

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

ANALISI DELLA DINAMICA SU NETWORK

Relatore:
Prof. Armando Bazzani

Presentata da:
Tommaso Pettinari

Anno Accademico 2020/2021

Sommario

Questa tesi si occupa di analizzare attraverso un approccio algebrico la dinamica di sistemi complessi descrivibili da network. Dopo aver definito concetti fondamentali per lo studio di un grafo, come la matrice di adiacenza e la matrice Laplaciana (e il relativo spettro), si mettono in relazione questi ultimi con le caratteristiche della dinamica degli elementi del sistema, in particolare per un tipo di dinamica diffusiva detto random walk. Di tale processo, considerato di rilevante importanza per via della sua applicabilità a numerosi fenomeni, se ne studiano l'esistenza e l'unicità delle soluzioni stazionarie e la risposta a perturbazioni dell'equilibrio.

I tipi di perturbazioni studiate sono di due categorie: perturbazioni della dinamica degli elementi del sistema rispetto allo stato stazionario, in cui viene tuttavia mantenuta fissa la struttura del network, oppure perturbazioni strutturali del network stesso, in cui sono gli stessi collegamenti tra i nodi a subire una modifica.

Attraverso alcune simulazioni numeriche vengono confermati sperimentalmente i principali risultati e, infine, si implementa un sistema di controllo basato su un algoritmo di *adaptive learning* capace di reagire a perturbazioni strutturali e riportare il sistema all'equilibrio originario.

Indice

Introduzione	2
1 Analisi spettrale di un grafo	4
1.1 Nozioni generali sui grafi	4
1.2 Matrice Laplaciana di grafi non direzionati	6
1.2.1 Grafi non pesati	6
1.2.2 Grafi pesati	8
1.2.3 Proprietà dello spettro	9
1.3 Matrice Laplaciana di grafi direzionati	10
2 Dinamica su network	14
2.1 Processi di Markov e Random walk discreto	14
2.2 Soluzione stazionaria e convergenza all'equilibrio	17
2.2.1 Simulazioni di convergenza alla soluzione stazionaria	19
2.2.2 Simulazioni sull'unicità della soluzione stazionaria	21
3 Perturbazioni strutturali del network	25
3.1 Perturbazione non direzionata su grafo non direzionato	25
3.2 Sviluppo di un sistema di controllo	27
3.2.1 Funzione di costo	28
3.2.2 Risultati ottenuti	29
3.3 Estensione al caso direzionato	31
Conclusione e discussione	36
Bibliografia	37

Introduzione

Un grafo, o network, è un ente matematico capace di descrivere in modo quantitativo le relazioni e i processi esistenti all'interno di un insieme di elementi. I sistemi formati da molteplici agenti che interagiscono tra di loro in modo complesso, trovano una naturale rappresentazione nei grafi, che permettono di studiare come la struttura della rete di relazioni influenzi la dinamica del sistema. Lo spostamento di automobili in una rete stradale, il flusso d'acqua nella rete idrica di una città, la regolazione enzimatica di reazioni chimiche che avvengono in una cellula, il *world wide web*; tutti questi sistemi apparentemente scollegati tra loro sono in realtà molto simili se visti dal punto di vista della teoria dei grafi e, una volta formalizzati nel corretto linguaggio, si comportano nello stesso modo, con una dinamica che segue le stesse leggi.

Nonostante la grande versatilità dei network, che ne permette l'impiego per modellizzare una miriade di diversi fenomeni, dalla biologia all'informatica, dalla sociologia all'urbanistica, passando ovviamente per la fisica e la chimica, lo sviluppo di questa branca della scienza è stato piuttosto lento e si è dovuto aspettare la seconda metà del XX secolo per far sì che la comunità scientifica iniziasse a studiarli in modo approfondito e sistematico, e soprattutto a capirne la potenza e duttilità (anche grazie alla possibilità di effettuare simulazioni numeriche al computer). Negli ultimi 50 anni la ricerca scientifica è progredita in modo incredibile sia dal punto di vista teorico, sia da quello applicativo e computazionale, ma tuttavia l'argomento è tutt'altro che esaurito e la ricerca in questo campo è vivace e attiva.

In tale contesto si inserisce questa tesi, che si propone di analizzare l'argomento della dinamica sui network studiandola prima da un punto di vista teorico, corredando quindi l'analisi con simulazioni che confermino i risultati teorici, e poi da un punto di vista applicativo, agendo sulla dinamica di un grafo in modo da controllarla e modificarla a nostra discrezione. Nel primo capitolo si presentano i fondamenti e alcuni risultati cardine della teoria spettrale (o algebrica) dei grafi, inizialmente sulle tracce del lavoro di F. R. K. Chung sullo spettro dei grafi non direzionati [1], che con il suo lavoro ha stabilito delle convenzioni ormai globalmente accettate e riconosciute. La trattazione viene poi estesa al caso più generale dei grafi direzionati [2, 3, 4, 5].

Il secondo capitolo riguarda lo studio della dinamica su network, in particolare delle catene di Markov [6, 7] e dei random walk [8], che rappresentano processi diffusivi su

domini discreti e finiti (in realtà i network possono essere anche infiniti, ma tale eventualità non verrà discussa in questo lavoro) e vengono assunti come esempio fondamentale di sistemi dinamici su grafi. L'argomento centrale del capitolo è il legame esistente tra la struttura del network e la soluzione stazionaria della dinamica. Ai risultati teorici si affiancano conferme sperimentali, ottenute tramite simulazioni al computer effettuate su varie tipologie di network.

Infine, il terzo capitolo si concentra sulle conseguenze che una perturbazione strutturale del grafo ha sulla dinamica, ovvero cosa succede ai processi tra gli elementi di un insieme se le relazioni tra di loro cambiano. Si propone, inoltre, l'implementazione di un sistema di controllo capace di reagire all'arrivo di perturbazioni strutturali incognite in un grafo, ristabilendo nel modo più fedele possibile l'equilibrio originario.

Capitolo 1

Analisi spettrale di un grafo

1.1 Nozioni generali sui grafi

Un grafo è la rappresentazione matematica di un insieme finito di oggetti, collegati tra loro da relazioni di un certo tipo. Per sua natura, dunque, un grafo è una struttura matematica discreta.

Definizione 1.1. Si definisce grafo $G = (V(G), E(G))$ una coppia ordinata di insiemi, in cui $V(G) = \{v_1, v_2, \dots, v_N\}$ indica l'insieme dei nodi, o vertici, mentre $E(G) = \{(v_i, v_j) \mid (v_i, v_j) \in V \times V\} \subseteq (V \times V)$ rappresenta l'insieme dei segmenti, o link.

Un grafo può essere direzionato o non direzionato se le coppie che costituiscono gli elementi di E sono coppie ordinate oppure no. Nel primo caso, dunque, si distingue il segmento (v_i, v_j) dal segmento con senso di percorrenza opposto (v_j, v_i) . Come si può vedere dalla definizione, in un grafo generico non sono impediti link tra un nodo e se stesso. In tal caso il segmento viene detto loop o cappio.

Definizione 1.2. Se la coppia $(v_i, v_j) \in E(G)$, allora v_i e v_j vengono detti adiacenti e si scrive $v_i \sim v_j$.

Nel caso di grafi non direzionati, la relazione di adiacenza è una relazione simmetrica, poiché se $(v_i, v_j) \in E(G)$ allora anche $(v_j, v_i) \in E(G)$. Se un nodo v_i non è adiacente a nessun altro nodo del grafo viene detto nodo isolato. Se, invece, ogni nodo del grafo è adiacente a ciascuno degli altri nodi, il grafo si dice completo.

Definizione 1.3. Si definisce funzione di peso, o semplicemente peso, la funzione $w: V(G) \times V(G) \rightarrow \mathbb{R}$ che associa ad ogni coppia di vertici del grafo un numero reale, il peso del collegamento tra i due vertici appunto, con le seguenti condizioni:

- i) $w(v_i, v_j) = w_{ij} \geq 0 \quad \forall i, j \in \{1, 2, \dots, N\}$
- ii) $w_{ij} = 0 \iff (v_i, v_j) \notin E(G)$

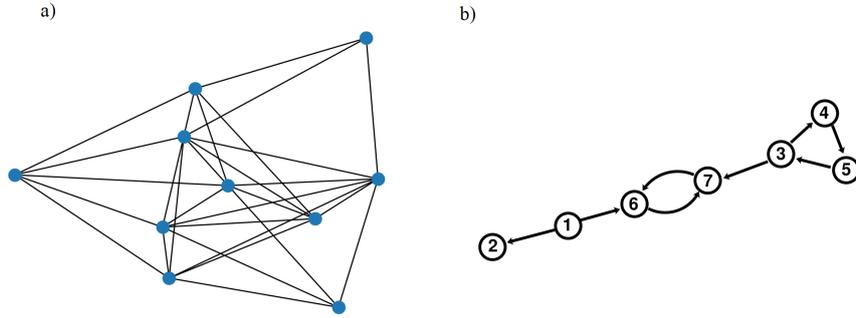


Figura 1.1: Esempi di grafi non direzionati (a) e direzionati (b). I nodi sono rappresentati da dei punti, i link da delle linee che li collegano. Si noti come nel caso direzionato, i link abbiano una freccia ad indicare il verso.

Un grafo con una funzione di peso associata si definisce grafo pesato. Chiaramente per un grafo non direzionato vale $w_{ij} = w_{ji} \quad \forall i, j \in \{1, 2, \dots, N\}$, mentre non ci sono vincoli di simmetria nel caso direzionato. Un sottoinsieme di notevole importanza dell'insieme dei grafi pesati è quello per cui le funzioni di peso assumono soltanto il valore 0 o 1. Un grafo con una funzione di peso del genere viene denominato grafo non pesato, dato che il peso associato ad ogni coppia di nodi indica semplicemente l'esistenza o meno di un segmento tra di essi.

Definizione 1.4. Si definiscono per un grafo non direzionato il grado d_i del vertice v_i e il volume del grafo $vol G$ rispettivamente

$$d_i = \sum_{j=1}^N w_{ij}$$

$$vol G = \sum_{i=1}^N d_i.$$

Similmente, nel caso di grafo direzionato, è possibile distinguere e definire un grado entrante $d_i^{in} = \sum_{j=1}^N w_{ji}$ e un grado uscente $d_i^{out} = \sum_{j=1}^N w_{ij}$. Per un grafo non direzionato $d_i^{out} = d_i^{in} = d_i$.

Nel caso di un grafo non pesato, in cui w_{ij} vale solo 0 o 1, il grado conta semplicemente il numero di collegamenti del nodo (entranti per d_i^{in} o uscenti per d_i^{out} , mentre è indifferente nel caso non direzionato).

Risulta naturale associare a ciascun grafo due particolari matrici: la matrice di adiacenza e la matrice dei gradi.

Definizione 1.5. Si definisce la matrice di adiacenza A come $A_{ij} = w_{ij}$. Nel caso di grafo non pesato la matrice di adiacenza diventa semplicemente:

$$A_{ij} = \begin{cases} 1 & \text{se } v_i \sim v_j \\ 0 & \text{altrimenti.} \end{cases}$$

Definizione 1.6. Si definisce la matrice dei gradi T come la matrice diagonale avente nell'entrata (i, i) l'elemento d_i (d_i^{out} nel caso di grafi direzionati).

1.2 Matrice Laplaciana di grafi non direzionati

Si procede ora con lo studio della matrice Laplaciana, che permette di ricavare numerose proprietà e caratteristiche del grafo a cui è associata. Verrà analizzato prima il caso di grafi non direzionati, partendo dal semplice grafo non pesato per poi considerare il generico grafo pesato. Nelle sezioni successive si prenderanno in esame anche i grafi direzionati.

1.2.1 Grafi non pesati

Sia G un grafo non direzionato non pesato, senza loop, con $V(G) = \{v_1, v_2, \dots, v_N\}$. Per iniziare si consideri la matrice L definita come $L = T - A$. In particolare:

$$L(v_i, v_j) = L_{ij} = \begin{cases} d_i & \text{se } v_i = v_j \\ -1 & \text{se } v_i \sim v_j \\ 0 & \text{altrimenti.} \end{cases}$$

A partire da questa matrice possiamo definire la matrice Laplaciana.

