

Imperial College London
Department of Physics

Paths and Directed Acyclic Graphs

Vaiva Vasiliauskaite

A thesis submitted for the degree of
Doctor of Philosophy of Imperial College London, May 2020

Declaration of Originality

I hereby declare that all work presented in this thesis is either my own or has been appropriately acknowledged.

V. V.

Copyright

The copyright of this thesis belongs to the author. The thesis is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence, making it free to copy, distribute or transmit on the condition that it is attributed, not used for commercial purposes and not altered.

Abstract

Network theory studies complex interdependencies, commonly observed in our interconnected world. Many of those networks express an important constraint leading to a characteristic order; examples include publication dates of papers in a citation network, dependency of packages in computer software, and prey species in a food web. If edges respect this order, they exist only if they are directed from one node to a node later in the order and there can be no cycles—we observe a Directed Acyclic Graph (DAG). So in a DAG a link between two nodes represents an *order* of the pair, not necessarily their similarity as links are often interpreted in standard network analysis. To better understand these common network topologies, we need to adapt our tools so that they respect the order implicit in a DAG.

In this thesis our question is what are the implications of this order on the paths, observed in a network. In particular, we study the relation between network paths and geometric geodesics, the statistics of the longest path, the meaning of centrality, and community detection in DAGs. We demonstrate how the presence of an order of nodes changes the network structure itself, as well as the analysis of it.

Acknowledgements

I would like to express my gratitude to my supervisor, Tim S. Evans without whom this thesis and the projects within would not have been possible. Multitude of times, your support and wisdom were sole reasons for my perseverance when the road was tough. Your brilliant knowledge of the field and passion for scientific endeavour will always be a source of inspiration to me and I will always be grateful to have been your student.

I would like to express my gratitude to Stephen Barnett, whose love for physics sparked my own. I wish to thank David Ireland for introducing me to the field of complexity. I would also like to thank Fernando Rosas, whose academic and non-academic advice had a great impact on me. Thank you to Paul Expert, conversations with whom were always thought-provoking and mind-bending and were often the highlight of my mid-week. I am expressing my gratitude to Ray Rivers, whose lifelong fondness of mind-boggling scientific questions inspired me greatly and expanded my horizons. I would like to thank the members of the Centre of Complexity Science as well as Theoretical Physics group at Imperial College London. I thank Henrik Jensen, Kim Christensen, Gunnar Pruessner, as well as their students for our conversations.

I wish to also thank my dear friends, whom I am happy to have met through my studies: Max Falkenberg-Mcgillivray, Hardik Rajpal, Goodwin Gibbins, James Clough, Michael Kitromilidis, Qing Yao, Matthew Garrod, Henry Price.

Whole-heartedly, I would like to thank my family and closest friends. I would like to thank my mother, sister, grandmother and my father for unconditional support, unbreakable belief in me and patience. I would like to express my gratitude to Ilona Tuminauskas, for advice, support, inspiration. Lastly, I am grateful and incredibly thankful to Tomas Aidukas, whose love and caring helped me in myriad ways. Thank you for always keeping the perspective. Out of all of our adventures, I thank you most for helping me with this one.

V. V.

Contents

1	Introduction	10
1.1	Complexity	10
1.2	Networks	11
1.3	Directed Acyclic Graphs	17
1.4	Objectives	27
2	Geodesics and Paths in DAGs	28
2.1	Introduction	28
2.2	Geometry and Paths	29
2.3	Networks and Space	33
2.4	Geodesics in DAGs	36
2.5	Discussion	43
3	The Longest Path in the Price Model	46
3.1	Analytic Result for Scaling of the Reverse Greedy Path	47
3.2	Numerical Methods	49
3.3	Conclusions	59

4 Centrality and Directed Acyclic Graphs	61
4.1 Introduction	61
4.2 Requirements for Centrality	64
4.3 Some Standard Measures of Centrality	66
4.4 DAGs and Standard Measures of Centrality	69
4.5 Centrality and Topological Order	72
4.6 Other DAG Centralities	84
4.7 Discussions	88
5 Order-respecting Communities	90
5.1 Related Network Partitioning Methods	93
5.2 Siblinarity Antichain Partitions	97
5.3 Siblinarity Communities in Some Networks	103
5.4 Conclusions	111
6 Discussions	114
Bibliography	116
Appendix	131
A Space-Time Lattice Model	132
B Price Model with Subject Fields	133
C Numerical Reverse Greedy Path Algorithm	135
Special terms, abbreviations, symbols	136

List of Tables

1.1	Network statistics of data.	25
3.1	Difference between the goodness of fit using a linear fitting function and a non-linear fitting function.	57
3.2	Variation in fit parameters obtained with various initial graphs. .	58
4.1	Definitions of some well-known centrality measures.	67
4.2	Examples of two-neighbour configurations, expected centrality or- der, neighbourhood inclusion preorder and neighbourhood size re- lation.	75
5.1	Siblinarity communities in the Florida bay network.	111

List of Figures

1.1	An example of a network and adjacency matrix.	14
1.2	Difference between an arborescence, a directed tree, a directed acyclic graph and a directed network.	16
1.3	DAG as space-time.	22
1.4	Example of the network growth in the Price model.	24
2.1	An illustration of Minkowski-RGG model.	38
2.2	Examples of Minkowski-RGGs.	40
2.3	Largest deviation from the shortest path and the longest path to geometrically defined geodesic for various ℓ_{\max}^2 values.	42
2.4	Largest deviation from the shortest path and the longest path to geometrically defined geodesic for various c values.	43
3.1	The difference between the reverse greedy path and the longest path.	48
3.2	Path length scaling with the $\ln t$ in a single network and in many network realisations.	50
3.3	Path length distributions.	52
3.4	The ratio between observed and expected path length values.	53
3.5	The ratio between the observed coefficient of $\ln t$ and the analytic prediction $m\bar{p}$.	55

3.6	The ratio of a_{\max}/a_{gr}	55
3.7	The ratio of b_{\max} and b_{gr}	56
3.8	Initial graph effect for the resulting path length scaling with t . . .	57
4.1	Centrality order.	68
4.2	Correlations between centrality measures in various networks. . .	79
4.3	Entropy of future sets.	82
4.4	Mutual information between centrality order and the topological order for out-degree in a network generated using the Price model.	83
4.5	Correlations between the shortest-path-based centrality and the longest-path-based centrality.	87
5.1	Example of an antichain and a siblinarity community.	92
5.2	An example of a modular network.	95
5.3	Various partitions of a space-time lattice.	104
5.4	Diversity of siblinarity antichains and height-based antichains in the networks generated using a variation of the Price model with subject fields.	106
5.5	Diversity of siblinarity antichains and height-based antichains in a single network generated using a variation of the Price model with subject fields.	108
5.6	Diversity of siblinarity antichains and height antichains in the <code>cora</code> network.	109
B1	An illustration of the Price model with subject fields.	134

Chapter 1

Introduction

Some parts of this chapter are based on work presented in:

Understanding Complexity via Network Theory: a Gentle Introduction,

V. V., F. E. Rosas,

arxiv.org/abs/2004.14845 (2020).

V. V. contributed to sections 1, 3 – 6.

1.1 Complexity

Many real-world systems are interconnected. A collection of closeby particles, a group of friends or financial stocks are seemingly very different things. Interactions between particles are governed by physical laws, which may not necessarily apply to interactions between people. We can know every physiological detail about the human body and it will not suffice to understand social interactions between two people. Learning about every single particle within the neuron will not be enough to explain how billions of neurons form a conscious brain. All of these COMPLEX systems are intractable using reductionist's approaches. Complexity science aims

to explain such patterns of complex interactions.

A complex system embodies a certain amount of hierarchy and interactions happen between and among its subsystems [1]. Furthermore, these interactions exhibit law-like and causal regularities, various kinds of symmetry, order, and periodic behaviour [2]. What distinguishes a complex system from a complicated system is non-linearity of interactions between its constituents. The aggregate of them exhibits properties, called emergent, that are not attainable by summation [3].

An example of a non-complex system is a watch, which, with many screws, gears, springs, is a complicated device. We may even say that its emergent behaviour is its time-keeping ability. However, we can understand the mechanisms underlying clock's architecture completely, if we take it apart. Thus although complicated, clocks are not complex [3]. In a complex system, "top-down" as well as "bottom-up" effects can be observed. For instance, the structure of the World Wide Web (WWW) is dependent on the individuals, contributing to creation and linkage of websites. On the other hand, the navigation within the WWW by a single individual is inevitably dependent on the structural features of the WWW itself.

1.2 Networks

Networks are a useful tool to model a wide variety of complex systems. A network is an abstraction of a system that depicts interdependencies within it. Just as a map neglects aspects of the territory in order to focus on particular features of interest, a network takes away most of the richness of individual sub-units of the system and only retains the structure of their interdependencies. Networks allow us to study the collective properties of these dependencies. Additionally,

as seemingly unrelated systems might have similar networks of interdependencies, networks sometimes allow us to establish non-trivial relationships between systems that share similar dependency structures.

In the following sections of this chapter we review the main technical definitions, used throughout the thesis. The notation, introduced here or elsewhere is also summarised in the glossary at the end of the thesis.

1.2.1 Basic Definitions

The interacting entities in a network are referred to as NODES (or VERTICES), and an interaction between two nodes is indicated by an EDGE. So in the simplest form, a network consists of a set of nodes with a set of edges between them [4, p. 1, 109]. We define a network (also called a graph) as a pair \mathcal{G} , consisting of a set of nodes \mathcal{V} and edges \mathcal{E} : $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set \mathcal{V} consists of all nodes: $\mathcal{V} = \{u, v, w\ldots\}$ and the size of this set is equal to the number of nodes in the network, N . The set \mathcal{E} consists of all edges among nodes in \mathcal{V} and its size is equal to the number of edges in the network, E .

The type of network just described is the most simple. Nodes in this network have no internal structure, and in this sense are identical. For instance, a network can be used to model relationships between people. An edge in a friendship network of people indicates that some two people, represented as nodes, are friends [4, p. 37]. In this network, all people are assumed to be identical: they have no individual personality, status, ethnicity.

If we want to incorporate more information, we can equip a network with richer structure. For instance, a node can be assigned a type. When there are two types of nodes in the network, and an edge only connects nodes of different type, a network is called BIPARTITE, see [4, p. 123]. Such networks are useful to model affiliations: a person belonging to a group, a company, a team. An

example of a bipartite network is a share-holder network where an edge connects a company and a person, if the person invested in the company [5]. Bipartite networks can be generalised to multipartite networks. A multilayer network on the other hand allows for multiple layers, as well as inter-layer connections [6]. Extra information can be added not only to nodes, but also to edges. Significance, strength of interaction can be represented with WEIGHTED edges, such a network is called a weighted network. In signed networks, a link can have a positive or a negative value. This type of network is useful to model polarised relationships, e.g. “like” and “hate” in social setting [7]. Lastly, edges can be equipped with orientation to obtain directed networks.

Examples of networks are numerous. To name a few significant ones, networks are useful to model social interactions (for example, see [8; 9; 10] and references therein), migration patterns, technological networks (such as the internet [11] and the World Wide Web [12]), biological systems (e.g. brain networks [13; 14; 15], protein-protein interactions [16]). In section 1.3.3 we will discuss examples of real world systems, that can be modelled as directed acyclic graphs—networks that are studied in this thesis.

Network theory is built upon a rich mathematical foundation of graph theory and statistical mechanics. To access the mathematical machinery, we define a network as a quantitative entity. Although many representations are possible, in this thesis we will always use an ADJACENCY MATRIX, **A**. In the simplest case this is a square $N \times N$ matrix that represents edge connectivity of a network with N nodes. For an undirected network each entry in the adjacency matrix is

$$\begin{aligned} A_{i_u j_v} &= A_{j_v i_u} = e \text{ if } (u, v) \in \mathcal{E}, \\ A_{i_u j_v} &= 0 \text{ otherwise.} \end{aligned} \tag{1.1}$$

Here e is the weight of the edge (u, v) and i_u, j_v are indexes of nodes u, v , respec-

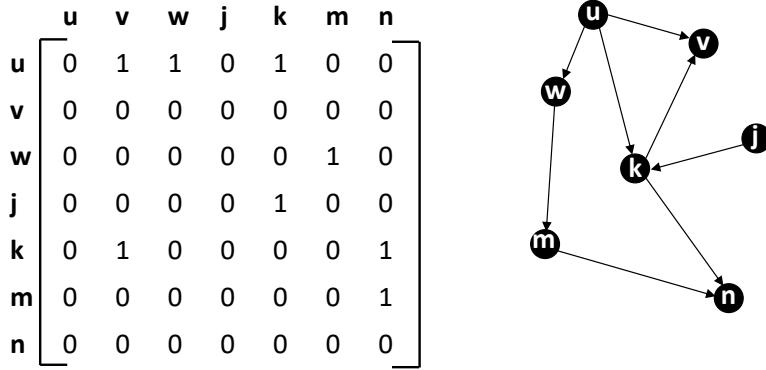


Figure 1.1: An example of a directed network on the right and its adjacency matrix on the left. There are $N = 7$ nodes in the network, $E = 8$ edges. Node u has an out-degree $k^{(\text{out})}(u) = 3$.

tively. If the network is directed, this matrix is not symmetric, and if an ordered pair $(u, v) \in \mathcal{E}$, $A_{i_u j_v} = e$, but $A_{j_v i_u} = 0$ if $(v, u) \notin \mathcal{E}$. An example of a directed network and its adjacency matrix is given in Fig. 1.1. Most often we will consider unweighted networks, unless otherwise stated. Having a mathematical representation of a network, we can define various characteristics of individual nodes, edges, as well as the network as a whole. For instance, DEGREE of a node is the number of edges that are attached to that node: $k(u) = \sum_{v \in \mathcal{G}} A_{i_u j_v}$. When the network is directed, two types of degrees can be distinguished: degree, based on edges pointing to the node (in-degree, $k^{(\text{in})}$), and degree based on edges pointing outwards, $k^{(\text{out})}$. For instance, in Fig. 1.1, $k^{(\text{in})}(u) = 0$, $k^{(\text{out})}(u) = 3$. We can calculate the degrees of nodes directly from the adjacency matrix:

$$k^{(\text{in})}(u) = \sum_{v \in \mathcal{G}} A_{j_v i_u}, \quad k^{(\text{out})}(u) = \sum_{v \in \mathcal{G}} A_{i_u j_v}. \quad (1.2)$$

When two nodes are connected with an edge, we call them each other's NEIGHBOURS. A set of all neighbours of u is called the NEIGHBOURHOOD \mathcal{N} of u . In directed networks, \mathcal{N} can be divided into a set of nodes that have an edge pointing to u , PREDECESSORS \mathcal{N}^{pre} : $\mathcal{N}^{\text{pre}}(u) = \{v \in \mathcal{V} | (v, u) \in \mathcal{E}\}$, and those that are pointed to from u , SUCCESSORS \mathcal{N}^{suc} : $\mathcal{N}^{\text{suc}}(u) = \{v \in \mathcal{V} | (u, v) \in \mathcal{E}\}$:

$\mathcal{N} = \mathcal{N}^{\text{suc}} \cup \mathcal{N}^{\text{pre}}$. Those nodes which have no successors $k^{(\text{out})} = 0$ will be called SINKS, and those that have no predecessors will be SOURCES.

1.2.2 Paths

Networks become truly useful when there is a need to understand collective behaviour of the system via its large- or meso-scale organisations. To understand such behaviours we study indirect, long-range interactions between nodes that operate via paths, walks, cycles. A WALK is defined as a sequence of vertices, such that there is an edge from each vertex to the next vertex in the sequence. A number of walks can be calculated from the adjacency matrix. For instance, in an undirected network, for $u, v \in \mathcal{V}$, $[\mathbf{A}^k]_{i_u j_v}$ is the number of walks of length k between the nodes [4, p. 136]. In a walk, the sequence of traversed nodes need not be unique.

A walk, in which each node of the sequence occurs at most once, is called a PATH. Formally, a path from u_0 to u_L , $\mathcal{L}(u_0, u_L) = \{u_0, \dots, u_L\}$ is a sequence of nodes, such that $(u_i, u_{i+1}) \in \mathcal{E} \forall i \in \{0, L - 1\}$. We will often look at the length of a path between two nodes, which is one less than the number of nodes in the path set, L in the above definition. We will most often assume the *shortest* path distance between two nodes and denote the length of a path between u, v as d_{uv} . In section 4.6 we will also consider the longest path distance between nodes. There we will call the length of the shortest path d_{uv}^{SP} , the length of the longest path— d_{uv}^{LP} . We will also discuss the number of shortest paths between nodes, denoted as σ_{uv} . The number of those paths that pass by a node w is σ_{uw}^w .

In all but one type of network there are at least two distinguishable paths between at least one pair of nodes. If for all pairs of nodes there is at most one path connecting them, such network is a TREE. The notion of a tree can be generalised to directed networks. An ARBORESCENCE is a network that has one

source node, called a root, and all paths from the root to any other node are unique. When multiple source nodes exist, the network is a DIRECTED TREE [17].

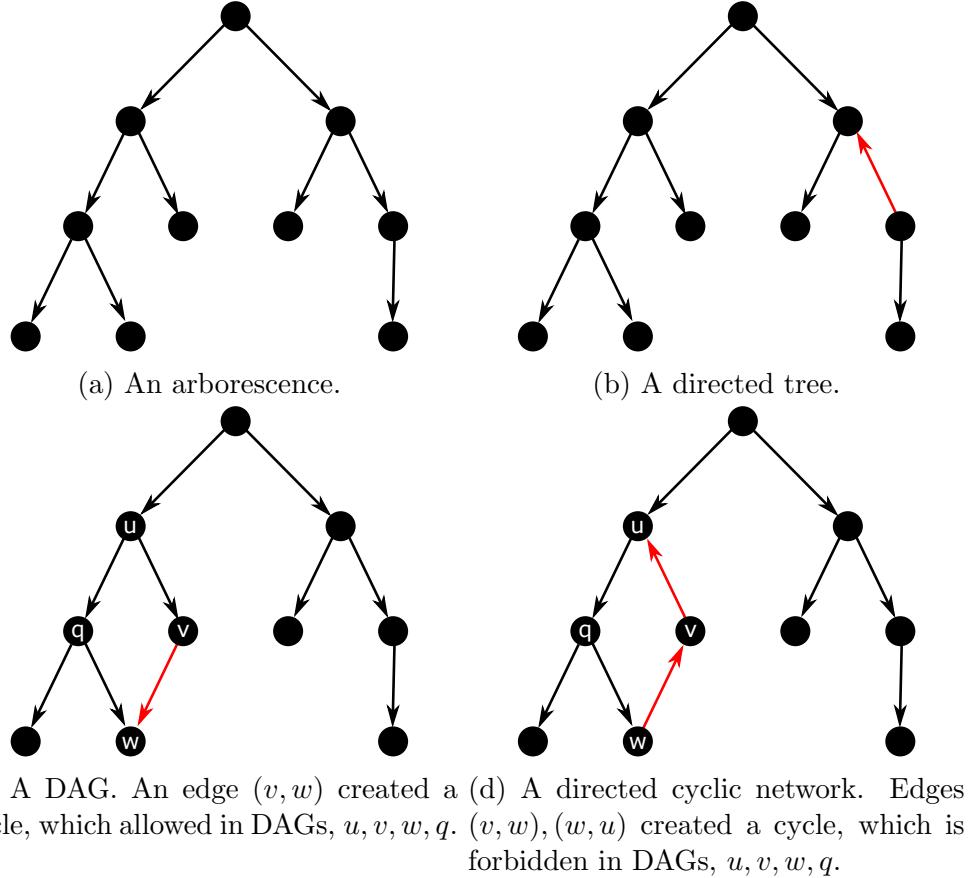


Figure 1.2: Difference between an arborescence, a directed tree, a directed acyclic graph and a directed network. Flipping the direction of a red edge produces a second source node in the arborescence of (a), resulting in a directed tree in (b). In both, arborescence and a directed tree, there is at most one path between any pair of nodes. Addition of an edge (v,w) in (c) or edges $(w,v), (v,u)$ in (d) produce cycles, composed of nodes u, v, w, q , so these networks are not directed trees, nor arborescences. A cycle u, v, w, q in (d) is a “feedback” cycle which is forbidden in DAGs.

Formally, when there is more than one path between a pair of nodes, a network has a CYCLE. A cycle is a special type of subgraph \mathcal{G}' . Generally \mathcal{G}' is a pair of nodes $\mathcal{V}' \subset \mathcal{V}$ and edges $\mathcal{E}' = \{(u,v) \in \mathcal{E} | u, v \in \mathcal{V}'\}$. A subgraph is a cycle if $k(u) = 2 \forall u \in \mathcal{V}'$. If the cycle is directed, we must have $k^{(\text{out})}(u) + k^{(\text{in})}(u) = 2 \forall u \in \mathcal{V}'$. Differences between an arborescence, a directed tree, a DAG, and a simple directed network are illustrated in Fig. 1.2.

Various insights about networks can be acquired from the statistics of different types of paths. THE SHORTEST PATH between the two nodes (interchangeably called a geodesic e.g. in [18],[4, p. 139]) is a path with smallest possible number of edges. A classic example of the use of the shortest path statistics is for assessing positional importance of nodes. Here we refer to measures of centrality which will be discussed in great detail in Chapter 4. Furthermore, shortest paths are also used to find optimal routes [19].

To study dynamical processes on networks one can employ various stochastic methods, for example random walk techniques. Random walks are often used to model various types of diffusion, and there has been intense research on the impact of network architecture on the dynamics of random walks. The finiteness of a network, along with its irregular structure can make diffusion on networks very different from diffusion on regular or infinite lattices. Thus the statistics and dynamical properties of a random walk can reveal information about the structure of a network [20].

1.3 Directed Acyclic Graphs

The main focus of this thesis is on a special type of directed network, in which all walks are paths, and none of the paths start and finish at the same node. We will call such network a DIRECTED ACYCLIC GRAPH (DAG). Although this is standard nomenclature, the name is misleading: if edge direction is ignored, cycles in DAGs do exist. So most formally, DAGs are networks in which there are no “feedback” cycles: walks that, following the edge direction, would start and finish at the same node. See Fig. 1.2(c) for an example of a cycle, allowed in DAGs, and Fig. 1.2(d) for an illustration of a feedback cycle, forbidden in DAGs.

Directed acyclic structure occurs in systems with underlying characteristic

constraint, which orders nodes and governs the direction of edges [21]. So if in the standard network analysis an edge indicates the presence of a relationship, interaction, or similarity between the two connected nodes, in DAGs an edge also indicates the hierarchical relation, causality, or order of the pair of vertices.

Perhaps the most significant example of DAGs is that of networks ordered in time¹. For instance, a CITATION GRAPH [23] is a network in which both edges and nodes respect temporal order. In citation networks nodes are publications and an edge indicates a citation of one publication by another. In this thesis we will use the convention of edges pointing forward in time: an edge starts *from* a referenced paper and points *to* a referencing paper. In a citation network, an edge not only marks a reference but also indicates the temporal order of papers: a paper can only reference older papers that have been published already. We can say that in part, the temporal structure can be recovered from the topological structure of the network (where we do not necessarily know the real “ages” of nodes).

Another example of a DAG is a food web. This network depicts prey-predator relationships between species in an ecosystem. Usually food webs are constructed by considering the flow of energy or mass between compartments (species) within the system [24]: there is an edge from u to v if v consumes energy from u , often meaning that species v eats species u .

In practice there are often “bad” links which are in the “wrong” direction, from a newer document to an older document in a citation network or cyclic energy flows in a food web (e.g. cannibalism). In citation networks, this is because documents are published in different versions and the text available may not have been created at the time associated with the document in the data set.

¹Note that these are not temporal networks. In a temporal network, time at which edges appear is relevant for dynamical processes of the system [22]. An example of a system that can be modelled as a temporal network is email communication between people where a sent message from one person (node) to another is represented as a directed edge. In these networks time does not govern the directionality of edges, so time is not a characteristic constraint ordering the edges.

For instance, a revised version of an arXiv paper carries the same index as the first version. A journal article has several associated dates: first submitted, date accepted, published online, formal publication date and so forth. Such bad links can introduce cycles and these must be mitigated. One can simply drop the links that do not respect the agreed order of time, or go against the order imposed by trophic layers. More sophisticated approaches also exist to remove the acyclicity-breaking edges, for example see [25; 26].

Networks can also represent a hierarchical structure. For example, within software, many packages, programmes, algorithms can depend on other pieces of software. For the whole package to run smoothly, one must ensure that there are no cyclic dependencies. E.g. in `python` programming language, `pandas` library is built upon `numpy` library. We may say `numpy` precedes `pandas` in the hierarchy of software dependencies. For `python` to run smoothly, `pandas` cannot be a prerequisite of `numpy`, so software package dependencies are another example of a DAG.

Another DAG closely related to computer package dependency network is a task scheduling network [27]. Here tasks are ordered: one task must necessarily be executed before beginning another (to make coffee, beans are ground before being put in a coffee brewer). All of these examples are DAGs as there exists inherent causality within the system. Indeed, causal networks (Bayesian networks which are always directed acyclic graphs) are structures that represent conditional dependencies between entities within the system [28, Chapter 2].

1.3.1 Basic DAG Properties

The lack of cycles in a DAG leads to many special properties that distinguish this network type. Every DAG is linked to a PARTIAL ORDER on its set of nodes. That is, DAG is related to a partially ordered set—POSET—of nodes \mathcal{V} in which nodes

are ordered via reachability criterion. We say that an element a in a partially ordered set precedes another element b if these elements are related in the order and a comes before b . We write $a \prec b$, with relation “ \prec ” satisfying the following requirements for all pairs of elements, ordered in the set [29, p. 1-2]:

- Reflexivity: $a \prec a$,
- Antisymmetry: if $a \prec b$ and $b \prec a$, then $a = b$,
- Transitivity: if $a \prec b$ and $b \prec c$, then $a \prec c$.

In a DAG, we can say that whenever there is a path $\mathcal{L}(u, v)$, $u \prec v$. This relation between the mathematical entity of a poset and any DAG means that we can exploit the mathematical machinery of ordered set theory to study our DAGs.

In the following chapters we will often do so.

On the other hand, DAGs are different from posets. In network representation, a partially ordered set is a TRANSITIVELY COMPLETE DAG, meaning that if $\exists \mathcal{L}(u, v)$, then $\exists(u, v) \in \mathcal{E}$. TRANSITIVE CLOSURE [30] is a process of injecting edges in a network, such that whenever there is a path between two nodes, there is also an edge between them. So a partially ordered set related to a DAG can be obtained by performing transitive closure.

Each directed acyclic graph can relate to at most one poset, however one poset is related to a collection of DAGs. DAGs, which have the same forms of transitively reduced network and transitively complete network, share the same partially ordered set. TRANSITIVE REDUCTION (TR) is a process of removing those edges from a DAG, which are not essential to maintain the connectivity within a network [31]. That is, if for an edge (u, v) there is another, longer path which connects these two nodes, the edge (u, v) is not essential and can be “reduced”.

We can also order nodes linearly, that is, find a TOPOLOGICAL ORDER of nodes, i.e. put them in a sequence such that u comes before v if $u \prec v$ [32]. Such topological order always exists, however it is not unique. If there are two nodes, which do not have a path connecting them, either node can be put forward in the topological order. Such a set of nodes for which there are no paths between any of the nodes is known as an ANTICHAIN. We will say that if $u \not\prec v$ and $v \not\prec u$, $u \parallel v$. An antichain will be denoted \mathcal{A} , $\mathcal{A} = \{u, v \in \mathcal{V} \text{ if } u \parallel v\}$. Conversely, a CHAIN is a generalisation of a path, that is, if $u \prec v$, then u, v form a chain. In this thesis, to order nodes we will most often use the information about the inherent constraint of the system. An example of a constraint is time in citation networks, so the index of a node in topological order will correspond to its publication date: first is the oldest paper, followed by the second oldest and so on. When there is no sensible metadata, we will use a `topological_sort` function of `networkx` library in `python` to sort the nodes. The algorithm is based on the description in [33, Chapter 7]. We will say that i is an INDEX of a node u if it is in the i^{th} place in the linear topological order.

Another important feature of DAGs is that there is always a well-defined longest path between any two connected nodes. We will denote the longest path as \mathcal{L}^{LP} and its length L . Furthermore, we can use the longest path to define HEIGHT and DEPTH of each node. The height of a node u , $h(u)$ is defined as the longest path distance from any source node to u ; the depth of u , $d(u)$ is the longest path distance from the node to any sink node.

It is also important to note that adjacency matrices of DAGs are always upper-triangular, which are nilpotent matrices that have all eigenvalues equal to zero. As will be discussed in Chapter 4, this feature has important effects on measuring centrality in DAGs.

1.3.2 Relation to Special Relativity

DAGs in which nodes are ordered with respect to time are similar to causal sets.

Causal sets theory states that at quantum level, space-time is a partially ordered set in which the order relation \prec represents causality of space-time events [34].

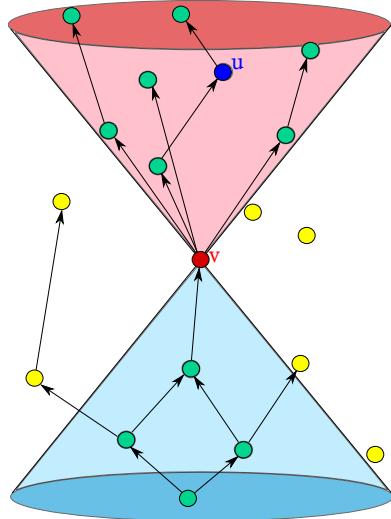


Figure 1.3: DAG as space-time. If nodes are space-time points, then each node has a “future” and a “past”. The future of a node v , $\mathcal{F}(v)$ is the set of its descendants (red cone), $u \in \mathcal{F}(v)$. Past of v , $\mathcal{P}(v)$ (blue cone) is the set of its ancestors. We can say that nodes that v are not connected to (in yellow) are space-like separated with the node.

