Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Matematica

Hub persistenti in reti

Tesi di Laurea Magistrale in Topologia Algebrica

Relatore:
Prof.
Massimo Ferri

Correlatore:
Dott.
Mattia G. Bergomi

Candidato:
Antonella Tavaglione

II Sessione
Anno Accademico 2017/2018

Indice

Elenco delle figure

Elenco delle tabelle

Lista degli algoritmi

Introduzione

Al giorno d'oggi lavoriamo con quantità di dati impensabili fino a pochi anni fa. La rappresentazione di fenomeni complessi, come l'elaborazione ad alto livello di dati grezzi, fanno parte di algoritmi utilizzati quotidianamente da chiunque possegga uno smartphone. La comprensione e di conseguenza la riduzione della dimensionalità di tali problemi è dunque un oggetto di grande interesse.
È possibile trovare i centri nevralgici di dati rappresentati sotto forma di un grafo pesato?

In questo elaborato affrontiamo il problema usando una recentissima tecnica matematica. Senza ricorrere a mediazioni topologiche descriveremo come sia possibile studiare i grafi pesati nel dominio della persistenza. In [2] viene proposto e implementato un metodo per slegare la persistenza dall'omologia persistente, fornendola di una teoria propria e generale che non ricorre a spazi topologici o a complessi simpliciali. Tale teoria viene applicata in questo lavoro a grafi pesati estendendola con la definizione di particolari *funzioni di persistenza*, che sono generalizzazioni delle più note funzioni dei numeri di Betti persistenti. Dopo aver introdotto in maniera intuitiva il concetto di hub, definiamo formalmente le funzioni di persistenza che chiameremo steady e ranging hub. Utilizzeremo queste funzioni per analizzare tre reti. Questi esempi che coprono campi estremamente differenti mirano a dare un'indicazione sulla generalità del metodo proposto.

Nel primo capitolo riportiamo la strategia di soluzione di alcuni ricercatori che tentano di dare una risposta al problema combinando strumenti di base di teoria dei grafi con la persistenza omologica.
Nel secondo capitolo riportiamo nel dettaglio il recente studio che mira a slegare la persistenza dall'omologia persistente e illustriamo un metodo di selezione dei cornerpoint di un diagramma di persistenza.
Nel terzo capitolo sperimentiamo la teoria individuando gli hub di tre reti: la rete dei collegamenti tra aeroporti degli Stati Uniti d'America, la rete delle relazioni tra i personaggi de *Les Misérables* e la rete delle lingue ufficiali dell'Unione Europea.

Capitolo 1

Premesse

Una rete che rappresenti le interazioni di un gruppo di agenti è un oggetto di grande complessità combinatorica. Davanti a un tale oggetto è naturale chiedersi quali dei suoi nodi—agenti—siano più rilevanti rispetto all'organizzazione complessa dell'intero sistema.
In letteratura spesso si usano le *misure di centralità* per dare una risposta a questo problema ([1] [7] [8] [9]).
Introduciamo nel primo capitolo di questo elaborato i concetti di base di teoria dei grafi e persistenza. Inoltre parliamo di un interessante metodo [1] basato sulle misure di centralità che combina tra loro questi strumenti.
Abbiamo scelto di porre l'attenzione su questo metodo non perché sia quello che utilizzeremo nelle nostre applicazioni, ma perché può essere utile per comprendere il diverso ruolo che attribuiremo alla persistenza nel secondo capitolo: qui la persistenza è lo strumento che porta alla definizione di una misura di centralità; nel secondo capitolo, pur non facendo riferimento ad alcuna misura di centralità, usando la persistenza in termini più astratti, saremo in grado di individuare i nodi centrali di una rete.

1.1 Preliminari di teoria dei grafi

Per iniziare, introduciamo alcune definizioni di base in teoria dei grafi che aiutano l'analisi e la comprensione di una rete complessa.

Definizione 1 (Grafo)**.** Un *grafo* G è una coppia ordinata $G = (V, E)$, dove V è l'insieme dei vertici del grafo ed $E = \{\{a, b\} | a, b \in V, a \neq b\}$ è l'insieme dei lati, e rappresenta le relazioni tra i vertici.

Definizione 2 (Sottografo)**.** Un grafo $G' = (V', E')$ è detto *sottografo* di $G = (V, E)$ se $V' \subseteq V$ e $E' \subseteq E$

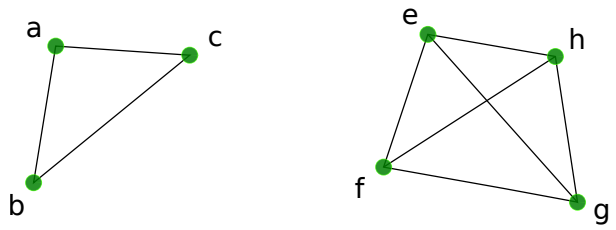

Figura 1.1: A sinistra un esempio di *3*-cricca e a destra un esempio di *4*-cricca.

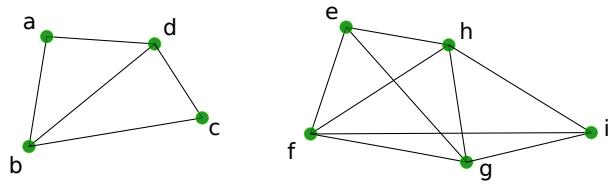

Figura 1.2: Le *3*-cricche $\{a, b, d\}$ e $\{b, c, d\}$ sono adiacenti poiché la loro intersezione è la *2*-cricca $\{b, d\}$. Analogamente, le *4*-cricche $\{e, f, g, h\}$ e $\{f, g, h, i\}$ si intersecano nella *3*-cricca $\{f, g, h\}$ e sono quindi adiacenti

Definizione 3 (Sottografo indotto)**.** Dato un grafo $G = (V, E)$, il grafo $G_W = (W, E_W)$ si dice *sottografo di G indotto da W* se $W \subseteq V$ e $E_W = \{\{j, k\} \in E | j, k \in W\}$.

Definizione 4 (k-cricca)**.** Una *k-cricca* è un sottoinsieme di V di cardinalità k, il cui sottografo indotto è un grafo completo su k vertici (Figura 1.1), cioè ogni vertice della k-cricca è collegato ad ogni altro da un lato.

Definizione 5 (k-cricche adiacenti)**.** Due k-cricche ρ e ρ' di G si dicono *adiacenti* se condividono (*k-1*) vertici (Figura 1.2) o, equivalentemente, se la loro intersezione è una (*k-1*)-cricca.

Definizione 6 (k-cricche connesse)**.** Due k-cricche ρ e ρ' di G si dicono *connesse* (Figura 1.3) se esiste una sequenza $\{\rho_1 = \rho, \ldots, \rho' = \rho_n\}$ tale che ρ_i e ρ_{i+1} sono adiacenti per ogni $i \in \{1, \ldots, n-1\}$

Definizione 7 (comunità di k-cricche)**.** Una *comunità di k-cricche* di G è un'unione massimale di k-cricche connesse a due a due. Figura 1.4

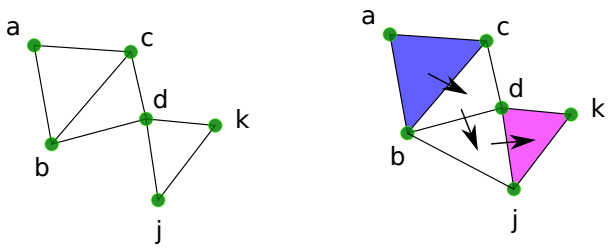

Figura 1.3: Nella figura a sinistra le 3-cricche $\{a, b, c\}$ e $\{d, j, k\}$ non sono connesse, infatti le 3-cricche $\{b, c, d\}$ e $\{d, j, k\}$ non sono adiacenti. Aggiungendo il lato $\{b, j\}$ otteniamo una sequenza di 3-cricche in cui due 3-cricche consecutive sono adiacenti. Pertanto nella figura a destra le 3-cricche $\{a, b, c\}$ e $\{d, j, k\}$ sono connesse.

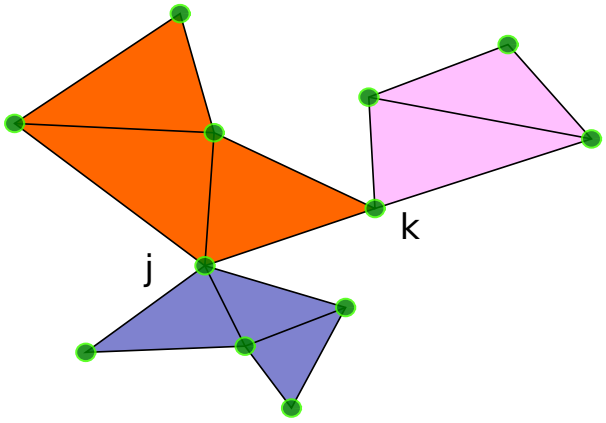

Figura 1.4: In questa figura è mostrato un semplice esempio di comunità di 3-cricche. Le 3-cricche che fanno parte di una stessa comunità sono colorate con lo stesso colore.

1.2 Cenni di omologia persistente

L'omologia persistente è uno strumento molto utile per individuare alcune caratteristiche topologiche di uno spazio. In generale, la teoria è stata sviluppata per spazi topologici, ma si conoscono delle varianti. Noi esporremo brevemente la teoria per complessi simpliciali seguendo [2] e [3].

Definizione 8 (complesso simpliciale)**.** Un *complesso simpliciale* K è un insieme di *simplessi*, sottoinsiemi finiti non vuoti di un insieme di vertici $V(K)$, tali che

- ogni sottoinsieme di $V(K)$ con esattamente un elemento è un simplesso
- ogni sottoinsieme non vuoto di un simplesso è un simplesso

Definizione 9 (p-simplesso)**.** Un *p-simplesso* $\sigma = [u_0, \ldots, u_p]$ è un simplesso che contiene esattamente $p + 1$ vertici e la sua dimensione è p.

La dimensione massima dei simplessi di un complesso simpliciale K è la *dimensione del complesso.*

Osservazione 1. Un grafo è un complesso simpliciale di dimensione al più 1

Definizione 10 (p-catena)**.** Dato un complesso simpliciale finito K, ogni combinazione lineare formale a coefficienti in $\mathbb{Z}_2$ di p-simplessi è detta *p-catena* di K
Indichiamo con C_p lo $\mathbb{Z}_2$-spazio vettoriale formato dalle p-catene di K

Definizione 11 (operatore bordo)**.** Per ogni $p \in \mathbb{Z}$ chiamiamo *operatore bordo* la trasformazione lineare $\partial_p(\sigma) : C_p \longrightarrow C_{p-1}$ definita da

$$\partial_p(\sigma) = \sum_{j=0}^{n} [u_0, \ldots, \widehat{u}_j, \ldots, u_p]$$

dove $\sigma = [u_0, \ldots, u_p]$ è un p-simplesso e $[u_0, \ldots, \widehat{u}_j, \ldots, u_p]$ denota la sua faccia generata da tutti i suoi vertici tranne u_j $(j = 0, \ldots, p)$

Osservazione 2. Poiché ∂_p è una trasformazione lineare basta definirla sui suoi p-simplessi e poi estenderla per linearità.

Si prova che $\partial_p \partial_{p+1} = 0$, cioè $Im\partial_{p+1} \subseteq Ker\partial_p$.
Indichiamo con B_p l'immagine dell'operatore ∂_{p+1} e chiamiamo i suoi elementi *p-bordi.* Z_p indica invece il nucleo dell'operatore ∂_p e i suoi elementi sono detti *p-cicli.*

Definizione 12 (gruppo di omologia)**.** Il *p-esimo gruppo di omologia simpliciale di K* è il quoziente

$$H_p(K) = \frac{Z_p(K)}{B_p(K)}$$

in cui le classi di omologia non nulle sono rappresentate da cicli che non sono bordi.

Il *p-esimo numero di Betti* di un complesso simpliciale K è la dimensione del p-esimo gruppo di omologia $H_p(K)$ e si indica con $\beta_p(K)$

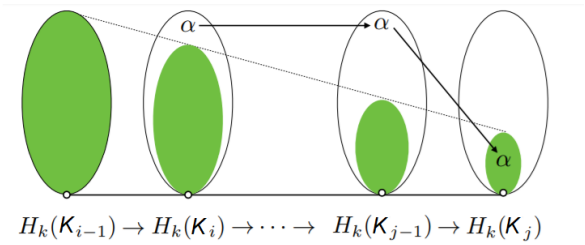

Figura 1.5: La classe di omologia α è nata in K_i poiché non è nell'immagine di $i_k^{i-1,i}$ colorata in verde. Muore in K_j poiché è contenuta nell'immagine della mappa indotta dall'inclusione $K_{j-1} \subset K_j$ ma non nell'immagine della mappa indotta dall'inclusione $K_{i-1} \subset K_{j-1}$

Definizione 13 (filtrazione)**.** Dato un complesso simpliciale K finito, consideriamo una funzione $f : K \longrightarrow \mathbb{R}$ tale che se σ e τ sono simplessi di K con $\sigma \subset \tau$ si avrà $f(\sigma) \leq f(\tau)$. Sia inoltre per ogni $a \in \mathbb{R}$, $K(a) = \{\sigma \in K | f(\sigma) \leq a\}$ l'*insieme sottolivello*. Una *filtrazione di* K è una sequenza di sottocomplessi di K

$$\varnothing = K_0 \subsetneq K_1 \subsetneq \cdots \subsetneq K_p = K$$

dove $K_i = K(\alpha_i)$ con $\alpha_i \in [a_i, a_{i+1})$ per qualche $a_i \in \mathbb{R}$

Data una filtrazione di un complesso simpliciale K la mappa di inclusione $i^{r,s} : K_r \hookrightarrow K_s$, per ogni coppia di r, s con $0 \leq r \leq s \leq p$, induce un omomorfismo $i_k^{r,s} : H_k(K_r) \longrightarrow H_k(K_s)$ per ogni grado k.

Definizione 14. Il *k-esimo gruppo di omologia (r,s)-persistente* è il gruppo

$$PH_k(r, s) := i_k^{r,s}(H_k(K_r)) \subseteq H_k(K_s)$$

Il *k-esimo Numero di Betti Persistente* $\beta_{r,s}^k$ è la dimensione di $PH_k(r, s)$ e conta quante classi di k-cicli di $H_k(K_r)$ sopravvivono in $H_k(K_s)$

Possiamo costruire un *diagramma di persistenza* per ogni grado k nel modo seguente: se una classe di omologia nasce nel sottocomplesso K_i e muore nel sottocomplesso K_j (Figura 1.5), le associamo le sue coordinate di nascita e morte (i, j) che individuano un *cornerpoint*. Si chiama *persistenza di una coppia* il valore $pers(i, j) := |j - i|$. Se una classe di omologia nasce ma non muore, verrà rappresentata nel diagramma di persistenza da una retta detta *cornerline* e le assoceremo la coppia (i, ∞). Il diagramma di persistenza sarà l'insieme di cornerpoint e cornerline (Figura 1.6).

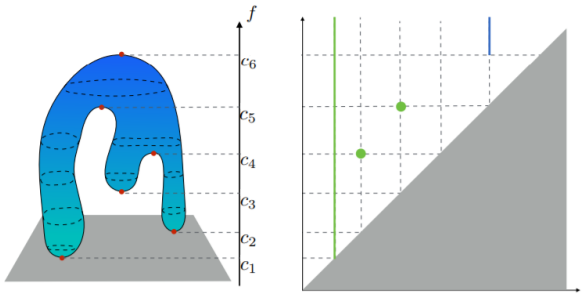

Figura 1.6: Esempio di diagramma di persistenza della sfera topologica: i cornerpoint e la cornerline in verde sono relativi al grado 0, la cornerline in blu si riferisce al grado 2

1.3 Individuare nodi centrali in una rete

Siamo finalmente pronti a comprendere il metodo sviluppato in [1] che utilizza le nozioni viste nelle sezioni precedenti come strumenti fondamentali.

In teoria dei grafi la definizione di comunità di cricche è intuitiva, rispecchiando il concetto ordinario di comunità, cioè un gruppo di enti che condividono una o più caratteristiche. Di fatto, individuare le comunità di una rete [1] significa rintracciare gruppi di vertici maggiormente interconnessi ma con legami deboli tra i vari gruppi. In un certo senso significa decomporre la rete di partenza, di difficile comprensione, in tante sottoreti più semplici da analizzare. Ne viene che le comunità sono uno strumento utile per definire la struttura globale di una rete complessa e ora vedremo come intervengono anche nell'analisi locale della rete, facilitando l'individuazione di nodi rilevanti.
Conoscendo le comunità è possibile classificare i vertici a seconda della posizione che occupano [4]: vertici con una posizione centrale, cioè connessi a molti altri vertici della comunità, potrebbero avere un ruolo importante all'interno del gruppo; i vertici ai bordi tra due o più comunità, come ad esempio i vertici j e k della Figura 1.4, permettono relazioni e scambi tra comunità differenti.

In [1] si definisce una misura di centralità che deriva dalla seguente definizione

Definizione 15 (cross-clique connectivity)**.** La *cross-clique connectivity* di un vertice v è il numero di cricche di cui fa parte.

Estendendo tale nozione alle comunità di cricche possiamo definire la centralità di un vertice nel modo seguente:

Definizione 16 (centralità di un vertice)**.** La *centralità di un vertice* v è definita come

$$\Gamma_C(v) := \sum_{v \in C} pers(C)$$

dove C indica le comunità di cricche di cui v fa parte e $pers(C)$ è la persistenza di una comunità di cricche e ne verrà chiarito il significato nel seguito.

Dato $G = (V, E)$ un grafo con n vertici e una funzione peso $\alpha : V \longrightarrow \mathbb{R}$ definita sui suoi vertici, partendo dall'insieme vuoto, consideriamo la filtrazione del grafo G

$$G_0 \subseteq G_1 \subseteq \ldots \subseteq G_n$$

dove ogni $G_k = (V_k, E_k)$ della filtrazione è tale che

$$V_k = \{v \in V | \alpha(v) \leq \alpha_k\} \quad e \quad E_k = \{e \in E | e = \{u, v\}, \alpha(e) := max(\alpha(u), \alpha(v)) \leq \alpha_k\}$$

Data una cricca ρ possiamo definire $\alpha(\rho) := \max_{v \in v(\rho)} \alpha(v)$, cioè il peso di una cricca è il massimo tra i pesi dei suoi vertici.
A questo punto si può costruire il grafo $G^k = (V^k, E^k)$ di k-cricche connesse che ha un vertice per ogni k-cricca di G e $E^k := \{\{\rho, \rho'\} | \rho \text{ e } \rho' \in V^k, k\text{-cricche adiacenti }\}$.

La funzione peso sui lati di G^k sarà definita da $\alpha(\{\rho, \rho'\}) := \max(\alpha(\rho), \alpha(\rho'))$.

Quello che si fa quindi è definire la filtrazione di un nuovo grafo che ha un numero di vertici e lati inferiore a quello di partenza e che evidenzia le relazioni tra le cricche e le comunità di cricche.

Poiché un grafo è un complesso simpliciale di dimensione al più uno, possiamo considerare i gruppi di omologia persistenti *0*-dimensionali associati alla filtrazione e ottenere il suo diagramma di persistenza $\mathcal{D}$. Così facendo la durata della vita di una componente connessa non sarà altro che la persistenza di una comunità di cricche e si può così comprendere pienamente il significato della Definizione 16. Più è alto il valore della persistenza e più sarà ritenuta importante quella comunità all'interno del grafo.

Ma perché è così importante questa definizione per il nostro scopo? Si evince che i vertici che fanno parte di molte comunità di cricche con bassa persistenza, avranno un valore di centralità basso [1]; invece, i vertici che fanno parte anche solo di poche comunità che però hanno una persistenza alta, cioè riferita a punti più lontani dalla diagonale del diagramma e quindi a comunità più importanti, avranno un valore di centralità alto. Questo vuol dire in sostanza che nodi con alto grado non saranno automaticamente considerati i più importanti di un grafo.