Definizione 1.7. Si definisce matrice Laplaciana (o matrice Laplaciana normalizzata) la matrice $\mathcal{L} = T^{-1/2}LT^{-1/2}$, con la convenzione che $T_{i,i}^{-1} = 0$ se $d_i = 0$, cioè se il nodo i -esimo è isolato. Per un grafo non pesato le entrate della matrice sono:

$$\mathcal{L}(v_i, v_j) = \mathcal{L}_{ij} = \begin{cases} 1 & \text{se } v_i = v_j \text{ e } d_i \neq 0 \\ -\frac{1}{\sqrt{d_i d_j}} & \text{se } v_i \sim v_j \\ 0 & \text{altrimenti.} \end{cases}$$

La matrice Laplaciana è anche esprimibile come $\mathcal{L} = I - T^{-1/2}AT^{-1/2}$, con A matrice di adiacenza.

\mathcal{L} può essere vista come un operatore sullo spazio $C(G)$ delle funzioni g che associano a ciascun nodo v_i del grafo G un numero reale $g(v_i)$. Ogni elemento di tale spazio può essere visto come un vettore colonna $g = (g(v_1), \dots, g(v_N))^T = (g_1, \dots, g_N)^T$. Si ha dunque che l'azione di $\mathcal{L} : C(G) \rightarrow C(G)$ è:

$$\begin{aligned} (\mathcal{L}g)_j &= g_j - \sum_{\substack{v_i \\ v_j \sim v_i}} \frac{1}{\sqrt{d_j d_i}} g_i \\ &= \sum_{\substack{v_i \\ v_j \sim v_i}} \left(\frac{g_j}{d_j} - \frac{g_i}{\sqrt{d_j d_i}} \right) \\ &= \frac{1}{\sqrt{d_j}} \sum_{\substack{v_i \\ v_j \sim v_i}} \left(\frac{g_j}{\sqrt{d_j}} - \frac{g_i}{\sqrt{d_i}} \right) \end{aligned} \quad (1.1)$$

Come si può evincere dalla definizione 1.7, per un grafo non direzionato la matrice \mathcal{L} risulta simmetrica, perciò diagonalizzabile e con autovalori reali. Per trovare tali autovalori utilizziamo il metodo variazionale di Rayleigh-Ritz [1]. Consideriamo, quindi, una funzione arbitraria $g \in C(G)$. Definiamo il funzionale

$$\mathfrak{F}[g] = \frac{\langle g, \mathcal{L}g \rangle}{\langle g, g \rangle} = \frac{\langle g, T^{-1/2} L T^{-1/2} g \rangle}{\langle g, g \rangle}$$

dove $\langle a, b \rangle$ denota il prodotto scalare standard di \mathbb{R}^n . Si effettua quindi la sostituzione $g = T^{1/2} f$ ottenendo

$$\begin{aligned} \mathfrak{F}[f] &= \frac{\langle f, Lf \rangle}{\langle T^{1/2} f, T^{1/2} f \rangle} \\ &= \frac{\sum_{v_i \sim v_j} (f_i - f_j)^2}{\sum_i f_i^2 d_i}, \end{aligned} \quad (1.2)$$

in cui il simbolo $\sum_{v_i \sim v_j}$ indica la sommatoria su tutte le coppie $\{v_i, v_j\}$ non ordinate tali per cui v_i e v_j sono adiacenti. Tale rapporto prende il nome di quoziente di Rayleigh. Il minimo di \mathfrak{F} al variare di f corrisponde al più piccolo autovalore λ_0 di \mathcal{L} , che risulta essere 0 quando la funzione f è uguale al vettore $\mathbf{1} = (1, 1, \dots, 1)$. A tale autovalore corrisponde l'autofunzione $g = T^{1/2} f = T^{1/2} \mathbf{1}$. Dall'equazione 1.2 si possono ottenere tutti gli autovalori di \mathcal{L} , che risultano chiaramente essere tutti non negativi.

Definizione 1.8. Siano $0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}$ gli autovalori di \mathcal{L} . L'insieme $\{\lambda_0, \lambda_1, \dots, \lambda_{n-1}\}$ viene denominato spettro del grafo G .

L'espressione dei successivi autovalori si ottiene escludendo dallo spazio di ricerca gli autospazi generati dalle autofunzioni degli autovalori già trovati. Ovvero:

$$\lambda_G = \lambda_1 = \inf_{f \perp \mathbf{1}} \frac{\sum_{v_i \sim v_j} (f_i - f_j)^2}{\sum_i f_i^2 d_i} \quad (1.3)$$

mentre in generale per l'autovalore λ_k si ha

$$\lambda_k = \inf_{f \perp P_{k-1}} \frac{\sum_{v_i \sim v_j} (f_i - f_j)^2}{\sum_i f_i^2 d_i} \quad , \quad (1.4)$$

dove P_{k-1} indica lo spazio generato dalle funzioni f_i corrispondenti agli autovalori λ_i con $i \leq k-1$. Le funzioni f_i vengono anche dette autofunzioni armoniche di \mathcal{L} . L'autovalore maggiore, infine, è esprimibile come

$$\lambda_{n-1} = \sup_f \frac{\sum_{v_i \sim v_j} (f_i - f_j)^2}{\sum_i f_i^2 d_i} \quad . \quad (1.5)$$

1.2.2 Grafi pesati

Generalizziamo le definizioni della sezione precedente al caso di grafi pesati, eventualmente con loop. Ricordando che, secondo la definizione 1.4, il grado d_i di un nodo v_i di un grafo pesato è la somma dei pesi w_{ij} dei segmenti aventi come estremo il nodo considerato, ridefiniamo la matrice L :

$$L(v_i, v_j) = L_{ij} = \begin{cases} d_i - w_{ii} & \text{se } v_i = v_j \\ -w_{ij} & \text{se } v_i \sim v_j \\ 0 & \text{altrimenti.} \end{cases}$$

La matrice Laplaciana, definita sempre come $\mathcal{L} = T^{-1/2} L T^{-1/2}$, diventa:

$$\mathcal{L}(v_i, v_j) = \mathcal{L}_{ij} = \begin{cases} 1 - \frac{w_{ij}}{d_i} & \text{se } v_i = v_j \text{ e } d_i \neq 0 \\ -\frac{w_{ij}}{\sqrt{d_i d_j}} & \text{se } v_i \sim v_j \\ 0 & \text{altrimenti.} \end{cases}$$

L'azione della Laplaciana su una funzione $f \in G(C)$ ora deve tenere conto del peso di ciascun link. Considerando per semplicità la matrice L (che differisce da \mathcal{L} solo per fattori di normalizzazione) si ha che:

$$(Lf)_j = \sum_{\substack{v_i \\ v_j \sim v_i}} (f_j - f_i) w_{ji} \quad . \quad (1.6)$$

Gli autovalori di \mathcal{L} si possono definire nello stesso modo del caso non pesato attraverso il metodo variazionale. Si ottiene il quoziente di Rayleigh leggermente modificato, con il peso di ciascun link che compare come nell'equazione 1.6. Ad esempio, per il generico autovalore λ_k si ha:

$$\begin{aligned}\lambda_k &= \inf_{g \perp T^{1/2} P_{k-1}} \frac{\langle g, Lg \rangle}{\langle g, g \rangle} \\ &= \inf_{f \perp P_{k-1}} \frac{\sum_{v_i \sim v_j} (f_i - f_j)^2 w_{ij}}{\sum_i f_i^2 d_i} .\end{aligned}\tag{1.7}$$

Il primo autovalore λ_0 sarà chiaramente nullo anche in questo caso.

1.2.3 Proprietà dello spettro

Per conoscere con esattezza gli autovalori è necessario sapere i collegamenti, e i pesi di tali collegamenti, tra tutti i nodi del grafo. È possibile, tuttavia, delimitare degli intervalli in cui gli autovalori λ_i sono confinati senza conoscere a fondo le connessioni di G , ma a partire da proprietà generali o da grandezze caratteristiche. Si enuncia a tale proposito il seguente lemma [1]:

Teorema 1.1. *Sia G un grafo con n nodi. Si ha che:*

i)

$$\sum_{i=0}^{n-1} \lambda_i \leq n$$

con l'uguaglianza valida se e solo se G non ha vertici isolati.

ii) Per un grafo completo con $n \geq 2$, vale

$$\lambda_i = \frac{n}{n-1} \quad \forall i \in \{1, \dots, n-1\}.$$

Per un grafo non completo, invece

$$\lambda_1 \leq 1$$

Inoltre, per un grafo senza nodi isolati si ha che

$$\lambda_{n-1} \geq \frac{n}{n-1}$$

iii) Se G è un grafo connesso, ovvero esiste un percorso che collega due qualsiasi suoi nodi, allora $\lambda_1 > 0$. Se $\lambda_i = 0$ e $\lambda_{i+1} > 0$, allora il grafo ha esattamente $i + 1$ componenti connesse.

iv) Per ogni $i \leq n - 1$

$$\lambda_i \leq 2$$

con l'uguaglianza valida solo per $i = n - 1$ e se e solo se il grafo è bipartito (ovvero un grafo in cui l'insieme dei vertici è divisibile in due sottoinsiemi tali per cui ogni nodo di una delle due parti è collegato soltanto ad un vertice dell'altro sottoinsieme).

v) Lo spettro di un grafo è l'unione degli spettri delle sue componenti connesse.

1.3 Matrice Laplaciana di grafi direzionati

Per i grafi direzionati la definizione di matrice Laplaciana non è immediata come per il caso non direzionato. La ragione di questo fatto si trova nella perdita di simmetria della matrice di adiacenza A , e quindi anche della matrice Laplaciana per come definita nella definizione 1.7. Una Laplaciana così definita non è necessariamente diagonalizzabile con valori reali, ma nonostante ciò non si perde del tutto la capacità di descrivere un grafo tramite lo spettro di \mathcal{L} . Occorre soltanto adattare la definizione della Laplaciana, generalizzandola in modo appropriato.

Prima di tutto, è necessario introdurre alcune definizioni. Se per i grafi non direzionati, la nozione di connessione è immediata ed è facilmente visualizzabile in forma grafica, non vale lo stesso per i digrafi. La direzionalità dei collegamenti tra un nodo e un altro permette di definire diversi gradi di connessione per un digrafo. È necessario introdurre la nozione di “grafo sottostante”, cioè il grafo non direzionato ottenuto trascurando la direzionalità dei collegamenti di un digrafo.

Definizione 1.9. Sia G un digrafo.

- i) G si dice fortemente connesso se per ogni coppia ordinata di nodi (i, j) , esiste un cammino che congiunge i con j .
- ii) G si dice connesso unilateralmente se per ogni coppia ordinata di nodi (i, j) , esiste un cammino che congiunge i con j oppure j con i .
- iii) G si dice debolmente connesso se il grafo sottostante è connesso.
- iv) G è un grafo non connesso se non è debolmente connesso.

Un sottografo fortemente connesso viene anche detto componente fortemente connessa (SSC abbreviato, dall'inglese *strongly connected component*). La relazione esistente tra due nodi qualsiasi appartenenti ad una SSC è una relazione di equivalenza e permette perciò di partizionare univocamente un grafo direzionato G [5].

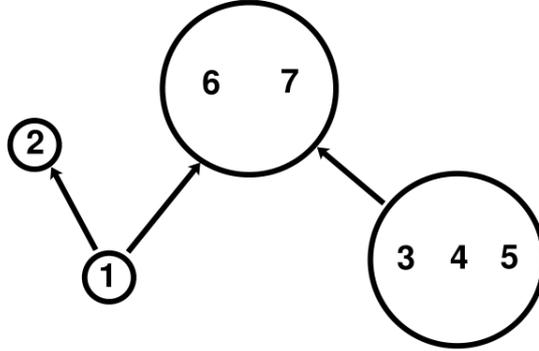


Figura 1.2: Condensazione del grafo direzionato di figura 1.1. Ciascun nodo ora rappresenta una SCC.

Definizione 1.10. Sia G un grafo direzionato, definiamo la condensazione di G , $SC[G]$, come il grafo ottenuto identificando come un unico nodo ciascuna SSC.

Per introdurre la matrice Laplaciana per i digrafi, consideriamo la forma della matrice L per grafi non direzionati, che rappresenta la matrice \mathcal{L} prima della procedura di normalizzazione. Per definizione $L = T - A$, con T matrice dei gradi e A matrice di adiacenza. Le matrici A e T possono essere utilizzate per generare una matrice $P = T^{-1}A$ stocastica, ovvero tale per cui le sue entrate sono tutte non negative e la somma degli elementi di ciascuna riga è uguale ad 1 (P rappresenta, ad esempio, la probabilità di transizione da un nodo ad un altro in un random walk su grafo, come vedremo nei capitoli successivi). La matrice P che si ottiene ha componenti

$$p_{ij} = \begin{cases} \frac{w_{ij}}{d_i} & \text{se } v_i \sim v_j \\ 0 & \text{altrimenti,} \end{cases} \quad (1.8)$$

dove si ricorda che con d_i , nel caso di grafi direzionati, si intende d_i^{out} . La matrice L , dunque, si può scrivere come

$$L = T - TP \quad . \quad (1.9)$$

Si utilizzerà questa come definizione di matrice Laplaciana per un grafo direzionato. La matrice L avrà una struttura fortemente dipendente dalla connettività del grafo.

