aperiodica, e positiva ricorrente al fine di caratterizzare la distribuzione limite π .

Teorema 3.11. *Se il processo LDQBD con matrice di probabilità di transizioni data da (3.3.1) è irriducibile, aperiodica e positiva ricorrente, allora, esistono matrici $\{R_n : n \geq 1\}$, tali che*

$$\pi_n = \pi_{n-1} R_n, \quad n \geq 1, \quad (3.3.2)$$

ove la matrice R_n registra il numero atteso di visite all'insieme $\ell(n+1)$ tra due visite all'insieme $\ell(n)$, e la sequenza $\{R_n\}$ è la soluzione non negativa minima del sistema di equazioni

$$R_n = A_0^{(n-1)} + R_n A_1^{(n)} + R_n R_{n+1} A_2^{(n+1)}, \quad n \geq 1.$$

□

Più precisamente, $(R_n)_{jk}$ è uguale al numero atteso di visite allo stato $(n+1, k)$ prima di un ritorno all'insieme $\ell_0 \cup \ell(1) \cup \dots \cup \ell(n)$, a condizione che il processo inizia nello stato (n, j) . Notare che il numero atteso di visite dipende esplicitamente del livello “ n ” da cui si parte. Per determinare le condizioni per la positiva ricorrenza, di seguito discuteremo i tempi di primo passaggio di $\{\Phi_t : t \in \mathbb{N}_0\}$ e introdurremo alcune matrici importanti che sono molto simili a quelli tipicamente utilizzati per caratterizzare il comportamento dei processi level-independent QBD.

Supponiamo che $\{\Phi_t : t \in \mathbb{N}_0\}$ inizia in qualche stato (n, j) , cioè, nel livello $n > 0$ e la fase “ j ” al tempo “0”, e definiamo S_1 come il primo tempo di ritorno al livello “ n ”. Allora,

$$S_1 = \inf\{t \geq 1 : \Phi_t \in \ell(n) | \Phi_0 = (n, j)\}$$

Inoltre, adottando la notazione in [39], τ denota il primo tempo di passaggio di $\{\Phi_t : t \in \mathbb{N}_0\}$ al livello $n-1$. Definiamo ora le seguenti due probabilità:

$$(U_n)_{jk} = \mathbb{P}\{S_1 < \tau, \Phi_{S_1} = (n, k) | \Phi_0 = (n, j)\}, \quad (3.3.3)$$

$$(G_n)_{jk} = \mathbb{P}\{\tau < \infty, \Phi_\tau = (n-1, k) | \Phi_0 = (n, j)\}, \quad (3.3.4)$$

Note che questi due equazioni hanno una sorprendente somiglianza con le equazioni (3.1.4) e (3.1.5) con un'importante eccezione: le probabilità sopra dipendono esplicitamente dal livello “ n ” mentre loro sono completamente indipendente da “ n ” quando il QBD è level-independent. Vedremo che questa dipendenza complica notevolmente il processo di calcolo della distribuzione limite di un processo LDQBD.

Il “ j, k ” –esimo elemento della matrice U_n è la probabilità che, iniziando nel livello “ n ”, il processo ritorna al livello “ n ” prima di colpire il livello “ $n-1$ ”, e il “ j, k ” –esimo elemento G_n indica la probabilità che il processo colpisce il livello “ $n-1$ ” in un periodo limitato di tempo, dato che è iniziato da qualche parte del livello “ n ”. Non è sorprendente che le sequenze di matrici, $\{R_n\}$, $\{G_n\}$ e $\{U_n\}$ sono correlati. Infatti, una qualsiasi delle sequenze determina gli altri due.

Teorema 3.12. *Una qualsiasi delle sequenze $\{R_n\}$, $\{G_n\}$ e $\{U_n\}$ determina gli altri due tramite le relazioni:*

$$\begin{aligned} G_n &= (I - U_n)^{-1} A_2^{(n)}, \\ R_n &= A_0^{(n-1)} (I - U_n)^{-1}, \\ U_n &= A_1^{(n)} + A_0^{(n)} G_{n+1}, \\ U_n &= A_1^{(n)} + R_{n+1} A_2^{(n)}. \end{aligned}$$

□

Note che le stesse relazioni valgono per R , G , e U nel caso level-independent in cui nessuna delle matrici dipende dal livello “ n ”. Per scopi di calcolo, il seguente teorema mostra le relazioni esplicite tra queste matrici

Teorema 3.13. *Per ogni $n \geq 1$, le matrici R_n , G_n , e U_n verificano il seguente sistema di equazioni:*

$$\begin{aligned} G_n &= A_2^{(n)} + A_1^{(n)} G_n + A_0^{(n)} G_{n+1} G_n, \\ R_n &= A_0^{(n-1)} + R_n A_1^{(n)} + R_n R_{n+1} A_2^{(n+1)}, \\ U_n &= A_1^{(n)} + A_0^{(n)} (I - U_{n+1})^{-1} A_2^{(n+1)}. \end{aligned}$$

□

Alla luce di queste due teoremi e l’equazione (3.3.2), è evidente che la condizione di positiva ricorrenza può essere riscritta in termini della matrice G_1 . Il seguente teorema fornisce la condizione necessaria e sufficiente che assicura la positiva ricorrenza di P .

Teorema 3.14. *Il processo LDQBD è positivo ricorrente se e solo se esiste una soluzione strettamente positiva per il sistema di equazioni*

$$\pi_0 = \pi_0 (A_1^{(0)} + A_0^{(0)} G_1) \quad (3.3.5)$$

soggetto alla condizione di normalizzazione

$$\pi_0 \left(\sum_{k=0}^{\infty} \prod_{n=1}^k R_n \right) e = 1, \quad (3.3.6)$$

dove il prodotto vuoto (quando $n = 0$) risulta nella matrice identità.

□

Dobbiamo rimarcare che la matrice G_1 nella pratica solo può essere ottenuta approssimativamente; pertanto, è difficile affermare la positiva ricorrenza usando (3.3.5) e (3.3.6). Tuttavia, l’equazione matriciale (3.3.5) può essere sostituito da

$$\pi_0 = \pi_0 (A_1^{(0)} + R_1 A_2^{(1)}). \quad (3.3.7)$$

Si noti che, se $m = \infty$, allora abbiamo anche bisogno che la *embedded Markov chain* di $\ell(0)$ sia positivo ricorrente. Tale condizione è ovviamente soddisfatta quando $m < \infty$. Prendendo insieme l’equazioni (3.3.2), (3.3.6) e (3.3.7) forniscono la distribuzione limite π . È ovvio che dobbiamo calcolare la famiglia di matrici $\{R_n : n \geq 1\}$ per ottenere questa distribuzione limite. Al contrario, abbiamo bisogno solo di calcolare la matrice R (o G o U) nel caso level-independent per caratterizzare completamente π .

3.3.2 LDQBD a tempo continuo

Definizione 3.4. *Un level-dependent QBD a tempo continuo, è un processo di Markov $\{\Phi_t = (N_t, \phi_t) : t \geq 0\}$ sullo spazio degli stati bidimensionale $S = \{(n, j) : n \in \mathbb{N}_0, j = 1, 2, \dots, m_n\}$ con matrice generatrice infinitesimale della forma:*

$$Q = \begin{pmatrix} A_1^{(0)} & A_0^{(0)} & 0 & 0 & \cdots \\ A_2^{(1)} & A_1^{(1)} & A_0^{(1)} & 0 & \cdots \\ 0 & A_2^{(2)} & A_1^{(2)} & A_0^{(2)} & \cdots \\ 0 & 0 & A_2^{(3)} & A_1^{(3)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (3.3.8)$$

dove la matrice $A_2^{(n)}$ è di dimensione $m_n \times m_{n-1}$, $A_1^{(n)}$ è di dimensione $m_n \times m_n$, $A_0^{(n)}$ è di dimensione $m_n \times m_{n+1}$, le matrici $A_0^{(0)}$, $A_0^{(n)}$, $A_2^{(n)}$ sono non negative. Inoltre, per ogni $n \geq 0$: $A_1^{(n)}(i, j) \geq 0$ per $1 \leq i \neq j \leq m_n$ e $A_1^{(n)}(i, i) < 0$ per $1 \leq i \leq m_n$, e la somma degli elementi nelle righe di Q devono essere uguale a zero, cioè

$$A_0^{(n)} \mathbf{e} + A_1^{(n)} \mathbf{e} + A_2^{(n)} \mathbf{e} = \mathbf{0}, \quad \forall n \geq 1$$

e

$$A_0^{(0)} \mathbf{e} + A_1^{(0)} \mathbf{e} = \mathbf{0}.$$

ove ogni \mathbf{e} ha una dimensione appropriata.

Le stesse osservazioni fatte per il caso discreto vengono anche indicate per il caso continuo, dove in questo caso cambiamo la matrice P per il generatore infinitesimale Q il quale sarà irriducibile. Per interpretare gli elementi della matrice Q , notare che:

Quando $n \geq 1$, le uniche possibili transizioni di stato sono:

- ▶ Dallo stato (n, j) allo stato $(n-1, k)$, con tasso $(A_2^{(n)})_{jk}$, $\forall j, k = 1, 2, \dots, m_n$.
- ▶ Dallo stato (n, j) allo stato (n, k) , con tasso $(A_1^{(n)})_{jk}$, $\forall j, k = 1, 2, \dots, m_n$.
- ▶ Dallo stato (n, j) allo stato $(n+1, k)$, con tasso $(A_0^{(n)})_{jk}$, $\forall j, k = 1, 2, \dots, m_n$.

Quando $n = 0$, le uniche possibili transizioni di stato sono:

- ▶ Dallo stato $(0, j)$ allo stato $(0, k)$, con tasso $(A_1^{(0)})_{jk}$, $\forall j, k = 1, 2, \dots, m_0$.
- ▶ Dallo stato $(0, j)$ allo stato $(1, k)$, con tasso $(A_0^{(0)})_{jk}$, $\forall j, k = 1, 2, \dots, m_0$.

Se il processo ha una distribuzione limite $\pi = (\pi_0, \pi_1, \dots)$, allora deve soddisfare il sistema di equazioni

$$\begin{cases} \pi Q = \mathbf{0} \\ \pi \mathbf{e} = 1 \end{cases}$$

Come ci si potrebbe aspettare, la distribuzione limite della versione a tempo continuo è analoga alla sua controparte a tempo discreto. Vedremo ora la condizione necessaria e sufficiente che

assicura la positiva ricorrenza di Q e forniremo un'espressione per la sua distribuzione limite. Il seguente risultato principale mostra che la distribuzione limite ristretto a uno qualsiasi dei livelli dipende esplicitamente da π_0 .

Teorema 3.15. *Il processo LDQBD con matrice generatrice infinitesimale data da (3.3.8) è positivo ricorrente se e solo se esiste una soluzione strettamente positiva per il sistema di equazioni*

$$\pi_0(A_1^{(0)} + R_0A_2^{(1)}) = \mathbf{0}, \quad (3.3.9)$$

soggetto alla condizione di normalizzazione

$$\pi_0 \left(\sum_{n=0}^{\infty} \prod_{k=0}^{n-1} R_k \right) \mathbf{e} = 1, \quad (3.3.10)$$

In tal caso, il vettore riga π_n di ordine “ m ” è dato da

$$\pi_n = \pi_0 \prod_{k=0}^{n-1} R_k, \quad n \geq 0. \quad (3.3.11)$$

dove il prodotto vuoto (quando $n = 0$) risulta nella matrice identità. □

Analogamente alla versione a tempo discreto, la distribuzione limite è caratterizzato da una sequenza di matrici level-dependent $\{R_n : n \geq 0\}$ che, in generale, può essere determinato solo numericamente. Gli elementi di queste matrici possono essere interpretati come segue. Il “ j, k ” –esimo elemento di R_n , denotata da $(R_n)_{jk}$, è il tempo di permanenza previsto (*expected sojourn time*) nello stato $(n+1, k)$ per unità di “sojourn” nello stato (n, j) , dato che il processo è iniziato nello stato (n, j) . È noto che la sequenza $\{R_n\}$ è la soluzione non negativa minima del sistema di equazioni

$$A_0^{(n)} + R_n A_1^{(n+1)} + R_n (R_{n+1} A_2^{(n+2)}) = 0, \quad n \geq 0. \quad (3.3.12)$$

Se il numero di livelli e/o fasi è infinito, il calcolo della distribuzione limite ha due principali complicanze. In primo luogo, si deve troncatura la serie infinita dell'equazione (3.3.10) in modo appropriato, e in secondo luogo, le matrici $\{R_n\}$ devono essere calcolati in modo efficiente. La sezione successiva fornisce una breve panoramica di alcuni degli approcci algoritmici per il calcolo della distribuzione limite dei processi LDQBD a tempo discreto e a tempo continuo.

3.4 Algoritmi per LDQBD

Nella sezione (3.2) abbiamo visto diversi algoritmi per calcolare la distribuzione stazionaria dei processi LIQBD, tra di loro forse il metodo di *riduzione logaritmica* è il più notevole. Nel caso dei processi LDQBD attualmente ci sono pochi approcci notevoli; quando il numero di fasi e il numero di livelli del LDQBD è finito, ci sono alcuni lavori, ad esempio: Gaver e al. [18] hanno fornito un algoritmo stabile per calcolare la distribuzione limite ed i momenti del primo tempo di passaggio nel caso LDQBD a tempo continuo, Ye - Li [59] hanno sviluppato l'algoritmo “*Folding algorithm*” per calcolare la distribuzione limite, questo algoritmo risolve direttamente $\pi Q = 0$ utilizzando tecniche di riduzione spazio-stato, il che riduce la complessità indotta dalla dipendenza del livello.

Quando il numero di livelli e/o fasi è infinito, il metodo ampiamente accettato per calcolare la distribuzione limite è dovuto a Bright-Taylor [9], che fornisce un'estensione dell'algoritmo RL di Latouche e Ramaswami [38] per il caso *level-dependent*. Vediamo un po' questo'ultimo approccio.

Bright e Taylor [9] suggeriscono di troncatura la serie infinita dell'equazione (3.3.10) ad un certo livello K^* e poi renormalizzare per calcolare un subvettore approssimato della forma

$$x_k(K^*) = x_0(K^*) \prod_{m=0}^{k-1} R_m \quad (3.4.1)$$

con la condizione che $x_0(K^*)$ soddisfa (3.3.9) e

$$x_0(K^*) \sum_{k=0}^{K^*} \left[\prod_{m=0}^{k-1} R_m \right] \mathbf{e} = 1. \quad (3.4.2)$$

dove $(x_k(K^*))_j$, $0 \leq k \leq K^*$ è la probabilità stazionaria che Φ_t si trovi nello stato (k, j) condizionata al fatto che Φ_t sia nello insieme $\{(i, j) | 0 \leq i \leq K^*, 1 \leq j \leq m_i\}$.

Si noti che non è stato troncato Φ_t a uno spazio di stati finiti, anzi, si ha calcolato una misura invariante per la distribuzione di equilibrio per tutti gli stati corrispondenti ai livelli minori e pari a K^* , quindi ha senso la normalizzazione di questa misura invariante su questi stati. È ovvio che per ogni $K^* \geq 0$, $x_k(K^*)$ è un limite superiore per x_k . Quindi abbiamo

$$x_k \leq x_k(K^*) \quad \forall K^* \geq 0,$$

e

$$x_k = \lim_{K^* \rightarrow \infty} x_k(K^*) \quad \forall K^* \geq 0,$$

dove la disuguaglianza e il limite valgono "element-wise". Se prendiamo una K^* abbastanza grande, si dovrebbe avere $x_k \approx x_k(K^*)$. L'algoritmo presentato di seguito, permette di valutare $x_k(K^*)$ per un dato valore K^* , ma prima è necessario un lemma.

Lemma 3.4. *Se Φ_t è positivo ricorrente, allora R_k è dato da*

$$R_k = \sum_{\ell}^{\infty} U_k^{\ell} \prod_{i=0}^{\ell-1} D_{k+2^{\ell-i}}^{\ell-1-i}, \quad k \geq 0, \quad (3.4.3)$$

dove U_k^{ℓ} e D_k^{ℓ} sono matrici di dimensioni $m_{k-1} \times m_{k-1+2^{\ell}}$ e $m_{k-1} \times m_{k-1+2^{\ell}}$ rispettivamente e sono definiti ricorsivamente da

$$U_k^0 = A_0^{(k)} (-A_1^{(k+1)})^{-1}, \quad k \geq 1,$$

$$D_k^0 = A_2^{(k)} (-A_1^{(k-1)})^{-1}, \quad k \geq 1,$$

$$U_k^{\ell+1} = U_k^{\ell} U_{k+2^{\ell}} \left[I - U_{k+2^{\ell+1}}^{\ell} D_{k+3 \cdot 2^{\ell}}^{\ell} - D_{k+2^{\ell+1}}^{\ell} U_{k+2^{\ell}}^{\ell} \right]^{-1},$$

$$D_k^{\ell+1} = D_k^{\ell} D_{k-2^{\ell}} \left[I - U_{k-2^{\ell+1}}^{\ell} D_{k-2^{\ell}}^{\ell} - D_{k-2^{\ell+1}}^{\ell} U_{k-3 \cdot 2^{\ell}}^{\ell} \right]^{-1}.$$

Quando il Lemma (3.4) viene utilizzato per calcolare R_k , è necessario troncare la somma infinita in (3.4.3). Un modo semplice per farlo è di definire $R_k(n)$ come la somma dei primi “ $n + 1$ ” termini in (3.4.3) e poi prendere $R_k = R_k(L)$ tale che $[R_k(L) - R_k(L - 1)]_{max} < \epsilon$ per alcuna tolleranza ϵ , dove $[M]_{max}$ rappresenta il massimo di tutti gli elementi della matrice M . Per calcolare $\{R_k, 0 \leq k \leq K^*\}$, non è necessario usare l’equazione (3.4.3) ripetutamente. Riordinando l’equazione (3.3.12) abbiamo

$$R_k = A_0^{(k)}(-A_1^{(k+1)} - R_{k+1}A_2^{(k+2)})^{-1}, \quad (3.4.4)$$

sempre che l’inversa esista, abbiamo quindi il seguente algoritmo:

Algoritmo: Calcolo di $\{x_k(K^*), 0 \leq k \leq K^*\}$ dato K^*

- Calcolare R_{K^*-1} usando l’equazione (3.4.3).
- Calcolare ricorsivamente $R_{K^*-2}, R_{K^*-3}, \dots, R_0$ usando l’equazione (3.4.4).
- Risolvere il sistema di equazioni:

$$\begin{aligned} x_0(K^*) \left[A_1^{(0)} + R_0 A_2^{(1)} \right] &= 0, \\ x_0(K^*) \mathbf{e} &= 1. \end{aligned}$$

- Per ogni $i = 1$ fino a K^* , fare

$$x_k(K^*) = x_{k-1}(K^*)R_{k-1},$$

e poi normalizzare $x_n(K^*)$, $n = 0, 1, \dots, k$, tale che

$$\sum_{n=0}^k x_n(K^*) \mathbf{e} = 1.$$

È imperativo selezionare un intero K^* lo sufficientemente grande per assicurare che la probabilità limite di essere in uno stato dentro o sopra del livello K^* sia chiuso a zero. In [9], l’algoritmo 4 descrive un metodo generale per la scelta di un punto di troncamento che funziona bene in pratica e l’algoritmo 2 permette calcolare i valori di R_k in modo più efficiente. Il passo critico di questi algoritmi è quello di trovare un valore sufficientemente grande di K^* . Attualmente sono in corso ulteriori ricerche su metodi più elaborati per la determinazione di K^* e anche metodi per determinare la positiva ricorrenza.

Capitolo 4

Markovian Binary Trees

In questa sezione definiremo e costruiremo *Markovian Binary Trees* come un esempio particolare di un *QBD level-dependent* che è stato discusso nel Capitolo 3. Tali processi furono definiti per la prima volta in ([6],[35]) e utilizzati per modellare i processi evolutivi.

Definizione 4.1. *Un Markovian Binary Tree è un processo QBD level-dependent a tempo continuo $\{\Psi(t) : t \geq 0\}$ con lo spazio degli stati $\Psi(t) = (N(t), \phi_1(t), \dots, \phi_{N(t)}(t))$ definite su $\cup_{k=0}^{\infty} \{k\} \times \{1, \dots, n\}^k$ e con matrice generatrice infinitesimale della forma*

$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ A_2^{(1)} & A_1^{(1)} & A_0^{(1)} & 0 & 0 & 0 & \dots \\ 0 & A_2^{(2)} & A_1^{(2)} & A_0^{(2)} & 0 & 0 & \ddots \\ 0 & 0 & A_2^{(3)} & A_1^{(3)} & A_0^{(3)} & 0 & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \quad (4.0.1)$$

La variabile aleatoria $N(t)$ indica il numero dei rami viventi al tempo t e le variabili aleatorie $\phi_k(t) \in \{1, \dots, n\}$ per ogni $k = 1, 2, \dots, N(t)$ indicano le fasi del k -esimo ramo al tempo t .

Il processo fase in ciascuno dei rami si evolve come il processo fase di una MAP con “ n ” fasi (τ, D_0, D_1, d) , come è stato descritto nella sezione (2.2.3), con $d_0 = d$, $d_1 = \mathbf{0}$, cioè

$$D_0^* = \begin{pmatrix} 0 & \mathbf{0} \\ d & D_0 \end{pmatrix} \quad e \quad D_1^* = \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{0} & D_1 \end{pmatrix}$$

con $D_1 \geq 0$, $d \geq \mathbf{0}$ e $D_0(i, j) \geq 0$ per $1 \leq i \neq j \leq n$, $D_0(i, i) < 0$ per $1 \leq i \leq n$, inoltre

$$D_0 \mathbf{e} + D_1 \mathbf{e} + d = \mathbf{0}. \quad (4.0.2)$$

Lo spazio degli stati del processo può essere partizionato in livelli, proprio come nei QBDs. Lo spazio degli stati nel livello $k \in \{0\} \cup \mathbb{Z}_+$, denotato con $\ell(k)$ sono quegli stati che consistono dei “ k ” rami che si muovono attivamente con n^k possibili fasi per l’intero MBT, cioè,

$$\ell(k) = \{(k, \phi_1, \dots, \phi_k) | (\phi_1, \dots, \phi_k) \in \{1, 2, \dots, n\}^k\}.$$

Più specificamente:

$$\begin{aligned}
\triangleright \ell(0) &= \{(0)\} && \Rightarrow n^0 \text{ possibili stati} \\
\triangleright \ell(1) &= \{(1, 1); (1, 2); \dots; (1, n)\} && \Rightarrow n^1 \text{ possibili stati} \\
\triangleright \ell(2) &= \left\{ \begin{array}{l} (2, 1, 1); (2, 1, 2); \dots; (2, 1, n) \\ (2, 2, 1); (2, 2, 2); \dots; (2, 2, n) \\ \vdots \\ (2, n, 1); (2, n, 2); \dots; (2, n, n) \end{array} \right\} && \Rightarrow n^2 \text{ possibili stati} \\
&\vdots && \vdots \\
\triangleright \ell(k) &= \text{così via} && \Rightarrow n^k \text{ possibili stati}
\end{aligned}$$

Di seguito esporremo e spiegheremo la matrice Q del processo e poi discuteremo il suo comportamento qualitativo, ma prima definiremo le *operazioni Kronecker* sullo spazio delle matrici.

Definizione 4.2 (Prodotto di Kronecker). *Siano le matrici Z di dimensione $a \times b$ e Y di dimensione $c \times d$, allora il prodotto di Kronecker di Z e Y , denotato con $Z \otimes Y$ è un'altra matrice di dimensione $ac \times bd$ data da*

$$Z \otimes Y = \begin{pmatrix} z_{11}Y & z_{12}Y & \cdots & z_{1b}Y \\ z_{21}Y & z_{22}Y & \cdots & z_{2b}Y \\ \vdots & \vdots & \ddots & \vdots \\ z_{a1}Y & z_{a2}Y & \cdots & z_{ab}Y \end{pmatrix}_{ac \times bd} \quad (4.0.3)$$

Definizione 4.3 (Somma di Kronecker). *Siano le matrici Z e Y di dimensioni $a \times a$ e $b \times b$, allora la somma di Kronecker di Z e Y , denotato con $Z \oplus Y$ è un'altra matrice di dimensione $ab \times ab$ data da*

$$Z \oplus Y = Z \otimes I_b + I_a \otimes Y, \quad (4.0.4)$$

dove I_n è la matrice identità di dimensione $n \times n$. □

Analizziamo ora la matrice Q . Sia $I^{(k)}$ la matrice identità di dimensione $n^k \times n^k$, con $I^{(0)} = 1$. Allora possiamo descrivere le matrici $A_0^{(k)}$, $A_1^{(k)}$ e $A_2^{(k)}$ che compongono Q come segue:

- La matrice $A_2^{(k)}$ di dimensione $n^k \times n^{k-1}$ per ogni $k \geq 1$ è data da

$$A_2^{(k)} = \sum_{j=0}^{k-1} I^{(j)} \otimes \mathbf{d} \otimes I^{(k-1-j)}. \quad (4.0.5)$$

Questa equazione incarna il fatto che solo un ramo può estinguersi in qualsiasi momento del tempo.

- La matrice $A_1^{(k)}$ di dimensione $n^k \times n^k$ per ogni $k \geq 1$ soddisfa la ricorsione

$$A_1^{(k)} = A_1^{(k-1)} \oplus D_0, \quad (4.0.6)$$

con $A_1^{(0)} = 0$. Questa equazione indica che non esiste alcuna interazione tra qualsiasi dei 'k' processi-fase che sono attualmente in evoluzione.

- Infine, la matrice $A_0^{(k)}$ di dimensione $n^k \times n^{k+1}$ per ogni $k \geq 1$ è data da

$$A_0^{(k)} = \sum_{j=0}^{k-1} I^{(j)} \otimes B \otimes I^{(k-1-j)}, \quad (4.0.7)$$

ove la matrice B di dimensione $n \times n^2$ governa le transizioni osservabili del processo. L'espressione nell'ultima equazione ha “ k ” termini che riflettono il fatto che ci sono dei rami che si evolvono attivamente e che qualsiasi di questi rami può dare origine ad un nuovo ramo figlio attraverso l'azione della matrice B . L'indipendenza dei “ k ” rami in evoluzione è evidente, dal fatto che non esiste alcuna interazione tra le “ k ” copie della matrice “ B ”.

Sviluppiamo un po' di più questi concetti. Si osserva che le matrici $A_2^{(k)}$, $A_1^{(k)}$ e $A_0^{(k)}$ dipendono di d , D_0 , D_1 e B .

- Gli elementi del vettore \mathbf{d} rappresentano i tassi di transizione nei quali ciascun ramo entra nello stato assorbente e muore, cioè, il ramo termina la sua evoluzione.
- Gli elementi della matrice \mathbf{D}_0 controllano le transizioni delle fasi che vengono eseguite all'interno di ciascun ramo. A seconda del modello che rappresentano, queste transizioni possono avere un'interpretazione fisica, come l'età o la condizione di salute degli individui, o possono essere puramente fittizi. Durante la loro vita, gli individui si riproducono e subiscono delle transizioni tra le fasi transitorie finché non entrano nella fase assorbente e muoiono.
- Gli elementi $B_{i,jk}$ della matrice \mathbf{B} rappresentano i tassi nei quali il ramo padre, che si trova nella fase “ i ”, genera un punto di ramificazione (transizione osservabile) in modo che il ramo figlio sia nella fase “ j ”, mentre il ramo parentale sia nella fase “ k ” immediatamente dopo il punto di ramificazione.

Quando si rappresenta pittoricamente un punto di ramificazione, il ramo parentale viene disegnato come il ramo destro e il ramo figlio viene disegnato come ramo sinistro.

Questi tre elementi soddisfano l'equazione:

$$D_0 \mathbf{e} + B \mathbf{e} + d = \mathbf{0}. \quad (4.0.8)$$

Usando (4.0.2) e (4.0.8) si può dimostrare facilmente che effettivamente, G è un generatore infinitesimale.

Di seguito diamo un esempio di come sarebbe la matrice B quando $n = 3$ (3 fasi).

	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
1	$(D_1)_{11}$	0	0	0	$(D_1)_{12}$	0	0	0	$(D_1)_{13}$
2	$(D_1)_{21}$	0	0	0	$(D_1)_{22}$	0	0	0	$(D_1)_{23}$
3	$(D_1)_{31}$	0	0	0	$(D_1)_{32}$	0	0	0	$(D_1)_{33}$

Le righe indicano la fase in cui il ramo era immediatamente prima che fosse sottoposto a una transizione osservabile e le colonne danno la fase di nascita del ramo figlio (la cifra sinistra) e la fase del ramo parentale (cifra destra) immediatamente dopo la transizione osservabile. Così, un

$$\begin{array}{cccccccc}
(\mathbf{m+1}, & a, & \cdots, & b, & \mathbf{i}, & \mathbf{j} & c, & \cdots, & d) \\
& \downarrow & \cdots & \downarrow & \downarrow & \downarrow & \downarrow & \cdots, & \downarrow \\
& 1 & \cdots, & k-1 & \mathbf{k} & \mathbf{k+1} & k+2 & \cdots, & m+1
\end{array}$$

L'albero è orientato in modo tale che il ramo parentale sia il ramo destro ed è associato alla “ $k + 1 -esima$ ” etichetta e il ramo figlio sia il ramo sinistro ed è associato alla $k -esima$ etichetta. I rami precedentemente etichettati con $k + 1, \dots, m$ sono stati nuovamente etichettati con $k + 2, \dots, m + 1$ rispettivamente.

- Infine, si verifica una catastrofe sul ramo “ k ” con un tasso d_r . Ciò provoca che il ramo k cessi di esistere e il nuovo stato sarà

$$\begin{array}{cccccccc}
(\mathbf{m-1}, & a, & \cdots, & b, & c, & \cdots, & d) \\
& \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \\
& 1 & \cdots, & k-1 & k & \cdots, & m-1
\end{array}$$

dove i rami che erano precedentemente etichettati da $k + 1, \dots, m$ sono stati nuovamente etichettati con $k, \dots, m - 1$ rispettivamente.