A questo punto abbiamo un primo metodo per individuare i nodi centrali di un grafo. Concludiamo questo primo capitolo ponendoci un'ultima domanda la cui risposta è il filo conduttore di tutto questo elaborato.
Perché ci interessano questi nodi? I nodi centrali di una rete sono quelli che, una volta individuati, si possono "tenere d'occhio", quelli su cui si può agire per avere effetto anche su tutti gli altri nodi a lui collegati, sono quelli che possiamo considerare centri di propagazione di informazioni. Di conseguenza, eliminare un nodo centrale potrebbe voler dire bloccare la propagazione di una informazione o far sparire informazioni importanti di una rete come le relazioni tra varie comunità.
Togliere un nodo centrale da una rete è come togliere uno dei personaggi principali di una storia: la parte di narrazione che lo riguarda perderebbe completamente di significato.
Lo studio di nodi centrali estende in qualche senso lo studio della connettività, nato nella seconda guerra mondiale quando gli Alleati cercavano un *vertex cut* ottimale con cui disconnettere la rete di approvvigionamento della prima linea tedesca dopo lo sbarco in Normandia. In un certo senso, individuare i nodi centrali di una tale rete permette di infliggere il massimo danno con il minimo costo, pur senza la pretesa di una effettiva disconnessione.

Capitolo 2

Persistenza combinatoria per hub in reti

È naturale interpretare un grafo come uno spazio topologico pensandolo come un complesso simpliciale finito di dimensione uno: il teorema di realizzazione geometrica ci assicura che possiamo associare ad ogni complesso simpliciale finito, uno spazio topologico [5], detto *corpo* del complesso. Questa interpretazione in bassa dimensione è naturalmente limitata in termini di quantità di informazione che può comunicare. Nella 1.2 abbiamo visto come l'utilizzo di concetti basilari della teoria dei grafi e costruzioni topologiche permettano di estrarre nuova informazione da un grafo pesato dato. In questo capitolo, basato su [2], descriveremo come sia possibile studiare grafi pesati nel dominio della persistenza, senza ricorrere a mediazioni topologiche.
Uno studio in via di sviluppo [2], condotto da Massimo Ferri dell'Università di Bologna, e da Mattia G. Bergomi, del centro di ricerca Champalimaud di Lisbona, mira a slegare la persistenza dall'omologia persistente, fornendola di una teoria propria e generale applicabile non solo in contesti riconducibili a spazi topologici o complessi simpliciali. La teoria non è ancora assestata nel suo contesto più generale, possiamo però formularla nel caso specifico dei grafi pesati, oggetti delle nostre applicazioni.

2.1 Funzioni di persistenza

Nel seguito definiremo le *funzioni di persistenza* di cui i numeri di Betti persistenti, ad ogni grado, sono un esempio. Quanto diremo sulle funzioni di persistenza, fornisce loro la struttura a triangoli sovrapposti, tipica delle funzioni dei numeri di Betti persistenti, evidenziando il fatto che ne sono una generalizzazione.

Definizione 17 (grafo pesato)**.** Un *grafo pesato* è una coppia (G, f), dove $G = (V, E)$ è un grafo e $f : E \longrightarrow \mathbb{R}$ è una funzione definita sui lati del grafo.

La funzione f è detta *funzione peso* o *funzione filtrante* e induce una filtrazione sul grafo G. Per ogni numero reale u, $G_u = (V, f^{-1}((-\infty, u]))$ è il sottografo di G indotto dall'insieme dei lati $f^{-1}((-\infty, u])$.

Poniamo $\Delta^+ = \{(u, v) \in \mathbb{R}^2 | u < v\}$ e $\Delta = \{(u, v) \in \mathbb{R}^2 | u = v\}$

Definizione 18 (funzione di persistenza)**.** Dato un grafo pesato (G, f), ogni funzione $\lambda_{(G,f)}(u, v) : \Delta^+ \longrightarrow \mathbb{Z}$ è detta *funzione di persistenza* se soddisfa le seguenti condizioni

1. $\lambda_{(G,f)}(u, v)$ è non decrescente in u e non crescente in v;
2. per ogni $u_1, u_2, v_1, v_2 \in \mathbb{R}$ tali che $u_1 \leq u_2 < v_1 \leq v_2$, vale la disuguaglianza:

$$\lambda_{(G,f)}(u_2, v_1) - \lambda_{(G,f)}(u_1, v_1) \geq \lambda_{(G,f)}(u_2, v_2) - \lambda_{(G,f)}(u_1, v_2)$$

.

Proposizione 1. *[2, Prop. 12] Valgono le seguenti affermazioni*

1. *Se $\overline{u}$ è un punto di discontinuità per $\lambda_{(G,f)}(\cdot, \overline{v})$ e $\overline{u} < v < \overline{v}$ allora $\overline{u}$ è un punto di discontinuità anche per $\lambda_{(G,f)}(\cdot, v)$*
2. *Se $\overline{v}$ è un punto di discontinuità per $\lambda_{(G,f)}(\overline{u}, \cdot)$ e $\overline{u} < u < \overline{v}$ allora $\overline{v}$ è un punto di discontinuità anche per $\lambda_{(G,f)}(u, \cdot)$* □

Osservazione 3. La proposizione precedente suggerisce che i punti di discontinuità di una funzione di persistenza formano segmenti paralleli agli assi coordinati.

Proposizione 2. *[2, Prop. 13] Ogni intorno aperto e connesso per archi di un punto di discontinuità di $\lambda_{(G,f)}$ contiene almeno un punto di discontinuità o nella prima o nella seconda variabile.* □

Osservazione 4. La proposizione suggerisce che i punti di discontinuità di una funzione di persistenza non sono isolati.

Proposizione 3. *[2, Prop. 14] Per ogni punto $\overline{p} = (\overline{u}, \overline{v}) \in \Delta^+$, esiste un $\varepsilon > 0$ tale che l'aperto $W_\varepsilon = \{(u, v) \in \mathbb{R}^2 | 0 <| u - \overline{u} |< \varepsilon, 0 <| v - \overline{v} |< \varepsilon\}$ non contiene punti di discontinuità di $\lambda_{(G,f)}$.* □

Osservazione 5. La proposizione suggerisce che intorno ai segmenti di discontinuità di una funzione di persistenza, ci sono aree che non contengono alcun punto di discontinuità.

Nella sezione 1.2 abbiamo parlato dei cornerpoint e delle cornerline, i punti dei diagrammi di persistenza. Diamo ora la definizione di molteplicità che serve a stabilire se un punto $p \in \Delta^+$ è un punto di un diagramma di persistenza.

Definizione 19 (molteplicità punti propri)**.** La *molteplicità* di un punto $p = (u, v) \in \Delta^+$ per $\lambda_{(G,f)}$ è il numero $\mu(p)$ uguale al minimo, su tutti i positivi reali ε con $u + \varepsilon < v - \varepsilon$, di

$$\lambda_{(G,f)}(u + \varepsilon, v - \varepsilon) - \lambda_{(G,f)}(u - \varepsilon, v - \varepsilon) - \lambda_{(G,f)}(u + \varepsilon, v + \varepsilon) + \lambda_{(G,f)}(u - \varepsilon, v + \varepsilon)$$

Un punto $p \in \Delta^+$ si dice *cornerpoint proprio* se la sua molteplicità è positiva.

Definizione 20 (molteplicità punti impropri)**.** Identifichiamo ogni retta verticale r, di equazione $u = k$ con la coppia (k, ∞) e definiamo la sua *molteplicità* per $\lambda_{(G,f)}$, il numero $\mu(r)$ uguale al minimo, su tutti i positivi reali ε con $k + \varepsilon < 1/\varepsilon$, di

$$\lambda_{(G,f)}(k + \varepsilon, 1/\varepsilon) - \lambda_{(G,f)}(k - \varepsilon, 1/\varepsilon)$$

Una retta r si dice *cornerpoint all'infinito* se la sua molteplicità è positiva.

Proposizione 4. *[2, Prop. 15] Se $(\overline{u}, \overline{v})$ è un cornerpoint, valgono le seguenti affermazioni:*

- *Se $\overline{u} < v < \overline{v}$, allora $\overline{u}$ è un punto di discontinuità per $\lambda_{(G,f)}(\cdot, v)$*
- *Se $\overline{u} < u < \overline{v} < +\infty$ allora $\overline{v}$ è un punto di discontinuità per $\lambda_{(G,f)}(u, \cdot)$* □

Proposizione 5. *[2, Prop. 16]*

1. *Se $\overline{u}$ è un punto di discontinuità per $\lambda_{(G,f)}(\cdot, \overline{v})$ con $\overline{u} < \overline{v}$, allora o un cornerpoint di $\lambda_{(G,f)}$ si trova sulla semiretta chiusa $\{(\overline{u}, v) \in \mathbb{R}^2 | \overline{v} < v\}$, o la retta $u = \overline{u}$ è un cornerpoint all'infinito, o si verificano entrambi i casi.*
2. *Se $\overline{v}$ è un punto di discontinuità per $\lambda_{(G,f)}(\overline{u}, \cdot)$ con $\overline{u} < \overline{v}$, allora un cornerpoint di $\lambda_{(G,f)}$ si trova sulla semiretta chiusa $\{(u, \overline{v}) \in \mathbb{R}^2 | u < \overline{u}\}$.* □

Osservazione 6. Le due proposizioni precedenti mostrano che un cornerpoint proprio è il punto di intersezione tra un segmento di discontinuità orizzontale e uno verticale; un cornerpoint all'infinito fornisce la posizione di una semiretta di discontinuità.

Definizione 21 (diagramma di persistenza)**.** Il *diagramma di persistenza D(f) di $\lambda_{(G,f)}$* è il multiinsieme dei suoi cornerpoint, propri e all'infinito, ognuno ripetuto secondo la sua molteplicità, e di tutti i punti della diagonale Δ, ognuno contato con molteplicità infinita ($\aleph_0$).

2.2 Hubs

Nella sezione precedente abbiamo definito le funzioni di persistenza e le loro caratteristiche senza la mediazione di costrutti simpliciali ausiliari. In questa sezione proveremo a formalizzare l'idea intuitiva di "hub" in un grafo e daremo due esempi di funzioni di persistenza legate a questi.
Come accade nel metodo visto nel primo capitolo, avere grado maggiore degli altri vertici, non è sufficiente per definire un hub. Anche proprietà più complesse come ad esempio che un vertice abbia somma dei pesi dei lati incidenti maggiore degli altri, non è di per sé sufficiente perché quel vertice sia un hub. In questa sezione costruiremo delle funzioni di persistenza, quindi otterremo dei diagrammi di persistenza e, con il metodo che illustreremo nella prossima sezione, selezioneremo i cornerpoint più rilevanti.

Sia (G, f) un grafo pesato e per ogni numero reale u, sia G_u il sottografo di G indotto dall'insieme dei lati $f^{-1}((-\infty, u])$.

Definizione 22 (t-hub)**.** Un *hub temporaneo(t-hub) al livello u* è un vertice di G_u che ha grado strettamente maggiore rispetto agli altri vertici con cui è direttamente collegato.

Osservazione 7. È possibile dare varianti di questa definizione scegliendo una qualsiasi funzione definita su $V \cup E$. Nelle nostre applicazioni useremo come definizioni di hub temporaneo al livello u quella data, cioè $grado(v) > grado(Vicini(v))$, e la variante $grado(v) \geqslant grado(Vicini(v))$.

2.2.1 Steady hubs

Come si può intuire, nulla vieta che un vertice possa essere un t-hub a più livelli consecutivi. Formalizziamo questa intuizione e in seguito, a partire da questa proprietà, arriveremo alla definizione della prima funzione di persistenza che considereremo nelle nostre applicazioni.

Definizione 23 (s-hub)**.** Dati due livelli u e v, uno *steady hub (s-hub) a (u, v)* $\in \Delta^+$ è un vertice che è t-hub a tutti i livelli w compresi tra u e v, $u \leq w \leq v$.

Dato $(u, v) \in \Delta^+$ e una coppia (G, f), indichiamo con $S_{(G,f)}(u, v)$ l'insieme degli steady hub a (u, v).

Lemma 1. *Se* $u \leq u' < v' \leq v$*, allora* $S_{(G,f)}(u, v) \subseteq S_{(G,f)}(u', v')$.

Dimostrazione. Per definizione, se un vertice è s-hub a (u, v) lo è anche a (u', v'). □

Sia $\overline{\sigma}_{(G,f)} : \Delta^+ \longrightarrow \mathcal{P}(V)$, con $\mathcal{P}(V)$ insieme delle parti dell'insieme dei vertici V, definita da $\overline{\sigma}_{(G,f)}(u, v) = S_{(G,f)}(u, v)\ \forall (u, v) \in \Delta^+$, cioè $\overline{\sigma}_{(G,f)}$ associa ad ogni coppia di

livelli (u, v) l'insieme dei vertici che sono steady hub a (u, v)

Definiamo ora la funzione $\sigma_{(G,f)} : \Delta^+ \longrightarrow \mathbb{Z}$ come segue: per ogni $(u, v) \in \Delta^+$, $\sigma_{(G,f)}(u, v) = |\overline{\sigma}_{(G,f)}(u, v)|$. Tale funzione conta quanti vertici sono s-hub tra i due livelli e vale la seguente proposizione:

Proposizione 6. *σ è una funzione di persistenza.*

Dimostrazione. Per provare che σ è una funzione di persistenza, bisogna provare le due condizioni della Definizione 18. Procediamo.

1. Per il Lemma 1, se $u < \overline{u} < v$ allora $S_{(G,f)}(u, v) \subseteq S_{(G,f)}(\overline{u}, v)$, pertanto $\sigma_{(G,f)}(u, v) \leq \sigma_{(G,f)}(\overline{u}, v)$.
   Se $u < v < \overline{v}$ allora $S_{(G,f)}(u, v) \supseteq S_{(G,f)}(u, \overline{v})$, pertanto $\sigma_{(G,f)}(u, v) \geq \sigma_{(G,f)}(u, \overline{v})$.

2. Dati $u_1 \leq u_2 < v_1 \leq v_2$, si ha che $S_{(G,f)}(u_1, v_1) \subseteq S_{(G,f)}(u_2, v_1)$. Allora $\sigma_{(G,f)}(u_2, v_1) - \sigma_{(G,f)}(u_1, v_1)$ è il numero di s-hub a (u_2, v_1) che non sono t-hub per qualche w con $u_1 \leq w < u_2$. Analogamente per $\sigma_{(G,f)}(u_2, v_2) - \sigma_{(G,f)}(u_1, v_2)$.
   Ora, ogni s-hub a (u_1, v_2) che non è t-hub a w con $u_1 \leq w < u_2$, è anche un s-hub a (u_1, v_1) che non è t-hub per lo stesso w. Pertanto $S_{(G,f)}(u_2, v_1) - S_{(G,f)}(u_1, v_1) \supseteq S_{(G,f)}(u_2, v_2) - S_{(G,f)}(u_1, v_2)$ e $\sigma_{(G,f)}(u_2, v_1) - \sigma_{(G,f)}(u_1, v_1) \geq \sigma_{(G,f)}(u_2, v_2) - \sigma_{(G,f)}(u_1, v_2)$.

□

2.2.2 Ranging hubs

Dalla definizione di t-hub si intuisce anche che un vertice può essere un t-hub "a intermittenza". Ciò vuol dire che un hub potrebbe venire oscurato dall'aggiunta di lati in sottolivelli successivi della filtrazione, per poi riapparire nel corso della stessa. Questa, nello studio dei centri nevralgici di un grafo può rappresentare una proprietà chiave, come ad esempio dal punto di vista discusso alla fine della sezione 1.3 sulla disconnessione delle linee di approvvigionamento nemiche. Riformuliamo quindi quanto appena visto per gli steady hub, per vertici che hanno questa proprietà

Definizione 24 (r-hub)**.** Dati due livelli u e v, un *ranging hub (r-hub) a (u, v)* $\in \Delta^+$ è un vertice per cui esistono livelli $w \leq u$ e $w' \geq v$ ai quali è un t-hub.

Dato $(u, v) \in \Delta^+$, indichiamo con $R_{(G,f)}(u, v)$ l'insieme dei ranging hub a (u, v).

Lemma 2. *Se $u \leq u' < v' \leq v$, allora $R_{(G,f)}(u, v) \subseteq R_{(G,f)}(u', v')$.*

Dimostrazione. Per definizione, se un vertice è r-hub a (u, v) lo è anche a (u', v') □

Sia $\overline{\rho}_{(G,f)} : \Delta^+ \longrightarrow \mathcal{P}(V)$, definita da $\overline{\rho}_{(G,f)}(u,v) = R_{(G,f)}(u,v)\ \forall (u,v) \in \Delta^+$, cioè $\overline{\rho}_{(G,f)}$ associa ad ogni coppia di livelli (u,v) l'insieme dei vertici che sono ranging hub a (u,v)

Definiamo ora la funzione $\rho_{(G,f)} : \Delta^+ \longrightarrow \mathbb{Z}$ come segue: per ogni $(u,v) \in \Delta^+$, $\rho_{(G,f)}(u,v) = |\overline{\rho}_{(G,f)}(u,v)|$. Tale funzione conta quanti vertici sono r-hub a (u,v) e vale la seguente proposizione:

Proposizione 7. *ρ è una funzione di persistenza.*

Dimostrazione. Come prima, per provare che ρ è una funzione di persistenza, bisogna provare le due condizioni della Definizione 18.

1. Vale quanto detto per l'analoga dimostrazione sugli steady hub.

2. Dati $u_1 \leq u_2 < v_1 \leq v_2$, per il Lemma 2, si ha che $R_{(G,f)}(u_1,v_1) \subseteq R_{(G,f)}(u_2,v_1)$. Allora $\rho_{(G,f)}(u_2,v_1) - \rho_{(G,f)}(u_1,v_1)$ è il numero di r-hub a (u_2,v_1) che non sono t-hub per tutti i livelli w con $w \leq u_1$. Analogamente per $\rho_{(G,f)}(u_2,v_2) - \rho_{(G,f)}(u_1,v_2)$. Ora, ogni r-hub a (u_1,v_2) che non è t-hub per tutti i livelli w con $w \leq u_1$, è anche un r-hub a (u_1,v_1) che non è t-hub per gli stessi livelli w. Pertanto $R_{(G,f)}(u_2,v_1) - R_{(G,f)}(u_1,v_1) \supseteq R_{(G,f)}(u_2,v_2) - R_{(G,f)}(u_1,v_2)$ e $\rho_{(G,f)}(u_2,v_1) - \rho_{(G,f)}(u_1,v_1) \geq \rho_{(G,f)}(u_2,v_2) - \rho_{(G,f)}(u_1,v_2)$.

□

Osservazione 8. È importante sottolineare che σ e ρ non sono *le* funzioni di persistenza per gli steady e i ranging hub, ma sono delle possibilità, delle proposte di soluzione. Ogni altra funzione che soddisfi le condizioni *1.* e *2.* della Definizione 18 può essere considerata valida a seconda del contesto.

2.2.3 Persistenza di steady e ranging hub

Date delle funzioni di persistenza, otteniamo dei diagrammi di persistenza. Consideriamo il grafo pesato in alto nella Figura 2.1 e individuiamone gli s-hub e gli r-hub applicando le definizioni delle sezioni precedenti.