In time-ordered DAGs, when $u \prec v$, u must necessarily be older than v . We will say that u is in the PAST of v . The set of all nodes in the past of v is then the PAST of v , $\mathcal{P}(v)$ (note that this set does not include v itself. When it does, we will write $\mathcal{P}[v]$). In networks, this set is also called a set of ANCESTORS. On the other hand, if $u \in \mathcal{P}(v)$, then v is in the FUTURE of u , $\mathcal{F}(u)$ (v is a DESCENDANT of u). Future and past sets of a node are shown in Fig. 1.3. Lastly, we can take an intersection $\mathcal{P}[v] \cap \mathcal{F}[u]$ and define an interval as a subgraph $\mathcal{I}[u, v] = \{w \in \mathcal{P}[v] \cap \mathcal{F}[u], (w, q) \in \mathcal{E} | w, q \in \mathcal{P}[v] \cap \mathcal{F}[u]\}$. An interval is a subgraph with nodes which are causally related to both u and v . More links to the notion of space-time and special relativity will be explored in Chapter 2.

1.3.3 Random DAG Models

In the following chapters we will study properties of DAGs via analysis of real data and network models. The latter are random graphs, in which some parameters are fixed but other features vary. For example, we can take an observed network and shuffle the edges in a way which preserves the degrees of nodes (a configuration model [35, Chapter 7]). Other parameters that can be fixed include the number of edges, diameter, clustering coefficient and so on.

Here we review two well-known network models, which will be used in the thesis, namely a random (Erdős-Rényi) network, and the Price model of cumulative advantage (or preferential attachment). The Erdős-Rényi model is well-known as a model of an undirected network, and here we discuss how it can be used as a random model of DAGs. The undirected version of the Price model is also known as a Barabási-Albert model [36]. In some chapters, we will also use more unconventional random models, which will be discussed there and in the appendices.

Erdős-Rényi Model

The Erdős-Rényi network model [37] is one of the simplest network models, the essential element of which is its randomness. In this model one assumes no prior knowledge about the network, nodes and edges. To build a representative network of this model, one follows a simple rule: consider each pair of nodes, and with some probability p add an edge between them. To build a directed acyclic graph, we will first assign each node u an index i_u and add directional edges which point from nodes with smaller-valued indices to larger-valued indices.

Price Model

In the Price model [38] (for instance see sec.14.1 of Newman[4]) we start from a network $G(t)$ defined at an integer ‘time’ t . We create a new graph $G(t + 1)$ by

first adding one new vertex, which we label with the time $(t + 1)$. This new node, $(t + 1)$, is connected to m existing vertices s in the graph $G(t)$. These m existing vertices $\{s\}$ are each chosen with probability $\Pi(t, s)$. We will use a convention that these edges point from older to newer vertices, from s to $(t + 1)$. Once these edges have been added we have our new graph $G(t + 1)$. The process is then repeated. For an example of how a network grows according to the Price model, see Fig. 1.4.

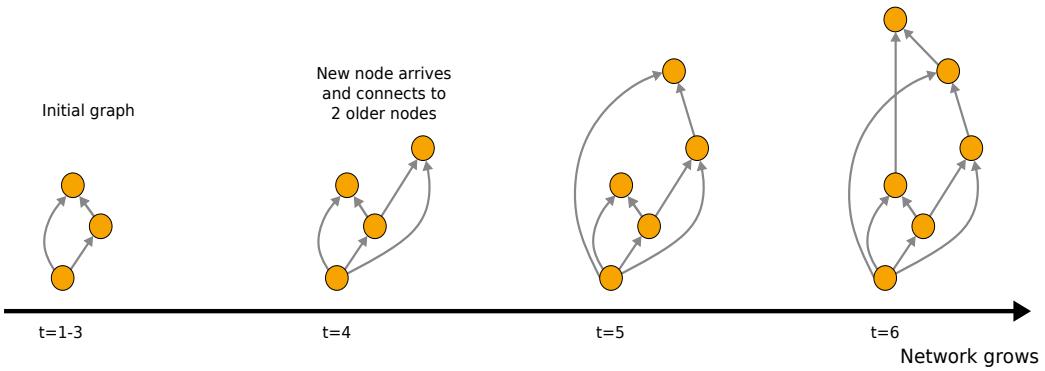


Figure 1.4: An illustration of the Price Model. Here the height of the node on the page indicates the time with the first node at the bottom being the node $t = 1$. At each stage we show the graph $\mathcal{G}(t)$ so after the new nodes and its incoming edges have been added. (Figure taken from [39]).

The mathematical and numerical simplicity of this model comes from the simple definition of $\Pi(t, s)$. To define the probability $\Pi(t, s)$ we first define $N(t) = N_0 + t$ be the number of nodes in the graph $G(t)$ for some constant N_0 . The number of edges in the graph $G(t)$, after all m edges have been added to node t , is $E(t) = E_0 + mt$ where E_0 is some constant. Finally in the graph $G(t)$ let the node created at time s have out-degree $k^{(\text{out})}(t, s)$, the number of edges leaving s and connecting it to later nodes.

In this model, the connection of edges to new node $(t + 1)$ is made in one of two ways. With probability p node $(t + 1)$ is connected to an existing vertex

s chosen in proportion to the number of edges leaving s for later nodes in $G(t)$, $k^{(\text{out})}(t, s)$. Price called this CUMULATIVE ADVANTAGE and, after normalisation, we have that the probability of choosing s is $k^{(\text{out})}(t, s)/E(t)$. The second process happens with probability $\bar{p} = (1 - p)$ and in this case we choose the source vertex s uniformly at random from the set of vertices in $G(t)$, i.e. with probability $1/N(t)$. If we start the process at time equal to 1, then the probability of connecting the vertex $(t + 1)$ to existing vertex s is $\Pi(t, s)$ where

$$\Pi(t, s) = p \frac{k^{(\text{out})}(t, s)}{E(t)} + \bar{p} \frac{1}{N(t)} \text{ if } t \geq s \geq 1, \quad (1.3)$$

unless $s = t = 1$ when $\Pi(t, s) = 1$, otherwise $\Pi(t, s) = 0$. Note that in his original paper, Price considered $p = m/(m + 1)$ where $\Pi \propto k^{(\text{out})} + 1$.

1.3.4 Datasets

We will also use data of citation networks and a food web, described in this section. Summary statistics are given in Table 1.1

Name	Type	N	E	E_{DAG}	Source
hep-th	Citation	27,770	352,806	351,500	[40]
hep-ph	Citation	34,529	419,528	419,528	[40]
astro	Citation	103,526	918,853	912,880	[41]
cora	Citation	2,708	5,429	5,255	[42; 43]
Florida bay	Food web	128	2,106	1,930	[44]

Table 1.1: Network statistics of data used in the thesis.

First, a citation network of papers in arXiv High Energy Physics-Theory repository, for papers published between 1992 and 2003, taken from [40]. We will refer to this dataset as **hep-th**. It contains 27,770 nodes and 351,500 edges (edges which did not respect the time order, i.e. pointed from newer papers to older papers, and were thus cycle-inducing, were dropped). We also used the citation network of arXiv High Energy Physics-Phenomenology repository, referred to as

`hep-ph`, also taken from [40]. For this network we also used the temporal stamps of nodes to identify time order-disobeying edges.

Web of Science Astronomy and Astrophysics papers, published between 2003 and 2010 (referred to as `astro`) is another citation network we used [41]. This dataset has 103,526 papers and 921,880 acyclic edges. For this data we used ISI identifiers and assumed that the larger the value, the older the paper, and only added links when there is a citation from one paper of higher ISI identifier than the paper being cited. This eliminates all cycles but 5973 citations are not encoded as edges in our network as a result, which is 0.64% of the total in this data.

The last citation network we will use is a cora dataset consisting of a selection of 2,708 scientific publications on Machine Learning, classified into one of seven classes, based on their topics. The papers were selected in a way such that in the final network every paper cites or is cited by at least one other paper. This citation network consists of 5,429 edges. Since this dataset does not include publication dates, we used the approach, discussed in [25] to recover a DAG from a directed network, leaving us with a network, composed of 5,255 edges.

Lastly, we will also study a food web of the Florida Bay [44]. In this network, the nodes are compartments and edges represent directed carbon exchange in the Florida Bay. There is an edge from u to v if compartment v consumes carbon from compartment u . The compartments are mostly organisms but also encompass special nodes such as “input” and “output”. We also had group classification labels. Example groups include “Zooplankton Microfauna” and “Pelagic Fishes” [45]. The original network consists of 128 nodes and 2106 edges but this contains cycles. We used the breadth-first search approach, described in [25] to recover a DAG, removing 176 edges (less than 9%).

1.4 Objectives

My broad goal is to understand what implications the intrinsic order of nodes in directed acyclic graphs has. As much of network analysis is based on statistics of network paths, my focus will be on understanding how paths in DAGs are affected by the order of nodes. For instance, since in directed acyclic graphs the longest path between nodes exists, one of my goals is to understand where in DAG analysis the notion of the longest path can be used.

This thesis will begin with Chapter 2 in which the aim is to motivate the use of the longest paths in the analysis of DAGs. We will see that although the shortest path is often considered the *fundamental* proxy of distance between nodes, and so is called the *geodesic*, the longest path may be a better-suited path for such role in DAGs. We will then study the properties of the longest paths in the Price model in Chapter 3.

In Chapter 4 we will study the path-based measures of importance—centrality—and how centrality is affected by the intrinsic order of nodes. We will also look into variations of centrality measures, based on the longest paths, and see whether they are better-suited to evaluate importance of nodes.

Lastly, in Chapter 5 we will discuss community detection, a standard network measure of partitioning networks into sets of similar nodes. We will examine an adapted version of community detection algorithm which allows us to detect “hierarchically equivalent” and notionally similar nodes.

The aim of this work is to emphasise that the order found in DAGs and data represented by them is an important feature which should not be ignored. However, this ordering constraint is not accounted for in most traditional network measures suggesting that network analysis methods need to be reexamined carefully for the use in DAGs.

Chapter 2

Geodesics and Paths in DAGs

2.1 Introduction

Paths are an important concept used to explain features of observed networks. The presence of a path between two nodes indicates that some communication between them is possible. Based on various statistics of paths in our networks we can better understand how well-connected, optimised and navigable networks are. For instance, one of the key characteristics of the so-called “small-world” networks is that even though most nodes are not immediate neighbours, they can be reached from any other node in a few hops.

One of the most interesting examples of a small-world network is the well-known Milgram’s experiment [46]. In it, Stanley Milgram, psychologist at Yale University, sent out packages to some people across the United States. These people were not the recipients of these packages but rather were asked to pass them along to other people they knew well and who may know the recipients (if they did not know them themselves). It turned out that the average path length that a letter travelled was around six—the six degrees of separation. The conclusion was that although the social network (people in the country) is large,

everyone is separated by a comparably much smaller number of people—six.

The shortest path has received particularly big attention. The distribution of the shortest paths between nodes is thought to indicate efficient communication in the network. For instance, in brain networks short characteristic path lengths may relate to more efficient internodal information transmission [47, Chapter 7]. A short characteristic path length means that information can, on average, be routed between pairs of nodes using only a few edges. In this way, short path lengths minimise the metabolic cost associated with routing action potentials across axons and synaptic contacts, and hence could provide faster, more direct, and less noisy information transfer [48]. Moreover, large degree of heterogeneity and small amounts of synchronisation in the network can be expected if the longest of all shortest paths, the **DIAMETER** is relatively small [49].

The shortest paths are often referred to as the **GEODESICS**—a word which is meaningful only when there exists a space in which the path is defined. If nodes are points in the space, referring to the shortest path as a geodesic path implies a particular space. In this chapter, we explore the link between networks, spaces, and geodesics. The goal of this chapter is to define DAGs based on appropriate metric spaces and compare geometric geodesic paths with specific network paths.

2.2 Geometry and Paths

2.2.1 Metric Spaces and Riemannian Manifolds

A **METRIC SPACE** is necessary to define paths, geodesics, and distances. Simply put, space X is a collection of geometric points. If such space is also equipped with some function $d : X \times X \rightarrow \mathbb{R}$, the space is called **METRIC SPACE** and d is called a **METRIC**. The metric must satisfy the following requirements for all pairs

of points in X [50, p. 4]:

- $d(x, y) \geq 0$ non-negativity,
- $d(x, y) = 0 \rightarrow x = y$ identity of indiscernibles,
- $d(x, y) = d(y, x)$ symmetry,
- $d(x, y) + d(y, z) \geq d(x, z)$ triangle inequality.

In metric spaces, the metric gives the length of a path between two points. For example, in EUCLIDEAN SPACE the distance between two points is the length of the straight line segment connecting them. Hence Euclidean space is the simplest form of a metric space.

We are interested in such metric spaces, upon which geodesics exist (a generalisation of the shortest path between points, soon to be defined). Geodesics are always well-defined between all points of DIFFERENTIABLE RIEMANNIAN MANIFOLDS. These are metric spaces which are, at all points, real, smooth, and locally flat. The metric of a differential manifold is defined on its TANGENT SPACE: at point p , $T_p M$. The tangent space is a real vector space of the same dimension as the manifold. It contains all possible directions in which one can tangentially pass through p . The elements of the tangent space at p are called the TANGENT VECTORS at p [51, p. 63]. If the tangent space is equipped with a function $g : T_p M \times T_p M \rightarrow \mathbb{R}$, a METRIC TENSOR, which is positive definite for all pairs of vectors of the tangent space at any point on the manifold, such a manifold is called RIEMANNIAN.

The metric tensor is used to measure distances and angles on the manifold. It appears as a real, symmetric $D \times D$ matrix $g_{\mu\nu}$, where D is the dimension of the space [51, p. 72-5]. A signature of a metric is the trace of this matrix. For example, on a three-dimensional Euclidean manifold, g is a 3×3 identity matrix

and the signature of the three-dimensional Euclidean space is three. Given the metric, we can define an inner product on $T_p M$ as [50, p. 130]

$$\langle x, y \rangle_p = \sum_{\mu, \nu=1}^D g_{\mu\nu} x_\mu y_\nu, \quad (2.1)$$

where (x_1, \dots, x_D) and (y_1, \dots, y_D) are vectors in $T_p M$.

Given dx^μ a μ^{th} component of an infinitesimal coordinate displacement vector, the metric determines the invariant square of an infinitesimal line element, often referred to as an interval, defined as

$$ds^2 = \sum_{\mu, \nu=1}^D g_{\mu\nu} dx^\mu dx^\nu. \quad (2.2)$$

Recall that if the manifold is Riemannian, $ds^2 \geq 0$ for all vectors of the tangent space. For instance, in three-dimensional Euclidean space where $g_{\mu\nu}$ is the identity matrix, $ds^2 = g_{xx} dx^x dx^x + g_{yy} dx^y dx^y + g_{zz} dx^z dx^z = (dx)^2 + (dy)^2 + (dz)^2$.

In Riemannian manifolds, the role of straight lines of Euclidean space is replaced by GEODESICS—the shortest paths between points. The distance between two points is then measured along the geodesic. Given a curve that connects two points and a tangent vector, the direction of the tangent vector, transported along the line from the original point to the end point, remains unchanged. To find a geodesic on a differential manifold one can use geodesic equations, defined in terms of the metric (and its derivatives) as well as the components of an infinitesimal coordinate displacement vector (e.g. see [51, p. 102-8]). For example, a sphere is another simple example of Riemannian manifold. Here we find that there are two geodesics between any two points, both following a great circle.

2.2.2 Pseudo-Riemannian Manifolds

If one relaxes the requirement of metric positivity, that is $ds^2 < 0$ is allowed, such manifolds are referred to as pseudo-Riemannian manifolds. Of particular interest are Lorentzian manifolds, with a metric of a signature d where $d + 1 = D$ is the dimension of the manifold. The space-time is modelled as a $D = 4$ Lorentzian manifold, with three spatial dimensions and the fourth, representing time.

What Euclidean space is to Riemannian manifold, MINKOWSKI SPACE-TIME is to pseudo-Riemannian manifold. Minkowski space-time is a flat, four-dimensional space-time with the metric tensor represented as a diagonal matrix with entries $g = (1, -1, -1, -1)$. The square of a distance ds^2 is

$$ds^2 = (cdt)^2 - dx^2 - dy^2 - dz^2, \quad (2.3)$$

where c is the constant speed of light, $c = 1$ in natural units. Strictly, this is a pseudo-distance, as ds^2 is not positive everywhere on Lorentzian manifold. Three different types of separations are distinguished:

- time-like: $ds^2 > 0$,
- space-like: $ds^2 < 0$,
- light-like: $ds^2 = 0$.

Time-like separated events are said to be CAUSAL, space-like separated events are ACAUSAL. Although we will not study curved Lorentzian manifolds, it is worth pointing the reader to perhaps the simplest examples of such curved spaces, namely de Sitter and anti-de Sitter space-times, e.g. see [52] for an introductory text on de Sitter space.

2.3 Networks and Space

2.3.1 Networks Based on Geometry

A fundamental element of a network, a node, is reminiscent of a fundamental element of space, a geometric point. However, the relationship between networks and space is vague. It is not clear what space a network represents, as nodes in networks, unlike points in space, do not have coordinates, and there is no implicit metric.

Spatial networks are those in which vertices (or edges) are spatial elements associated with geometric points, i.e. nodes are located in space equipped with a metric [53]. The simplest form of a spatial network is a network that represents a geographic system. In such systems geometry is an intrinsic and important feature that influences network processes. Examples of such systems are numerous. Spatial networks include transport networks (e.g. networks of connections between public transport stops [54]), power grids (refer to a review on power grids [55] and references therein), so-called “urban” networks. An urban network can be constructed by considering street intersections as network nodes. One obvious network of such nodes can be obtained by calling portions of streets between intersections to be edges, e.g. see [56]. Clearly, most geography-representing networks are based on two- or three-dimensional Euclidean space. Infrastructure-representing networks are often **PLANAR GRAPHS**—networks that can be plotted on a 2-dimensional plane without any crossing edges.

The Euclidean geometry of a spatial network implies that the further apart the nodes are, the less likely there is an edge between them [53]. This is exactly the idea behind a **RANDOM GEOMETRIC GRAPH** (RGG). To create an RGG, a first step is to generate N nodes and assign each of them coordinates in a metric space. The parameter N and the volume of the space then define the density

of points. In the simplest case the space is Euclidean and the coordinates are obtained using Poisson (PPP) or other random point process, thereby defining an ensemble of RGGs [57].

To connect nodes in RGG, one defines a condition, based on the metric. For instance, two nodes u, v , geometric distance $d(u, v)$ apart, are connected with an edge if $d(u, v) \leq R$, where R is some constant. RGGs can be generalised in multiple ways. Of course, any space equipped with a distance function $d(u, v)$ can be used instead of Euclidean space. RGGs, generated in hyperbolic (negatively curved Riemannian) manifolds seem to be of particular importance. It is claimed that hyperbolic RGGs provide a geometric explanation of common structural and dynamical properties of many real networks, including scale-free degree distributions, strong clustering, community structure, and network growth dynamics [58; 59; 60]. They also give a model underpinning optimal navigation without the global knowledge of the network topology [61]. Furthermore, the node connection condition (e.g. $d(u, v) \leq R$) can also take numerous forms and can be thought of as a probability function. The previously stated condition for the RGG is then a Heaviside step function, yielding “hard” RGGs. Monotonically decreasing probability functions of the connectivity of RGG yield so-called “soft” random geometric graphs [62].

Pseudo-Riemannian manifolds were also studied as spaces of RGGs [57]. Here, the so-called Lorentzian RGGs are *undirected* random geometric networks, generated by first sprinkling nodes onto a Lorentzian manifold using PPP and connecting two nodes with an undirected edge if the nodes are time-like separated. Directed Lorentzian RGGs are known as CAUSAL SETS (causets). In network representation, causets are transitively complete directed acyclic graphs.

2.3.2 Non-geometric Networks

Generally, networks are not “directly” embedded in space but the proximity of nodes can still be important and lead to a natural geometry for a graph. Social networks for instance connect individuals through friendship relations, which in principle happens without any constraints derived from a finite dimensional space. But in this case, geographical space intervenes in the fact that the connection probability between two individuals usually decreases with the distance between them [53].

Without the prior knowledge of the space, appropriate for a network, one may want to find the most suitable, low dimensional space for it. The NETWORK EMBEDDING task addresses this question. Given a network and some similarity (or distance) function between its nodes the goal of network embedding is to find a low dimensional representation of the network nodes in some metric space so that the given similarity (or distance) function is preserved as much as possible [63]. Network embedding is an important tool for network visualisation and understanding of functional properties of networks.

2.3.3 Geometric Networks and Paths

When a network is implicitly linked with a metric space, the relation between geometric paths and paths in networks is apparent. If a network is based on or embedded in Euclidean space, the straight line between points is the geometric geodesic, and the shortest path between them is also the network geodesic [64; 65].

In simple models the longest path has been shown to be the best approximation to the geodesic for models of DAGs embedded in Minkowski space [66] where there is a single time direction. Here the considered graphs are directed Lorentzian RGGs, or causal sets (i.e. transitively complete DAGs). The statement that the

geodesic in directed Lorentzian RGGs is closest to the network longest path was further confirmed numerically [67]. In box spaces [68] which include those built from Poisson Point Processes in Minkowski spaces where all causal nodes are connected to form a network, the longest path also approximates a straight line, although a straight line is only known to be the geodesic in Minkowski space. It is not clear whether this statement holds true in non-transitively complete DAGs. Perhaps in this case, the shortest path is a good representative of a geometric geodesic?

2.4 Geodesics in DAGs

To understand what is a good “DAG geodesic”, we will study paths in those RGGs that yield directed acyclic graphs. In particular, we are interested in the relation between real geodesics of the geometric space and network paths in RGGs. To build a DAG as an RGG, we can use a Lorentzian manifold. Then node similarity relates to spatial proximity of nodes, and the order of edges is imposed by the temporal coordinate. One can define a Lorentzian manifold in which some nodes are space-like separated—form an antichain and are not connected in the DAG. The relation between Lorentzian manifolds and DAGs has been previously used in network embedding [69; 70].

Brightwell and Gregory [66] (basing their work on Bollobás and Brightwell [68]) showed that in transitively complete DAGs, modelled as space-time particles in Minkowski space, the longest network path approximates the geodesic in the continuous Minkowski space. They considered a Minkowski manifold M^D where D is the dimension of the space-time ($D - 1$ spatial dimensions with one temporal dimension). This manifold is further equipped with a causal structure “ \prec ”: $v \prec u$ if $u \in \mathcal{F}(v)$. Nodes are sprinkled on this manifold using PPP of constant density ρ

in a finite space-time volume V . Once nodes are sprinkled onto the manifold, the causal structure induces a causal set—a transitively complete DAG—with space-like separated nodes forming antichains and time-like separated nodes connected. In other words, we connect with edges all causal nodes (for which ds^2 of (2.3) is positive).

We find that the longest network path is a curve in M^D which becomes a geodesic when spacing between nodes approaches 0. The longest path length L scales with the number of nodes, that is $L \propto (\rho V)^{1/D} m_D$, where m_D is some constant (e.g. $m_2 = 2$) [66]. This was first discussed in [68] by Bollobás and Brightwell in terms of box spaces (random partially ordered sets), and related to causal sets by Brightwell and Gregory in [66]. Note that the length of a geodesic should scale with the increased separation between points, a requirement which is not satisfied by the shortest path in this model (as the shortest path is equal to one edge). We will study scaling of the longest path in Chapter 3.

Here we will consider DAGs which are non-transitively complete *directed* random geometric graphs. Note, this definition is different from the Lorentzian random geometric graphs, mentioned in section 2.3.3 where Lorentzian metric was used to obtain transitively complete DAGs of which the directionality was omitted for further analysis. We will study DAGs, generated using PPP in Minkowski space. To ensure that our networks are not transitively complete, time-like separated nodes will only be connected if the square of a distance for time-like separated points does not exceed a certain value. Although this study could be generalised to curved spaces, let us now consider Minkowski space-time, so each network we will generate will be a Minkowski-RGG. An illustration of the model is given in Fig. 2.1.

We will restrict ourselves to cases of $D = 2, 3$. To generate a Minkowski-RGG, we will first add two nodes $u^{\text{source}}, u^{\text{sink}}$, separated by one unit in time and

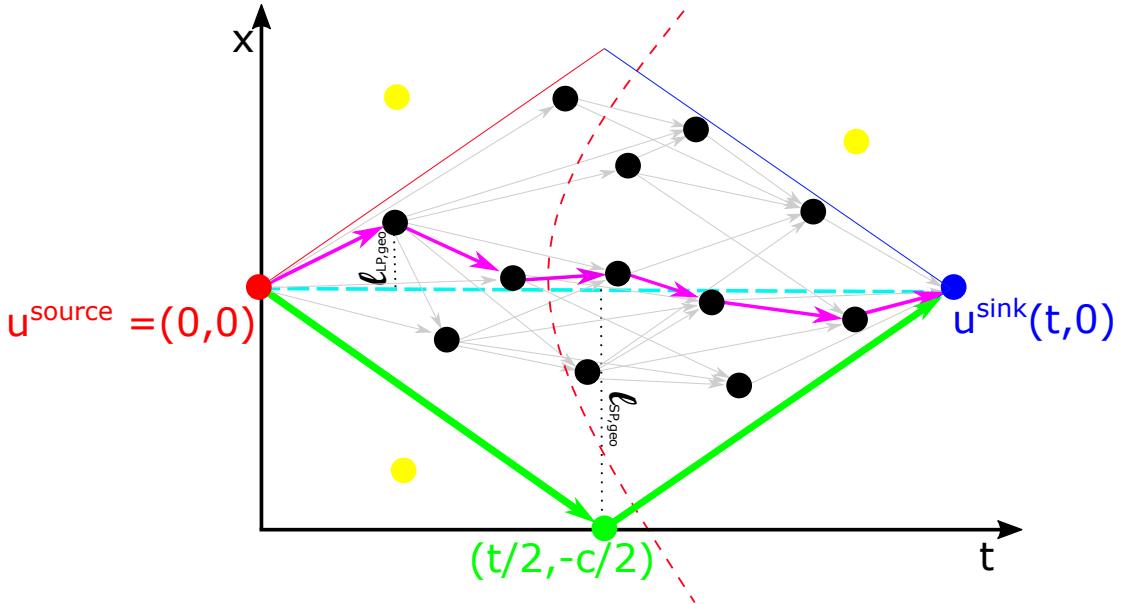


Figure 2.1: An illustration of a Minkowski random geometric graph model in $D = 2$. The ordinate axis corresponds to spatial coordinate of nodes, whereas the abscissa axis corresponds to the temporal coordinate. First, two nodes are added at coordinates $(0, 0)$ and $(t, 0)$, the source node u^{source} and the sink node u^{sink} . The interval between the two nodes will be generated by sprinkling nodes in the space and keeping those, with which both u^{source} and u^{sink} are time-like separated (yellow nodes do not satisfy this condition). The nodes within the interval are those inside the rhombus bounded by $\mathcal{F}(u^{\text{source}})$ and $\mathcal{P}(u^{\text{sink}})$. Once the interval is populated with the required number of nodes, edges are created between those nodes for which (2.5) is satisfied. For instance, the red node will have edges to those nodes which are not farther than the dotted red line. The longest (fuschia) and the corresponding shortest (green) paths are found, and the distance from either to the geodesic (dotted cyan line) are calculated.

by zero in space. Let us choose these two nodes to be placed at points with coordinates $(0, 0)$ and $(1, 0)$ in $D = 2$ and $(0, 0, 0)$ and $(1, 0, 0)$ in $D = 3$. In general, we will say that the point $(0, 0)$ in $D = 2$ ($(0, 0, 0)$ in $D = 3$) is the origin and a node u will be placed at a point located in the space with coordinate $\mathbf{u} = (u_0, u_1)$ in $D = 2$ ($\mathbf{u} = (u_0, u_1, u_2)$ in $D = 3$). The future light-cone of u^{source} and the past light-cone of u^{sink} will intersect at $(0.5, \pm c/2)$ in $D = 2$. In $D = 3$ we can work in polar spatial coordinates to find where future and past light-cones of the source and the sink intersect. Consider x, y as two spatial coordinates. We have $x = r \cos \theta$, $y = r \sin \theta$, for $\theta \in [0, 2\pi]$ and $r \in [0, c/2]$. Then points

lying on the boundary between the two light-cones will be those with coordinates $(0.5, c/2 \cos \theta, c/2 \sin \theta)$ in $D = 3$. Volume of these intervals are $c/2$ in $D = 2$ and $c^2\pi/6$ in $D = 3$ in units of $[L^{D-1}T]$ (L —length unit, T —time unit).