4.1 Markovian Binary Trees and Branching Processes

Si può interpretare un MBT come un caso speciale del *continuous-time Markovian multi-type Branching Process (ctMMTBP)*, visualizzando le fasi da 1 a “ n ” del MAP come i diversi tipi di particelle e dove ciascun punto di ramificazione può avere 0, 1 o 2 discendenti. In questa prospettiva, una particella di tipo “ i ” vivrà per un intervallo di tempo distribuito esponenzialmente con media $1/(d_0)_i$, al termine di questo intervallo la particella o,

- (1) muore senza dare origine a nessuna particella e questo avviene con probabilità $d_i/(d_0)_i$ (questo accade quando il MAP entra nella sua fase di assorbimento), o
- (2) muore e si trasforma in una singola particella di tipo $j \neq i$, e questo avviene con probabilità $(D_0)_{ij}/(d_0)_i$, (ciò corrisponde ad un cambiamento di fase nascosto) o finalmente,
- (3) muore e da luogo a due nuove particelle dei tipi “ j ” e “ k ” e questo si verifica con probabilità $B_{i,jk}/(d_0)_i$ (ciò corrisponde ad un evento osservabile).

Ricordiamo ora dalla sezione (1.4) che le funzioni generatrici di probabilità della distribuzione dei figli per ciascuna particella “ i ” sono date da,

$$\begin{aligned}
f^{(i)}(s_1, s_2, \dots, s_n) &= \sum_{j_1, j_2, \dots, j_n \geq 0} p^i(j_1, j_2, \dots, j_n) s_1^{j_1} s_2^{j_2} \dots s_n^{j_n} & (4.1.1) \\
0 \leq s_\alpha &\leq 1, \quad \alpha = 1, 2, \dots, n,
\end{aligned}$$

ove $p^{(i)}(j_1, j_2, \dots, j_n) =$ la probabilità che un genitore del tipo “ i ” produca j_1 particelle del tipo 1, j_2 particelle del tipo 2, ..., j_n particelle del tipo n .

Siano $\mathbf{s} = s_1, s_2, \dots, s_n$ e $\mathbf{j} = j_1, j_2, \dots, j_n$, allora l'equazione (4.1.1) può essere scritto come

$$f^{(i)}(\mathbf{s}) = \sum_{j_1, j_2, \dots, j_n \geq 0} p^i(\mathbf{j}) s_1^{j_1} s_2^{j_2} \dots s_n^{j_n} \quad (4.1.2)$$

Il MBT è un caso speciale del ctMMTBP con funzione generatrice di probabilità dei discendenti che sono quadratiche, quindi

$$f^{(i)}(\mathbf{s}) = \frac{d_i}{(d_0)_i} + \sum_{k=1, k \neq i}^n \frac{(D_0)_{ik}}{(d_0)_i} s_k + \sum_{k,j=1}^n \frac{B_{i,jk}}{(d_0)_i} s_j s_k, \quad (4.1.3)$$

dove $(d_0)_i = -(D_0)_{ii}$ per ogni $i \in \{1, 2, \dots, n\}$.

4.2 Regolarità e Numero medio di Branches

Poiché tutte le derivate di $f^{(i)}(\mathbf{s})$ sono chiaramente finiti allora il processo non esplose. Dall'equazione (1.4.10) abbiamo che la matrice del *expected number of branches* del MBT sarà

$$\mathcal{M}(t) = \exp(At), \quad (4.2.1)$$

dove

$$A_{ij} = (d_0)_i \kappa_{ij}, \quad (4.2.2)$$

con

$$\kappa_{ij} = \frac{1}{(d_0)_i} \left((1 - \delta_{ij})(D_0)_{ij} + \sum_{k=1}^n (B_{i,jk} + B_{i,kj}) \right) - \delta_{ij}, \quad (4.2.3)$$

dove δ_{ij} è il delta di Kronecker. Dalle equazioni (4.2.2) e (4.2.3) si può dedurre che

$$A = D_0 + BC, \quad (4.2.4)$$

dove C è una *counting matrix* di dimensione $n^2 \times n$, con

$$C_{ij,k} = I\{i = k\} + I\{j = k\}. \quad (4.2.5)$$

Quindi, *the expected number of branches* al tempo “ t ” viene dato da

$$\mathcal{M}(t) = \exp[(D_0 + BC)t]. \quad (4.2.6)$$

Nella sezione (1.4) abbiamo visto che il processo era *sub-critical*, *critical* o *super-critical* a seconda dell'autovalore dominante della matrice A . Così, per il MBT abbiamo la stessa classificazione. Dalle equazioni (4.2.4) e (4.2.5) abbiamo

$$A = D_0 + B(\mathbf{1} \oplus \mathbf{1}), \quad (4.2.7)$$

e sia λ_A l'autovalore della matrice A con parte reale massimo, allora

- se $\mathcal{R}e[\lambda_A] < 0$ il processo è *sub-critical*,
- se $\mathcal{R}e[\lambda_A] = 0$ il processo è *critical*, infine
- se $\mathcal{R}e[\lambda_A] > 0$ il processo è *super-critical*.

4.3 La probabilità di estinzione

Ricordiamo che la soluzione minima non negativa di

$$\mathbf{u}(\mathbf{s}) = \mathbf{0}, \quad (4.3.1)$$

è la probabilità di estinzione definitiva del processo, (vedere (1.4.12)). Ora se $\lambda_A < 0$ il processo si estinguerà q.c., e se $\lambda_A > 0$ allora $\mathbf{q} < \mathbf{e}$ *component-wise*. Per il *MBT* abbiamo,

$$\begin{aligned} 0 &= \mathbf{u}^{(i)}(\mathbf{s}) \\ &= (d_0)_i (f^{(i)}(\mathbf{s}) - s_i) \\ &= (d_0)_i \left(\frac{d_i}{(d_0)_i} + \sum_{k=1, k \neq i}^n \frac{(D_0)_{ik}}{(d_0)_i} s_k + \sum_{k,j=1}^n \frac{B_{i,jk}}{(d_0)_i} s_j s_k - s_i \right) \\ &= d_i + \sum_{k=1}^n (D_0)_{ik} s_k + \sum_{k=1}^n \sum_{j=1}^m B_{i,jk} s_k s_j \end{aligned}$$

Questo può essere riscritto in forma di matrice come,

$$0 = d + D_0 \mathbf{s} + B(\mathbf{s} \otimes \mathbf{s}). \quad (4.3.2)$$

Quindi, la probabilità di una eventuale estinzione è la soluzione minima non negativa dell'equazione (4.3.2) (vedere [4]). Questa equazione è la genesi dei metodi numerici per il MBT che sarà risolto nella sezione successiva. Diamo ora un modo alternativo per classificare i MBTs in sub-critical, critical o super-critical. Sia la matrice $\mathfrak{M}_{ij} = \partial f_i(\mathbf{s}) / \partial s_j |_{\mathbf{s}=\mathbf{1}}$, $i, j = 1, \dots, n$, cioè, \mathfrak{M}_{ij} è il numero medio di figli di tipo “ j ” nati da un genitore di tipo “ i ”, quindi abbiamo.

$$\mathfrak{M} = (-D_0)^{-1} B(\mathbf{1} \oplus \mathbf{1}) = (-D_0)^{-1} B(\mathbf{1} \otimes I + I \otimes \mathbf{1}), \quad (4.3.3)$$

Sia $\lambda_{\mathfrak{M}}$ il raggio spettrale della matrice \mathfrak{M} , allora

- se $\lambda_{\mathfrak{M}} < 1$ il processo è *sub-critical*, quindi $q = 1$;
- se $\lambda_{\mathfrak{M}} = 1$ il processo è *critical*, quindi anche $q = 1$ infine
- se $\lambda_{\mathfrak{M}} > 1$ il processo è *super-critical*, quindi $q \leq 1$. Inoltre, se il processo è *positive regular*, cioè, se \mathfrak{M} è irriducibile allora $q < 1$.

4.4 Approcci algoritmici per il MBT

La base teorica del *ctMMTBP* è ben consolidata in ([4],[19] e [44]), ciò nonostante, poco è stato fatto nello sviluppo di una base algoritmica che possa essere utilizzata per determinare le misure che sono usate in un contesto di modellazione. Dorman, Sinsheimer e Lange [15] hanno fornito un approccio algoritmico per un *ctMMTBP* con *Poissonian immigration*. Ad esempio, loro hanno potuto calcolare la probabilità di estinzione del processo in qualsiasi momento del tempo, tuttavia, loro hanno riconosciuto che i loro algoritmi hanno fallito nell'importante caso *super-critical* quando il tempo diventa grande. Il caso *super-critical* è il caso più importante dato che quando il processo è *sub-critical* e *critical* la probabilità di eventuali estinzioni è sempre “1”.

In questa sezione presentiamo i 4 principali algoritmi sviluppati fino ad oggi che possono determinare la probabilità di una eventuale estinzione, questi sono: *the Depth algorithm*, *the Order algorithm* (due tipi) e *the Thicknesses algorithm*, ognuno di loro con la sua versione accelerata di Newton.

4.4.1 Algoritmi con convergenza lineare

The Depth algorithm

Si consideri l'equazione (4.3.2) della sezione precedente, questa equazione può anche essere espresso come segue

$$\mathbf{s} = (-D_0)^{-1}d + (-D_0)^{-1}B(\mathbf{s} \otimes \mathbf{s}). \quad (4.4.1)$$

In generale, l'equazione (4.4.1) non può essere risolto in modo esplicito. Bean et al. ([6],[35]) hanno sviluppato due algoritmi iterativi, chiamati *the Depth algorithm* and *the Order algorithm* per risolvere l'equazione (4.4.1). L'algoritmo *Depth* è basato su iterazioni funzionali e procede come segue:

$$\begin{aligned} s(0) &= M \\ s(k) &= M + N(s(k-1) \otimes s(k-1)), \quad k \geq 1. \end{aligned} \quad (4.4.2)$$

dove $M = (-D_0)^{-1}d$ e $N = (-D_0)^{-1}B$. Riportiamo di seguito l'algoritmo.

Algoritmo: Depth

1. Inizializzare:
 - $M := (-D_0)^{-1}d$;
 - $N := (-D_0)^{-1}B$;
 - $q_0 = M$;
2. Fino alla convergenza:

Do:

 - $q = M + N * (q_0 \otimes q_0)$
 - Verificare se: $\|q - q_0\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: $q_0 = q$

EndDo

Definiamo ora la successione $\{q(k)\}_{k \geq 1}$, dove $q(k)$ rappresenta la probabilità che un albero inizi con un ramo ed eventualmente si estingua *under the taboo* che la sua profondità (*depth*) sia $\delta < k + 1$. Matematicamente,

$$q(k) = \mathbb{P}\{\delta(\mathcal{T}_{[\psi]}) < k + 1 | \phi(\alpha(\psi))\}, \quad (4.4.3)$$

per ogni $k \geq 0$. In altre parole, $q(k)$ è la probabilità che un albero avrà una profondità minore di $k + 1$ quando $t \rightarrow \infty$, che equivale a dire che $q(k)$ è la misura di probabilità di tutti gli alberi estinti con $\delta < k + 1$.

Teorema 4.1. *La successione $\{q(k)\}$ per $k \geq 0$ definita dall'equazione (4.4.3) è monotona crescente e converge al vettore “ q ”. La successione $\{q(k)\}$ soddisfa anche (4.4.2), cioè,*

$$\begin{aligned} q(0) &= M \\ q(k) &= M + N(q(k-1) \otimes q(k-1)), \quad k \geq 1. \end{aligned} \tag{4.4.4}$$

Dimostrazione. Il fatto che $\{q(k)\}$ è monotona crescente è ovvio. Che converga a q è anche evidente dal momento che $\lim_{k \rightarrow \infty} q(k) = \lim_{k \rightarrow \infty} \mathbb{P}\{\delta(\mathcal{T}_{[\psi]}) < k + 1 | \phi(\alpha(\psi))\} = \mathbb{P}\{\delta(\mathcal{T}_{[\psi]}) < \infty | \phi(\alpha(\psi))\} = q$. Il passo successivo è quello di dimostrare che $\{q(k)\}$ effettivamente soddisfa l'equazione (4.4.4). Un *tree* può eventualmente essere estinto in uno dei due modi: tramite una catastrofe su $[\psi]$, o quando un *branch point* in $[\psi]$ e $([\alpha(\psi)], [\psi])$ diventi un *internal branch*. I due *subtrees* che si basano attorno a $[\psi, 0]$ e $[\psi, 1]$ devono quindi essere estinti in modo indipendente.

La probabilità dal primo scenario è già dato da $M = (-D_0)^{-1}d$. Il secondo scenario è un po' più complicato. La probabilità che il *root branch* $([\alpha(\psi)], [\psi])$, eventualmente darà alla luce un *daughter branch* prima di subire una catastrofe è dato da $N = (-D_0)^{-1}B$. I *daughter* e *parental subtrees* che si basano intorno a $[\psi, 0]$ e $[\psi, 1]$ possono eventualmente essere estinti sotto il *taboo* che ogni *subtree* abbia un *depth* di al massimo $k - 1$, in modo che l'intero *tree* abbia un *depth* di al massimo k . Ma la probabilità di una eventuale estinzione di un *tree* con al massimo *depth* uguale a $k - 1$ è $q(k - 1)$. Poiché ognuno dei due *subtrees* generati dai *daughter* e *parent branches* sono indipendenti, abbiamo che,

$$q(k) = M + N(q(k-1) \otimes q(k-1)), \tag{4.4.5}$$

e la dimostrazione è completa. □

Nella teoria di QBDs (capitolo (3.1)), abbiamo visto che l'algoritmo di Neuts è stato derivato dalla seguente equazione

$$G = (-A_1)^{-1}A_2 + (-A_1)^{-1}A_0G^2 \tag{4.4.6}$$

con la procedura iterativa data in (3.2.3), cioè,

$$\begin{aligned} G_0 &= (-A_1)^{-1}A_2, \\ G_n &= (-A_1)^{-1}A_2 + (-A_1)^{-1}A_0G_{n-1}^2, \end{aligned} \tag{4.4.7}$$

dove $(-A_1)^{-1}A_2$ e $(-A_1)^{-1}A_0$ sono le probabilità di transizioni dopo aver lasciato il livello di partenza, per spostarsi un livello verso il basso o verso l'alto rispettivamente, e G è la probabilità di *first passage* dal livello “1” al livello “0”. La somiglianza di (4.4.6) con (4.4.1) è molto chiaro, anche se ci sono differenze. In (4.4.1), i termini quadratici indicano che sia il ramo sinistro che quello destro dell'albero devono finire indipendentemente in modo che l'intero albero diventi estinto; nell'equazione (4.4.6), il termine quadratico significa che entrambi i tempi di ritorno dal livello 2 al livello uno e quello successivo dal livello uno al livello zero devono essere finiti affinché l'intero primo tempo di ritorno a zero sia finito. Pertanto, si prevede che q sia “*come*” Ge in un certo senso, ma non identico a esso. Ulteriori dettagli su questo argomento sono riportati in [35].

The Order algorithm

Come abbiamo accennato in precedenza, questo algoritmo è stato anche sviluppato da Bean et al. ([6],[35]). Consideriamo di nuovo l'equazione (4.3.2), questa equazione può essere riscritto come

$$s = [I - N(s \otimes I)]^{-1}M; \quad (4.4.8)$$

a condizione che la matrice inversa nel lato destro di (4.4.8) esista. Ciò suggerisce un altro algoritmo iterativo basato sull'iterazione funzionale:

$$\begin{aligned} s(0) &= M \\ s(k) &= [I - N(s(k-1) \otimes I)]^{-1}M, \quad k \geq 1. \end{aligned} \quad (4.4.9)$$

dove $M = (-D_0)^{-1}d$ e $N = (-D_0)^{-1}B$. L'approssimazione $s(k)$ è la probabilità che l'albero eventualmente si estingua con un "order" al massimo uguale a "k". Di seguito daremo una definizione precisa di "order" di un albero e il teorema che assicura che il processo iterativo (4.4.9) realmente converga a "q", ma prima riportiamo l'algoritmo che chiameremo *algoritmo Order-1*.

Algoritmo: Order-1

1. Inizializzare:
 - $M := (-D_0)^{-1}d$;
 - $N := (-D_0)^{-1}B$;
 - $q_0 = M$;
2. Fino alla convergenza:

Do:

 - $U = N * (q_0 \otimes I)$;
 - $q = (I - U)^{-1} * M$;
 - Verificare se: $\|q - q_0\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: $q_0 = q$

EndDo

Definizione 4.4. L'ordine ("the order") di un nodo interno, $\theta^k(\psi)$, sul ramo parentale di una topologia $\mathcal{T}_{[\psi]}$ (vedere app. (B)), è indicato per $\mathbb{O}_n(\theta^k(\psi))$ e è data da,

$$\mathbb{O}_n(\theta^k(\psi)) = \mathcal{O}(\mathcal{T}_{[\theta^k(\psi), 0]}),$$

per $k = 0, 1, 2, \dots, N(\mathcal{T}_\psi)$. L'ordine di un singolo ramo è 0. Infine, l'ordine di \mathcal{T}_ψ è data da,

$$1 + \max_{k=0,1,2,\dots,N(\mathcal{T}_\psi)} \mathbb{O}_n(\theta^k(\psi)).$$

Dalla definizione è chiaro che abbiamo bisogno di calcolare l'ordine di un MBT in modo ricorsivo. Si comincia con tutti i nodi dei *parental branch* poi lavoriamo verso giù per le sottostrutture fino a raggiungere *the leaf branches*. L'ordine dell'albero è poi 1 in più rispetto al nodo di ordine

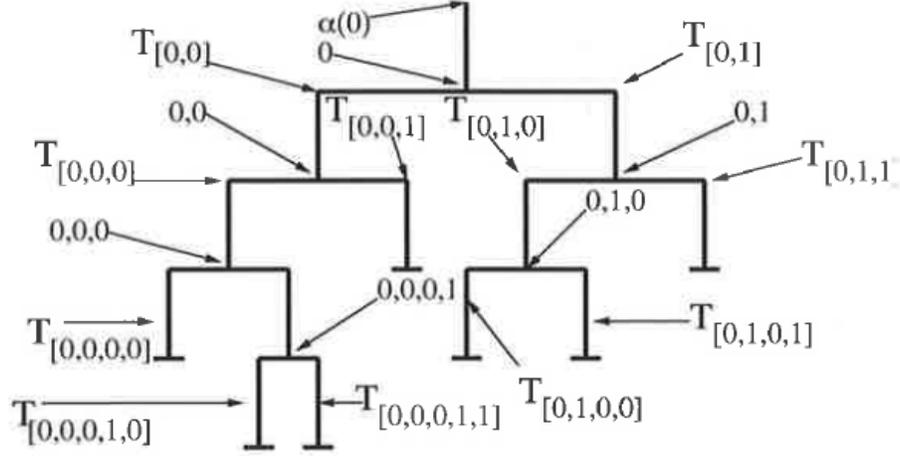


Figure 4.2: Un esempio del calcolo dell'ordine di un albero markoviano.

più alto lungo il *parental branch*. La figura (4.2) mostra un esempio di calcolo del ordine di un albero markoviano.

Per dimostrare che effettivamente il processo iterativo proposto in (4.4.9) converga a “ q ” occorre definire una matrice:

Definizione 4.5. La matrice $U = (U_{ij})$ per $i, j = 1, 2, \dots, n$ è definita da

$$U_{ij} = \mathbb{P}\{\mathcal{B}[\theta(\psi)], \mathcal{O}_n(\theta(\psi)) < \infty \ \& \ \phi_p(\theta(\psi))j | \phi_p(\psi) = i\}. \quad (4.4.10)$$

Il vettore “ q ” può anche essere definito come segue:

Definizione 4.6. Sia “ q ” la misura di probabilità di tutti i “*sample paths*” che hanno eventualmente zero particelle viventi, allora “ q ” è definito come

$$q = \mathbb{P}\{\mathcal{O}(\mathcal{T}_{[\psi]}) < \infty | \phi(\alpha(\psi))\}. \quad (4.4.11)$$

L’algoritmo per determinare la soluzione minima non negativa dell’equazione (4.4.8) è

$$\begin{aligned} s(0) &= M \\ U(k) &= N(s(k-1) \otimes I) \\ s(k) &= \sum_{k=0}^{\infty} [U(k)]^k M, \quad k \geq 1. \end{aligned} \quad (4.4.12)$$

dove $M = (-D_0)^{-1}d$ e $N = (-D_0)^{-1}B$.

Teorema 4.2. Le successioni $\{U(k) : k \geq 1\}$ e $\{q(k) : k \geq 1\}$ definite da

$$U(k) = \mathbb{P}\{\mathcal{B}[\theta(\psi)], \mathcal{O}_n(\theta(\psi)) < k \ \& \ \phi_p(\theta(\psi))j | \phi_p(\psi) = i\}. \quad (4.4.13)$$

e

$$q(k) = \mathbb{P}\{\mathcal{O}(\mathcal{T}_{[\psi]}) < k + 1 | \phi(\alpha(\psi))\}. \quad (4.4.14)$$

soddisfano (4.4.12), inoltre, le due successioni sono monotone crescenti e convergono rispettivamente alla matrice U e al vettore q .

Dimostrazione. Banalmente abbiamo che $\{U(k)\}$ e $\{q(k)\}$ sono monotone crescenti. Inoltre,

$$\begin{aligned}\lim_{k \rightarrow \infty} U(k) &= \lim_{k \rightarrow \infty} \mathbb{P}\{\mathcal{B}[\theta(\psi)], \mathbb{O}_n(\theta(\psi)) < k \ \& \ \phi_p(\theta(\psi)) | \phi_p(\psi)\} \\ &= \mathbb{P}\{\mathcal{B}[\theta(\psi)], \mathbb{O}_n(\theta(\psi)) < \infty \ \& \ \phi_p(\theta(\psi)) | \phi_p(\psi)\} \\ &= U,\end{aligned}$$

e

$$\begin{aligned}\lim_{k \rightarrow \infty} q(k) &= \lim_{k \rightarrow \infty} \mathbb{P}\{k + 1 | \phi(\alpha(0))\} \\ &= \mathbb{P}\{\mathbb{O}(\mathcal{T}) < \infty | \phi(\alpha(0))\} \\ &= q.\end{aligned}$$

quindi, abbiamo che

$$U(k) = \mathbb{P}\{\mathcal{B}[\theta(\psi)], \mathbb{O}_n(\theta(\psi)) < k \ \& \ \phi_p(\theta(\psi)) | \phi_p(\psi)\} \quad (4.4.15)$$

Tuttavia, la condizione $\mathbb{O}_n(\theta(\psi)) < k$ equivale a dire che $\mathbb{O}(\mathcal{T}_{[\psi,0]}) < k$, quindi possiamo scrivere

$$U(k) = \mathbb{P}\{\mathcal{B}[\theta(\psi)], \mathbb{O}_n(\mathcal{T}_{[\psi,0]}) < k \ \& \ \phi_p(\theta(\psi)) | \phi_p(\psi)\} \quad (4.4.16)$$

Poiché ogni evento si verifica in modo indipendente, possiamo riscrivere l'equazione sopra come

$$U(k) = \sum_{\phi_p(\theta(\psi))} \sum_{\phi(\alpha(\psi,0))} \mathbb{P}\{([\psi], [\theta(\psi)])^i \ \& \ \phi_p(\theta(\psi)) \ \& \ \phi(\alpha([\psi,0]) | \phi_p(\psi)) \times \mathbb{P}\{\mathbb{O}(\mathcal{T}_{[\psi,0]}) < k | \phi(\alpha([\psi,0])\},$$

quindi otteniamo

$$U(k) = (-D_0)^{-1} B(q(k-1) \otimes I^{(1)}), \quad (4.4.17)$$

dove il prodotto *Kronecker* con $I^{(1)}$ rappresenta che il *branch* presente è fissato con la probabilità $\mathbf{1}$.

Sia \mathcal{T}_ψ^κ una topologia che si basa sul nodo $[\psi]$ il cui *parental branch* ha esattamente κ nodi interni prima di terminare. Il primo subnode del *parental branch* interno è ovviamente $[\psi]$. Supponiamo che questa topologia abbia $\mathbb{O}(\mathcal{T}_{[\psi]}^\kappa) < k + 1$. La probabilità che un *random tree* eventualmente avrà questa proprietà è,

$$\mathbb{P}\{\mathbb{O}(\mathcal{T}_{[\psi]}^\kappa) < k + 1 | \phi(\alpha(\psi))\}. \quad (4.4.18)$$

Non è difficile vedere che l'espressione sopra è equivalente a,

$$\mathbb{P}\{\mathcal{A}[\theta^\kappa(\psi)] \ \& \ \mathbb{O}(\mathcal{T}_{[\psi]}^\kappa) < k + 1 | \phi(\alpha(\psi))\}, \quad (4.4.19)$$

poiché ci sono $\kappa - 1$ nodi interni di $[\psi]$, il primo nodo interno. Al $\kappa - th$ *parental subnode* del nodo *parent* $[\psi]$, si verifica un evento catastrofico. Ora, l'ordine di un *tree* è uno in più rispetto l'ordine del nodo del ramo parentale con l'ordine più alto e così abbiamo che

$$\mathbb{P}\{\mathcal{A}[\theta^\kappa(\psi)] \ \& \ \mathbb{O}(\mathcal{T}_{[\psi]}^\kappa) < k + 1 | \phi(\alpha(\psi))\} = \mathbb{P}\{\mathcal{A}[\theta^\kappa(\psi)] \ \& \ \max_{i=0, \dots, \kappa-1} \mathbb{O}_n(\theta^i(\psi)) < k | \phi(\alpha(\psi))\}. \quad (4.4.20)$$

L'espressione nell'equazione (4.4.20) è identica a dire che gli ordini di ciascuno dei nodi devono essere minore di k , quindi abbiamo

$$\begin{aligned}\mathbb{P}\{\mathcal{A}[\theta^\kappa(\psi)] \ \& \ \max_{i=0, \dots, \kappa-1} \mathbb{O}_n(\theta^i(\psi)) < k | \phi(\alpha(\psi))\} = \\ \mathbb{P}\{\mathcal{A}[\theta^\kappa(\psi)], \mathbb{O}_n(\psi) < k, \mathbb{O}_n(\theta(\psi)) < k, \dots, \mathbb{O}_n(\theta^{\kappa-1}(\psi)) < k | \phi(\alpha(\psi))\}.\end{aligned} \quad (4.4.21)$$

Ora ogni nodo e pertanto ogni *daughter subtree* si evolve in modo indipendente, quindi

$$\begin{aligned}
& \mathbb{P}\{\mathcal{A}[\theta^\kappa(\psi)], \mathbb{O}_n(\psi) < k, \mathbb{O}_n(\theta(\psi)) < k, \dots, \mathbb{O}_n(\theta^{\kappa-1}(\psi)) < k | \phi(\alpha(\psi))\} = \\
& \sum_{\phi_p(\psi)} \dots \sum_{\phi_p(\theta^{\kappa-1}(\psi))} \mathbb{P}\{\mathcal{B}[\psi], \mathbb{O}_n(\psi) < k \ \& \ \phi_p(\psi) | \phi(\alpha(\psi))\} \\
& \times \mathbb{P}\{\mathcal{B}[\theta(\psi)], \mathbb{O}_n(\theta(\psi)) < k \ \& \ \phi_p(\theta(\psi)) | \phi_p(\psi)\} \\
& \vdots \\
& \times \mathbb{P}\{\mathcal{A}[\theta^\kappa(\psi)] | \phi_p(\theta^{\kappa-1}(\psi))\}.
\end{aligned} \tag{4.4.22}$$

I primi κ termini sono i soli termini della forma di $U(k)$ e l'ultimo termine è uguale a $(-D_0^{-1})d$, perché dopo il κ -th nodo interno del *parental branch* si verifica una catastrofe. In altre parole,

$$\mathbb{P}\{\mathbb{O}(\mathcal{T}_{[\psi]}^\kappa) < k + 1 | \phi(\alpha(\psi))\} = U^\kappa(k)(-D_0)^{-1}d. \tag{4.4.23}$$

Tuttavia, per ottenere $q(k)$ dobbiamo sommare su tutte le κ , cioè, su tutto il numero possibile di *internal branch points* lungo il *parental branch*, quindi

$$q(k) = \sum_{\kappa=0}^{\infty} \mathbb{P}\{\mathbb{O}(\mathcal{T}_{[\psi]}^\kappa) < k + 1 | \phi(\alpha(\psi))\} = \sum_{\kappa=0}^{\infty} U^\kappa(k)(-D_0)^{-1}d. \tag{4.4.24}$$

□

Senza troppe difficoltà si può dimostrare che i processi iterativi funzionali (4.4.9) e (4.4.12) sono equivalenti.

Consideriamo ancora una volta l'equazione (4.3.2), questa equazione può essere riscritto come

$$s = [I - N(I \otimes s)]^{-1}M; \tag{4.4.25}$$

a condizione che la matrice inversa nel lato destro di (4.4.25) esista. Ciò suggerisce un altro algoritmo iterativo basato sull'iterazione funzionale:

$$\begin{aligned}
s(0) &= M \\
s(k) &= [I - N(I \otimes s(k-1))]^{-1}M, \quad k \geq 1.
\end{aligned} \tag{4.4.26}$$

dove $M = (-D_0)^{-1}d$ e $N = (-D_0)^{-1}B$. Chiameremo a questo algoritmo, *algoritmo Order-2*. Il modo costruttivo di questo algoritmo segue la medesima procedura dell'algoritmo *Order-1*, in modo che non è più necessario descriverlo nuovamente. Piuttosto, è interessante notare la somiglianza di questo algoritmo con l'algoritmo "U" dato nella sezione (3.2). In ([6],[35]) si possono trovare tutti i dettagli costruttivi dell'algoritmo *Order-1* ed in [21] troviamo un modo alternativo per ricavare questi due algoritmi appena visti con un approccio diverso. Riportiamo di seguito l'algoritmo.

Algoritmo: Order-2

1. Inizializzare:
 - $M := (-D_0)^{-1}d$;
 - $N := (-D_0)^{-1}B$;
 - $q_0 = M$;
2. Fino alla convergenza:

Do:

 - $U = N * (I \otimes q_0)$;
 - $q = (I - U)^{-1} * M$;
 - Verificare se: $\|q - q_0\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: $q_0 = q$

EndDo

The Thicknesses algorithm

Esiste un altro algoritmo proposto da Hautphenne-Latouche-Remiche ([26]-2009) chiamata “*The Thicknesses algorithm*”. Questo algoritmo procede per iterazione funzionale, in alternativa utilizzando le equazioni (4.4.8) e (4.4.25), in modo da definire

$$\begin{aligned}
 s_2(0) &= M, \\
 s_1(2k+1) &= [I - N(I \otimes s_2(2k))]^{-1}M, \\
 s_2(2k+2) &= [I - N(s_1(2k+1) \otimes I)]^{-1}M,
 \end{aligned} \tag{4.4.27}$$

dove $M = (-D_0)^{-1}d$ e $N = (-D_0)^{-1}B$ e $k \geq 0$. Per dimostrare che le successioni $\{s_2(2k)\}$ e $\{s_1(2k+1)\}$ sono ben definite e convergono monotonicamente a “ q ”, dobbiamo ridefinire “*the order*” di un albero in modo che sia più atta alla definizione di “*thicknesses*” di un albero.