Sia G come in figura e sia $W = \{w_{ij}\}$ l'insieme dei pesi dei suoi lati. Costruiamo la filtrazione visibile in figura considerando come funzione filtrante la funzione $f : W \subset \mathbb{R} \longrightarrow \mathbb{R}$ con $f(w_{ij}) = \frac{1}{w_{ij}}$, in modo tale che i primi lati a comparire nella filtrazione siano quelli con peso maggiore. Si avrà quindi come filtrazione del grafo pesato (G,f) la sequenza di sottografi

$$\emptyset \subseteq G_{\frac{1}{9}} \subseteq G_{\frac{1}{8}} \subseteq \cdots \subseteq G_1 = G$$

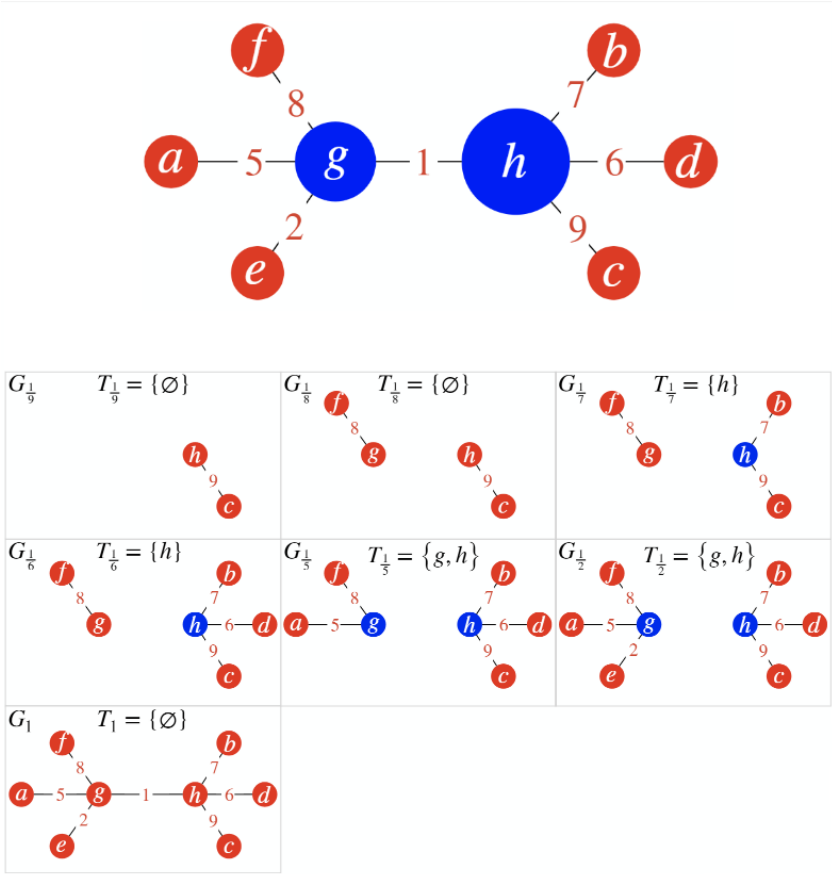

Figura 2.1: In alto il grafo pesato preso in esame. In basso la sua filtrazione costruita considerando sui lati l'inverso del peso. Per ogni sottografo G_i della filtrazione è definito l'insieme T_i degli hub temporanei

Poniamo ora l'attenzione su ogni sottografo per individuare, se esistono, i t-hub presenti in ciascuno. Come è possibile notare, per ogni sottografo G_i della filtrazione è definito l'insieme T_i degli hub temporanei e si ha che quelli non vuoti sono:

$$T_{\frac{1}{7}} = \{h\}, T_{\frac{1}{6}} = \{h\}, T_{\frac{1}{5}} = \{h, g\}, T_{\frac{1}{2}} = \{h, g\}$$

A questo punto, siamo in grado di individuare gli steady hub e i ranging hub applicando le definizioni 24 e 23 : il vertice h è uno steady hub perché è t-hub per $G_{\frac{1}{7}}$, $G_{\frac{1}{6}}$, $G_{\frac{1}{5}}$ e $G_{\frac{1}{2}}$ che sono sottografi consecutivi della filtrazione. Anche il vertice g è uno steady hub in quanto compare come t-hub nei sottografi consecutivi $G_{\frac{1}{5}}$ e $G_{\frac{1}{2}}$.

Osservazione 9. Per definizione, uno steady hub è sempre un ranging hub. Dunque, i vertici h e g oltre ad essere steady hub sono anche ranging hub.

Costruiamo infine i diagrammi di persistenza relativi agli hub. Come sappiamo, per farlo dobbiamo individuare i cornerpoint. Poiché il vertice h inizia ad essere un s-hub nel

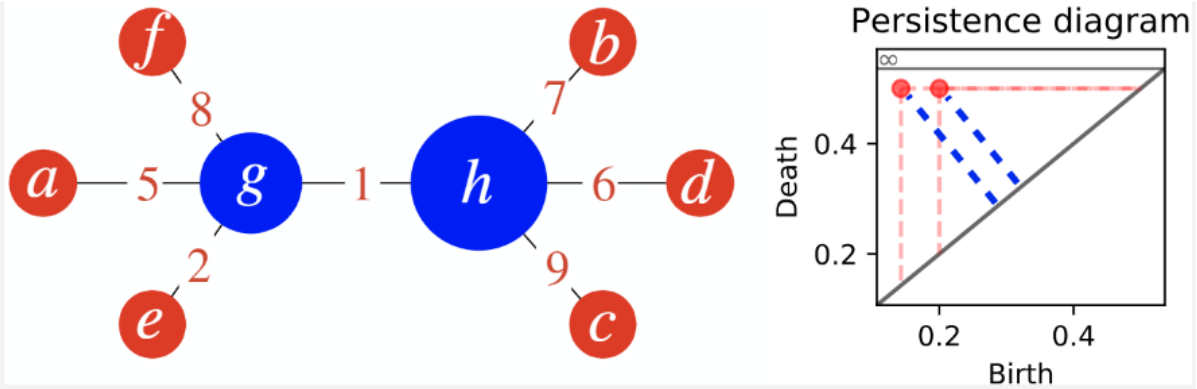

Figura 2.2: A sinistra il grafo di partenza in cui sono evidenziati in blu gli steady hub; il raggio del cerchio rappresenta la persistenza, distanza dalla diagonale. A destra il diagramma di presistenza per gli steady hub

terzo sottografo della filtrazione e smette di esserlo nel sesto, è associato al cornerpoint di coordinate (0.1,0.5) in $D(f)$; con un ragionamento analogo associamo al cornerpoint di coordinate (0.2,0.5) il vertice g. Figura 2.2

Osservazione 10. Se un vertice v risulta steady hub per più intervalli distinti di sottografi, sarà associato a tanti cornerpoint quanti sono gli intervalli.
Per quanto riguarda i ranging hub, se il vertice $\overline{v}$ è t-hub in più di due sottografi della filtrazione, è associamo al cornerpoint di coordinate (j, k) che sono rispettivamente il minimo e il massimo indice in cui $\overline{v}$ è t-hub.

2.3 Selezione di cornerpoint

Se ci troviamo in presenza di una rete in cui interagiscono molti nodi, i diagrammi di persistenza per steady e ranging hub potrebbero avere un elevato numero di cornerpoint. Sarà quindi utile avere uno strumento capace di selezionare gli hub più rilevanti. Quello che descriveremo in questa sezione è proprio un metodo di questo tipo: il metodo di Kurlin. Partiamo da alcune definizioni di base [6] per capire come funziona il metodo.

Definizione 25 (Complesso di Delaunay)**.** Data una nuvola $C = \{p_1, \cdots, p_n\} \subset \mathbb{R}^2$ di n punti, il *complesso di Delaunay Del(C)* associato a C è un complesso 2-dimensionale formato da tutti i triangoli con vertici $p_i, p_j, p_k \in C$ tali che la circonferenza a loro circoscritta non contiene altri punti di C. (Figura 2.3)

Definizione 26 (Cella di Voronoi)**.** Sia $p_i \in C$, la *cella di Voronoi di p* è $V(p_i) = \{q \in \mathbb{R}^2 | d(p_i, q) \leq d(p_j, q), \forall j \neq i\}$, cioè l'insieme di tutti i punti $q \in \mathbb{R}^2$ più vicini a p_i che a tutti gli altri punti di C.

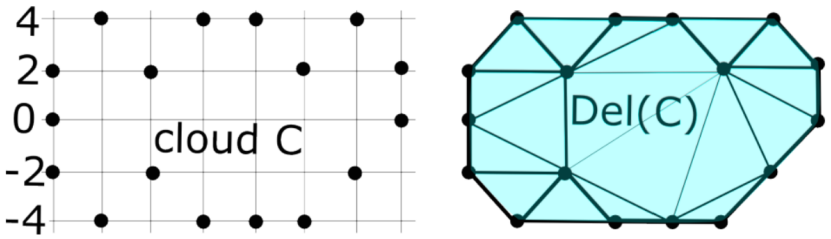

Figura 2.3: A sinistra una nuvola di punti C. A destra il relativo complesso di Delaunay.

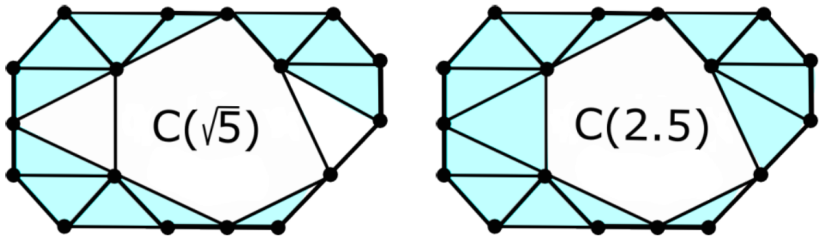

Figura 2.4: Due esempi di α-complessi per $\alpha = \sqrt{5}$ e $\alpha = 2.5$

Osservazione 11. In questi termini $Del(C)$ è il complesso dato da tutti i triangoli con vertici $p, q, r \in C$ tali che l'intersezione $V(p) \cap V(q) \cap V(r)$ è non vuota.

Definizione 27 (α-complesso)**.** Per ogni $p \in \mathbb{R}^2$ e $\alpha > 0$, sia $B(p, \alpha)$ il disco chiuso di centro p e raggio r. Per una nuvola di punti finita $C \subset \mathbb{R}^2$ l' *α-complesso $C(\alpha)$* $\subset \mathbb{R}^2$ è (Figura 2.4) il complesso che contiene

- tutti i lati tra i punti $p, q \in C$ tali che l'intersezione $(V(p) \cap B(p, \alpha)) \cap (V(q) \cap B(q, \alpha))$ è non vuota.
- tutti i triangoli con vertici $p, q, r \in C$ tali che l'intersezione $(V(p) \cap B(p, \alpha)) \cap (V(q) \cap B(q, \alpha)) \cap (V(r) \cap B(r, \alpha))$ è non vuota.

Siamo ora pronti ad illustrare brevemente il metodo di selezione di cornerpoint di Kurlin. Per approfondimenti si veda [6].

Questo metodo utilizza i diagrammi di persistenza per trovare regioni rilevanti date nuvole di punti nel piano. Una delle caratterstiche che rende questo approccio interessante è la sua resistenza al rumore, come discuteremo in seguito. È in grado di selezionare in modo automatico un sottoinsieme di cornerpoint costruendone una gerarchia che si basa sulla loro persistenza: più un punto è lontano dalla diagonale e più la relativa classe di simplessi sarà ritenuta rilevante.
Data una nuvola di punti C si costruisce una filtrazione $\{C(\alpha)\}$ di α-complessi

$$C = C(0) \subset \cdots \subset C(\alpha) \subset \cdots \subset C(\infty) = Del(C)$$

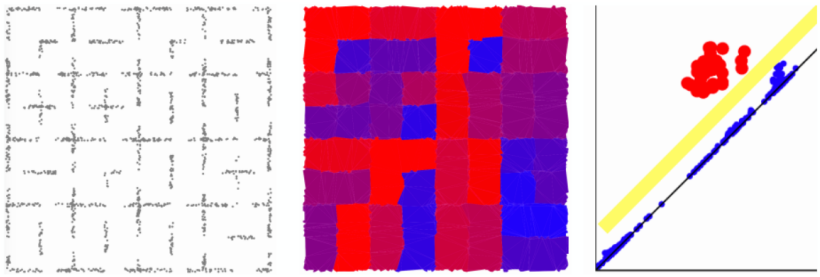

Figura 2.5: A sinistra una nuvola di punti 2D. Al centro la corrispondente segmentazione ottenuta con il metodo di Kurlin. A destra: in giallo è evidenziato il widest gap in $D_1\{C(\alpha)\}$ e in rosso sono evidenziati i punti relativi alle regioni più persistenti.

e si ottiene una segmentazione a partire dalle informazioni contenute in $D_1\{C(\alpha)\}$. Infatti, ad ogni cornerpoint in $D_1\{C(\alpha)\}$ possiamo associare una regione dell'immagine, i cui bordi sono cicli che uniscono punti di C.
Una volta trovati, i cornerpoint di $D_1\{C(\alpha)\}$ vengono ordinati secondo la loro persistenza, ottenendo l'insieme ordinato di punti $\sigma = \{p_1, \cdots, p_n\}$. Infine si trovano i gap diagonali che sono le distanze tra le persistenze di coppie di punti consecutivi di σ.
Si considereranno più significative le regioni corrispondenti ai cornerpoint al di sopra del gap diagonale più ampio. (Figura 2.5)

La particolarità di questo metodo sta nel fatto che non fornisce un'unica segmentazione, ma varie proposte di segmentazione e se ne può scegliere una piuttosto che un'altra a seconda di quello che è più significativo per lo studio in questione. Le varie proposte di segmentazione vengono dal fatto che non è obbligatorio scegliere il gap diagonale più ampio: possiamo sceglierne anche uno di larghezza inferiore e considerare i cornerpoint al di sopra di questo nuovo gap e le relative regioni. Il risultato sarà una nuova segmentazione.
Questa idea nasce dal fatto che normalmente vicino alla diagonale di un diagramma di persistenza si accumula tutto ciò che dipende dal rumore, allora si sceglie di fare quella che viene chiamata "ceretta": vengono eliminati tutti i cornerpoint di una striscia contenente la diagonale nel bordo. Ma come decidere la larghezza della striscia per essere certi di non eliminare informazioni importanti? Il metodo di Kurlin fornisce una soluzione a questo problema in quanto scegliere il gap ottimale in ogni contesto equivale, allo stesso tempo, a scegliere la larghezza della fascia da non prendere in considerazione.

Osservazione 12. Il metodo di Kurlin è sicuramente molto utile, ma presenta un limite: funziona bene quando le regioni della segmentazione hanno area comparabile. Scegliendo un gap decidiamo in qualche modo l'area delle regioni e questo è un limite che può essere superato con studi successivi.

Il modo in cui utilizzeremo il metodo di Kurlin in relazione alla scelta degli hub sarà il seguente: una volta ottenuti i diagrammi di persistenza per steady e ranging hub,

scegliendo un gap, selezioneremo i cornerpoint e da questi risaliremo ai vertici che hanno dato loro origine. Questi saranno gli hub persistenti della rete presa in esame.

Capitolo 3

Applicazioni

In questo capitolo applichiamo la teoria descritta nel Capitolo 2 a tre problemi ben distinti: la rete dei collegamenti tra aeroporti degli Stati Uniti d'America, la rete delle relazioni tra i personaggi de *Les Misérables* e la rete delle lingue ufficiali europee.
Tali problemi riguardano ambiti molto lontani tra loro, che non hanno nulla in comune, se non il fatto di poter essere tutti rappresentati da grafi pesati. La scelta fatta dunque ci permetterà di sperimentare la validità del nostro metodo nei più svariati ambiti, per poterne affermare la generalità.

3.1 I Miserabili

In questa sezione studieremo il grafo che rappresenta la rete delle relazioni tra i personaggi del celebre romanzo di Victor Hugo, *Les Misérables*. Poiché tale rete viene presa come esempio in [1], il nostro scopo è quello di verificare l'efficacia del metodo da noi proposto nella sezione 2.2, confrontando i risultati ottenuti con quelli acquisiti usando il metodo descritto nella sezione 1.3.

3.1.1 Dati raccolti

La rete che studiamo descrive gli incontri tra i personaggi del romanzo *Les Misérables*. L'insieme dei dati in formato CSV utilizzato per costruire il relativo grafo è lo stesso preso in considerazione in [1] e disponibile in rete (Data *Les Misérables*). Il grafo ottenuto ha un totale di 77 vertici e 254 lati e i dati raccolti sono riassumibili come una funzione peso $\alpha : E \longrightarrow \mathbb{N}$ che associa ad ogni lato il numero di compresenza di due personaggi in una stessa scena.
Dato l'insieme dei pesi $W = \{w_{ij}\}$, consideriamo come funzione filtrante la funzione $f(w_{ij}) = \dfrac{1}{w_{ij}}$; così facendo i primi lati a comparire lungo la filtrazione saranno quelli con peso maggiore.

3.1.2 Implementazione

L'implementazione è stata svolta in Python. Il file CSV contenente la struttura del grafo, viene letto dall'algoritmo 3.1, che riportiamo nel seguito.
Per la gestione del dataframe utilizziamo la libreria Pandas di Python rinominata *pd*.

```
def read_graph_structure_from_csv(path_to_csv, sep = ";"):
    d = pd.read_csv(path_to_csv, sep = sep)

    graph_structure = []
    for index, row in d.iterrows():
        graph_structure.append(tuple([row[0],row[1],row[2]]))

    return graph_structure
```

Algoritmo 3.1: Lettura del file CSV e costruzione della struttura del grafo relativo a *Les Misérables*

Questa prima funzione legge il file.csv che raccoglie i dati utilizzati per costruire il grafo e restituisce la struttura che andremo a studiare. Poiché il file è composto da 254 righe ognuna contenente due parole e un numero, quello che fa la funzione è trasformare ogni riga in una terna e aggiungerla ad una lista. In questo modo, alla fine del processo, il grafo avrà la struttura di una lista di terne del tipo *(personaggio1, personaggio2, numero di compresenze)*.

Data la struttura del grafo, ne costruiamo la filtrazione come da algoritmo 3.2. Nella sezione 2.2.3 abbiamo costruito un esempio per calcolare gli steady e i ranging hub in cui è stata considerata la funzione $f : W \subset \mathbb{R} \longrightarrow \mathbb{R}$ con $f(w_{ij}) = \frac{1}{w_{ij}}$ come funzione filtrante, dove $W = \{w_{ij}\}$ è l'insieme dei pesi dati ai lati del grafo. Anche per studiare questo grafo consideriamo la stessa funzione filtrante e i relativi sottolivelli.

```
path_to_csv = '../../data/literature/lesmiserables.csv'
graph_structure = read_graph_structure_from_csv(path_to_csv)
graph = WeightedGraph(graph_structure)
graph.build_graph()
graph.build_filtered_subgraphs()
graph.get_temporary_hubs_along_filtration()
```

Algoritmo 3.2: Filtrazione del grafo relativo a *Les Misérables* usando la funzione filtrante $f(w_{ij}) = \dfrac{1}{w_{ij}}$ e i sottolivelli.

Calcoliamo infine gli steady e i ranging hub considerando il grafo ottenuto come oggetto della classe *WeightedGraph* che, insieme alle altre funzioni visibili nell'algoritmo 3.3, implementa quanto descritto nel Capitolo 2.

```
# Steady
graph.steady_hubs_persistence()
fig, ax = plt.subplots()
graph.steady_pd.plot_gudhi(ax,
        persistence_to_plot = graph.steady_pd.
           persistence_to_plot)
if hasattr(graph.steady_pd, 'proper_cornerpoints_above_gap'):
    graph.steady_pd.plot_nth_widest_gap(ax_handle =ax, n =
       graph.gap_number)
    graph.steady_pd.mark_points_above_diagonal_gaps(ax)
plt.savefig('steady' + '.pdf', dpi= 300)
print('steady hubs:', [c.vertex for c in graph.
   steady_cornerpoints])

# Ranging
graph.ranging_hubs_persistence()
fig, ax = plt.subplots()
graph.ranging_pd.plot_gudhi(ax,
        persistence_to_plot = graph.ranging_pd.
           persistence_to_plot)
if hasattr(graph.ranging_pd, 'proper_cornerpoints_above_gap'):
    graph.ranging_pd.plot_nth_widest_gap(ax_handle =ax, n =
       graph.gap_number)
    graph.ranging_pd.mark_points_above_diagonal_gaps(ax)
plt.savefig('ranging' + '.pdf', dpi= 300)

print('ranging hubs:', [c.vertex for c in graph.
   steady_cornerpoints])
```

Algoritmo 3.3: Diagrammi di persistenza di steady e ranging hub per *Les Misérables* usando la funzione filtrante $f(w_{ij}) = \frac{1}{w_{ij}}$ e i sottolivelli.