For each value of c we will generate N points, such that the density of points ρ is independent of c , so $N = \rho V$. For the rest of $N - 2$ nodes, we will assign a temporal coordinate $v_0 \in [0, 1]$ and a spatial coordinate between $-c/2$ and $c/2$ such that

$$\begin{aligned}\ell^2(u^{\text{source}}, v) &= c^2(u_0^{\text{source}} - v_0)^2 - \sum_{i=1}^D (u_i^{\text{source}} - v_i)^2 \geq 0 \quad \text{and} \\ \ell^2(v, u^{\text{sink}}) &= c^2(v_0 - u_0^{\text{sink}})^2 - \sum_{i=1}^D (v_i - u_i^{\text{sink}})^2 \geq 0.\end{aligned}\quad (2.4)$$

We will use PPP to find the coordinates. These requirements will ensure that all nodes are placed in an interval between u^{source} and u^{sink} . We will then create edges between those nodes u, v , for which $0 < \ell^2(u, v) \leq \ell_{\max}^2$ for some value of ℓ_{\max}^2 . If

$$\ell^2(u, v) = c^2(u_0 - v_0)^2 - \sum_{i=1}^D (u_i - v_i)^2 \leq \ell_{\max}^2, \quad (2.5)$$

then we will add an edge (u, v) if $u_0 < v_0$ and (v, u) otherwise. When $\ell_{\max}^2 > c^2$, we recover a transitively complete DAG in our interval, while smaller ℓ_{\max} results in DAGs which are less dense. The value of c determines the angle of light-cones: small c results in narrow light-cones (we expect few edges), larger c makes them more open (more edges are expected).

Our hypothesis is that in this model the longest network path approximates the geometric geodesic of the space for various values of D, c, ℓ_{\max}^2 . For visual inspection, let us first consider Fig. 2.2, where examples of Minkowski-RGGs, created using $c = 1, D = 2$ and several ℓ_{\max}^2 values are shown. For each case, the longest path in the network (blue edges), and the shortest path between the first

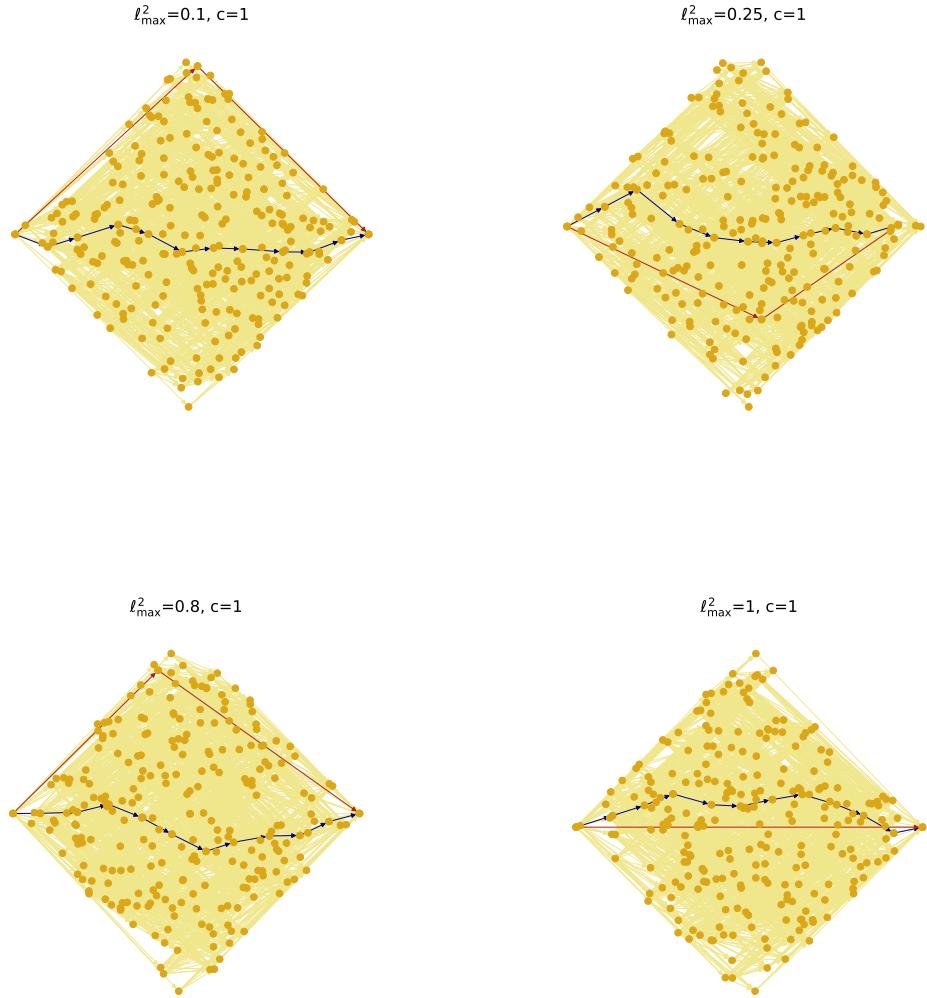


Figure 2.2: Comparison between the longest path (blue edges) in the network and the shortest path (red edges) between the same two nodes in Minkowski-RGGs generated using different ℓ_{\max}^2 . Each network contains 250 nodes, $c = 1$. The horizontal axis represents time coordinate, the vertical axis represents spatial coordinate. Two nodes u, v are connected if $\ell(u, v)^2 \leq \ell_{\max}^2$. For clarity, edges that do not feature in either the longest path or the corresponding shortest path are transitively reduced [31; 71]. This figure shows that in transitively incomplete random geometric DAGs, the shortest path deviates from the straight line much more than the longest path. The maximum distance was taken as the largest distance between a coordinate of any node in the path and the straight line joining the end points of a path.

and the last nodes in the longest path (red edges) are highlighted (the rest of the edges in the network are transitively reduced [31; 71] for clarity). In Minkowski

space the straight line between the two nodes is a true geometric geodesic. It is clear from the figures that for all but $\ell_{\max}^2 = 1$ case, the shortest path diverges from the straight line, whereas the longest path does so to a lesser extent. Note that we take *a* longest path and *a* shortest path, without any preference. It could be that more than one path of the same length exists between the same two nodes in the network, one of which is closer to the geodesic than another. However, in the following, we will sample these paths from many Minkowski-RGGs, so an average value of the distance between the network path and the straight line should be representative of the true mean value.

To quantify the disparity between the geodesic of the Minkowski space and the shortest or the longest paths in Minkowski-RGG, we proceed as follows. We expect that for a path which is close to a geodesic, all points lie close-by the straight line, connecting the two nodes. Then the largest perpendicular distance from any point in the path to a line defining spatial geodesic says in how big of a volume the path is contained in [65]. Ideally it is equal to zero and for non-geodesic paths we expect the volume to be bounded only by the volume of the interval $\mathcal{I}[u^{\text{source}}, u^{\text{sink}}]$. So we calculate the perpendicular distance from the most distant point on a path to the geodesic line and call this the quality of the approximation of the geodesic path by the path in the network. Given the start and end nodes at a network path, u^{start} , u^{end} and the node v , farthest from the straight geometric line connecting nodes u^{start} , u^{end} , the perpendicular distance to the line from u is calculated considering their coordinates in space as follows [72; 73]

$$\begin{aligned}\ell &= \frac{|(u_0^{\text{end}} - u_0^{\text{start}})(u_1^{\text{start}} - u_1) - (u_0^{\text{start}} - u_0)(u_1^{\text{end}} - u_1^{\text{start}})|}{\sqrt{(u_0^{\text{end}} - u_0^{\text{start}})^2 + (u_1^{\text{end}} - u_1^{\text{start}})^2}} \text{ in } D = 2, \\ \ell &= \frac{|(\mathbf{u}^{\text{end}} - \mathbf{u}^{\text{start}}) \times (\mathbf{u}^{\text{start}} - \mathbf{u})|}{|\mathbf{u}^{\text{start}} - \mathbf{u}^{\text{end}}|} \text{ in } D = 3.\end{aligned}\quad (2.6)$$

Here $\mathbf{u} = (u_0, u_1)$ are Euclidean coordinates of a node at point u in $D = 2$ and

$\mathbf{u} = (u_0, u_1, u_2)$ in $D = 3$. Let us call the largest distance from the longest path to the straight line $\ell_{LP,geo}$ and the largest distance from the shortest path to the straight line $\ell_{SP,geo}$.

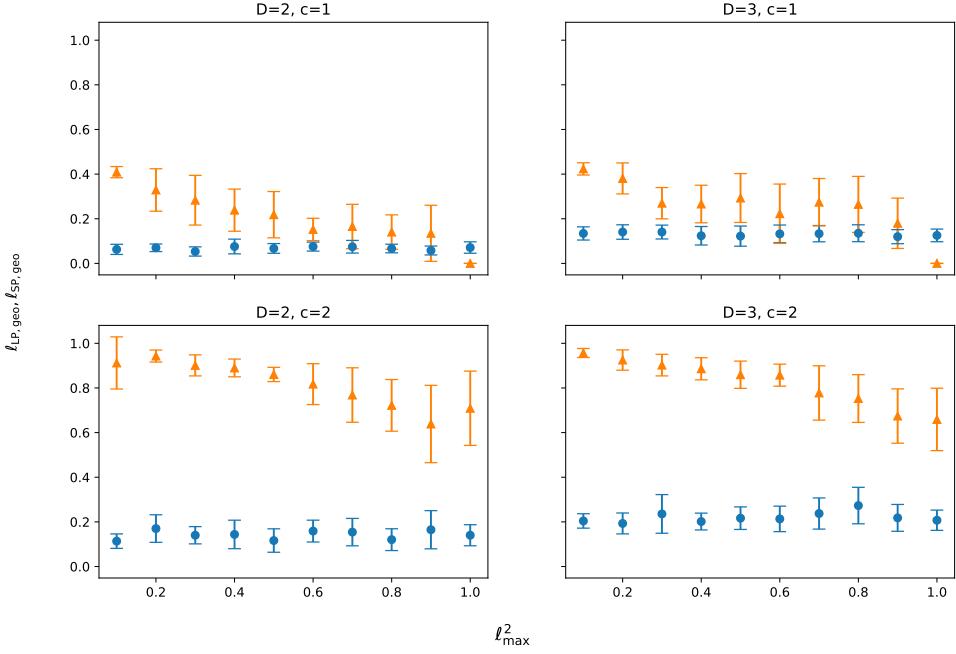


Figure 2.3: Largest distance from the straight line for the network longest path (blue circles) and a corresponding shortest path (orange triangles) for Minkowski-RGGs with $\rho = 1500$ and various dimension D and cutoff distance squared ℓ_{\max}^2 parameters for speed of light $c = 1, 2$. Errors are given as the standard deviation of obtained values.

Fig. 2.3 shows that the longest path is close to the straight line—the geodesic—for various values of D, c, ℓ_{\max}^2 . It is clear from the figure that this distance is a constant of approximately 0.1 for the longest path and seems to increase with dimension. However, $\ell_{SP,geo}$ depends on ℓ_{\max}^2 : as expected, for small values of ℓ_{\max}^2 , $\ell_{SP,geo}$ approaches the values, bounded by the size of the interval, and for large values of ℓ_{\max}^2 it approaches the minimum of 0, always obtained when the network is transitively complete.

Fig. 2.4 shows that the deviation from the geodesic path for each of the two network paths increases with c . For small values of c , the network is transitively

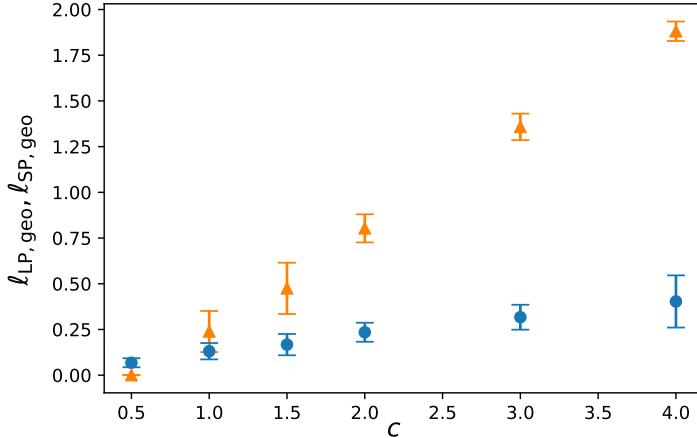


Figure 2.4: Largest distance from the straight line for the network longest path (blue circles) and a corresponding shortest path between the first and the last nodes on the longest path for a model of Minkowski-RGG networks with $\rho = 1000$ nodes, $D = 2$, $\ell_{\max}^2 = 0.8$, and c varying from 0.5 to 4. The network is transitively complete when $\ell_{\max}^2 > c^2$, so for those values of c , the largest distance from the shortest path to the geodesic is 0, as it is simply an edge. For larger values of c , when network is not transitively complete, one can see that the farthest point on the longest path comes closer to the geodesic than the farthest point on the corresponding shortest path. Errors are given as the standard deviation of obtained values.

complete, so the shortest path is simply an edge between the source and the sink, however when the network is not transitively complete, the longest path is closer to the geodesic than the shortest path. For larger values of c the network would become disconnected, so such values of c are not considered.

2.5 Discussion

Geometry has recently caught attention of network scientists, who seek geometric interpretations of non-geometric networks. Space is often interlinked with and intrinsic to the topology of the network, and when it is not (for instance, in hyperbolic space network models [58]), it explains features of observable networks. Network can in principle be thought of as a space itself, with nodes as spatial

points. However, the spaces of networks are complex, just as networks themselves. If a network is a land, it is filled with valleys (which are easy to pass) and hills (which are difficult to climb). Projecting a multi-dimensional terrain to a 2-dimensional map we can lose the information about the hills and valleys, so it is sensible to question the quality and reason behind our projection.

This chapter has served as an invitation to question the meaning of a network geodesic. Often, without discussion, the shortest path between nodes is called a geodesic. When the network is undirected and there is no knowledge of the space, underlying the network, calling the shortest path a geodesic seems natural. If we do not have any further information about the network terrain, by definition, there is no faster way to reach u from v , so the network shortest path must be a geodesic in a formal sense. When we generate an RGG based on Euclidean or other metric space, we find that the space-defined geodesic is only approximated by a network path, as was shown in e.g. [65] and discussed in this chapter. In a way, we do not want the network path to be an exact replica of a spatial geodesic, unless our goal is to recover a continuous space with our discrete network (as is the goal in [66]).

The main result of this chapter is that in a Minkowski-RGG, which is a non-transitively complete DAG, the longest path is a better approximation of a geodesic, defined by Minkowski space. Although the relation between the longest path of the spatial DAGs and the geodesics in the Lorentzian spaces is known, it has only been proven for transitively complete DAGs. Here we saw that the longest path lies close to the geodesic in non-transitively complete DAGs as well. On the contrary, the shortest path deviates from a straight line. We observed this result in many random networks, obtained using a range of parameter values. The results also suggest that the deviation of a network path from a geometric geodesic is independent of the parameter ℓ_{\max}^2 value for the longest path. However,

the shortest path seems to approach the straight line for large values of ℓ_{\max}^2, c . It is important to note that here we considered the largest deviation of a network path from a geodesic. It is possible that if we measured the cumulative sum of perpendicular distances from all points on a network path to the line defining a spatial geodesic, we would observe the shortest path as that closer to the geodesic, not the longest path. Testing this hypothesis is left for future work.

Our work, presented in this chapter poses more questions that provides answers. What is the natural space-time for any given dataset with an order? Is such a space-time locally (globally) curved and if so, how? What are the implications for the analysis of DAGs given the importance of the longest path shown in this chapter? We will look for answers to some of these questions in the following chapters.

Chapter 3

The Longest Path in the Price Model

This chapter is based on work presented in:

The Longest Path in Price Model,

T. S. Evans, L. Calmon, V. V.

To be published in Scientific Reports,

V. V. contributed to data collection and processing,

as well as interpreting the results and writing the manuscript.

The Price model [74; 38], previously discussed in section 1, is one of the oldest network models, motivated by the pattern of citations in academic papers. Recall that in a citation network, each node represents a document while every entry in the bibliography of a document t is represented by a directed edge from an older document, node s , to node t . The arrow of time thus ensures our network of documents is a DAG: bibliographies can only refer to older documents.

The previous chapter explored the relation between the longest path and geodesics. Here we will analyse the properties of the longest path in a model in

citation networks, and exploit the mathematical simplicity of the Price model to understand the mechanics of the longest path in this particular network model.

Although some properties of the longest path, such as its scaling with the network size, have been investigated in transitively complete networks [68], it is not clear whether the longest path behaves similarly in citation networks that are not transitively complete, nor transitively reduced DAGs. The limited size of a bibliography means no document ever cites every older paper to which it has some connection. We will begin by outlining the analytic results. We will then compare the analytical result to numerical simulations and summarise our findings.

3.1 Analytic Result for Scaling of the Reverse Greedy Path

The Price model was first introduced in section 1.3.3. Given the probability of connecting to a node s at time $t + 1$, (1.3), we can further simplify this expression by assuming that $E(t) = mt$ and $N(t) = t$ (this is a reasonable approximation for large values of t). We will use $\Pi(t, s)$ to derive the expression of scaling of the so-called REVERSE GREEDY PATH, which can be thought of as a lower-bound approximation of the longest path. This path is defined in opposite direction to the direction of edges. The reverse greedy path arriving at a vertex t is defined by following the edge arriving at that vertex t which links back to the most recent predecessor of t , vertex s . Iteratively, for vertex s the reverse greedy path follows the edge (against its direction) to the most recent predecessor of node s . The process is continued until the global source node is found, see Fig. 3.1 for an illustration. Let us denote the length of the greedy path of a network with t nodes $\ell(t)$, whereas the length of the longest path will be $L(t)$.

Let us sketch the algebraic solution for the reverse greedy path scaling in

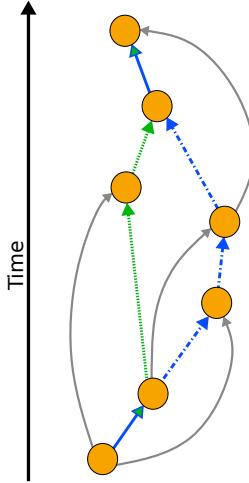


Figure 3.1: Example of the reverse greedy path in green and the longest path in blue. The two paths are distinct in general. Although the reverse greedy path can in principle be of the same length as the longest path, it is usually shorter. The length of the greedy path is the lower bound for the length of the longest path. Figure reproduced from [39].

the Price model. Full details are provided in [39]. First, one needs to define the cumulative probability function which is the probability that all m edges from the node created at time $(t + 1)$ are attached to a node created at time s or earlier

$$\Pi_{\leq}(t, s) = \begin{cases} \sum_{r=1}^s \Pi(t, r) & \text{if } t \geq s \geq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3.1)$$

One can see the $\Pi_{\leq}(t, s)$ as the probability to connect to an “effective node”, which represents all nodes created between the initial time to time s . Its degree is furthermore the degree of all such nodes, and edges, pointing from nodes in the effective node to other nodes in the effective node represent the self-loops of it. To find the scaling of the reverse greedy path, we need to know $\Pi_{\max}(t, s)$, the probability that of the m predecessor nodes connected to a new node at $(t + 1)$, the oldest of them is $s = \max(\mathcal{P}(t + 1))$. The probability that we connect a new node $(t + 1)$ to nodes of age s or less is simply $(\Pi_{\leq}(t, s))^m$ where $\Pi_{\leq}(t, s)$ is the

cumulative probability of attachment of (3.1). This gives us that

$$\begin{aligned}\Pi_{\max}(t, s) &= (\Pi_{\leq}(t, s))^m - (\Pi_{\leq}(t, s-1))^m \\ \text{for } t \geq s \geq 1.\end{aligned}\tag{3.2}$$

Then the master equation for the length of the reverse greedy path is

$$P(\ell, t+1) = \sum_{s=1}^t P(\ell-1, s) \Pi_{\max}(t, s).\tag{3.3}$$

The master equation has a solution of the form [39]

$$\begin{aligned}\lim_{t \rightarrow \infty} \langle \ell(t) \rangle &= m\bar{p} \ln(t) - m\bar{p}\psi(m\bar{p} + 1) \\ &+ \sum_{n=2}^m \binom{m}{n} (-1)^{n-1} (\bar{p})^n \zeta(n) + O(t^{-1}).\end{aligned}\tag{3.4}$$

Here $\psi(z)$ is the digamma function and $\zeta(n)$ is the Riemann zeta-function. The solution suggests that the length of the greedy path (and so the longest path) will scale as the logarithm of the size of the network.

3.2 Numerical Methods

To confirm the analytical result, (3.4), we studied the Price model numerically. The Price model is straightforward to implement, however there are some subtleties related to its implementation.

The details of the algorithm are provided in Appendix C. For completeness, here we sketch the algorithmic procedure qualitatively. The algorithm begins with an initial graph. This step is essential, as otherwise equation (1.3) is undefined. In theory, any directed acyclic network could be used as an initial graph. However, one may want to consider the effect of the initial graph choice for the results, and

minimise the initial bias as much as possible. Our choice for the initial graph was a transitively complete DAG with $(2m + 1)$ nodes. Of course, (3.4) does not relate to our network at this point, as a transitively complete DAG is not a graph, generated using the Price model. However, this equation is not supposed to be accurate in small networks, and this initial graph is assumed to have a weak effect on the asymptotic behaviour of the scaling of the longest path in the long time limit. The initial graph effect will be discussed in more detail in section 3.2.4.

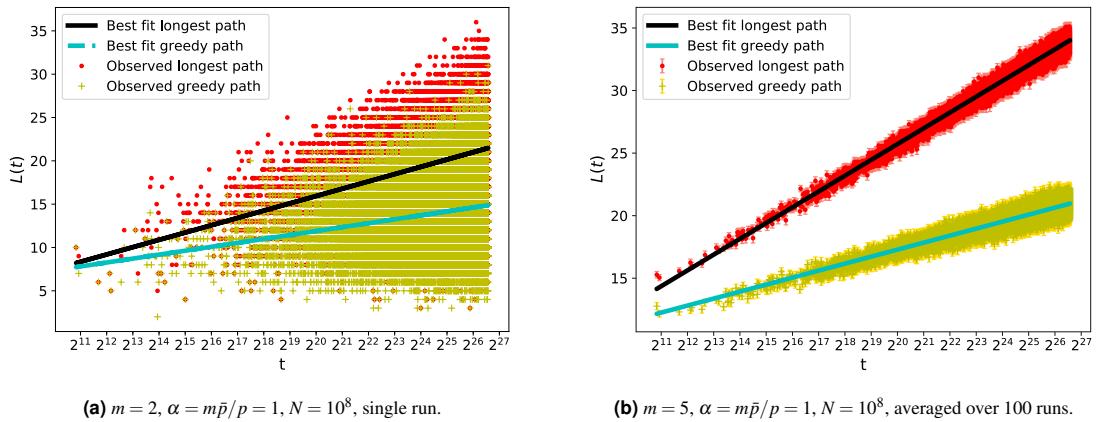


Figure 3.2: On the left is a plot of path length against t for a single network realisation showing how noisy the data is. Averaging over 100 runs greatly reduces the fluctuations as shown in the righthand plot. Fitted lines are of the form $a \ln(t) + b$. In both figures, a random sample of 10^5 points is plotted. The abscissa axes are logarithmic to show the linear behaviour between the path lengths and t . Figure reproduced from [39].

After the initialisation, the network is grown sequentially. At each time step t , a node t is added to the network, and new edges are placed from older nodes to the node t , according to (1.3). The lengths of the greedy path and the longest path at time t are recorded and the new iteration step begins. It is easy to track the lengths of the two paths at each time step. For the greedy path, one notes the lengths of the greedy paths of the nodes the node t drawn edges to, the largest value is $\ell(t) - 1$. For the longest path, one must check if one of the nodes t connected to, also has the largest value of the longest path length. If so, this length is $L(t) - 1$.

The results, obtained from a single network are noisy, as the exemplary Fig. 3.2(a) shows. However, despite the large fluctuations, there is a clear positive relation between t and the length of the longest and greedy paths. To obtain statistically significant results, we looked at the path length distributions, averaged over 100 runs. Fig. 3.2(b) shows greatly reduced fluctuations and a clear linear relation between path lengths and the logarithm of the size of the network, confirming the correctness of the analytic result. We further confirmed this for a wide range of parameter values, as will be discussed in the following sections.

3.2.1 Fitting

In order to compare our numerical data with the analytical results we fitted the path lengths found to the function $f(t)$ where

$$f(t) = a \ln(t) + b. \quad (3.5)$$

The fit was made by using a fitting routine based on the optimisation of the chi-squared measure of goodness of fit (for instance see [75]).

For each set of parameter values $p, m, N, R = 100$ networks were generated. Furthermore, for computational convenience, we used such values of p, m that $m\bar{p}/p = \alpha$ is an integer. From each network, the data of the reverse greedy path and the longest path at each t was collected. Note that strictly this makes our data points not entirely independent. We also assumed that the data points at each t are normally distributed—this feature is an essential prerequisite of the chi-squared statistics. However, it is strictly not possible for an integer-valued data such as ours to be normally distributed. Yet, as the Fig. 3.3 shows, Gaussian distribution with the same mean and standard deviation as the data approximates our data well. Let us call a path length (either greedy or the longest), recorded at

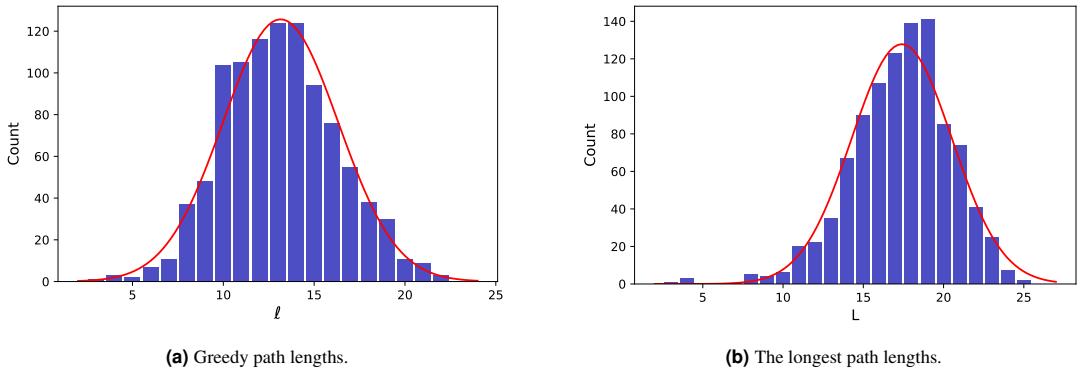


Figure 3.3: Distributions of the reverse greedy path length (left) and longest path lengths (right) for a node with index $t = 10^5$ in the Price model with $m = 5$, $c = 1$ and 10^6 total number of nodes, averaged over 1000 networks. For comparison, a Gaussian is plotted with the same mean and standard deviation as each data set. We can see that a normal distribution, while not perfect, is a sufficiently good description of the distribution to justify our use of Gaussian-based fitting statistics. Figure reproduced from [39].

run r for node t , y_{tr} . To evaluate the quality of our fit, parameters that minimise the value of the chi-squared function [75]

$$\chi^2 = \sum_{t,r} \frac{(y_{tr} - f_t)^2}{(\sigma_t)^2} \quad (3.6)$$

have to be found. The errors on the parameter values were taken from the covariance matrix produced by the method. The fitting function f_t used here is $f_t = a \ln(t) + b$ but we also tried $f_t = a \ln(t) + b + (c/t)$ where a, b, c are constants to be found by fitting, see section 3.2.3 for discussion.

For numerical convenience (minimising memory requirements), for each of the path length we kept track of two values for each node occurring at t , the total sum of values T_t and the sum of squares S_t , as given by

$$T_t = \sum_r y_{tr}, \quad S_t = \sum_r y_{tr}^2. \quad (3.7)$$

We do not expect the fit to be valid for small system sizes. For instance, in most of our work the initial graph is a complete DAG (see section 3.2.4 for a

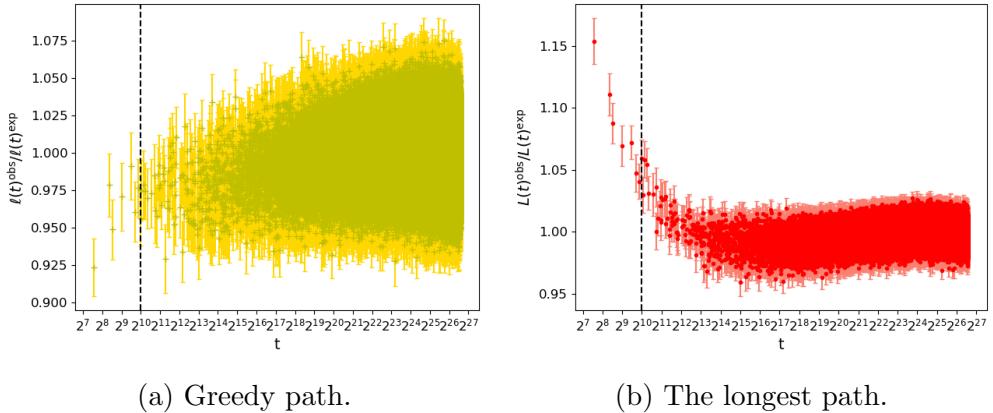


Figure 3.4: The ratio of the average over 100 runs of the path length measured numerically, L_{obs} and ℓ_{obs} , divided by the expected values, as described by the numerical best fit. The data for $p = 0.625$ $m = 5$ from $t = 1000$ (represented by a vertical dashed line) to $t = 10^8$ was fitted to (3.5). The reverse greedy path results are shown on the left, and the longest path are shown on the right. The error bar on each point is calculated from the standard error in the mean from the results for each node over 100 runs. A random sample of 10^6 points is plotted in both figures.

discussion of initial graph effects) in which the path lengths scale as $y_t = L(t) = \ell(t) = t$. To deal with this we fit our data from time t_0 , an additional parameter in our fit, to the largest time which is equal to the number of nodes N . So, in terms of the totals and the squares of totals, our chi-squared function becomes

$$\chi^2 = \sum_{t=t_0}^N \frac{S_t - 2T_t f_t + R(f_t)^2}{(S_t/R) - (T_t/R)^2}. \quad (3.8)$$

Under our assumptions of normal fluctuations and independence as noted above, the probability of obtaining a particular χ^2 value is given by the integral of the complementary cumulative distribution function of the χ^2 distribution for the given degrees of freedom. The number of degrees of freedom in the model is equal to $(N - t_0 + 1)R$. t_0 parameter was varied between 100 and 10^4 but this cutoff parameter t_0 had no significant influence on the resulting fits. This is to be expected, as even the largest t_0 value considered, 10^4 , constitutes a mere 0.01% of the data we were using. So, in our work we used a fixed value of $t_0 = 10^3$. As

Fig. 3.4 shows, there is a deviation between the numerical and analytic values for small t , however the number of data points in this regime is small, so in practice the effect on the overall quality of the fit is weak.