Sia \mathcal{T} un *binary tree*, segniamo con “1” ogni *child arc* e con “0” ogni *parent arc*. Se l’albero \mathcal{T} è costituito dalla radice e da una singola foglia, cioè, se l’albero si estende senza un punto di ramificazione, allora l’unico arc è segnato con un valore di “0”. Infine, associamo ad ogni nodo $x \in \mathcal{T}$ (sia esso un punto di ramificazione o una foglia) la stringa di 0’s e 1’s che segnano il percorso più breve alla radice e definiamo $L(x)$ come la quantità di 1’s in questa sequenza. L’ordine $\mathbb{O}(\mathcal{T})$ è definito come

$$\mathbb{O}(\mathcal{T}) = \max\{L(x) : x \in \mathcal{T}\}.$$

Ad esempio, se prendiamo la foglia “ w ” nella figura (4.3), si osserva per un esame esaustivo di tutti i nodi che $\mathbb{O}(\mathcal{T}) = L(w) = 5$. Per riferimenti successivi, chiameremo *parental branch* dell’albero \mathcal{T} , quel ramo dalla radice costituita solo da *parental arcs*, cioè il ramo più a destra di \mathcal{T} e sarà indicato con \mathcal{R} . Analogamente, chiameremo il *first child branch* dell’albero \mathcal{T} , quel ramo dalla radice costituita solo da *child arcs* che è il ramo più a sinistra di \mathcal{T} e sarà indicato da \mathcal{L} .

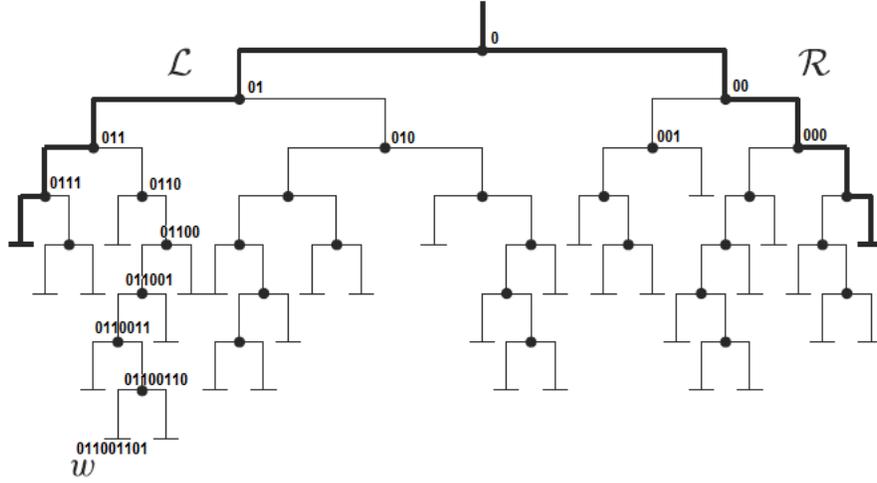


Figure 4.3: Un albero con $depth = 8$ e $order = 5$ per w .

Ad ogni nodo $x \in \mathcal{T}$ associamo ora due sequenze, $path1(x)$ e $path2(x)$: $path1(x)$ è la stringa di 0's e 1's che segnano il percorso più breve al *child branch* \mathcal{L} e $path2(x)$ è la stringa di 0's e 1's nel *branch* più a destra \mathcal{R} .

Successivamente, si definisce $N_1(x)$ e $N_2(x)$ rispettivamente come il numero di blocchi nelle sequenze $path1(x)$ e $path2(x)$, dove un blocco è una sequenza di 0's consecutivi o 1's consecutivi. Brevemente, questi contano il numero di cambi di direzione sui percorsi da un nodo a \mathcal{L} e a \mathcal{R} .

Il “*left thickness*” $\mathbb{S}_1(\mathcal{T})$ e il “*right thickness*” $\mathbb{S}_2(\mathcal{T})$ di un albero \mathcal{T} sono definiti come segue:

$$\begin{aligned} \mathbb{S}_1(\mathcal{T}) &= \max\{N_1(x) : x \in \mathcal{T}\}, \\ \mathbb{S}_2(\mathcal{T}) &= \max\{N_2(x) : x \in \mathcal{T}\}, \end{aligned} \quad (4.4.28)$$

Se \mathcal{T} consiste di una sola radice, allora $\mathbb{S}_1(\mathcal{T}) = \mathbb{S}_2(\mathcal{T}) = 0$. Nella figura (4.3) si vede che $N_1(w) = 4$ e $N_2(w) = 5$ e, mediante un esame esaustivo di tutti i nodi, possiamo dimostrare che $\mathbb{S}_1(\mathcal{T}) = 5$ e $\mathbb{S}_2(\mathcal{T}) = 6$.

Lemma 4.1. *Se \mathcal{T} è un MBT che diventa estinto, allora $\mathbb{S}_1(\mathcal{T}) < \infty$ e $\mathbb{S}_2(\mathcal{T}) < \infty$.* \square

Il contrario di questo lemma non è vero poiché è possibile per un albero con infiniti nodi, avere finiti *thicknesses*. Indichiamo ora con $[\mathcal{T} < \infty]$ l'evento per il quale il MBT diventa estinto. Quindi, $q_i = \mathbb{P}\{\mathcal{T} < \infty | \phi_0 = i\}$, dove ϕ_0 è la fase iniziale e scriviamo in breve come $q = \mathbb{P}\{\mathcal{T} < \infty | \phi_0\}$. Definiamo anche i vettori di probabilità

$$\begin{aligned} r_1(2k+1) &= \mathbb{P}\{\mathcal{T} < \infty \cap \mathbb{S}_1\mathcal{T} \leq 2k+1 | \phi_0\}, \\ r_2(2k) &= \mathbb{P}\{\mathcal{T} < \infty \cap \mathbb{S}_2\mathcal{T} \leq 2k | \phi_0\}, \end{aligned} \quad (4.4.29)$$

per $k \geq 0$. Allora abbiamo il seguente teorema.

Teorema 4.3. *Le successioni $\{r_1(2k+1) : k \geq 0\}$ e $\{r_2(2k) : k \geq 0\}$ soddisfano (4.4.27). Loro sono anche monotone crescenti e convergono a “ q ”. Inoltre, $r_2(2k) \leq r_1(2k+1) \leq r_2(2k+2)$, per ogni $k \geq 0$.*

Dimostrazione. Mostriamo che $[\mathbb{S}_2 \leq 2k] \subseteq [\mathbb{S}_1 \leq 2k+1]$, così, $r_2(2k) \leq r_1(2k+1)$. Se un *tree* ha un *right thickness* uguale a $2k$, allora il suo *left thickness* sarà al massimo pari a $2k+1$ perché, dopo

aver raggiunto il ramo più a destra, è necessario un blocco aggiuntivo di 0's solo per raggiungere il ramo più a sinistra. La prova che $r_1(2k + 1) \leq r_2(2k + 2)$ è simile.

Ora dimostreremo che la successione $\{r_1(2k + 1), k \geq 0\}$ converge a q . Utilizzando la prima equazione di (4.4.29) e il lemma (4.1), abbiamo

$$\begin{aligned} \lim_{k \rightarrow \infty} r_1(2k + 1) &= \lim_{k \rightarrow \infty} \mathbb{P}\{\mathcal{T} < \infty \cap \mathbb{S}_1(\mathcal{T}) \leq 2k + 1 | \phi_0\} \\ &= \mathbb{P}\{\mathcal{T} < \infty \cap \mathbb{S}_1(\mathcal{T}) < \infty | \phi_0\} \\ &= \mathbb{P}\{\mathcal{T} < \infty | \phi_0\}. \end{aligned} \tag{4.4.30}$$

La prova della seconda successione è simile. Infine, dobbiamo dimostrare che le due successioni $\{r_1(2k + 1), k \geq 0\}$ e $\{r_2(2k), k \geq 0\}$ soddisfano effettivamente l'equazione (4.4.27).

Per definizione, $\mathbb{S}_2(\mathcal{T}) = 0$ se e solo se il *root individual* muore prima di qualsiasi nascita, che accade con probabilità $M = (-D_0)^{-1}d$, in modo che $r_2(0) = q_2(0)$. Esistono due casi in cui un albero, eventualmente, viene estinto con un *left thickness* di al massimo pari a $2k + 1$: o la morte si verifica prima della prima nascita o una nascita avviene prima. Nel secondo caso,

- Il *subtree* sinistro \mathcal{T}^l eventualmente si estinguerà con un *thickness* sinistro al massimo pari a $2k + 1$ poiché il ramo più a sinistra di \mathcal{T}^l fa parte del ramo più a sinistra dell'intero albero \mathcal{T} ;
- Il *subtree* destro \mathcal{T}^r eventualmente si estinguerà con un *thickness* destro al massimo pari a $2k$ poiché per raggiungere il ramo \mathcal{L} più lontano dell'intero albero \mathcal{T} con al massimo $2k + 1$ cambi di direzione, si deve prima raggiungere il ramo più a destra di \mathcal{T}^r con al massimo $2k$ cambi di direzione e poi aggiungere un cambiamento di direzione per raggiungere \mathcal{L} .

La probabilità che esista una nascita prima di una morte è dato da $N = (-D_0)^{-1}B$ e le *random trees* \mathcal{T}^l e \mathcal{T}^r sono indipendenti date le loro fasi iniziali, quindi

$$\begin{aligned} r_1(2k + 1) &= M + N[r_1(2k + 1) \otimes r_2(2k)] \\ &= M + N[I \otimes r_2(2k)]r_1(2k + 1) \\ &= M + N[I \otimes r_2(2k)]M + (N[I \otimes r_2(2k)])^2 r_1(2k + 1) \\ &= \dots = \sum_{n=0}^{\infty} (N[I \otimes r_2(2k)])^n M. \end{aligned}$$

Poiché la serie converge, ciò è uguale a $(I - N[I \otimes r_2(2k)])^{-1}$, che dimostra la prima successione dell'equazione (4.4.27). Un argomento analogo è usato per dimostrare la seconda successione dell'equazione (4.4.27). \square

L'algoritmo è descritto di seguito.

Algoritmo: Thicknesses

1. Inizializzare:
 - $M := (-D_0)^{-1}d;$
 - $N := (-D_0)^{-1}B;$
 - $q2_0 = M;$
2. Fino alla convergenza:

Do:

 - $q1 = [I - N * (I \otimes q2_0)]^{-1} * M;$
 - $q2 = [I - N * (q1 \otimes I)]^{-1} * M;$
 - Verificare se: $\|q2 - q1\|_\infty \leq \epsilon$
 - SI: $q = q2, break$
 - NO: $q2_0 = q2$

EndDo

Comparing the Depth, Order and Thicknesses algorithms

In questa sezione confrontiamo l'efficienza dei 3 algoritmi descritti nella sezione precedente. Siccome il MBT è controllato da un MAP con “ n ” fasi, allora, a prima vista la complessità dell'algoritmo *Depth* è $\mathcal{O}(n^3)$, mentre la complessità degli algoritmi *Order-1*, *Order-2* e *Thicknesses* è $\mathcal{O}(n^4)$. In realtà, tuttavia, se si utilizza la struttura delle matrici considerate nell'espressione di questi algoritmi, in particolare il prodotto Kronecker con la matrice identità, si mostra che ogni iterazione è $\mathcal{O}(n^3)$ in tutti i casi. Quindi, il *tasso di convergenza* delle tre sequenze è il fattore chiave per confrontare l'efficienza dei tre algoritmi.

Definizione 4.7. *Si dice che un MBT è **positive regular** se la matrice*

$$\mathfrak{M} = N(I \otimes e + e \otimes I).$$

è irriducibile, con $N = (-D_0)^{-1}B$. In questo caso, $q = 1$ o $q < 1$, dove la disuguaglianza è stretta per tutti i componenti del vettore “ q ”.

Più avanti useremo il teorema che viene di seguito, ma prima ricordare che $\rho[X]$ indica il raggio spettrale della matrice X .

Teorema 4.4. *Supponiamo che il MBT sia positive regular. Nei casi sub-critical e super-critical, abbiamo che*

$$\rho[N(q \otimes I + I \otimes q)] < 1.$$

in modo che $[I - N(q \otimes I + I \otimes q)]^{-1}$ esiste e è non negativo. Nel caso critico, abbiamo,

$$\rho[N(q \otimes I + I \otimes q)] = 1.$$

Dimostrazione. La dimostrazione di questo teorema la possiamo trovare in [26] (pag 11-12). \square

Corollario 4.1. *Nei casi sub-critical e super-critical, $\rho[N(q \otimes I)] < 1$ e $\rho[N(I \otimes q)] < 1$, inoltre, $[I - N(q \otimes I)]^{-1}$ e $[I - N(I \otimes q)]^{-1}$ esistono e sono non negativi.*

Nel caso critical, $\rho[N(q \otimes I)] \leq 1$ e $\rho[N(I \otimes q)] \leq 1$.

Dimostrazione. Ciò segue subito da $N(q \otimes I) \leq N(q \otimes I + I \otimes q)$ e $N(I \otimes q) \leq N(q \otimes I + I \otimes q)$, e dalle proprietà di monotonicità del raggio spettrale per matrici non negativi. \square

Sia ora $\langle 0, q \rangle$ l'insieme dei vettori "x" tale che $0 \leq x \leq q$, con l'ordine parziale naturale, cioè, dove $x \leq q$ se $x_i \leq q_i$ per ogni i .

Corollario 4.2. *Nei casi sub-critical e super-critical, la matrice $[I - N(x \otimes I + I \otimes x)]^{-1}$ esiste e è non negativa per ogni $x \in \langle 0, q \rangle$, inoltre, $[I - N(x \otimes I + I \otimes x)]^{-1} \leq [I - N(y \otimes I + I \otimes y)]^{-1}$ per $0 \leq x \leq y \leq q$.*

Dimostrazione. Se $0 \leq x \leq q$, allora $0 \leq N(x \otimes I + I \otimes x) \leq N(q \otimes I + I \otimes q)$, quindi, nel caso sub- e supercritical abbiamo

$$0 \leq \sum_{i=0}^{\infty} [N(x \otimes I + I \otimes x)]^i \leq \sum_{i=0}^{\infty} [N(q \otimes I + I \otimes q)]^i < \infty$$

e così $[I - N(x \otimes I + I \otimes x)]^{-1}$ esiste e è non negativa. La seconda affermazione è dimostrata da un'argomentazione simile. \square

Denotiamo con $q_n^{(d)}$, $q_n^{(o)}$ e $q_n^{(t)}$ la n -esima approssimazione prodotta dall'algoritmo Depth, Order e Thicknesses rispettivamente, e definiamo gli errori di approssimazione:

$$E_n^{(d)} = q - q_n^{(d)}, \quad E_n^{(o)} = q - q_n^{(o)}, \quad E_n^{(t)} = q - q_n^{(t)}.$$

Abbiamo il seguente teorema,

Teorema 4.5. *Se il MBT è non critical, allora, i limiti superiori degli errori di approssimazione per i tre algoritmi sono i seguenti:*

- Per l'algoritmo "Depth",

$$E_n^{(d)} \leq N(q \otimes I + I \otimes q)E_{n-1}^{(d)}, \quad \forall n \quad (4.4.31)$$

- Per l'algoritmo "Order",

$$E_n^{(o)} \leq [I - N(q \otimes I)]^{-1}N(I \otimes q)E_{n-1}^{(o)}, \quad \forall n \quad (4.4.32)$$

- Per l'algoritmo "Thicknesses",

$$E_{2n+1}^{(t)} \leq [I - N(I \otimes q)]^{-1}N(q \otimes I)E_{2n}^{(t)}, \quad \forall n \quad (4.4.33)$$

$$E_{2n}^{(t)} \leq [I - N(q \otimes I)]^{-1}N(I \otimes q)E_{2n-1}^{(t)}, \quad \forall n \quad (4.4.34)$$

Dimostrazione. Proveremo solo l'equazione (4.4.32), le altre diseuguaglianze si dimostrano in modo simile. Riscriviamo quindi l'equazione (4.4.9) come $q_n^{(o)} = [I - N(q_{n-1}^{(o)} \otimes I)]^{-1} M = M + N(q_{n-1}^{(o)} \otimes q_n^{(o)})$. Siccome $q_{n-1}^{(o)} \leq q$, allora

$$\begin{aligned} E_n^{(o)} &= N(q \otimes q) - N(q_{n-1}^{(o)} \otimes q_n^{(o)}) \\ &= N((q - q_{n-1}^{(o)}) \otimes q) + N(q_{n-1}^{(o)} \otimes (q - q_n^{(o)})) \\ &= N(E_{n-1}^{(o)} \otimes q) + N(q_{n-1}^{(o)} \otimes E_n^{(o)}) \\ &\leq N(I \otimes q)E_{n-1}^{(o)} + N(q \otimes I)E_n^{(o)}. \end{aligned}$$

Così,

$$(I - N(q \otimes I))E_n^{(o)} \leq N(I \otimes q)E_{n-1}^{(o)}.$$

Se moltiplichiamo entrambi i lati da $[I - N(q \otimes I)]^{-1}$, che esiste e è non negativo, usando il corollario 4.1 si raggiunge il risultato desiderato. \square

Una conseguenza diretta è che nel caso non critical, tutti i tre algoritmi convergono linearmente. Se indichiamo con $\gamma^{(d)}$, $\gamma^{(o)}$ e $\gamma^{(t)}$ le loro velocità di convergenza, vediamo immediatamente che

$$\begin{aligned} \gamma^{(d)} &\leq \rho[\Omega(q)], \\ \gamma^{(o)} &\leq \rho[R_1(q)], \\ \gamma^{(t)} &\leq \{\rho[R_1(q)R_2(q)]\}^{1/2}, \end{aligned} \tag{4.4.35}$$

dove

$$\begin{aligned} \Omega(q) &= N(q \otimes I + I \otimes q), \\ R_1(q) &= [I - N(q \otimes I)]^{-1}N(I \otimes q), \\ R_2(q) &= [I - N(I \otimes q)]^{-1}N(q \otimes I). \end{aligned} \tag{4.4.36}$$

Il teorema successivo ci permette di confrontare il raggio spettrale delle tre matrici appena viste.

Teorema 4.6.

- Se il processo è non critical, allora

$$\rho[R_1(q)] \leq \rho[\Omega(q)] < 1, \tag{4.4.37}$$

e

$$\rho[R_2(q)] \leq \rho[\Omega(q)] < 1, \tag{4.4.38}$$

- Se il processo è critical e se $[I - N(q \otimes I)]^{-1}$ e $[I - N(I \otimes q)]^{-1}$ esistono, allora

$$\rho[R_2(q)] = \rho[R_1(q)] = \rho[\Omega(q)] = 1. \tag{4.4.39}$$

Dimostrazione. Utilizziamo i seguenti tre *splitting* regolari della *M-matrix* $A = I - \Omega(q)$:

$$\begin{aligned} A &= B_1 - C_1 \quad \text{con} \quad B_1 = I, \quad C_1 = \Omega(q), \\ A &= B_2 - C_2 \quad \text{con} \quad B_2 = I - N(q \otimes I), \quad C_2 = N(I \otimes q), \\ A &= B_3 - C_3 \quad \text{con} \quad B_3 = I - N(I \otimes q), \quad C_3 = N(q \otimes I). \end{aligned}$$

Siccome $C_1 \geq C_2$ e $C_1 \geq C_3$, usando il corollario 5.7, (p. 184) di Berman e Plemmons [7], risulta che $\rho[B_2^{-1}C_2] \leq \rho[C_1]$ e $\rho[B_3^{-1}C_3] \leq \rho[C_1]$ se $\rho[\Omega(q)] < 1$. Questo ultimo risultato, insieme con il teorema 4.4 e l'equazione (4.4.36) dimostra gli equazioni (4.4.37) e (4.4.38). Se il processo è *critical*, allora A è singolare poiché $\rho[\Omega(q)] = 1$. Il risultato in Berman e Plemmons [7] è ancora applicabile se B_2 e B_3 sono non singolari, quindi risulta l'equazione (4.4.39). \square

Il teorema (4.6) spiega in modo analitico perché l'algoritmo "Order" converge più velocemente rispetto all'algoritmo "Depth", questo è dimostrato in Bean et al. ([6],[35]) per argomenti probabilistici. Gli stessi argomenti mostrano che l'algoritmo "Thickneses" converge più velocemente in confronto all'algoritmo "Depth".

Non c'è, tuttavia, alcun confronto in forma generale tra la velocità di convergenza dell'algoritmo Order e Thickneses, nelle equazioni in (4.4.35) si vede che questi sono controllati dal raggio spettrale della matrice $R_1(q)$ e quella del prodotto $R_1(q)R_2(q)$ rispettivamente, due quantità che possono essere molto diverse. In [26], alla fine della sezione 4 vengono date alcune circostanze dalle quali ci si aspetta che l'algoritmo Thickneses sia più veloce.

Un'altra conseguenza del teorema (4.6) è che ci si aspetta una convergenza lenta per tutti i tre algoritmi nel caso in cui il processo sia *quasi critical*, cioè, quando $\rho[\Omega(q)]$ è molto vicino a 1.

4.4.2 Sperimentazione numerica con gli algoritmi lineari

Esempio 4.1 (Feminine offsprings of a Belgian woman). *L'obiettivo di questo primo esempio sarà confrontare la velocità di convergenza dei 4 primi algoritmi studiati. Si consideri la progenie delle donne. Siccome i MBTs sono modelli per la riproduzione asessuale, supponiamo che il processo inizia con una donna e limitiamo la nostra attenzione ai suoi discendenti femminili. Modelliamo la vita di una donna che progredisce in fasi corrispondenti a 22 classi di età come si indica nella tabella (4.1): alla nascita, da 1 a 4 anni, da 5 a 9 anni, ecc; fino all'intervallo tra 95 e 99 anni, e infine una classe per oltre 100 anni.*

Durante l'intervallo i -esimo, una donna può dare alla luce una figlia con tasso β_i e può morire con tasso d_i . La matrice B è tale che $B_{i,jk} = 0$ per $k \neq i$, che indica che l'età di una madre non cambia quando fa nascere una bambina; e $B_{i,jk} = 0$ per $j \neq 1$, poiché una bambina inizia automaticamente nella prima classe; infine, $B_{i,jk} = \beta_i$, per tutti "i".

Gli elementi non diagonali di D_0 sono $(D_0)_{1,2} = 1$, $(D_0)_{2,3} = 1/4$ e $(D_0)_{i,i+1} = 1/5$, per $i = 3, \dots, 21$ (l'unità di tempo è un anno) e gli elementi diagonali di D_0 sono tali che $D_0\mathbf{e} + B\mathbf{e} + d = 0$.

In questo caso si vede subito che $\rho[\mathfrak{M}] = 0.8631 < 1$, quindi il processo è sub-critical e sappiamo realmente che $q = 1$. Allora, se applichiamo i quattro algoritmi appena studiati, potremo verificare la loro precisione e la loro velocità di convergenza. In tutti gli esempi riportati in questo documento, è stato usato $\|q_n - q_{n-1}\|_\infty < 10^{-10}$ come criterio di arresto.

Diamo nella tabella (4.2) il numero di iterazioni (N.I.) e anche il valore dei due residui

$$r_1 = \|q^* - M - N(q^* \otimes q^*)\|_\infty \quad e \quad r_2 = \frac{\|q^* - M - N(q^* \otimes q^*)\|_1}{(\|q^*\|_1 + \|M\|_1 + \|N(q^* \otimes q^*)\|_1)}.$$

per i quattro algoritmi, dove q^* è l'approssimazione di q ottenuta numericamente. La figura (4.4a) mostra la velocità di convergenza; inoltre, diamo su una scala logaritmica (base 10) l'errore

<i>Classi</i>	β	d
0	0	0.00375
1-4	0	0.00021
5-9	0	0.00010
10-14	0	0.00011
15-19	0.0035	0.00024
20-24	0.02548	0.00022
25-29	0.06105	0.00035
30-34	0.04907	0.00047
35-39	0.01868	0.00070
40-44	0.00271	0.00113
45-49	0.00008	0.00191
50-54	0	0.00313
55-59	0	0.00443
60-64	0	0.00639
65-69	0	0.00955
70-74	0	0.01571
75-79	0	0.02967
80-84	0	0.05785
85-89	0	0.10650
90-94	0	0.18513
95-99	0	0.30392
≥ 100	0	0.47113

Tabella 4.1: Belgian women: birth and death rates by age classes.

<i>Algoritmi</i>	<i>N.I</i>	r_1	r_2
<i>Depth</i>	134	0.7488×10^{-10}	0.1250×10^{-10}
<i>Order - 1</i>	88	0.4597×10^{-10}	0.0793×10^{-10}
<i>Order - 2</i>	60	0.2669×10^{-10}	0.0422×10^{-10}
<i>Thicknesses</i>	74	0.5038×10^{-10}	0.0869×10^{-10}

Tabella 4.2: Confronto del numero di iterazioni (N.I.) e dei residui per i quattro algoritmi.

$\|E_n\|_\infty = \|\mathbf{e} - \mathbf{q}_n\|_\infty$ per i quattro algoritmi (fig. (4.4b)). La convergenza lineare è molto chiara e, come ci aspettavamo, l'algoritmo “*Depth*” è notevolmente più lento rispetto ad altri tre.

Esempio 4.2 (Order versus Thicknesses). *Analizziamo qui l'effetto della forma dell'albero sui due algoritmi: Order e Thicknesses. In questo esempio, gli individui hanno l'opportunità di avere molti figli, in modo che l'albero avrà “long branches” nella direzione parentale. Il numero di generazioni, che si traduce in numero di nodi in direzione a sinistra, è controllato da un parametro indicato con δ . Gli elementi della matrice di transizione \mathbf{B} hanno la forma $\mathbf{B}_{i,jk} = (\mathbf{D}_1)_{ik}(\mathbf{P}_1)_{ij}$, dove gli elementi di $(\mathbf{D}_1)_{ik}$ possono essere scomposti come un prodotto $\beta_i(\mathbf{P}_0)_{ik}$, dove β dà l'effetto dei tassi di nascita in ogni fase e la matrice stocastica \mathbf{P}_0 dà le probabilità condizionali per la fase parentale dopo la nascita, data la sua fase prima della nascita.*

Ci sono nove fasi, un individuo può morire solo quando è nella fase 5, quindi $d_i = 0$ per $i \neq 5$ e $d_5 = 1$, in modo che l'unità di tempo sia il tempo previsto fino alla morte quando è nella fase 5. Un individuo che si trova nella fase 1 attraversa le fasi da 1 a 4 (vedere l'angolo superiore della matrice \mathbf{D}_0); ogni volta che si trova nella fase 4, può anche passare alla fase 5, dove è probabile che muoia, o alla fase 6 dove il suo comportamento sarà molto diverso.