Poiché le classi e le funzioni richiamate nel resto del codice sono molto complesse e di non immediata comprensione, si invitano i lettori interessati a prendere visione dei dati e dei codici presenti nell'intero capitolo a contattare

- M.G. Bergomi e-mail: mattia.bergomi@neuro.fchampalimaud.org

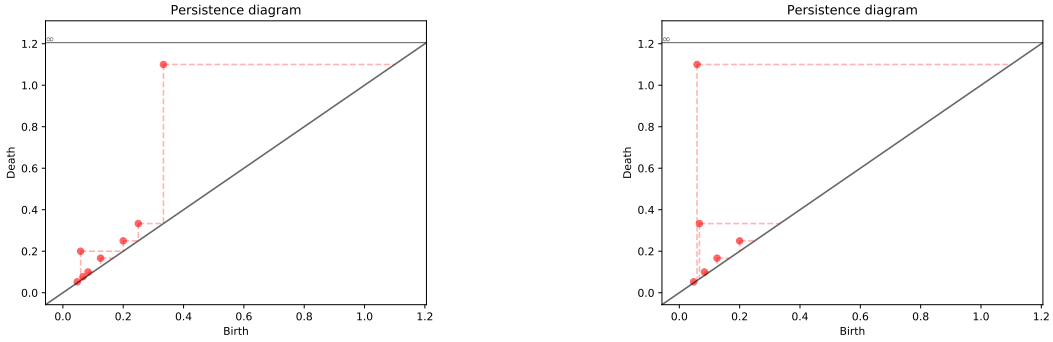

Figura 3.1: Diagrammi di persistenza per steady e ranging hub per il grafo relativo a *Les Misérables*.

Steady hub					
8 cornerpoint 6 vertici					
Cosette	1	Courfeyrac	1	Enjolras	2
Marius	1	Myriel	1	Valjean	2

Tabella 3.1: In tabella sono riportati i personaggi associati ai cornerpoint del diagramma di persistenza per steady hub relativo a *Les Misérables*. I numeri riportati a destra di ogni personaggio indicano il numero di cornerpoint ai quali è assciato.

- M. Ferri e-mail: massimo.ferri@unibo.it
- A. Tavaglione e-mail: antonella.tavaglione@studio.unibo.it

3.1.3 Risultati ottenuti

Come si può vedere dalla figura 3.1 e dalle tabelle 3.2 e 3.1, otteniamo un diagramma di persistenza per steady hub con 8 cornerpoint e uno per ranging hub con 6 cornerpoint. Ad ogni cornerpoint è associato l'insieme dei vertici del grafo che sono steady o ranging hub per la coppia di valori data dalle coordinate del cornerpoint. Per analizzare i risultati

Ranging hub		
6 cornerpoint 6 vertici		
Cosette	Courfeyrac	Enjolras
Marius	Myriel	Valjean

Tabella 3.2: In tabella sono riportati i personaggi associati ai cornerpoint del diagramma di persistenza per ranging hub relativo a *Les Misérables*.

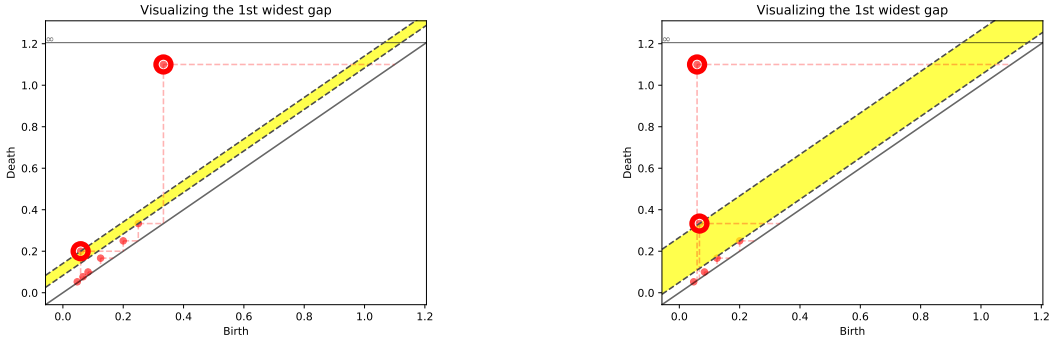

Figura 3.2: Dati i diagrammi di persistenza per steady e ranging hub per il grafo relativo a *Les Misérables* ne selezioniamo i cornerpoint più persistenti con il primo gap più ampio.

Steady hub	
2 cornerpoint	**1 vertice**
Valjean	2

Tabella 3.3: In tabella sono riportati i personaggi associati ai cornerpoint evidenziati nella figura 3.2 (a sinistra) del diagramma di persistenza per steady hub relativo a *Les Misérables*. Il numero riportato a destra del personaggio indica il numero di cornerpoint ai quali è assciato.

ottenuti ci chiediamo se questi siano coerenti con la teoria introdotta nel capitolo 2, oltre che darne un'interpretazione rispetto all'applicazione specifica considerata. Confronteremo infine tali risultati con quelli ottenuti in [1] e riportati nella tabella 3.5.

Come potevamo aspettarci risultano hub le figure di Valjean, protagonista indiscusso di tutto il romanzo e presente in tutti i tomi del libro e Cosette, sua figlia adottiva per la quale si prodiga portando a compimento la promessa fatta alla madre naturale nel primo tomo del libro. Cosette è presente in quattro dei cinque tomi del romanzo e dal terzo tomo in poi la sua figura è strettamente legata a quella di Marius, giovane di buona famiglia che si innamora di lei e tra i principali membri del gruppo di studenti rivoluzionari che combattono sulla barricata. A lui viene dedicato quasi interamente il

Ranging hub	
2 cornerpoint	**2 vertici**
Enjolras	Valjean

Tabella 3.4: In tabella sono riportati i personaggi associati ai cornerpoint evidenziati nella figura 3.2 (a destra) del diagramma di persistenza per ranging hub relativo a *Les Misérables*.

Risultati con CCC		
Enjolras	Fantine	Gavroche
Marius	Valjean	

Tabella 3.5: In tabella sono riportati i personaggi che risultano centrali per il grafo relativo a *Les Misérables* usando la misura di centralità per comunità di cricche (CCC) definita nella sezione 1.3

terzo tomo del romanzo e compare anche nel quarto e nel quinto specialmente in relazione a Cosette. Altra figura di spicco che rientra opportunamente tra gli hub è quella di Enjolras, leader degli Amici dell'ABC, gli studenti rivoluzionari. Anche Courfeyrac è un esponente dell'ABC ed è il miglior amico di Marius, tuttavia non troviamo, al momento, un'argomentazione valida per spiegare il suo essere hub. Il Vescovo di Digne, Myriel, è solitamente considerato un personaggio secondario del romanzo. Riteniamo però che la sua presenza tra gli hub sia del tutto giustificata: è il primo personaggio ad essere presentato, a lui è dedicato il primo tomo in cui si narra il suo incontro con Valjean. Gli insegnamenti e l'esempio di Myriel condizioneranno tutto l'agire di Valjean e di conseguenza tutta la trama.
Potevamo aspettarci tra gli hub la figura di Javert, l'ispettore che tenta in tutti i modi di catturare Valjean e fa di questo la sua ragione di vita. Tuttavia proprio questa sua caratteristica fa sì che la sua storia all'interno del romanzo sia piuttosto slegata da tutte le altre vicende narrate, è come se la sua fosse una storia a sé e di conseguenza ci sembra ragionevole che non risulti tra gli hub. Applichiamo il metodo descritto nella sezione 2.3 per individuare i cornerpoint più persistenti usando il primo gap diagonale più ampio sia per steady che per ranging hub.

Come mostrato nella figura 3.2 e nelle tabelle 3.3 3.4 otteniamo un diagramma di persistenza per steady hub con 2 cornerpoint ai quali è associato il solo personaggio di Valjean, e un diagramma per ranging hub con 2 cornerpoint ai quali sono associati i personaggi di Valjean e Enjolras. Tali risultati ci sembrano ragionevoli: come detto prima, Valjean è senz'altro il protagonista principale di tutta la storia. Immaginiamo ora di dividere il romanzo in due parti, la prima in cui viene narrata la vicenda di Valjean e la seconda in cui si parla dei problemi sociali del paese e delle barricate. In questa seconda parte il protagonista principale sarebbe Enjolras che di fatto risulta come ranging hub.

Come si vede dalla tabella 3.5, i personaggi che risultano centrali usando la misura di centralità per comunità di cricche (CCC) ma non per il nostro metodo sono Fantine e Gavroche. Gavroche, monello di strada, compare come ragazzino furbo e molto attivo che, oltre a partecipare alla barricata, si prende cura, a sua insaputa, di due suoi fratelli minori. La sua figura però non sembra così essenziale per lo svolgimento della narrazione. Potrebbe essere un personaggio costruito dall'autore per muovere le coscienze dei lettori più ostili. Per quanto riguarda invece la figura di Fantine, potevamo aspettarcela tra i

nostri hub. A lei è dedicata solo parte del primo tomo, incarna la sorte che toccava a molte donne del tempo e le sue caratteristiche come personaggio sono di essere la madre naturale di Cosette e di supplicare Valjean di prendersi cura di sua figlia. Per il resto della storia la sua figura è quasi totalmente assente.
In conclusione, possiamo dire che entrambi i metodi forniscono una valida analisi per questa rete e visto il risultato positivo, nel seguito sperimenteremo il metodo da noi presentato su grafi per i quali non abbiamo possibilità di confronto.

3.2 Rete degli aeroporti

La seconda rete sulla quale ci siamo concentrati riguarda i collegamenti tra aeroporti dislocati in tutto il territorio degli Stati Uniti d'America (più due canadesi), scelti seguendo dei criteri stabiliti. Una volta deciso di voler studiare questa rete, il problema principale da risolvere, era quello di stabilire con che criterio scegliere la lista degli aeroporti — nodi — della nostra rete. Ad esempio, non sarebbe stato utile scegliere quelli delle capitali degli stati, in quanto quasi sempre sono città poco significative e di conseguenza poco utili per verificare l'efficacia del nostro metodo.

3.2.1 Dati raccolti

Abbiamo fatto riferimento alla mappa *USA railway map* disponibile sul sito Ontheworldmap.com che riporta gli itinerari ferroviari forniti da Amtrak, la società nazionale per passeggeri ferroviari.
Forse al lettore verrà spontaneo chiedersi perché non aver optato direttamente per lo studio della rete ferroviaria; ebbene, questo per due motivi principali: primo, molte città importanti, ad esempio Boston o Miami, hanno grado basso in quella figura; secondo, avremmo dovuto inserire le città presenti ai bivi, che spesso sono davvero poco rilevanti.

Dunque i nodi, in totale 44, scelti per la nostra rete sono le città riportate in maiuscolo nella suddetta mappa, che per semplicità abbiamo individuato con degli indicatori sulla più comune mappa fornita da Goolge Maps (Figura 3.3, Mappa interattiva):

nodi				
Albuquerque	Atlanta	Baltimore	Boston	Buffalo
Cheyenne	Chicago	Cincinnati	Cleveland	Dallas
Denver	Detroit	El Paso	Houston	Indianapolis
Jacksonville	Kansas City	Las Vegas	Los Angeles	Memphis
Miami	Milwaukee	Mobile	Montreal	New Orleans
New York	Oakland/Emeryville	Philadelphia	Phoenix	Pittsburgh
Portland	Sacramento	Salt Lake City	San Antonio	San Diego
San Francisco	Seattle	St. Louis	St. Paul-Minneapolis	Tampa
Toronto	Tucson	Vancouver	Washington	

Figura 3.3: Mappa delle città USA (Mappa interattiva) considerate come nodi dei grafi presi in esame.

A queste città abbiamo associato il loro principale aeroporto scelto con i seguenti criteri:

- se una città ha un aeroporto internazionale, scegliamo quello
- se una città ha più aeroporti internazionali, consideriamo i dati raccolti da tutti questi come dati provenienti da un unico aeroporto
- se una città non ha aeroporti internazionali, si considera come principale l'aeroporto presente
- se una città non ha aeroporti, si considera il più vicino

Le città per le quali sono stati considerati più aeroporti internazionali sono New York con tre aeroporti, Buffalo e Miami con due. Invece la città per cui è stato considerato l'aeroporto più vicino è Cheyenne.

Dati i nodi, bisogna esplicitare i lati e i pesi su di essi. Abbiamo scelto di considerare tre funzioni peso:

1. frequenza dei voli settimanali tra due aeroporti;

2. se previsto almeno un volo, distanza tra due aeroporti;

3. prodotto tra i due pesi precedenti.

Frequenza dei voli settimanali tra due aeroporti

Con questa funzione peso due città sono state collegate da un lato se tra loro era previsto almeno un volo in una settimana.
I dati relativi ai voli sono stati raccolti usando il motore di ricerca Google Flights e il codice IATA (International Air Transport Association) di ciascun aeroporto. Per avere uniformità sui dati si è scelta come riferimento una settimana specifica, quella dall'11 al 17 giugno 2018. Impostando come filtri i voli diretti e la classe di volo Business, è stata compilata con Excel una matrice completa, contenente il numero di voli settimanali di andata e ritorno tra due città.
Nel caso di più aeroporti per una stessa città si è considerato come peso il numero totale di voli settimanali; invece se tra due aeroporti non erano previsti voli, si è lasciata un'entrata vuota.
Associando ad ogni valore un colore, è stato possibile visualizzare la matrice come nella Figura 3.4. Come si può notare dalla matrice, intuitivamente la città di New York dovrebbe risultare tra gli hub quando applichiamo al grafo questa funzione peso. Vedremo se sarà così.

Distanza tra due aeroporti se previsto almeno un volo

Per calcolare la distanza tra due aeroporti, abbiamo fatto riferimento al sito Prokerala.com, in cui inserendo i due codici IATA è possibile ottenere la distanza in miglia o in chilometri.
Per una maggiore completezza dei dati, si è scelto di compilare una matrice in cui sono presenti anche le distanze tra aeroporti tra i quali non sono previsti voli. Quello che abbiamo ottenuto è una matrice triangolare superiore con le distanze calcolate in chilometri. Tuttavia i lati ai quali è stato assegnato come peso la distanza, sono gli stessi presi in considerazione per la prima funzione peso: due città sono state collegate da un lato se tra loro era previsto almeno un volo e, a quel lato, è stato assegnato come peso la distanza tra gli aeroporti associati alle città.
Anche in questo caso è possibile visualizzare la matrice come nella Figura 3.5.
Come si può notare dalla figura i dati raccolti sono molto eterogenei, di conseguenza non possiamo evidenziare dalla matrice gli eventuali hub.

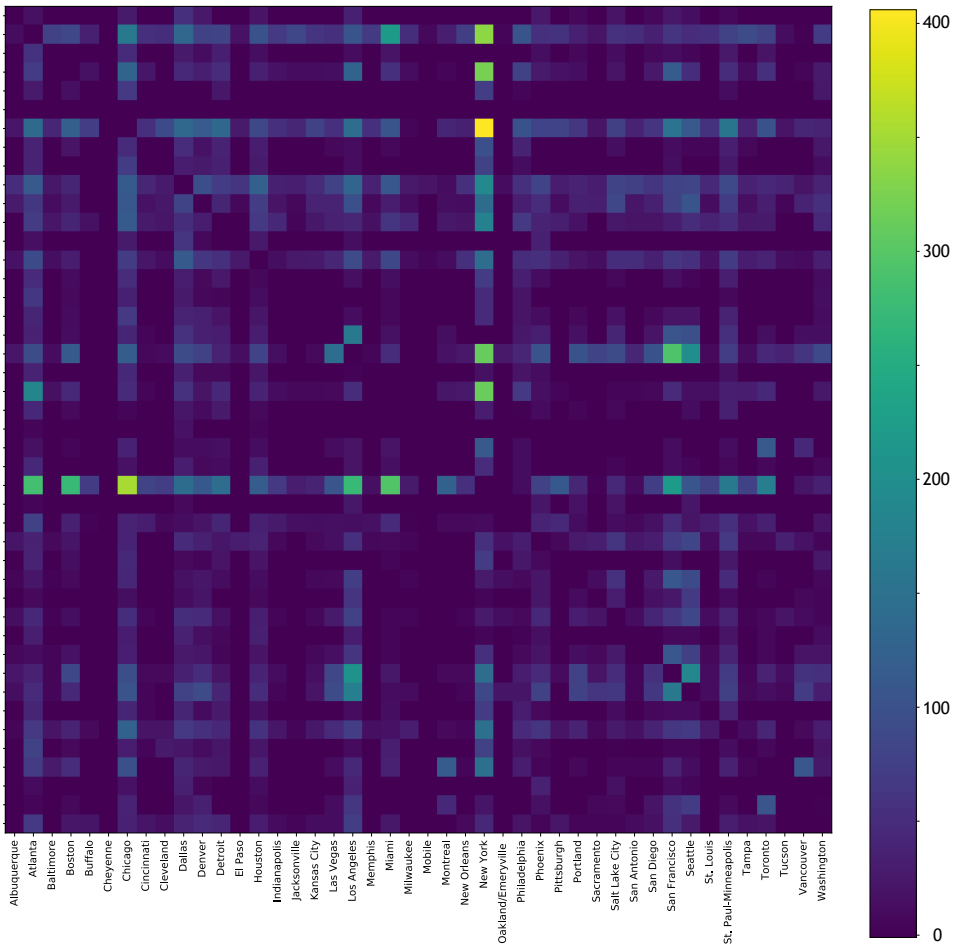

Figura 3.4: Prima matrice dei pesi. Matrice completa con valori il numero di voli settimanali tra due città. Il peso assegnato ad un lato del grafo sarà la somma degli elementi a_{ij} e a_{ji}.

Prodotto tra frequenza dei voli settimanali e distanza tra due aeroporti

Date le due funzioni peso precedenti, abbiamo scelto di considerare una terza funzione peso: se tra due città è previsto almeno un volo, queste vengono collegate da un lato a cui assegniamo come peso il prodotto tra il numero di voli settimanali previsti e la distanza tra i due aeroporti.

3.2.2 Metodo

I dati raccolti sono dunque riassumibili come tre funzioni peso $f, d, \pi : E \to \mathbb{N}$ rispettivamente frequenze dei voli, distanza tra gli aeroporti e la funzione prodotto tra le prime due, che associano pesi differenti ai lati E dello stesso grafo G.

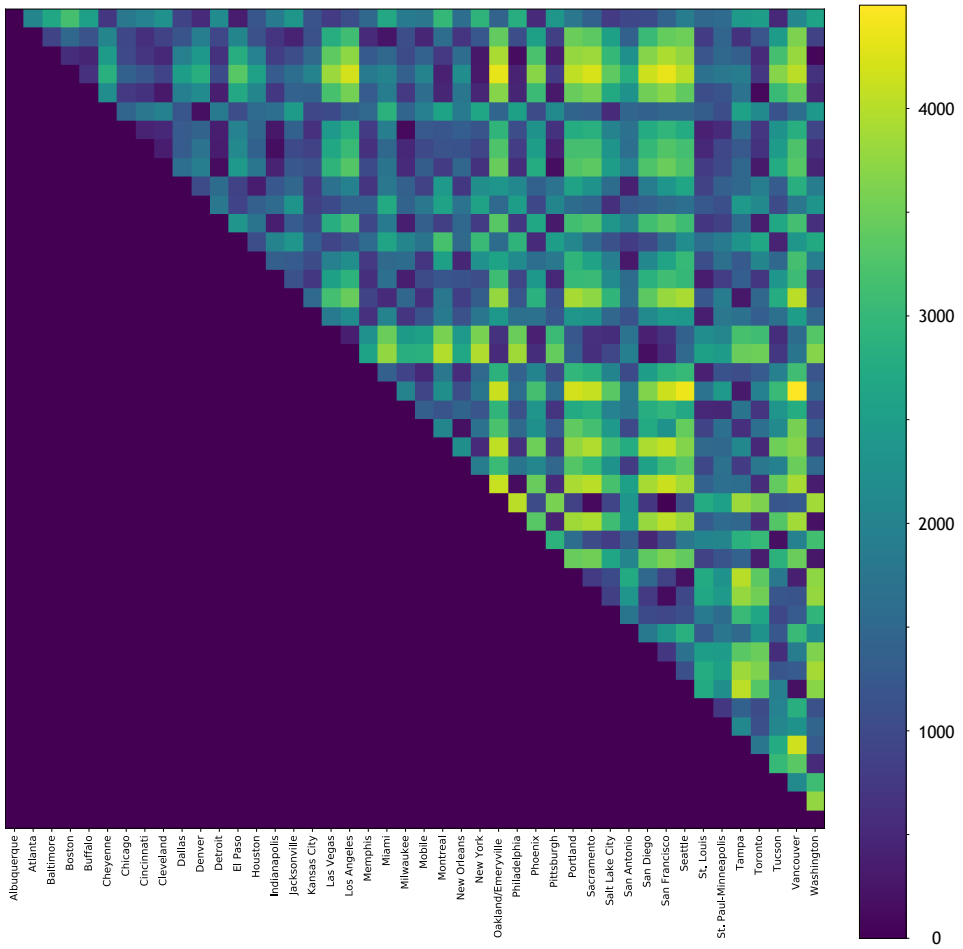

Figura 3.5: Seconda matrice dei pesi. Matrice triangolare superiore con valori le distanze tra due aeroporti . Se tra due città è previsto almeno un volo, il peso assegnato al lato del grafo che le collega sarà il corrispondente valore della distanza tra i loro aeroporti.