Per analizzare la dipendenza di L dalla connettività si procede per step: prima si fornisce un risultato valido esclusivamente per un grafo fortemente connesso, successivamente si estende l'analisi ad un generico digrafo. Per un grafo fortemente connesso vale la seguente proposizione.

Proposizione 1.2. Sia G un grafo fortemente connesso. La Laplaciana $L(G)$ ha un autovalore 0 con molteplicità geometrica e algebrica pari a 1.

La Laplaciana di un digrafo fortemente connesso, dunque, possiede un nucleo unidimensionale così come la Laplaciana di un grafo non direzionato connesso.

$$\dim \ker L(G) = 1$$

Nel caso di un generico digrafo si considerino due qualsiasi componenti fortemente connesse di G , SCC_i e SCC_j . Si dice che SCC_i precede SCC_j se esiste un link (ovviamente direzionato) tra un nodo $v_{i'} \in SCC_i$ e un nodo $v_{j'} \in SCC_j$. Si indica questa relazione con $SCC_i \preceq SCC_j$. La relazione di precedenza tra componenti fortemente connesse è una relazione di ordine parziale: è transitiva, riflessiva e antisimmetrica. Riguardo l'antisimmetria, è chiaro che se $SCC_i \preceq SCC_j$ e $SCC_j \preceq SCC_i$, allora $SCC_i = SCC_j$, poiché qualsiasi coppia di nodi delle due SCC sarebbe mutualmente connessa, così che SCC_i e SCC_j costituirebbero una unica componente SCC . Attraverso questa relazione di ordine parziale, è possibile effettuare una reindicizzazione dei nodi del grafo. Si enuncia, quindi, la seguente proposizione [5].

Proposizione 1.3. La matrice Laplaciana di un grafo direzionato con p componenti fortemente connesse, in seguito alla reindicizzazione dei suoi nodi secondo la relazione di precedenza tra le SCC , è in forma triangolare superiore a blocchi.

$$L(G) = \left(\begin{array}{ccc|ccc} L_1 & * & * & * & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & L_{p-q} & * & \cdots & * \\ \hline 0 & \cdots & 0 & L_{p-q+1} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & L_p \end{array} \right) \quad (1.10)$$

Inoltre, il polinomio caratteristico di L è il prodotto dei polinomi caratteristici dei vari blocchi quadrati L_i , con L_i che rappresenta la Laplaciana ristretta alla singola SCC_i .

Nella matrice 1.10 gli asterischi $*$ denotano blocchi non necessariamente nulli. Si noti come gli ultimi q blocchi lungo la diagonale (da L_{p-q+1} a L_p) rappresentino delle SCC terminali, ovvero elementi massimali secondo la relazione d'ordine parziale definita precedentemente. Questi blocchi hanno collegamenti entranti provenienti dalle prime $p - q$ componenti, ma non hanno collegamenti uscenti (il quadrante in basso a destra è diagonale a blocchi, quello in basso a sinistra è totalmente nullo). Dato che la matrice Laplaciana è stata costruita utilizzando i gradi uscenti, d_i^{out} , il blocco L_i relativo ad una SCC terminale è esattamente uguale alla matrice Laplaciana che si sarebbe ottenuta considerando la SCC_i come un grafo fortemente connesso a sé stante (cioè trascurando i link con i vertici delle altre componenti).

$$L_i = L(SCC_i)$$

Vale dunque la proposizione 1.2 e ciascun blocco L_i ha un nucleo monodimensionale. Sia m la dimensione del blocco L_i , l'autovettore $\rho_i \in \mathbb{R}^m$ relativo all'autovalore nullo può essere esteso dimensionalmente in \mathbb{R}^n , con n numero di nodi del grafo G , semplicemente fissando a 0 le coordinate per i vertici non appartenenti a SCC_i . I nuovi $\rho_i^* \in \mathbb{R}^n$ sono evidentemente linearmente indipendenti tra loro.

Si fornisce ora un lemma [5] a riguardo dei blocchi L_i relativi alle componenti SCC non terminali.

Lemma 1.4. *In un grafo G reindicizzato in modo tale da avere la matrice Laplaciana nella forma 1.10, i blocchi diagonali corrispondenti a SCC non terminali sono invertibili.*

L'invertibilità ha come conseguenza immediata che 0 non è autovalore per questi blocchi. Si può quindi enunciare il seguente teorema.

Teorema 1.5. *Sia G un digrafo. La molteplicità algebrica e geometrica dell'autovalore 0 della matrice Laplaciana $L(G)$ è pari al numero di componenti fortemente connesse terminali. Tutti gli autovalori non nulli hanno parte reale positiva.*

Capitolo 2

Dinamica su network

Ci occupiamo ora di definire ed analizzare dei processi dinamici sui network, focalizzando l'attenzione in particolare su un tipo di processo stocastico denominato random walk, capace di modellizzare un gran numero di possibili tipologie di dinamica su grafo. In generale possiamo pensare ad un processo dinamico su un grafo come allo spostamento di alcuni elementi (siano essi particelle, persone o eventualmente concetti astratti come informazioni) da un nodo ad un altro attraverso i link che li uniscono secondo un processo diffusivo. La direzione verso cui tali elementi si spostano può essere decisa in modo casuale o in modo "consapevole". Il random walk è chiaramente adatto per descrivere processi del primo tipo, data la loro natura stocastica, e in alcune situazioni può essere esteso anche al secondo caso. Se ciascuno degli elementi che si muovono nel grafo prende la decisione sullo spostamento successivo in modo indipendente dalle decisioni degli altri e dai suoi step precedenti, la dinamica è descrivibile attraverso una catena di Markov, di cui il random walk rappresenta un'applicazione. Se, inoltre, il numero di elementi è tale da poter permettere una statistica sulle scelte prese per muoversi, la frequenza con cui una particolare direzione viene scelta può essere interpretata anche come una probabilità.

2.1 Processi di Markov e Random walk discreto

Si consideri un insieme finito I di N elementi e una variabile aleatoria X che può assumere come valore uno qualsiasi degli elementi $i \in I$, detti stati. A ciascuno degli stati è inoltre associato un valore μ_i finito e non negativo. Si definisca il vettore μ con componenti μ_i misura su I , e in particolare *distribuzione* se $\sum_{i \in I} \mu_i = 1$. La distribuzione μ rappresenta la probabilità di trovare la variabile aleatoria in uno degli stati, ovvero

$$\mu_i = \mathbb{P}(X = i) \quad .$$

Sia, infine, P una matrice stocastica di dimensioni $N \times N$, per cui ogni riga di P rappresenta una distribuzione su I .

Definizione 2.1. Si definisce un cosiddetto processo (o catena) di Markov $(X_n)_{n \geq 0}$ una qualsiasi sequenza di variabili aleatorie a valori in I che rispetti le seguenti proprietà:

- i) X_0 ha distribuzione di probabilità μ ;
- ii) per ogni $n \geq 0$, sapendo che $X_n = i$, il successivo elemento della catena di Markov X_{n+1} ha distribuzione data dalla riga i -esima della matrice P , ovvero dagli elementi di matrice p_{ij} con $j \in I$.

Si dice, in breve, che $(X_n)_{n \geq 0}$ è *Markov* (μ, P) . Si può notare chiaramente che questa definizione è valida esclusivamente per step temporali discreti. Ci occuperemo in seguito del caso continuo, mentre ora si ritiene necessario enunciare la particolarità caratterizzante dei processi di Markov, ovvero l'indipendenza di ciascuna variabile aleatoria X_n da tutti i precedenti stati assunti dalla catena, eccezion fatta per il diretto precedente X_{n-1} . Tale caratteristica è conosciuta sotto il nome di proprietà di Markov. Da questa proprietà deriva il seguente teorema, che permette di valutare la distribuzione di probabilità dopo un qualsiasi numero di step temporali [6, 7].

Teorema 2.1. Sia $(X_n)_{n \geq 0}$ un processo Markov (μ, P) . Allora, per ogni $n \geq 0$, si ha che

$$\mathbb{P}(X_n = j) = (\mu P^n)_j \quad . \quad (2.1)$$

Si definisca, quindi, un random walk come un processo in cui un certo numero di elementi, detti *walker*, si muovono in modo casuale e con passi discreti in uno spazio che può essere infinito (random walk sulla retta o in generale in \mathbb{R}^n) o finito (come in un grafo). Un random walk su un grafo G rappresenta un chiaro esempio di catena di Markov. È sufficiente considerare come insieme I di stati possibili l'insieme dei vertici $V(G)$, mentre la concentrazione iniziale degli walker nei diversi nodi rappresenta la distribuzione di probabilità μ della variabile aleatoria X . Per quanto riguarda la matrice di probabilità P , questa è univocamente determinata dalla struttura del grafo: come già detto nel paragrafo 1.3, date le matrici di adiacenza A e dei gradi T , si può ottenere una matrice stocastica $P = T^{-1}A$. Dunque i collegamenti e i pesi del grafo determinano in modo univoco le probabilità di transizione del random walk. A titolo di esempio, si consideri il più semplice random walk possibile: dato un grafo non direzionato e non pesato, la probabilità di transizione da un nodo i ad un suo nodo adiacente j è pari a $p_{ij} = \frac{1}{d_i}$, ovvero ogni walker può capitare in un nodo adiacente ad i con probabilità uniforme. È banale osservare che $\sum_j p_{ij} = 1$, e altrettanto facilmente si può estendere questo tipo di random walk al caso pesato:

$$p_{ij} = \frac{A_{ij}}{d_i} = \frac{w_{ij}}{d_i} \quad . \quad (2.2)$$

L'equazione 2.2 ha validità generale, e vedremo ora il perché.

Si consideri un altro tipico esempio di random walk, cioè il cosiddetto *degree-biased* random walk, in cui la probabilità p_{ij} dipende dal prodotto tra i gradi d_i e d_j dei nodi di

partenza e di arrivo. In pratica un walker ha maggiore probabilità di andare in un nodo con molti collegamenti rispetto ad un altro con meno link. In generale si può scrivere

$$p_{ij} = \frac{(d_i \cdot d_j)^\alpha}{\sum_{i \sim l} (d_i \cdot d_l)^\alpha} \quad , \quad (2.3)$$

che è chiaramente diversa dall'equazione 2.2. Tuttavia, ridefinendo la matrice di adiacenza $A'_{ij} = (d_i \cdot d_j)^\alpha$, si riottiene l'espressione 2.2, a conferma del fatto che è possibile associare in modo univoco un random walk ad un grafo. Un *degree-biased* random walk su un grafo con matrice di adiacenza A , così come qualsiasi altro random walk che si voglia definire, può essere ricondotto al random walk standard semplicemente ridefinendo la matrice di adiacenza, ovvero riscalandolo i pesi del grafo.

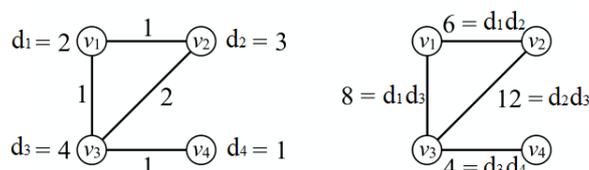


Figura 2.1: Esempio di *degree biased* random walk. I pesi dei collegamenti del grafo a destra sono ottenuti moltiplicando tra loro i gradi del grafo originale (a sinistra).

Una nozione importante per i processi di Markov, e dunque anche per i random walk, è la *comunicabilità*. Siano i e j due elementi di I , si dice che i porta a j , $i \rightarrow j$, se $\mathbb{P}(X_n = j \mid X_0 = i, \text{ per qualche } n \geq 0) = p_{ij}^n > 0$, dove p_{ij}^n indica per convenzione l'elemento (i, j) della matrice P^n , mentre si dice che i comunica con j , $i \leftrightarrow j$, se i porta a j e viceversa. La relazione di comunicazione è una relazione di equivalenza (transitiva, simmetrica e riflessiva) perciò divide gli stati di I in classi di equivalenza, dette classi di comunicazione. Una classe C si dice chiusa se

$$(i \in C, i \rightarrow j) \implies j \in C \quad .$$

Ciò significa che una volta che la variabile aleatoria X ha assunto uno stato della classe chiusa, non può assumere valori al di fuori della classe. Se una classe chiusa è composta da un solo stato, tale stato viene detto *assorbente*, mentre se l'intero insieme I è una classe chiusa, la catena di Markov viene detta irriducibile. La relazione che queste classi hanno con i grafi è chiara: se consideriamo $I = V(G)$, una classe di comunicazione corrisponde ad una componente fortemente connessa (*SCC*) di un grafo direzionato, mentre una classe chiusa è l'equivalente di una *SCC* terminale. Dunque, parlare di catena di Markov irriducibile o di random walk su un grafo fortemente connesso è la stessa identica cosa.