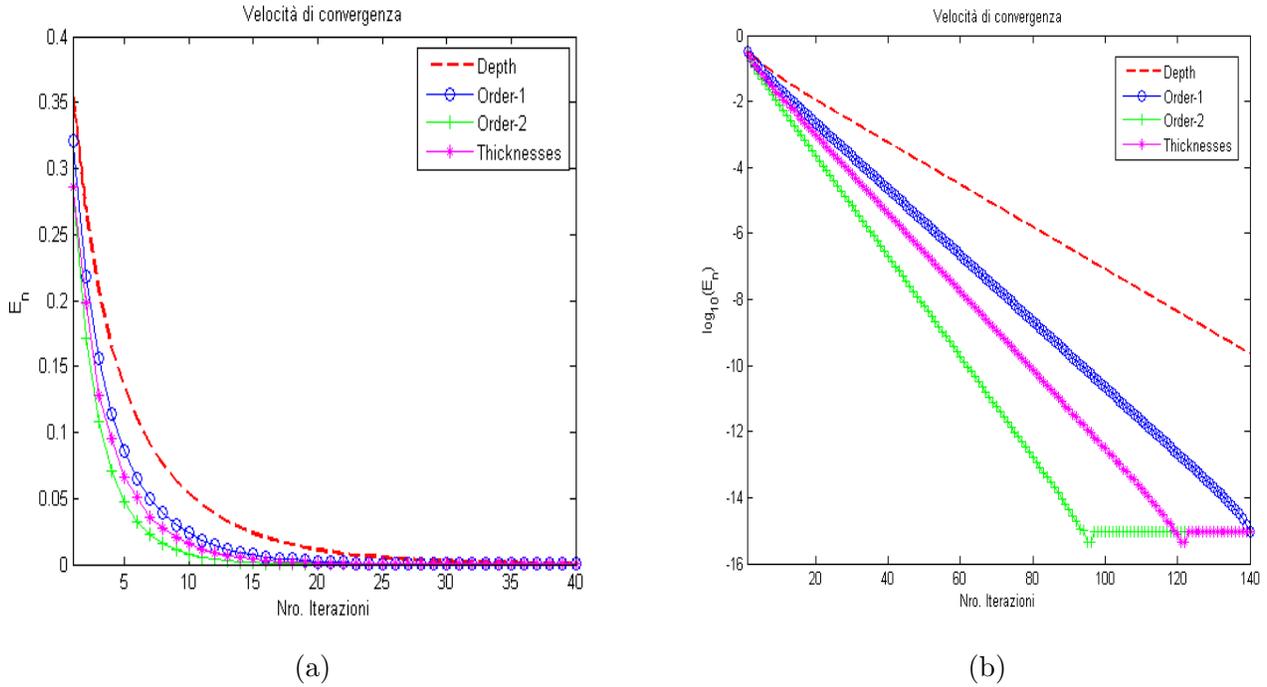


Figure 4.4: (a) Errori dei quattro algoritmi. (b) Logaritmi degli errori dei quattro algoritmi.

$$D_0 = 10^{-3} \cdot \left(\begin{array}{cccc|cccc} \star & 6 & \cdot \\ \cdot & \star & 6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \star & 6 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 6 & \cdot & \cdot & \star & 1 & 1 & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \star & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \star & 6 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \star & 6 & \cdot \\ \cdot & \star & 6 \\ 1 & \cdot & \cdot & \cdot & 1 & 6 & \cdot & \cdot & \star \end{array} \right)$$

$$P_0 = \left(\begin{array}{cccc|c|cccc} \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \hline 0.1 & \cdot & \cdot & \cdot & 0.9 & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot \\ \cdot & 1 \end{array} \right)$$

$$P_1 = \left(\begin{array}{cccc|c|cccc} 1 & \cdot \\ \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline 0.1 & \cdot & \cdot & \cdot & 0.9 & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot \\ \cdot & 1 \end{array} \right)$$

Le fasi da 6 a 9 costituiscono anche un ciclo, e dalla fase 9 un individuo può passare alla fase 5 o alla fase 1. Nelle fasi da 6 a 9, il tasso di nascita è elevato e, alla nascita, il genitore rimarrà nella stessa fase, in modo che i rami a destra siano molto lunghi; i figli iniziano nella fase 5, tuttavia, dove hanno un'elevata probabilità di morire senza lasciare figli in modo che i "children branches" a sinistra siano molto brevi. Il contrario è vero nelle fasi da 1 a 4: dopo la nascita, il genitore

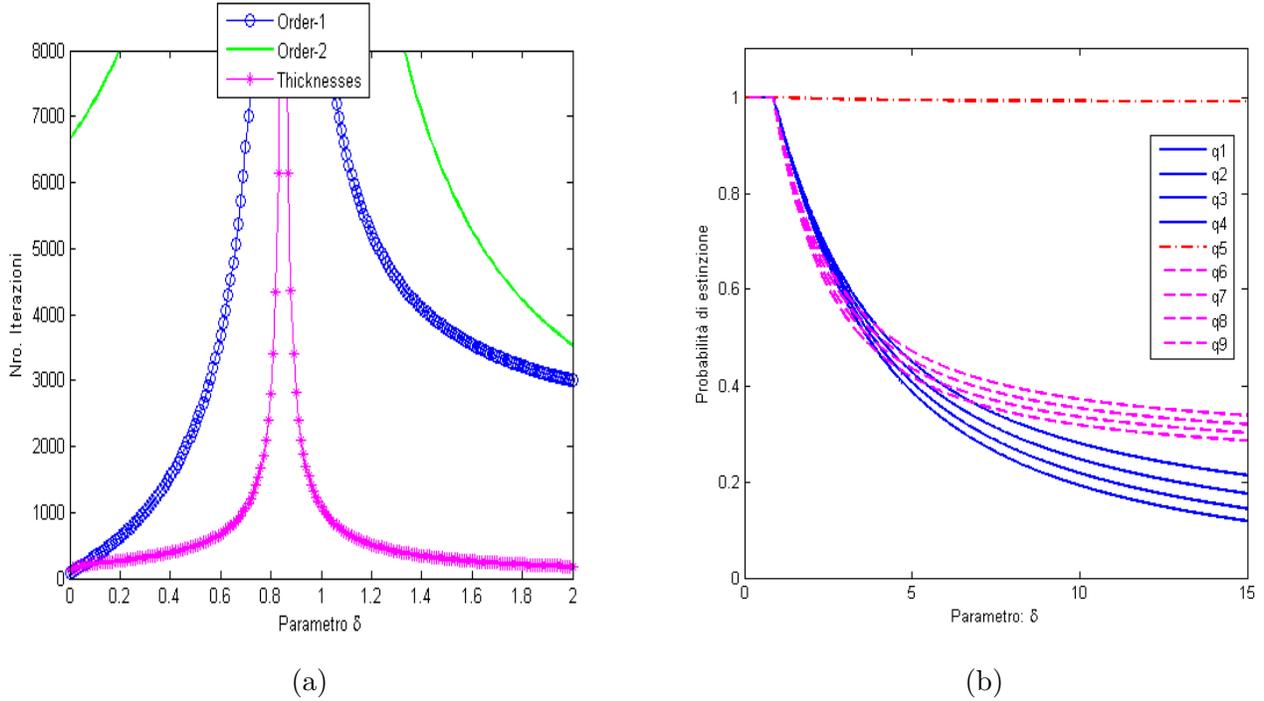


Figure 4.5: (a) Confronto del numero di iterazioni per gli algoritmi Order-1, Order-2 e Thicknesses, con $0 \leq \delta \leq 2$. (b) Calcolo del vettore probabilità di estinzione q per $0 \leq \delta \leq 15$.

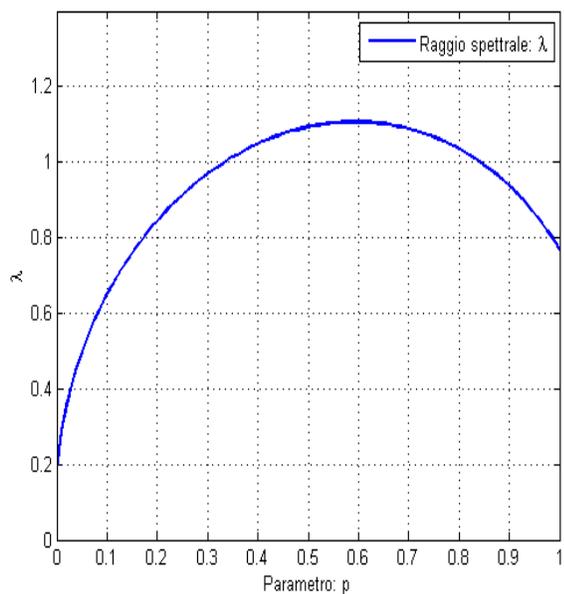
si muove alla fase 5, in modo che i “*right branches*” non crescano molto, mentre il figlio eredita la fase del suo genitore, in modo che i rami di sinistra hanno il potenziale di crescere. I tassi di nascita sono parametrizzate da δ , in modo che se δ è grande, le ramificazioni a sinistra hanno l’opportunità di crescere, mentre se δ è piccolo, i rami di sinistra non diventeranno molto lunghi. In sintesi, $\beta_i = 10^{-2}\delta$ per $i = 1, \dots, 4$; $\beta_i = 0.04$ per $i = 6, \dots, 9$ e $\beta_5 = 0.05$; le matrici P_0 e P_1 sono dati qui sopra e D_0 è tale che $D_0e + Be + d = 0$.

Per $\delta = 0$, raggio spettrale della matrice $\mathfrak{M} = N(e \oplus e)$ è $\rho = 0.998$ quindi il processo è sub-critical. Quando δ aumenta, aumenta anche ρ , il processo diventa critical per $\delta = 0.83$ approssimativamente e diventa super-critical al di là di quello. Mostriamo nella Fig. (4.5a) il numero di iterazioni necessarie per calcolare il vettore q con l’algoritmo Order-1 (linea interrotta con cerchi blu), l’algoritmo Order-2 (linea continua verde) e con l’algoritmo Thicknesses (linea interrotta con asterischi di colore lilla), in funzione di δ .

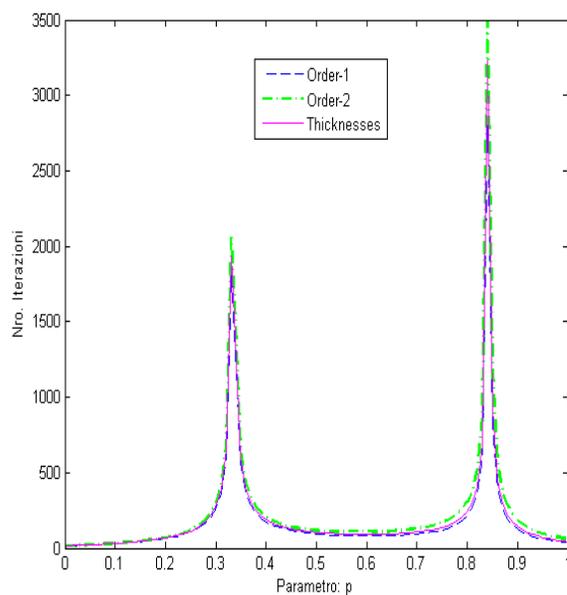
Come previsto alla fine della sezione precedente, osserviamo che il numero di iterazioni aumenta notevolmente intorno a $\delta = 0.83$, dove il processo è quasi critical.

Per valori grandi di δ , l’algoritmo Thicknesses si comporta in modo più efficiente che l’algoritmo Order, come ci aspettavamo. Per piccoli valori di δ , le cose sono migliori per l’algoritmo Order-1, anche se l’algoritmo Thicknesses rimane il più veloce dei tre. In Fig. (4.5b) mostra la soluzione q in funzione di δ . Nel caso super-critical, la probabilità q_5 non è mai esattamente 1, perché esiste una piccola probabilità, per un individuo nella fase 5, di generare un discendente e di passare alla fase 1.

Esempio 4.3 (Nearly critical MBTs). Per l’ultimo esempio di questa sezione prendiamo le matrici da Kontoleon [35].



(a)



(b)

Figure 4.6: (a) L'autovalore massimo λ della matrice \mathfrak{M} . (b) Confronto del numero di iterazioni degli algoritmi Order-1, Order-2 e Thicknesses, in funzione di p .

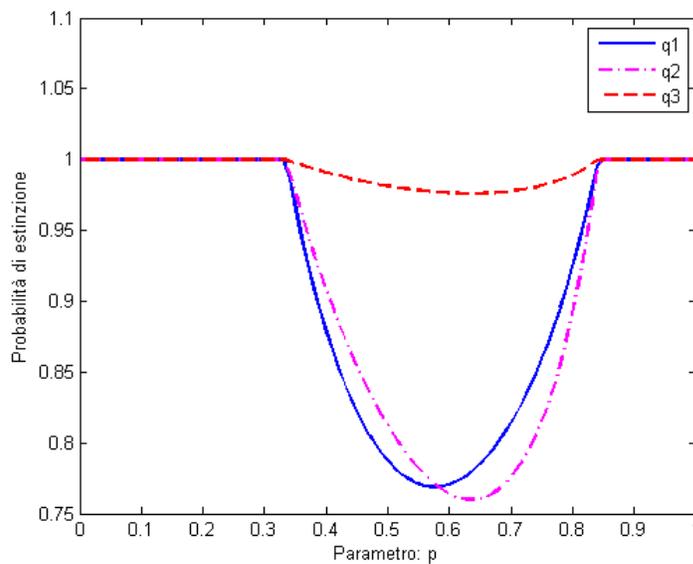


Figure 4.7: Calcolo del vettore probabilità di estinzione q per $0 \leq p \leq 1$.

$$d = \begin{pmatrix} 1 \\ 1 \\ 9 \end{pmatrix} \quad D_0 = \begin{pmatrix} -10 & 0 & 0 \\ 0 & -10 & 0 \\ 0 & 1 & -10 \end{pmatrix} \quad B = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 9(1-p) & 0 \\ 9p & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \middle| \begin{array}{ccc} 4.5p & 0 & 4.5p \\ 0 & 0 & 9(1-p) \\ 0 & 0 & 0 \end{array} \right)$$

Poiché “ p ” varia da 0 a 1, il processo varia da essere sub-critical a critical, super-critical, critical ed infine sub-critical di nuovo. Questo è mostrato nella Fig. (4.6a), mentre nella fig (4.6b) si mostra il numero di iterazioni richieste con tutti i tre algoritmi. Qui la differenza tra i tre algoritmi è minore, l’algoritmo Order-1 converge leggermente più velocemente rispetto all’algoritmo Thicknesses, mentre che l’algoritmo Thicknesses converge leggermente più velocemente rispetto all’algoritmo Order-2. Si vede chiaramente il rallentamento di tutti i tre algoritmi come previsto nella sezione precedente quando il processo si avvicina ad essere critical. Infine, la fig. (4.7) mostra la soluzione “ q ” in funzione del parametro p .

4.4.3 Algoritmi con convergenza quadratica

Abbiamo visto nella sezione (4.4) quattro metodi per calcolare il vettore di probabilità di estinzione “ q ”. Questi metodi utilizzano iterazioni di punto fisso su equazioni specifiche, questi sono:

- L’algoritmo *Depth*, che itera sull’equazione

$$\mathbf{s} = M + N(\mathbf{s} \otimes \mathbf{s}); \quad (4.4.40)$$

- L’algoritmo *Order-1*, che itera sull’equazione

$$\mathbf{s} = [I - N(\mathbf{s} \otimes I)]^{-1}M; \quad (4.4.41)$$

- L’algoritmo *Order-2*, che itera sull’equazione

$$\mathbf{s} = [I - N(I \otimes \mathbf{s})]^{-1}M; \quad (4.4.42)$$

- L’algoritmo *Thicknesses*, che itera sulle equazioni

$$\begin{cases} \mathbf{s} = [I - N(I \otimes \mathbf{s})]^{-1}M; \\ e \\ \mathbf{s} = [I - N(\mathbf{s} \otimes I)]^{-1}M; \end{cases} \quad (4.4.43)$$

Dove $M = (-D_0)^{-1}d$ e $N = (-D_0)^{-1}B$.

Ciò suggerisce che potremmo vedere queste equazioni come operatori definiti su spazi di Banach e tentare di risolverle usando il metodo di Newton, cioè, si potrebbero riformulare le equazioni (4.4.40), (4.4.41), (4.4.42) e (4.4.43) rispettivamente come segue

$${}_d\mathcal{F}(\mathbf{s}) = \mathbf{s} - M - N(\mathbf{s} \otimes \mathbf{s}); \quad (4.4.44)$$

$${}_{o1}\mathcal{F}(\mathbf{s}) = \mathbf{s} - [I - N(\mathbf{s} \otimes I)]^{-1}M; \quad (4.4.45)$$

$${}_{o2}\mathcal{F}(\mathbf{s}) = \mathbf{s} - [I - N(I \otimes \mathbf{s})]^{-1}M; \quad (4.4.46)$$

$$\begin{cases} {}_{o1}\mathcal{F}(\mathbf{s}) = \mathbf{s} - [I - N(\mathbf{s} \otimes I)]^{-1}M \\ e \\ {}_{o2}\mathcal{F}(\mathbf{s}) = \mathbf{s} - [I - N(I \otimes \mathbf{s})]^{-1}M; \end{cases} \quad (4.4.47)$$

Dove il pedice sinistro significa che la funzione \mathcal{F} viene applicata a ciascuno dei quattro metodi sopra descritti. Vediamo di seguito come viene fatta questa procedura.

Newton's iteration for the Extinction Probability

Sia \mathcal{F} un operatore differenziabile secondo Fréchet¹, tale che $\mathcal{F} : \Omega_0 \subseteq X \rightarrow Y$, dove X e Y sono spazi di Banach e Ω_0 è un insieme convesso aperto.

Per trovare una soluzione $x = x^*$ dell'equazione $\mathcal{F}(x) = 0$, si applica spesso il *metodo di Newton*, che consiste nella costruzione della successione $\{x\}_k$ definita da

$$x_{k+1} = x_k - [\mathcal{F}'(x_k)]^{-1}\mathcal{F}(x_k), \quad k = 0, 1, 2, \dots \quad (4.4.48)$$

iniziando da un $x_0 \in \Omega_0$ opportunamente scelto. Le condizioni sufficienti per il successo di questa procedura sono fornite dal famoso *teorema di L. V. Kantorovich* ([49],[58]).

Teorema 4.7 (Teorema di Kantorovich). *Supponiamo che per un qualche $x_0 \in \Omega_0$, esiste $[\mathcal{F}'(x_0)]^{-1}$ e sono soddisfatte le seguenti condizioni*

$$(i) \quad \|[\mathcal{F}'(x_0)]^{-1}\| \leq \mathfrak{B},$$

$$(ii) \quad \|[\mathcal{F}'(x_0)]^{-1}\mathcal{F}(x_0)\| \leq \mathfrak{R},$$

$$(iii) \quad \|\mathcal{F}'(x) - \mathcal{F}'(y)\| \leq \mathfrak{L}\|x - y\| \quad \forall x, y \in \Omega_0$$

con $\mathfrak{h} = \mathfrak{B}\mathfrak{R}\mathfrak{L} \leq \frac{1}{2}$. Sia

$$\Omega_* = \{x : \|x - x_0\| \leq t^*\} \quad \text{con } t^* = \left(\frac{1 - \sqrt{1 - 2\mathfrak{h}}}{\mathfrak{h}}\right)\mathfrak{R},$$

allora, se $\Omega_* \subseteq \Omega_0$, abbiamo che la procedura iterativa proposta in (4.4.48) è ben definita per ogni $x_k \in \Omega_*$ e converge a $x^* \in \Omega_*$, tale che

$$\mathcal{F}(x^*) = 0,$$

inoltre

$$\|x^* - x_k\| \leq \left(\frac{\mathfrak{R}}{\mathfrak{h}}\right) \left(\frac{(1 - \sqrt{1 - 2\mathfrak{h}})^{2^k}}{2^k}\right), \quad k = 0, 1, 2, \dots$$

Il valore della costante \mathfrak{h} è significativo per la convergenza del metodo di Newton. Se $\mathfrak{h} = 1/2$, allora ci sono degli esempi dove la successione di Newton non converge più rapidamente di una progressione geometrica con il rapporto $1/2$ e inoltre $\mathcal{F}'(x^*)$ non è invertibile. D'altra parte, se

¹L'operatore $\mathcal{F} : \Omega_0 \subseteq X \rightarrow Y$ è Fréchet-differenziabile in ogni $x \in \Omega_0$ se esiste un operatore lineare $A : X \rightarrow Y$ tale che

$$\lim_{h \rightarrow 0} \frac{\|\mathcal{F}(x+h) - \mathcal{F}(x) - Ah\|}{\|h\|} = 0.$$

L'operatore lineare A è denotato con \mathcal{F}'_x e è chiamato il *Fréchet derivative* di \mathcal{F} in x ([49]).

$\flat < 1/2$ allora la successione di Newton converge molto più rapidamente di una progressione geometrica; inoltre, è ben noto che $[\mathcal{F}'(x^*)]^{-1}$ esiste in questo caso.

Per implementare il metodo di Newton per le equazioni descritte (4.4.44), (4.4.45), (4.4.46) e (4.4.47), in appresso, supporremo che il *MBT* sia *positive regular*, il quale implica che $o \ q = 1$ o $q < 1$. Possiamo anche supporre che il *MBT* sia *super-critical*. Altrimenti, la probabilità di estinzione sarebbe esattamente uguale al vettore $\mathbf{1}$ e allora non ci sarebbe bisogno di algoritmi per risolvere l'equazione di estinzione.

Denotiamo da $\langle u, v \rangle$ l'insieme dei vettori x tale che $u \leq x \leq v$, dove usiamo l'ordine parziale naturale tale che $x \leq y$ se $x_i \leq y_i$ per tutti "i".

In Hautphenne et al. [23], gli autori hanno sviluppato un algoritmo quadratico per calcolare la probabilità di estinzione del MBT basato sul metodo di iterazione di Newton per la soluzione dell'equazione (4.4.44).

La procedura è come segue, per qualsiasi norma matriciale, \mathbb{R}^n è uno spazio di Banach, e la funzione ${}_d\mathcal{F}$ è una mappa da \mathbb{R}^n in se stessa. Il *Fréchet derivative* di ${}_d\mathcal{F}$ in x è una mappa lineare ${}_d\mathcal{F}'_x$ data da

$${}_d\mathcal{F}'_x : z \mapsto [I - N(x \oplus x)]z = z - N(z \otimes x + x \otimes z). \quad (4.4.49)$$

Dato un x_0 , la successione di Newton per la soluzione di ${}_d\mathcal{F}(x) = 0$ è

$$\begin{aligned} x_{k+1} &= x_k - ({}_d\mathcal{F}'_{x_k})^{-1} {}_d\mathcal{F}(x_k) \\ &= x_k - [I - N(x_k \oplus x_k)]^{-1} [x_k - M - N(x_k \oplus x_k)] \\ &= [I - N(x_k \oplus x_k)]^{-1} [M - N(x_k \oplus x_k)] \end{aligned} \quad (4.4.50)$$

per $k = 0, 1, 2, \dots$ a condizione che ${}_d\mathcal{F}'_{x_k}$ sia invertibile per ogni k .

Non è difficile verificare che per $x_0 \in \langle 0, M \rangle$ tutte le ipotesi del Teorema di Kantorovich (4.7) sono soddisfatte, oltre a ciò, abbiamo il seguente teorema

Teorema 4.8. *Per ogni $x_0 \in \langle 0, M \rangle$, la procedura iterativa proposta in (4.4.50) è tale che*

(a) *la successione $\{x_k\}$ è ben definita,*

(b) $x_0 \leq x_1 \leq x_2 \leq \dots$,

(c) $\lim_{k \rightarrow \infty} x_k = q$.

Inoltre, esiste una costante positiva c tale che

$$\|x_{k+1} - q\| \leq c \|x_k - q\|^2, \quad (4.4.51)$$

in modo che l'algoritmo di Newton converga (almeno) in modo quadratico.

La dimostrazione di questo teorema è riportato in [23]. In linea di principio, il punto di partenza x_0 può essere scelto ovunque in $\langle 0, M \rangle$, ma in pratica si sceglie un punto il più vicino possibile alla soluzione q , quindi si può prendere $x_0 = M$ come punto di partenza. Questo è il primo algoritmo di newton e lo chiameremo *Newton-Depth*. L'algoritmo è riportato qui sotto.

Algoritmo: Newton-depth

1. Inizializzare:

- $M := (-D_0)^{-1}d;$
- $N := (-D_0)^{-1}B;$
- $q_0 = M;$

2. Fino alla convergenza:

Do:

- $q = [I - N * (I \otimes q_0 + q_0 \otimes I)]^{-1} * (M - N * q_0 \otimes q_0);$
- Verificare se: $\|q - q_0\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: $q_0 = q$

EndDo

Consideriamo ora l'equazione (4.4.45). Analogamente al caso precedente per qualsiasi norma matriciale, \mathbb{R}^n è uno spazio di Banach, e la funzione ${}_{o1}\mathcal{F}$ è una mappa da \mathbb{R}^n in se stessa. Il *Fréchet derivative* di ${}_{o1}\mathcal{F}$ in x è una mappa lineare ${}_{o1}\mathcal{F}'_x$ data da

$${}_{o1}\mathcal{F}'_x : z \mapsto [I - T_x N(I \otimes T_x M)]z = [z - T_x N(z \otimes T_x M)], \quad (4.4.52)$$

dove $T_x = [I - N(x \otimes I)]^{-1}$. Dato un x_0 , la successione di Newton per la soluzione di ${}_{o1}\mathcal{F}(x) = 0$ è

$$\begin{aligned} x_{k+1} &= x_k - ({}_{o1}\mathcal{F}'_{x_k})^{-1} {}_{o1}\mathcal{F}(x_k), \\ &= x_k - [I - T_{x_k} N(I \otimes T_{x_k} M)]^{-1} [x_k - T_{x_k} M], \end{aligned} \quad (4.4.53)$$

con $T_{x_k} = [I - N(x_k \otimes I)]^{-1}$, per $k = 0, 1, 2, \dots$ a condizione che ${}_{o1}\mathcal{F}'_{x_k}$ sia invertibile per ogni k . In questo modo abbiamo ottenuto il secondo algoritmo di Newton, che chiameremo *Newton-Order-1*. L'algoritmo è riportato qui sotto.

Algoritmo: Newton-Order-1

1. Inizializzare:

- $M := (-D_0)^{-1}d;$
- $N := (-D_0)^{-1}B;$
- $q_0 = M;$

2. Fino alla convergenza:

Do:

- $T = [I - N * q_0 \otimes I]^{-1};$
- $q = q_0 - [I - T * N * (I \otimes T * M)]^{-1} * (q_0 - T * M);$
- Verificare se: $\|q - q_0\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: $q_0 = q$

EndDo

La procedura per ottenere il terzo algoritmo di Newton ripete gli stessi passi di cui sopra, in questo caso il *Fréchet derivative* di $o_2\mathcal{F}$ in x è una mappa lineare $o_2\mathcal{F}'_x$ data da

$$o_2\mathcal{F}'_x : z \mapsto [I - S_x N(S_x M \otimes I)]z = [z - S_x N(S_x M \otimes z)], \quad (4.4.54)$$

dove $S_x = [I - N(I \otimes x)]^{-1}$. Dato un x_0 , la successione di Newton per la soluzione di $o_2\mathcal{F}(x) = 0$ è

$$\begin{aligned} x_{k+1} &= x_k - (o_2\mathcal{F}'_{x_k})^{-1} o_2\mathcal{F}(x_k), \\ &= x_k - [I - S_{x_k} N(S_{x_k} M \otimes I)]^{-1} [x_k - S_{x_k} M], \end{aligned} \quad (4.4.55)$$

con $S_{x_k} = [I - N(I \otimes x_k)]^{-1}$, per $k = 0, 1, 2, \dots$ a condizione che $o_2\mathcal{F}'_{x_k}$ sia invertibile per ogni k . Questo terzo algoritmo di Newton lo chiameremo *Newton-Order-2*. Infine, possiamo ottenere il quarto algoritmo di newton considerando le funzioni (4.4.52) e (4.4.54), allora la procedura iterativa sarebbe:

$$\begin{cases} x_{2k+1} = x_{2k} - [I - S_{x_{2k}} N(S_{x_{2k}} M \otimes I)]^{-1} [x_{2k} - S_{x_{2k}} M], \\ x_{2k+2} = x_{2k+1} - [I - T_{x_{2k+1}} N(I \otimes T_{x_{2k+1}} M)]^{-1} [x_{2k+1} - T_{x_{2k+1}} M], \end{cases} \quad (4.4.56)$$

dove $T_{x_{2k+1}} = [I - N(x_{2k+1} \otimes I)]^{-1}$ e $S_{x_{2k}} = [I - N(I \otimes x_{2k})]^{-1}$, per $k = 0, 1, 2, \dots$. Questo quarto algoritmo lo chiameremo *Newton-Thicknesses*, riportiamo di seguito i due algoritmi.

Algoritmo: Newton-Order-2

1. Inizializzare:

- $M := (-D_0)^{-1}d;$
- $N := (-D_0)^{-1}B;$
- $q_0 = M;$

2. Fino alla convergenza:

Do:

- $S = [I - N * I \otimes q_0]^{-1};$
- $q = q_0 - [I - S * N * (S * M \otimes I)]^{-1} * (q_0 - S * M);$
- Verificare se: $\|q - q_0\|_\infty \leq \epsilon$
 - SI: *break*
 - NO: $q_0 = q$

EndDo

Algoritmo: Newton-Thicknesses

1. Inizializzare:

- $M := (-D_0)^{-1}d;$
- $N := (-D_0)^{-1}B;$
- $q2_0 = M;$

2. Fino alla convergenza:

Do:

- $S = [I - N * I \otimes q2_0]^{-1};$
- $q1 = q2_0 - [I - S * N * (S * M \otimes I)]^{-1} * (q2_0 - S * M);$
- $T = [I - N * q1 \otimes I]^{-1};$
- $q2 = q1 - [I - T * N * (I \otimes T * M)]^{-1} * (q1 - T * M);$
- Verificare se: $\|q2 - q1\|_\infty \leq \epsilon$
 - SI: $q = q2$, *break*
 - NO: $q2_0 = q2$

EndDo

Si può dimostrare che le inverse delle matrici $[I - T_{x_k} N(I \otimes T_{x_k} M)]^{-1}$ e $[I - S_{x_k} N(S_{x_k} M \otimes I)]^{-1}$ nelle equazioni (4.4.53) e (4.4.55) rispettivamente esistono per tutti $x_k \leq q$, utilizzando un'interpretazione delle equazioni ${}_{o1}\mathcal{F}(x) = 0$ e ${}_{o2}\mathcal{F}(x) = 0$ come equazioni di estinzione per un nuovo processo di ramificazione *embedded* nel MBT, inoltre, che per ogni $x_0 \in \langle 0, M \rangle$, i processi iterativi in (4.4.53), (4.4.55) e (4.4.56) sono monotoni, crescenti e convergono (almeno) in modo

quadratico alla probabilità di estinzione “ q ” del MBT. Nel caso dell’algoritmo Newton-Depth, Hautphenne et al. [23] hanno dimostrato utilizzando il “*Newton Attraction Theorem*” [49] che la convergenza è esattamente quadratica.

In [21] viene data una costruzione dettagliata dell’algoritmo Newton-Order-2 usando “*tree-structured Markov chains*”, e vengono dati anche suggerimenti per le dimostrazioni delle proprietà appena menzionate degli ultimi tre algoritmi di Newton.

Numerical complexity

Gli algoritmi di Newton sono stati ottimizzati utilizzando la particolare struttura delle matrici coinvolte in ogni iterazione, in particolare il prodotto Kronecker con la matrice identità. Facendo i calcoli rispettivi, si ottiene una complessità di $(20/3)n^3 + \mathcal{O}(n^2)$ flops per iterazione per l’algoritmo *Newton-Depth*, mentre per gli altri tre algoritmi: *Newton-Order-1*, *Newton-Order-2* e *Newton-Thicknesses* si ottiene una complessità di $8n^3 + \mathcal{O}(n^2)$ flops per iterazione. Ci aspettiamo quindi che l’algoritmo *Newton-Depth* sia un po’ più veloce degli altri algoritmi, tuttavia, Hautphenne e Benny in [21] hanno dimostrato tramite una proposizione che l’algoritmo *Newton-Order-2* converge più velocemente che l’algoritmo *Newton-Depth* (in termini di numero di iterazioni) verso la soluzione “ q ”, cioè, la proposizione mostra che in ogni iterazione, l’algoritmo *Newton-Order-2* è più vicino alla soluzione “ q ” rispetto all’algoritmo *Newton-Depth*. Più specificamente, la proposizione stabilisce quanto segue:

Proposizione 4.1 (Proposizione (3.1) in [21]). *Siano \tilde{x}_k le approssimazioni dell’algoritmo Newton-Depth e x_k le approssimazioni dell’algoritmo Newton-Order-2 $\forall k \geq 0$. Indichiamo la differenza tra la k -esima iterazione nei due algoritmi di Newton da $\Delta_k = x_k - \tilde{x}_k$, allora, la differenza Δ_k è non negativa per ogni $k \geq 0$.*

Ciò non significa che l’algoritmo Newton-Depth sia più lento rispetto all’altro, solo che può richiedere più iterazioni per ottenere il risultato, ma in generale raggiunge il risultato in meno tempo. Questo sarà confermato in seguito facendo la sperimentazione numerica

4.4.4 Sperimentazione numerica con gli algoritmi quadratici

Esempio 4.4. *L’obiettivo di questo esercizio è quello di verificare la convergenza quadratica dei quattro algoritmi di Newton, perciò, abbiamo preso gli stessi dati dell’esempio (4.1) della sezione (4.4.2).*

Siccome $\rho[\mathfrak{M}] = 0.8631 < 1$, quindi il processo è sub-critical e sappiamo che $q = 1$. Applicando i quattro algoritmi di Newton con una tolleranza di $\epsilon = 10^{-10}$ si hanno ottenuto i risultati che seguono.