3.2.2 Leading and Next-to-leading Order Coefficients

The dependence of the coefficient of the $\ln(t)$ term found from the fit, a , on the model parameters is shown in Fig. 3.5 and Fig. 3.6. The abscissa in these figures, $m + p$ was chosen for visualisation purposes. In the range $[m, m + 1)$ lie all points corresponding to networks where the parameter m is the same and p varies. Fig. 3.5 shows that for a wide range of parameter values m, p the observed coefficient a_{gr} agrees with the theoretically expected value of $m\bar{p}$, whereas for the longest path, the observed leading order coefficient is around double the expected coefficient. Fig. 3.6 further shows the relation of the leading order coefficient, found for the scaling of the greedy path length, and for the longest path length. The coefficient for the greedy path length is always smaller than the coefficient for the longest path length scaling. For the parameter values considered, the longest path seems to be around twice as long as the reversed greedy path.

The next-to-leading order coefficient, b of (3.4) did not reveal as clear trends as the coefficient a of the same equation. There does not seem to be an obvious relation between b_{max} and b_{gr} , as Fig. 3.7 shows.

3.2.3 Fit to a Non-linear Function

We also considered a fit with a term of c/t added to the expression in the (3.5), that is $f(t) = a \ln(t) + b + (c/t)$. We found that in practice, this term had little influence on the remaining parameters of interest, namely, a and b . Furthermore, the errors in c were found to be relatively large in comparison to the errors of the

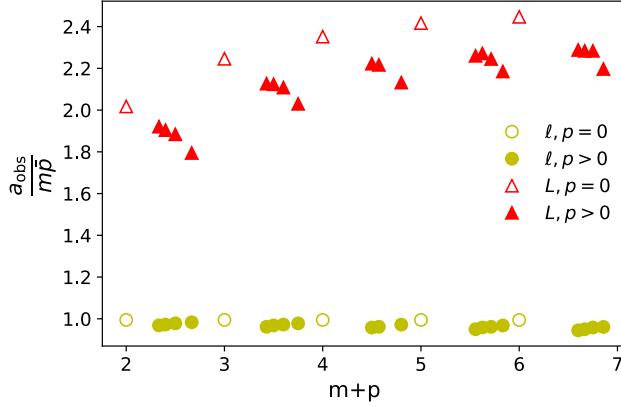


Figure 3.5: The ratio of $a_{\text{obs}}/(m\bar{p})$ where a_{obs} is the coefficient of $\ln(t)$ derived from the best fit of the numerical path length data to $a \ln(t) + b$ (3.5) while $m\bar{p}$ is the analytical prediction for the value of a when looking at the length of the reverse greedy path. The red triangles show the results for the longest path value of a while yellow circles are the reverse greedy path values. These values were obtained by fitting the form to nodes created between $t = 1,000$ and $t = 10^8$ from 100 realisations. Errors on the fitted values of a were smaller than the marker size and are not shown.

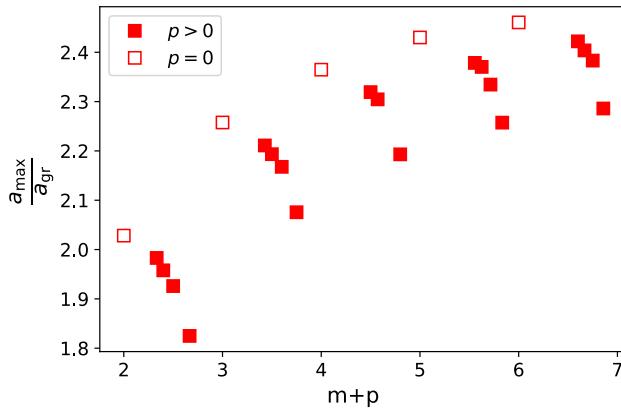


Figure 3.6: The ratio of $a_{\text{max}}/a_{\text{gr}}$ where a is the coefficient of $\ln(t)$ in the best fit of the numerical path length data to (3.5), a_{max} for the longest path data and a_{gr} for the reverse greedy path data. These values were obtained by fitting the form to nodes created between $t = 1,000$ and $t = 10^8$ from 100 realisations. As a result the errors on the fitted values of a , as estimated from the covariance matrix of the linear fitting algorithm, were smaller than the marker size and so these are not shown.

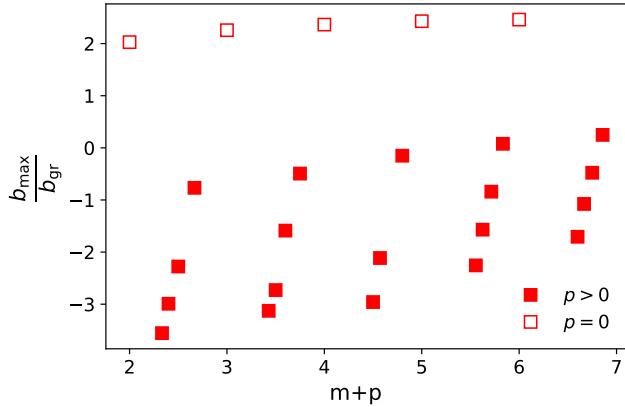


Figure 3.7: The ratio of b_{\max} and b_{gr} where b is the next-to-leading order coefficient in the best fit of the numerical path length data to $a \ln(t) + b$: b_{\max} for the longest path data and b_{gr} for the reverse greedy path data. These values were obtained by fitting the form to nodes created between $t = 1,000$ and $t = 10^8$ from 100 realisations. The errors on the fitted values of b , as estimated from the covariance matrix of the linear fitting algorithm, were smaller than the marker size so the uncertainties are not shown.

parameters a , see Table 3.1.

3.2.4 Initial Graph Effect

Although in this work, all simulations were performed with a single type of initial graph (a transitively complete DAG with $2m + 1$ nodes), $\mathcal{G}_{\text{compl}}$, here we look at the importance of this choice on the obtained results. Due to the choice of the initial graph, we always have exactly m more edges than nodes which is an assumption in the analytical work. A possible drawback is that this initial graph has long path lengths, $\ell(t) = L(t) = (t - 1)$ for nodes in this graph. As both the longest and reverse greedy paths will start with paths in the initial graph, these long initial graph path lengths will produce a significant addition to those we measure. For instance, for $m = 5$ this initial graph contributes up to 10 to any path we measure while the typical length scales we measure at late times are less than 100 in general.

m	α	$\chi^2_{\text{gr,lin}} - \chi^2_{\text{gr,nonlin}}$	$\frac{\chi^2_{\text{gr,lin}} - \chi^2_{\text{gr,nonlin}}}{\chi^2_{\text{gr,lin}}} \times 10^7$	$\chi^2_{\text{max,lin}} - \chi^2_{\text{max,nonlin}}$	$\frac{\chi^2_{\text{max,lin}} - \chi^2_{\text{max,nonlin}}}{\chi^2_{\text{max,lin}}} \times 10^7$
2	1	1914.679	1.8958	52115.2	51.6
2	2	5500.539	5.4468	47292.36	46.9
2	3	10679.31	10.574	32425.11	32.1
2	4	15793.18	15.638	11345.38	11.2
3	1	4128.344	4.0881	53148.67	52.7
3	2	10891.1	10.784	30573.03	30.3
3	3	18021.77	17.849	14582.06	14.5
3	4	29150.27	28.865	535.4066	0.531
4	1	6381.005	6.3185	35694.73	35.4
4	2	15188.28	15.041	13707.66	13.6
4	3	30406.43	30.109	10.1752	0.0101
4	4	41108.91	40.72	3412.088	3.39
5	1	7572.032	7.4983	31364.85	31.1
5	2	23749.52	23.52	3088.395	3.06
5	3	36774.17	36.424	1128.841	1.12
5	4	66946.58	66.314	47772.77	47.3
6	1	11489.57	11.378	22020.32	21.8
6	2	26992.08	26.731	149.04	0.148
6	3	61386.35	60.782	45868.17	45.4
6	4	89429.041	88.563	93100.79	92.2

Table 3.1: Absolute and relative difference between chi-squared values obtained by fitting path data to a linear function, $f(t) = a \ln(t) + b$, χ^2_{lin} and to a non-linear function $f(t) = a \ln(t) + b + (c/t)$, χ^2_{nonlin} . The non-linear fit results in smaller χ^2 values, but the improvement is marginal. Table reproduced from [39].

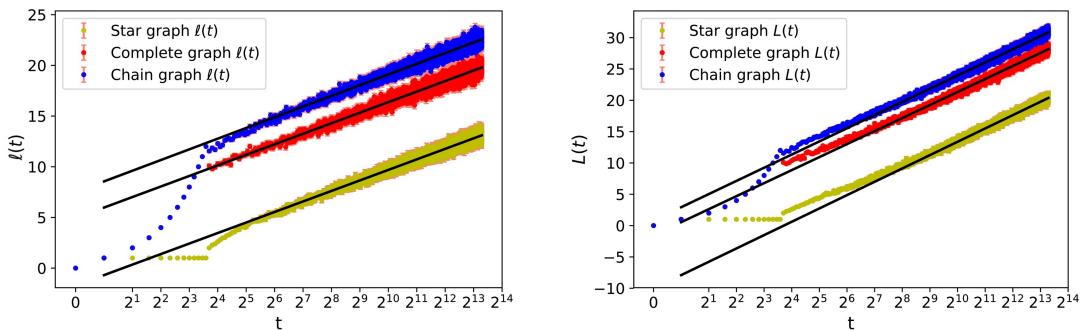


Figure 3.8: Path length (greedy on the left and the longest on the right) vs. $\ln(t)$ for 100 runs with $m = 6$, fitness of 2, $N = 10^5$ using different initial graphs, all composed of $2m + 1$ nodes. Fitted lines are of the form $a \ln(t) + b$. The fit parameter a seems to be unaffected by the initial graph, whereas the fit parameter b varies significantly, see Table 3.2. This causes the slope to remain stable but shifts the graphs vertically. The fit was obtained for nodes created from $t = 1,000$ onwards. Figure reproduced from [39].

A transitively reduced version of the $\mathcal{G}_{\text{compl}}$ is another potential choice for the initial graph. It is a single path, or a chain, thus let us call it $\mathcal{G}_{\text{chain}}(t)$. This type of initial graph is as sparse as possible, all nodes have $k^{(\text{in})}$ and $k^{(\text{out})}$ equal to zero or one, but this initial graph has the same long path lengths as the complete graph. The reason the chain and the complete graph lead to different behaviour in the shift in the average path length at long times is because of their different initial degree distributions which alters the pattern of attachment at early times. That in turn alters the likelihood that the paths we measure join the initial graph at a particular node. We expect that the paths we measure in models starting with the complete graph are more likely to contain a smaller fraction of the initial graph as the earliest nodes will have higher out-degrees but add shorter path lengths to our measurements. By way of contrast, in the chain, most nodes have the same initial degree, they are likely to have similar degree over time so the paths we measure are more likely to leave the initial graph at a later node so giving a longer contribution to the path length coming from this initial chain graph.

Path type	Initial graph	a	b
Greedy	Star	1.4973 ± 0.0052	-0.688 ± 0.04417
Greedy	Complete	1.5001 ± 0.0038	5.9741 ± 0.0309
Greedy	Chain	1.5213 ± 0.0035	8.5548 ± 0.0290
Longest	Star	3.0701 ± 0.0056	-7.9019 ± 0.0476
Longest	Complete	2.9970 ± 0.0034	0.5458 ± 0.0279
Longest	Chain	3.0303 ± 0.0034	2.9345 ± 0.02789

Table 3.2: Variation in obtained fitting parameters for the greedy path scaling and the longest path scaling when using various initial graphs in the Price model with $m = 6$, $\alpha = 2$. The data from $t = 1,000$ to $t = 10^5$ was fitted to the form $a \ln(t) + b$. The initial graph seems to have a small effect on the slope, parameter a , but causes significant changes in the intercept, parameter b . Table reproduced from [39].

The last initial graph we studied was a “directed star graph” $\mathcal{G}_{\text{star}}(t)$, where a single global source node points to the remaining nodes, which in turn are not pairwise connected. Now the degree distribution is highly skewed, with the

central node going to dominate attachment through the cumulative advantage mechanism. However, connecting to one of the other nodes only adds one to the paths we measure so we expect our models using this initial star graph will have the smallest paths lengths of the three cases considered.

As Fig. 3.8 shows, the results obtained using either of the three studied variants of the initial graph are distinguishable. The asymptotic scaling of the path lengths as $a \ln t$ remains the same with the same value of the parameter a . However the constant contribution, b of (3.1), seems to depend on the choice of the initial graph. As Fig. 3.8 and Table 3.2 show, the b coefficient behaves exactly as suggested above: the star graph gives the shortest path lengths while the chain gives the longest.

3.3 Conclusions

In this chapter, we considered a directed preferential attachment model, the Price model of citation networks. As it is a directed acyclic graph, the longest path is well defined. Here we showed that the length of the longest path scales as the logarithm of the size of the network, more precisely, asymptotically as $m\bar{p}\ln(t)$ for a wide range of parameter values. The analytic result agrees with the numerical result within the margin of error.

The reverse greedy path length is a lower bound on the length of the longest path so it is no surprise that the longest path length also scales as $\ln(t)$ with a coefficient, a_{\max} , which is larger than the corresponding scaling factor for the reverse greedy path length, a_{gr} . We found that the ratio a_{\max}/a_{gr} is approximately equal to two, indicating that the longest path in the Price model scales twice as fast with the network size, as the reverse greedy path.

In network models, the directionality of edges is often ignored, however,

whenever the nodes and edges are added sequentially to the growing network, and edges respect the arrow of time, we obtain a DAG, with special properties such as well-defined longest paths. Future studies may look into other statistics, not only the length of the longest path in the Price model, e.g. the distribution of all longest paths. Other random networks may also be studied. In effect, any generative network model could be a model of random DAGs, as was already discussed in section 1.3.3. Studies of the longest path scaling in alternative DAG random models is another future direction.

Chapter 4

Centrality and Directed Acyclic Graphs

4.1 Introduction

Nodes in a network are equivalent in terms of their internal structure: each node represents the same type of entity (in the simplest type of networks). However, positional importance of nodes is far from equivalent: some nodes are in structurally important (central) parts of a network, and some are positioned in peripheral areas. CENTRALITY is a function that evaluates node's positional importance with respect to the topology of the rest of the network.

Centrality is one of the oldest, primary tools of network analysis. Bavelas was first to coin the term “centrality”, remarking in his experiments that centrality of a social network is related to group efficiency in problem solving [76]. The relation between centrality and other group processes was thoroughly studied in the following decades by sociologists [77; 74; 78], later adapted to other sciences. Centrality has found endless applications and is used to study numerous social, technological, transport, biological systems. To name just a few examples, central-

ity is a useful tool to find most relevant items in recommendation systems [79], or to understand key participants in protein-protein interaction networks [80]. Centrality is also used in transportation networks, e.g. to understand traffic and congestion [81; 82]. Furthermore, it seems to be linked to socioeconomic indicators in air transport networks [83]. Centrality is also used to find vulnerabilities in different types of criminal organisations [84].

There is no strict definition of centrality. When a node is called central, it is assumed to be in a structurally important part of a network. However this importance manifests differently in different contexts. So it is not surprising that centrality is also evaluated in various ways. Which centrality measure to use for a given network then depends on what defines an important region in a certain network. One measure can be useful to evaluate importance in one network, but it can have no meaning in another.

For instance, consider a citation network. Recall that an edge indicates a citation and can only point from an older paper to a more recent one. In this picture, out-degree gives a citation count of a node and in-degree is equal to the length of paper's bibliography. In bibliometric studies, a count of paper's citations is considered an indicator of its success or impact [85]. Conversely, to use the length of a bibliography would be a far more controversial quantifier of paper's significance. As the length of bibliography reflects on the amount of resources used by the authors, it can vary vastly based on the research field and the type of paper (e.g. whether it is a letter or a review). So centrality reflecting upon importance in one context may be less insightful in another.

An important feature of centrality that will be studied in this chapter is that centrality creates an order of nodes. Centrality values of nodes are real numbers which can be used to order nodes from the most central to the least central. Generally, this order depends only on the topology of the network. DAGs are

unique in that one order of nodes is quintessentially a part of the network topology itself (the topological order of nodes, see section 1.3.1).

As we will see, in many cases, centrality scores are correlated with the intrinsic topological order of nodes: old nodes are deemed to be indicated as central, regardless of whether they are structurally peripheral or positioned in a core area. So tension between the two types of node orders exists: in some cases, a node is “central” because it is important, but other nodes are “central” because they are old but not necessarily important.

We will begin this chapter with a review of relevant centrality literature and will define centrality as a function with specified features. We will then create axioms for a function to be a centrality measure for DAGs.

To understand the relation between the two orders of nodes in DAGs, we will concentrate on a quintessential feature of centrality function, namely, the NEIGHBOURHOOD INCLUSION PREORDER and its behaviour in directed networks. We will also show that in DAGs there are some cases when neighbourhood inclusion preorder need not exist, yet we can create a centrality order. It means that centrality order reflects on topological order of nodes. Thus we demand a second requirement for a useful DAG centrality measure, namely, that it also cannot be trivially related to topological order.

To make use of centrality in DAGs, we will propose several methods. First, we will discuss an information-theoretic measure based on similarity of partial orders. Such tool may be used to understand whether a node, indicated as central, is expected to be central given its index in the topological order. We will also look into several variations of centrality, such as centrality based on the longest paths, future and past light-cones.

4.2 Requirements for Centrality

Over the years, network centrality acquired many meanings and definitions. To make sense of this multitude of measures requires their classification and description of its key properties. In this section we will review work done in this area.

It is sensible to have many centrality measures, so that we would always have at least one centrality measure which has a meaning in our network. However, an ever-growing number of measures makes one sceptical about the need to continuously “improve” centrality. Very early in the formation of the topic of network centrality, Freeman pointed out that “several measures are often only vaguely related to the intuitive ideas they purport to index, and many are so complex that it is difficult or impossible to discover what, if anything, they are measuring” [86].

Perhaps a formal definition of centrality is needed. Sabidussi was first to formalise centrality by stating requirements for a function which takes an undirected network as an argument [87]. He first pointed out that the function should be invariant under graph isomorphisms. Secondly, it should be monotonic under graph modification: a vertex receiving a new incident edge (via operations called edge addition and edge switching) can only become more central. Sabidussi’s work was followed by Nieminen, who discussed similar axiomatisation for directed graphs [88]. More recently, Boldi and Vigna [89] proposed axioms to characterise the effect of size, density and addition of an edge in a directed network. They focused on axiomatic validity of centralities applied to graphs composed of k-cliques making their work specific to such type of graph. Bandyopadhyay et al. worked on universally applicable sets of axioms [90].

Most axiomatic approaches are limited to centrality applied for a specific network type and to date there seems to be only one key property that is an uncontested assertion of what centrality is, the star property, put forward by

Freeman [86; 91]: “a person located in the centre of a star is universally assumed to be structurally more central than any other person in any other position in any other network of similar size”. A generalisation of the star property was proposed by Schoch and Brandes [91] as an essential criterion of any centrality measure—NEIGHBOURHOOD INCLUSION PREORDER. It states that if the neighbourhood of a node u includes that of v , u is dominated by v . The relation

$$u \preceq_N v \text{ indicates that } \mathcal{N}(u) \subseteq \mathcal{N}(v) \cup \{v\} \quad (4.1)$$

and we say that u is DOMINATED by v .

Neighbourhood inclusion preorder can be used to define centrality as a function acting on nodes in \mathcal{G} in such a way that if the neighbourhood of a vertex v dominates that of a vertex u , then v is at least as central as u . For our purposes, we define CENTRALITY as a function $C: C(\mathcal{G}, u) \rightarrow \mathbb{R}$. The function must preserve the neighbourhood inclusion preorder: if $u \preceq_N v$, then $C(u) \leq C(v)$. We will discuss the neighbourhood inclusion preorder in detail in section 4.5.1.

We expect centrality to behave in the following way. The centrality score of a node increases if the distances between the node and the rest of the network decrease. Furthermore, if a graph is not strongly connected (i.e. there is at least one node which does not have a path to at least one other node), then the centrality of a node increases with the number of reachable nodes. This is often considered in centrality functions that do not increase monotonically with the number of terms in a relevant sum (which, as will be discussed in the following section, is a standard form a centrality function takes). For instance, closeness centrality is equal to an inverse of a sum of paths from a node to the rest of a network. If a node does not have a path to at least one other node, then the closeness centrality of such node is equal to zero—a result one may want to avoid. To go around this issue, usually only distances to reachable nodes are accounted for. We call this

type of centrality SHORTEST-PATH-BASED CENTRALITY.

4.3 Some Standard Measures of Centrality

Since a large variety of centrality measures exists, a thorough review is left out of this thesis. Here an emphasis is put on centralities that consider paths originating or terminating at a node to evaluate its importance. Indeed, many centrality measures are based on either the shortest paths in the network or spreading processes, such as diffusion or broadcasting. An extensive review of a variety of centrality measures can be found in [92]. Some traditional centrality measures are outlined in Table 4.1. Note that these measures assume undirected, unweighted network with one strongly connected component.

Often measures that are formulated for undirected networks can be used in directed networks without changes. Furthermore, some centrality measures, such as PageRank [94], demand directionality. What directionality of edges means is that $A \neq A^T$. Measures, given in Table 4.1 are then measures related to paths which originate at a node, and, since the networks we are interested in are DAGs, they point forward in time—to node's future. Let us call such measures FUTURE CENTRALITY and denote them by $C^{(\text{future})}$ ¹.

When centrality of a node is revealed through paths which end at that node, we can reverse the direction in the definitions, use d_{vu} instead of d_{uv} to quantify how central u is. In the space-time picture of DAGs, when d_{uv} is a path directed to u 's future, then d_{vu} starts in the past of u . Let us call such centrality PAST CENTRALITY and denote such functions $C^{(\text{past})}$. Instead of changing d_{uv} to d_{vu} in the formulation of centrality one can simply reverse the direction

¹Note that the definition of PageRank in Table 4.1 is related to node's successors, or nodes it points to. This is opposite to the original definition, used for search engine in Google. It is due to our opposite convention. Originally, in a website network an edge (u, v) indicates that a website u is referenced in a website v .

Measure	Mathematical Formulation for $C(u)$	Interpretation
Degree	$\sum_{v \in \mathcal{G}} A_{i_u j_v}$	The number of adjacent edges. Related to node's activity in the network.
Closeness	$\frac{N-1}{\sum_{v \in \mathcal{G}} d_{uv}}$	The inverse of the mean distance to other nodes. Which individuals can influence the entire network most quickly.
Harmonic closeness	$\frac{1}{N-1} \sum_{v \in \mathcal{G}} \frac{1}{d_{uv}}$	The harmonic mean of the inverse distances to other nodes.
Eigenvalue	$\lambda_1 C(u) = \sum_{v \in \mathcal{G}}^N A_{i_u j_v} C(v)$	"Node's centrality is its summed connections to others, weighted by their centralities" [93]. λ_1 here is the largest eigenvalue of \mathbf{A} .
Katz	$\sum_{l=1}^{\infty} \sum_{v \in \mathcal{G}} \alpha^l [A^l]_{i_u j_v}$	Node's centrality depends on all paths, but the shorter the paths to other nodes, the larger $C(u)$ ($0 < \alpha < \frac{1}{\lambda_1}$).
PageRank	$\frac{1-\alpha}{N} + \alpha \sum_{v \in \mathcal{N}^{\text{suc}}(u)} \frac{C(v)}{k^{(\text{in})}(v)}$	Important websites are pointed to more often by other important websites.
Betweenness	$\sum_{v,w \in \mathcal{G}} \frac{\sigma_{vw}^u}{\sigma_{vw}}$	The number of times a node appears on paths between other nodes. Recall that σ_{vw}^u is the number of shortest paths from v to w that pass through u . High betweenness nodes act as bridges.

Table 4.1: A few examples of centrality measures for an undirected, unweighted network with one connected component.

of edges and calculate centrality of a REVERSED GRAPH $\mathcal{G}_{\text{rev}} = (\mathcal{V}, \mathcal{E}_{\text{rev}})$, where $\mathcal{E}_{\text{rev}} = \{(v, u) | (u, v) \in \mathcal{E}\}$. Similarly, if centrality of a node depends on both, its future and its past sets, the scores of past centrality and future centrality may be joined into a global TOTAL CENTRALITY score. For example, one can define it as

$$C^{(\text{total})}(u) = C^{(\text{past})}(u) + C^{(\text{future})}(u).$$

It is worth highlighting that both closeness centrality and harmonic closeness centrality should be slightly modified to be useful in DAGs, which are always

weakly connected networks². Both functions are supposed to give a mean value of the length of paths originating or ending at a node, which holds true only when a network is strongly connected [95]. When it is weakly connected, we can normalise future closeness centrality with $|\mathcal{F}(u)|$ instead of $N - 1$ and future harmonic closeness centrality with $1/(|\mathcal{F}(u)|)$ instead of $1/(N - 1)$ (refer to Table 4.1 for the definitions of these measures). This is a convention we will use in this chapter.

Without the loss of generality, in this chapter we will also concentrate on FUTURE CENTRALITY unless otherwise stated. Any of our arguments can be translated to PAST CENTRALITY as we simply need to change from \mathcal{G} to \mathcal{G}_{rev} . Future centrality will be denoted $C^{(\text{future})}$; $C^{(\text{future})} : C^{(\text{future})}(\mathcal{G}, u) \rightarrow \mathbb{R}$. Similarly, past centrality is defined as $C^{(\text{past})} : C^{(\text{past})}(\mathcal{G}_{\text{rev}}, u) \rightarrow \mathbb{R}$.

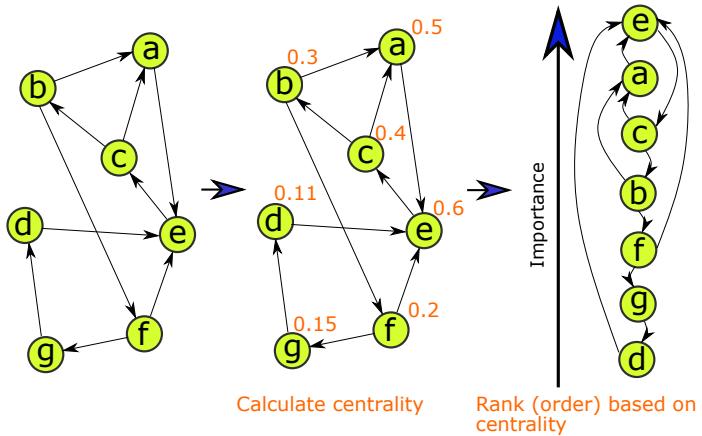


Figure 4.1: An illustration of how centrality measure can be used to obtain an order of nodes—a CENTRALITY ORDER. On the left, there is no notion of an order (rank) in the network. Centrality of nodes (in this example, eigenvalue centrality) provides scores for each node, as shown in the middle figure. These scores are used to order nodes from the most central, e in this case, to the least central, d in this case, see the figure on the right.

It would be cumbersome to analyse all existing centrality measures, so in the further analysis, we will limit ourselves to the seven measures outlined in

²In a (strongly) connected graph, every node has a path to every other node. A directed network is called weakly connected if replacing all its directed edges by undirected edges the result is a strongly connected undirected graph.

Table 4.1. As mentioned earlier, most centrality measures are derivatives of these “fundamental” centrality measures. The extent to which our results apply to other centrality measures is a task left for future studies.

It will be useful to consider centrality as a function that creates a PREORDER: nodes of a network are (partially) ordered given a centrality function. We will call this order a CENTRALITY ORDER. See Fig. 4.1 for an illustration of how eigenvalue centrality measure can impose an order of nodes in a directed network.

4.4 DAGs and Standard Measures of Centrality

Centrality provides many interesting insights about nodes in our DAGs. A “central” paper in terms of an out-degree in a citation network may be considered an important paper in a research field [85]. A “central” species in terms of degree in a food web is an actively participating species and changes to the abundances of such species directly affect many other species [96]. A “central” package in a network which represents software package dependencies is a software package which is a prerequisite to many other software packages.

Centrality in directed acyclic graphs is not a new topic. A large contribution to the study of centrality in DAGs comes from bibliometric studies of citation networks. In a citation network, node’s out-degree (the number of times a node is cited) is a well-known indicator of its impact [85]. What type of impact citation count indicates exactly is not entirely clear. For instance, Leydesdorff et al. argued that citation is a composite indicator: short-term citations can be considered as currency at the research front, whereas long-term citations can contribute to the codification of knowledge claims into concept symbols [97]. Recently, degree in citation networks was also used as a predictor of scientific awards winners [98].