2.2 Soluzione stazionaria e convergenza all'equilibrio

L'evoluzione della concentrazione (o della distribuzione di probabilità) μ di walkers sui diversi nodi del grafo evolve, come stabilito dal teorema 2.1, secondo la seguente legge:

$$\mu_{n+1} = \mu_n P \quad . \quad (2.4)$$

Si definisce, dunque, la distribuzione stazionaria o invariante π del random walk l'autovettore sinistro con autovalore pari a 1 della matrice stocastica P :

$$\pi = \pi P \quad . \quad (2.5)$$

Teorema 2.2. *Sia P la matrice stocastica associata ad un random walk irriducibile su un grafo finito. Allora il random walk ammette un'unica distribuzione stazionaria π tale per cui $\pi = \pi P$.*

Il teorema 2.2 implica che ogni grafo non direzionato connesso e ogni digrafo fortemente connesso ammettono una soluzione stazionaria per il random walk. Non esiste tuttavia un risultato analogo per un generico digrafo. Si noti inoltre come tale teorema valga soltanto nel caso di grafi finiti.

Il fatto che lo stato stazionario del random walk corrisponda all'autovettore con autovalore unitario della matrice P ci suggerisce di analizzare lo spettro della matrice $P = T^{-1}A$. Ricordiamo che $\mathcal{L} = I - T^{-1/2}AT^{-1/2}$, che può essere riscritto come

$$\mathcal{L} = I - T^{1/2}PT^{-1/2} \quad . \quad (2.6)$$

Dall'equazione 2.6 vediamo che lo spettro della matrice P e della laplaciana sono legati da una relazione. Infatti $T^{1/2}PT^{-1/2}$ è una matrice simile a P e ne condivide lo spettro. Inoltre

$$\text{se } Mv = \lambda v \quad \implies \quad (M + \mathbf{1})v = (\lambda + 1)v \quad ,$$

da cui si ottiene che gli autovalori di \mathcal{L} sono gli opposti di quelli di P , traslati di un'unità. L'autovalore 1 di P corrisponde all'autovalore nullo della laplaciana normalizzata, e in generale l'intervallo $[0, 2]$ in cui gli autovalori di \mathcal{L} sono contenuti, viene traslato nell'intervallo $[-1, 1]$. Valgono, perciò, tutti i teoremi enunciati nel primo capitolo, tra cui quelli che affermano l'unicità dell'autovalore nullo per grafi non direzionati connessi e per digrafi fortemente connessi, a conferma del teorema 2.2. Anche gli autovettori delle due matrici sono legati tra loro, dato che la relazione di similarità tra $T^{1/2}PT^{-1/2}$ e P comporta un riscaldamento delle componenti degli autovettori: sia v_i un autovettore sinistro della matrice Laplaciana \mathcal{L} , allora $\tilde{v}_i = v_i T^{1/2}$ è l'autovettore corrispondente per P (ogni componente viene moltiplicata per la radice del grado del nodo corrispondente).

Per i grafi non direzionati è possibile ottenere analiticamente l'autovettore di P che rappresenta la distribuzione stazionaria, ovvero

$$\pi_i = \frac{d_i}{\sum_{l=1}^N d_l} \quad \text{o, in forma vettoriale,} \quad \pi = \frac{\mathbf{1}T}{\text{vol } G} \quad (2.7)$$

che può essere esteso anche a digrafi fortemente connessi, purché $d_{in} = d_{out} \equiv d_i$. Nell'equazione, N è il numero di nodi del grafo, mentre $\mathbf{1}$ è il vettore contenente tutti 1.

La velocità con cui la distribuzione di walker nel grafo tende alla distribuzione stazionaria dipende dalla nozione di convergenza che si utilizza. In ogni caso, comunque, è possibile stabilire dei parametri che indicano quanto velocemente la soluzione di equilibrio viene raggiunta. Sviluppando la matrice P^n in forma diagonale si ottiene [1, 8]

$$\mu_n = \mu_0 P^n = \sum_{i=0}^{N-1} \lambda_i^n \tilde{v}_i a_i, \quad (2.8)$$

dove \tilde{v}_i è l'autovettore corrispondente all'autovalore λ_i , mentre a_i rappresenta una sorta di proiezione della distribuzione di probabilità di partenza sugli autovettori di P . Complessivamente, l'equazione 2.8 può essere considerata come una trasformata di Fourier su grafo, che scompone una funzione lungo i modi principali. Per $n \rightarrow \infty$, sapendo che $\lambda_0 = 1$ e che per gli altri autovalori vale $|\lambda_i| < 1$ (non consideriamo il caso di grafi bipartiti, per cui, come visto nel capitolo 1, l'ultimo autovalore della Laplaciana vale 2) la 2.8 diventa

$$\mu_n \approx \pi + \lambda^n \tilde{v}' a'. \quad (2.9)$$

Quando n cresce, infatti, λ_i^n tende a zero per $\lambda_i \neq 1$, e così i modi relativi a tali autovalori si “spengono” velocemente. Per n grande ma non infinito, il modo relativo al secondo autovalore più grande (in modulo), indicato con l'apice nell'equazione 2.9, controlla la velocità con cui la soluzione tende alla stazionarietà (la corretta relazione di proporzionalità dipende dalla nozione di convergenza che si sceglie). Utilizzando la stessa numerazione adottata per gli autovalori della Laplaciana nel capitolo 1, avremmo che

$$\lambda' = \begin{cases} \lambda_1 & \text{se } |\lambda_1| > |\lambda_{N-1}| \\ \lambda_{N-1} & \text{altrimenti} \end{cases},$$

tuttavia si può dimostrare che è sempre possibile migliorare la stima del cosiddetto *tempo caratteristico di rilassamento* τ in funzione esclusivamente di λ_1 [1, 6]. Il secondo autovalore λ_1 viene detto anche *connettività algebrica*, o autovalore di Fiedler, e la sua grandezza riflette quanto un grafo sia ben collegato: risulta chiara la relazione con il tempo di rilassamento, poiché più il grafo è ben collegato, più in fretta riusciranno i *walkers* a muoversi e raggiungere le concentrazioni all'equilibrio. Con la semplice nozione di distanza in L_2 (Euclidea) si ha che $\tau = \frac{1}{1-\lambda_1}$. Si fa presente che τ non è in realtà un “tempo” vero e proprio ma rappresenta il numero di step temporali (dato che stiamo considerando un processo discreto) necessari affinché la distribuzione sia prossima alla soluzione stazionaria (secondo la nozione di convergenza scelta).

2.2.1 Simulazioni di convergenza alla soluzione stazionaria

Per verificare la correttezza di quanto affermato precedentemente sono state effettuate delle simulazioni di random walk su tre diversi tipi di grafi [9]: un grafo random costruito secondo il modello Erdős-Rényi, uno di tipo *small world* ottenuto con il modello Watts-Strogatz e infine un grafo *scale free* costruito con l'algoritmo Barabási-Albert (figure 2.2–2.4). I grafi sono stati generati in modo pseudo-casuale utilizzando la libreria Python *networkx* [10].

Grafo random

Il grafo random (fig. 2.2) è stato generato a partire da 50 nodi, con una probabilità di esistenza di ciascun collegamento pari a $p = 0.2$. Ogni collegamento ha un peso unitario, per semplicità. Il grafo ha un volume pari a $vol G = 524$, mentre la distribuzione dei gradi dei nodi è riassunta nella tabella seguente.

Grado	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Frequenza	1	0	4	2	8	5	5	6	6	6	3	0	3	1

La distribuzione stazionaria combacia con quanto previsto dall'equazione 2.7. Inoltre si ha che il secondo autovalore è $\lambda_1 \cong 0.56$, da cui si ottiene $\tau \cong 2.32$, che è compatibile con i risultati ottenuti dalla simulazione.

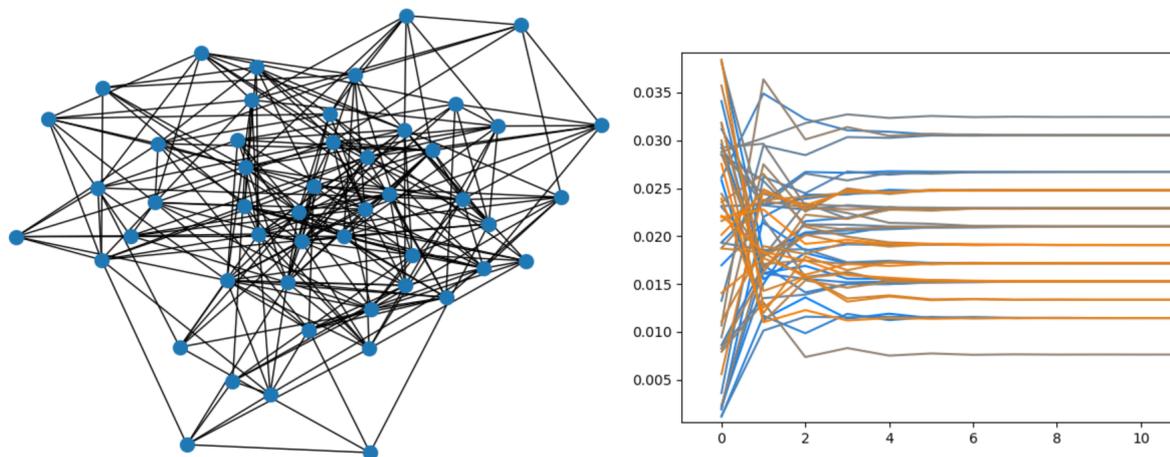


Figura 2.2: A sinistra una rappresentazione del grafo random, a destra l'andamento della distribuzione di probabilità. Ogni linea corrisponde ad uno specifico nodo.

Grafo small world

Tramite il modello Watts-Strogatz è stato generato un grafo small world con 50 nodi e un numero medio di collegamenti per nodo pari a 4 (fig. 2.3). Il volume del grafo è $vol G = 200$ e la distribuzione dei gradi dei nodi è illustrata nella tabella seguente.

Grado	2	3	4	5	6
Frequenza	1	6	36	6	1

Anche in questo caso la distribuzione stazionaria ottenuta dalla simulazione corrisponde al risultato predetto dall'equazione 2.7. Confrontando la tabella con l'andamento della distribuzione degli walker sui nodi, infatti, si nota che gran parte dei nodi tende ad un valore di concentrazione pari a $\pi_i \cong 0.02 = \frac{4}{vol G}$, dove 4 è il grado con frequenza maggiore. Per quanto riguarda il tempo caratteristico di rilassamento, si ha che il secondo autovalore è $\lambda_1 \cong 0.95$. Un autovalore così prossimo all'unità comporta un tempo di rilassamento molto elevato, infatti $\tau \cong 20$.

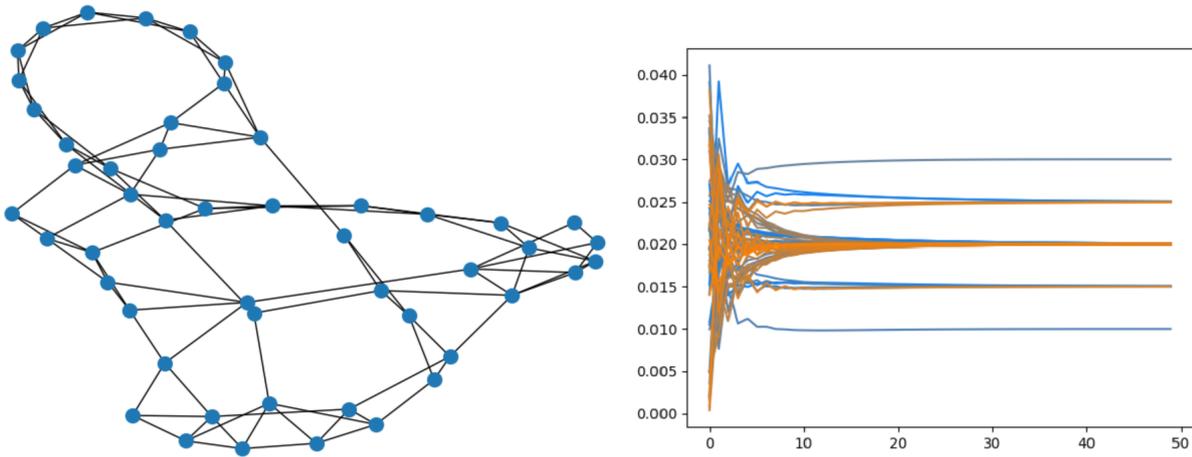


Figura 2.3: A sinistra si ha la rappresentazione del grafo small world, mentre a destra è stato rappresentato l'andamento della concentrazione di walker nei diversi nodi. Si nota subito come la scala temporale con cui la distribuzione tende alla soluzione stazionaria sia decisamente più grande rispetto al caso del random network. Ciò si rispecchia nel valore assunto da λ_1 , molto prossimo all'unità. Dal grafico a destra si nota inoltre come la maggioranza dei nodi abbia lo stesso grado (pari a 4).