Diamo nella tabella (4.3) gli errori in ogni iterazione per i quattro algoritmi di Newton, si vede chiaramente che l’algoritmo di Newton-Depth richiede 6 iterazioni per raggiungere la tolleranza specificata, mentre gli altri algoritmi soltanto 5 iterazioni. La figura (4.8a) mostra la velocità di convergenza per i quattro algoritmi. Inoltre, diamo su una scala logaritmica (base 10) l’errore $\|E_n\|_\infty = \|\mathbf{1} - q_n\|_\infty$ per i quattro algoritmi (fig. (4.8b)). La convergenza quadratica è molto chiara e, come avevamo previsto, l’algoritmo “*Newton-Depth*” richiede un numero maggiore o uguale di iterazioni rispetto agli altri algoritmi per raggiungere una tolleranza specifica.

	$E_n = \ \mathbf{1} - q_n\ _\infty$			
<i>Iter.</i>	<i>Newton – Depth</i>	<i>Newton – Order – 1</i>	<i>Newton – Order – 2</i>	<i>Newton – Thicknesses</i>
1	1.9527×10^{-1}	1.3881×10^{-1}	1.2840×10^{-1}	1.2840×10^{-1}
2	4.7634×10^{-2}	2.2482×10^{-2}	1.7220×10^{-2}	1.9857×10^{-2}
3	4.3971×10^{-3}	8.9516×10^{-4}	4.5892×10^{-4}	6.0755×10^{-4}
4	4.4917×10^{-5}	1.5650×10^{-6}	3.5114×10^{-7}	7.2191×10^{-7}
5	4.7830×10^{-9}	0	0	0
6	0	0	0	0

Tabella 4.3: Errori dei quattro algoritmi rispetto al numero di iterazioni.

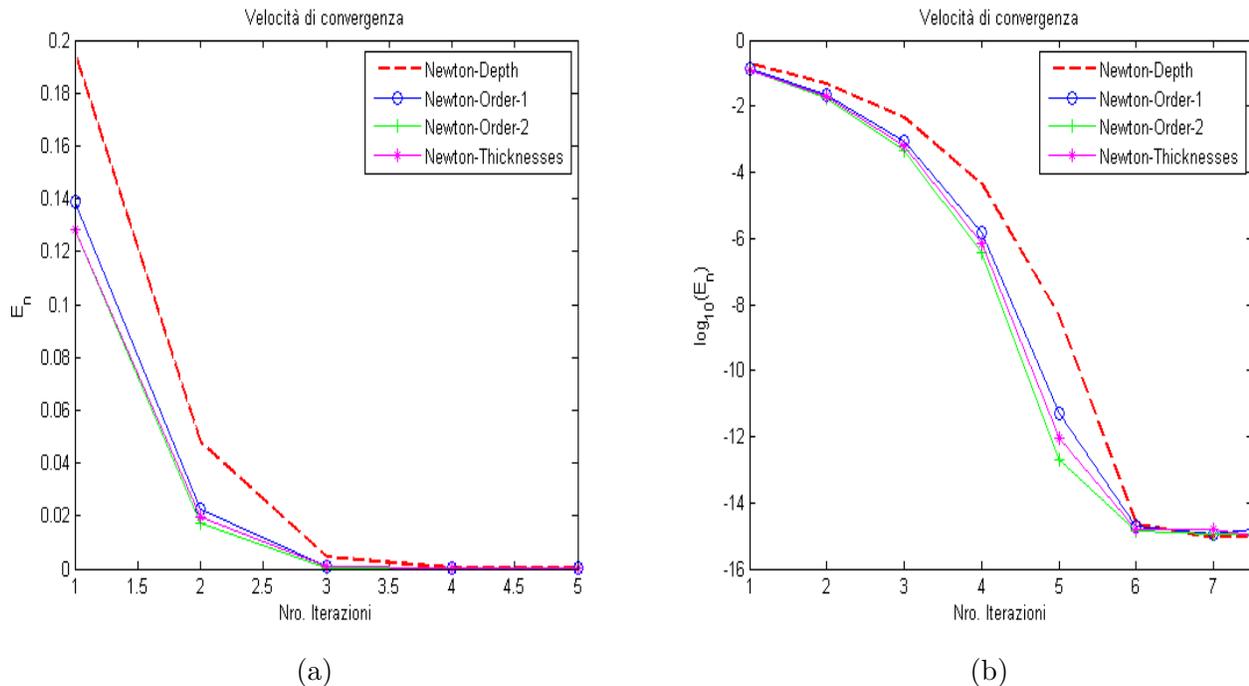


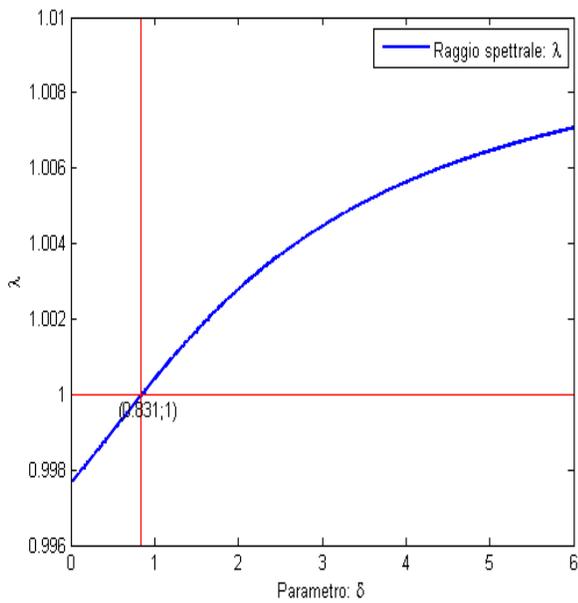
Figure 4.8: (a) Errori dei quattro algoritmi. (b) Logaritmi degli errori dei quattro algoritmi.

Esempio 4.5. In questo esercizio confronteremo il numero di iterazioni e il tempo di esecuzione dei quattro algoritmi di Newton quando il vettore di probabilità di estinzione per diversi MBTs viene calcolato. Per questo, usiamo i dati del esercizio (4.2) quando il parametro δ varia da 0 a 6.

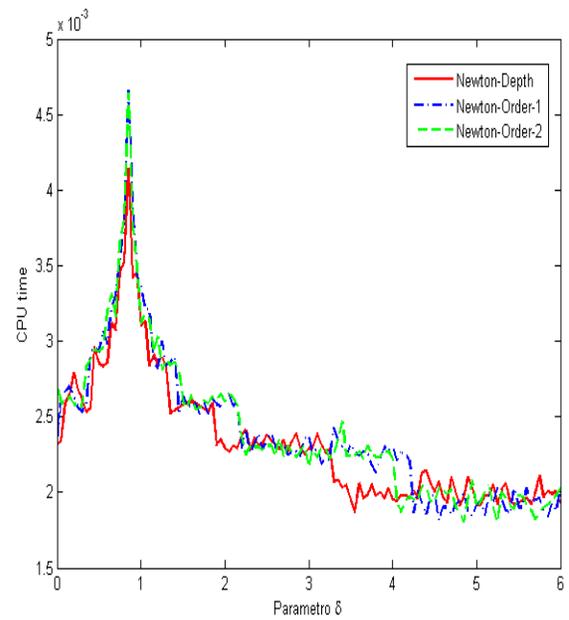
La figura (4.9a) mostra la variazione del raggio spettrale quando il parametro δ varia da 0 a 6, è chiaro che per $\delta = 0.83$ circa, il processo è *critical*, mentre per $\delta > 0.83$ il processo è *super-critical* e per $\delta < 0.83$ il processo è *sub-critical*.

Se guardiamo le figure (4.9b) e (4.10a), si noti che in questo caso, l'algoritmo Newton-Depth richiede un maggior numero di iterazioni per calcolare “ q ”, tuttavia, il tempo impiegato per lo generale è inferiore rispetto agli altri algoritmi di Newton; che avevamo già accennato nella sezione (4.4.3). Ma, in termini di tasso di convergenza espresso nel numero di iterazioni, l'algoritmo di Newton più veloce tra i quattro presentati sopra dipenderà dall'esempio in esame.

Un'altra osservazione da considerare è la seguente, in generale non possiamo supporre che se uno degli algoritmi lineari, diciamo A , è migliore di un altro, B , per un particolare esempio, allora il

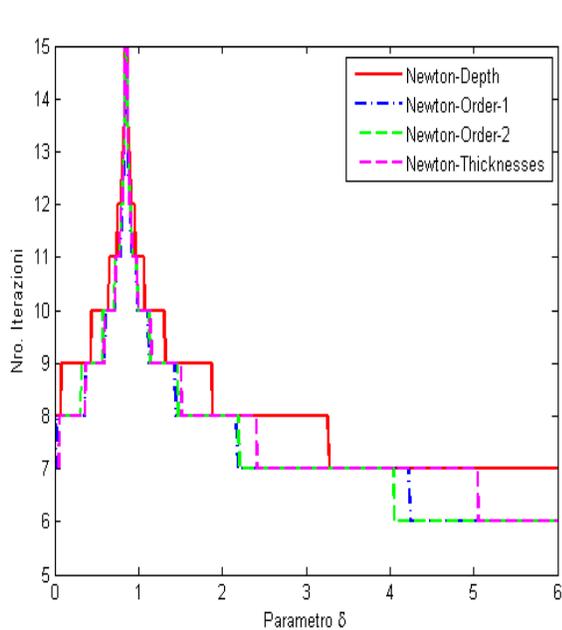


(a)

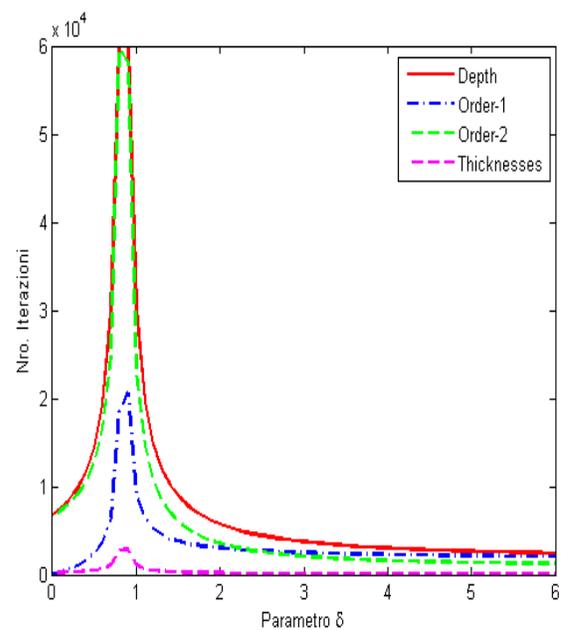


(b)

Figure 4.9: (a) Raggio spettrale di \mathfrak{M} quando $0 \leq \delta \leq 6$. (b) Tempo di CPU per calcolare q quando $0 \leq \delta \leq 6$ (l'algoritmo *Newton-Thicknesses* si comporta in modo simile all'algoritmo *Newton-Order*).



(a) N.I. per gli algoritmi quadratici.



(b) N.I. per gli algoritmi lineari.

Figure 4.10: Numero di iterazioni necessarie per calcolare q quando $0 \leq \delta \leq 6$.

corrispondente algoritmo di Newton-A convergerà più velocemente del corrispondente algoritmo Newton-B sullo stesso esempio. Questo è illustrato nelle figure (4.10a) e (4.10b), dove si vede che l'algoritmo Thicknesses è notevolmente il miglior algoritmo lineare per calcolare q , ma l'algoritmo Newton-Thicknesses non funziona così significativamente bene, è addirittura uguale o peggiore rispetto all'algoritmo di Newton-Depth in alcune occasioni.

A questo punto possiamo dedurre che gli algoritmi quadratici sono più efficienti degli algoritmi lineari, studieremo questo nel prossimo paragrafo.

4.4.5 Confronto tra gli algoritmi lineari e gli algoritmi quadratici

In questa sezione confrontiamo gli algoritmi di Newton con gli algoritmi lineari mediante almeno due esempi rappresentativi.

Esempio 4.6. *Mostreremo come i otto algoritmi si comportano su l'esempio (4.2) visto sopra . La struttura delle matrici D_0 , D_1 , P_1 e P_0 e del vettore d è riassunta in (4.2). M e N dipendono dal parametro δ .*

La Fig. (4.9a) mostra la variazione del massimo autovalore λ della matrice \mathfrak{M} e vediamo che il processo si muove da essere sub-critical a super-critical quando aumenta δ . Si noti che il processo è critical intorno a $\delta = 0.83$. La figura (4.11) rappresenta il numero di iterazioni necessarie per calcolare q con gli algoritmi di Newton in funzione di δ (figura (a)) e con gli algoritmi lineari (figura (b)). Vediamo che in tutti i casi gli algoritmi di Newton richiedono un numero di iterazioni notevolmente inferiore rispetto agli algoritmi lineari, inoltre, che il numero di iterazioni cresce in modo significativo quando il processo è quasi critico.

Ora confrontiamo la convergenza quando $\delta = 1$. La soluzione ottenuta è:

$$q = (0.9620 \quad 0.9626 \quad 0.9632 \quad 0.9639 \quad 0.9996 \quad 0.9434 \quad 0.9459 \quad 0.9483 \quad 0.9508) \quad (4.4.57)$$

Sia q_n la n -esima approssimazione di q , sappiamo che l'errore di approssimazione è data da

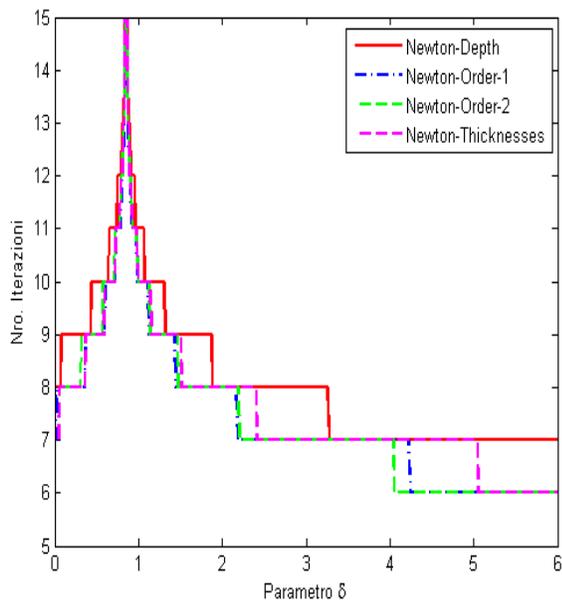
$$E_n = q - q_n,$$

poiché non conosciamo a priori il valore reale di q , non abbiamo il valore esatto di E_n . Invece, useremo la differenza $q^* - q_n$ dove q^* è la probabilità di estinzione ottenuta in (4.4.57) eseguendo l'algoritmo di Newton durante un numero enorme di passi (più di 10^4 in questo caso). Vediamo nella figura (4.12) che dopo alcune iterazioni, l'algoritmo di Newton (per i quattro tipi) è convergente: al di là di $n = 12$, la grafica riflette solo gli errori di arrotondamento. Dopo un po' più di 1600 iterazioni, l'errore per l'algoritmo Thicknesses è paragonabile a quello dello schema di Newton mentre gli altri algoritmi lineari richiedono più di 3000 iterazioni. Qui e in tutti gli esempi abbiamo usato $\|q_n - q_{n-1}\|_\infty$ come criterio di arresto.

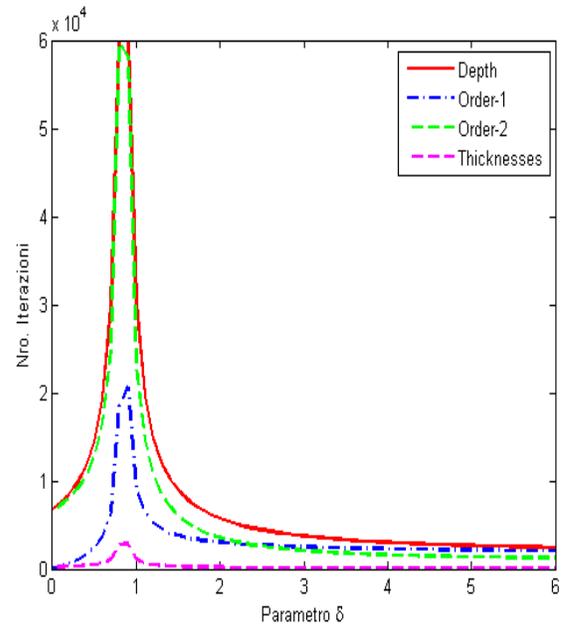
Sia q^* l'approssimazione di q ottenuta numericamente, diamo nella tabella (4.4) il numero di iterazioni (N.I.), il valore dei due residui

$$r_1 = \|q^* - M - N(q^* \otimes q^*)\|_\infty \quad e \quad r_2 = \frac{\|q^* - M - N(q^* \otimes q^*)\|_1}{(\|q^*\|_1 + \|M\|_1 + \|N(q^* \otimes q^*)\|_1)}.$$

e “CPU execution time” per i otto algoritmi (eseguiti con Matlab su un processore Core-i5 a 1.70 GHz con 4.00 GB di RAM).



(a) N.I. per gli algoritmi quadratici.



(b) N.I. per gli algoritmi lineari.

Figure 4.11: Numero di iterazioni necessarie per calcolare q quando $0 \leq \delta \leq 6$.

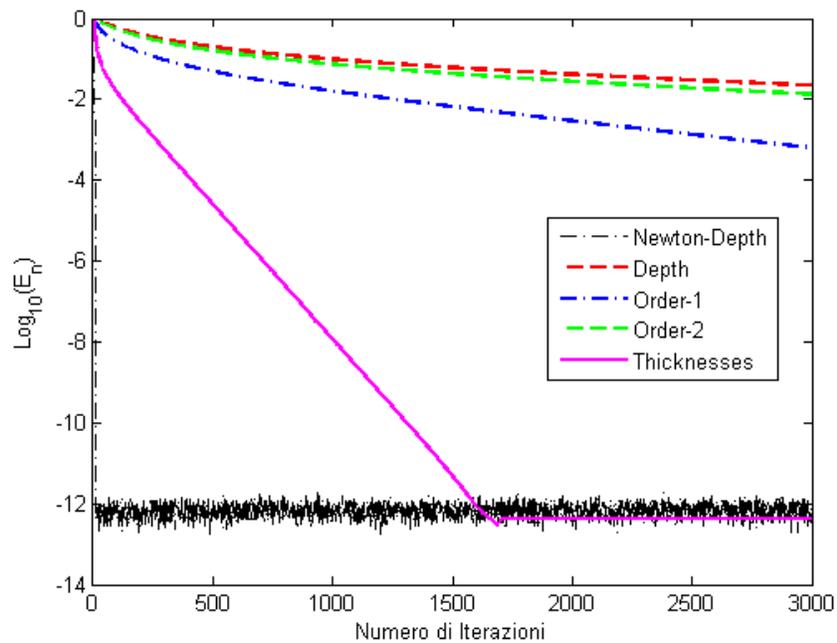


Figure 4.12: Logaritmi di $\|E_n\|_\infty$ per i cinque algoritmi con $\delta = 1$ (gli altri algoritmi di Newton hanno un comportamento simile al Newton-Depth).

<i>Algoritmo</i>	<i>N.I</i>	r_1	r_2	<i>CPUtime(s)</i>
<i>Depth</i>	30244	0.9994×10^{-10}	0.3679×10^{-10}	3.6940
<i>Order - 1</i>	9076	0.6850×10^{-10}	0.1525×10^{-10}	2.0070
<i>Order - 2</i>	22963	0.9926×10^{-10}	0.2203×10^{-10}	4.8640
<i>Thicknesses</i>	1082	0.0398×10^{-10}	0.0070×10^{-10}	0.2030
<i>Newton - Depth</i>	11	0.9770×10^{-13}	0.1371×10^{-13}	0.0050
<i>Newton - Order - 1</i>	10	0.0044×10^{-13}	0.0007×10^{-13}	0.0070
<i>Newton - Order - 2</i>	10	0.0022×10^{-13}	0.0005×10^{-13}	0.0080
<i>Newton - Thicknesses</i>	10	0.0022×10^{-13}	0.0005×10^{-13}	0.0060

Tabella 4.4: Confronto tra il numero di iterazioni, i residui e il tempo di esecuzione dei otto algoritmi con $\delta = 1$.

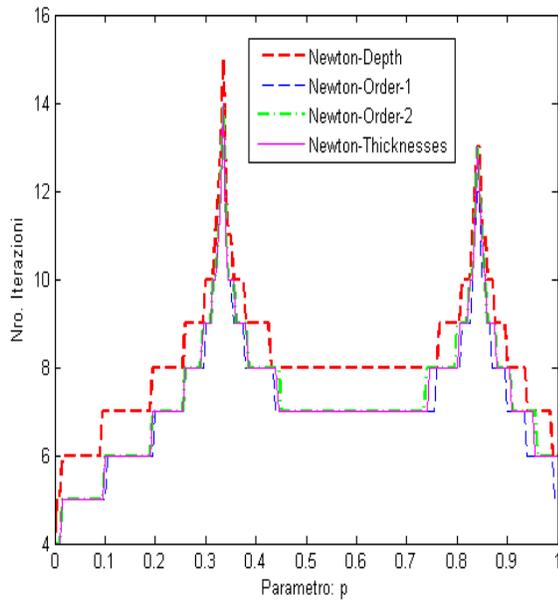
Esempio 4.7. *Per il nostro ultimo esempio, prendiamo le matrici di Kontoleon date in (4.3), la figura (4.6a) mostra la variazione del massimo autovalore λ della matrice \mathfrak{M} rispetto al parametro “ p ” quando varia da 0 a 1, si vede che il processo varia da essere sub-critical a critical, super-critical, critical ed infine sub-critical di nuovo.*

La figura (4.13) rappresenta il numero di iterazioni necessarie per calcolare q con gli algoritmi di Newton in funzione del parametro p (figura (a)) e con gli algoritmi lineari (figura (b)). Ancora una volta, vediamo che il numero di iterazioni richieste dagli algoritmi di Newton è ancora molto piccolo rispetto agli algoritmi lineari. Chiaramente, la convergenza degli algoritmi è sempre più lenta in quanto si avvicina ai casi critici (circa $p = 0,34$ e $p = 0,84$).

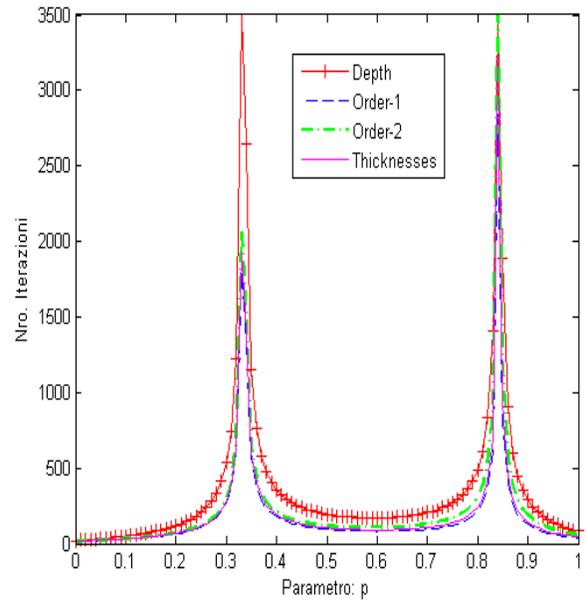
Nella tabella (4.5) confrontiamo il numero di iterazioni, i residui e il tempo di esecuzione dei tre algoritmi per il valore $p = 0.6$ dove il processo sembra essere il più super-critical.

Numerical complexity

Come osservazione finale notare che i due algoritmi lineari più efficienti (*Order* e *Thicknesses*) hanno una complessità numerica di $C_l(n) = (8/3)n^3 + 3n^2 + (4/3)n - 2$ flops per iterazione, dove “ n ” è la dimensione del vettore “ q ”, mentre l’algoritmo di Newton-Depth (il quale richiede generalmente più iterazioni) ha una complessità di $C_q(n) = (20/3)n^3 + 6n^2 + (7/3)n - 2$ flops per iterazione, che è maggiore di $c_l(n)$. Poiché il numero di iterazioni richieste con l’algoritmo di Newton è molto inferiore rispetto agli algoritmi lineari, il costo computazionale complessivo degli algoritmi di Newton è inferiore rispetto agli algoritmi lineari, come è stato dimostrato dai tempi di CPU riportati nelle tabelle (4.4) e (4.5).



(a) N.I. per gli algoritmi quadratici.



(b) N.I. per gli algoritmi lineari.

Figure 4.13: Numero di iterazioni necessarie per calcolare q per $0 \leq p \leq 1$.

<i>Algoritmo</i>	<i>N.I</i>	r_1	r_2	<i>CPUtime(s)</i>
<i>Depth</i>	169	0.8887×10^{-10}	0.3698×10^{-10}	0.0200
<i>Order - 1</i>	81	0.4381×10^{-10}	0.1601×10^{-10}	0.0160
<i>Order - 2</i>	106	0.4933×10^{-10}	0.1847×10^{-10}	0.0190
<i>Thicknesses</i>	92	0.3194×10^{-10}	0.1061×10^{-10}	0.0150
<i>Newton - Depth</i>	8	0.2665×10^{-14}	0.8512×10^{-15}	0.0030
<i>Newton - Order - 1</i>	7	0.0069×10^{-14}	0.0138×10^{-15}	0.0040
<i>Newton - Order - 2</i>	7	0.0111×10^{-14}	0.0276×10^{-15}	0.0040
<i>Newton - Thicknesses</i>	7	0.0222×10^{-14}	0.0774×10^{-15}	0.0020

Tabella 4.5: Confronto tra il numero di iterazioni, i residui e il tempo di esecuzione dei otto algoritmi con $\delta = 1$.

Capitolo 5

Analisi transitorio dei Markovian binary trees

Nel capitolo precedente abbiamo studiato la probabilità di estinzione del *MBT*, tuttavia, ci sono altre caratteristiche transitorie importanti che possono essere derivati dall'analisi del processo in evoluzione, ad esempio: la dimensione della popolazione per un certo tempo finito, la distribuzione di probabilità del tempo fino all'estinzione, ecc. In questo capitolo studieremo queste caratteristiche che saranno formulati come equazioni differenziali per le funzioni generatrici di probabilità. Bisogna però notare che queste caratteristiche possono essere ricavati direttamente dall'analisi dei *ctMMTBPs*, nonostante, si ricava qui nel contesto dei *MBTs* per presentare un'immagine coerente. Iniziamo richiamando alcune proprietà della derivazione di matrici che sono necessari per ottenere espressioni di vari momenti fattoriali per le funzioni generatrici di probabilità.

5.1 Derivati di matrici

Le regole riportate di seguito sono ispirate dalle regole per le derivati di una matrice descritte da MacRae ([42]). Sia “ s ” un vettore di dimensione $n \times 1$ e sia d/ds^T un vettore riga di operatori derivate $[d/ds_1, \dots, d/ds_n]$.

Definizione 5.1 (Matrix differentiation). *Se Y è una matrice di dimensione $p \times q$ dove i suoi elementi sono funzioni del vettore “ s ”, allora, la derivata di Y rispetto a s^T è la matrice delle derivate parziali dY/ds^T di dimensione $p \times nq$, data da*

$$\frac{dY}{ds^T} = Y \otimes d/ds^T,$$

con la comprensione che $Y_{ij} \cdot d/ds_k = \partial Y_{ij} / \partial s_k$. □

Notare che $ds/ds^T = I_n$, dove I_n indica la matrice identità di dimensione n . Diamo ora tre teoremi generali che verranno utilizzati nelle sezioni successive per ottenere i momenti fattoriali.

Teorema 5.1 (Sum Rule). *Siano Y e Z funzioni matriciali di s , tale che è definita la somma tra di loro, allora,*

$$\frac{d(Y + Z)}{ds^T} = \frac{dY}{ds^T} + \frac{dZ}{ds^T}.$$

Teorema 5.2 (Product Rule). *Siano Y e Z funzioni matriciali di s , tale che è definita il prodotto tra di loro, allora,*

$$\frac{d(YZ)}{ds^T} = \frac{dY}{ds^T}(Z \otimes I_n) + Y \frac{dZ}{ds^T}.$$

Per dare la regola per i derivati dei prodotti Kronecker, dobbiamo introdurre la “*permuted identity matrix*” $I_{(m,n)}$.

Definizione 5.2. *La “permuted identity matrix” $I_{(m,n)}$ è una matrice quadrata di dimensione mn partizionata in $m \times n$ sotto-matrici tale che la (i, j) – esima sotto-matrice ha il valore di 1 nella sua (j, i) – esima posizione e zeri altrove.*

Per esempio,

$$I_{(2,2)} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right)$$

Le seguenti identità possono essere verificate mediante un esame diretto: $I_{(m,1)} = I_{(1,m)} = I_m$, $I_{(m,n)} = I_{(n,m)}^T$, e $I_{(m,n)} \cdot I_{(n,m)} = I_{mn}$.

La “*permuted identity matrix*” può essere anche utilizzata per invertire l’ordine di un prodotto Kronecker: se A e B sono matrici di dimensioni $m \times n$ e $p \times q$ risp., allora,

$$B \otimes A = I_{(m,p)}(A \otimes B)I_{(q,n)}.$$

Teorema 5.3 (Kronecker Product Rule). *Se Y e Z sono matrici di dimensioni $s \times t$ e $p \times q$ rispettivamente, allora,*

$$\begin{aligned} \frac{d(y \otimes Z)}{ds^T} &= Y \times \frac{dZ}{ds^T} + I_{(p,s)} \left(Z \times \frac{dY}{ds^T} \right) (I_{(t,q)} \otimes I_n) \\ &= Y \times \frac{dZ}{ds^T} + \left(\frac{dY}{ds^T} \times Z \right) I_{(q,t)} (I_{(t,q)} \otimes I_n). \end{aligned}$$

5.2 Dimensione della Popolazione al tempo T

Supponiamo che $Z(t) = [Z_1(t), Z_2(t), \dots, Z_n(t)]^T$ sia il vettore “*population size*” al tempo t in un MBT, dove $Z_i(t)$ è il numero di rami nella fase “ i ” al tempo t . Sia ϕ_0 la fase iniziale del MBT.

La funzione generatrice di probabilità vettoriale $F(s, t)$, data la fase del primo individuo, si definisce come

$$F_i(\mathbf{s}, t) = \sum_{\mathbf{k} \geq 0} \mathbb{P}\{Z(t) = \mathbf{k} | \phi_0 = i\} \mathbf{s}^{\mathbf{k}}, \quad (5.2.1)$$

dove $\mathbf{s}^{\mathbf{k}} = s_1^{k_1} s_2^{k_2} \dots s_n^{k_n}$, e $|s_i| < 1$ per ogni i . In appresso, a volte scriveremo $F(\mathbf{s}, t) = \sum_{\mathbf{k} \geq 0} \mathbb{P}_{\phi_0}\{Z(t) = \mathbf{k}\} \mathbf{s}^{\mathbf{k}}$. Si noti che $F(\mathbf{1}, t) = \mathbf{1}$, dato che in ogni tempo finito t , la popolazione totale è sempre finita. Abbiamo già visto in (1.4) che la funzione generatrice di probabilità $F(\mathbf{s}, t)$ soddisfa il sistema di equazioni *backward and forward Kolmogorov* (per ulteriori dettagli vedere [4] e [19]), quindi diamo le espressioni di questi sistemi di equazioni differenziali nel caso speciale del MBT.