Degree is not the only centrality measure used in DAGs. With several excep-

tions, one can use almost any centrality measure. One particular counter-example is eigenvalue centrality. Here centrality of a node is given by the corresponding eigenvalue of the adjacency matrix. As the adjacency matrix of any DAG is upper-triangular, it is also a nilpotent matrix with no non-zero eigenvalues, so all nodes are given centrality of zero.

Some centrality measures, such as PageRank, are used off-the-shelf to study citation networks [99; 100]. However, quite often some adaptions of centrality measures also take place. These changes account for the bias created by the arrow of time. It is well-known that old nodes are more likely to be indicated as central. Some measures that account for this effect include “time-rescaled” PageRank [101; 102], VoteRank [103].

TR (see section 1.3.1) and its effect on citation count was also studied [71]. REDUCED DEGREE is the degree of a node after transitive edges [31] are removed from the network. The so-called reduced citation count may be related to the topic diversity of a publication [71].

When centrality measures account for the “age bias” they are “treating symptoms” rather than “healing the disease”. Here the disease is that the topological order, an inevitable element of DAGs, creates the “older is more central” effect. In the following sections we will try to uncover the relation between the topological order on the centrality order. We will also address potential treatments which can help reveal “truly central” nodes.

4.4.1 Centrality Axioms for DAGs

To understand the mechanisms of centrality we will begin by formulating axioms for a function to be called a centrality measure. Based on work on general centrality axioms, here we list an adapted, DAG-specific axiom list.

1. *Isomorphic property.* If graphs \mathcal{G}, \mathcal{H} are ISOMORPHIC GRAPHS, centrality scores are equal. Isomorphic graphs are such that contain the same number of nodes and edges. Furthermore, there is a function f which maps each node v of the first graph to a node $f(v)$ in a second graph in such a way, that if $\exists(u, v) \in \mathcal{E}_{\mathcal{G}}$, $\exists(f(u), f(v)) \in \mathcal{E}_{\mathcal{H}}$. We say that this function is a structure-preserving bijection. Then centrality of a node $v \in \mathcal{G}$ should be equal to $f(v)$ in \mathcal{H} : $C(\mathcal{G}, v) = C(\mathcal{H}, f(v)) \forall v \in \mathcal{G}$.
2. *Locality.* Future centrality of a node v depends only on the connections to its future:

$$\begin{aligned} C^{(\text{future})}(\mathcal{G}, v) &= C((\mathcal{V}, \mathcal{E}), v) \\ &= C((\mathcal{F}(v) \cup \{v\}, \{(u, w) | u, w \in \mathcal{V} \text{ if } u, w \in \mathcal{F}(v) \cup \{v\}\}), v). \end{aligned} \quad (4.2)$$

An equivalent axiom for $C^{(\text{past})}$ is constructed by replacing \mathcal{G} with \mathcal{G}_{rev} . Similarly, total centrality of v must only depend on the union of node's future and past. This axiom necessitates that centrality of a node must depend on paths related to it but should not be perturbed by any change in the part of a network, that is not connected to the node.

3. *Isolated minima.* Centrality of a node v is zero if v is isolated, that is, $(u, w) \notin \mathcal{E}$ where $u = v$ or $w = v \forall u, w \in \mathcal{V}$. If future (past) of a node v is an empty set, its future (past) centrality, respectively, is minimal: $C^{(\text{future})}(\mathcal{G}, v) = \min(C(\mathcal{G}))$ if $\mathcal{F}(v) = \emptyset$ (similarly if $\mathcal{P}(v) = \emptyset$). Of course, we can be less strict and ask only for a minimal centrality, rather than its value to be zero.
4. *Edge monotonicity.* For a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $v, u \in V$, $(u, v) \notin \mathcal{E}$, adding an edge (u, v) increases $C^{(\text{past})}(v)$ and $C^{(\text{future})}(u)$. This axiom requires that the larger the number of connections a node has, the more central it is.

5. *Impact range.* A centrality measure is long-range if a local perturbation has a non-diminishing effect at large distances. Consider two graphs, $\mathcal{G} : (u, v \in V; v \succ u; (u, v) \notin \mathcal{E})$ and $\mathcal{G}' : (\mathcal{V}; \mathcal{E} \cup \{(u, v)\})$. Suppose further that $z, y \in \mathcal{F}(u); z \prec y$. Then centrality measure is considered long-range if

$$|C(\mathcal{G}', z) - C(\mathcal{G}, z)| < |C(\mathcal{G}', y) - C(\mathcal{G}, y)|. \quad (4.3)$$

If the reverse is true, centrality is short-range. Note that centrality measure can be either short-range or long-range, so this axiom should be looked at as a classifying axiom.

We can use DAG centrality axioms for a systematic analysis of centrality measures: if a centrality function does not admit at least one of the axioms, we would say that it cannot be a centrality function. For instance, these axioms would show that closeness centrality, as defined in Table 4.1, is not a measure of centrality in DAGs, whereas the modified version, which was described in section 4.4, would be a valid centrality measure.

4.5 Centrality and Topological Order

Let us now turn our attention to the relation between order, created using centrality, and the topological order. We will do so via the study of the ephemeral element of centrality—the neighbourhood inclusion preorder. We will discuss in detail what it is, and how it acts in directed networks. Lastly, we will show how the centrality order is dictated by the topological order of nodes. We will often discuss a node’s index (defined in section 1.3.1) as an indicator of its position in the topological order. Recall that the INDEX is defined as a node’s position in the topological order. We will use the convention that the index begins at the oldest node.

4.5.1 Directed Neighbourhood Inclusion Preorder

Neighbourhood inclusion, introduced in section 4.3, states that if we have a pair of nodes for which the neighbourhood inclusion preorder exists, the centrality function should order nodes such that the one with the larger neighbourhood has a larger centrality score, regardless of the choice of a centrality measure.

Schoch and Brandes discussed the neighbourhood inclusion in terms of an undirected, strongly connected network [91]. In DAGs we have three neighbourhoods to consider, namely, $\mathcal{N}^{\text{suc}}(u)$, $\mathcal{N}^{\text{pre}}(u)$, $\mathcal{N}(u)$ for each $u \in \mathcal{G}$. Intuitively, if a centrality score is based on node's future, we should consider $\mathcal{N}^{\text{suc}}(u)$, rather than its $\mathcal{N}^{\text{pre}}(u)$ set, as the latter has little to do with node's centrality score. In a DAG no paths that start at a node ever return to it, so centrality of u only depends on its successors, successors of those successors and so on (as is also required by the axiom 2 from section 4.4.1). If in an undirected network u is a neighbour of v as much as v is a neighbour of u , in directed networks, node's predecessors are not equivalent to node's successors and this asymmetry needs to be accounted for.

Let us consider all possible two-neighbour configurations and how the replacement of \mathcal{N} by either \mathcal{N}^{suc} or \mathcal{N}^{pre} in (4.1) relates to an expected centrality order. The node configurations are the following: $u||v$, $u \in \mathcal{N}^{\text{suc}}(v)$ (which also includes the case $u \in \mathcal{F}(v)$), $u \in \mathcal{N}^{\text{pre}}(v)$ (which also includes the case $u \in \mathcal{P}(v)$). Let us also assume shortest-path-based future centrality. These node configurations are given in Table 4.2.

In the case when $u||v$ (first and second rows in Table 4.2), neither u is a successor of v nor u is a predecessor of v , we can only have $\mathcal{N}^{\text{suc}}(u) \subseteq \mathcal{N}^{\text{suc}}(v)$ or $\mathcal{N}^{\text{suc}}(v) \subseteq \mathcal{N}^{\text{suc}}(u)$. A node with a dominant open neighbourhood is that of a larger centrality. Thus the neighbourhood inclusion preorder works as expected. Similarly, in the cases shown in rows three and four, centrality orders are also validated correctly by the inclusion preorder of successors.

Note there is a special case, given in the last row of the table, when the original neighbourhood inclusion preorder is inconclusive. The issue created by the directionality of edges is related to the size of neighbourhoods. In the last row of the table we have two valid neighbourhood inclusions: $\{w\} \subset \{u, w, v\}$ ($\mathcal{N}^{\text{suc}}(u) \subseteq \mathcal{N}^{\text{suc}}[v] \rightarrow C^{(\text{future})}(u) \leq C^{(\text{future})}(v)$, which allows $C^{(\text{future})}(v) > C^{(\text{future})}(u)$), but also $\{u, w\} = \{u, w\}$ ($\mathcal{N}^{\text{suc}}(v) = \mathcal{N}^{\text{suc}}[u] \rightarrow C^{(\text{future})}(v) = C^{(\text{future})}(u)$, which forbids $C^{(\text{future})}(v) > C^{(\text{future})}(u)$). Of course, we could simply choose $C^{(\text{future})}(v) = C^{(\text{future})}(u)$, or choose not to compare the neighbourhoods of two nodes. However in simple cases of out-degree centrality and Katz centrality $C^{(\text{future})}(u)$ is always smaller than $C^{(\text{future})}(v)$ in such two-node configuration.

To avoid the confusion, we may add a second requirement for the DIRECTED NEIGHBOURHOOD INCLUSION PREORDER, namely that a node on the right-hand-side of the (4.1) must be that of the larger open neighbourhood. Such choice removes the ambiguity in the case shown in the fifth row of Table 4.2, as we then consider the following neighbourhood inclusion condition: $\mathcal{N}^{\text{suc}}(u) \subseteq \mathcal{N}^{\text{suc}}[v]$ but not reverse. Here node u can be less central, as is the case for out-degree centrality, or as central as v , which is the case in harmonic closeness centrality.

Each centrality measure is different and the orders created by various centralities will differ. However what directed neighbourhood inclusion preorder requires is that nodes, for which a given neighbourhood preorder exists, are ordered with respect to the neighbourhood inclusion preorder in the centrality order.

4.5.2 Centrality without Neighbourhood Inclusion Preorder

We know that if neighbourhood inclusion preorder exists for two nodes, then centrality must respect the preorder for these nodes. However, in DAGs the notion of neighbourhoods of u, v are not always needed to compare their centrality. This information is already encoded in the topological order.

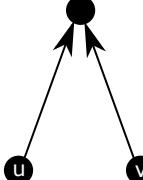
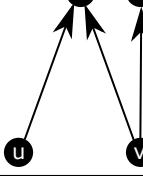
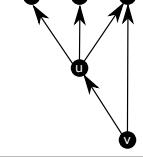
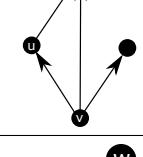
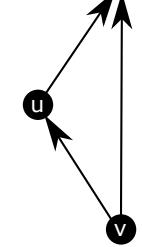
Node configuration	Centrality order	$(v, u) \in \mathcal{E}$	Neighbourhood inclusion	Neighbourhood size
	$C(u) = C(v)$	\times	$\mathcal{N}^{\text{suc}}(u) = \mathcal{N}^{\text{suc}}(v)$	$ \mathcal{N}^{\text{suc}}(u) = \mathcal{N}^{\text{suc}}(v) $
	$C(u) \leq C(v)$	\times	$\mathcal{N}^{\text{suc}}(u) \subset \mathcal{N}^{\text{suc}}[v]$	$ \mathcal{N}^{\text{suc}}(u) < \mathcal{N}^{\text{suc}}(v) $
	$C(u) \geq C(v)$	\checkmark	$\mathcal{N}^{\text{suc}}[u] \supset \mathcal{N}^{\text{suc}}(v)$	$ \mathcal{N}^{\text{suc}}(u) > \mathcal{N}^{\text{suc}}(v) $
	$C(u) \leq C(v)$	\checkmark	$\mathcal{N}^{\text{suc}}(u) \subset \mathcal{N}^{\text{suc}}[v]$	$ \mathcal{N}^{\text{suc}}(u) < \mathcal{N}^{\text{suc}}(v) $
	$C(u) \leq C(v)$	\checkmark	$\mathcal{N}^{\text{suc}}(u) \subseteq \mathcal{N}^{\text{suc}}[v]$, but also $\mathcal{N}^{\text{suc}}(v) = \mathcal{N}^{\text{suc}}[u]$, so both, $C(u) \leq C(v)$ and $C(u) = C(v)$ are allowed.	$ \mathcal{N}^{\text{suc}}(u) < \mathcal{N}^{\text{suc}}(v) $

Table 4.2: Examples of two-neighbour configurations, expected centrality order, and neighbourhood inclusion preorder. The case in the last row shows that the original neighbourhood inclusion preorder gives an unclear result about the centrality order of the two nodes u, v . If we consider the sizes of neighbourhoods in addition to the neighbourhood inclusion preorder, we are able to determine the centrality relation correctly.

First let us consider degree. It can be calculated using adjacency matrix, which is an upper-triangular matrix for a DAG. In terms of the adjacency matrix, we write the out-degree of a node as

$$k^{(\text{out})}(u) = \sum_{v \in \mathcal{G}} A_{i_u j_v} = \sum_{v \in \mathcal{N}^{(\text{suc})}} A_{i_u j_v} = \sum_{v > u} A_{i_u j_v}^{\Delta}.$$

The last equality uses \mathbf{A}^{Δ} which is equivalent to \mathbf{A} , ordered in such a way that $A_{i_u j_v}^{\Delta} = 0 \forall i_u < j_v$. So the closer i_u is to 0, the larger the range of possible indices j_v . So $k^{(\text{out})}(u)$ is likely to be larger if u is a node with a small index.

As another example, Katz centrality, defined as

$$C_K^{(\text{future})}(u) = \sum_{l=1}^{\infty} \sum_{v \in \mathcal{V}} \alpha^l [A^l]_{i_u j_v}, \quad (4.4)$$

takes into account all paths originating at a node. In terms of its neighbour $w \in \mathcal{N}^{(\text{suc})}(u)$ we have that

$$\begin{aligned} C_K^{(\text{future})}(u) &= \sum_{l=1}^{\infty} \left(\sum_{v \in \mathcal{F}[w]} + \sum_{v \in \mathcal{V} \setminus \{w\}} \right) \alpha^l [A^l]_{i_u j_v} \\ &\geq \left(\sum_{l=1}^{\infty} \sum_{v \in \mathcal{F}[w]} \alpha A_{i_u k_w} \alpha^{l-1} [A^{l-1}]_{k_w j_v} \right) + \sum_{l=1}^{\infty} \sum_{v \in \mathcal{V} \setminus \{w\}} \alpha^l [A^l]_{i_u j_v} \\ &\geq \alpha + \alpha C_K^{(\text{future})}(w) + \sum_{l=1}^{\infty} \sum_{v \in \mathcal{V} \setminus \{w\}} \alpha^l [A^l]_{i_u j_v}, \end{aligned} \quad (4.5)$$

resulting in $C_K^{(\text{future})}(u) > C_K^{(\text{future})}(w)$ if $\alpha > 0$ and $C_K^{(\text{future})}(w)$ is greater than $(\alpha + \sum_{l=1}^{\infty} \sum_{v \in \mathcal{V} \setminus \{w\}} \alpha^l [A^l]_{i_u j_v}) / (1 - \alpha)$. More so, if $\alpha > 1$, $C_K^{(\text{future})}(u)$ is always greater than $C_K^{(\text{future})}(w)$.

In harmonic closeness centrality of u in terms of its neighbour's centrality

gives the following inequality:

$$\begin{aligned}
C_{HC}^{(\text{future})}(u) &= \frac{1}{|\mathcal{F}(u)|} \sum_{v \in \mathcal{F}(u)} \frac{1}{d_{uv}} \\
&\geq \frac{1}{|\mathcal{F}(u)|} \left(\sum_{v \in \mathcal{G}/\mathcal{F}(w)} \frac{1}{d_{uv}} + \sum_{v \in \mathcal{F}(w)} \frac{1}{d_{wv} + 1} \right) \\
&\geq \frac{1}{|\mathcal{F}(u)|} \left(\sum_{v \in \mathcal{G}/\mathcal{F}(w)} \frac{1}{d_{uv}} - \sum_{v \in \mathcal{F}(w)} \frac{1}{d_{wv}^2 + d_{wv}} \right) + \frac{|\mathcal{F}(w)|}{|\mathcal{F}(u)|} C_{HC}^{(\text{future})}(w),
\end{aligned} \tag{4.6}$$

because the shortest paths from u to $v \in \mathcal{F}(w)$ may or may not pass w itself. $C_{HC}^{(\text{future})}(u)$ can be larger or smaller than $C_{HC}^{(\text{future})}(w)$ and the values of centrality are not entirely predetermined by the relative positions of the two nodes. In future harmonic closeness centrality, we need to evaluate the successors' inclusion preorder in order to find out which of the two nodes is more central.

In this section we showed that in some cases, such as for Katz centrality measure, the directed neighbourhood inclusion preorder need not exist to order two neighbouring nodes based on their centrality scores. It suggests the relation between the order of centrality and the topological order of a DAG. However, it is worth pointing out that similar arguments derived for other centrality measures from Table 4.1 do not show obvious relation between the topological order of nodes and the centrality order. In light of this, the following section will consider correlations of centrality and topological order and will reveal the relationship between the two orders numerically.

4.5.3 Centrality Correlations in DAGs

It is well-known that centrality measures are inherently correlated [104; 105; 106]. Amongst other analyses of centrality correlations, such work was also done in DAGs, in the context of food webs [107]. Typically, Pearson or Spearman correlation coefficients are used to evaluate similarity of centrality values, obtained using

various centrality measures on the same network. Here we will study correlations between centrality measures from Table 4.1 as observed in some networks from Table 1.1 and in random networks from section 1.3.3.

Fig. 4.2 shows that centrality measures divide into several similarity classes: past centrality measures, future centrality measures, and centralities, related to both futures and pasts of nodes. Past centralities, such as past harmonic closeness, in-degree, in-PageRank are correlated. Likewise, future centralities form a block of similar measures. Two measures based on both in- and out-going paths are degree and betweenness centrality. These measures also seem to be positively correlated. We see that node index is positively correlated with past centrality measures and negatively correlated with future centrality measures. These correlations are expected: as the arrow of time is in the same direction as the increasing index, older nodes have larger out-degrees. As the index of a node increases, its out-degree and future set decrease, while in-degree and past set increase.

This correlation analysis is a proof that topological order and centrality order are correlated in a wide range of DAGs. Future centrality of an old node is larger than that of a recent one, as the former has a larger future light-cone than the latter. However we also discovered that in some networks the correlations between indices are stronger than in others, indicating that the effect of the topological order on centrality order is not equally strong in all networks.

4.5.4 Mutual Information Between Centrality and Topological Order

If we expect the observed centrality score of a node to be strongly influenced by its position in the topological order, we may want to quantify this influence, as it may differ from node to node. In order to separate topological order and the centrality order, we can think in terms of comparing ranks. How much topological order

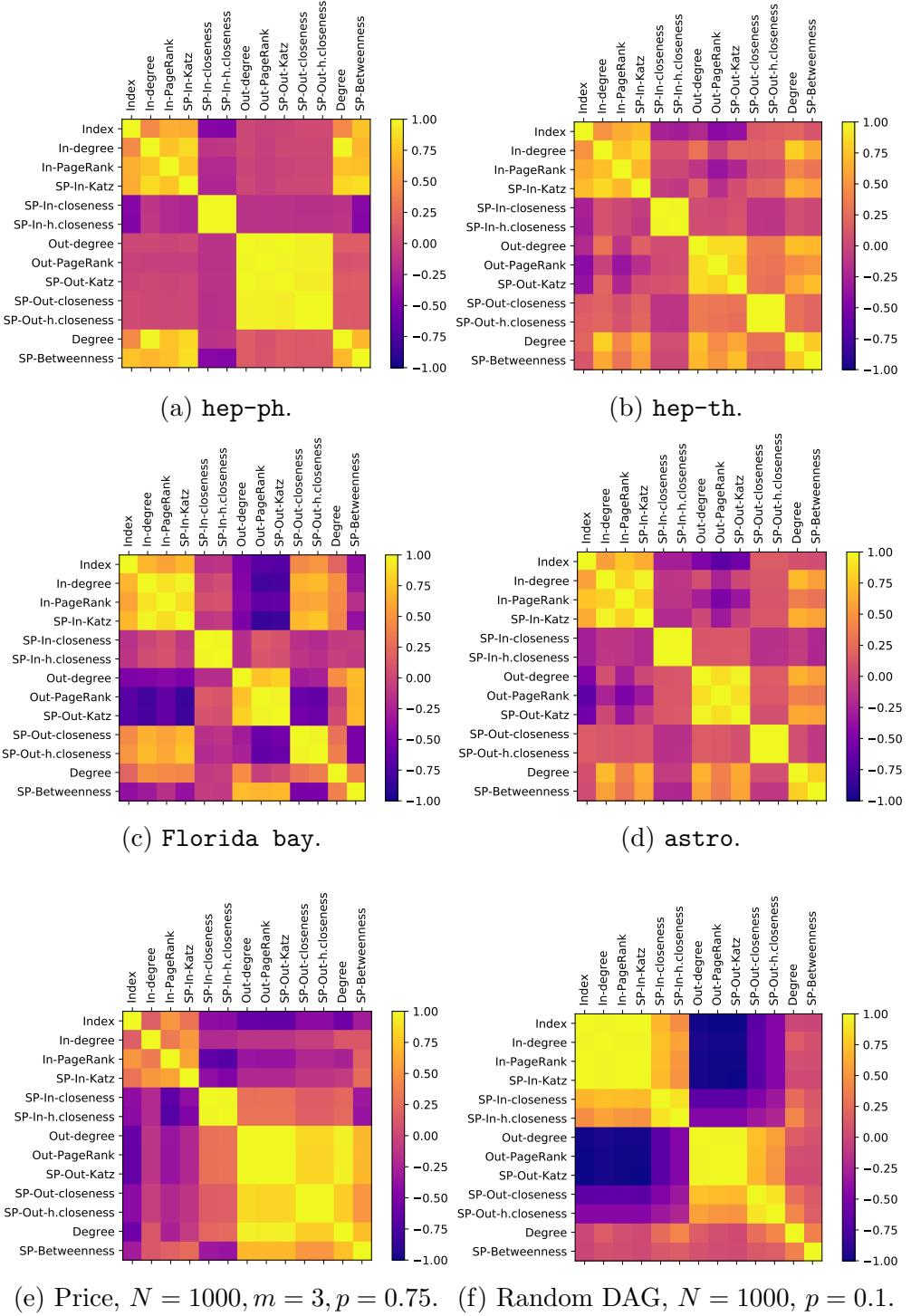


Figure 4.2: Correlations of classical centrality measures in various networks. The prefix “in” indicates that centrality is based on paths that terminate at each node; “out” indicates that paths, that originate at each node were used. These are $C^{(\text{past})}$ and $C^{(\text{future})}$ in our convention. In Fig. 4.2e and Fig. 4.2f mean values of correlation coefficient, obtained from 10 random networks are shown.

reordering is required to recover the centrality order? Statistical tests of rank comparison, such as Kendall's τ [108], would be appropriate, however these tests are limited to comparison of total orders, whereas we have partial orders.

Instead, we will consider mutual information between posets: if they are equal, the mutual information between them will be maximal and we can say that centrality order does not provide any new information about our nodes. We will take the approach proposed in [109] to compare our two partial orders: one, obtained from centrality and second, the intrinsic order of the DAG. To illustrate this idea we will employ a running example of out-degree centrality in the network, generated using the Price model with $N = 5000, m = 5, p = 0.625$.

First, we build a DAG with nodes of the original network and edges which represent the partial order of centrality. We will say that if $C(\mathcal{G}, u) > C(\mathcal{G}, v)$, then $\exists(u, v) \in \mathcal{E}_C$, where \mathcal{E}_C is a set of edges in the new DAG. Whenever we have a tie in centrality values, we do not connect the two nodes—they form an antichain (see 1.3.1 for definition). Note that in this centrality graph the directed neighbourhood inclusion preorder exists for all pairs $(u, \{v \in \mathcal{F}(u)\})$. Let us denote the network as \mathcal{G} and the centrality graph as \mathcal{G}_C . Having obtained \mathcal{G}_C we can compare it to \mathcal{G} for future centrality and \mathcal{G}_{rev} for past centrality³. Thus we compare the closed future set of a node in \mathcal{G}_C , $\mathcal{F}_C[u]$, to the closed future set of a node in the original \mathcal{G} or \mathcal{G}_{rev} , $\mathcal{F}_T[u]$ ⁴.

As in the original paper, we formulate the joint distribution between two random variables based on the sizes of the set intersections. More specifically, we use two indicator variables $i_C(u)$ and $i_T(u)$, associated with each candidate in \mathcal{G}_C

³In the original paper these sets are called DOWN SETS and a down set of a node contains all of its descendants.

⁴Although previously, we used the notation of $\mathcal{F}[u]$ to note the future set of u , but for the clarity of this section we will use the subscript “T” in order to accentuate the topological order from centrality order. Furthermore, note that we need to look at the future set $\mathcal{F}_T[u]$ defined on \mathcal{G} when considering future centrality and $\mathcal{F}_T[u]$ defined on \mathcal{G}_{rev} when the past centrality is considered.

and \mathcal{G} as follows. By choosing a candidate v from \mathcal{V} at random we require

$$i_C(u) = \begin{cases} 1 & \text{if } v \in \mathcal{F}_C[u], \\ 0 & \text{otherwise,} \end{cases} \quad (4.7)$$

for a node in centrality DAG \mathcal{G}_C and

$$i_T(u) = \begin{cases} 1 & \text{if } v \in \mathcal{F}_T[u], \\ 0 & \text{otherwise,} \end{cases} \quad (4.8)$$

in the original DAG \mathcal{G} . The joint probability distributions for the status of a randomly chosen node v given a fixed node u is

$$\begin{aligned} P(i_C(u) = 1, i_T(u) = 1) &= \frac{|\mathcal{F}_C[u] \cap \mathcal{F}_T[u]|}{|\mathcal{G}|}, \\ P(i_C(u) = 1, i_T(u) = 0) &= \frac{|\mathcal{F}_C[u]|}{|\mathcal{G}|} - \frac{|\mathcal{F}_C[u] \cap \mathcal{F}_T[u]|}{|\mathcal{G}|}, \\ P(i_C(u) = 0, i_T(u) = 1) &= \frac{|\mathcal{F}_T[u]|}{|\mathcal{G}|} - \frac{|\mathcal{F}_C[u] \cap \mathcal{F}_T[u]|}{|\mathcal{G}|}, \\ P(i_C(u) = 0, i_T(u) = 0) &= 1 - \frac{|\mathcal{F}_C[u]|}{|\mathcal{G}|} - \frac{|\mathcal{F}_T[u]|}{|\mathcal{G}|} + \frac{|\mathcal{F}_C[u] \cap \mathcal{F}_T[u]|}{|\mathcal{G}|}, \end{aligned} \quad (4.9)$$

and the marginals are

$$\begin{aligned} P(i_C(u) = 1) &= \frac{|\mathcal{F}_C[u]|}{|\mathcal{G}|}, & P(i_T(u) = 1) &= \frac{|\mathcal{F}_T[u]|}{|\mathcal{G}|} \\ P(i_C(u) = 0) &= 1 - \frac{|\mathcal{F}_C[u]|}{|\mathcal{G}|}, & P(i_T(u) = 0) &= 1 - \frac{|\mathcal{F}_T[u]|}{|\mathcal{G}|}. \end{aligned} \quad (4.10)$$

Given these probabilities we can construct various information-theoretic measures that will quantify the similarity of our two DAGs. Let us first define entropy of each of the two orderings. For instance, for a node u in a centrality graph, its

entropy is defined as

$$H_C(u) = - \sum_{i_C(u)=0}^1 P(i_C(u)) \log(P(i_C(u))) \quad (4.11)$$

and similarly in terms of $i_T(u)$ for a node u in the topological ordering. We will be measuring entropy in bits, so the logarithm has a base of 2. By definition, entropy is maximal when the amount of “disorder” is maximal. In our context, the largest disorder happens when a node dominates one half of all nodes, as then $P(i_C(u)) = 1/2$, $1 - P(i_C(u)) = 1/2$. It follows that the probability to randomly draw a node which u does not dominate is equal to the probability to draw a node which u dominates.

For our exemplary network, we see the largest entropy in both H_T and H_C when $|\mathcal{F}_T|/N = 1/2$ and $|\mathcal{F}_C|/N = 1/2$, see Fig. 4.3. The lack of data points at high values of $|\mathcal{F}_T|/N$ and low values of $|\mathcal{F}_C|/N$ reflects on the network model: there are few nodes that have a descendant set that constitutes large numbers of total nodes. For centrality order, we have many nodes with low degree (power law degree distribution), so most nodes do not have large future sets \mathcal{F}_C .

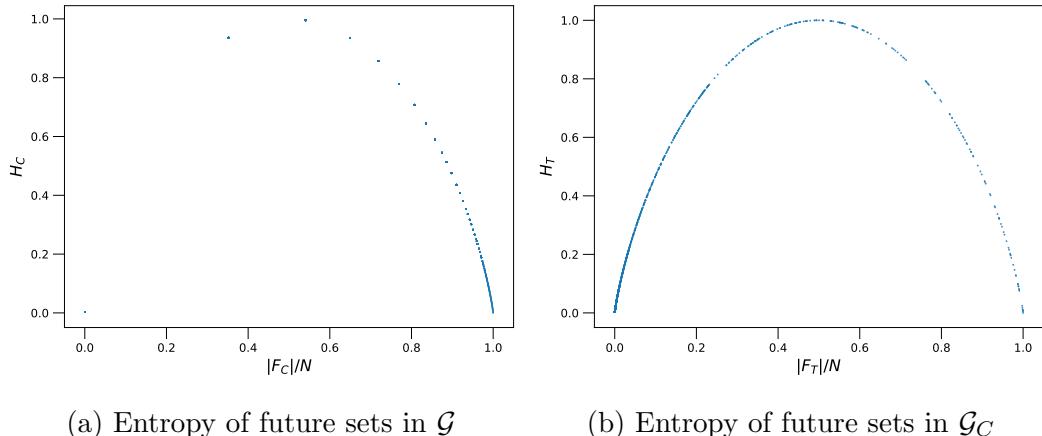


Figure 4.3: Entropy of future sets for nodes in the running example of the Price model. Fig. 4.3a shows entropy based on a future sets of nodes in \mathcal{G} , Fig. 4.3b shows entropy based on future sets of nodes in \mathcal{G}_C .