Grafo scale free

Il grafo di tipo scale free è stato ottenuto applicando l'algoritmo Barabási-Albert per un grafo di 50 nodi, ottenendo un volume del grafo pari a $vol G = 192$. La distribuzione dei

gradi è riassunta nella tabella seguente.

Grado	1	2	3	4	5	7	8	9	10	11	12	17
Frequenza	1	26	8	5	2	1	1	1	1	2	1	1

Si può notare la caratteristica dei grafi di tipo scale free, per cui la distribuzione dei gradi si concentra a bassi valore di d , come si può notare anche dalla diversa densità di linee nel grafico dell'evoluzione della concentrazione di popolazione nei diversi nodi. Il secondo autovalore è $\lambda_1 \cong 0.80$, da cui $\tau \cong 5$, che risulta compatibile con quanto mostrato in figura 2.4: nonostante siano ancora presenti lievi fluttuazioni nella distribuzione di probabilità, è evidente come la concentrazione di popolazione di ogni nodo tenda al limite stazionario.

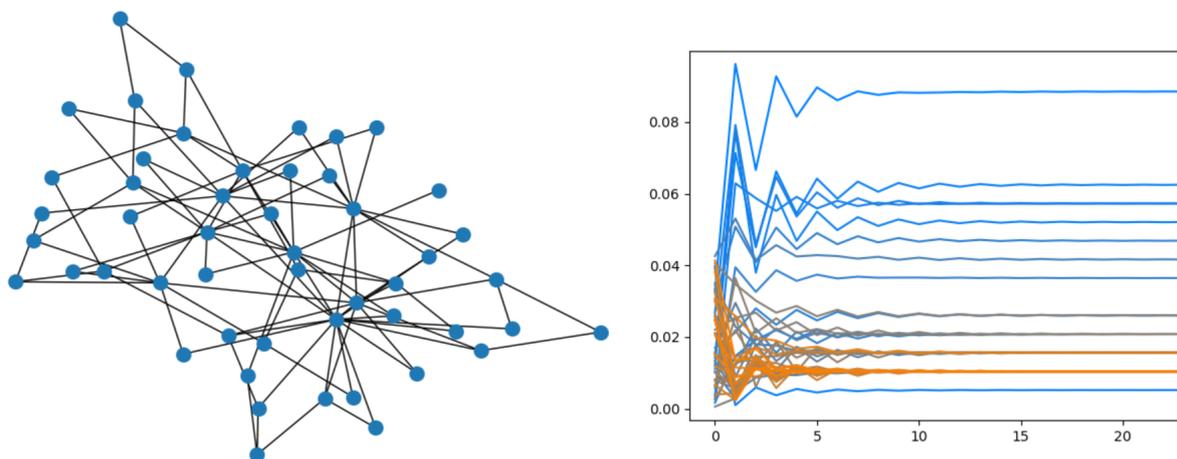


Figura 2.4: Rappresentazione del grafo e andamento della concentrazione di popolazione nei nodi per un network di tipo scale free.

2.2.2 Simulazioni sull'unicità della soluzione stazionaria

Dopo aver confermato l'esistenza, si procede con il verificare l'unicità di tali soluzioni stazionarie. Il teorema 2.2 richiede soltanto che il random walk sia irriducibile, il che, come già detto, equivale ad avere un grafo direzionato fortemente connesso. Le simulazioni sono state effettuate generando in modo casuale un grafo con le caratteristiche desiderate e facendo evolvere una distribuzione di popolazione (anch'essa generata in modo random) secondo la matrice stocastica P fino al raggiungimento della soluzione stazionaria. L'equilibrio viene quindi perturbato bruscamente con un *kick* casuale che modifica i valori della concentrazione di walkers sui diversi nodi (mantenendo la distribuzione normalizzata ad 1) e poi il sistema viene fatto nuovamente rilassare. Come si

può notare in figura 2.5 lo stato di equilibrio raggiunto dopo la perturbazione è lo stesso di quello iniziale. In altre parole, qualsiasi sia la distribuzione di popolazione iniziale, la configurazione del sistema all'equilibrio è sempre la stessa.

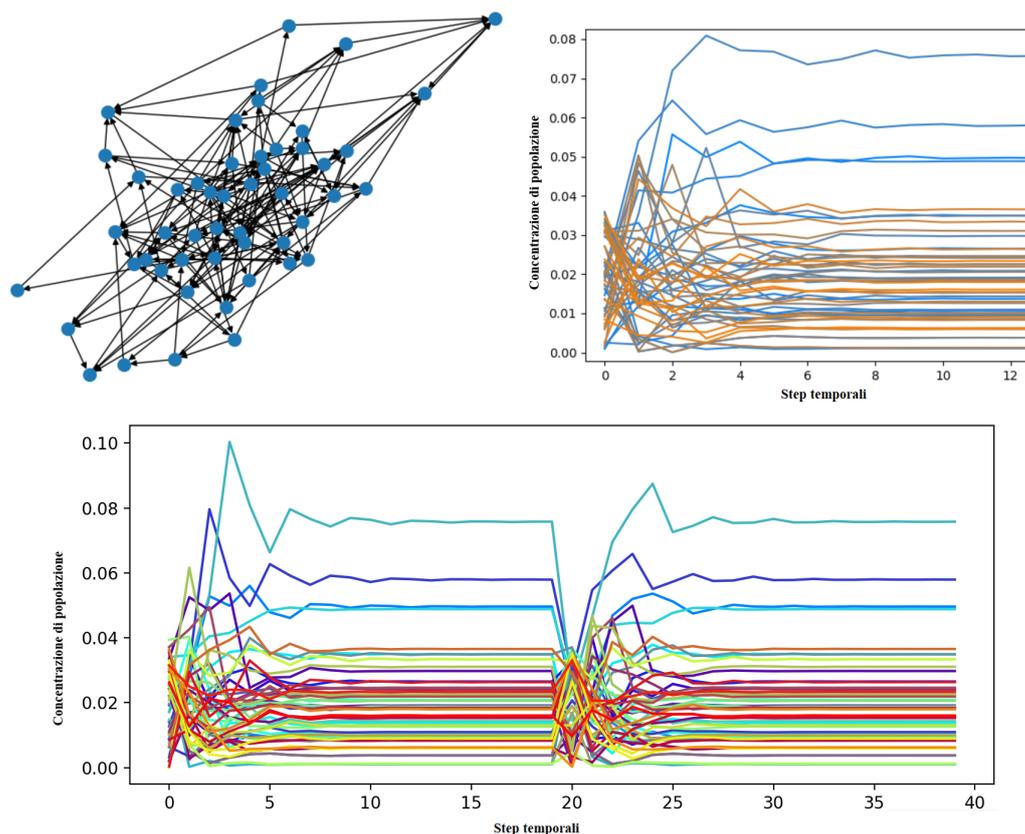


Figura 2.5: In alto a sinistra il grafo direzionato pesato e fortemente connesso, formato da 50 nodi. A fianco, è stata rappresentata l'evoluzione della popolazione verso l'equilibrio, che come è evidente, non si distingue molto da quella di un grafo random non pesato. Nel grafico in basso invece si può notare come dopo aver subito una brusca perturbazione, la concentrazione di popolazione in ciascun nodo ritorna al precedente valore di equilibrio (i colori sono gli stessi).

La condizione richiesta dal teorema 2.2 che il grafo sia fortemente connesso è sufficiente, ma non è in realtà necessaria ed è possibile indebolirla. Se il grafo possiede una sola componente fortemente connessa terminale, la distribuzione di probabilità all'equilibrio sarà unica [4], come si può vedere in figura 2.6.

Si perde l'unicità della soluzione, invece, nel caso in cui le componenti terminali del grafo siano molteplici. La distribuzione di concentrazione di popolazione all'equilibrio dipenderà, infatti, dalle condizioni iniziali, che determineranno la percentuale di popolazione "intrappolata" nei nodi terminali. In figura 2.7 si riporta il risultato di una

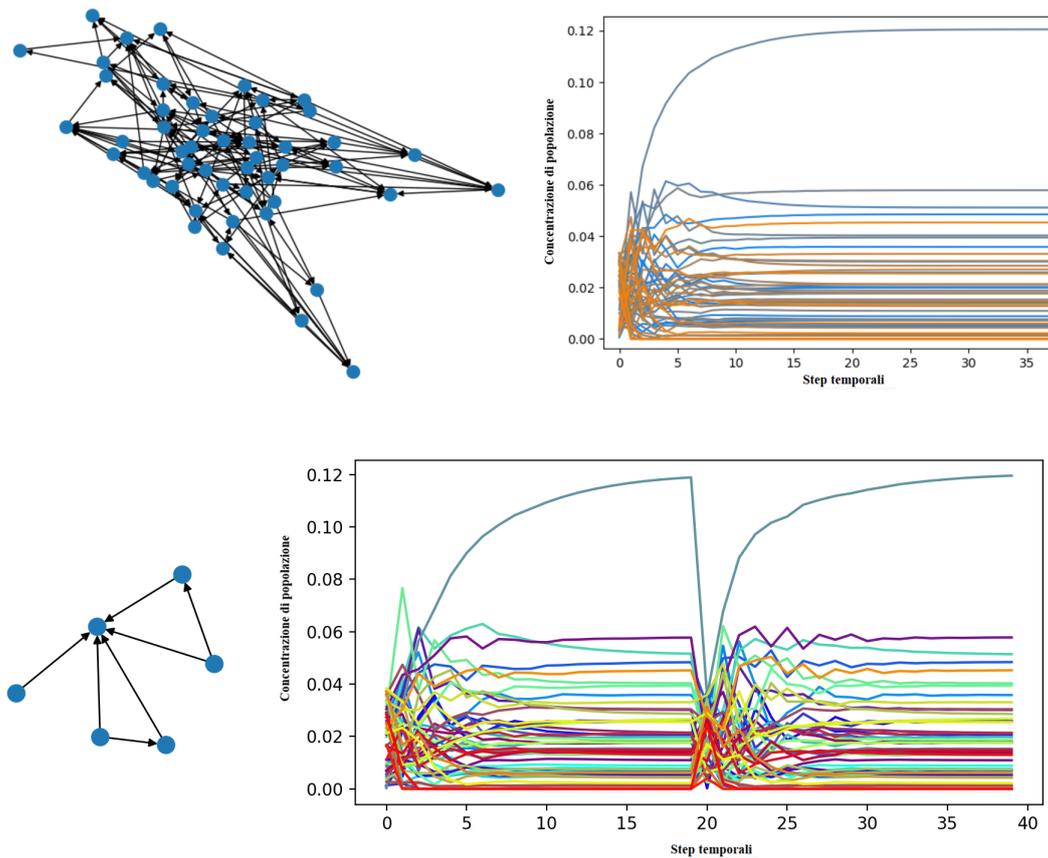


Figura 2.6: In alto sono visibili il grafo (direzionato, non fortemente connesso) e la convergenza alla distribuzione di probabilità stazionaria. In basso a sinistra è stata rappresentata la condensazione del grafo: ogni nodo ora rappresenta una componente fortemente connessa e si può vedere chiaramente come il grafo possieda una sola SCC terminale (quella al centro, che ha solo collegamenti in ingresso ma non in uscita). A causa di questa struttura, i nodi delle componenti non terminali (detti nodi transienti) si svuotano progressivamente, e gli unici a rimanere popolati sono quelli della SCC terminale. In seguito al *kick* che perturba l'equilibrio, la popolazione sui nodi si riassetta nella stessa soluzione precedente, come evidenziato dai colori.

simulazione effettuata con un grafo con 3 SCC terminali, ciascuna di esse composta esclusivamente da un unico nodo.