Vogliamo calcolare e discutere eventuali hubs nei grafi ottenuti considerando $G_1 = (V, d(E))$, $G_2 = (V, f(E))$ e $G_3 = (V, \pi(E))$. Procederemo considerando sia la definizione originale di hub temporaneo data nella sezione 2.2 che una definizione più lasca ottenuta sostituendo il minore stretto nella formula con un minore o uguale. Infatti pensiamo che la sola condizione di minore stretto possa oscurare uno o più hub lungo la filtrazione facendo così perdere risultati importanti per le nostre applicazioni. Tuttavia questo non è sempre vero, motivo per cui scegliamo di studiare la condizione di minore uguale come variante.
Inoltre, dato l'insieme dei pesi $W = \{w_{ij}\}$, prenderemo in considerazione due funzioni filtranti per costruire le filtrazioni dei nostri grafi: la funzione identità $f(w_{ij}) = w_{ij}$ e la funzione $f(w_{ij}) = max(w) - w_{ij}$. Una volta calcolati gli hubs andremo a evidenziare

Grafi studiati		
$(G_1, <, identità)$ (3.2.4)	$(G_2, <, identità)$ (3.2.4)	$(G_3, <, identità)$ (3.2.4)
$(G_1, \leq, identità)$ (3.2.4)	$(G_2, \leq, identità)$ (3.2.4)	$(G_3, \leq, identità)$ (3.2.4)
$(G_1, <, max\_)$ (3.2.4)	$(G_2, <, max\_)$ (3.2.4)	$(G_3, <, max\_)$ (3.2.4)
$(G_1, \leq, max\_)$ (3.2.4)	$(G_2, \leq, max\_)$ (3.2.4)	$(G_3, \leq, max\_)$ (3.2.4)

Tabella 3.6: In tabella sono riportati i grafi studiati nella sezione 3.2.4. $G_1 = (V, d(E))$, $G_2 = (V, f(E))$ e $G_3 = (V, \pi(E))$, dunque ogni terna indica a quale funzione peso, definizione di hub temporaneo e funzione filtrante ci stiamo riferendo.

quelli di grande persistenza applicando il metodo descritto in 2.3.

Nel seguito utilizzeremo le notazioni $(G_i, <, identità)$, $(G_i, \leq, identità)$, $(G_i, <, max\_)$ e $(G_i, \leq, max\_)$ per indicare a quale funzione peso, definizione di hub temporaneo e funzione filtrante ci stiamo riferendo. Tutti i grafi studiati sono riportati nella Tabella 3.6.

3.2.3 Implementazione

L'implementazione di questi algoritmi è stata svolta in Python. Le matrici sopracitate sono state compilate con fogli Excel e poi salvate come file CSV. Per la lettura di questi abbiamo implementato due funzioni, una per ogni matrice, che riportiamo nel seguito. Per la gestione dei dataframe utilizziamo la libreria Pandas di Python rinominata *pd* nelle porzioni di codice che mostreremo nel seguito, i.e. 3.4 3.5.

```
def read_csv_distance_matrix(path_to_csv, skip_character =-1):
    d = pd.read_csv(path_to_csv, sep = ",", index_col = 0)
    dd = pd.DataFrame(data = d.values , columns = d.columns,
       index = d.columns)
    dd = dd.fillna(skip_character)
    graph_structure = []

    for pair in itertools.combinations(dd.columns, r = 2):
        if dd[pair[1]][pair[0]] != skip_character:
            graph_structure.append(pair+tuple([dd[pair[1]][
               pair[0]]]))

    return graph_structure
```

Algoritmo 3.4: Lettura della matrice delle distanze e costruzione della struttura del grafo

Questa prima funzione legge il file.csv relativo alla matrice delle distanze e restituisce una matrice in cui le entrate vuote sono state sostituite da dei -1, in quanto non consideriamo pesi negativi nelle nostre applicazioni.
Per individuare un valore, consideriamo tutte le combinazioni senza ripetizione di due elementi scelti tra la lista delle città, e poiché la nostra matrice delle distanze è triangolare superiore, accediamo a tutti i valori relativi al secondo elemento della coppia con $dd[pair[1]]$ e selezioniamo in valore relativo al primo elemento con $dd[pair[1]][pair[0]]$. Non stiamo facendo altro che leggere il valore a_{ji} con $j > i$ della nostra matrice. Se questo valore è diverso da *skip_character*, lo aggiungiamo alla coppia di città ottenendo una terna. Alla fine il nostro grafo avrà la struttura di una lista di terne del tipo *(nodo1, nodo2, peso)*.

La funzione che legge la matrice delle frequenze dei voli settimanali è invece riportata nell'algoritmo 3.5

```
def read_csv_frequencies_matrix(path_to_csv, skip_character =
    -1):
    d = pd.read_csv(path_to_csv, sep = ",", index_col = 0)
    dd = pd.DataFrame(data = d.values , columns = d.columns,
        index = d.columns)
    dd = dd.fillna(skip_character)
    graph_structure = []

    for pair in itertools.combinations(dd.columns, r = 2):
        s = 0
        if dd[pair[0]][pair[1]] != skip_character:
            s += dd[pair[0]][pair[1]]
        if dd[pair[1]][pair[0]] != skip_character:
            s += dd[pair[1]][pair[0]]

        if s > 0:
            graph_structure.append(pair + tuple([s]))

    return graph_structure
```

Algoritmo 3.5: Lettura della matrice delle frequenze da CSV e costruzione della struttura del grafo

Come prima, creiamo una matrice a partire dal file.csv, dopodiché si inizializza una variabile somma s con il valore 0. Nel ciclo for, controlliamo che l'elemento a_{ij} con $i > j$ sia diverso da *skip_character*, e in caso affermativo si aggiunge il suo valore alla somma. Lo stesso facciamo per l'elemento a_{ji}. Alla fine, al lato che collega i due nodi, viene assegnato come peso il numero totale di voli di andata e ritorno previsti in una settimana. Anche in questo caso il grafo che consideriamo avrà la struttura di una lista di terne $(nodo1, nodo2, peso)$

Poiché abbiamo tre funzioni peso avremo tre grafi da analizzare: i nodi, 44, e i lati, 508, non cambieranno; gli unici a cambiare saranno i pesi assegnati ai lati. Nel seguente frammento di codice (3.6) vediamo in che modo abbiamo costruito i tre grafi presi in considerazione nelle nostre applicazioni a partire dalle due funzioni appena definite.

```
path_to_csv_1 = '../hub_in_reti/MATRICE DISTANZE.csv'
path_to_csv_2 = '../hub_in_reti/MATRICE FREQUENZE.csv'
graph_structure = read_csv_distance_matrix(path_to_csv_1)
graph_structure_2 = read_csv_frequencies_matrix(path_to_csv_2)


graph_structure_1 = []

for j in graph_structure:
    for i in graph_structure_2:
        if i[0] == j[0] and i[1] == j[1]:
            graph_structure_1.append((j[0], j[1], j[2]))


graph_structure_3 = []

for i in graph_structure_2:
    for j in graph_structure:
        if i[0] == j[0] and i[1] == j[1]:
            graph_structure_3.append((i[0], i[1], i[2]*j[2]))
```

Algoritmo 3.6: Costruzione dei tre grafi usati nelle applicazioni

Il primo grafo su cui lavoriamo è rappresentato dalla lista di terne *graph_structure_1*. Costruiamo due liste richiamando le funzioni sopra definite e poi scegliamo come terne del nostro primo grafo quelle per cui i primi due elementi, in ordine, sono presenti in entrambe le liste costruite, e il terzo elemento è il valore della distanza. Questa non è

altro che l'implementazione del fatto che consideriamo come peso la distanza solo se tra due città è previsto almeno un volo.
Il secondo grafo su cui lavoriamo è *graph_structure_2* che otteniamo già leggendo il file.csv relativo alle frequenze dei voli settimanali con la funzione sopra descritta. Infine, il terzo grafo preso in considerazione è *graph_structure_3*. Questo grafo si costruisce in modo analogo al primo, con la differenza che l'ultimo elemento di ogni tripla è il prodotto tra i pesi precedenti.

Nella sezione 2.2.3 abbiamo costruito un esempio per calcolare gli steady e i ranging hub in cui è stata considerata la funzione $f : W \subset \mathbb{R} \longrightarrow \mathbb{R}$ con $f(w_{ij}) = \frac{1}{w_{ij}}$ come funzione filtrante, dove $W = \{w_{ij}\}$ è l'insieme dei pesi dati ai lati del grafo. Per i tre grafi che prendiamo ora in considerazione, useremo due funzioni filtranti: la funzione identità, $f(w_{ij}) = w_{ij}$, e i relativi sottolivelli di ogni peso w_{ij} partendo dal peso minore, e la funzione $f(w_{ij}) = max(w) - w_{ij}$ e i relativi sottolivelli, partendo dal peso maggiore. Inoltre per ogni funzione si è analizzato ogni grafo prendendo come definizione di t-hub sia quella data nella definizione 22 che la sua variante con il $\leq$.

```
def identity(array):
        return array

def max_(array):
    return(max(array) - array)

graph = WeightedGraph(graph_structure_*)
graph.build_graph()
graph.build_filtered_subgraphs(weight_transform = identity,
   sublevel =True)
graph.get_temporary_hubs_along_filtration()
```

Algoritmo 3.7: Filtrazione del grafo usando le funzioni identità o (max(pesi)- peso) e i sottolivelli

A questo punto non resta che calcolare i nostri steady e ranging hub considerando questi grafi come oggetti della classe *WeightedGraph* che, insieme alle altre funzioni visibili nel frammento di codice 3.8, implementa quanto descritto nel Capitolo 2.

```
above_max_diagonal_gap = False

# Steady
graph.steady_hubs_persistence(above_max_diagonal_gap =
   above_max_diagonal_gap , gap_number = 0)
```

```
fig, ax = plt.subplots()
graph.steady_pd.plot_gudhi(ax,
        persistence_to_plot = graph.steady_pd.
           persistence_to_plot)
if hasattr(graph.steady_pd, 'proper_cornerpoints_above_gap'):
    graph.steady_pd.plot_nth_widest_gap(ax_handle =ax, n =
       graph.gap_number)
    graph.steady_pd.mark_points_above_diagonal_gaps(ax)
plt.savefig('steady.pdf')
print('steady hubs:', [c.vertex for c in graph.
   steady_cornerpoints])
print ('steady hubs above gap:', [(c.birth, c.death, c.vertex)
    for c in graph.steady_pd.proper_cornerpoints_above_gap])



# Ranging
graph.ranging_hubs_persistence(above_max_diagonal_gap =
   above_max_diagonal_gap , gap_number = 0)
fig, ax = plt.subplots()
graph.ranging_pd.plot_gudhi(ax,
        persistence_to_plot = graph.ranging_pd.
           persistence_to_plot)
if hasattr(graph.ranging_pd, 'proper_cornerpoints_above_gap'):
    graph.ranging_pd.plot_nth_widest_gap(ax_handle =ax, n =
       graph.gap_number)
    graph.ranging_pd.mark_points_above_diagonal_gaps(ax)
plt.savefig('ranging.pdf')
print('ranging hubs:', [c.vertex for c in graph.
   ranging_cornerpoints])
print ('ranging hubs above gap:', [(c.birth, c.death, c.vertex
   ) for c in graph.ranging_pd.proper_cornerpoints_above_gap])
```

Algoritmo 3.8: codice che visualizza i diagrammi di persistenza di steady e ranging hub con e senza il metodo descritto nella sezione 2.3 e stampa i nomi delle città che risultano hub

La prima riga di questo frammento di codice ci permette di applicare o meno il metodo descritto nella sezione 2.3: con *above_max_diagonal_gap = False* scegliamo di non applicare il metodo e pertanto il codice ci restituirà la lista di tutti gli steady e ranging hub. Ponendo *above_max_diagonal_gap = True* scegliamo di applicare il metodo e l'argomento *gap_number* ci permette di selezionare il gap che preferiamo. *gap_number=0* prende in esame il primo gap più largo, *gap_number=1* il secondo più largo e così via.

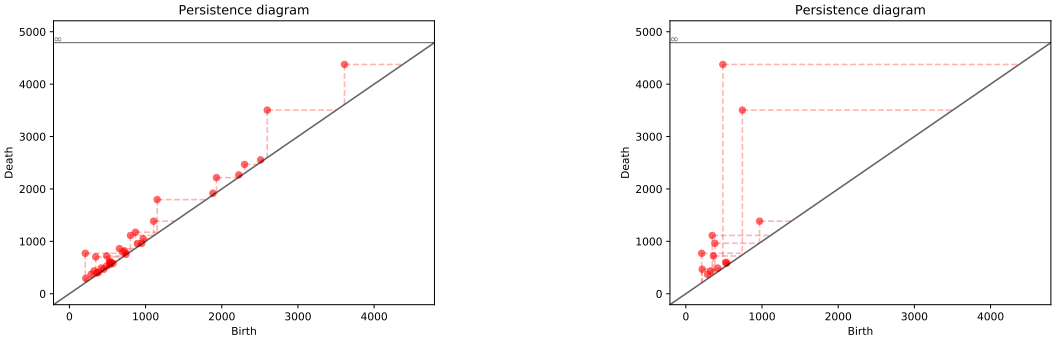

Figura 3.6: Diagrammi di persistenza per steady e ranging hub per $(G_1, <, \textit{identità})$

Steady hub							
34 cornerpoint 13 vertici							
Atlanta	6	Chicago	1	Dallas	8	Detroit	3
Houston	2	Las Vegas	1	Los Angeles	4	Miami	1
New York	2	Philadelphia	2	Phoenix	1	Salt Lake City	2
Seattle	1						

Tabella 3.7: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_1, <, \textit{identità})$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

3.2.4 Risultati

$(G_1, <, \textit{identità})$

Lavoriamo con $(G_1, <, \textit{identità})$, cioè *graph_structure_1* con la condizione di $<$ nella definizione di t-hub, prima senza applicare il metodo descritto nella sezione 2.3. La prima funzione filtrante che utilizziamo è l'identità e consideriamo i sottolivelli della filtrazione. Questo significa che i primi lati a comparire sono quelli con peso minore, fino ad arrivare al peso con lato maggiore.
Quello che otteniamo è un diagramma di persistenza per gli steady hub con 34 cornerpoint e uno per i ranging hub con 13 cornerpoint. Ad ogni cornerpoint è associato l'insieme dei vertici del grafo che sono steady o ranging hub per la coppia di valori data dalle coordinate del cornerpoint. I vertici che risultano hub sono riportati nelle tabelle.
Ci chiediamo prima di tutto se i risultati siano coerenti con la teoria introdotta nel capitolo 2, in un secondo momento invece cercheremo di interpretare quei risultati rispetto all'applicazione specifica che stiamo considerando, discutendo alcune varianti in termini di scelta dei parametri dell'algoritmo.

Considerando le matrici rappresentate nelle figure 3.5 e 3.4, i risultati ottenuti sem-

Ranging hub			
13 cornerpoint 13 vertici			
Atlanta	Chicago	Dallas	Detroit
Houston	Las Vegas	Los Angeles	Miami
New York	Philadelphia	Phoenix	Salt Lake City
Seattle			

Tabella 3.8: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_1, <, identità)$.

Steady hub					
4 cornerpoint 3 vertici					
Atlanta	2	Dallas	1	Seattle	1

Tabella 3.9: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.7 del diagramma di persistenza per steady hub relativo a $(G_1, <, identità)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

brano essere molto validi (Tabelle 3.7 3.8). Infatti le città che risultano hub sono collegate a quasi ogni altra con distanze che variano molto. Dunque queste città potranno risultare t-hub sia nei primi sottografi della filtrazione, sia nei sottografi successivi, il che le rende hub almeno da un certo peso in poi o a intermittenza. Potevamo aspettarci città come Baltimore, Washington e St. Louis che sono le città i cui lati nascono per primi lungo la filtrazione, ma poiché sono collegate ad un numero ristretto di città, prima che il loro grado aumenti, le città precedenti hanno avuto modo di aumentare il loro grado. Dunque questi primi risultati ci sembrano coerenti.
Considerato l'elevato numero di cornerpoint nel diagramma per gli steady hub (Figura 3.6), applichiamo il metodo descritto nella sezione 2.3 per capire quali tra questi 13 vertici sono i più rilevanti. Selezionando i cornerpoint con lo 0-esimo gap diagonale più ampio, vengono evidenziati 4 cornerpoint ai quali sono associati 3 vertici (Tabella 3.9) Atlanta, Dallas e Seattle. Tra tutti gli hub considerati, Atlanta e Dallas sono sicuramente giustificate ad essere tra i selezionati del metodo per i motivi detti. È un po' più strana la presenza di Seattle. Tuttavia Seattle è associata al cornerpoint più in basso, meno persistente e hub per distanze minori. Considerando questo la presenza di Seattle non sembra più strana, infatti tra le città a lui vicine possiamo sicuramente considerarla la più centrale. Atlanta invece è associata ai due cornerpoint centrali della Figura 3.7, proprio ad evidenziare la centralità di Atlanta per medie e grandi distanze.

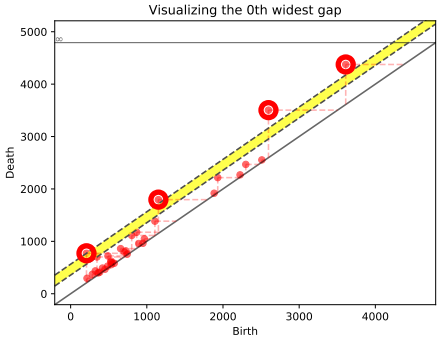

Figura 3.7: Dato il diagramma di persistenza per gli steady hub, ottenuto da $(G_1, <, identità)$, ne selezioniamo i cornerpoint più persistenti con lo 0-esimo gap più ampio.