We can now define the conditional entropy $H_{C|T}(u)$ of a node u , which tells how much information about node's centrality can be explained by its index. Conditional entropy of a node u is defined as

$$H_{C|T}(u) = - \sum_{i_C(u)=0}^1 \sum_{i_T(u)=0}^1 P(i_C(u), i_T(u)) \log \frac{P(i_C(u), i_T(u))}{P(i_T(u))} \quad (4.12)$$

and quantifies the amount of information needed to describe the outcome of the random variable $i_C(u)$ given that the value of $i_T(u)$ is known.

So the fraction $(H_C(u) - H_{C|T}(u))/H_C(u)$ gives the amount of information on centrality that can be extracted from the topological order. When this fraction is large, we can expect the centrality order to be very similar to the topological order. When this fraction is small, topological order is not very useful for gauging information about centrality.

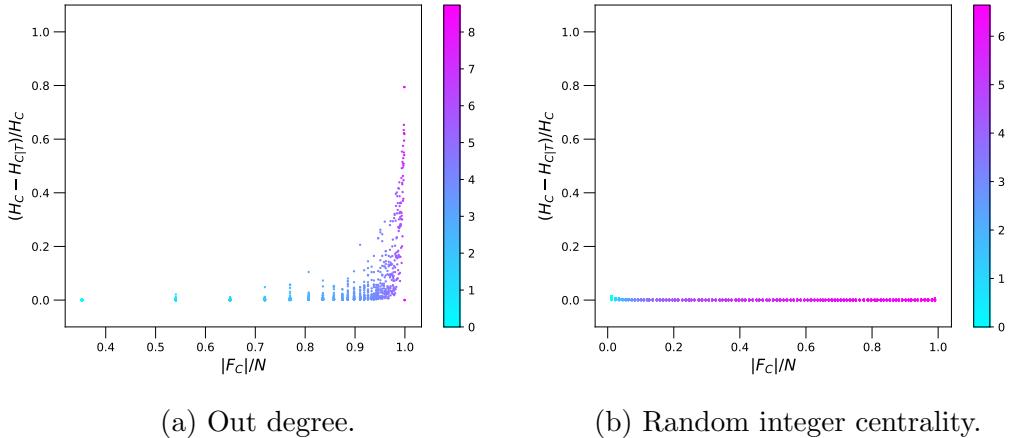


Figure 4.4: Mutual information between centrality order and the topological order for out-degree centrality in Fig. 4.4a and for centrality given at random in range from 0 to 100 for nodes in the running example of the Price model network. Mutual information between centrality order and topological order, obtained from out-degree centrality is large for nodes which have large degrees. These are also nodes that are old, because of the cumulative advantage nature of the network. Mutual information is close to negligible between random centrality order and topological order. Colour indicates the logarithm of out-degree of a node.

Fig. 4.4 shows the ratio $(H_C(u) - H_{C|T}(u))/H_C(u)$ as a function of the size of node's future set in \mathcal{G}_C , in which colour indicates paper's citation count. Here

all high degree nodes, as expected, are on the right sides of both figures, as they are dominating the majority of other nodes in the centrality order. However, the wide range of $(H_C(u) - H_{C|T}(u))/H_C(u)$ for those nodes is large, indicating, that in the Price model, old nodes (large future sets \mathcal{F}_T) are also important nodes and age can explain centrality. In comparison, if we allocate a random number as the centrality value of each node, the mutual information between this randomised centrality and topological orders is negligibly small.

Information-theoretic analysis could be useful to evaluate how much information about a node's apparent importance can be explained by its position in the topological order. The possible avenues with this approach are many and we did not exhaust all possibilities. For instance, we could use the ratio of $\frac{H_C(u) - H_{C|T}(u)}{H_C(u)}$ to evaluate the confidence level for the centrality value we have. Alternatively, we could weigh the obtained centralities with the ratio of the entropies H_C/H_T . Lastly, we could analyse the temporal patterns of the conditional entropy for a single node to understand its peak importance periods.

4.6 Other DAG Centralities

In the previous section of this chapter, we have studied the shortest-path-based centrality behaviour in DAGs. Let us now turn our attention to potential new candidates of well-behaved DAG-specific centrality measures. We have learnt that good centrality measures should order nodes in a way, respecting the neighbourhood inclusion preorder, and provide information about the importance of nodes, not necessarily present in the topological order.

4.6.1 Centrality Based on the Longest Paths

The longest paths may be a more natural choice for “geodesics”, as discussed in Chapter 2. This fact gives us motivation to test whether the longest-path-based centrality measures provide useful insights about centrality of nodes in DAGs. However, to make this unconventional choice, we need to convince ourselves the change of paths produces a measure which calculates centrality. More specifically, we want a measure which satisfies DAG centrality axioms (thus is centrality), adheres to the directed neighbourhood inclusion preorder, and provides information, not attainable from the topological order.

Let us first consider adaptions of the measures we studied previously. For instance, closeness centrality based on the longest paths is defined as

$$C_C^{(\text{future})\text{LP}}(u) = \frac{|\mathcal{F}(u)|}{\sum_{v \in \mathcal{G}} d_{uv}^{\text{LP}}}, \quad (4.13)$$

where d_{uv}^{LP} is the length of the longest path from u to v . We can similarly replace the shortest path with the longest path between nodes in a harmonic closeness centrality and betweenness centrality. In Katz centrality, $\alpha > 1$ indicates that longer paths are preferred over shorter paths. To adapt degree, we may want to consider transitive reduction (TR). Edges which remain in the network after TR are edges which participate in at least one longest path between some two nodes.

Let us have a look at the behaviour of these new centrality measures to see whether the longest-path-based centrality is related to topological order in the network. Let us also denote these centrality measures with a superscript “LP”.

For Katz centrality, as was shown in (4.5), when $\alpha > 1$, centrality of an older node is always larger than that of its successor. Thus C_K^{LP} will be strongly affected by the topological order of nodes and will likely be very similar to the shortest-path-based Katz centrality.

Harmonic longest-path-based closeness centrality of a node u is related to centrality of its future neighbour v in a similar way as was shown in (4.6), however now the inequality is reversed, as the longest path d_{uv}^{LP} where $v \in \mathcal{F}(w)$ is at least equal to $d_{wv}^{\text{LP}} + 1$ but can be longer (making $C_{HC}^{\text{LP}}(u)$ smaller). So centrality of u , again, can be lesser or greater than centrality of its successors w .

It is also not clear whether we prefer longer longest paths over shorter in centrality measures, such as closeness. If longer paths are preferred over shorter, the longest path centrality is likely to be strongly related to topological order of nodes. To see this, consider a simple future centrality, defined as

$$C^{(\text{future})\text{LP}}(u) = \frac{1}{|\mathcal{F}(u)|} \sum_{v \in \mathcal{F}(u)} d_{uv}^{\text{LP}}. \quad (4.14)$$

Intuitively, it calculates the average length of the longest paths, originating at u . We can show that in terms of $w \in \mathcal{N}^{\text{suc}}(u)$, $C^{(\text{future})\text{LP}}(u) \geq C^{(\text{future})\text{LP}}(w)$, because the longest path from u to $v \in \mathcal{F}(w)$ will be via w or will be even longer:

$$\begin{aligned} C^{(\text{future})\text{LP}}(u) &\geq \frac{1}{|\mathcal{F}(u)|} \sum_{v \in \mathcal{F}(u)/\mathcal{F}(w)} d_{uv}^{\text{LP}} + \frac{1}{|\mathcal{F}(u)|} \sum_{v \in \mathcal{F}(w)} (d_{wv}^{\text{LP}} + 1). \\ &\geq \frac{1}{|\mathcal{F}(u)|} \sum_{v \in \mathcal{F}(u)/\mathcal{F}(w)} d_{uv}^{\text{LP}} + \frac{|\mathcal{F}(v)|}{|\mathcal{F}(u)|} (C^{(\text{future})\text{LP}}(w) + 1). \end{aligned} \quad (4.15)$$

So there are some analytic indications that the longest-path-based centrality is influenced by the topological order, similarly as the shortest-path-based centrality is. Furthermore, correlation analysis in Fig. 4.5 shows a clear relation between the shortest-path-based centrality measures and their longest-path-based counterparts. The longest-path-based centrality measures are also correlated with node indices, but this relation is likely to vary, depending on the network topology. In order to make use of centrality without topological order, simply replacing one

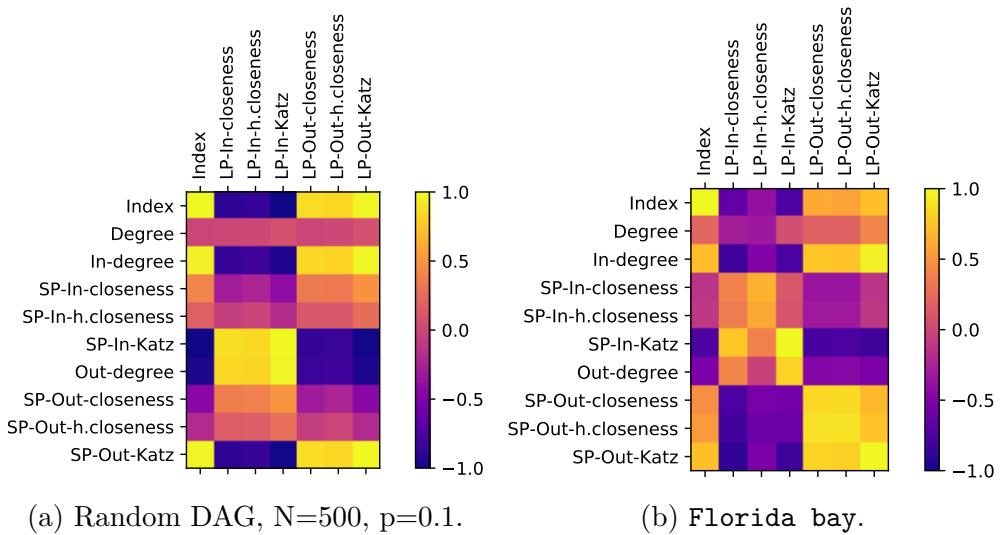


Figure 4.5: SpearmanR correlations between the shortest path (“SP”) and the longest path (“LP”)-based centrality measures in random networks, generated using random DAG model, $N = 500, p = 0.1$ in Fig. 4.5a and in Florida bay food web network in Fig. 4.5b. Figures show several separate groups of centrality measures. One consists of “in-” centrality measures, another consists of “out-” centrality measures. The corresponding shortest path centrality and the longest path centrality measures seem to be very similar.

type of path with another is not sufficient.

4.6.2 Centrality Using Future and Past Sets

Other features of nodes in DAGs are also worth considering for use in centrality. For instance, we can use the future (past) set and height (depth) of a node to define centrality in the following form:

$$C^{(\text{future})}(u) = |\mathcal{F}(u)| \cdot h(u), \quad C^{(\text{past})}(u) = |\mathcal{P}(u)| \cdot d(u). \quad (4.16)$$

Here $h(u)$ and $d(u)$ are, respectively, the height and the depth of a node u . Such centrality measures satisfy our directed neighbourhood inclusion preorder requirement: for two nodes at the same height, depth, centrality depends only on their light-cone volumes $\mathcal{F}(u)$ or $\mathcal{P}(u)$. For nodes $u, v | (u, v) \in \mathcal{E}$, $C^{(\text{future})}(v)$ is at least

$(h(u) + 1)(|\mathcal{F}(u)| - 1)$ but can be larger. Node u would be less central than v if all its neighbours are neighbours of v , but if some of them are not successors of v , u 's centrality can be larger.

Alternative DAG centralities could be based on a network after transitive reduction. Transitively reduced degree would satisfy directed neighbourhood inclusion preorder, as whenever $(u, v) \in \mathcal{E}$ in the graph after transitive reduction, we have $\mathcal{N}^{\text{suc}}(u) = \{v\}$. TR degree would also not be correlated with node's index.

4.7 Discussions

Centrality as any other important concept is polysemous. Although much effort has been put in finding the answer to a question of “*which* centrality is appropriate in this situation?”, not much work has been done for answering “*is* centrality appropriate in this situation?”.

The aim of this chapter was to challenge the concept of centrality in ordered networks. We hope to have convinced the reader that centrality and the order of directed acyclic graphs are intertwined. We have shown this by analysing centrality axioms, the neighbourhood inclusion preorder and correlations between centrality measures and the topological order.

Once the inter-connectedness between the two node orderings was revealed, we embarked on a search for a measure of centrality, not influenced by the topological ordering of nodes. We did it in a twofold way. First, in section 4.5.4 we suggested evaluating the mutual information between two partial orders, one based on the topological ordering of nodes, and other, based on centrality. If a node is more central than would be expected from its position in the topological order, the lack of mutual information between the orders would signify its importance. Secondly, in section 4.6 we looked at unconventional centrality measures

that could potentially replace the traditional centrality measures. We concluded that centrality based on longest paths would also be related to the topological ordering of nodes.

Chapter 5

Order-respecting Communities

This chapter is based on work presented in:

Making Communities Show Respect for Order,

V. V., T. S. Evans,

Applied Network Science 5, 15 (2020).

V. V. contributed to all parts of this work.

In Chapter 4 we considered one of many possible node orders—centrality—and how it works in directed acyclic graphs. In traditional network analysis, an edge or a path indicates relation between two elements, so when a node has many relations, we assume it is central. We discussed how centrality should be changed, if edges and paths show not only relation between two nodes but also their order. A natural follow-up question is whether this presence of an intrinsic order has implications on any other network analysis tools.

In this chapter, we turn our attention to community detection algorithms. There we assume that two nodes are similar, if many of their connections are alike. However, in DAGs certain connections between similar nodes cannot exist due to the inherent order in the system. For example, two papers can be produced

independently at the same time with similar new results yet by definition they can not cite each other. So the absence of connection between such nodes can be indicative of their hierarchical equivalence.

The development of the relativistic model for the Higgs mechanism is a good illustration of an event where very similar publications are not citing each other. The discovery was made by three independent groups: Brout and Englert (August 1964), Higgs (October 1964), and Guralnik, Hagen and Kibble (November 1964). This is an example of the so-called “multiple independent discovery”, in which “similar discoveries are made by scientists working independently of each other” [110]. Rather than a coincidence, multiple discovery is frequent in sciences [111]. A large list of multiple discoveries is available on Wikipedia [112].

Our goal is to capture similar, yet not connected nodes. To accomplish it, we employ a variation of community detection. The approach we take is to look for communities of similar nodes in a DAG which are antichains, denoted as \mathcal{A} . Recall that an antichain is a set of nodes in a graph, such that no nodes in an antichain are pairwise connected with paths of any length. See Fig. 5.1 for an illustration of an antichain. As there is no direct connection between nodes in an antichain, as a community, it is very different from that produced in general network community detection. To see how nodes in a DAG may be disconnected if they are in the same community, consider the following examples.

In the case of a computer package dependence DAG, two computer packages which fill a very similar role will not depend on each other but they will draw on similar “lower level” packages reflecting a hierarchy and an order in packages. For instance, the python network package `networkx` is a prerequisite for two python community detection algorithms, `python-louvain` and `demon`. In a food web, such as the `Florida-bay` food web we study in Section 5.3.4, pinfish, parrotfish and manatee are grouped together because they feed on similar species, such as

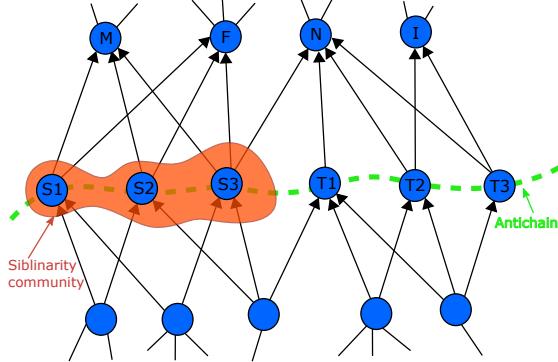


Figure 5.1: An antichain is a subset of nodes in the graph, such that none of the nodes are pairwise connected with edges or paths. In this illustration, an antichain is composed of nodes $\{S_1, S_2, S_3, T_1, T_2, T_3\}$ (the green dotted line is drawn as a guide of eye). Although in the same antichain, not all nodes share many neighbours. Nodes $\{S_1, S_2, S_3\}$ share many successors (they are all connected to nodes M, F), so we join them in a Siblinarity community. Figure reproduced from [113].

detritivorous polychaetes, sponges, bivalves. In a DAG a natural property of any nodes at the same level in the hierarchy is that they are *not* connected, directly by an edge or even indirectly via a longer path. In the context of a network with an order, with a hierarchy, it is very natural to cluster nodes which are not connected by any path, as then we can instantly assume they may be hierarchically equivalent.

For simplicity, we restrict ourselves to the case where each community is an antichain, and the set of all communities is a partition \mathfrak{A} , which we call an ANTICHAIN PARTITION. That is every node is in exactly one element (one community) of our antichain partition \mathfrak{A} . This is hard or non-fuzzy clustering in the language of data science.

The next problem is that there are many possible antichains and each node can be in many different antichains. Also nodes in useful communities will need to be similar in some sense. In a set of computer packages, we do not want to cluster a package on games with one on networks; we want to collect all the different network packages together in one community. Likewise, we do not want to cluster species

at the same level in food webs if they are from completely different environments. So we will aim to find the antichains which contain nodes that are similar by some appropriate measure. We take our inspiration from classic measures used to assess the similarity of two documents from their citation network alone (for example see [114] for a more recent application).

5.1 Related Network Partitioning Methods

Our method is related to two different types of network algorithms: community detection and DAG layering. Community detection does not include a notion of “layers”, where a layer would consist of nodes, which are hierarchically equivalent. As a consequence, a conventional community can be composed of nodes from many different layers. DAG layering, discussed in the subsequent sections, on the other hand, does not employ the notion of similarity, so a layer is composed of hierarchically equivalent nodes but a layer does not represent a community. Our algorithm bridges this gap and finds a layer decomposition of a DAG with a notion of similarity.

5.1.1 Community Detection via Modularity Optimisation

Community detection is an active research topic. This field is wide and many different approaches to finding groups of similar nodes were proposed. An extensive discussion of the main approaches to community detection in networks is given in [115]. Traditional community detection algorithms include graph partitioning methods (for a review of various graph partitioning methods see [116]) hierarchical clustering methods [116], optimisation methods [117; 118; 119] and stochastic block modelling (e.g. see a review in [115; 120]). Here we concentrate on a specific approach to community detection via modularity optimisation.

Recall that a network in which clustering coefficient varies significantly in different parts of the network is said to be “modular”. A quantity, which measures the extent to which the network is modular, is called **MODULARITY**. The idea behind modularity is that true community structure in a network corresponds to a, statistically speaking, significantly denser arrangement of edges. So in modularity the number of edges falling within groups is compared to the expected number in an equivalent network with edges placed at random [121]. The more the arrangement of edges deviates from the null model, the larger the modularity score.

To evaluate the modularity, one compares the observed edges between pairs of nodes to the expected value of this number. The observed number of edges between two nodes is given by the adjacency matrix entry $A_{i_u j_v}$, and the expected number is approximately $k_u k_v / E$, obtained from configuration model. The modularity of a network as a whole is counted by considering all pairs of nodes.

The Louvain algorithm is a type of community detection algorithm which aims to find a partition that maximises modularity in an efficient way. The quantitative expression of modularity

$$Q(\mathfrak{C}) = \frac{1}{2E} \sum_{c \in \mathfrak{C}} \sum_{u, v \in c} \left(A_{i_u j_v} - \frac{k_u k_v}{E} \right) \quad (5.1)$$

yields a high value if the partition \mathfrak{C} consists of high-quality modules. In order to maximise this value efficiently, the Louvain algorithm has two phases that are repeated iteratively.

First, each node in the network is assigned to its own community. Then for each node u , the change in modularity is calculated, for the change of it being moved from the community of itself to the community of its neighbour. If a significantly large change in modularity is observed, node u is moved to the new

community. This process is performed repeatedly and sequentially to all nodes until no significant change in modularity is observed.

During the second stage of the algorithm an INDUCED GRAPH is created. In it, the communities from the first stage are turned into nodes of the induced graph. Moreover, edges between nodes from different communities become edges between nodes in the induced graph, and edges between nodes in the same community are self-loops. After the creation of the induced graph the algorithm returns to the first stage. An example of a modular network and the communities, found by the Louvain algorithm, is shown in Fig. 5.2.

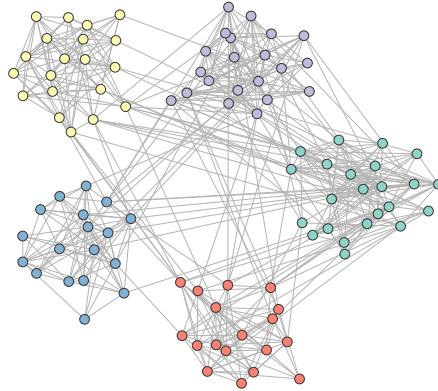


Figure 5.2: An example of a random modular network, composed of 100 nodes which have an average degree of 10.3. Degree distribution is Poissonian. Nodes in the same community are pictured of the same colour. The network was generated using the methodoly in [122], and communities were found using Louvain algorithm [119]). Modularity of this partition is 0.63. Figure taken from [123].

5.1.2 Layering of Hierarchical Graphs

Partitioning a DAG into maximal antichains is equivalent to the problem of graph layering. It is used for drawing hierarchical graphs, such as trees and DAGs. Sugiyama graph drawing algorithms use various layering techniques to remove edge overlap as much as possible [124]. The longest-path algorithm is used to find a layering with the smallest number of layers [125]. More sophisticated methods aim to limit the number of layers in the graph, variation in the size of each

layer, number of layers with crossing edges or maximum width of any of the layers [126; 127; 128; 129; 130; 131].

Probably the simplest partitions of a DAG into layers are obtained via the longest path algorithm. It uses the property of a DAG that we can always assign a HEIGHT to every node. Recall that the height $h(u)$ of a node u is the length of the longest path to u from any node with zero in-degree. It is straightforward to see that each node on a path must have different heights with the height increasing as you move along the path. Conversely, nodes of the same height cannot have any path between them so they form an antichain. Thus we can define the HEIGHT PARTITION $\mathfrak{A}^{(h)}$ to be the set of antichains $\{\mathcal{A}_a^{(h)}\}$, each of which contains all the nodes of a given height

$$\mathfrak{A}^{(h)} = \{\mathcal{A}_a^{(h)}\}, \quad \mathcal{A}_a^{(h)} = \{u | u \in \mathcal{V}, h(u) = a\}. \quad (5.2)$$

Similarly, we defined the DEPTH $d(u)$ of a node u to be the length of the longest path from u to a node with zero out-degree. Nodes with the same depth are guaranteed to form an antichain so we can define the DEPTH PARTITION $\mathfrak{A}^{(d)}$ to be a partition of the set of nodes by their depth,

$$\mathfrak{A}^{(d)} = \{\mathcal{A}_a^{(d)}\}, \quad \mathcal{A}_a^{(d)} = \{u | u \in \mathcal{V}, d(u) = a\}. \quad (5.3)$$

The height and depth antichains are examples of maximal antichains, each antichain is not a proper subset of any other antichain. Many of the layering algorithms are designed to produce the same or similar numbers of antichains, so again those are often maximal antichains or close to that. For this reason, the height and depth antichains are good representatives of the type of antichains produced by traditional layering algorithms, so they will be used to illustrate how our siblinarity antichain partitions are very different.

5.2 Siblinarity Antichain Partitions

Often the elements of either the height or depth antichain partitions (an element here is one community, one antichain in a partition) provide one definition of a level in the hierarchy. However, while these antichains respect the order of the DAG, we want to highlight much smaller groups which contain nodes which are much more similar than just the similarity imposed by the order as encoded by an antichain. All the nodes at one height, or those at one depth need not be very similar in general.

To add similarity to the hierarchy constraint encoded through our restriction to antichains, we can take inspiration from classic similarity measures used in bibliometrics. In that context, one way to assess the similarity of two publications is to look at overlap of their neighbours in the citation network. The more two publications share the same neighbours, the more similar they are said to be. The size of an intersection between two paper's bibliographies is called BIBLIOGRAPHIC COUPLING [132], whereas the size of an overlap between articles that they were referenced by is called CO-CITATION [133]. So by looking into similarities of neighbourhoods of two nodes, we can say something about how similar they are themselves.

A family tree provides a good example where people are the nodes and edges are from a parent to a child and so point forward in time in terms of birth date. There will be many people in a single generation but, by definition, none will be a parent or a child of any other person in the same generation so each generation forms an antichain. Generations are layers in a natural hierarchy for this DAG. However, almost all people in one generation will have little genetic biological relationship to each other so this large antichain, a single generation, may not be very interesting in many problems. However, if we also look for clusters of people within this generation, people who have common predecessors and so share one or

two parents, then these smaller communities may be of more interest. Using such a predecessor similarity measure on top of an antichain constraint would mean a community detection method would be producing communities of siblings.

These ideas of antichain and neighbour similarity are encoded in a function which measures the quality of a given partition \mathfrak{A} of our DAG into antichains, \mathcal{A} . Motivated by the family tree example, we call our function SIBLINARITY $S(\mathfrak{A})$ and we define it to be

$$S(\mathfrak{A}) = \sum_{\mathcal{A} \in \mathfrak{A}} \sum_{u \in \mathcal{A}} \sum_{v \in \mathcal{A} \setminus u} (\text{sim}(u, v) - \text{sim}_{\text{null}}(u, v)) . \quad (5.4)$$

Here the first term $\text{sim}(u, v)$ is some measure of the similarity of two nodes u and v . The second term, $\text{sim}_{\text{null}}(u, v)$, is the expected value of similarity of these two nodes in some suitable null model. There is a lot of freedom in choosing a null model but in general it is some randomised version of the DAG. As u and v are in the same antichain \mathcal{A} there is no path between nodes contributing to siblinarity. We have excluded the case $u = v$ so any node in a community by itself contributes zero and $S(\mathfrak{A}) = 0$ for the case where every community is a single node. Including the $u = v$ terms only adds an irrelevant overall constant, however, computationally it is more efficient to exclude this term, hence our choice.

Any similarity measure could be used but a logical choice for the similarity function in our context is the number of neighbours that u and v have in common, so $\text{sim}(u, v) = |\mathcal{N}(u) \cap \mathcal{N}(v)|$ where $\mathcal{N}(u)$ is the neighbourhood of node u . There are two obvious choices for this neighbourhood: one in terms of its predecessors $\mathcal{N}^{(\text{pre})}(u)$, as used in the family tree example above, and another in terms of the successors, $\mathcal{N}^{(\text{suc})}(u)$. It is useful to express this in a matrix form as follows

$$S(\mathfrak{A}) = \sum_{\mathcal{A} \in \mathfrak{A}} \sum_{u \in \mathcal{A}} \sum_{v \in \mathcal{A} \setminus u} \left(\tilde{A}_{i_u j_v} - \frac{\kappa_u \kappa_v}{W} \right), \quad \kappa_u = \sum_v \tilde{A}_{i_u j_v}, \quad W = \sum_{u,v} \tilde{A}_{i_u j_v} \quad (5.5)$$

The neighbourhood overlap is captured by the matrix $\tilde{\mathbf{A}}$ which is the product of the adjacency matrix \mathbf{A} of the DAG and its transpose. The $\tilde{\mathbf{A}}$ matrix can be regarded as the adjacency matrix for a derived graph $\tilde{\mathcal{G}}$ which is a directed weighted graph with the same node set as our DAG. In the case where we have an unweighted DAG we define this to be either $\tilde{\mathbf{A}}^{(\text{suc})}$, our successors-based similarity matrix, or $\tilde{\mathbf{A}}^{(\text{pre})}$ is a similarity matrix based on predecessors or $\tilde{\mathbf{A}}^{(\text{both})}$ is a similarity based on both, successors and predecessors. Here

$$\tilde{\mathbf{A}}^{(\text{suc})} = \mathbf{A} \cdot \mathbf{A}^T, \quad \tilde{\mathbf{A}}^{(\text{pre})} = \mathbf{A}^T \cdot \mathbf{A}, \quad , \tilde{\mathbf{A}}^{(\text{both})} = \tilde{\mathbf{A}}^{(\text{suc})} + \tilde{\mathbf{A}}^{(\text{pre})}. \quad (5.6)$$

The κ_u is the strength of edges attached to a node u in a graph with similarity matrix $\tilde{\mathbf{A}}$ and W is the total strength of edges in that graph. The second term in (5.5) also defines the null model, here a configuration model applied to $\tilde{\mathcal{G}}$.

The form we use (5.5) emulates the definition of modularity [134], a similarity measure minus expected value of that similarity measure in some null model. The biggest difference between modularity and siblinarity is that we impose the antichain constraint in the communities we study. This comparison with modularity suggests that we can modify siblinarity to adjust the typical number of antichains found, the resolution of our method. One simple method is to scale the null model term [135] so that

$$S(\mathfrak{A}, \lambda) = \sum_{\mathcal{A} \in \mathfrak{A}} \sum_{u \in \mathcal{A}} \sum_{v \in \mathcal{A} \setminus u} \left(\tilde{A}_{i_u j_v} - \lambda \frac{\kappa_u \kappa_v}{W} \right). \quad (5.7)$$

It is clear that for large λ we expect many small antichain communities. In particular, for $\lambda \gtrsim W$ adding any node in a community by itself to any other antichain will reduce siblinarity so we expect all the antichains to be the trivial case where each antichain has just one node. Conversely, we expect $\lambda = 0$ to produce large antichain communities.