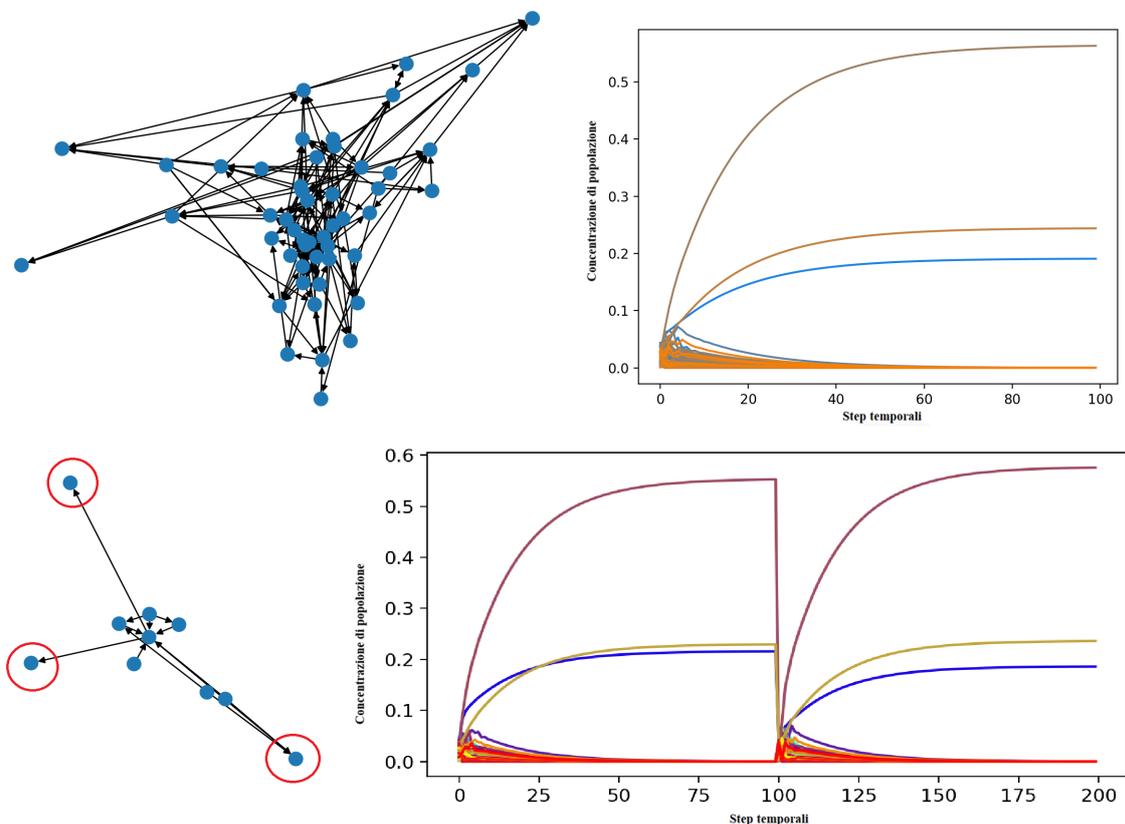


Figura 2.7: In alto la rappresentazione del grafo e la convergenza ad una particolare configurazione di equilibrio. In basso la condensazione del grafo, con cerchiati in rosso le tre componenti terminali. Il fatto che ciascuna delle SCC terminali sia composta da un unico nodo si rispecchia nel grafico delle concentrazioni all'equilibrio: la popolazione di tutti i nodi crolla a zero, con l'eccezione dei tre nodi terminali che rimangono popolati e assorbono la popolazione proveniente dal resto del grafo. In basso a destra si può osservare come la distribuzione stazionaria non sia unica: dopo la perturbazione la popolazione dei nodi si ristabilizza a valori diversi rispetto ai precedenti.

Capitolo 3

Perturbazioni strutturali del network

Dopo aver introdotto il tipo di dinamica su network che ci interessava analizzare e dopo aver studiato le condizioni sulla struttura del grafo affinché le soluzioni stazionarie siano stabili o instabili, ci preoccupiamo di vedere cosa accadrebbe se, invece di perturbare la distribuzione definita sul grafo G , perturbassimo il grafo stesso. Questa perturbazione viene detta strutturale, dato che agisce sulla struttura stessa del grafo G , modificandone i vertici o i collegamenti. Una perturbazione strutturale che coinvolge i vertici deve necessariamente agire aumentando o diminuendo il numero dei nodi del grafo, mentre una perturbazione che interessa i collegamenti può consistere nella leggera variazione dei pesi dei collegamenti oppure in modo più drastico, aggiungendo o rimuovendo del tutto alcuni link. Considereremo soltanto perturbazioni strutturali non drastiche che coinvolgono esclusivamente i link. Ci poniamo inoltre l'obiettivo di creare un sistema di controllo capace di agire sul network in modo da neutralizzare la perturbazione in arrivo e riportare il grafo nella situazione di equilibrio precedente.

3.1 Perturbazione non direzionata su grafo non direzionato

Si consideri inizialmente il caso più semplice, ovvero una perturbazione non drastica che colpisce un grafo non direzionato, mantenendo la simmetria della matrice di adiacenza del grafo. Una perturbazione di questo tipo può essere descritta dalla seguente matrice

$$\epsilon_{ij} = \begin{cases} \epsilon_{ij} & \text{se } v_i \sim v_j \\ 0 & \text{altrimenti,} \end{cases}$$

con la condizione che $\epsilon_{ij} = \epsilon_{ji}$. Si ricorda che $v_i \sim v_j$ indica la relazione di adiacenza tra una coppia di vertici del grafo. La perturbazione agisce sulla matrice di adiacenza, trasformandola in questo modo:

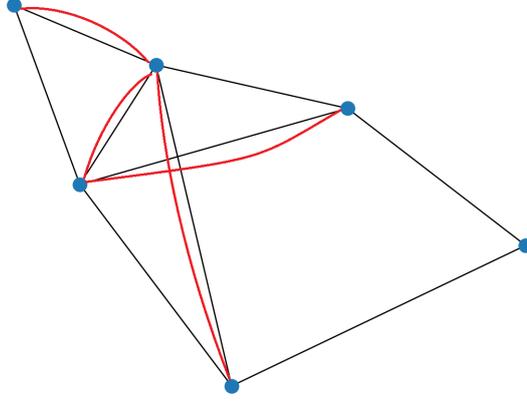


Figura 3.1: In figura si può vedere un grafo non direzionato (nodi in blu, collegamenti in nero), sottoposto ad una perturbazione strutturale non drastica (in rosso) anch'essa non direzionata. Come si può notare, la perturbazione agisce solo su collegamenti preesistenti, senza crearne di nuovi.

$$A_{ij} \rightarrow A'_{ij} = \begin{cases} w_{ij} + \epsilon_{ij} & \text{se } v_i \sim v_j \\ 0 & \text{altrimenti.} \end{cases}$$

Ricordando la distribuzione stazionaria vista nel capitolo 2 nel caso di un grafo non direzionato

$$\pi_i = \frac{d_i}{\sum_{l=1}^N d_l}, \quad (3.1)$$

e considerando la trasformazione a cui sono sottoposti i gradi dei nodi del network e il volume totale

$$d_i \rightarrow d'_i = \sum_{j=1}^N (w_{ij} + \epsilon_{ij}) \equiv d_i + \epsilon_i,$$

$$vol G \rightarrow vol G' = \sum_{j=1}^N d'_j = \sum_{j=1}^N (d_j + \epsilon_j) \equiv vol G + vol \epsilon,$$

otteniamo che l'equazione 3.1 diventa

$$\pi_i \rightarrow \pi'_i = \frac{d'_i}{vol G'} = \frac{d_i + \epsilon_i}{vol G + vol \epsilon}. \quad (3.2)$$

L'equazione 3.2 rappresenta la nuova distribuzione di popolazione all'equilibrio. Si ipotizzi ora di avere un grafo G completamente noto (ne conosciamo la matrice di adiacenza, dunque ne conosciamo i gradi, il volume e la distribuzione stazionaria) in cui in

un certo istante arrivi una perturbazione incognita che ne modifichi la matrice di adiacenza in modo a noi sconosciuto. Si ipotizzi che della perturbazione sia possibile tuttavia misurare l'effetto che ha sulla soluzione stazionaria $\pi \rightarrow \pi'$, in altre parole, si conoscono tutti i $\Delta_i \equiv \pi'_i - \pi_i \forall i \in 1, \dots, N$. L'obiettivo che ci si pone è di ricostruire la forma della perturbazione in arrivo.

Dalla 3.1 e dalla 3.2 otteniamo un'espressione che lega Δ_i alla perturbazione:

$$\Delta_i \equiv \pi'_i - \pi_i = \frac{\epsilon_i \text{vol } G - d_i \text{vol } \epsilon}{\text{vol } G (\text{vol } G + \text{vol } \epsilon)} \quad (3.3)$$

che può essere riscritta come un sistema lineare in ϵ_i

$$\sum_{j=1}^N [\epsilon_j (\delta_{ij} \text{vol } G - d_i - \Delta_i)] = \Delta_i (\text{vol } G)^2 \quad (3.4)$$

con $i = 1, 2, \dots, N$.

Si può dimostrare che il determinante di questo sistema (la cui matrice associata verrà chiamata M) dipende da

$$\det(M) \propto (\text{vol } G)^N \sum \Delta_i. \quad (3.5)$$

Tuttavia, considerando la definizione di Δ_i in funzione di π'_i e di π_i , che sono entrambe delle distribuzioni, e dunque normalizzate a uno, si ottiene che

$$\sum_{i=1}^N \Delta_i = \sum_{i=1}^N \pi'_i - \sum_{i=1}^N \pi_i = 1 - 1 = 0. \quad (3.6)$$

Perciò il determinante del sistema è nullo e non è possibile ricostruire analiticamente in modo univoco la perturbazione.

3.2 Sviluppo di un sistema di controllo

Data l'impossibilità di risolvere univocamente in maniera analitica il sistema 3.4, ci proponiamo di ricostruire numericamente la perturbazione attraverso un algoritmo di ottimizzazione. Il sistema di controllo progettato avrà l'obiettivo di ristabilire l'equilibrio in cui il network si trovava prima dell'arrivo della perturbazione.

Il sistema si basa sull'algoritmo "adadelta" [11], un algoritmo di *adaptive learning* che attua la cosiddetta discesa del gradiente per minimizzare una funzione (la funzione di costo, che quantifica la "distanza" tra la soluzione numerica trovata e la soluzione ideale). L'algoritmo è stato modificato in modo da includere tecniche di *differential evolution*, che permettono di esplorare lo spazio delle soluzioni anche in maniera euristica. Ciò

rende la convergenza più efficiente e riduce la probabilità di rimanere intrappolati in un minimo locale.

Anche in questo caso, chiaramente, la soluzione trovata non sarà univoca, ma lo scopo del sistema di controllo sarà quello di riportare il network ad una concentrazione stazionaria il più possibile simile a quella iniziale nel modo più efficiente. Si noti come non sia stata posta alcuna condizione sulla reale perturbazione che colpisce il network, se non che $w_{ij} + \epsilon_{ij} > 0$, per mantenere positiva la matrice di adiacenza. Il sistema di controllo ci permette, invece, di scegliere dei limiti entro cui far variare la soluzione, così che si potrà decidere di restringere lo spazio di ricerca delle soluzioni a quelle che permettono di tornare all'equilibrio di partenza con il "minor sforzo possibile" (dal punto di vista applicativo, ciò potrebbe essere utile nel caso in cui il network in questione rappresenti un sistema non facilmente controllabile, in cui creare delle modifiche notevoli ai pesi dei collegamenti risulta molto dispendioso).

Si tratterà il caso in cui il network abbia un tempo di rilassamento molto breve, perciò il sistema di controllo sarà progettato per agire direttamente sul nuovo stato di equilibrio π' e non su stati transienti.

3.2.1 Funzione di costo

Un elemento essenziale per l'algoritmo è la funzione di costo, che è stata definita nel modo seguente

$$F(\epsilon_{ij}, \text{ con } i, j \in \{1, \dots, N\}) = \sum_{i=1}^N \left(\frac{(\sum_{j=1}^N \epsilon_{ij}) \text{vol } G - d_i \text{vol } \epsilon}{\text{vol } G (\text{vol } G + \text{vol } \epsilon)} - \Delta_i \right)^2. \quad (3.7)$$

Si può notare innanzitutto come le variabili della funzione, in questo caso, siano direttamente gli elementi della matrice che definisce la perturbazione, poiché, infatti, il metodo numerico ci permette di ottenere in un unico procedimento direttamente gli elementi ϵ_{ij} , senza dover passare attraverso la loro somma lungo le righe, cioè ϵ_i . In secondo luogo notiamo che la funzione è sempre positiva o nulla, e in particolare che $F = 0$ se e solo se l'insieme delle variabili ϵ_{ij} è tale da causare una differenza di Δ_i nello stato stazionario (si può riconoscere che il numeratore della frazione nella sommatoria è esattamente l'equazione 3.3). Perciò i punti di minimo globale di F corrispondono alle soluzioni che si vogliono trovare. Il fattore Δ_i al denominatore, infine, è utile per velocizzare l'algoritmo, poiché dividendo per $\Delta_i \ll 1$ si ottiene il doppio vantaggio di lavorare con numeri non eccessivamente piccoli e di rendere più ripidi i cambiamenti di gradiente.