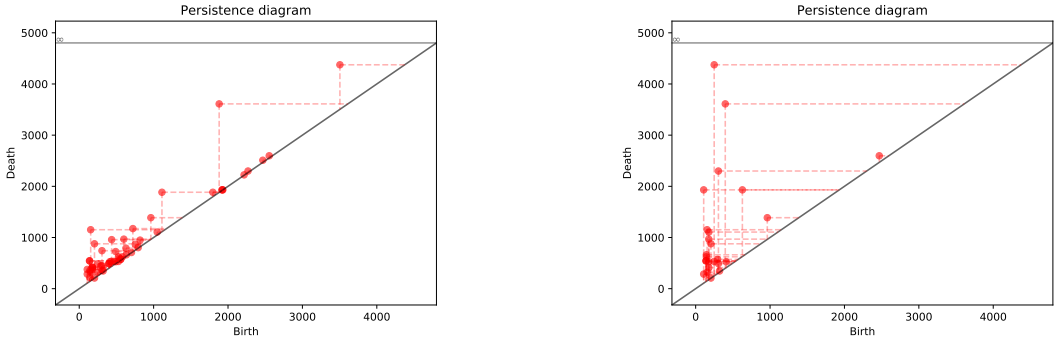

Figura 3.8: Diagrammi di persistenza per steady e ranging hub per $(G_1, \leq, identità)$

$(G_1, \leq, identità)$

Analizziamo ora $(G_1, \leq, identità)$, cioè lo stesso grafo con la stessa funzione filtrante ma con la condizione di $\leq$ nella definizione di vertice t-hub. Tale condizione non stravolge i risultati (Tabelle 3.11 3.10), infatti i vertici risultati prima come hub restano, ma se ne aggiungono di nuovi. Questo perché se nella filtrazione ci sono due vertici legati da un lato con lo stesso grado, nessuno dei due risulterebbe t-hub senza questa condizione. Così invece risultano entrambi hub e di conseguenza in numero di vertici associati ai cornerpoint di un diagramma cresce. In questo caso siamo passati da 13 a 25 vertici hub. I cornerpoint che compaiono come conseguenza della modifica della definizione di hub temporaneo sembrano (Figura 3.8), in questo caso, concentrarsi intorno alla diagonale, utilizzando sia funzioni di persistenza steady che ranging. Questo ci suggerisce due cose: primo, le città in più risultano steady o ranging hub per pochi sottografi della filtrazione; secondo, nel caso precedente sono state oscurate da vertici che avevano caratteristiche simili a loro come ad esempio grado o ordine di nascita dei lati che li congiungono a città comuni. Dato l'elevato numero di vertici in gioco, risulta difficile fare un confronto

Steady hub							
56 cornerpoint 25 vertici							
Atlanta	5	Chicago	5	Cleveland	1	Dallas	3
Denver	2	Detroit	1	Houston	4	Las Vegas	1
Los Angeles	3	Miami	2	Milwaukee	1	New York	3
Philadelphia	5	Phoenix	4	Pittsburg	1	Sacramento	1
St. Paul-Minneapolis	2	Salt Lake City	1	San Antonio	2	San Diego	2
San Francisco	1	Seattle	1	Tucson	1	Vancouver	1
Washington	3						

Tabella 3.10: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_1, \leq, identità)$ I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

Ranging hub				
25 cornerpoint 25 vertici				
Atlanta	Chicago	Cleveland	Dallas	Denver
Detroit	Houston	Las Vegas	Los Angeles	Miami
Milwaukee	New York	Philadelphia	Phoenix	Pittsburg
Sacramento	St. Paul-Minneapolis	Salt Lake City	San Antonio	San Diego
San Francisco	Seattle	Tucson	Vancouver	Washington

Tabella 3.11: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_1, \leq, identità)$

tra tutti guardando le rappresentazioni delle matrici dei pesi nelle figure 3.5 e 3.4. Ad esempio, però, analizzando le due matrici si può intuire che la città di San Francisco e quella di New York probabilmente si oscuravano a vicenda: infatti le città alle quali sono collegate sono quasi tutte in comune e dalla matrice 3.5 si vede chiaramente che hanno valori di distanze molto simili.
Ricorriamo anche qui al metodo descritto nella sezione 2.3 per evidenziare i veri hub. I risultati ottenuti (Tabelle 3.12 3.13) sono simili a quelli precedenti e soprattutto non vengono evidenziati cornerpoint ai quali sono associate città nuove. Questo vuole forse significare che, nel caso che stiamo studiando, la condizione di $\leq$ aggiunge sì qualcosa ma che questo qualcosa non è fondamentale? Proviamo a capirlo studiando gli altri grafi.

$(G_2, <, identità)$

Lavoriamo ora con $(G_2, <, identità)$. Otteniamo un diagramma di persistenza per steady hub con 32 cornerpoint e 14 vertici e uno per ranging hub con 14 cornerpoint e 14 vertici

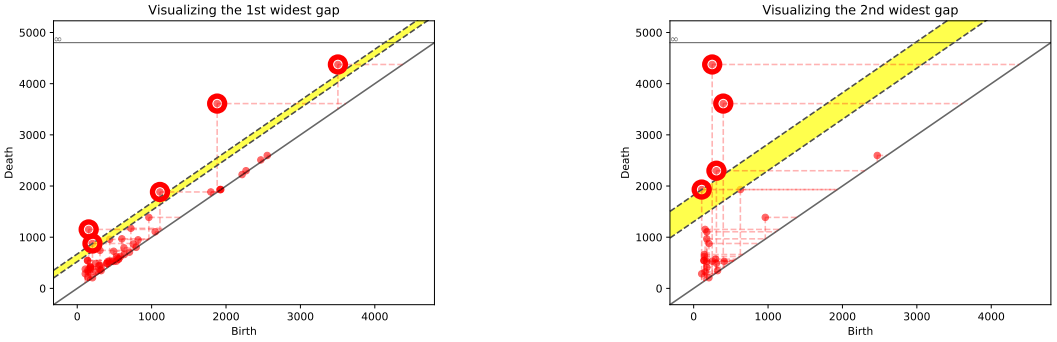

Figura 3.9: Dati i diagrammi di persistenza per steady e ranging hub, ottenuti da $(G_1, \leq, identità)$, ne selezioniamo i cornerpoint rispettivamente con il primo e il secondo gap diagonale più ampio.

Steady hub							
5 cornerpoint 4 vertici							
Atlanta	2	Dallas	1	Detroit	1	Seattle	1

Tabella 3.12: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.9 (a sinistra) del diagramma di persistenza per steady hub relativo a $(G_1, \leq, identità)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

(Figura 3.10). Come per le distanze Molte delle città risultate hub sono giustificate ad esserlo (Tabelle3.15 3.14): Atlanta, Dallas, Houston, Phoenix, Philadelphia, San Francisco, Salt Lake City sono tutte collegate a quasi ogni altra città con frequenze piccole e grandi. Inoltre è interessante vedere come le città che risultano hub sono ben distribuite all'interno di tutto il territorio considerato e a corto raggio possono essere considerate centrali rispetto alle città a loro vicine.
Molti dei cornerpoint sono concentrati nella parte in basso, molto vicino alla diagonale. Questo perché risultano hub per pochi sottografi e per minori frequenze di voli. Ci saremmo aspettati tra gli hub anche le città di Chicago e New York che invece non compaiono. Una possibile risposta può essere che entrambe queste città hanno frequenze

Ranging hub			
4 cornerpoint 4 vertici			
Atlanta	Chicago	Dallas	Houston

Tabella 3.13: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.9 (a destra) del diagramma di persistenza per steady hub relativo a $(G_1, \leq, identità)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

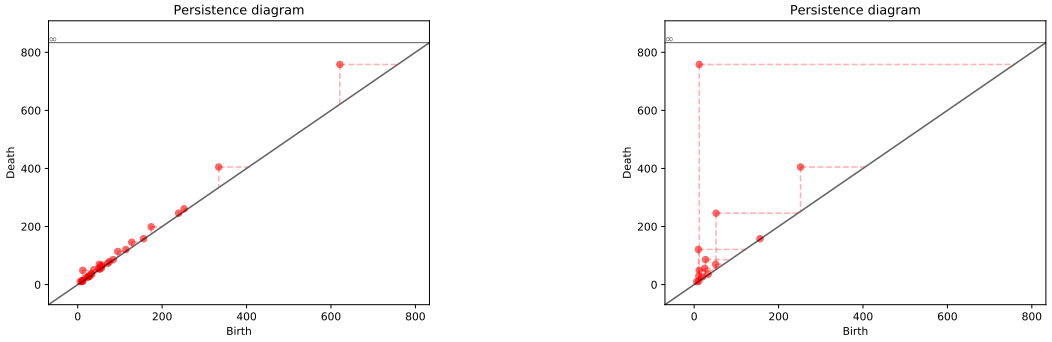

Figura 3.10: Diagrammi di persistenza per steady e ranging hub per $(G_2, <, \textit{identità})$

Steady hub							
32 cornerpoint 14 vertici							
Atlanta	2	Dallas	2	Denver	7	Houston	6
Jacksonville	1	Philadelphia	3	Phoenix	2	Portland	1
St. Paul - Minneapolis	1	Salt Lake City	1	San Francisco	2	Seattle	2
Vancouver	1	Washington	1				

Tabella 3.14: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_2, <, \textit{identità})$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

di voli molto alte, pertanto i loro lati nascono dopo, quando città come Atlanta o Dallas hanno già grado alto e pertanto non riescono ad emergere come hub. Una conferma di questo l'avremo quando studieremo lo stesso grafo con la seconda funzione filtrante. Selezioniamo anche qui i principali cornerpoint con il metodo descritto nella sezione 2.3 considerando il secondo gap più ampio. Otteniamo un diagramma con 3 cornerpoint ai quali sono associati tre vertici (Tabella 3.16) Atlanta, Dallas e Salt Lake City. Le prime due hanno tutto il diritto di comparire come hub in evidenza, infatti se da un lato la caratteristica di essere collegate a quasi ogni altra città le accomuna agli altri hub, dall'altro queste sono le città che hanno frequenze meno omogenee e pertanto il numero di lati che incidono su di esse continua a crescere, rendendole più persistenti. Per quanto riguarda la città di Salt Lake City, non possiamo di certo dire che spicchi per le alte frequenze, sicuramente però è ben collegata alle altre città e con frequenze di voli meno alte. Potremmo dire che questa città è un hub rispetto alle altre città con frequenze di voli medio-basse.

Ranging hub			
14 cornerpoint 14 vertici			
Atlanta	Dallas	Denver	Houston
Jacksonville	Philadelphia	Phoenix	Portland
St. Paul - Minneapolis	Salt Lake City	San Francisco	Seattle
Vancouver	Washington		

Tabella 3.15: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_2, <, identità)$

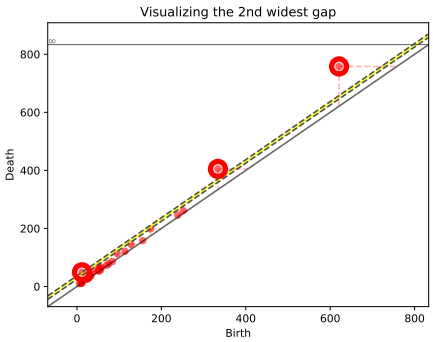

Figura 3.11: Dato il diagramma di persistenza per gli steady hub, ottenuto da $(G_2, <, identità)$, ne selezioniamo i cornerpoint più persistenti con il secondo gap diagonale più ampio.

$(G_2, \leq, identità)$

Come per il primo grafo consideriamo ora la condizione di $\leq$ nella definizione di vertice t-hub e vediamo cosa succede. Anche qui il numero di vertici che risultano hub viene incrementato, passando da 14 a 25. Come mostrano i diagrammi 3.12 compaiono molti più cornerpoint a ridosso della diagonale e analizzando le tabelle 3.17 3.18 e le matrici ci rendiamo conto che compaiono in più soprattutto città con frequenze di voli basse, come St. Paul-Minneapolis, Milwaukee e Memphis. Anche lo studio di questo secondo grafo dunque sembra suggerire che ciò che aggiungiamo con la condizione di $\leq$ non sono

Steady hub		
3 cornerpoint 3 vertici		
Atlanta	Dallas	Salt Lake City

Tabella 3.16: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.11 del diagramma di persistenza per steady hub relativo al grafo graph_structure_2 con funzione filtrante l'identità e condizione di $<$ nella definizione di hub temporaneo.

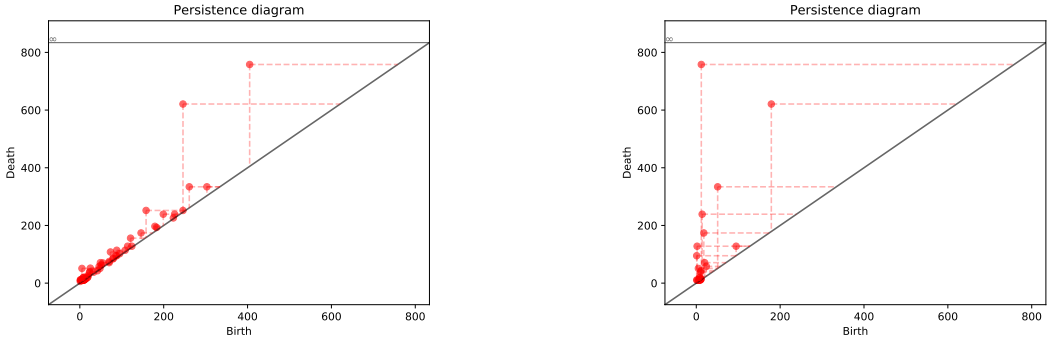

Figura 3.12: Diagrammi di persistenza per steady e ranging hub per $(G_2, \leq, identità)$.

Steady hub							
59 cornerpoint 25 vertici							
Atlanta	6	Boston	1	Dallas	3	Denver	6
Detroit	3	Houston	5	Jacksonville	2	Los Angeles	1
Memphis	1	Miami	1	Milwaukee	1	Montreal	1
New York	1	Philadelphia	5	Phoenix	2	Portland	2
Sacramento	1	St. Paul-Minneapolis	3	Salt Lake City	1	San Antonio	1
San Francisco	2	Seattle	3	Tampa	1	Vancouver	1
Washington	5						

Tabella 3.17: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_2, \leq, identità)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

vertici che poi risultano più persistenti. Selezionare i cornerpoint (Tabelle 3.19 3.20) ci può aiutare anche a capire meglio questa cosa, vediamo come. Consideriamo i cornerpoint al di sopra del primo gap diagonale più ampio per il diagramma degli steady hub e al di sopra del secondo per il diagramma dei ranging hub. In entrambi i casi ai cornerpoint selezionati sono associate città già presenti come hub nello studio dello stesso grafo ma con la condizione di $<$. Anche questi risultati per la matrice delle frequenze trovano un riscontro positivo.

$(G_3, <, identità)$

Analizziamo infine $(G_3, <, identità)$ e $(G_3, \leq, identità)$.
Nel primo caso otteniamo un diagramma di persistenza per steady hub con 43 cornerpoint ai quali sono associati 15 vertici e un diagramma per ranging hub con 15 cornerpoint (Figura 3.14). Ricordando che questo terzo grafo ha come pesi sui lati il prodotto dei

Ranging hub				
25 cornerpoint 25 vertici				
Atlanta	Boston	Dallas	Denver	Detroit
Houston	Jacksonville	Los Angeles	Memphis	Miami
Milwaukee	Montreal	New York	Philadelphia	Phoenix
Portland	Sacramento	St. Paul-Minneapolis	Salt Lake City	San Antonio
San Francisco	Seattle	Tampa	Vancouver	Washington

Tabella 3.18: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_2, \leq, \textit{identità})$.

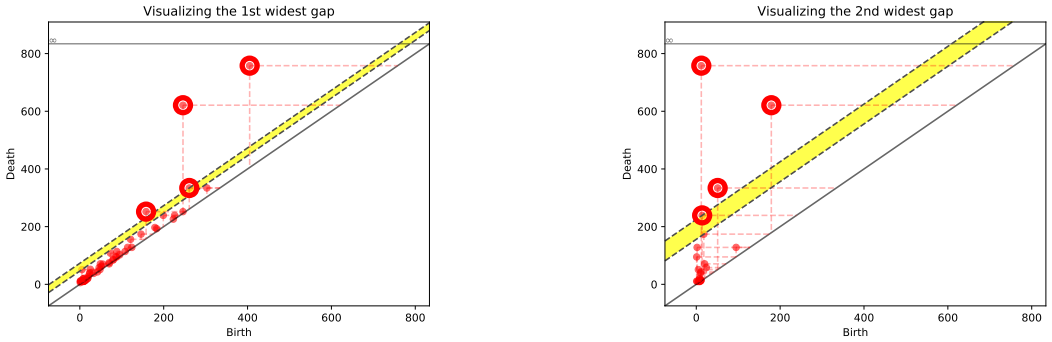

Figura 3.13: Dati i diagrammi di persistenza per steady e ranging hub, ottenuti da $(G_2, \leq, \textit{identità})$, ne selezioniamo i cornerpoint più persistenti rispettivamente con il primo e il secondo gap diagonale più ampio.

pesi precedenti, non sorprende vedere tra i risultanti hub (Tabelle 3.22 3.21) città come Atlanta, Houston, Philadelphia e Phoenix, già hub per i grafi precedenti. Tuttavia anche le altre città compaiono per un giusto motivo: guardando le matrici 3.5 3.4 molte di loro hanno valori di frequenze e distanze medio-basse, pertanto i loro lati sono primi a nascere e sono quelle città che risultano hub per i primi sottografi. Anche qui, visto l'elevato numero di cornerpoint selezioniamo i cornerpoint più rilevanti con il secondo gap più largo. Otteniamo 5 cornerpoint (figura 3.15 e tabella 3.23), 3 ai quali è associata la città di Atlanta e 2 ai quali è associata la città di Houston. Il risultato è esattamente quello che ci aspettavamo. Inoltre considerando che con questa scelta di gap siamo passati da 43 a 5 cornerpoint e tra questi ci sono tutti e due quelli ai quali è associata la città di Houston e tre dei cinque ai quali è associata Atlanta possiamo dire che queste due città sono *super*-hub.

Steady hub					
4 cornerpoint 3 vertici					
Atlanta	2	Dallas	1	Houston	1

Tabella 3.19: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.13 (a destra) del diagramma di persistenza per steady hub relativo a $(G_2, \leq, identità)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata.

Ranging hub			
4 cornerpoint 3 vertici			
Atlanta	Dallas	Denver	Houston

Tabella 3.20: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.13 (a sinistra) del diagramma di persistenza per ranging hub relativo a $(G_2, \leq, identità)$.

$(G_3, \leq, identità)$

Consideriamo ora $(G_3, \leq, identità)$. Otteniamo un diagramma di persistenza per gli steady hub con ben 82 cornerpoint e 32 per i ranging, raddoppiando il numero di vertici che risultano hub. Come si può vedere dai diagrammi di persistenza aumentano i cornerpoint in basso (Figura 3.16), in prossimità della diagonale. Compaiono città come Vancouver che ha valori alti per le distanze ma relativamente bassi per le frequenze. Le città già presenti con la condizione di $<$, oltre ad essere ancora presenti, sono associate ad un numero maggiore di cornerpoint rispetto a prima (Tabelle 3.25 3.24). Selezionando i principali cornerpoint con il secondo gap più ampio per gli steady hub e con il terzo per i ranging hub, otteniamo (Tabelle 3.27 3.26) in entrambi i casi tre vertici principali (Figura 3.17) Altanta, Dallas e Houston. Potevamo sicuramente prevedere Atlanta e Houston; Dallas invece fa eccezione a quanto detto finora. Infatti è tra i vertici comparsi

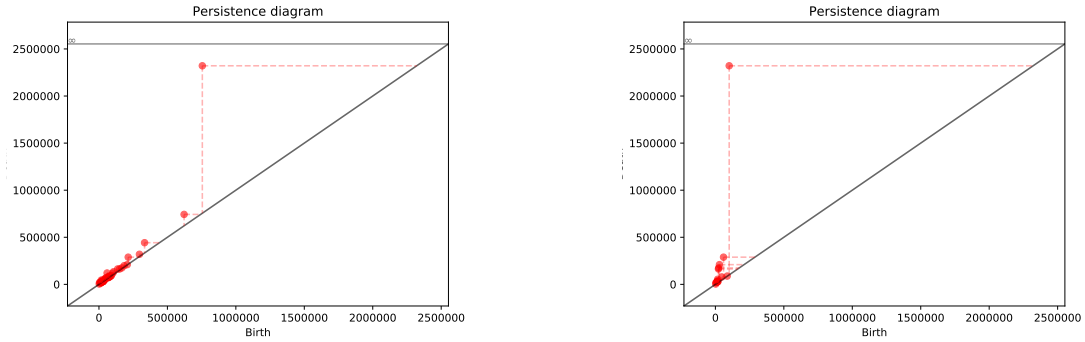

Figura 3.14: Diagrammi di persistenza per steady e ranging hub per $(G_3, <, identità)$.