Teorema 5.4. Le equazioni “*forward and backward Kolmogorov*” per la funzione generatrice $F(\mathbf{s}, t)$ di un MBT sono rispettivamente

$$\frac{\partial}{\partial t} F(\mathbf{s}, t) - \frac{\partial}{\partial \mathbf{s}^T} F(\mathbf{s}, t) \cdot a(\mathbf{s}) = \mathbf{0}, \quad (5.2.2)$$

dove $\partial/\partial \mathbf{s}^T F(\mathbf{s}, t) = F(\mathbf{s}, t) \otimes \partial/\partial \mathbf{s}^T$, e

$$\frac{\partial}{\partial t} F(\mathbf{s}, t) = a(F(\mathbf{s}, t)), \quad (5.2.3)$$

con $F(\mathbf{s}, t) = \mathbf{s}$ e $a(x) = d + D_0 x + B(x \otimes x)$. □

Le dimostrazioni di questo teorema e quelli che seguono in questo capitolo si possono trovare in ([25],[24]). Nel caso $n \geq 2$, nessuna delle equazioni differenziali matriciali (parziali) (5.2.2) e (5.2.3) può essere risolta esplicitamente. Tuttavia, esistono potenti strumenti numerici per risolvere (5.2.3) come la funzione *ode45* in MATLAB, che si basa su una formula esplicita di Runge-Kutta (4,5). Ulteriori dettagli si trovano in Dormand e Prince [16]. Siamo quindi in grado di valutare numericamente $F(\mathbf{s}, t)$ per ogni coppia (\mathbf{s}, t) . Allo stesso modo, possiamo anche calcolare la funzione generatrice

$$F(\mathbf{s}, t) = \sum_{k \geq 0} \mathbb{P}\{Z(t) \cdot \mathbf{1} = k | \phi_0\} s^k \quad (5.2.4)$$

del “*total population size*” al tempo t risolvendo numericamente (5.2.3) con $F(\mathbf{s}, 0) = \mathbf{s} \cdot \mathbf{1}$ come condizione iniziale (notare che qui, s è scalare).

Per ottenere la distribuzione del “*total population size*” in un momento dato, possiamo usare tecniche numeriche per l’inversione di funzioni generatrici di probabilità, ad esempio l’algoritmo proposto da Abate e Whitt [1], che si basa sulla regola trapezoidale e una serie di Fourier per il “*error bound*”, e che brevemente presentiamo adesso.

Siano “ t ” un valore fisso ed “ i ” la radice complessa di -1 con $Re(z)$ che denota la parte reale di z . Il seguente teorema fornisce un semplice algoritmo, con “*error bound*”, per approssimarsi alle probabilità $q_t(k) = \mathbb{P}_{\phi_0}\{Z(t) \cdot \mathbf{1} = k\}$, per $k \geq 1$, dove $q_t(0)$ viene calcolato come $F(0, t)$.

Teorema 5.5. Per $0 < r < 1$ e $k \geq 1$,

$$|q_t(k) - \tilde{q}_t(k)| \leq \frac{r^{2k}}{1 - r^{2k}},$$

dove

$$\begin{aligned} \tilde{q}_t(k) &= \frac{1}{2kr^k} \sum_{j=1}^{2k} (-1)^j Re \left[F(re^{\pi j i/k}, t) \right] \\ &= \frac{1}{2kr^k} \left\{ F(r, t) + (-1)^k F(-r, t) + 2 \sum_{j=1}^{k-1} Re[F(re^{\pi j i/k}, t)] \right\}. \end{aligned}$$

In questo teorema, “*the bound*” $r^{2k}/(1 - r^{2k})$ è approssimativamente uguale a r^{2k} quando r^{2k} è piccolo. Quindi, per ottenere una precisione del $10^{-\gamma}$, prendiamo $r = 10^{-\gamma/2k}$.

Osservazione:

Il teorema (5.5) si basa sull’ipotesi che la funzione generatrice $F(\mathbf{s}, t)$ sia esattamente conosciuta. Poiché è calcolato numericamente con alcuni errori di stima, esiste il rischio che l’errore aumenti quando $F(\cdot, t)$ viene invertito. Anzi, spesso si ottengono scarsi risultati.

5.2.1 Momenti Fattoriali

Il k -esimo momento fattoriale $M^{(k)}(t)$ è una matrice di dimensione $n \times n^k$ data da

$$\begin{aligned} M^{(k)}(t) &= \frac{\partial^k}{(\partial s^T)^k} F(\mathbf{s}, t)|_{s=1} \\ &= F(\mathbf{s}, t) \otimes \underbrace{\frac{\partial}{\partial s^T} \otimes \cdots \otimes \frac{\partial}{\partial s^T}}_{\text{"k" volte}} |_{s=1}. \end{aligned}$$

Definiamo ora la matrice

$$\Omega = D_0 + B(\mathbf{1} \oplus \mathbf{1}). \quad (5.2.5)$$

I momenti fattoriali vengono forniti ricorsivamente dalla seguente proposizione.

Teorema 5.6. *Le matrici $M^{(k)}(t)$, $k \geq 1$, sono le soluzioni di due equazioni differenziali matriciali ricorsive, con condizioni iniziali $M^{(1)}(0) = I_n$ e $M^{(k)}(0) = 0$ per $k \geq 2$, date da,*

- Il primo sistema è

$$\frac{d}{dt} M^{(k)}(t) = M^{(k)}(t)(\Omega \otimes I_{n^{k-1}})A(k) + M^{(k-1)}(t)[B(I_{n^2} + I_{(n,n)}) \otimes I_{n^{k-2}}]C(k), \quad (5.2.6)$$

dove Ω è definita in (5.2.5) e le $n^k \times n^k$ matrici di coefficienti $A(k)$ e $C(k)$ sono definite ricorsivamente come

$$\begin{aligned} A(1) &= I_n \\ C(1) &= 0_n \\ A(k) &= I_{(n^{k-1}, n)} + A(k-1) \otimes I_n \\ C(k) &= I_{(n^{k-2}, n^2)} [I_{(n, n^{k-2})} \otimes I_n] [A(k-1) \otimes I_n] + C(k-1) \otimes I_n \end{aligned} \quad (5.2.7)$$

per $k \geq 2$.

- Il secondo sistema è

$$\frac{d}{dt} M^{(k)}(t) = \Omega M^{(k)}(t) + \sum_{i=1}^{k-1} B[M^{(i)}(t) \otimes M^{(k-i)}(t)]C(i, k-i), \quad (5.2.8)$$

dove una somma vuota è zero e la matrice di coefficienti $C(i, k-i)$ è definita ricorsivamente come

$$\begin{aligned} C(0, k-i) &= I_{n^{k-i}} \\ C(i, 0) &= I_{n^i} \\ C(i, k-i) &= I_{(n^{k-i}, n^i)} [I_{(n^{i-1}, n^{k-i})} C(i-1, k-i) \otimes I_n] + C(i, k-i-1) \otimes I_n. \end{aligned} \quad (5.2.9)$$

Osservazione:

La matrice di coefficienti $C(i, k-i)$ in (5.2.8) sono matrici che generalizzano i coefficienti binomiali $\binom{k}{i}$

La dimensione delle matrici $M^{(k)}(t)$ aumenta rapidamente con k , che in qualche modo limita l'utilità del teorema (5.6) al di là dei primi momenti, ma la "mean" e la "standard deviation" sono facilmente ottenuti, come mostriamo ora.

Corollario 5.1. *I primi due momenti sono dati da*

$$M^{(1)} = e^{\Omega t} \quad (5.2.10)$$

$$M^{(2)} = X(t)[I_{n^2} + I_{(n,n)}] \quad (5.2.11)$$

dove $X(t)$ è la soluzione dell'equazione di Lyapunov

$$X(t)[\Omega \oplus \Omega] - \Omega X(t) + e^{\Omega t} B - B[e^{\Omega t} \otimes e^{\Omega t}] = 0. \quad (5.2.12)$$

L'equazione di Lyapunov (5.2.12) si può risolvere usando la funzione *lyap* in MATLAB. Questo metodo utilizza la decomposizione triangolare, quindi la complessità è $\mathcal{O}(n^3)$. La deviazione standard del “total population size” al tempo t , data la fase iniziale, è ottenuta come

$$\sigma_M(t) = \left[M^{(2)}(t)\mathbf{1} + \text{diag}(M^{(1)}(t)\mathbf{1})[\mathbf{1} - M^{(1)}\mathbf{1}] \right]^{1/2}. \quad (5.2.13)$$

Osservazione:

La matrice Ω , che è la stessa matrice A definita in (4.2.7) nel capitolo precedente, svolge un ruolo simile a “the mean progeny matrix” \mathcal{M} . Infatti, dato che il primo momento della “population size” al tempo t è dato da $e^{\Omega t}$, si vede che l'autovalore di parte reale maximal $\mu(\Omega)$ determina la natura del MBT: se $\mu(\Omega) < 0$, allora in media, la dimensione della popolazione tende a zero quando il tempo aumenta e pertanto il MBT è *sub-critical*; se $\mu(\Omega) = 0$, allora in media, la dimensione della popolazione è limitata, diverso da zero e pertanto il MBT è *critical*; e infine, se $\mu(\Omega) > 0$, allora in media, la dimensione della popolazione cresce senza limiti e pertanto il MBT è *super-critical*.

5.2.2 Dimensione della Popolazione condizionata all'estinzione

Finiamo la sezione osservando la “conditional distribution of the population size”, dato che si verifichi l'estinzione. Supponiamo che E indica l'evento di una totale estinzione, la funzione generatrice di probabilità della “conditional population size” è definita come

$$F_e(\mathbf{s}, t) = \sum_{k \geq 0} \mathbb{P}\{Z(t) = \mathbf{k} | E, \phi_0\} \mathbf{s}^{\mathbf{k}}. \quad (5.2.14)$$

Definiamo anche

$$\Phi = \text{diag}(q). \quad (5.2.15)$$

L'inversa di Φ esiste, sempre che $q_i \neq 0$. Il seguente risultato è stato già stabilito da Athreya e Ney nel caso del processo di Galton-Watson, (vedere [4], cap. I.12, teorema 3).

Proposizione 5.1. *La distribuzione della dimensione della popolazione (“population size”) con la condizione che si verifichi l'estinzione è data da*

$$F_e(\mathbf{s}, t) = \Phi^{-1} F(\Phi \mathbf{s}, t). \quad (5.2.16)$$

Definiamo la seguente matrice

$$\Theta = D_0 + B(q \oplus q), \quad (5.2.17)$$

che è uguale alla matrice Ω definita sopra quando $q = 1$, cioè, nei casi *sub-critical* e *critical*.

Corollario 5.2. *La matrice di “conditional mean population size” al tempo t dato che si verifichi l’estinzione è data da*

$$M_e^{(1)}(t) = \Phi^{-1} e^{\Theta t} \Phi \quad (5.2.18)$$

Il risultato successivo mostra che, nel caso irriducibile e *non-critical* esiste sempre l’inverso di Θ .

Proposizione 5.2. *Se il MBT è irriducibile e non-critical, allora tutti gli autovalori di Θ hanno parte reale strettamente negativa. Nel caso critical, Θ è singolare.*

5.3 Tempo fino all’estinzione: T_e

Supporremo qui che non siamo nel caso critico. Sia $F(t) = \mathbb{P}\{T_e \leq t | \phi_0\}$ la funzione di distribuzione di T_e data la fase iniziale, e osserviamo che la probabilità di estinzione q è in realtà il limite di $F(t)$ quando t tende all’infinito. È chiaro che $T_e \leq t$ se e solo se $Z(t) = 0$ in modo che $F(t) = F(s, t)|_{s=0}$. Quindi, prendendo $s = 0$ in (5.2.3), si vede che la funzione di distribuzione $F(t)$ soddisfa il sistema differenziale

$$\frac{d}{dt} F(t) = d + D_0 F(t) + B[F(t) \otimes F(t)], \quad (5.3.1)$$

con $F(0) = 0$.

Questa è un’equazione differenziale matriciale quadratica che non può essere risolta esplicitamente quando $n \geq 2$, ma gli stessi strumenti numerici usati per la funzione generatrice $F(s, t)$ ci permettono di approssimare la funzione in alcuni punti discreti.

Se $q < 1$, allora T_e è infinito con una probabilità positiva e il suo valore atteso è infinito. Per studiare una quantità significativa, definiamo M_e come il valore atteso condizionale di T_e , data la fase iniziale e dato che l’estinzione accada:

$$M_e = \Phi^{-1} \int_0^\infty [q - F(u)] du = \int_0^\infty [1 - \Phi^{-1} F(u)] du \quad (5.3.2)$$

dove Φ è data in (5.2.15).

Per calcolare M_e , procediamo come segue: fissiamo un certo tempo T^* tale che

$$\|1 - \Phi^{-1} F(T^*)\| < \epsilon \quad (5.3.3)$$

dove ϵ è arbitrariamente piccolo e (5.3.2) lo possiamo scrivere come

$$M_e = \int_0^{T^*} [1 - \Phi^{-1} F(u)] du + \int_{T^*}^\infty [1 - \Phi^{-1} F(u)] du. \quad (5.3.4)$$

- Il primo integrale può essere approssimato usando la regola trapezoidale: scegliamo un numero k^* di intervalli e scriviamo

$$\int_0^{T^*} [1 - \Phi^{-1} F(u)] du = h \left\{ 1 - \Phi^{-1} \frac{[F(0) + F(T^*)]}{2} + \sum_{i=1}^{k^*-1} [1 - \Phi^{-1} F(ih)] \right\} + E, \quad (5.3.5)$$

dove $h = T^*/k^*$ e $E = T^* h^2 \Phi^{-1} F''(c)/12$ per alcun $c \in \langle 0, T^* \rangle$ è l’errore di approssimazione. In pratica, scegliamo ϵ e h e calcoliamo $F(ih)$ per i valori successivi di i finché $F(k^*h)$ soddisfi la disuguaglianza (5.3.3).

- Il secondo integrale può essere calcolato usando una funzione $\tilde{F}(t)$ che approssima $F(t)$ per i valori grandi di t , cioè, per i valori di t tale che $F(t) \approx q$. Questa approssimazione è data da

$$\tilde{F}(t) = [I - e^{\Theta t}]q, \quad (5.3.6)$$

dove Θ è stato definito in (5.2.17) e viene ottenuto come segue. Definiamo $\varrho(t) = q - F(t)$. Il vettore q soddisfa

$$0 = d + D_0q + B(q \otimes q), \quad (5.3.7)$$

sottraendo questa ultima equazione da (5.3.1), otteniamo

$$\begin{aligned} \varrho'(t) &= D_0\varrho(t) + B[(q \otimes q) - (F(t) \otimes F(t))] \\ &= D_0\varrho(t) + B[(q - F(t) \otimes q) + (F(t) \otimes q - F(t))] \\ &= D_0\varrho(t) + B[(I \otimes q) + (F(t) \otimes I)]\varrho(t). \end{aligned}$$

Ora, se t è abbastanza grande, allora $F(t)$ può essere sostituito da q nell'equazione sopra e otteniamo il sistema approssimativo $\tilde{\varrho}'(t) = \Theta\tilde{\varrho}(t)$, che porta a

$$\tilde{\varrho}(t) = e^{\Theta t}q, \quad (5.3.8)$$

dato che $\varrho(0) = q$ per definizione e questo ci dà (5.3.6).

Pertanto, il secondo integrale può essere approssimato da

$$\int_{T^*}^{\infty} [1 - \Phi^{-1}\tilde{F}(u)]du = \int_{k^*h}^{\infty} \Phi^{-1}e^{\Theta u}qdu = \Phi^{-1}[-\Theta]^{-1}e^{\Theta k^*h}q \quad (5.3.9)$$

che è ben definita nel caso *non-critical* per la Proposizione (5.3).

Si noti che l'espressione $\Phi^{-1}e^{\Theta u}$ in (5.3.9) non è altro che “*the conditional mean total population size*” al tempo u , dato che l'estinzione accada, cioè, questo è $M_e^{(1)}(u)\mathbf{1}$ come indicato dal Corollario (5.2).

In sintesi, “*the mean time until extinction*”, dato che si verifichi l'estinzione, può essere approssimato da

$$\tilde{M}_e = h \left\{ 1 - \Phi^{-1} \frac{[F(0) + F(T^*)]}{2} + \sum_{i=1}^{k^*-1} [1 - \Phi^{-1}F(ih)] \right\} + \Phi^{-1}[-\Theta]^{-1}e^{\Theta T^*}q. \quad (5.3.10)$$

5.4 Dimensione totale della progenie

Si può dimostrare che la funzione generatrice di probabilità $g(s)$ del “*total progeny until extinction*” soddisfa l'equazione funzionale $g(s) = sP(g(s))$, dove $P(\cdot)$ è la funzione generatrice del progenie. In linea di principio, si può invertire $g(s)$ con la formula di inversione di Lagrange, ma i risultati forniti nel caso multi-type sono difficilmente implementabili, quindi qui si segue un approccio diverso che fa uso delle caratteristiche specifiche dei MBT.

Sia $g(k, t) = \mathbb{P}\{N(t) = k | \phi_0\}$ che indica la probabilità che un totale di k individui sono nati prima del tempo t , dato la fase iniziale del primo individuo ϕ_0 .

Proposizione 5.3. *Le probabilità $g(k, t)$ sono date ricorsivamente da*

$$\begin{aligned}
g(0, t) &= \mathbf{0} \\
g(1, t) &= e^{D_0 t} \mathbf{1} + \int_0^t e^{D_0 u} \mathbf{d} du, \\
g(k, t) &= \int_0^t e^{D_0 u} B \cdot \sum_{i=1}^{k-1} [g(i, t-u) \otimes g(k-i, t-u)] du.
\end{aligned} \tag{5.4.1}$$

per $k \geq 2$, $t \geq 0$.

Possiamo esprimere la funzione generatrice di probabilità di $N(t)$, $G(z, t) = \sum_{k \geq 0} g(k, t) z^k$ come soluzione di un'equazione differenziale utilizzando le espressioni ricorsive della proposizione (5.3) e differenziando rispetto a t :

$$\frac{\partial}{\partial t} G(z, t) = dz + D_0 G(z, t) + B[G(z, t) \otimes G(z, t)], \tag{5.4.2}$$

con $G(z, 0) = z$, Notare che $G(1, t) = \mathbf{1}$ per ogni t finito.

Potremmo risolvere numericamente (5.4.2) e prendere la trasformazione inversa, come è suggerito per la funzione generatrice $F(s, t)$ del “total population size” al tempo t ; qui, tuttavia, la distribuzione del “total progeny size” può essere calcolato direttamente da (5.4.1).

Sia $D^{(k)} = \partial^k G(z, t) / (\partial z)^k |_{z=1}$ il k -esimo momento fattoriale (vettore di dimensione $n \times 1$) di $N(t)$. Questi momenti sono caratterizzati ricorsivamente come segue,

Proposizione 5.4. *I vettori $D^{(k)}(t)$ soddisfano la seguente ricorrenza*

$$\begin{aligned}
\frac{d}{dt} D^{(1)}(t) &= \Omega D^{(1)}(t) + d, \\
\frac{d}{dt} D^{(k)}(t) &= \Omega D^{(k)}(t) + \sum_{i=1}^{k-1} \binom{k}{i} B[D^{(i)}(t) \otimes D^{(k-i)}(t)],
\end{aligned} \tag{5.4.3}$$

per $k \geq 2$, con $D^{(1)}(0) = \mathbf{1}$ e $D^{(k)}(0) = \mathbf{0}$ per $k \geq 2$.

Si ottiene facilmente un'espressione esplicita per il primo momento data da

$$D^{(1)}(t) = e^{\Omega t} \left[\int_0^t e^{-\Omega u} du \mathbf{d} + \mathbf{1} \right]. \tag{5.4.4}$$

Se l'inverso di Ω esiste, allora (5.4.4) è equivalente a

$$D^{(1)}(t) = [I - e^{\Omega t}] (-\Omega)^{-1} \mathbf{d} + e^{\Omega t} \mathbf{1}. \tag{5.4.5}$$

I momenti superiori non possono essere scritti esplicitamente, ma possono essere facilmente calcolati numericamente. Il secondo momento fattoriale per esempio, è dato da

$$D^{(2)}(t) = e^{\Omega t} \left(\int_0^t e^{-\Omega u} 2B[D^{(1)}(u) \otimes D^{(1)}(u)] du \right). \tag{5.4.6}$$

Ora rivolgiamo la nostra attenzione alla distribuzione asintotica di $N(t)$ quando t tende all'infinito. Definiamo $g(k) = \lim_{t \rightarrow \infty} g(k, t)$. Possiamo ottenere facilmente da (5.3) che

$$\begin{aligned} g(0) &= \mathbf{0} \\ g(1) &= \theta \\ g(k) &= \psi \sum_{i=1}^{k-1} [g(i) \otimes g(k-i)], \quad \text{per } k \geq 2. \end{aligned} \quad (5.4.7)$$

La funzione generatrice $G(z) = \sum_{k \geq 0} g(k)z^k$, è tale che $G(1) = q$ e per ogni z , cioè soddisfa l'equazione funzionale del punto fisso

$$G(z) = \theta z + \psi[G(z) \otimes G(z)], \quad (5.4.8)$$

che ha la stessa struttura dell'equazione di estinzione (4.4.1).

L'equazione (5.4.8) ha la seguente interpretazione. Condizionata rispetto al primo evento osservabile: o il primo individuo muore prima di generare un discendente, che si verifica con la probabilità θ e $N(\infty) = 0$, o genera un discendente con probabilità ψ , e la progenie totale è la somma dei progeneri totali nei "sub-trees" generati da un figlio e da un genitore dopo della nascita di questo figlio, entrambi si evolvono in modo indipendente.

Le derivate $D^{(k)} = d^k G(z)/(dz)^k|_{z=1}$ per $k \geq 1$ sono qui uguali ai vettori momenti fattoriali della progenie totale $N(\infty)$ sulle traiettorie in cui si verificano l'estinzione. Differenziando (5.4.8), è facile dimostrare che,

$$\begin{aligned} D^{(1)} &= (-\Theta)^{-1} \mathbf{d}, \\ D^{(k)} &= (-\Theta)^{-1} \sum_{i=1}^{k-1} \binom{k}{i} B[D^{(i)} \otimes D^{(k-i)}], \end{aligned} \quad (5.4.9)$$

per $k \geq 2$, dove Θ è definita in (5.2.17) e a condizione che il MBT sia non-critical.

Qui, i momenti possono essere espressi esplicitamente. Per esempio, il secondo momento fattoriale è dato da

$$D^{(2)} = 2(-\Theta)^{-1} B[(-\Theta)^{-1} \mathbf{d} \otimes (-\Theta)^{-1} \mathbf{d}] \quad (5.4.10)$$

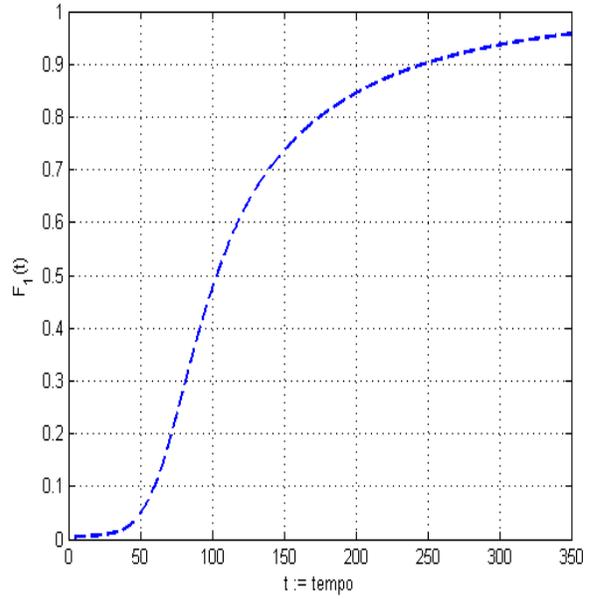
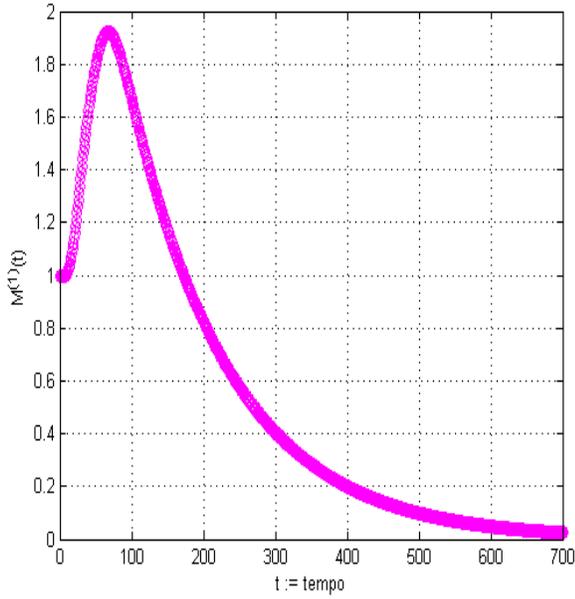
ma queste espressioni diventano rapidamente molto ingombranti.

È più naturale considerare la distribuzione condizionale di $N(\infty)$ dato che si verifichi l'estinzione; è sufficiente quindi moltiplicare per Φ^{-1} le espressioni ottenute finora. Otteniamo ad esempio, che "the conditional mean total progeny size", dato che si verifichi l'estinzione, è

$$D_e^{(1)} = \Phi^{-1}(-\Theta)^{-1} \mathbf{d} = -(\Theta\Phi)^{-1} \mathbf{d}, \quad (5.4.11)$$

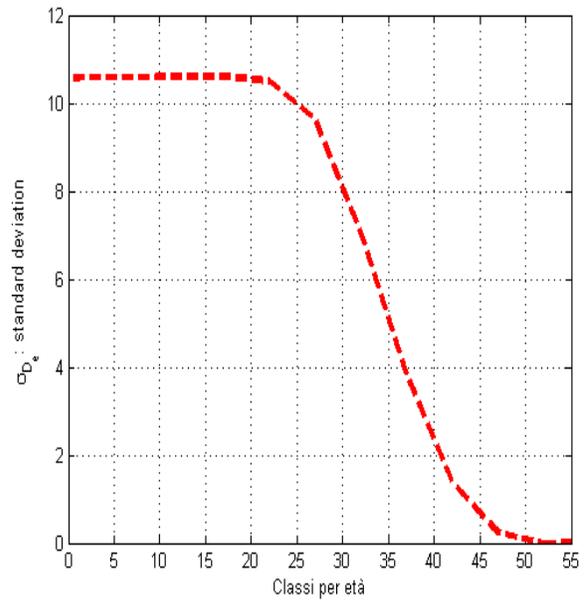
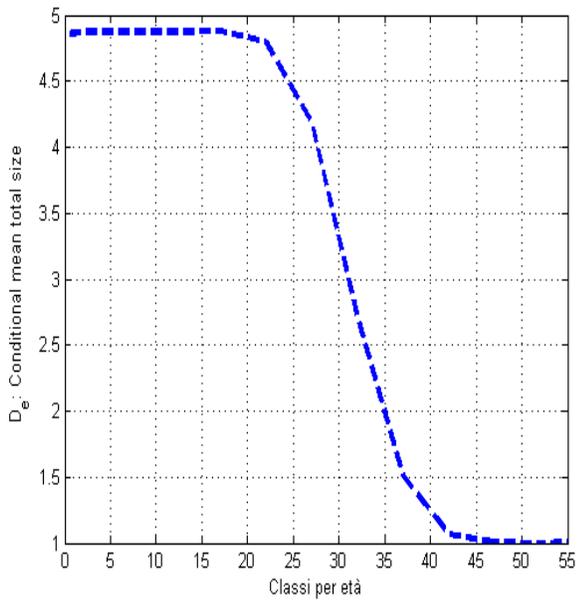
e la "conditional standard deviation" è

$$\sigma_{D_e} = \left[\Phi^{-1} D^{(2)} + \text{diag}(D_e^{(1)})(1 - D_e^{(1)}) \right]^{1/2}. \quad (5.4.12)$$



(a) Mean family size generated by a new-born woman as a function of time. (b) Distribution function of the time until extinction of the family generated by a first new-born woman.

Figure 5.1



(a) Conditional mean total size of the family generated by a first woman as a function of her age class, given that extinction occurs. (b) Standard deviation of the family generated by a new-born woman as a function of her age class, given that extinction occurs.

Figure 5.2

Capitolo 6

Modellazione dell'estinzione nel processo di diffusione delle informazioni nei Wireless Sensor Networks

6.1 Introduzione

Le reti di sensori wireless WSNs (*“Wireless Sensor Nodes”*) sono state ampiamente considerate come una delle tecnologie più importanti per il ventunesimo secolo. Grazie ai recenti progressi nei sistemi di microelettronica meccanica (MEMS) e tecnologie di comunicazione wireless, i sensori minuscoli, economici e intelligenti impiegati in un'area fisica e collegati in rete tramite collegamenti wireless e Internet offrono opportunità senza precedenti per una varietà di applicazioni civili e militari, per esempio, il monitoraggio ambientale, la sorveglianza del campo di battaglia e controllo dei processi industriali [13]. A differenza dalle reti tradizionali di comunicazione wireless, ad esempio, sistemi cellulari e reti mobili ad hoc (MANET), le WSNs hanno caratteristiche uniche, come: un livello più denso di distribuzione dei nodi, una maggiore affidabilità dei nodi sensori e severe restrizioni in materia di energia, di calcolo e di memorizzazione dei dati [2], che presentano molte nuove sfide nello sviluppo e nell'applicazione dei WSNs.

Negli ultimi dieci anni, i WSNs hanno ricevuto un'attenzione straordinaria sia dal mondo accademico che dall'industria in tutto il mondo. Una grande quantità di attività di ricerca è stata condotta per esplorare e risolvere vari problemi sul design e applicazione e sono stati compiuti notevoli progressi nello sviluppo e nella distribuzione di WSNs. Si prevede che in un futuro prossimo, i WSNs saranno ampiamente utilizzati in vari settori civili e militari e rivoluzioneranno il nostro modo di vivere, lavorare e interagire con il mondo fisico [17].