5.2.1 Louvain Siblinarity Optimisation

Having defined a quality function, we can look for a partition of our nodes into antichains, \mathfrak{A} , which maximises the siblinarity $S(\mathfrak{A})$. This task faces the same challenges as most network community detection methods; there are many local minima and only approximate solutions can be found in a reasonable amount of computational time. We chose to adopt the Louvain approach [119] to community detection in networks. We chose this because it is fast and successful at finding communities in networks and because it proved easy to adapt to our context. Here, we will discuss how to adapt the Louvain algorithm [119] which is a widely used and successful methods to find communities in networks which maximise modularity. Emulating the Louvain algorithm, our siblinarity maximisation method is an iterative greedy algorithm in which each iteration has two phases.

In the first phase of our algorithm, we start with an initial partition into antichains in which each node is assigned to its own antichain. At each subsequent step, we try to move a single node n from its current antichain, \mathcal{A}_a , to another antichain \mathcal{A}_b , always choosing the configuration which maximises the siblinarity, even if that means leaving the antichains unchanged. In our implementation, we visit each node in a fixed sequence. Once the sequence is exhausted, we sweep through the same sequence once again. This process is continued until there are no more changes in the optimal antichain partition possible whatever node n we choose to examine. That is when changing the antichain partition by moving just one node can not increase the siblinarity. This marks the end of the first phase. In principle, we can also stop this first phase at any point as every new antichain partition is, by definition better than the last. So in our algorithm we also stop this phase if we have completed a given number of sweeps since the second phase is almost certain to simplify the problem and so speed up subsequent iterations.

For each node u we choose for a possible move, the change in siblinarity is

calculated for removing u from its current antichain, \mathcal{A}_a , and placing in a new antichain \mathcal{A}_b . It is important to enforce the constraint that the node u must not be connected to any existing node v in the potential new antichain \mathcal{A}_b , i.e. we want $\mathcal{A}_b \cup \{u\}$ to be an antichain.

In our algorithm, we further limit the choice of which new antichains \mathcal{A}_b we examine. If we are using siblinarity defined using a similarity measure using a neighbourhood set $\mathcal{N}(u)$ for our nodes u , then we look for antichains \mathcal{A}_b which contain at least one node $v \in \mathcal{A}_b$ which has a non-trivial similarity measure with our chosen node u , i.e. for us we require $|\mathcal{N}(u) \cap \mathcal{N}(v)| > 0$. These potential new antichains for node u are easy to find as this involves a two-step walk on the network starting from u . If we use a neighbourhood based on successors, that is $\mathcal{N}^{(\text{suc})}(u)$, then we only look at antichains \mathcal{A}_b which contain a predecessor v of a successor of u . We will call these **SUCCESSOR ANTICHAINS**. If we look only at predecessor neighbourhoods and so use $\mathcal{N}^{(\text{pre})}(u)$ for our neighbourhood sets, we shall refer to the resulting antichain partitions as **PREDECESSOR ANTICHAINS**. There is only one other case we examine, and that is we also check the case where we allow u to join a new antichain consisting of the node u alone, i.e. $\mathcal{A}_b = \emptyset$.

The change in siblinarity ΔS is given by

$$\begin{aligned} \Delta S(\mathcal{A}_a, \mathcal{A}_b \rightarrow \mathcal{A}_a \setminus u, \mathcal{A}_b \cup \{u\}) &= \sum_{v \in \mathcal{A}_b} (|\mathcal{N}(u) \cap \mathcal{N}(v)| - \mathbb{E}(|\mathcal{N}(u) \cap \mathcal{N}(v)|)) \\ &\quad - \sum_{q \in \mathcal{A}_a \setminus u} (|\mathcal{N}(u) \cap \mathcal{N}(q)| - \mathbb{E}(|\mathcal{N}(u) \cap \mathcal{N}(q)|)) , \end{aligned} \quad \text{provided } u \not\sim \mathcal{A}_b . \quad (5.8)$$

The first term is the contribution from the addition of node u to the antichain \mathcal{A}_b , while the second term is the effect of the removal of node u from its current antichain \mathcal{A}_a . Note the condition that u is not connected to any node in the existing antichain \mathcal{A}_b which we denote as $u \not\sim \mathcal{A}_b$. This is needed to ensure \mathcal{A}_b

remains an antichain when u is added.

In the matrix notation of (5.8), the change in siblinarity is given by

$$\Delta S(\mathcal{A}_a, \mathcal{A}_b \rightarrow \mathcal{A}_a \setminus u, \mathcal{A}_b \cup \{u\}) = \sum_{v \in \mathcal{A}_b} \left(\tilde{A}_{i_u j_v} - \frac{\kappa_u \kappa_v}{W} \right) - \sum_{q \in \mathcal{A}_a \setminus u} \left(\tilde{A}_{i_u j_q} - \frac{\kappa_u \kappa_q}{W} \right)$$

provided $u \not\sim \mathcal{A}_b$.

(5.9)

In the second phase we create an INDUCED GRAPH $\mathcal{H} = \{\mathcal{V}_H, \mathcal{E}_H\}$ from the original graph \mathcal{G} and the antichain partition \mathfrak{A} left at the end of phase one. Each node $a \in \mathcal{V}_H$ in this induced graph \mathcal{H} represents a single antichain, $\mathcal{A}_a \in \mathfrak{A}$, as given at the end of the previous phase. The edges between nodes of induced graph are given a weight equal to the sum of the weights of all the edges between the equivalent antichain nodes in the original graph \mathcal{G} of the induced graph. For instance, if there were k_{ba} edges all of weight 1 pointing from nodes in the antichain \mathcal{A}_a to the antichain \mathcal{A}_b at the end of phase one, there would be a directed edge $(a, b) \in \mathcal{E}_H$ in the induced graph with weight equal to k_{ab} in the induced graph¹. In terms of matrices, if H_{ab} is equal to the weight of the edge from node a to b in the adjacency matrix for the induced graph, then we have that

$$H_{ab} = \sum_{v \in \mathcal{A}_a} \sum_{u \in \mathcal{A}_b} A_{j_v i_u}. \quad (5.10)$$

Once the induced graph is created, the algorithm continues by finding an antichain partition of the induced graph using siblinarity, starting with the phase one. We continue until there is no substantial increase in the siblinarity function (5.7).

¹The induced graph does not have to be a DAG: antichains are possible in graphs with cycles: by definition, an antichain is a subset of nodes such that there is no path between any of two of them in this subset. This is perfectly valid in any graph, however, in some they are more interesting than in others.

5.3 Siblinarity Communities in Some Networks

5.3.1 Space-Time Lattice

To illustrate the key differences between different communities in DAGs, we employ the space-time lattice model (refer to Appendix A for definition) on a small scale. In Fig. 5.3 we show the same instance of the space-time lattice model in which nodes are placed on an eight-by-eight square lattice. We then use colour to show the different partitions produced using different algorithms.

The bottom two examples of Fig. 5.3 illustrate standard community detection algorithms applied to our network. Here these are found by optimising modularity with two different resolutions (obtained by scaling the null model term). The communities found tend to be located in one region of space reflecting the spatial constraint in the model. However, they extend over several values of time so these communities do not respect the order inherent in the DAG. That is, the nodes in these Louvain communities are not similar in terms of their time coordinates.

If this was a citation network, we would be grouping papers together with different publication dates. If we were interested in comparing the impact of each paper through citation counts, the older papers would have an advantage making the comparison unfair. If this toy model represented a food web, these communities derived from the undirected network might capture aspects such as distinct parts of the ecosystem. However it fails to find the hierarchy, each group would contain predators and their prey.

So these communities from the undirected graph may be of use in some contexts, but they are not particularly sensitive to the inherent structure of the DAG coming from the time direction.

The middle two examples in Fig. 5.3 use the height- and depth-antichain partitions. Now the hierarchical structure coming from the time direction is clearly

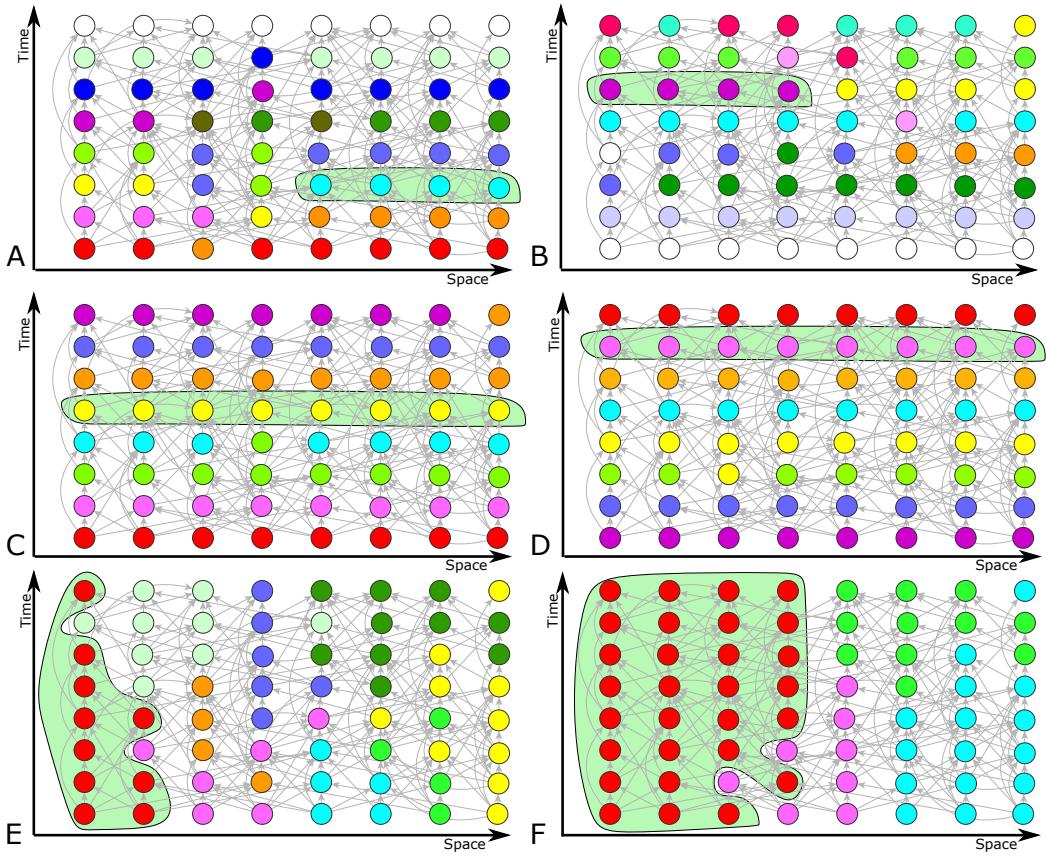


Figure 5.3: Examples of various partitions of DAG from a simple space-time model. In this DAG, edges are more probable between two nodes if they are a shorter Manhattan distance apart (in the space-time). The time coordinate, which induces the acyclicity in the network, is vertical. Colour indicates community of a node. The top row shows siblinarity communities based on common successors (A), and predecessors (B). The central row shows layers based on height (C) and depth (D). The communities in the bottom row are from Louvain community detection using resolution parameter value of one (E) and two (F). Nodes coloured white are in a community of size one. Siblinarity partitioning tends not to form communities (top row) which stretch across the whole network unlike those communities based on height or depth shown in the centre row. The communities found using traditional methods respect the spatial (horizontal) constraint but show no respect for the order in the DAG as they are spread over several times vertically. Figure reproduced from [113].

exposed, with many communities running across one row, perhaps two. Where two rows are involved, it is showing how these communities detection methods are highlighting where the placement of the nodes in the visualisation does not reflect the topological reality because of the stochastic aspect of edge placement.

This shows that these antichain partitions can do a useful job, picking out the true topological based hierarchy as defined by the data.

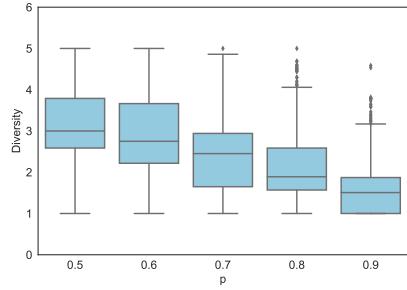
However the height- and depth-antichain communities show the opposite problem from the undirected graph communities. That is they respect the order coming from the arrow of time but they fail to pick up in any way the spatial clustering in the data. If this was a citation network, these would cluster papers published at a similar time regardless of academic field. For a food web, all predators at the same level would be grouped together regardless of whether or not they were competing in the same ecological niche.

The top two networks in Fig. 5.3 show the siblinarity antichain communities, one based on predecessor neighbours and the other using successors. This shows that the communities respect both the time and the spatial clustering in the data. The nodes tend to be in the same row and only contain nodes which are close by reflecting the spatial clustering in the model. This suggests that when applied to a citation network, this method will cluster papers which are directly comparable, similar publication date and similar field. Applied to food webs, siblinarity will only group predators at the same level competing in the same niche.

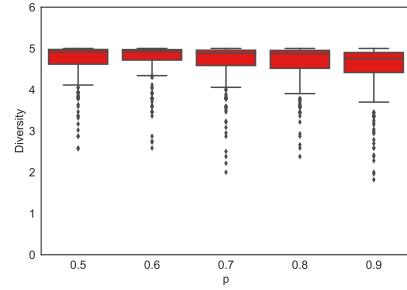
5.3.2 Diversity of Antichains: The Price Model with Fields

The space-time lattice model illustrated that the communities formed by our methods respect the order implicit in a DAG. Our method also aims to create groups which are similar in other ways as reflected in the network structure. To test this we use our modified version of the Price model, described in detail in Appendix B. In a nutshell, this is a modified model of Price, where we inject the notion of node “similarity” (when nodes are associated with the same group, cluster) and “dissimilarity” (when a pair of nodes is associated with two separate groups), and a node prefers to share an edge with other nodes it is similar to.

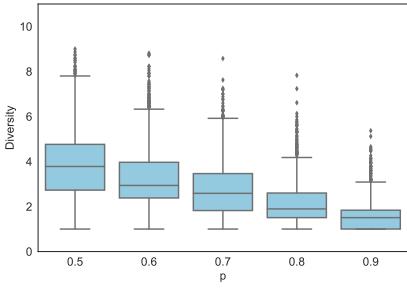
We will use the language of a citation model, reflecting Price's original context. So here we say we are aiming to group papers (nodes) of a similar age (in an antichain) published in the same academic field (similarity group) based on the network structure alone (using co-citation or bibliometric coupling).



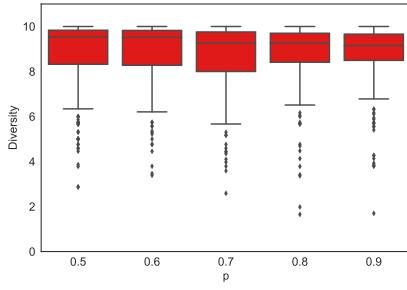
(a) Diversity of successor siblinarity antichains in a five-field Price model.



(b) Diversity of height antichains in a five-field Price model.



(c) Diversity of successor siblinarity antichains in a ten-field Price model.



(d) Diversity of height antichains in a ten-field Price model.

Figure 5.4: Comparison of diversity in height-based antichain partitions (red boxes on right) versus siblinarity communities, based on common successors (blue boxes on left) in the Price model with five fields (top row) and ten fields (bottom row) and different intraconnectivity probability p . We studied ten networks of 5,000 nodes for each set of parameters. Figures show that the diversity of siblinarity antichains is clearly significantly smaller than that of heights. The box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to show the range of the data. Flier points are those past the end of the whiskers. Figure reproduced from [113].

Here we consider the case when a field is assigned stochastically using a uniform distribution so that it is equally likely to assign a field $f = 1$ and $f = 5$. Other variations are possible, for instance, one could assign the fields sequentially deterministically or create non-uniform in size fields.

Our aim is to show that our antichain partitions are largely composed of papers from the same field which we measure using Shannon's diversity metric D [136]. That is given an antichain \mathcal{A} we find p_f , the fraction of nodes in the antichain which are in field f . The diversity of this antichain is then given by

$$D(\mathcal{A}) = \exp\left(-\sum_f p_f \ln(p_f)\right) \quad (5.11)$$

so that $1 \leq D(\mathcal{A}) \leq |\mathcal{F}|$. The results for the average diversity of a partition, $\sum_{\mathcal{A} \in \mathfrak{A}} D(\mathcal{A})/|\mathfrak{A}|$ are shown in Fig. 5.4, results for the distribution of $D(\mathcal{A})$ values within \mathfrak{A} are shown in Fig. 5.5. Not surprisingly the diversity of height and depth antichains is almost always larger than that of communities found from a siblinarity antichain partition. This is expected because the height and depth antichains typically contain more nodes, reflected by the low number of points on their plots, as their construction is not effected by the field labels. Since they are constructed without regard to the field labels, we would expect, and find, that the diversity of the height and depth partitions is close to the maximum value where $p_f \approx 1/|\mathcal{F}|$ and so $D \approx |\mathcal{F}|$.

We also noted that the diversity of antichains that are height or depth based does not approach maximum if the sizes of fields are not uniform. This is expected as Shannon's entropy not only accounts for number of species in the ecosystem but also their relative abundances. Uneven sizes of fields result in expected diversity score lower than maximum, as the maximum would be obtained if we had equally abundant species, as seen in the other two figures.

5.3.3 Citation Networks

We also performed analogous analysis of a real citation network. We considered the `cora` dataset, as here each node is labelled by one of seven different topics. We

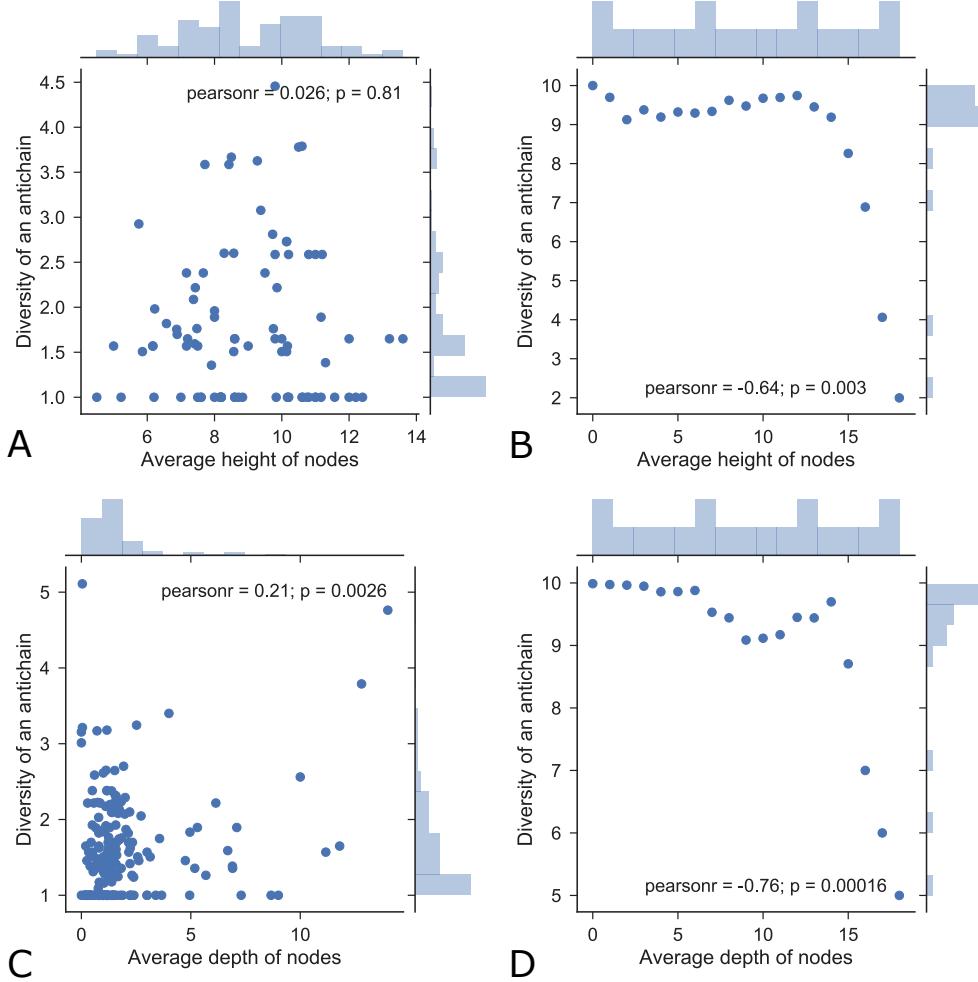


Figure 5.5: Shannon's diversity of antichains in a Price model with planned partition of nodes into five fields for networks of 5,000 nodes. Each node is assigned to a field sequentially and nine times out of ten a new node chooses to connect to an existing node of the same field, with a preference for large degree nodes, $\phi = 0.9$. Each new node attaches to $m = 3$ older nodes. Each point represents an antichain in a given partition, with the average h or d for nodes in the antichain and Diversity $D(\mathcal{A})$ (5.11) used as coordinates. Four different types of antichain partition are shown: A - successors-based siblinarity antichains, B - height-based antichains, C - predecessors-based siblinarity antichains, D - depth-based antichains. The height and depth antichains are clearly much bigger (few points on their plots) but much less diverse, with their diversity close to the theoretical maximum 10.0. On the other hand, the siblinarity antichains based on co-citation similarity (A) or bibliometric coupling (B) are drawn largely from the same field with diversities close to the minimum value of 1.0. Figure reproduced from [113].

use this label to quantify the diversity of the nodes in the antichain communities we find, similar to the analysis in Section 5.3.2. Fig. 5.6 shows that the diversity

of 134 siblinarity antichains is much smaller than the partitions based on heights. Furthermore, most of the time the diversity values for siblinarity antichains are equal to one, which indicates that all nodes in the antichain have the same label. This result confirms that the siblinarity communities are communities of similar papers, not just ones published around the same time.

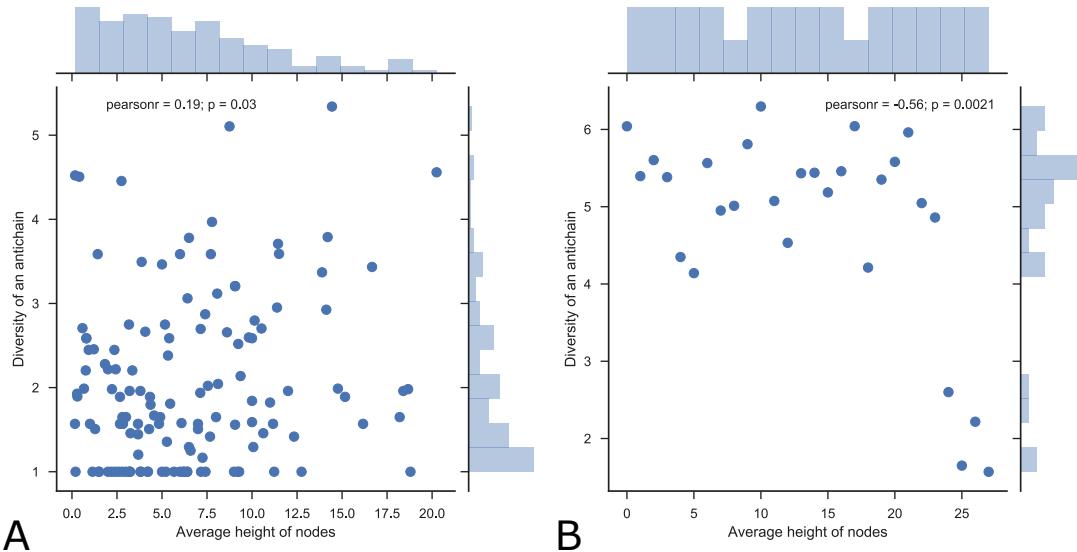


Figure 5.6: Shannon’s diversity of nodes in antichains of the `cora` citation network. Each node has a label assigned to it, based on the topic the paper is on. There are seven topic in total. Two different antichain partitions are shown: A — successors and predecessors-based siblinarity antichains, B — height-based antichains (diversity calculated only for antichains, that are composed of at least five nodes). The horizontal axis is the average of the heights of nodes in the antichain. The height antichains, with diversity values often close to the maximum value of seven, are much more diverse than the siblinarity comminties. Figure reproduced from [113].

Looking more broadly, the siblinarity communities can be an interesting and useful tool to study citation networks. In particular, in bibliometric studies context our siblinarity community detection is similar to “related records” search in Science Citation Index (SCI), introduced by Garfield in 1964. A related record is any record which shares at least one cited reference with the original source record. The more shared references, the more closely related the records are—an extension of the notion of citation searching to track a subject area. The related

records are further ranked based on the number of shared references. As pointed out by Newman, simply using bibliographic coupling or co-citations is flawed: strong bibliographic coupling or co-citations only occur between papers that have either large bibliographies or are highly cited, respectively [4]. Our approach naturally eliminates this flaw: by comparing the observed neighbourhood overlap to a null model, we can evaluate the significance of the similarity regardless of the degrees of two nodes.

5.3.4 Florida Bay Food Web

We also studied our antichain partitions in the DAG version of the Florida Bay food web data set [44]. Results are shown in Table 5.1.

We see that many of our siblinarity antichains consist of similar species as indicated by their similar names. For instance community 9 consists of, amongst other crustaceans, several types of amphipods. Another example of how our siblinarity antichains work is the example of the green turtle in community 25. The “Green Turtle” node has been placed in a community along with much smaller species, seemingly very different from turtles, such as shrimp and isopods. However, green turtles feed on thalassia (a type of seagrass, commonly known as turtlegrass), which is also a food source for organisms, represented with “DOC” (Dissolved Organic Carbon) node and isopods. Furthermore, all of the species in this green turtle antichain community feed on epiphytes. Nodes in this antichain all have the same height of two, but their depths range from 22 for “Epiphytic Gastropods” to 30 for “Isopods”, “Herbivorous Shrimp”, and “Thor Floridanus”. Another example of an antichain, in which nodes of a variety of heights and depths were collected into a siblinarity community is the community 27, consisting of “Other Snapper” (height 27, depth 4), “Other Pelagic Fishes” (height 28, depth 4) and “Spotted Seatrout” (height 29, depth 3).

No.	Species (type)
1	2 μ m Spherical Phytoplankt, Synedococcus, Oscillatoria, Small Diatoms (< 20 μ m), Big Diatoms(> 20 μ m), Dinoflagellates, Other Phytoplankton, Benthic Phytoplankton, Thalassia, Halodule, Syringodium, Roots, Drift Algae, Epiphytes.
3	Acartia Tonsa, Oithona nana, Paracalanus, Other Copepoda, Other Zooplankton, Sponges, Bivalves.
7	Coral, Other Cnidaridae.
9	Benthic Crustaceans, Detritivorous Amphipods, Herbivorous Amphipods, Detritivorous Gastropods, Detritivorous Polychaetes, Suspension Feeding Polych, Macroben-thos, Detritivorous Crabs.
18	Toadfish, Brotalus.
19	Other Killifish, Goldspotted killifish, Blennies, Clown Goby, Silverside, Lobster, Preda-tory Crabs, Callinectus sapidus, Bay Anchovy, Rainwater killifish, Mullet, Other Horsefish, Gulf Pipefish, Dwarf Seahorse, Code Goby, Halfbeaks.
22	Flatfish, Grunt, Pinfish, Rays, Porgy, Scianids, Parrotfish, Bonefish, Needlefish, Snook, Puffer, Manatee.
23	Omnivorous Crabs, Pink Shrimp.
25	DOC, Isopods, Herbivorous Shrimp, Thor Floridanus, Sailfin Molly, Green Turtle.
26	Sharks, Tarpon, Lizardfish, Grouper, Jacks, Pompano, Gray Snapper, Red Drum, Mackerel, Small Herons & Egrets, Ibis, Roseate Spoonbill, Herbivorous Ducks, Om-nivorous Ducks, Gruiformes, Small Shorebirds, Gulls & Terns, Kingfisher, Loggerhead Turtle, Hawksbill Turtle.
27	Other Snapper, Other Pelagic Fishes, Spotted Seatrout.
28	Stone Crab, Sardines, Anchovy, Other Demersal Fishes, Filefishes.
29	Barracuda, Loon, Greeb, Pelican, Comorant, Big Herons & Egrets, Predatory Ducks.
33	Free Bacteria, Dolphin.
35	Raptors, Crocodiles.
36	Output, Respiration.
Ind.	Input (0), Water Flagellates (2), Water Ciliates (4), Benthic Ciliates (5), Meroplankton (6), Meiofauna (8), Benthic Flagellates (10), Water POC (12), Predatory Gastropods (13), Echinoderma (14), Predatory Shrimp (15), Predatory Polychaetes (16), Mojarra (20), Benthic POC (24), Spadefish (31), Catfish (32), Eels (34).

Table 5.1: Siblinarity communities, based on common prey and common predators in the Florida Bay food web. Some communities consist of a single species and these are listed together under community number “Ind.” to reduce the size of the table. The numbers in brackets indicate the community index in our data. Note the green turtle in community 25 appears to be out of place with the other smaller species in the same community but they all tend to feed on similar species. Table reproduced from [113].