Steady hub							
43 cornerpoint 15 vertici							
Atlanta	5	Denver	7	Detroit	2	Houston	2
Miami	2	Milwaukee	1	Mobile	1	Philadelphia	1
Phoenix	1	Sacramento	2	St. Paul Minneapolis	3	Salt Lake City	5
San Diego	3	Tucson	2	Washington	6		

Tabella 3.21: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_3, <, \textit{identità})$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

Ranging hub				
15 cornerpoint 15 vertici				
Atlanta	Denver	Detroit	Houston	Miami
Milwaukee	Mobile	Philadelphia	Phoenix	Sacramento
St. Paul Minneapolis	Salt Lake City	San Diego	Tucson	Washington

Tabella 3.22: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_3, <, \textit{identità})$.

con la condizione di $\leq$.

$(G_1, <, max\_)$

Stiamo analizzando $(G_1, <, max\_)$, il grafo che ha come pesi le distanze se previsto almeno un volo tra due città. Guardando a posteriori la Figura 3.5 della matrice delle distanze, molti degli steady hub ottenuti (Figura 3.18, tabella 3.28), corrispondono alle colonne più chiare e questo dà un significato positivo al risultato. Potremmo essere in dubbio sul fatto che Atlanta, Denver, Houston e Phoenix siano hub guardando la stessa matrice, ma poiché consideriamo le distanze se previsto almeno un volo, tenendo presente anche matrice delle frequenze (Figura 3.4), anche queste città sono giustificate ad

Steady hub			
5 cornerpoint 2 vertici			
Atlanta	3	Houston	2

Tabella 3.23: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.15 del diagramma di persistenza per steady hub relativo a $(G_3, <, \textit{identità})$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata.

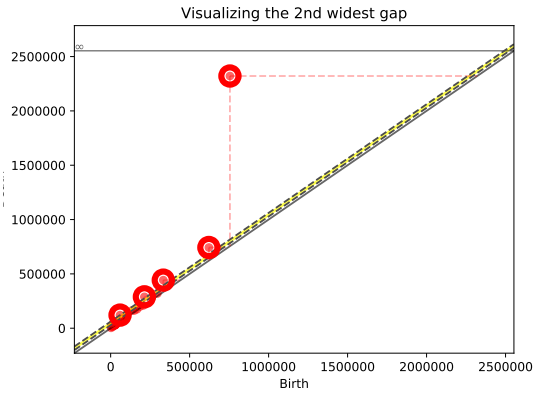

Figura 3.15: Dato il diagramma di persistenza per steady hub, ottenuto da $(G_3, <, identità)$, ne selezioniamo i cornerpoint più persistenti con il secondo gap più ampio.

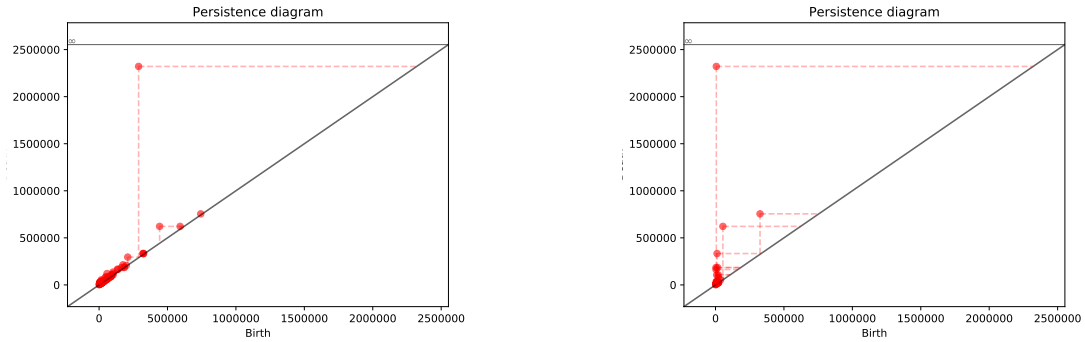

Figura 3.16: Diagrammi di persistenza per steady e ranging hub per $(G_3, \leq, identità)$.

essere hub. Infatti, sono città collegate a quasi ogni altra, di conseguenza su queste città incidono molti lati anche se per valori più bassi della distanza.
Ci sono città, ad esempio Saint Paul Minneapolis che non compaiono come hub pur avendo grado maggiore di alcune città che invece risultano hub. Questo conferma il fatto che per essere un nodo centrale non è sufficiente il grado. Potevamo aspettarci tra gli hub la città di Vancouver, ma analizzando le due matrici e la cartina, notiamo che è molto vicina a Seattle e ha meno edge di questa città e quindi viene oscurata. Altra città che ci saremmo aspettati è quella di New York, collegata con quasi ogni altra città se pur con valori delle distanze non molto alti. Questo, come prima, potrebbe dipendere dal fatto che sia vicina a Boston che ha valori delle distanze più alti, e quindi i suoi lati nascono prima nella filtrazione. Lo stesso argomento si applica per le città di San Antonio e San Diego, molto vicine rispettivamente a Houston e Los Angeles. In definitiva, per questo primo grafo sembra che i risultati siano quelli che, in certi casi non ci aspettavamo, ma che ad una seconda analisi risultano del tutto ragionevoli.

Steady hub							
82 cornerpoint 32 vertici							
Albuquerque	4	Atlanta	3	Boston	1	Chicago	1
Cincinnati	1	Cleveland	1	Dallas	3	Denver	12
Detroit	3	Houston	8	Jacksonville	1	Las Vegas	1
Los Angeles	1	Memphis	1	Miami	2	Milwaukee	1
Mobile	2	Montreal	1	New Orleans	1	Philadelphia	3
Phoenix	4	Sacramento	1	St. Paul-Minneapolis	2	Salt Lake City	8
San Diego	3	San Francisco	1	Seattle	3	St. Louis	1
Tampa	1	Tucson	1	Vancouver	1	Washington	5

Tabella 3.24: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_3, \leq, \textit{identità})$.I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

Ranging hub				
32 cornerpoint 32 vertici				
Albuquerque	Atlanta	Boston	Chicago	Cincinnati
Cleveland	Dallas	Denver	Detroit	Houston
Jacksonville	Las Vegas	Los Angeles	Memphis	Miami
Milwaukee	Mobile	Montreal	New Orleans	Philadelphia
Phoenix	Sacramento	St. Paul-Minneapolis	Salt Lake City	San Diego
San Francisco	Seattle	St. Louis	Tampa	Tucson
Vancouver	Washington			

Tabella 3.25: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_3, \leq, \textit{identità})$.

Avendo un numero considerevole di cornerpoint per gli steady hub possiamo applicare il metodo descritto nella sezione 2.3 per vedere quali vertici associati ai cornerpoint risultano i veri hub. Abbiamo applicato il metodo usando lo 0-esimo gap di ampiezza maggiore e il diagramma relativo è mostrato nella Figura 3.19. Utilizzando il primo gap più ampio otteniamo 3 cornerpoint e 3 vertici (Tabella 3.30). Se visualizziamo i 3 vertici ottenuti nella cartina ci rendiamo conto di due cose principali: primo sono tutte città lontane tra loro e a ovest, secondo sono vicine a molte delle altre che risultano hub ma che non vengono fuori con il metodo descritto nella sezione 2.3.

Il fatto che a risultare hub siano tutte città collocate ad ovest del territorio considerato, conferma da un lato la correttezza dei risultati ottenuti analizzando lo stesso grafo ma con la funzione filtrante *identity*. In quel caso infatti, a risultare come hub

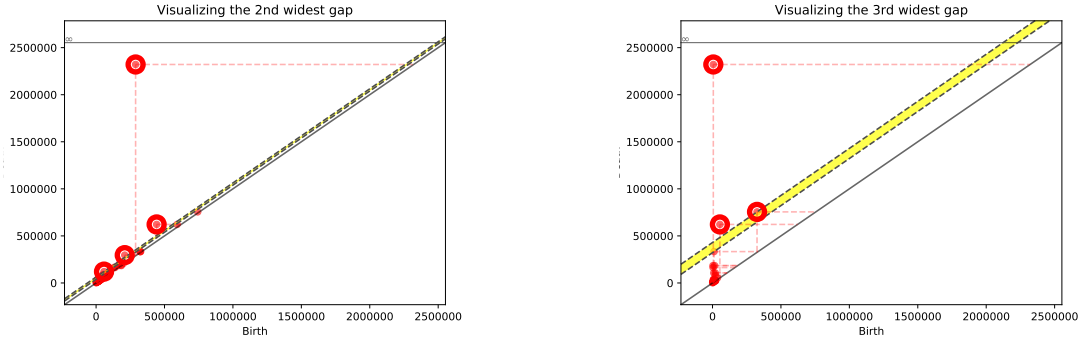

Figura 3.17: Dati i diagrammi di persistenza per steady e ranging hub, ottenuti da $(G_3, \leq, \textit{identità})$, ne selezioniamo i cornerpoint più persistenti rispettivamente con il secondo e il terzo gap più ampio.

Steady hub					
4 cornerpoint 3 vertici					
Atlanta	1	Dallas	1	Houston	2

Tabella 3.26: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.17 (a sinistra) del diagramma di persistenza per steady hub relativo a $(G_3, \leq, \textit{identità})$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata.

erano perlopiù città dislocate nella parte centro-orientale degli USA. Poiché in questa zona sono locati molti degli aeroporti considerati come nodi dei nostri grafi, risultavano hub per distanze minori. Allo stesso tempo, la posizione geografica di Los Angeles, San Francisco e Seattle, giustifica queste città ad essere hub per le distanze più grandi, sono infatti collegate a quasi tutte le città che si trovano ad est e molto lontane da loro. In conclusione, anche applicando il metodo descritto nella sezione 2.3, i risultati ci sembrano più che ragionevoli.

Ranging hub		
3 cornerpoint 3 vertici		
Atlanta	Dallas	Houston

Tabella 3.27: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.17 (a destra) del diagramma di persistenza per ranging hub relativo a $(G_3, \leq, \textit{identità})$.

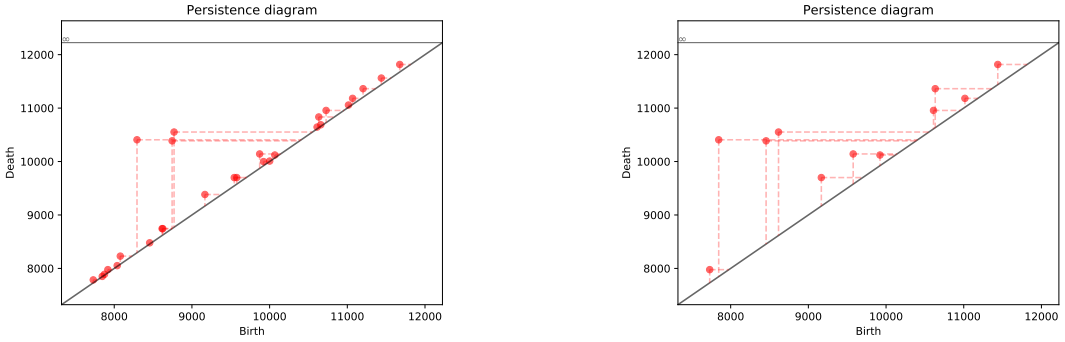

Figura 3.18: Diagrammi di persistenza per steady e ranging hub per $(G_1, <, max\_)$.

Steady hub							
28 cornerpoint 11 vertici							
Atlanta	2	Boston	2	Dallas	2	Denver	2
Houston	3	Los Angeles	3	Phoenix	2	Portland	2
Salt Lake City	3	San Francisco	5	Seattle	2		

Tabella 3.28: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_1, <, max\_)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

$(G_1, \leq, max\_)$

Come si può notare, anche per $(G_1, \leq, max\_)$, vale sempre la regola per cui i vertici che risultano hub sono gli stessi trovati con la condizione di $<$ e in più qualche altro. In questo caso (Tabelle 3.32 3.31), risultano in più come hub le città di Las Vegas, Miami, New York e San Diego. Le città di Las Vegas e San Diego sono molto vicine a quella di Los Angeles e, a parte qualche eccezione, sono collegate alle stesse città. Quelle poche eccezioni sono il motivo per cui con la sola condizione di $<$ nella definizione di t-hub Las Vegas e San Diego non figurano come hub. Come dicevamo nell'analisi precedente di questo grafo, ci

Ranging hub			
11 cornerpoint 11 vertici			
Atlanta	Boston	Dallas	Denver
Houston	Los Angeles	Phoenix	Portland
Salt Lake City	San Francisco	Seattle	

Tabella 3.29: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_1, <, max\_)$.

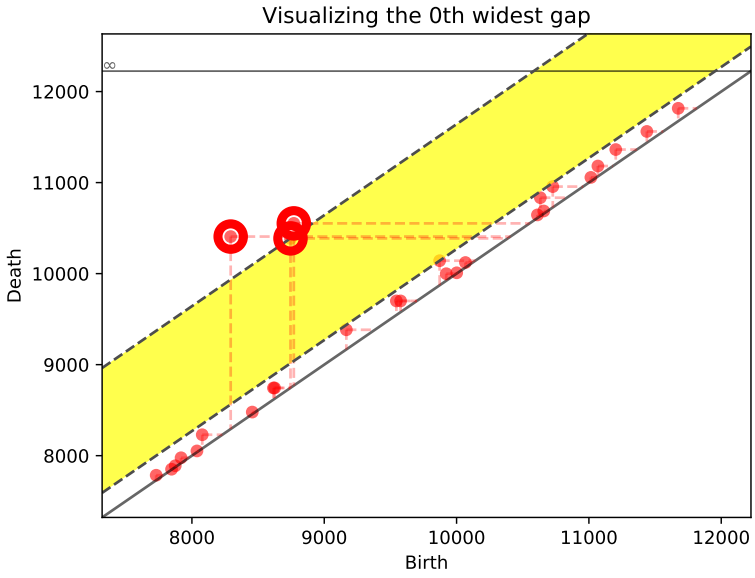

Figura 3.19: Dato il diagramma di persistenza per steady hub, ottenuto da $(G_1, <, max\_)$, ne selezioniamo i cornerpoint più persistenti con lo 0-esimo gap diagonale più ampio.

Steady hub con 0-esimo widest gap					
3 cornerpoint 3 vertici					
Los Angeles	1	San Francisco	1	Seattle	1

Tabella 3.30: In tabella sono riportate le città associate ai cornerpoint evidenziati nella Figura 3.19 del diagramma di persistenza per steady hub relativo a $(G_1, <, max\_)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata

saremmo aspettati la città di New York tra gli hub, ed eccola qui. Questo conferma che senza la condizione di $\leq$ nella definizione, questa città viene oscurata dall'essere hub di Boston. Infine, per la città di Miami, prendendo in esame le due matrici, si vede che le distanze tra Boston e le città con cui è collegata e Miami e le città con cui è collegata sono circa le stesse, ma questa ha meno collegamenti. Possiamo pertanto concludere che anche la città di Miami viene oscurata dall'essere t-hub di Boston con la sola condizione di $<$. In definitiva i risultati ottenuti risultano coerenti con la precedente analisi. Il metodo descritto nella sezione 2.3 applicato a questo grafo, porta alle stesse conclusioni dell'analisi fatta in precedenza, pertanto anche questo fatto conferma una coerenza nei risultati.

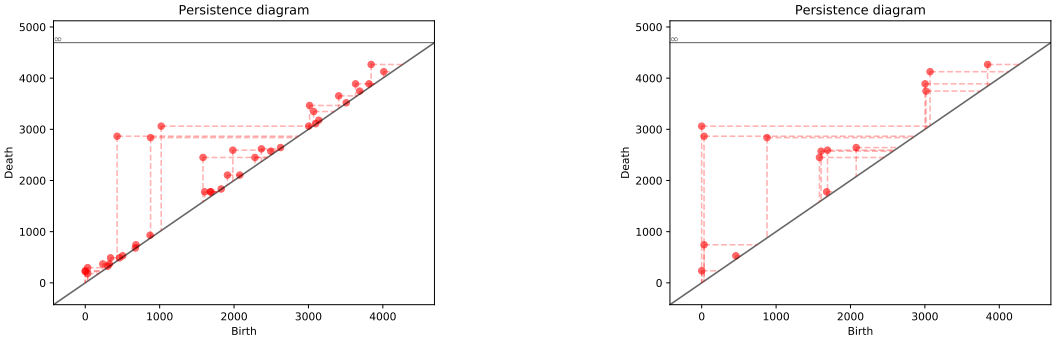

Figura 3.20: Diagrammi di persistenza per steady e ranging hub per $(G_1, \leq, max\_)$.

Steady hub							
40 cornerpoint 15 vertici							
Atlanta	1	Boston	4	Dallas	4	Denver	5
Houston	2	Las Vegas	1	Los Angeles	1	Miami	1
New York	2	Phoenix	3	Portland	1	Salt Lake City	3
San Diego	4	San Francisco	3	Seattle	5		

Tabella 3.31: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_1, \leq, max\_)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata.

$(G_2, <, max\_)$

Lavoriamo ora con il secondo grafo $(G_2, <, max\_)$, che ha come pesi le frequenze settimanali. Facciamo quindi riferimento alla matrice in Figura 3.4 per capire se anche in questo caso i risultati ottenuti assumono un significato ragionevole. Nel diagramma di persistenza per gli steady hub abbiamo 7 cornerpoint e in quello per i ranging hub ne abbiamo 4 (figura 3.21). Risultano hub quattro vertici: Atlanta, Chicago, Dallas, New York (Tabelle 3.34 3.33). Se guardiamo la matrice, queste sono proprio le città che corrispondono alle righe più chiare, i loro lati hanno dunque peso maggiore e nascono prima degli altri.
Come per il primo grafo, guardando la matrice, ci aspetteremmo come hub anche le città di Los Angeles, San Francisco e Seattle, ma non compaiono. La giustificazione di questo è dovuta all'essere steady di New York e Chicago. Infatti quando nascono dei lati per le città che non risultano hub, si aggiungono altri lati anche a New York e a Chicago che essendo già molto forti, continuano ad essere i più forte per quei sottografi. Dunque anche questi risultati ci sembrano ragionevoli.
Visto l'esiguo numero di cornerpoint ottenuto non ci sembra opportuno in questo caso

Ranging hub			
15 cornerpoint 15 vertici			
Atlanta	Boston	Dallas	Denver
Houston	Las Vegas	Los Angeles	Miami
New York	Phoenix	Portland	Salt Lake City
San Diego	San Francisco	Seattle	

Tabella 3.32: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_1, \leq, max\_)$.

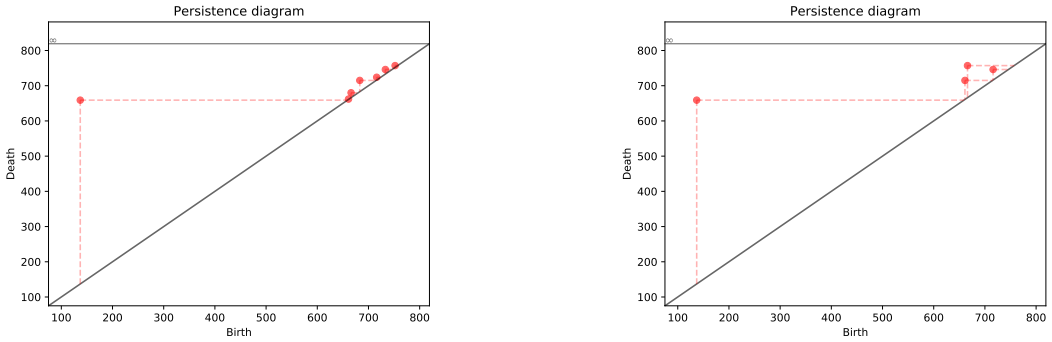

Figura 3.21: Diagrammi di persistenza per $(G_2, <, max\_)$.

applicare il metodo descritto nella sezione 2.3.

$(G_2, \leq, max\_)$

Modificando la condizione di minore stretto nella definizione di t-hub in questo caso viene aggiunto un cornerpoint, ma i vertici associati ai cornerpoint restano gli stessi (Tabelle 3.36 3.35). Chicago è associata ad un cornerpoint in più, Dallas ad uno in meno. Nella precedente analisi di questo grafo, al cornerpoint con maggiore persistenza era associato il

Steady hub							
7 cornerpoint 4 vertici							
Atlanta	2	Chicago	2	Dallas	2	New York	1

Tabella 3.33: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_2, <, max\_)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata.