3.2.2 Risultati ottenuti

Il sistema è stato testato inizialmente su un piccolo network di soli 3 nodi completamente connesso, la cui matrice di adiacenza imperturbata è

$$A = \begin{bmatrix} 0 & 10 & 2 \\ 10 & 0 & 4 \\ 2 & 4 & 0 \end{bmatrix}$$

mentre la perturbazione subita è

$$\epsilon = \begin{bmatrix} 0 & 1.5 & 2.3 \\ 1.5 & 0 & 0 \\ 2.3 & 0 & 0 \end{bmatrix}.$$

Il sistema di controllo ha ricostruito una perturbazione con la seguente forma:

$$\epsilon' = \begin{bmatrix} 0 & -0.064 & 1.715 \\ -0.64 & 0 & -0.544 \\ 1.715 & -0.544 & 0 \end{bmatrix},$$

chiaramente diversa dalla perturbazione effettivamente subita (anche per il fatto che è stato imposto al sistema di controllo $-2 < \epsilon_{ij} < 2$, perciò non avrebbe mai potuto ottenere il valore di 2.3). Tuttavia dalla figura 3.2 si può notare come la concentrazione di popolazione all'equilibrio venga ricostruita in modo molto soddisfacente. La distribuzione stazionaria sui 3 nodi prima dell'arrivo di ϵ era $\pi = (0.375, 0.4375, 0.1875)$ (ottenibile anche analiticamente dall'equazione 3.1), mentre quella ricostruita è $\pi_{finale} = (0.378, 0.4312, 0.1904)$.

Procedendo a testare il sistema di controllo con network più grandi, continuiamo ad ottenere ottimi risultati. In figura 3.3 è possibile vedere l'andamento della distribuzione nel caso di un grafo con 6 nodi, con le seguenti matrici di adiacenza e della perturbazione:

$$A = \begin{bmatrix} 0 & 12 & 2 & 6 & 5 & 0 \\ 12 & 0 & 0 & 5 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 1 \\ 6 & 5 & 2 & 0 & 3 & 0 \\ 5 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad \epsilon = \begin{bmatrix} 0 & 0.3 & 1 & -0.8 & -1.25 & 0 \\ 0.3 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -0.1 & 0 & 0 \\ -0.8 & 0 & -0.1 & 0 & -0.2 & 0 \\ -1.25 & 0 & 0 & -0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

La soluzione ottenuta è

$$\epsilon' = \begin{bmatrix} 0 & 0.24 & 0.71 & -0.24 & 0.61 & 0 \\ 0.24 & 0 & 0 & 1.49 & 0 & 0 \\ 0.71 & 0 & 0 & 0.68 & 0 & 0.09 \\ -0.24 & 1.49 & 0.68 & 0 & -1.64 & 0 \\ 0.61 & 0 & 0 & -1.64 & 0 & 0 \\ 0 & 0 & 0.09 & 0 & 0 & 0 \end{bmatrix},$$

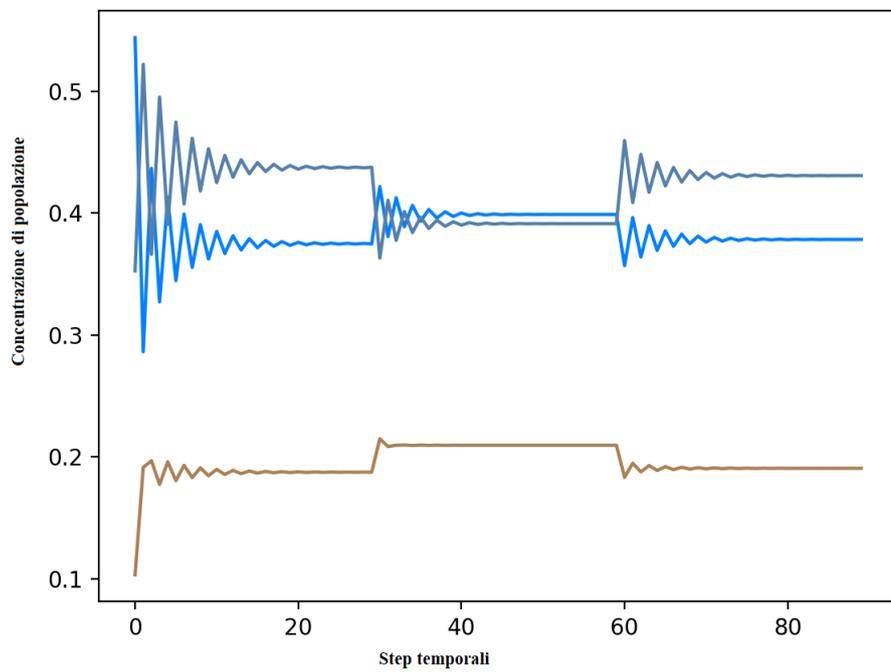


Figura 3.2: Andamento della distribuzione sui diversi nodi in funzione del tempo. Dopo l'arrivo della perturbazione, allo step temporale $t = 30$, il sistema rilassa in un nuovo equilibrio. Applicando, in $t = 60$, la perturbazione ricostruita, il sistema ritorna all'equilibrio originale.

mentre le distribuzioni di popolazioni sono estremamente compatibili:

$$\pi = (0.347, 0.236, 0.069, 0.222, 0.111, 0.014),$$

$$\pi_{finale} = (0.347, 0.236, 0.067, 0.221, 0.115, 0.014).$$

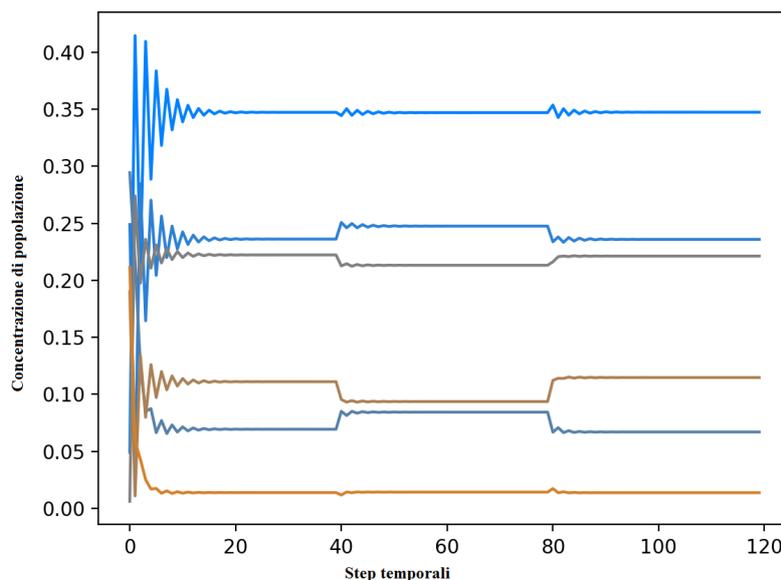


Figura 3.3: Andamento della concentrazione di popolazione in funzione del tempo in un grafo con 6 nodi. In $t = 40$ sopraggiunge una perturbazione strutturale, mentre in $t = 80$ viene applicata la perturbazione complementare ottenuta dall'algoritmo. L'equilibrio originale viene ricostruito in modo soddisfacente.

Infine, nella figura 3.4 il sistema di controllo è stato testato con un grafo random di tipo Erdős-Rényi con 20 nodi (il grafo random, come visto nel capitolo 2, è un ottimo candidato come network con un tempo di rilassamento veloce). Anche in questo caso il sistema di controllo è stato in grado di trovare una perturbazione complementare alla perturbazione effettiva, capace di riportare l'equilibrio originale nei nodi del grafo. In questa simulazione, è stato imposto il vincolo ancora più stringente di $-0.5 < \epsilon_{ij} < 0.5$.

3.3 Estensione al caso direzionato

Si vuole ora provare ad estendere il sistema di controllo sviluppato nel caso non direzionato al più generico caso di un grafo direzionato sottoposto ad una perturbazione strutturale (anch'essa generica). Due sono i problemi che sorgono. Il primo, più di natura computazionale, è che la perdita di simmetria raddoppia la dimensionalità dello spazio delle soluzioni, provocando inevitabilmente una perdita di precisione e un aumento del

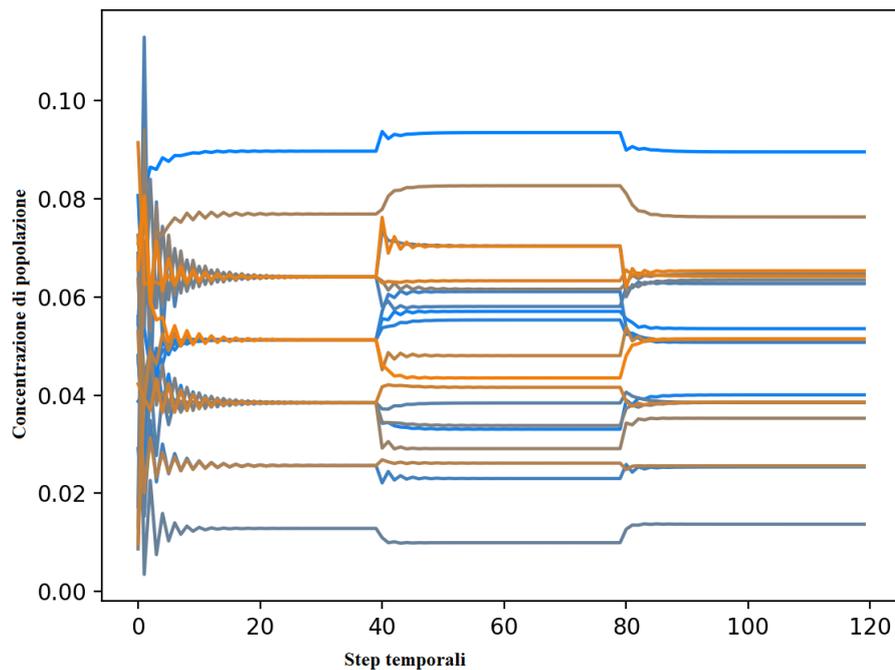


Figura 3.4: Andamento della distribuzione sui diversi nodi in funzione del tempo. In $t = 40$ sopraggiunge una perturbazione strutturale, mentre in $t = 80$ viene applicata la perturbazione complementare ottenuta dall'algoritmo. Si potrebbe credere che in questo caso il sistema di controllo sia stato meno efficace rispetto ai casi precedenti con 3 e 6 nodi, ma in realtà le concentrazioni di popolazione iniziale e finale sono uguali tra di loro fino alla seconda/terza cifra decimale, cioè un risultato migliore rispetto a quello ottenuto nel caso di un grafo con 3 nodi (e con un numero di iterazioni dell'algoritmo addirittura minore). L'apparente peggiore precisione è dovuta al fatto che l'asse verticale ha una scala più ingrandita.

tempo necessario per trovare la soluzione. Il secondo problema, invece, è che non si possiede un corrispettivo direzionato dell'equazione 3.1 che descrive in modo esatto la distribuzione all'equilibrio e perciò la precedente funzione di costo, che su di essa si basa, risulta inefficace.

Occorre dunque scegliere una nuova funzione di costo adatta alla situazione, e la scelta cade sulla seguente funzione:

$$F(\epsilon_{ij}, \text{ con } i, j \in \{1, \dots, N\}) = \sum_{l=1}^N \left((\Re(v_l) - \pi'_l)^2 + k \Im(v_l)^2 \right), \quad (3.8)$$

dove π' è la distribuzione stazionaria dopo la perturbazione (misurabile dal nostro sistema di controllo), v_l è l'autovettore corrispondente all'autovalore $\lambda = 1$ della matrice stocastica P' (con $\Re(v)$ e $\Im(v)$ rispettivamente la parte reale ed immaginaria) e k una costante positiva $k \gg 1$. Nella funzione 3.8 la dipendenza dalla perturbazione ϵ_{ij} è in $v_l = v_l(\epsilon_{ij})$, essendo esso un autovalore della matrice $P' = P'(\epsilon_{ij}) = T'^{-1}(\epsilon_{ij})A'(\epsilon_{ij})$. La forma della funzione è dovuta al fatto che l'autovettore $vP' = v$, che rappresenta la distribuzione stazionaria, deve essere un vettore reale. Il minimo di F infatti si ha per $\Re(v) = \pi'$ e $\Im(v) = 0$.