6.2 Descrizione del sistema

Mentre la ricerca attuale su WSNs si concentra principalmente sul design di protocolli efficienti dal punto di vista energetico che considerano il routing e la copertura, l'obiettivo qui è quello di studiare il processo di diffusione delle informazioni. Nel nostro scenario, i WSNs vengono distribuiti in modo casuale nell'area di monitoraggio. Poiché i nodi del sensore devono operare in modo efficiente in termini di energia o se l'obiettivo di monitoraggio non richiede che essi siano sempre attivi, allora, i nodi adottano uno schema di *“sleep scheduling”* per risparmiare energia. In altre parole, i nodi non

sono attivi per tutto il tempo, ma eseguono cicli di lavoro tra uno stato attivo e uno stato di sonno. Nello stato attivo, l'unità sensoriale del nodo fornisce determinati dati di misura in base al tipo di sensore (temperatura, luce, posizione, accelerometro, ecc.); e l'unità di ricetrasmittenza *RF* invia questi dati ai nodi vicini e sente il radio canale per i messaggi in arrivo dai nodi vicini. D'altra parte, nello "sleep state", il nodo spegne la sua unità sensore e il *RF transceiver*, poi entra in una modalità di risparmio energetico che consuma solo una frazione della potenza necessaria nello stato attivo. Il nostro scenario considera anche i dispositivi eterogenei che operano con le stesse informazioni, ma dove i nodi sono altrimenti indipendenti nella loro pianificazione del ciclo di lavoro, ad esempio, diversi tipi di dispositivi domestici che reagiscono alla presenza dell'utente. Tuttavia, per motivi di trattabilità, supporremo che tutti i nodi avranno la stessa probabilità stazionaria di essere attivi. Poiché la "wireless transmission power" di ciascun nodo è limitata, un nodo può raggiungere solo i nodi che si trovano all'interno del suo campo di trasmissione wireless. Mentre altri documenti spesso presumono che esista un singolo nodo "sink" a cui tutti i dati vengono propagati, qui si considera che tutti gli altri nodi della regione di monitoraggio sono anche interessati a ricevere i dati diffusi da un singolo nodo. Ad esempio, un nodo può registrare un certo evento che deve essere propagato a tanti nodi attivi come sia possibile, vedere la Fig.(6.1). Inoltre, supporremo che queste informazioni sono valide solo per un tempo limitato e diventano obsolete dopo un periodo di tempo aleatorio.

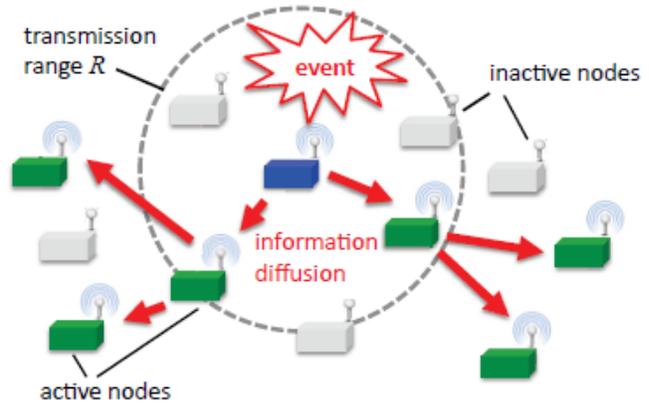


Figure 6.1: Sensor network scenario with information being disseminated from the (blue) node detecting an event. Inactive nodes are shown in grey.

L'obiettivo in questo documento è quello di modellare il processo di disseminazione dell'informazione e quantificare la proporzione dei nodi in una determinata area che hanno avuto le informazioni chiave per qualsiasi tempo, ma anche calcolare il tempo necessario in cui il processo raggiunge un determinato numero di nodi. Siamo anche interessati a calcolare la probabilità di estinzione del processo di disseminazione, cioè, la probabilità che queste informazioni non esistano più nel sistema.

6.3 Modello analitico

Supponiamo che i nodi siano situati in modo casuale in base a "a planar point Poisson process" con tasso λ (vedere [12], cap. 2 per una definizione formale). Ciò implica che, dato il numero di nodi in una zona, questi saranno sparsi in modo uniforme. Di conseguenza, la probabilità $p(k)$, $k \in \{1, 2, \dots\}$ che il k -esimo vicino più prossimo si trovi all'interno di una distanza R da qualsiasi nodo "source" è data da

$$\begin{aligned}
 p(k) &= \mathbb{P}\{N[C(0, R)] \geq k\} \\
 &= 1 - \sum_{i=0}^{k-1} \frac{(\pi R^2 \lambda)^i}{i!} \exp[-\pi R^2 \lambda],
 \end{aligned}
 \tag{6.3.1}$$

dato che R è la “*transmission range*” di ogni nodo del sistema, $N(A)$ è una misura casuale che conta il numero di nodi in un insieme A e $C(0, R)$ indica un cerchio di raggio R centrato in $(0, 0)$.

Un nodo subisce cambiamenti ciclici di stato durante la sua vita. Questi possono essere modellati con l’aiuto di un processo di Markov che possa risultare in un comportamento stazionario. Chiamiamo p_{on} la probabilità stazionaria che un nodo sia “*active*” al tempo t . Quindi, con probabilità complementare, il nodo sarà o “*inactive or sleeping*”. Le informazioni restano nel buffer di messaggi di un nodo solo per un tempo limitato fino a perdere la validità e verranno persi anche a causa del cambiamento di stato, da attivo a inattivo. Supponiamo qui come approssimazione che questa durata del messaggio è distribuita aleatoriamente ed esponenzialmente con parametro μ . A causa dell’accesso casuale al “*radio channel*” e agli errori di trasmissione, il processo di trasmissione di un messaggio è aleatorio e qui si assume che la ricezione dell’informazione di interesse senza errori per un nodo attivo richiede una quantità esponenziale di tempo (con parametro δ). Siamo interessati a studiare il numero di nodi che dispongono delle informazioni specifiche nel loro buffer in qualsiasi momento del tempo. Ciò è espresso dal processo

$$\{(N(t), \phi_0(t), \phi_1(t), \dots, \phi_m(t)) : t \in \mathbb{R}^+\} \quad (6.3.2)$$

dove $N(t)$ è il numero totale di nodi con l’informazione e $\phi_i(t)$, $i \in \{0, 1, \dots, m\}$ descrive il numero di nodi nella fase “ i ” nel sistema. Nel nostro modello, un nodo solo può ricevere le informazioni nella fase “ 0 ” e poi passa automaticamente alla fase successiva, e quando arriva alla fase m non può condividere in più l’informazione. Chiaramente, ne segue

$$\sum_{i=0}^m \phi_i(t) = N(t) \quad (6.3.3)$$

per ogni $t \in \mathbb{R}^+$. Questo processo è un processo di ramificazione, in particolare un *Markovian binary tree*.

Nella nostra particolare configurazione, il *MBT* è completamente descritto da due diverse matrici, vale a dire D_1 e P , e da un vettore colonna d . Ci proponiamo di identificarli in base al comportamento del sistema descritto in precedenza e di spiegare in modo appropriato le relative performance metriche. La matrice D_1 è una matrice quadrata di dimensione $m + 1$, che registra il tasso con cui un nodo che segue una trasmissione cambia il suo stato. Come accennato in precedenza, un nodo nella fase “ i ” può passare solo alla fase “ $i + 1$ ” e un nodo nella fase “ m ” non può già trasmettere l’informazione. Di conseguenza, D_1 è una matrice sparsa dove solo $(D_1)_{i,i+1}$, con $i \in \{0, 1, \dots, m - 1\}$ sono diversi da zero e sono uguali a

$$(D_1)_{i,i+1} = \delta p_{on} p(i + 1), \quad (6.3.4)$$

poiché solo i nodi attivi possono ricevere le informazioni trasmesse e questo avviene con un tasso δ .

Ricordare che, a causa della nostra assunzione di continuità, la probabilità che due nodi ricevano le informazioni esattamente allo stesso tempo è zero. Inoltre, vale la pena notare qui che il modello sovrastima questo tasso di transizione, non tenendo conto che “ i ” nodi avevano già le informazioni. Tuttavia, quanto maggiore sia “ i ”, minore sarà p_i e quindi minore sarà anche il tasso totale di transizione.

La matrice P ha dimensione $(m + 1) \times (m + 1)^2$ e è composta dagli elementi $P_{j|ik}$, con $i, j, k \in \{0, 1, 2, \dots, m\}$ che corrispondono alle probabilità condizionali che il nodo che riceve l’informazione

inizi nella fase “ j ”, sapendo che il nodo che trasmette l’informazione sia nella fase “ i ” e entra nella fase “ k ” subito dopo la trasmissione. Nel nostro scenario, P è una matrice sparsa dove solo il blocco $P_{0|ik}$ è diverso da zero e gli elementi esistono solo per $k = i + 1$ e $i = 0, 1, \dots, m - 1$.

Il vettore colonna d ha dimensione $m + 1$ e registra il tasso per cui un nodo può lasciare il sistema mentre sta nella fase “ i ”, quindi, i suoi elementi sono uguali a μ .

La “*mean progeny matrix*” \mathcal{M} fornisce, per gli elementi $i, j \in \{0, 1, \dots, m\}$, il numero medio di nodi diretti di tipo “ j ” che hanno registrato le informazioni trasmesse da un nodo di tipo “ i ”. Chiaramente, secondo la nostra configurazione, un nodo può trasmettere l’informazione al massimo a un nodo di tipo “1” alla volta. È stato stabilito nel capitolo (4) che la matrice \mathcal{M} soddisfa la seguente identità:

$$\mathcal{M} = N(\mathbf{1} \otimes \mathbf{1}),$$

con $N = (-D_0)^{-1}B$, dove D_0 è una matrice diagonale tale che

$$(D_0)_{ii} = - \left[\sum_{j=0}^m (D_1)_{ij} + \mu \right]. \quad (6.3.5)$$

e B è una matrice di dimensione $(m + 1) \times (m + 1)^2$ tale che $B_{i,jk} = (D_1)_{ik}P_{j|ik}$.

Definiamo il vettore colonna q di dimensione $m + 1$ dove q_i è la probabilità che il processo sperimenti l’estinzione partendo da un nodo nella fase “ i ”. Chiamiamo a q il vettore di probabilità di estinzione.

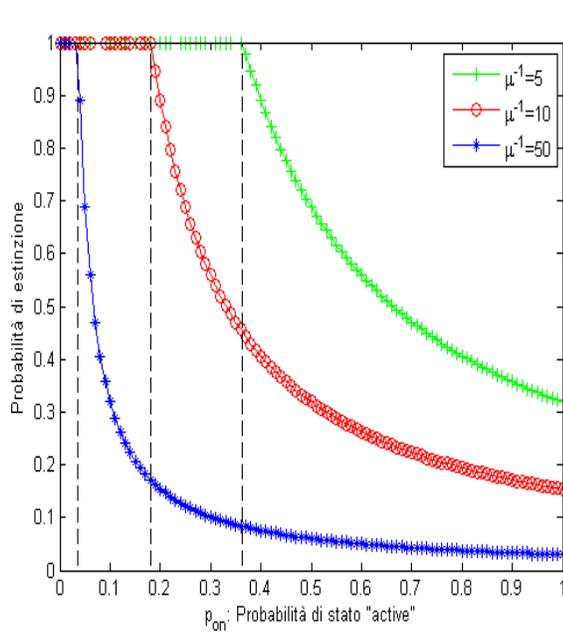
Utilizzando i risultati ottenuti nei capitoli (4) e (5), possiamo ora calcolare le caratteristiche desiderate del processo di diffusione dell’informazione nei “*Wireless Sensor Networks*”.

6.4 Sperimentazione numerica

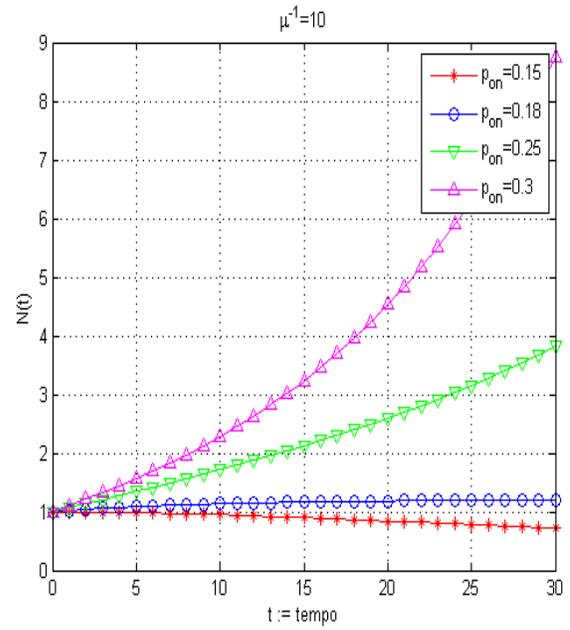
In questa sezione valutiamo numericamente il modello descritto nella sezione precedente, vedremo come tutti gli strumenti imparati finora ci aiuteranno a modellare questo fenomeno. Supporremo che “*the average transmission rate*” δ sia data per una specifica applicazione e “*packet size*”, pertanto, tutte le unità di tempo saranno espresse in relazione a δ^{-1} . Per la nostra sperimentazione, i seguenti parametri rimarranno sempre fissi:

- La densità dei nodi $\lambda = 0.01$.
- La gamma di trasmissione $R = 10$.
- Il tasso di durata del messaggio nel buffer $\mu^{-1} = 5, 10, 50$ (3 valori).

Ovviamente, la durata di un messaggio deve essere tale che $\mu^{-1} > \delta^{-1}$, altrimenti non sarebbe diffuso l’informazione al di là del primo nodo adiacente, quindi, per garantire questo useremo il valore $\delta^{-1} = 10/7 = 1.4286$ e inoltre $m = 6$, cioè 7 - *fasi*.



(a) Extinction probability over p_{on} .



(b) Mean number of nodes with information $N(t)$.

Figure 6.2

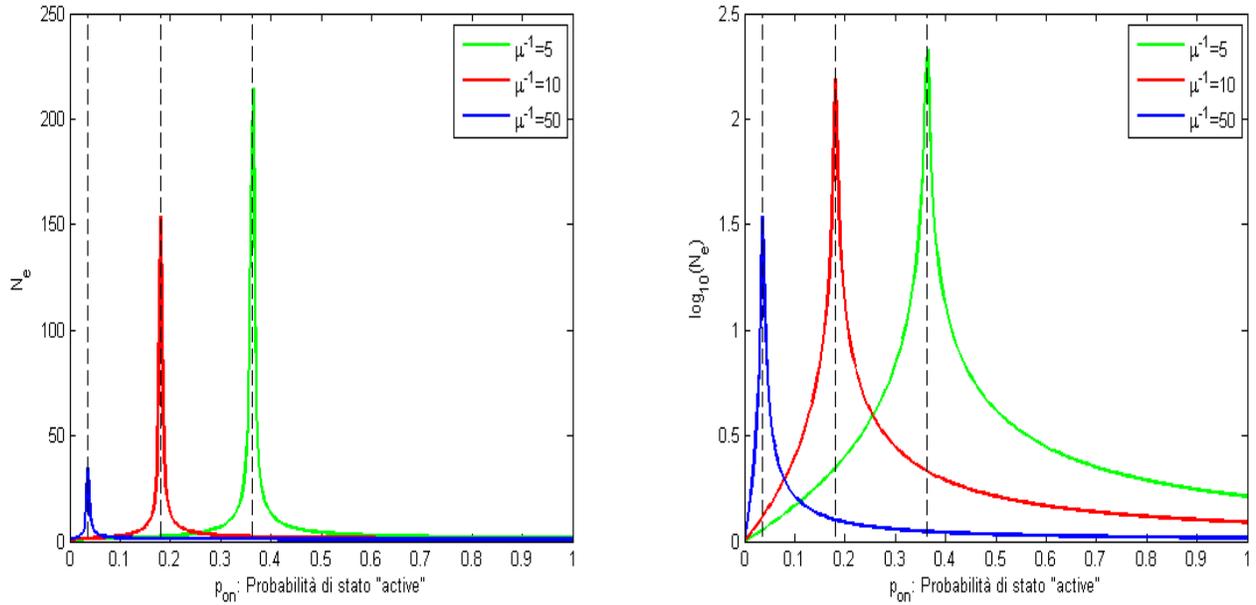
6.4.1 Probabilità di estinzione delle informazioni

Per calcolare la probabilità di estinzione possiamo usare qualsiasi degli otto metodi studiati nel capitolo (4), in questo caso, useremo l'algoritmo *Newton-Depth*. Dobbiamo inoltre notare che come la trasmissione inizia sempre con un nodo nella fase "0", quindi, i nostri risultati si concentreranno sempre sui primi componenti dei vettori che ne derivano, in questo caso q_1 . La Figura (6.2a) mostra la probabilità di estinzione rispetto alla probabilità che i nodi siano attivi p_{on} . Possiamo vedere che per $\mu^{-1} = 5, 10, 50$, quanto più lunga è la validità dei dati, minore è la probabilità di estinzione, anche per i piccoli valori di p_{on} . A causa della durata finita di un messaggio, il processo di diffusione è sempre propenso all'estinzione a un certo punto. Tuttavia, il processo di diffusione può fermarsi anche prima, seconda delle posizioni dei nodi. Vediamo che una certa estinzione potrebbe essere possibile anche se il messaggio è ancora valido.

Inoltre, questa figura ci mostra che se la validità del messaggio è in media 5, 10 o 50 volte superiore alla durata media del trasferimento dei messaggi, i nodi devono essere attivi per almeno p_{on} uguali a 0.364, 0.18 o 0.036, rispettivamente, per evitare una certa estinzione.

6.4.2 Numero medio di Nodi con le informazioni

Il numero medio di nodi che hanno ricevuto il messaggio di informazione: $N(t)$ può essere calcolato usando l'equazione (5.2.10) del corollario (5.1). La Fig. (6.2b) mostra i valori di $N(t)$ rispetto del tempo per il valore di $\mu^{-1} = 10$. Si noti che per $p_{on} = 0.18$ circa, il numero dei nodi $N(t)$ diminuisce nel tempo, mentre per $p_{on} > 0.18$ si dispone di una diffusione con successo dell'informazione e inoltre, $N(t)$ cresce rapidamente.



(a) Conditional number of nodes N_e under extinction. (b) Conditional number of nodes $\log_{10}(N_e)$ under extinction.

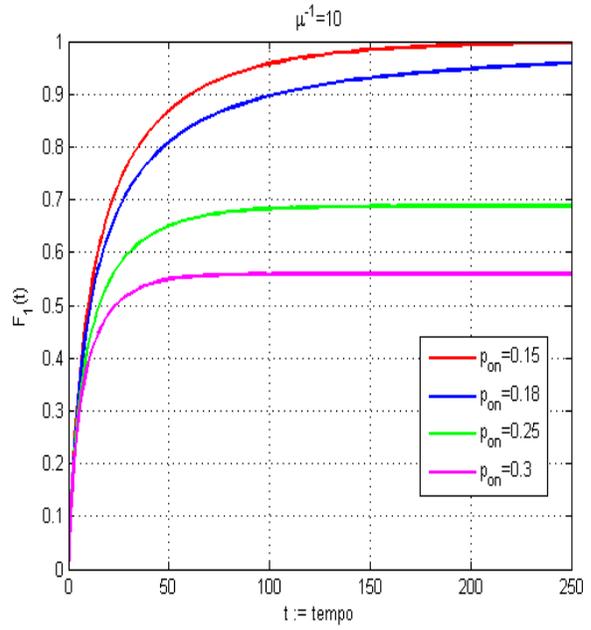
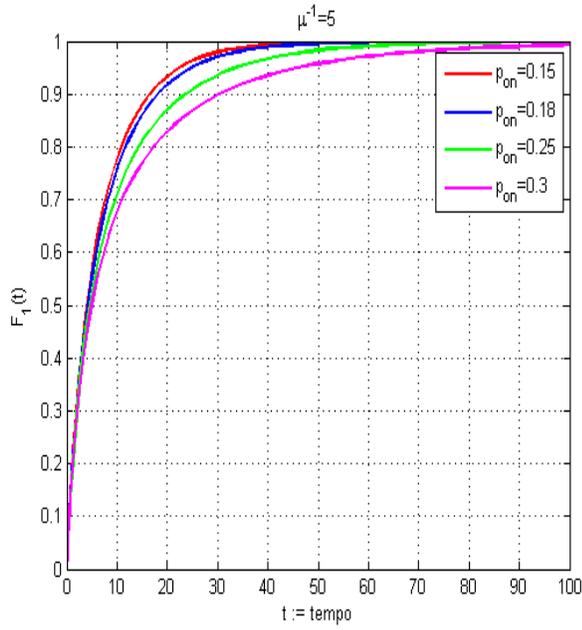
Figure 6.3

6.4.3 Numero medio totale di Nodi che avevano le informazioni

Se consideriamo il numero totale di nodi N_e che sono coinvolti nel processo di diffusione condizionato al fatto che si verifichi l'estinzione, otteniamo il suo comportamento nell'Fig. (6.3a). Questi valori sono stati calcolati con l'equazione (5.4.11). Le linee tratteggiate verticali indicano i punti dove la commutazione tra l'estinzione e l'eventuale diffusione con successo avviene per $\mu^{-1} = 5, 10, 50$. Quando la probabilità di attività del nodo diventa grande, N_e si avvicina a 1, per vedere questo, abbiamo tracciato la curva $\log_{10}(N_e)$ vs p_{on} , che è mostrato nella figura (6.3b); in questo caso, le curve si avvicinano allo zero. Ciò significa che l'unico modo per osservare l'estinzione del processo di diffusione corrisponde a quei casi in cui il nodo che trasmette l'informazione ("source node") non era in grado di raggiungere ai suoi vicini. Quindi, in totale, solo un nodo aveva l'informazione durante il processo di diffusione. Poiché questo accade sempre meno quando p_{on} cresce, si vede una diminuzione della probabilità di estinzione come era già notato nella Fig. (6.2a). Di conseguenza, se un "source node" riesce a trasmettere l'informazione almeno a uno dei suoi vicini, esiste una buona probabilità che l'estinzione del processo di diffusione non si verifichi.

6.4.4 Tempo fino all'estinzione delle informazioni

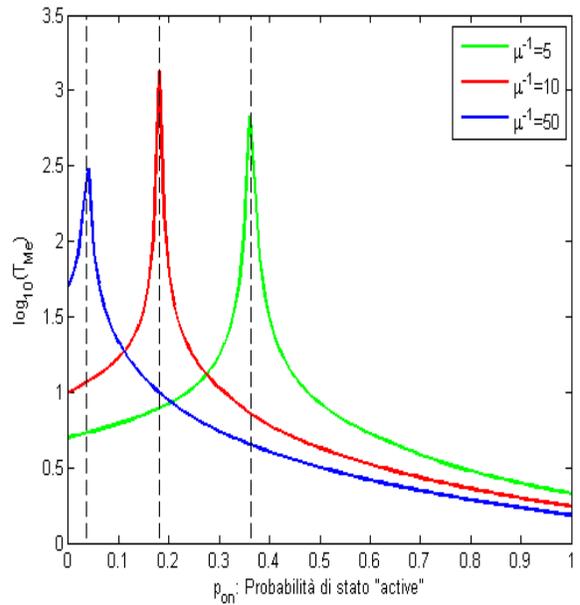
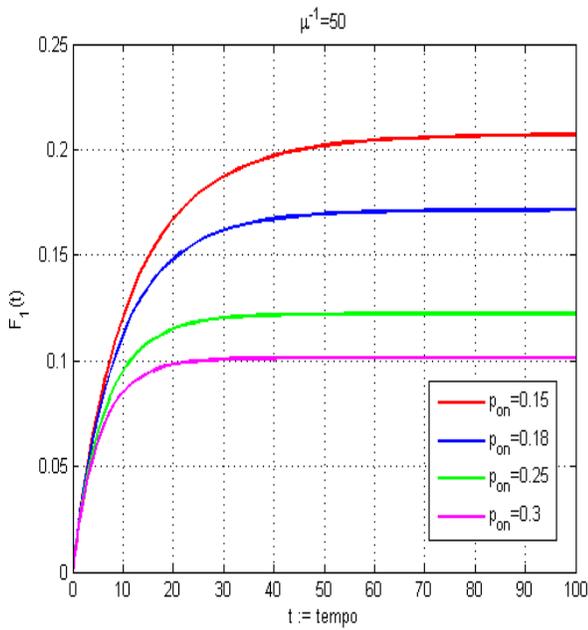
La distribuzione di "the time until extinction": T_e , viene calcolato con l'equazione (5.3.1). Nelle figure (6.4a), (6.4b) e (6.5a) abbiamo tracciato queste distribuzioni $F_1(t)$ di T_e quando la trasmissione inizia con un nodo nella fase "0" per i valori di $\mu^{-1} = 5, 10$ e 50 rispettivamente (che è la prima componente di $F(t)$); e per quattro valori diversi di p_{on} che sono $0.15, 0.18, 0.25$ e 0.3 . Osserviamo che quando il tempo va all'infinito, le curve tendono alla probabilità di estinzione in ogni caso.



(a) Distribution function of the time until extinction generated of a node in phase 0 ($\mu^{-1} = 5$).

(b) Distribution function of the time until extinction generated of a node in phase 0 ($\mu^{-1} = 10$).

Figure 6.4



(a) Distribution function of the time until extinction generated of a node in phase 0 ($\mu^{-1} = 50$).

(b) Logarithm of mean time until extinction: $\log_{10}(T_{Me})$ vs p_{on} .

Figure 6.5

Ricordare che il “*the mean time until extinction*”: T_{Me} , dato che si verifichi l'estinzione, viene approssimato dall'equazione (5.3.10). Quindi, in questo caso la Fig. (6.5b) mostra la grafica del $\log_{10}(T_{Me})$ rispetto a p_{on} per i valori di $h = 10^{-2}$ e $\epsilon = 5 \times 10^{-2}$. Notare che questo grafico è molto simile a (6.3b), con la differenza che (6.5b) non inizia dal punto $(0, 0)$, il motivo è molto semplice; quando $p_{on} = 0$, la probabilità di estinzione è uguale a “1” ma il tempo medio per questa estinzione è almeno il tempo medio per il quale l'informazione rimarrà ancora nel buffer di messaggi, in questo caso $\mu^{-1} = 5, 10, 50$ rispettivamente, quindi come il grafico rappresenta a $\log_{10}(T_{Me})$ avremo che le curve in (6.5b) inizieranno in $\log_{10}(5) = 0.699$, $\log_{10}(10) = 1$ e $\log_{10}(50) = 1.699$ rispettivamente, che è evidente quando si guarda la figura.

Osservazione:

Esiste un'altra caratteristica transitoria importante da studiare: “*The mean time to reach K nodes*” prima che l'estinzione accada, ma non c'è molta letteratura matematica su questo argomento. Tuttavia, in [20] sono stati sviluppati vari algoritmi per calcolare le stesse caratteristiche transitorie che abbiamo appena studiato però con un approccio un po' diverso, chiamato “*Structured Markov Chain to Branching Processes*”. Il vantaggio di questo approccio è che ci permette modellare fenomeni in un ambiente casuale, ma il suo principale svantaggio è che i metodi algoritmici proposti sono numericamente efficaci solo quando viene applicato a MBTs con un numero relativamente piccolo di fasi. Di seguito diamo la definizione di “*Structured Markov Chain to Branching Processes*” e per ulteriori dettagli rimandiamo a vedere l'articolo [20].

Structured Markov Chain Approach to Branching Processes

Un “*Markovian Binary Tree*” è un particolare processo di Markov con generatore infinitesimale che ha una struttura particolare. Vale a dire, in questo approccio, il processo

$$\{(N(t), \phi_0(t), \phi_1(t), \dots, \phi_m(t)) : t \in \mathbb{R}^+\}$$

è completamente caratterizzato nello spazio di stati \mathcal{N} dei possibili valori per $(N(t), \phi_0(t), \phi_1(t), \dots, \phi_m(t))$ tale che $N(t) \geq 0$, $\sum_{0 \leq i \leq m} \phi_i(t) = N(t)$ con $\phi_i(t) \geq 0$. Questo spazio di stati può essere partizionato come

$$\mathcal{N} = \bigcup_{k \geq 0} \mathcal{L}(k),$$

dove $\mathcal{L}(k)$ è chiamato *il livello* e consiste dagli stati

$$\mathcal{L}(k) = \{(\phi_0(t), \phi_1(t), \dots, \phi_m(t)) : \sum_{0 \leq i \leq m} \phi_i = k, \phi_i \geq 0\}.$$

Secondo questa partizione, il generatore Q ha una struttura a blocchi come segue

$$Q = \begin{pmatrix} A_1^{(0)} & A_0^{(0)} & 0 & 0 & \dots \\ A_2^{(1)} & A_1^{(1)} & A_0^{(1)} & 0 & \dots \\ 0 & A_2^{(2)} & A_1^{(2)} & A_0^{(2)} & \dots \\ 0 & 0 & A_2^{(3)} & A_1^{(3)} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

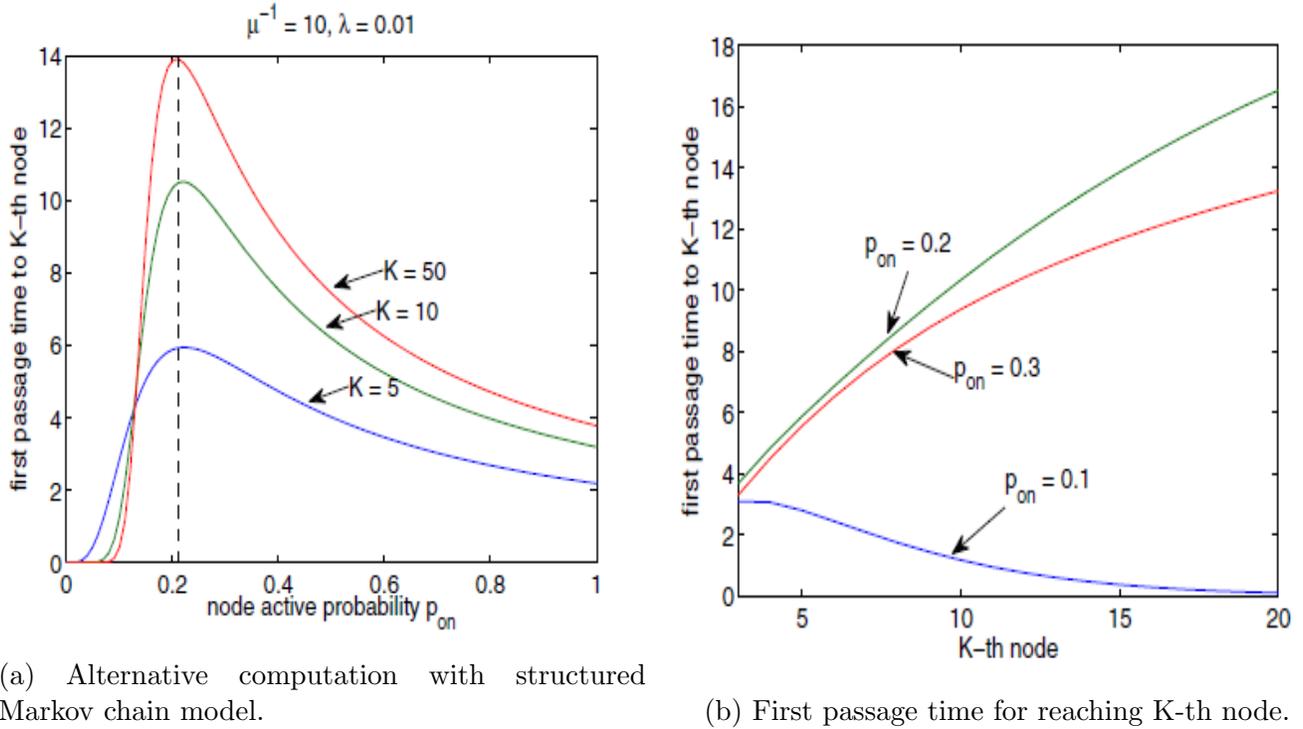


Figure 6.6

Con questa rappresentazione abbiamo che $A_0^{(0)}, A_1^{(0)} = 0$, ma non esiste un modo semplice per definire le altre matrici a blocchi $A_2^{(k)}, A_1^{(k)}$ e $A_0^{(k)}$ per $k \geq 1$ direttamente in termini di d, D_0 e B , la descrizione delle matrici a blocchi deve essere fatto in termini dei componenti. La struttura di Q semplifica l'elaborazione del "mean first passage time" ad un particolare numero K di nodi nel sistema prima dell'estinzione, l'algoritmo (6.1) in [20] ci permette calcolare K .