Ranging hub			
4 cornerpoint 4 vertici			
Atlanta	Chicago	Dallas	New York

Tabella 3.34: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_2, <, max\_)$.

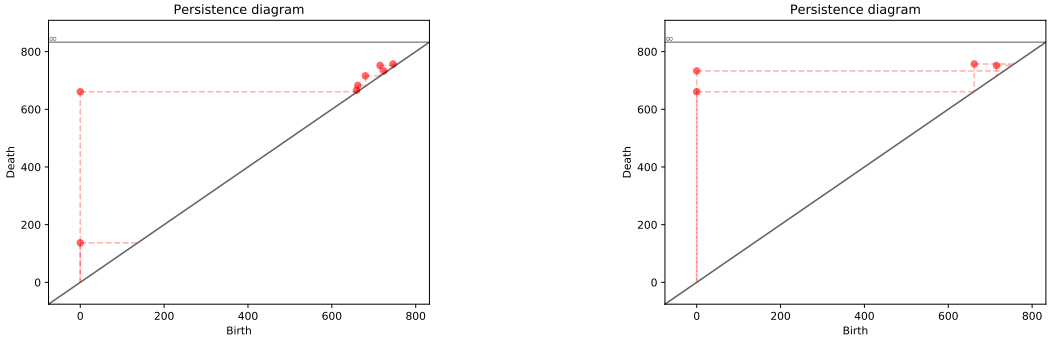

Figura 3.22: Diagrammi di persistenza per steady e ranging hub per $(G_2, \leq, max\_)$.

vertice relativo a New York. In questo caso, ai cornerpoint più persistenti sono associate New York e Chicago. E questo è un ottimo risultato in riferimento ai dati raccolti per la matrice delle frequenze.

$(G_3, <, max\_)$

Per quanto riguarda il terzo grafo analizzato $(G_3, <, max\_)$, con pesi il prodotto dei pesi precedenti, otteniamo 3.23 un diagramma di persistenza per gli steady hub con 10 cornerpoint e un diagramma di persistenza per ranging hub con 5 cornerpoint. Risultano hub cinque città (Tabelle 3.38 3.37): Atlanta, Chicago, Dallas, New York e Vancouver. Ci potevamo sicuramente aspettare tra i risultati le città di Atlanta e Dallas, che risultano hub già per entrambi i grafi precedenti. Anche le città di New York e Chicago non

Steady hub							
8 cornerpoint 4 vertici							
Atlanta	2	Chicago	4	Dallas	1	New York	1

Tabella 3.35: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_2, \leq, max\_)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata.

Ranging hub			
4 cornerpoint 4 vertici			
Atlanta	Chicago	Dallas	New York

Tabella 3.36: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_2, \leq, max\_)$.

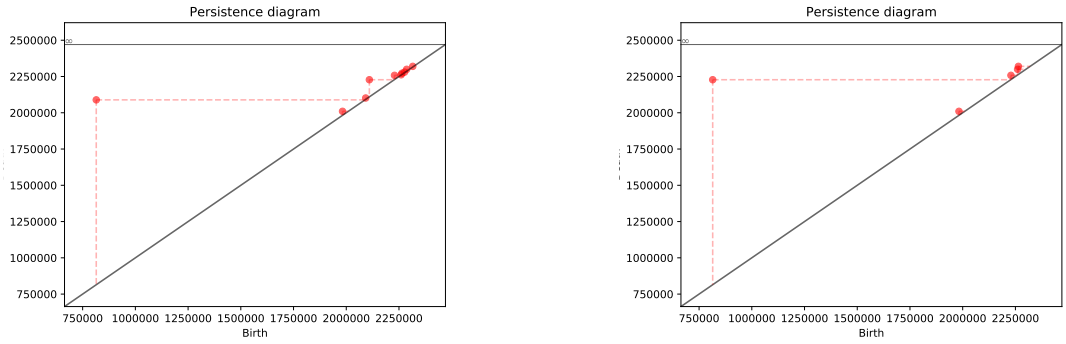

Figura 3.23: Diagrammi di persistenza per steady e ranging hub per $(G_3, <, max\_)$.

sorprendono come risultati, infatti seppur vicini a molti degli altri nodi, hanno una frequenza di voli molto alta oltre che ad essere collegati con quasi ogni altra città, e questo li giustifica ad essere considerati hub. L'unico risultato inaspettato è Vancouver. Questa è molto vicina alla città di Seattle e quindi le loro distanze dalle città a cui sono entrambe collegate, hanno valori molto vicini tra loro. Inoltre la città di Seattle è collegata a molte più città rispetto a Vancouver e con frequenze più alte. Ci saremmo aspettati dunque come hub questa città e non Vancouver. Tuttavia una giustificazione a questo può esserci e cioè, la città di Vancouver è a distanza molto alta dalla città di Toronto, e tra le due ci sono moltissimi voli settimanali. Facendone il prodotto otteniamo un peso che supera molti dei pesi dei lati collegati a Seattle e di conseguenza nasce prima di questi. I pesi successivi riguardano lati che incidono sulle stesse città e poiché, se pur vicini, Vancouver è più lontano tra le due sarà t-hub Vancouver. In definitiva anche

Steady hub									
10 cornerpoint 5 vertici									
Atlanta	2	Chicago	1	Dallas	3	New York	3	Vancouver	1

Tabella 3.37: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_3, <, max\_)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata.

Ranging hub				
5 cornerpoint 5 vertici				
Atlanta	Chicago	Dallas	New York	Vancouver

Tabella 3.38: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_3, <, max\_)$.

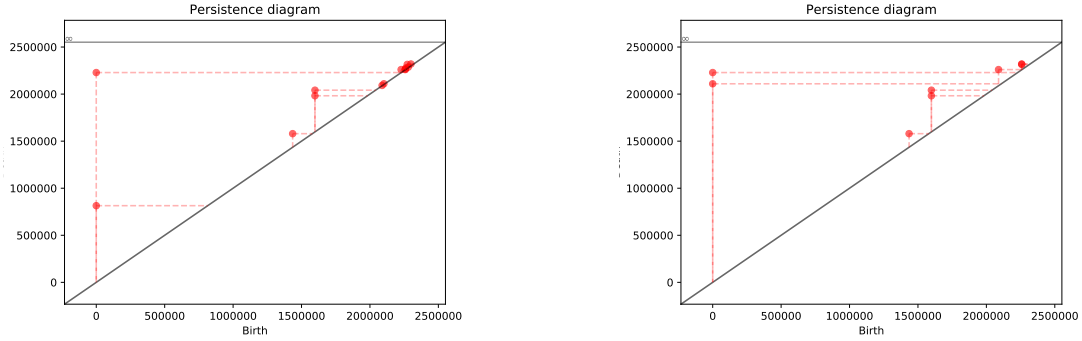

Figura 3.24: Diagrammi di persistenza per steady e ranging hub per $(G_3, \leq, max\_)$.

questo risultato ci sembra ragionevole.

$(G_3, \leq, max\_)$

Analizzando lo stesso grafo ma con la condizione di $\leq$ otteniamo un diagramma di persistenza per steady hub con 4 cornerpoint in più e risultano hub 8 vertici (Tabelle 3.40 3.39). Vengono aggiunte agli hub già trovati le città di Los Angeles, Boston e Toronto. Il fatto che ora compaia Toronto tra gli hub ci dà conferma della giustificazione data in precedenza per la presenza come hub di Vancouver al posto di Seattle. Guardando le due matrici ci si può rendere conto dell'assoluta coerenza tra i risultati della città di Los Angeles. Questa, se pur a distanze maggiori rispetto a Chicago e Dallas, ha frequenze

Steady hub							
14 cornerpoint 8 vertici							
Atlanta	4	Boston	1	Chicago	2	Dallas	2
Los Angeles	2	New York	1	Toronto	1	Vancouver	1

Tabella 3.39: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per steady hub relativo a $(G_3, \leq, max\_)$. I numeri riportati a destra di ogni città indicano il numero di cornerpoint ai quali è associata.

Ranging hub			
8 cornerpoint 8 vertici			
Atlanta	Boston	Chicago	Dallas
Los Angeles	New York	Toronto	Vancouver

Tabella 3.40: In tabella sono riportate le città associate ai cornerpoint del diagramma di persistenza per ranging hub relativo a $(G_3, \leq, max\_)$.

inferiori. Questo vuol dire che i prodotti che vengono fuori tra queste città sono molto simili e pertanto senza la condizione di $\leq$ la città di Los Angeles viene oscurata dalle altre. Infine Boston è assolutamente giustificato a comparire come hub in questo caso. È molto vicino a New York e come questa è collegato a quasi ogni altra città con frequenze simili. Questo fa presumere che in alcuni sottografi siano entrambi t-hub. Tuttavia New York ha alcune frequenze maggiori e di conseguenza i suoi lati nascono prima. In definitiva Possiamo dire che la città di Boston senza la condizione di $\leq$ viene oscurata dalla città di New York e così anche questi ultimi risultati trovano il loro significato positivo.

3.3 Lingue europee

In questa ultima sezione studieremo il grafo che rappresenta le relazioni tra le lingue ufficiali dell'Unione Europea a partire da proprietà comuni.

3.3.1 Dati raccolti

L'insieme dei dati in formato CSV utilizzato per costruire il grafo di questa rete è stato ottenuto dal sito TerraLing.com. Tale sito prende in esame 304 lingue e 165 proprietà grammaticali o fonetiche; per ognuna il valore può essere Vero se la lingua in questione ha quella proprietà o Falso se non l'ha.
Purtroppo per la maggior parte delle lingue il sito in questione non riporta i valori di verità di tutte e 165 le proprietà ma solo una parte. Pertanto nel costruire il grafo da noi studiato sono state considerate le lingue dell'Unione Europea, più il turco, che avessero almeno il 50% delle proprietà esaminate, per un totale di 19 lingue (Tabella 3.41).

Sono state compilate con Excel due matrici: la prima ha come elemento (i, j) la somma tra le proprietà vere e le proprietà false per entrambe le lingue i e j; la seconda fornisce il valore della funzione peso per quel lato: il valore è dato da $100 *$ il valore corrispondente nella prima matrice, diviso per la radice del prodotto delle completezze delle due lingue.

Lingue				
Castigliano	Catalano	Ceco	Croato	Danese
Finlandese	Francese	Galiziano	Greco	Inglese
Italiano	Olandese	Polacco	Portoghese	Rumeno
Svedese	Tedesco	Turco	Ungherese	

Tabella 3.41: In tabella sono riportate le lingue europee considerate come nodi del grafo preso in esame.

Per completezza di una lingua si intende la percentuale delle proprietà esaminate rispetto alle 165 totali. Dunque in questo caso la funzione peso è esplicata nella formula

$$f : E \to \mathbb{R}$$
$$e_{i,j} \mapsto 100 * \frac{m_{ij}}{\sqrt{c(l_i)c(l_j)}}$$

dove m_{ij} è la somma tra i numeri delle proprietà vere e delle proprietà false per entrambe le lingue i e j e $c(l_i)$ è la completezza dell'i-esima lingua.
Dato l'insieme dei pesi $W = \{w_{ij}\}$ consideriamo come funzione filtrante la funzione $f(w_{ij}) = max(w) - w_{ij}$; così facendo i primi lati a comparire lungo la filtrazione saranno quelli con peso maggiore.

3.3.2 Implementazione

L'implementazione è stata svolta in Python. Il file CSV contenente la struttura del grafo viene letto da un algoritmo molto simile a 3.4 che restituisce la struttura del grafo come lista di terne del tipo *(lingua1, lingua2, peso)*.

Data la struttura del grafo, ne costruiamo la filtrazione con un algoritmo identico a 3.5 usando la funzione $f : W \subset \mathbb{R} \longrightarrow \mathbb{R}$ con $f(w_{ij}) = max(w) - w_{ij}$ come funzione filtrante, dove $W = \{w_{ij}\}$ è l'insieme dei pesi del grafo.

Calcoliamo infine gli steady e i ranging hub considerando il grafo ottenuto come oggetto della classe *WeightedGraph* che, insieme alle altre funzioni già viste nell'algoritmo 3.8, implementa quanto descritto nel Capitolo 2.

Tale algoritmo oltre a fornire la possibilità di applicare o meno il metodo descritto nella sezione 2.3, fornisce i diagrammi di persistenza per steady e ranging hub e la lista delle lingue che risultano hub.

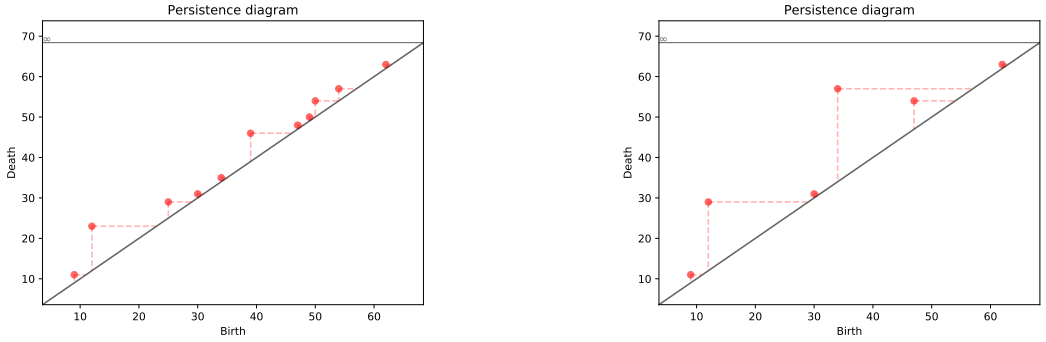

Figura 3.25: Diagrammi di persistenza per steady e ranging hub per il grafo relativo alle lingue europee.

Steady hub					
11 cornerpoint 6 vertici					
Castigliano	1	Catalano	2	Inglese	2
Olandese	1	Portoghese	4	Svedese	1

Tabella 3.42: In tabella sono riportate le lingue associate ai cornerpoint del diagramma di persistenza per steady hub relativo al grafo delle lingue. I numeri riportati a destra di ogni lingua indicano il numero di cornerpoint ai quali è associata.

3.3.3 Risultati ottenuti

I risultati visibili in figura 3.25 e nelle tabelle 3.42 3.43 sono stati ottenuti considerando la condizione di $<$ nella definizione di t-hub e come funzione filtrante la funzione $f(w_{ij}) = max(w) - w_{ij}$. Otteniamo un diagramma di persistenza per gli steady hub con 11 cornerpoint e uno per i ranging hub con 6 cornerpoint. Ad ogni cornerpoint è associato l'insieme dei vertici del grafo che sono steady o ranging hub per la coppia di valori data dalle coordinate del cornerpoint. Come fatto per le altre applicazioni ci chiediamo se i risultati siano coerenti con la teoria introdotta nel capitolo 2, e proveremo a dare un'interpretazione rispetto alla specifica rete considerata.

Ranging hub		
6 cornerpoint 6 vertici		
Castigliano	Catalano	Inglese
Olandese	Portoghese	Svedese

Tabella 3.43: In tabella sono riportate le lingue associate ai cornerpoint del diagramma di persistenza per ranging hub relativo al grafo delle lingue.

Non sorprende vedere tra gli hub l'inglese, seconda lingua più parlata al mondo, e il portoghese, lingua del primo impero coloniale e commerciale d'Europa e anch'essa diffusa in molte aree del mondo. Ci stupiscono sicuramente le altre lingue: svedese, castigliano, catalano e olandese. Bisogna però sottolineare che le nostre poche conoscenze in campo linguistico non ci permettono di stabilire se tali risultati possano sorprendere anche un esperto del settore. La morfologia svedese è molto simile a quella inglese; ciò vuol dire che potrebbe risultare tra gli hub non perché sia realmente importante da un punto di vista di significato ma perché ha molte proprietà in comune con altre lingue. Il catalano è una lingua parlata in varie regioni della Spagna (Catalogna, Isole Baleari, Comunità Valenzana e Frangia d'Aragona), in Andorra e in alcune parti della Francia e dell'Italia. Il castigliano, dopo il cinese mandarino e l'inglese, è la terza lingua più parlata al mondo e tra il 1000 e il 1900 d.C. ha avuto grande diffusione in tutta la penisola iberica. Attualmente è riconosciuta come lingua ufficiale, oltre che in Europa, in molti stati del centro e sud America, negli Stati Uniti e in Africa nord-occidentale. L'olandese è la lingua ufficiale dei Paesi Bassi, del Belgio e del Suriname, ma è parlato in altri 9 stati nel mondo.

Potevamo aspettarci tra gli hub il francese, il tedesco e l'italiano piuttosto che il catalano o lo svedese. Infatti sia l'italiano che il tedesco sono tra le lingue i cui lati a loro incidenti hanno pesi maggiori, indice del fatto che condividono molte proprietà con altre lingue.
In generale non riteniamo che i risultati ottenuti ci soddisfino pienamente. Questo può essere dovuto al fatto che i dati raccolti non siano completi e omogenei: prendere in considerazione le lingue che avessero almeno il 50% delle proprietà esaminate significa che nel compilare la matrice valutiamo anche la somiglianza tra una lingua che ha completezza 50 e una con completezza 98, e questo ci fa pensare che sia lecito ottenere risultati che non ci aspettavamo.

Conclusioni

In questa tesi abbiamo provato a dare una risposta alla domanda *È possibile trovare i centri nevralgici di dati rappresentati sotto forma di un grafo pesato?*
Abbiamo descritto come sia possibile studiare i grafi pesati usando la persistenza. Per fare questo, discostandoci dagli approcci basati sulla rappresentazione del grafo pesato come complesso simpliaciale, abbiamo discusso una teoria che permette di ottenere funzioni filtranti e diagrammi di persistenza direttamente a partire dal grafo stesso e una proprietà propria della teoria dei grafi.

Dopo aver stabilito una classe di funzioni di persistenza adatte all'analisi degli hub in un grafo pesato, abbiamo applicato la teoria e gli algoritmi descritti precedentemente a tre reti. La scelta di queste tre applicazioni è stata guidata dalla volontà di fornire un'intuizione riguardo alla generalità di tale metodo. Abbiamo considerato come prima rete quella che descrive le relazioni tra i personaggi de *Les Misérables*. La seconda rete studia i collegamenti tra aeroporti degli Stati Uniti d'America e la terza prende in esame proprietà comuni o non comuni alle varie lingue ufficiali dell'Unione Europea.

I risultati ottenuti su queste tre reti sono ragionevoli, sopratutto pensando alla semplicità con cui il contesto teorico permette di affrontare una tematica complessa come l'analisi dei centri nevralgici in una rete. Ad ogni modo, sarà necessario in futuro procedere a una validazione del metodo utilizzando un numero maggiore di reti di test e sopratutto tali che non necessitino dell'intervento di esperti perché possano essere interpretate. Si pensi in questo senso a problemi come il clustering. In aggiunta, sarà possibile sperimentare l'effetto di altre proprietà nella definizione di t-hub o funzioni di persistenza differenti.

Bibliografia

[1] B. Rieck, U. Fugacci, J. Lukasczyk, and H. Leitte. Clique community persistence: a topological visual analysis approach for complex networks. *IEEE transactions on visualisation and computer graphics*, 24(1):822-831,2018

[2] M. G. Bergomi, M. Ferri, L. Zuffi. Graph persistence. arXiv:1707.09670v2. 2017

[3] M. Ferri. Persistent topology for natural data analysis - a survey. arXiv:1706.00411. 2017.

[4] S. Fortunato. Community detection in graphs. *Physics Reports, 486(3-5):75-174,2010*

[5] Edwin H Spanier. *Algebraic topology*, volume 55. Springer Science & Business Media, 1994.

[6] V. Kurlin. A fast persistence-based segmentation of noisy 2D clouds with provable guarantees. *Pattern recognition letters*, 83:3-12, 2016.

[7] A. Landherr, B. Friedl, J. Heidemann. A critical review of centrality measures in social networks. *Business & Information System Engineering, 2(6):371-385, 2010*

[8] M. E. J. Newman. Networks: An Introduction. *Oxford: Oxford University Press, 2010*

[9] M. Benzi, Christine Klymko. On the limiting behavior of parameter-dependent network centrality measures. *SIAM J. Matrix Anal. App., Vol. 36, No. 2, pp. 686-706, (2015)*