Data una possibile forma per la perturbazione ϵ_{ij} , l'autovettore v viene calcolato numericamente cercando tra gli autovalori della matrice più prossimi a 1, ma poichè gli ϵ_{ij} variano nello spazio delle soluzioni, essi non sono necessariamente tali per cui l'algoritmo numerico converga. In tal caso interviene la componente euristica dell'algoritmo, che trascura la soluzione non convergente e fa proseguire l'algoritmo di discesa del gradiente generando nuove candidate soluzioni.

Il nuovo sistema di controllo è stato testato con delle simulazioni su grafi di 6 e 10 nodi ciascuno (figure 3.5 e 3.6). Nel caso del grafo con 6 nodi, con la seguente matrice di adiacenza e con la seguente perturbazione

$$A = \begin{bmatrix} 0 & 3.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.3 & 0 & 16.5 & 0 \\ 2.7 & 11.9 & 0 & 4.4 & 0 & 0 \\ 0 & 19.7 & 0 & 0 & 0 & 10.9 \\ 0 & 17.7 & 0 & 0 & 0 & 0 \\ 17.8 & 19.6 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \epsilon = \begin{bmatrix} 0 & 0.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.2 & 0 & -1.7 & 0 \\ 0.3 & -4.1 & 0 & -2.1 & 0 & 0 \\ 0 & 1.9 & 0 & 0 & 0 & -2.3 \\ 0 & -6.1 & 0 & 0 & 0 & 0 \\ 3.1 & 2.0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

la soluzione ottenuta è

$$\epsilon' = \begin{bmatrix} 0 & -1.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.2 & 0 & 3.7 & 0 \\ 0.4 & -1.8 & 0 & -0.9 & 0 & 0 \\ 0 & -1.9 & 0 & 0 & 0 & 2.1 \\ 0 & -2.8 & 0 & 0 & 0 & 0 \\ 0.6 & 1.6 & 0 & 0 & 0 & 0 \end{bmatrix},$$

e le distribuzioni stazionarie prima della perturbazione e dopo la correzione risultano compatibili:

$$\pi = (0.015, 0.480, 0.081, 0.019, 0.399, 0.007),$$

$$\pi_{finale} = (0.016, 0.481, 0.083, 0.017, 0.399, 0.004).$$

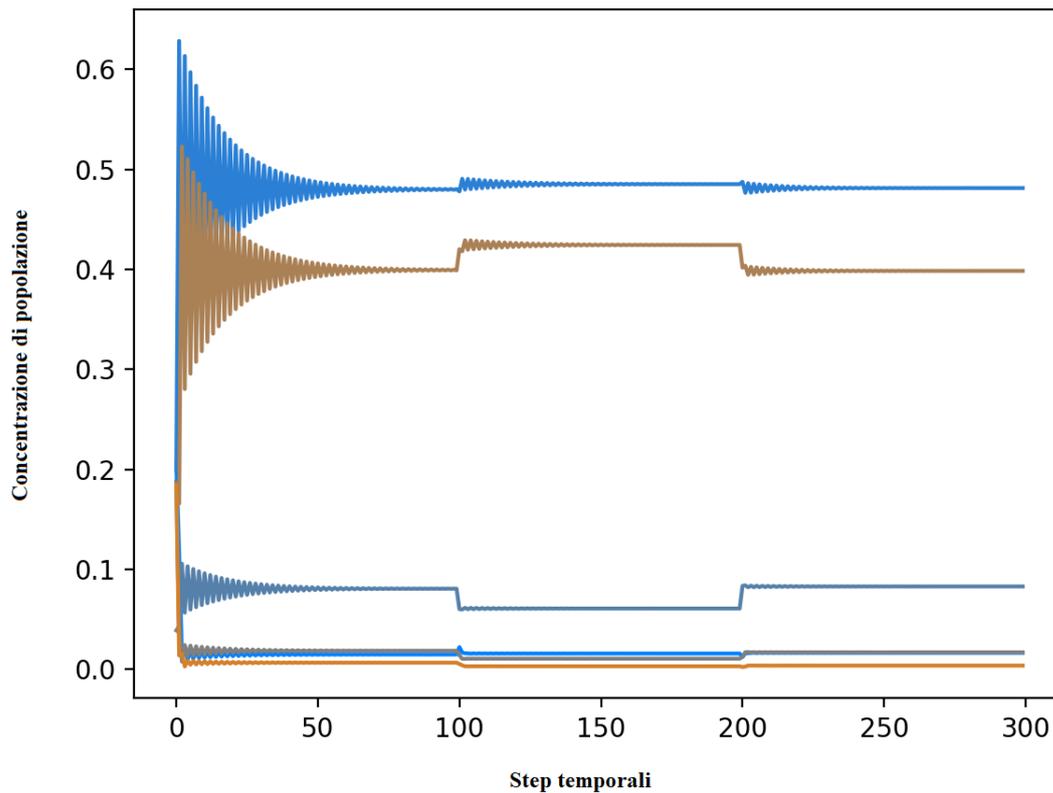


Figura 3.5: Andamento della concentrazione di popolazione in funzione del tempo in un grafo direzionato con 6 nodi. In $t = 100$ sopraggiunge una perturbazione strutturale, mentre in $t = 200$ viene applicata la perturbazione complementare ottenuta dall’algoritmo. L’equilibrio originale viene ricostruito in modo soddisfacente.

Nel caso del grafo con 10 nodi la distribuzione stazionaria originale viene ricostruita solo parzialmente, e anche se la maggior parte dei nodi torna alla concentrazione di *walkers* iniziali, ci sono alcune eccezioni per cui la differenza tra π e la distribuzione stazionaria ricostruita π_{finale} non è trascurabile. Inoltre testando il sistema di controllo con network più grandi di 10/15 nodi, si nota che il programma fallisce nel ricostruire l’equilibrio originale, riuscendo ad annullare l’effetto della perturbazione solo per una frazione di nodi.

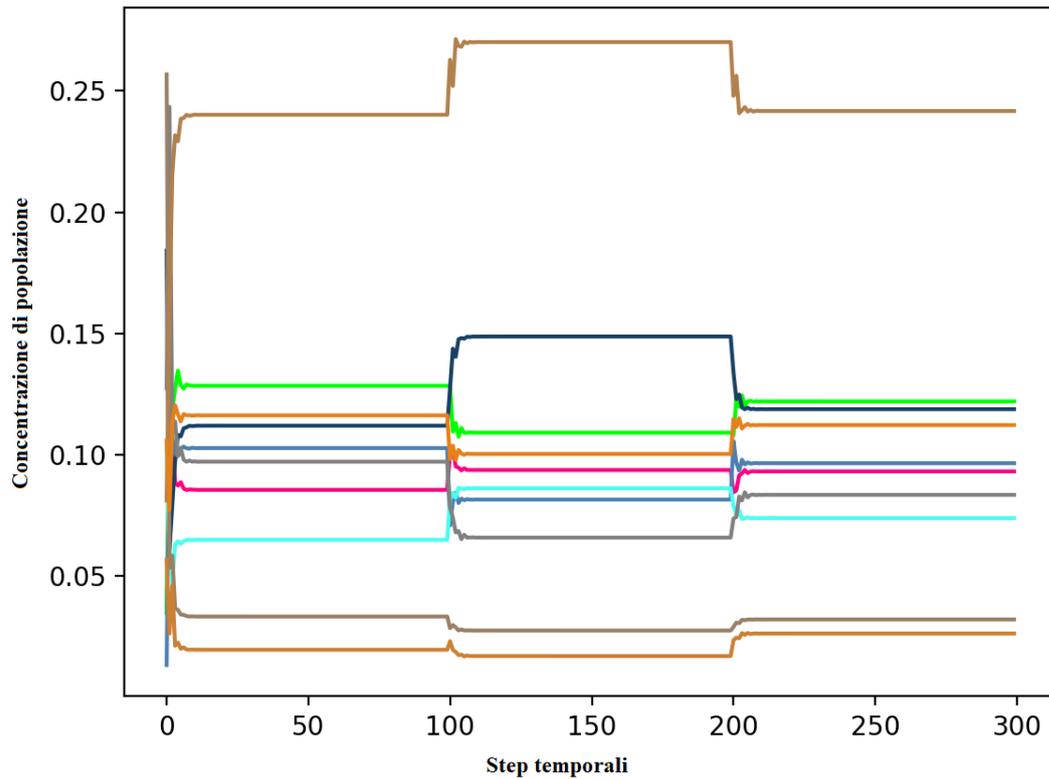


Figura 3.6: Andamento della concentrazione di popolazione in funzione del tempo in un grafo direzionato con 10 nodi. L'equilibrio originale viene ricostruito soltanto parzialmente. Si potrebbe pensare che con un numero maggiore di iterazioni dell'algoritmo, la convergenza migliorerebbe. Tuttavia, l'obiettivo del sistema di controllo è quello di ricostruire l'equilibrio originale in modo rapido, così da essere capace di rispondere quasi immediatamente a perturbazioni esterne, e perciò aumentare eccessivamente il numero di iterazioni finirebbe per snaturare lo scopo del sistema di controllo.

Conclusione e discussione

L'analisi svolta ha confermato sperimentalmente attraverso diverse simulazioni tutti i risultati teorici presentati.

Nel caso di grafi non direzionati si è ottenuto che la distribuzione stazionaria in un nodo è pari al suo grado, normalizzato rispetto al volume totale del network, confermando la previsione teorica. Tale distribuzione è unica e indipendente dalle condizioni iniziali e, inoltre, il tempo di rilassamento con cui il sistema tende ad essa dipende dalla connettività algebrica λ_1 (il secondo autovalore più piccolo della matrice Laplaciana normalizzata).

L'esistenza ed unicità della soluzione stazionaria rimangono ancora valide per i grafi direzionati in quasi tutti i casi: è condizione necessaria e sufficiente che il grafo abbia una ed una sola componente fortemente connessa terminale. È immediato interpretare questo risultato ottenuto nella teoria osservando l'andamento della concentrazione nei diversi nodi, ottenuto dalle simulazioni. I nodi che non appartengono alla SCC terminale si svuotano progressivamente, mentre quelli che le appartengono si riempiono. Il grafo, una volta raggiunto lo stato di equilibrio, è popolato soltanto nella componente terminale, che così funge da "sottografo" fortemente connesso, mentre gli altri nodi sono vuoti. Se ci fossero più componenti connesse terminali, lo stato stazionario finale dipenderebbe dalle condizioni iniziali: distribuzioni di partenza diverse provocherebbero una suddivisione diversa di *walkers* nelle varie SCC terminali, e quindi distribuzioni stazionarie differenti.

Infine, per quanto riguarda le perturbazioni strutturali, il programma di controllo implementato per reagire ad esse e ristabilire l'equilibrio precedente agisce correttamente per grafi non direzionati e per grafi direzionati con pochi nodi, ma fallisce per un generico grafo direzionato con molti vertici. Le sue limitazioni risiedono parzialmente nella funzione di costo, che potrebbe essere migliorata includendo altri parametri (come ad esempio il diametro del grafo o la sua connettività algebrica), ma parzialmente anche nella limitata capacità di calcolo a disposizione che inficia le performance delle simulazioni.

Bibliografia

- [1] Fan Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.
- [2] Fan Chung. Laplacians and the cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19, 2005.
- [3] JJP Veerman and Robert Lyons. A primer on laplacian dynamics in directed graphs. *arXiv preprint arXiv:2002.02605*, 2020.
- [4] JJP Veerman and Ewan Kummel. Diffusion and consensus on weakly connected directed graphs. *Linear Algebra and its Applications*, 578:184–206, 2019.
- [5] Inomzhon Mirzaev and Jeremy Gunawardena. Laplacian dynamics on general graphs. *Bulletin of mathematical biology*, 75(11):2118–2149, 2013.
- [6] David Aldous and James Allen Fill. Reversible markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014, available at <http://www.stat.berkeley.edu/~aldous/RWG/book.html>.
- [7] James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998.
- [8] Naoki Masuda, Mason A. Porter, and Renaud Lambiotte. Random walks and diffusion on networks. *Physics Reports*, 716-717:1–58, Nov 2017.
- [9] Mark Newman. Networks: An introduction. 2010: Oxford university press. *Artif. Life*, 18:241–2, 2012.
- [10] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
- [11] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017.