6.4.5 Tempo medio per raggiungere "K" nodi prima dell'estinzione

Le figure (6.6a) e (6.6b) sono stati presi di [41] su cui si basa questo capitolo della tesi. La Figura (6.6) rappresenta i risultati ottenuti utilizzando il "Structured Markov Chain Approach" riportato nella sezione precedente. Ricordare che per $\mu^{-1} = 10$ e $\lambda = 0.01$, la probabilità p_{on} deve essere maggiore di 0,18 per avere una probabilità di estinzione inferiore a 1, possiamo vedere in (6.6a) che il primo tempo di passaggio per raggiungere il K -esimo nodo vicino aumenta strettamente fino a raggiungere lo stesso picco, circa $p_{on} = 0.21$ indipendente da K . Come abbiamo visto prima, per $p_{on} = 0.1$ l'estinzione si verifica, in modo che il primo tempo di passaggio si avvicina a "0" per i grandi valori di K poiché i nodi non sono attivi lo sufficiente come per sostenere il processo di diffusione, questo comportamento è mostrato nella Fig. (6.6b). Quando $p_{on} = 0.2$ si richiede un tempo più lungo per raggiungere i nodi, ma valori più grandi aiutano alla diffusione per raggiungere i nodi più velocemente. Possiamo anche vedere che per i grandi valori di K , le curve si comprimono e a causa della diffusione dell'informazione, vengono raggiunti più nodi in quasi lo stesso tempo. Dobbiamo sottolineare che nella nostra sperimentazione noi abbiamo usato i valori $\delta^{-1} = 10/7$ e $m = 6$ ma in [41] non specifica i valori usati per m e δ , quindi, è probabile che questi due ultimi grafici potrebbero variare leggermente rispetto ai valori specifici dai nostri risultati, ma in quanto

riguarda al comportamento generale del processo, non c'è alcuna variazione significativa.

6.5 Conclusioni

In questa sezione abbiamo fornito un modello analitico per il processo di diffusione dell'informazione in una rete di sensori wireless usando un MBT. Dopo il rilevamento di un evento, abbiamo determinato la probabilità che tali informazioni verranno persi nella rete a causa di nodi di sonno o scadenza della validità. Nella definizione del processo di ramificazione, questa viene chiamata probabilità di estinzione. Supponiamo che i nodi siano distribuiti secondo un "*planar Poisson process*" abbiamo visto che sono possibili due tecniche complementari per l'analisi del processo di ramificazione. I nostri risultati mostrano che è possibile derivare il valore limite tra l'estinzione e la diffusione con successo dell'informazione, che può essere utile nel design di modelli ottimali del ciclo di lavoro e nell'aggiornamento di intervalli di ritrasmissione dell'informazione. Questi risultati ci aiutano a chiarire le condizioni in cui una simulazione potrebbe ora essere eseguita per evitare una certa estinzione e concentrarsi maggiormente sulla diffusione geografica.

Appendice

Capitolo A

Funzione generatrice di probabilità e Convoluzione

Funzione generatrice di probabilità

Premettiamo alcuni richiami sulle *serie di potenze*: data una successione di numeri $(a_n)_{n \geq 0}$, si chiama serie di potenze ad essa associata la serie $\sum_{n=0}^{\infty} a_n t^n$.

Il raggio di convergenza R verifica l'equazione:

$$R = \frac{1}{\limsup_{n \rightarrow \infty} \sqrt[n]{|a_n|}}.$$

(con la convenzione $\frac{1}{0} = +\infty$ e $\frac{1}{+\infty} = 0$). La serie di potenze converge per $|t| < R$ e non converge per $|t| > R$; inoltre se $R > 0$, posto $\varphi(t) = \sum_{n=0}^{\infty} a_n t^n$, si ha $a_n = \frac{1}{n!} \varphi^{(n)}(0)$ e di conseguenza due serie di potenze coincidono se e solo se tutti i coefficienti $(a_n)_{n \geq 0}$ sono uguali.

In questo paragrafo consideriamo solamente variabili aleatorie X, Y, \dots a *valori non negativi*.

Definizione A.1. Sia X una v.a. discreta a valori non negativi con distribuzione di probabilità

$$p_k = \mathbb{P}\{X = k\}, \quad k = 0, 1, 2, \dots$$

Si chiama *funzione generatrice di probabilità (-pgf)* (Probability-generating function) alla funzione $f_X(\cdot)$ definita da

$$f_X(t) = \mathbb{E}\{t^X\} = \sum_{n=0}^{\infty} p_n t^n.$$

□

Si noti che la funzione generatrice è sicuramente definita per $|t| \leq 1$ (infatti il raggio di convergenza è sicuramente maggiore o eguale a 1, e si verifica direttamente che la serie converge per $t = 1$).

Le proprietà più importanti delle funzioni generatrici di probabilità sono le seguenti (ved. [31]):

(1) $|f_X(t)| \leq 1$, per ogni $|t| \leq 1$.

(2) Il pgf è definito dalle probabilità; l'unicità della espansione di una serie di potenza implica che il pgf a sua volta definisce le probabilità, quindi

$$p_k = \left[\frac{1}{k!} \cdot \frac{d^k f_X(t)}{dt^k} \right]_{t=0}, \quad k = 0, 1, 2, \dots$$

(3) Se il r -esimo momento m_r esiste, allora

$$m_r = \sum_{j=0}^{\infty} j^r p_j = \lim_{t \nearrow 1} \left[\frac{d^r f_X(t)}{dt^r} \right], \quad r = 1, 2, \dots$$

In particolare abbiamo

$$\triangleright \mathbb{E}\{X\} = \lim_{t \nearrow 1} f'_X(t)$$

$$\triangleright \mathbb{E}\{X(X-1)\} = \lim_{t \nearrow 1} f''_X(t)$$

(4) $f_X(t) = f_Y(t) \Leftrightarrow X$ e Y sono equidistribuite.

(5) Se X e Y sono indipendenti $\Rightarrow f_{X+Y}(t) = f_X(t)f_Y(t)$.

Riportiamo qua sotto una tabella delle funzioni generatrici delle più usuali variabili aleatorie a valori interi, che si può facilmente verificare.

- $X \sim B(n, p) \Rightarrow f_X(t) = [1 + p(t-1)]^n$;
- X Geometrica di parametro $p \Rightarrow f_X(t) = \frac{tp}{1-t(1-p)}$;
- X di Poisson di parametro $\lambda \Rightarrow f_X(t) = e^{\lambda(t-1)}$;

Convoluzione

La convoluzione (*convolution*) delle distribuzioni di probabilità sorge nella teoria della probabilità e nella statistica come l'operazione che corrispondono alla somma di variabili casuali indipendenti e, per estensione, alla formazione di combinazioni lineari di variabili casuali. L'operazione qui è un caso speciale di convoluzione nel contesto delle distribuzioni di probabilità.

Formula della Convoluzione Discreta

Sarà conveniente supporre qui che queste funzioni di distribuzione siano definite per tutti gli interi, definendoli come 0 altrimenti.

Siano X e Y due v.a. indipendenti a valori interi con distribuzioni di probabilità p_X e p_Y . Sia $Z = X + Y$, vorremmo determinare la funzione di distribuzione p_Z di Z .

Suponiamo che $X = k$, dove $k \in \mathbb{Z}$, allora $Z = z$ se e solo se $Y = z - k$, quindi l'evento $Z = z$ è l'unione degli eventi pairwise disgiunti

$$\{X = k\} \text{ e } \{Y = z - k\},$$

Formula della Convoluzione Continua

In questa sezione consideriamo la versione continua del problema poste nella sezione precedente: Come vengono distribuite le somme di variabili casuali indipendenti?

Definizione A.3. *Siano X e Y due v.a continue con funzioni di densità f_X e f_Y , rispettivamente. Supponiamo che f_X e f_Y sono definite per tutti i numeri reali. Allora, la convoluzione $f_X * f_Y$ di f_X e f_Y è la funzione data da*

$$\begin{aligned}(f_X * f_Y)(z) &= \int_{-\infty}^{+\infty} f_X(z-y)f_Y(y)dy \\ &= \int_{-\infty}^{+\infty} f_Y(z-x)f_X(x)dx\end{aligned}$$

□

Questa definizione è analoga alla definizione (A.2). Quindi non dovrebbe sorprendere che se X e Y sono indipendenti, allora la densità della loro somma è la convoluzione delle loro densità. Questo fatto è indicato come un teorema sotto,

Teorema A.1. *Siano X e Y due v.a indipendenti con funzioni di densità $f_X(x)$ e $f_Y(y)$, rispettivamente. Allora, la somma $Z = X + Y$ è una variabile aleatoria con funzione di densità $f_Z(z)$, dove f_Z è la convoluzione di f_X e f_Y .*

□

Capitolo B

Tree Topology

La Topologia di un Albero (*Tre Topology*) nel contesto matematico è definito come il standard di ramificazione di quell'albero, quando le lunghezze dei rami e le etichette delle specie nelle foglie vengono ignorati. Pertanto, qualunque albero può essere disegnato in modo topologico se tutte le lunghezze dei rami sono rese uguali; questo è rappresentato nella figura (B.1). Gli alberi A e B sono topologicamente identiche, l'unica differenza è che i rami dell'albero A hanno lunghezze variabili, mentre i rami dell'albero B presentano la stessa lunghezza. Le lunghezze dei rami di un albero rappresentato topologicamente non riflettono le età dei rami. La topologia o la forma di un albero trasmette informazioni relative alle relazioni di posizione tra le specie.

A Labelling System for Binary Trees

Definiamo tre tipi di nodi per un albero binario (o per un albero generico):

- Nodi interni, chiamati anche punti di ramificazione,
- Nodi fogliari estinti, e
- Nodi fogliari instabili.

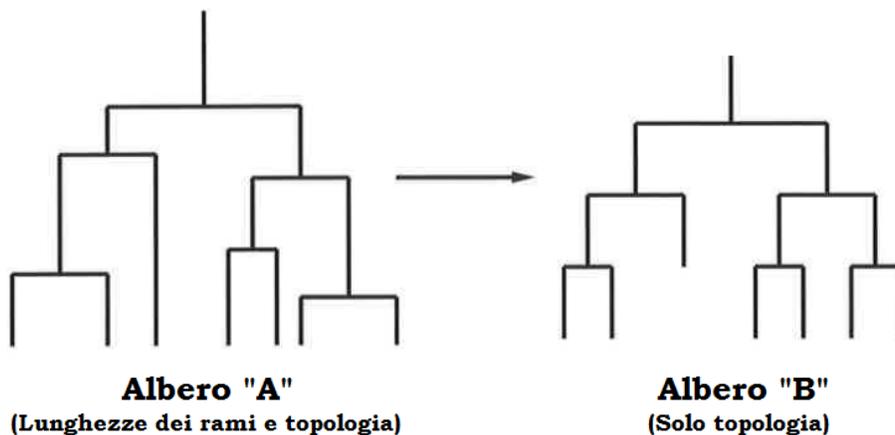


Figure B.1: Due rappresentazioni dello stesso albero.

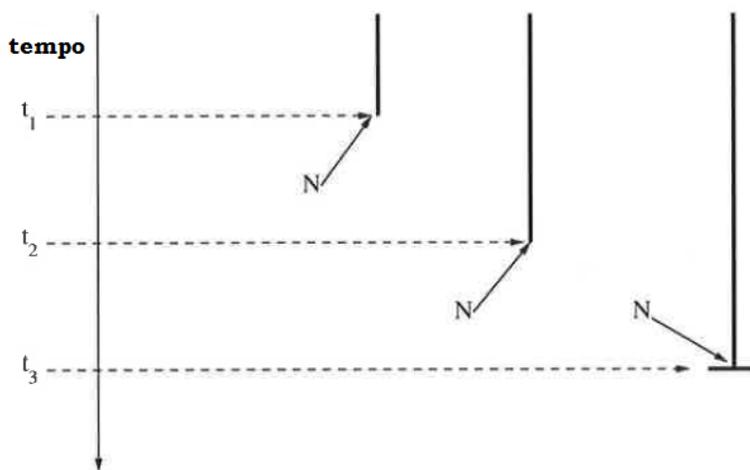


Figure B.2: Esempio dell'evoluzione di un nodo fogliare instabile.

Un ramo interno è definito come un ramo che ha completato la sua evoluzione e non è un ramo di foglie. Un ramo estinto è un ramo fogliare che ha completato la sua evoluzione. Un ramo instabile è definito come un ramo che non ha completato la sua evoluzione, quindi, o genererà un nuovo figlio e diventerà un ramo interno o semplicemente si estinguerà.

I nodi interni, conosciuti anche come punti di ramificazione, hanno una posizione fissa nell'albero e non cambiano quando l'albero si evolve e il loro nome suggerisce che non sono né nodi fogliari né nodi di origine (chiamati anche nodi di radice). Un nodo fogliare estinto è il nodo che si trova alla fine di un ramo estinto. La posizione di un nodo fogliare estinto è fissa e non cambia quando l'albero si evolve. Il modo migliore per spiegare un nodo fogliare instabile è quello di considerare la Figura (B.2).

In questa figura descriviamo un unico ramo, questo ramo è rappresentato in tre tempi diversi, t_1 , t_2 e t_3 . Poiché il ramo sta in evoluzione, la posizione del nodo fogliare N si altera da t_1 a t_2 e infine a t_3 . Al tempo t_3 il ramo diventa estinto e la posizione del nodo fogliare è finalmente fissato e quindi diventa un nodo fogliare estinto. Così, un nodo fogliare instabile non è fisso mentre un ramo continua ad evolversi ma diventa fisso se il ramo subisce un punto di ramificazione e poi diventa un nodo interno o se il ramo diventa estinto e quindi il nodo diventa un nodo fogliare estinto.

Da un punto di vista topologico, tuttavia, i singoli rami della Figura (B.2) sono tutti identici, non fa differenza che il nodo instabile N non fosse stato fissato fino al tempo t_3 .

Da qui in poi, rappresentiamo alle etichette dei nodi tra parentesi quadre per garantire che ciascun nodo possa essere riconosciuto in modo inequivocabile. Iniziamo ad etichettare dal nodo $[0]$. Questo nodo è, o il nodo fogliare instabile del ramo radice, il nodo fogliare estinto di un albero con un singolo ramo, o il primo nodo interno di un albero con almeno due rami. Successivamente forniamo un'etichetta al nodo radice, che è il nodo padre di $[0]$. Supponendo che $[\psi] = [0, i_1, \dots, i_m]$, dove $i_1, \dots, i_m \in \{0, 1\}$, è un nodo di un albero binario, dove $|\psi| = m + 1$ è definito come la profondità del nodo $[\psi]$ ("depth"). Il nodo che è collegato a sinistra di $[\psi]$, chiamato nodo figlio, è etichettato come

$$[\psi, 0] = [0, i_1, \dots, i_m, 0],$$

mentre il nodo che è collegato alla destra di $[\psi]$, chiamato il sub-nodo parentale di $[\psi]$, è etichettato da

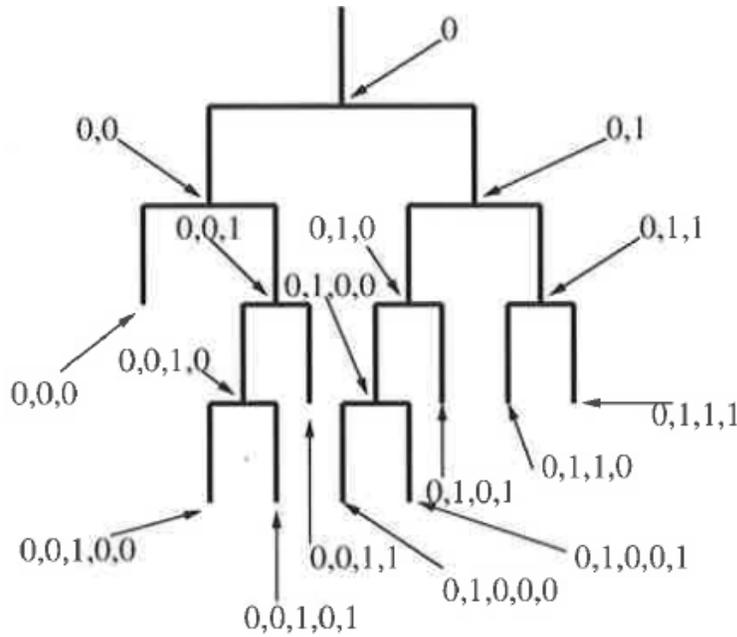


Figure B.3: Un esempio di etichettatura di un albero binario.

$$[\psi, 1] = [0, i_1, \dots, i_m, 1].$$

Come una questione di terminologia corretta, il nodo padre di $[\psi] = [0, i_1, \dots, i_m]$ è sempre quel nodo che si trova ad una profondità di “ m ” con l’etichetta $[0, i_1, \dots, i_{m-1}]$ e tutti i nodi, $[\psi, e_k]$, sono i sub-nodi parentali di $[\psi]$ a condizione che questi nodi esistano. La figura (B.3) rappresenta una topologia di un albero binario con tutti i suoi nodi etichettati.

Un’altra volta, sia $[\psi] = [0, i_1, \dots, i_m]$ qualsiasi nodo ad eccezione del nodo radice di un albero binario. La funzione “ α ” ha la seguente azione su $[\psi]$,

$$\alpha(\psi) = [0, i_1, \dots, i_{m-1}],$$

cioè, $\alpha(\psi)$ è il nodo padre di $[\psi]$. Di conseguenza possiamo etichettare il nodo radice di un albero binario con un $\alpha(0)$. La funzione “ θ ”, d’altra parte, agisce su qualsiasi nodo interno $[\psi]$, o, il nodo radice $\alpha(0)$ in modo che,

$$\theta(\psi) = [\psi, 1],$$

e

$$\theta(\alpha(0)) = [0].$$

Supponiamo che $[\psi] = [0, i_1, \dots, i_m]$ sia un nodo di un albero binario. Il segmento di ramo tra i nodi $[\psi]$ e $[\psi, i_{m+1}]$ è rappresentato da,

$$([\psi], [\psi, i_{m+1}]),$$

dove $i_{m+1} \in \{0, 1\}$. Sia il ramo $([\alpha(\psi)], [\psi])$, allora

- Per indicare che il ramo è estinto, scriviamo $([\alpha(\psi)], [\psi])^{(e)}$, e per indicare il nodo fogliare estinto di questo ramo scriviamo $[\psi]^{(e)}$.
- Per indicare il ramo interno, scriviamo $([\alpha(\psi)], [\psi])^{(i)}$, e per indicare il nodo interno di questo ramo scriviamo $[\psi]^{(i)}$.
- Per indicare il ramo instabile, scriviamo $([\alpha(\psi)], [\psi])^{(u)}$, e per indicare il nodo fogliare instabile di questo ramo scriviamo $[\psi]^{(u)}$.

Se un tipo di ramo non è importante, non specifichiamo un superscript.

Sia \mathcal{T} la topologia di un albero binario. Se $[\psi]$ è un nodo interno di un albero di questa topologia, allora, l'albero di topologia $\mathcal{T}_{[\psi]}$, basato sul nodo $[\psi]$ può essere scritto come un insieme ordinato,

$$\mathcal{T}_{[\psi]} = \{([\alpha(\psi)], [\psi])^{(i)}, \mathcal{T}_{[\psi,0]}, \mathcal{T}_{[\psi,1]}\},$$

dove $\mathcal{T}_{[\psi,0]}$ e $\mathcal{T}_{[\psi,1]}$ sono le topologie dei sub-alberi figlio e parentale dove i primi punti di ramificazione interno si verificano rispettivamente nei nodi $[\psi, 0]$ e $[\psi, 1]$, e per una topologia di ramo unico che è estinta abbiamo,

$$\mathcal{T}_{[\psi]} = \{([\alpha(\psi)], [\psi])^{(e)}\},$$

o per una topologia di ramo unico che è instabile abbiamo

$$\mathcal{T}_{[\psi]} = \{([\alpha(\psi)], [\psi])^{(u)}\}.$$

Si dice che $\mathcal{T}_{[\psi]}$ è l'albero genitore dei sub-alberi figlio $\mathcal{T}_{[\psi,0]}$, e parentale $\mathcal{T}_{[\psi,1]}$. Di conseguenza, a un punto di ramificazione, lo chiamamo il nodo $[\psi]$, al ramo $([\psi], [\psi, 0])$ chiamamo ramo figlio e al ramo $([\psi], [\psi, 1])$ chiamamo ramo parentale.

Per finire, vogliamo introdurre un'altro tipo di nodo e ramo, chiamati nodo quasi-stabile e ramo quasi-stabile. Un nodo quasi-stabile può essere considerato simile a un nodo instabile, eccetto per una importante differenza, un nodo quasi-stabile non può mai diventare un nodo interno o un nodo fogliare estinto, quindi una volta che si forma un nodo quasi-stabile, quella porzione del ramo diventa un ramo fogliare non estinto, in modo che il nodo non viene mai fissato. Si indica con $[\psi]^q$ un nodo quasi-stabile e con $([\alpha(\psi)], [\psi])^q$ un ramo fogliare quasi-stabile.

Bibliografia

- [1] J. Abate and W. Whitt. Numerical inversion of probability generating functions. *Operations Research Letters*, 12(4):245–251, 1992.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications magazine*, 40(8):102–114, 2002.
- [3] W. J. Anderson. *Continuous-time Markov chains: An applications-oriented approach*. Springer Science & Business Media, 2012.
- [4] K. B. Athreya and P. E. Ney. *Branching processes*, volume 196. Springer Science & Business Media, 2012.
- [5] P. Baldi. *Equazioni differenziali stocastiche e applicazioni*, volume 28. Pitagora, 1984.
- [6] N. G. Bean, N. Kontoleon, and P. G. Taylor. Markovian trees: properties and algorithms. *Annals of Operations Research*, 160(1):31–50, 2008.
- [7] A. Berman and R. J. Plemmons. *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.
- [8] L. Breuer and D. Baum. *An introduction to queueing theory: and matrix-analytic methods*. Springer Science & Business Media, 2005.
- [9] L. Bright and P. G. Taylor. Calculating the equilibrium distribution in level dependent Quasi-Birth-and-Death processes. *Stochastic Models*, 11(3):497–525, 1995.
- [10] Z. Brzezniak and T. Zastawniak. *Basic stochastic processes: a course through exercises*. Springer Science & Business Media, 2000.
- [11] P. Buchholz, J. Kriege, and I. Felko. *Input modeling with phase-type distributions and Markov models: theory and applications*. Springer, 2014.
- [12] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic geometry and its applications*. John Wiley & Sons, 2013.
- [13] C.-Y. Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.
- [14] D. de la Fuente García and R. P. Díez. *Teoría de líneas de espera: modelos de colas*. Universidad de Oviedo, 2001.

- [15] K. S. Dorman, J. S. Sinsheimer, and K. Lange. In the garden of branching processes. *SIAM review*, 46(2):202–229, 2004.
- [16] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.
- [17] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. *IEEE pervasive computing*, 1(1):59–69, 2002.
- [18] D. Gaver, P. Jacobs, and G. Latouche. Finite Birth-and-Death models in randomly changing environments. *Advances in applied probability*, 16(04):715–731, 1984.
- [19] T. E. Harris. *The theory of branching processes*. Courier Corporation, 2002.
- [20] S. Hautphenne. A structured Markov chain approach to branching processes. *Stochastic Models*, 31(3):403–432, 2015.
- [21] S. Hautphenne and B. Van Houdt. On the link between Markovian trees and tree-structured Markov chains. *European journal of operational research*, 201(3):791–798, 2010.
- [22] S. Hautphenne, K. Leibnitz, and M.-A. Remiche. Extinction probability in peer-to-peer file diffusion. *ACM SIGMETRICS Performance Evaluation Review*, 34(2):3–4, 2006.
- [23] S. Hautphenne, G. Latouche, and M.-A. Remiche. Newton’s iteration for the extinction probability of a Markovian binary tree. *Linear algebra and its applications*, 428(11-12):2791–2804, 2008.
- [24] S. Hautphenne, G. Latouche, and M.-A. Remiche. An algorithmic look at phase-controlled Branching processes. 2009.
- [25] S. Hautphenne, G. Latouche, and M.-A. Remiche. Transient features for Markovian binary trees. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, page 18. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.
- [26] S. Hautphenne, G. Latouche, and M.-A. Remiche. Algorithmic approach to the extinction probability of Branching processes. *Methodology and Computing in Applied Probability*, 13(1):171–192, 2011.
- [27] Q.-M. He. *Fundamentals of matrix-analytic methods*, volume 365. Springer, 2014.
- [28] P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to stochastic processes*. Waveland Press, 1986.
- [29] V. L. in Probability and Statistics. Probability, mathematical statistics, stochastic processes. URL <http://www.randomservices.org/stat/>.
- [30] J. Jacod and P. Protter. *Probability essentials*. Springer Science & Business Media, 2004.
- [31] N. L. Johnson, A. W. Kemp, and S. Kotz. *Univariate Discrete Distributions, Set*, volume 444. John Wiley & Sons, 2005.

- [32] J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov chains: with a chapter of Markov random fields by David Griffeth*, volume 40. Springer Science & Business Media, 2012.
- [33] J. P. Kharoufeh. Level-Dependent Quasi-Birth-and-Death Processes. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.
- [34] M. Kimmel and D. E. Axelrod. *Branching Processes in Biology. Interdisciplinary Applied Mathematics 19*. Springer, New York. MR1903571, 2002.
- [35] N. Kontoleon. *The Markovian binary tree: a model of the macroevolutionary process*. PhD thesis, The University of Adelaide, 2006.
- [36] G. Latouche. Algorithms for infinite Markov chains with repeating columns. In *Linear algebra, Markov chains, and queueing models*, pages 231–265. Springer, 1993.
- [37] G. Latouche. Newton’s iteration for non-linear equations in Markov chains. *IMA journal of numerical analysis*, 14(4):583–598, 1994.
- [38] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for Quasi-Birth-and-Death processes. *Journal of Applied Probability*, 30(03):650–674, 1993.
- [39] G. Latouche and V. Ramaswami. *Introduction to matrix analytic methods in stochastic modeling*. SIAM, 1999.
- [40] G. Latouche, M.-A. Remiche, P. Taylor, et al. Transient Markov arrival processes. *The Annals of Applied Probability*, 13(2):628–640, 2003.
- [41] K. Leibnitz and M.-A. Remiche. Modeling the Extinction in the Information Diffusion Process in Wireless Sensor Networks. In *ICCCN*, pages 1–5, 2010.
- [42] E. C. MacRae et al. Matrix derivatives with an application to an adaptive linear decision problem. *The Annals of Statistics*, 2(2):337–346, 1974.
- [43] K. V. Mitov and E. Omey. *Renewal processes*. Springer, 2014.
- [44] C. J. Mode. *Multitype branching processes: Theory and applications*, volume 34. American Elsevier Pub. Co., 1971.
- [45] M. F. Neuts. Computer power or the liberation of applied probability. *Conference on Mathematical Methods in Queueing Theory*, Mimeo Series(312):1–22, 1973.
- [46] M. F. Neuts. Moment formulas for the Markov renewal branching process. *Advances in Applied Probability*, 8(04):690–711, 1976.
- [47] M. F. Neuts. *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Corporation, 1981.
- [48] J. R. Norris. *Markov chains*. Number 2. Cambridge university press, 1998.
- [49] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

- [50] I. Pinelis. Evolutionary models of phylogenetic trees. *Proceedings of the Royal Society of London B: Biological Sciences*, 270(1522):1425–1431, 2003.
- [51] N. Privault. *Understanding Markov chains: examples and applications*. Springer Science & Business Media, 2013.
- [52] S. Ross. *A First Course in Probability 8th Edition*. Pearson, 2009.
- [53] S. M. Ross. *Introduction to probability models*. Academic press, 2014.
- [54] T. H. Savits. The explosion problem for branching Markov process. 1969.
- [55] E. Seneta. *Non-negative matrices and Markov chains*. Springer Science & Business Media, 2006.
- [56] B. Sericola. *Markov Chains: Theory and Applications*. John Wiley & Sons, 2013.
- [57] W. J. Stewart. *Introduction to the numerical solutions of Markov chains*. Princeton Univ. Press, 1994.
- [58] R. Tapia. The Kantorovich theorem for Newton’s method. *The American Mathematical Monthly*, 78(4):389–392, 1971.
- [59] J. Ye and S.-q. Li. Folding algorithm: A computational method for finite QBD processes with level-dependent transitions. *IEEE Transactions on Communications*, 42(234):625–639, 1994.