5.4 Conclusions

In this chapter we have highlighted that in an ordered network, the absence of a pairwise relationship of nodes can be just as useful a signal as the presence of it. In DAGs the implicit order prohibits the direct connections of many similar nodes. As this order is intrinsic to the very nature of a DAG, our response has

been to embrace it. To find meaningful communities, we searched for antichains, sets of disconnected nodes which often play a useful role in the study of DAGs and which reflect the topological properties on the networks we consider. To ensure our antichains contain similar nodes, we have used neighbourhood overlap to define the siblinarity of a partition of a DAG into antichain communities.

We have shown using many examples how the antichains that maximise siblinarity are interesting and relevant. Our communities are extremely different from usual communities found in network analysis as the simple examples in Fig. 5.3 show. The communities in traditional network analysis consist of strongly interconnected nodes while siblinarity communities consist of nodes that are strictly not connected, but which share similar neighbours.

This novel type of node partitioning has several potential uses whenever data is well represented by a DAG. In a citation network, each antichain community would be composed of publications that are similar thematically and which were published at a similar time. So this approach could be used as a recommendation system for publication search: given an input paper, one could find alternative similar publications by looking into the antichain of the input paper. Another common practice in bibliometrics is partitioning a citation network into subsets of nodes published in the same year in order to make comparisons between those publications. This is to avoid the bias that the arrow-of-time introduces, for instance older papers have more citations on average than younger papers. However, such clustering by publication date is very similar to clustering by height and this means papers from completely different fields are in the same community. Our method produces clusters of similar age and topic.

Our siblinarity algorithm, like any clustering algorithm, can also be used to coarse-grain a DAG which can help navigate a large amount of data. In our case we would contend the order of a DAG is important. Having a predator and its

prey in the same community may not be a useful classification when examining food webs. Indeed we exploit this coarse graining in our implementation of the Louvain algorithm to find antichain partitions which have approximate maximal values of siblinarity.

Looking ahead, a good future direction would be in incorporating “fuzziness” into our approach. For a community to be a strict antichain is a very strong requirement, especially when a network is sparse. So allowing some nodes into a siblinarity community, even if that would break the antichain requirement, would enable forming larger communities.

It is worth pointing out that although we developed our methodology in terms of DAGs, this type of partitioning can be applied to other types of networks, such as multilayer networks, bipartite networks, directed networks and even undirected networks (although in the case of the last type, antichains would most likely be composed of individual nodes). Studying siblinarity communities in non-DAG networks is another natural future perspective.

Chapter 6

Discussions

In this thesis, we studied the meaning of paths in directed acyclic graphs. Our aim was to understand what implications an intrinsic order of nodes and edges has on both, paths, as well as network analysis tools based on them. As discussion followed each chapter of the thesis, here we briefly review the key findings and future directions.

In Chapter 2 we studied the relation between a path defined in a network and a geometrically defined geodesic. We emphasised that in both transitively complete DAGs (such as those that would represent causal sets), as well as DAGs that are neither transitively reduced nor transitively complete, the longest path in the network may play a role of a network geodesic. We showed this by studying how closely the longest path approximates a straight line—the geodesic—in an RGG built using a Minkowski manifold. In future, analogous models to the Minkowski-RGG studied here may look into the validity of this result in curved spaces as well as real acyclic networks, embedded in an appropriate space.

The order of nodes and edges, inherent in every DAG, gives way to a well-defined longest path in the network. We thus studied the properties of the longest path in a cumulative advantage model of DAGs—the Price model. We have shown

by using both, analytic and numerical analysis, that the length of the lower-bound approximation of the longest path—the reversed greedy path—scales as the logarithm of the size of a network in the Price model. Numerically, we also found the true longest path also to be scaling with the logarithm of the network size. This result holds true for a large number of parameter values.

Many important network tools are based on network statistics. In Chapter 4 and Chapter 5 we studied two of those tools: centrality and community detection. Here we emphasised that an edge in a DAG represents an order in the data and is an important feature which should not be ignored. This constraint and its effects are not always accounted for in traditional network measures, suggesting that network analysis methods need to be reexamined when applied to DAGs.

First we showed that centrality creates an order of nodes which, in many ways, closely replicates the natural order of the system. This is an unwanted feature, as we observe old nodes as inevitably central. We proposed several solutions how to determine centrality of nodes, irrespective of their positions in topological ordering. These solutions were novel centrality measures and a measure of mutual information between topological order and centrality order. Future work in this area needs to concentrate on testing the usefulness of the proposed methods in real networks, such as citation graphs.

In Chapter 5 we proposed a novel tool to find communities of nodes at the same hierarchical layer in the DAG. Such nodes may be research articles with similar publication dates, species in a food web at the same trophic level, or related libraries which do not depend on each other in a software package. Conventional DAG layering algorithms have no notion of similarity, so all nodes within a layer would be “similar”. Conversely, community detection algorithms would not consider nodes at the same hierarchical layer as very similar. We combined the two DAG partitioning tools to obtain the siblinarity communities. Our tool

may potentially be useful to coarse grain large data sets that represent hierarchical relations (e.g. group together similar species in a food web), as well as a recommendation tool of research articles (similar to “related records” in the Science Citation Index [23]). Again, how practical our approach is in large-scale real-world data sets is an important direction of future research.

During this work, I have come to understand the importance of the order often found in complex systems. In fact, order is inevitably related to hierarchy, which, in turn, is implicit to complexity. In H. Simon’s words, “complexity frequently takes the form of hierarchy” and it is “one of the central structural schemes that the architect of complexity uses” [137]. Hierarchy can partake multiple forms, e.g. “order”, “level”, “control”, “inclusion” types [138] and perhaps even more. A unifying feature of any type of hierarchy is that quantitatively, they can all be pictured as a partially ordered set that represent relationships between elements within hierarchy. However, what partially ordered sets do not reveal is the lack of some of transitive relations (i.e. when we do not observe an edge for each element, related in a poset). The absence of some order relations is a key distinguishing feature between DAGs and partially ordered sets, which adds an extra layer of complexity to observed hierarchical structures. In turn, the absence of some edges is an important and valuable signal. So in a way, DAGs are “doubly-complex” systems with the first level of complexity relating to the order a DAG represents, and second, coming from the “missing information” encrypted with missing edges of the underlying partially ordered set. It is my hope that with my work I highlighted the peculiarity of DAGs and shed some light upon their structural features.

Bibliography

- [1] H. Simon, “On a class of skew distribution functions,” *Biometrika*, vol. 42, no. 3-4, pp. 425–440, 1955.
- [2] J. Ladyman, J. Lambert, and K. Wiesner, “What is a complex system?,” *European Journal for Philosophy of Science*, vol. 3, pp. 33–67, June 2012.
- [3] J. H. Holland, *Complexity: A Very Short Introduction*. Oxford University Press, July 2014.
- [4] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.
- [5] Q. Yao, T. S. Evans, and K. Christensen, “How the network properties of shareholders vary with investor type and country,” *PLoS ONE*, vol. 14, p. e0220965, Aug. 2019.
- [6] S. Boccaletti, G. Bianconi, R. Criado, C. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, “The structure and dynamics of multilayer networks,” *Physics Reports*, vol. 544, pp. 1–122, Nov. 2014.
- [7] A. Kirkley, G. T. Cantwell, and M. E. J. Newman, “Balance in signed networks,” *Physical Review E*, vol. 99, Jan. 2019.
- [8] G. Kossinets, “Empirical analysis of an evolving social network,” *Science*, vol. 311, pp. 88–90, Jan. 2006.

- [9] J. Scott, “Social network analysis,” *Sociology*, vol. 22, pp. 109–127, Feb. 1988.
- [10] R. A. Hill and R. I. M. Dunbar, “Social network size in humans,” *Human Nature*, vol. 14, pp. 53–72, Mar. 2003.
- [11] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach*. New York, NY, USA: Cambridge University Press, 2004.
- [12] H. W. Park and M. Thelwall, “Hyperlink analyses of the World Wide Web: a review,” *Journal of Computer-Mediated Communication*, vol. 8, pp. 0–0, June 2006.
- [13] S. E. Petersen and O. Sporns, “Brain networks and cognitive architectures,” *Neuron*, vol. 88, pp. 207–219, Oct. 2015.
- [14] C. Seguin, M. P. van den Heuvel, and A. Zalesky, “Navigation of brain networks,” *Proceedings of the National Academy of Sciences*, vol. 115, pp. 6297–6302, May 2018.
- [15] H.-J. Park and K. Friston, “Structural and functional brain networks: From connections to cognition,” *Science*, vol. 342, pp. 1238411–1238411, Oct. 2013.
- [16] V. Colizza, A. Flammini, A. Maritan, and A. Vespignani, “Characterization and modeling of protein–protein interaction networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 352, pp. 1–27, July 2005.
- [17] M. Coscia, “Using arborescences to estimate hierarchicalness in directed complex networks,” *PLOS ONE*, vol. 13, p. e0190825, jan 2018.

- [18] L. Papadopoulos, M. A. Porter, K. E. Daniels, and D. S. Bassett, “Network analysis of particles and grains,” *Journal of Complex Networks*, vol. 6, pp. 485–565, Apr. 2018.
- [19] D. B. Johnson, “Efficient algorithms for shortest paths in sparse networks,” *Journal of the ACM (JACM)*, vol. 24, pp. 1–13, Jan. 1977.
- [20] N. Masuda, M. A. Porter, and R. Lambiotte, “Random walks and diffusion on networks,” *Physics Reports*, vol. 716-717, pp. 1–58, Nov. 2017.
- [21] B. Karrer and M. E. J. Newman, “Random acyclic networks,” *Physical Review Letters*, vol. 102, Mar. 2009.
- [22] P. Holme and J. Saramäki, “Temporal networks,” *Physics Reports*, vol. 519, pp. 97–125, Oct. 2012.
- [23] E. Garfield, “Citation indexes for science: a new dimension in documentation through association of ideas,” *Science*, vol. 122, pp. 108–111, July 1955.
- [24] J. K. Kones, K. Soetaert, D. van Oevelen, and J. O. Owino, “Are network indices robust indicators of food web functioning? A Monte Carlo approach,” *Ecological Modelling*, vol. 220, pp. 370–382, Feb. 2009.
- [25] J. Sun, D. Ajwani, P. K. Nicholson, A. Sala, and S. Parthasarathy, “Breaking cycles in noisy hierarchies,” in *Proceedings of the 2017 ACM on Web Science Conference - WebSci '17*, (New York), ACM Press, 2017.
- [26] E. Letizia, P. Barucca, and F. Lillo, “Resolution of ranking hierarchies in directed networks,” *PLoS ONE*, vol. 13, pp. 1–25, 02 2018.
- [27] E. Hou, N. Ansari, and H. Ren, “A genetic algorithm for multiprocessor scheduling,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 2, pp. 113–120, 1994.

- [28] T. G.-N. Finn B. Jensen, *Bayesian networks and decision graphs*. Information Science and Statistics, Springer, 2nd ed., 2007.
- [29] B. Schröder, *Ordered Sets*. Springer International Publishing, 2016.
- [30] M. Yannakakis, “Graph-theoretic methods in database theory,” in *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pp. 230–242, 1990.
- [31] A. V. Aho, M. R. Garey, and J. D. Ullman, “The transitive reduction of a directed graph,” *SIAM Journal on Computing*, vol. 1, pp. 131–137, June 1972.
- [32] A. B. Kahn, “Topological sorting of large networks,” *Communications of the ACM*, vol. 5, pp. 558–562, Nov. 1962.
- [33] U. Manber, *Introduction to algorithms: a creative approach*. Addison-Wesley Longman Publishing Co., Inc., 1989.
- [34] S. Surya, “The causal set approach to quantum gravity,” *Living Reviews in Relativity*, vol. 22, Sept. 2019.
- [35] R. van der Hofstad, *Random Graphs and Complex Networks*. Cambridge University Press, 2016.
- [36] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [37] P. Erdős and A. Rényi, “On the evolution of random graphs,” *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.
- [38] D. S. Price, “A general theory of bibliometric and other cumulative advantage processes,” *J.Amer.Soc.Inform.Sci.*, vol. 27, pp. 292–306, 1976.

- [39] T. S. Evans, L. Calmon, and V. Vasiliauskaite, “Longest Path in the Price Model,” *arXiv preprint arXiv:1903.03667*, 2019.
- [40] “Kdd cup.” cs.cornell.edu/projects/kddcup/datasets.html, 2003. Online; accessed 10/10/17.
- [41] J. Gläser, W. Glänzel, and A. Scharnhorst, “Same data—different results? Towards a comparative approach to the identification of thematic structures in science,” *Scientometrics*, vol. 111, pp. 981–998, Mar. 2017.
- [42] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI Magazine*, vol. 29, p. 93, Sept. 2008.
- [43] L. Getoor, “Link-based classification,” in *Advanced methods for knowledge discovery from complex data*, pp. 189–207, Springer, 2005.
- [44] R. Ulanowicz, C. Bondavalli, and M. S Egnatovich, “Network analysis of trophic dynamics in South Florida ecosystem, FY 97: The Florida bay ecosystem,” *Annual Report to the United States Geological Service Biological Resources Division. Ref. No. [UMCES]CBL, 01* 1998.
- [45] A. R. Benson, D. F. Gleich, and J. Leskovec, “Higher-order organization of complex networks,” *Science*, vol. 353, no. 6295, pp. 163–166, 2016.
- [46] S. Milgram, “The small world problem,” *Psychology today*, vol. 2, no. 1, pp. 60–67, 1967.
- [47] *Fundamentals of Brain Network Analysis*. Elsevier, 2016.
- [48] E. Bullmore and O. Sporns, “Complex brain networks: graph theoretical analysis of structural and functional systems,” *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 186–198, 2009.

- [49] T. Nishikawa, A. E. Motter, Y.-C. Lai, and F. C. Hoppensteadt, “Heterogeneity in oscillator networks: Are smaller worlds easier to synchronize?,” *Physical Review Letters*, vol. 91, July 2003.
- [50] E. Deza and M. M. Deza, *Encyclopedia of distances*. Springer Berlin Heidelberg, 2009.
- [51] S. M. Carroll, *Spacetime and geometry*. Cambridge University Press, 2004.
- [52] M. Spradlin, A. Strominger, and A. Volovich, “Les Houches lectures on de Sitter space,” *arXiv preprint arXiv:hep-th/0110007*, 2001.
- [53] M. Barthélemy, “Spatial networks,” *Physics Reports*, vol. 499, pp. 1–101, Feb. 2011.
- [54] J. Sienkiewicz and J. A. Hołyst, “Statistical analysis of 22 public transport networks in Poland,” *Physical Review E*, vol. 72, Oct. 2005.
- [55] G. A. Pagani and M. Aiello, “The power grid as a complex network: survey,” *Physica A: Statistical Mechanics and its Applications*, vol. 392, pp. 2688–2700, June 2013.
- [56] M. Rahmani and H. N. Koutsopoulos, “Path inference from sparse floating car data for urban networks,” *Transportation Research Part C: Emerging Technologies*, vol. 30, pp. 41–54, May 2013.
- [57] W. Cunningham, K. Zuev, and D. Krioukov, “Navigability of random geometric graphs in the universe and other spacetimes,” *Scientific Reports*, vol. 7, Aug. 2017.
- [58] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá, “Hyperbolic geometry of complex networks,” *Physical Review E*, vol. 82, Sep 2010.

- [59] F. Papadopoulos, M. Kitsak, M. Á. Serrano, M. Boguñá, and D. Krioukov, “Popularity versus similarity in growing networks,” *Nature*, vol. 489, no. 7417, pp. 537–540, 2012.
- [60] K. Zuev, M. Boguñá, G. Bianconi, and D. Krioukov, “Emergence of soft communities from geometric preferential attachment,” *Scientific reports*, vol. 5, p. 9421, 2015.
- [61] M. Boguñá, D. Krioukov, and K. C. Claffy, “Navigability of complex networks,” *Nature Physics*, vol. 5, no. 1, pp. 74–80, 2009.
- [62] B. Waxman, “Routing of multipoint connections,” *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.
- [63] W. Nelson, M. Zitnik, B. Wang, J. Leskovec, A. Goldenberg, and R. Sharan, “To embed or not: network embedding as a paradigm in computational biology,” *Frontiers in Genetics*, vol. 10, May 2019.
- [64] R. B. Ellis, J. L. Martin, and C. Yan, “Random geometric graph diameter in the unit ball,” *Algorithmica*, vol. 47, no. 4, pp. 421–438, 2007.
- [65] J. Díaz, D. Mitsche, G. Perarnau, and X. Pérez-Giménez, “On the relation between graph distance and Euclidean distance in random geometric graphs,” *Advances in Applied Probability*, vol. 48, no. 3, pp. 848–864, 2016.
- [66] G. Brightwell and R. Gregory, “Structure of random discrete spacetime,” *Physical Review Letters*, vol. 66, no. 3, pp. 260–263, 1991.
- [67] R. Ilie, G. B. Thompson, and D. D. Reid, “A numerical study of the correspondence between paths in a causal set and geodesics in the continuum,” *Classical and Quantum Gravity*, vol. 23, no. 10, p. 3275, 2006.

- [68] B. Bollobás and G. Brightwell, “Box-spaces and random partial orders,” *Transactions of the American Mathematical Society*, vol. 324, no. 1, p. 59, 1991.
- [69] J. R. Clough and T. S. Evans, “Embedding graphs in Lorentzian spacetime,” *PLoS ONE*, vol. 12, no. 11, p. e0187301, 2017.
- [70] J. R. Clough and T. S. Evans, “What is the dimension of citation space?,” *Physica A: Statistical Mechanics and its Applications*, vol. 448, pp. 235–247, Apr. 2016.
- [71] J. R. Clough, J. Gollings, T. V. Loach, and T. S. Evans, “Transitive reduction of citation networks,” *Journal of Complex Networks*, vol. 3, no. 2, pp. 189–203, 2014.
- [72] E. W. Weisstein, “Point-line distance—2-dimensional.” mathworld.wolfram.com, 2002. Online; accessed 03/03/20.
- [73] E. W. Weisstein, “Point-line distance—3-dimensional.” mathworld.wolfram.com, 2002. Online; accessed 03/03/20.
- [74] D. J. de Solla Price, “Networks of scientific papers,” *Science*, vol. 149, pp. 510–515, July 1965.
- [75] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, “Numerical recipes in C.,” *Cambridge: Cambridge University*, 1992.
- [76] A. Bavelas, “A mathematical model for group structures,” *Human Organization*, vol. 7, no. 3, pp. 16–30, 1948.
- [77] K. D. Mackenzie, “Structural centrality in communications networks,” *Psychometrika*, vol. 31, no. 1, pp. 17–25, 1966.

- [78] M. A. Beauchamp, “An improved index of centrality,” *Behavioral Science*, vol. 10, no. 2, pp. 161–163, 1965.
- [79] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: Bringing order to the web.,” Technical Report 1999-66, Stanford InfoLab, November 1999.
- [80] M. Ashtiani, A. Salehzadeh-Yazdi, Z. Razaghi-Moghadam, H. Hennig, O. Wolkenhauer, M. Mirzaie, and M. Jafari, “A systematic survey of centrality measures for protein-protein interaction networks,” *BMC Systems Biology*, vol. 12, July 2018.
- [81] P. Holme, “Congestion and centrality in traffic flow on complex networks,” *Advances in Complex Systems*, vol. 06, pp. 163–176, June 2003.
- [82] S. Derrible, “Network centrality of metro systems,” *PLoS ONE*, vol. 7, p. e40575, July 2012.
- [83] J. Wang, H. Mo, F. Wang, and F. Jin, “Exploring the network structure and nodal centrality of China’s air transport network: a complex network approach,” *Journal of Transport Geography*, vol. 19, pp. 712–721, July 2011.
- [84] M. K. Sparrow, “The application of network analysis to criminal intelligence: An assessment of the prospects,” *Social Networks*, vol. 13, pp. 251–274, Sept. 1991.
- [85] E. Garfield, “Citation frequency as a measure of research activity and performance,” *Essays of an Information Scientist*, vol. 1, no. 2, pp. 406–408, 1973.
- [86] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Social Networks*, vol. 1, no. 3, pp. 215 – 239, 1978.

- [87] G. Sabidussi, “The centrality index of a graph,” *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.
- [88] J. Nieminen, “On the centrality in a graph,” *Scandinavian Journal of Psychology*, vol. 15, no. 1, pp. 332–336, 1974.
- [89] P. Boldi and S. Vigna, “Axioms for centrality,” *Internet Mathematics*, vol. 10, no. 3-4, pp. 222–262, 2014.
- [90] S. Bandyopadhyay, M. N. Murty, and R. Narayanan, “A generic axiomatic characterization of centrality measures in social network,” *arXiv preprint arXiv:1703.07580*, 2017.
- [91] D. Schoch and U. Brandes, “Re-conceptualizing centrality in social networks,” *European Journal of Applied Mathematics*, vol. 27, no. 06, pp. 971–985, 2016.
- [92] A. Landherr, B. Friedl, and J. Heidemann, “A critical review of centrality measures in social networks,” *Business & Information Systems Engineering*, vol. 2, pp. 371–385, Oct. 2010.
- [93] P. Bonacich, “Power and centrality: A family of measures,” *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.
- [94] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Computer Networks and ISDN Systems*, vol. 30, pp. 107–117, Apr. 1998.
- [95] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, 1994.

- [96] A. R. Cirtwill, G. V. D. Riva, M. P. Gaiarsa, M. D. Bimler, E. F. Cagua, C. Coux, and D. M. Dehling, “A review of species role concepts in food webs,” *Food Webs*, vol. 16, p. e00093, Sept. 2018.
- [97] L. Leydesdorff, L. Bornmann, J. A. Comins, and S. Milojević, “Citations: indicators of quality? The impact fallacy,” *Frontiers in Research Metrics and Analytics*, vol. 1, 2016.
- [98] O. Kinouchi, A. J. Holanda, and G. C. Cardoso, “Citation network centrality: a scientific awards predictor?,” *arXiv preprint arXiv:1910.02369*.
- [99] P. Chen, H. Xie, S. Maslov, and S. Redner, “Finding scientific gems with Google’s PageRank algorithm,” *Journal of Informetrics*, vol. 1, pp. 8–15, Jan. 2007.
- [100] M. T. Joseph and D. R. Radev, “Citation analysis, centrality, and the acl anthology,” Tech. Rep. CSE-TR-535-07, University of Michigan. Department of Electrical Engineering and Computer Science, 2007.
- [101] M. S. Mariani, M. Medo, and F. Lafond, “Early identification of important patents: design and validation of citation network metrics,” *Technological Forecasting and Social Change*, vol. 146, pp. 644–654, Sept. 2019.
- [102] M. S. Mariani, M. Medo, and Y.-C. Zhang, “Identification of milestone papers through time-balanced network centrality,” *Journal of Informetrics*, vol. 10, pp. 1207–1223, Nov. 2016.
- [103] J. C. S. Freitas, R. Barbastefano, and D. Carvalho, “Identifying influential patents in citation networks using enhanced VoteRank centrality,” *arXiv preprint arXiv:1811.01638*.
- [104] T. W. Valente, K. Coronges, C. Lakon, and E. Costenbader, “How correlated

- are network centrality measures?,” *Connections (Toronto, Ont.)*, vol. 28, no. 1, p. 16, 2008.
- [105] C.-Y. Lee, “Correlations among centrality measures in complex networks,” *arXiv preprint arXiv:physics/0605220*, 2006.
- [106] C. Li, Q. Li, P. V. Mieghem, H. E. Stanley, and H. Wang, “Correlation between centrality metrics and their application to the opinion model,” *The European Physical Journal B*, vol. 88, Mar. 2015.
- [107] F. Jordán, Z. Benedek, and J. Podani, “Quantifying positional importance in food webs: a comparison of centrality indices,” *Ecological Modelling*, vol. 205, pp. 270–275, July 2007.
- [108] M. G. Kendall, “A new measure of rank correlation,” *Biometrika*, vol. 30, pp. 81–93, June 1938.
- [109] G. Tibély, P. Pollner, and G. Palla, “Partial order similarity based on mutual information,” *arXiv preprint arXiv:1601.05922*, 2016.
- [110] R. K. Merton, “Resistance to the systematic study of multiple discoveries in science,” *European Journal of Sociology*, vol. 4, pp. 237–282, Dec. 1963.
- [111] R. K. Merton, “Singletons and multiples in scientific discovery: a chapter in the sociology of science,” *Proceedings of the American Philosophical Society*, vol. 105, no. 5, pp. 470–486, 1961.
- [112] Wikipedia contributors, “List of multiple discoveries.” en.wikipedia.org/wiki/List_of_multiple_discoveries, 2020. Online; accessed 7/02/20.
- [113] V. Vasiliauskaite and T. S. Evans, “Making communities show respect for order,” *Applied Network Science*, vol. 5, Feb. 2020.

- [114] K. W. Boyack and R. Klavans, “Co-citation analysis, bibliographic coupling, and direct citation: which citation approach represents the research front most accurately?,” *Journal of the American Society for Information Science and Technology*, vol. 61, pp. 2389–2404, Dec. 2010.
- [115] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, pp. 75–174, Feb. 2010.
- [116] M. Newman, “Detecting community structure in networks,” *The European Physical Journal B / Condensed Matter and Complex Systems*, vol. 38, 03 2004.
- [117] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences*, vol. 99, pp. 7821–7826, June 2002.
- [118] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, “Defining and identifying communities in networks,” *Proceedings of the National Academy of Sciences*, vol. 101, pp. 2658–2663, Feb. 2004.
- [119] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. P10008, Oct. 2008.
- [120] E. Abbe, “Community detection and stochastic block models: recent developments,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6446–6531, 2017.
- [121] M. E. J. Newman, “Modularity and community structure in networks,” *Proceedings of the National Academy of Sciences*, vol. 103, pp. 8577–8582, May 2006.

- [122] P. Sah, L. O. Singh, A. Clauset, and S. Bansal, “Exploring community structure in biological networks with random graphs,” *BMC Bioinformatics*, vol. 15, no. 1, p. 220, 2014.
- [123] V. Vasiliauskaite and F. E. Rosas, “Understanding complexity via network theory: a gentle introduction,” *arXiv preprint arXiv:2004.14845*, 2020.
- [124] K. Sugiyama, S. Tagawa, and M. Toda, “Methods for visual understanding of hierarchical system structures,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 2, pp. 109–125, 1981.
- [125] L. Mirsky, “A dual of Dilworth’s decomposition theorem,” *The American Mathematical Monthly*, vol. 78, no. 8, pp. 876–877, 1971.
- [126] E. Gansner, E. Koutsofios, S. North, and K.-P. Vo, “A technique for drawing directed graphs,” *IEEE Transactions on Software Engineering*, vol. 19, pp. 214–230, Mar. 1993.
- [127] A. Tarassov, N. S. Nikolov, and J. Branke, “A heuristic for minimum-width graph layering with consideration of dummy nodes,” in *Experimental and Efficient Algorithms*, pp. 570–583, Springer Berlin Heidelberg, 2004.
- [128] N. S. Nikolov and A. Tarassov, “Graph layering by promotion of nodes,” *Discrete Applied Mathematics*, vol. 154, pp. 848–860, Apr. 2006.
- [129] P. Healy and N. S. Nikolov, “A branch-and-cut approach to the directed acyclic graph layering problem,” in *Graph Drawing* (M. T. Goodrich and S. G. Kobourov, eds.), (Berlin, Heidelberg), pp. 98–109, Springer Berlin Heidelberg, 2002.
- [130] M. Hopkins, “Layerwidth: Analysis of a new metric for directed acyclic graphs,” *arXiv preprint arXiv:1212.2479*, 2012.

- [131] P. Healy and N. Nikolov, “Hierarchical drawing algorithms,” in *Handbook of Graph Drawing and Visualization* (R. Tamassia, ed.), ch. 13, pp. 409–454, Florida: CRC Press, 08 2013.
- [132] J. Martyn, “Bibliographic coupling,” *Journal of Documentation*, vol. 20, no. 4, pp. 236–236, 1964.
- [133] H. Small, “Co-citation in the scientific literature: a new measure of the relationship between two documents,” *Journal of the American Society for Information Science*, vol. 24, no. 4, pp. 265–269, 1973.
- [134] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, no. 2, p. 026113, 2004.
- [135] J. Reichardt and S. Bornholdt, “Statistical mechanics of community detection,” *Phys. Rev. E*, vol. 74, no. 1, p. 016110, 2006.
- [136] L. Jost, “Entropy and diversity,” *Oikos*, vol. 113, no. 2, pp. 363–375, 2006.
- [137] H. A. Simon, “The architecture of complexity,” *Proceedings of the American Philosophical Society*, vol. 106, no. 6, pp. 467–482, 1962.
- [138] D. Lane, “Hierarchy, complexity, society,” in *Hierarchy in natural and social sciences*, pp. 81–119, Springer, 2006.

A Space-Time Lattice Model

This model is a simple DAG where the nodes are placed on a square lattice at (t, x) where tL and xL are integers between 0 and $(L - 1)$. To place edges between the node at the points of our lattice, we use the Manhattan distance $d(n, m) = |t_n - t_m| + |x_n - x_m|$ for the distance between nodes n and m of coordinates (t_n, x_n) and (t_m, x_m) respectively. We add a directed edge from n to m with probability $p(n, m)$ where

$$p(n, m) = \begin{cases} 0 & \text{if } t_n \geq t_m \\ 1 - \frac{d(n, m)}{D} & \text{if } t_n < t_m. \end{cases} \quad (1)$$

The edges from n to m only exist if $t_n < t_m$ and it is this arrow-of-time which ensures that we will always have a DAG.

The idea behind this DAG model is there is a natural hierarchy given by the t -coordinate which guarantees acyclicity. In addition, nodes which are close in their space and time coordinates will often have large numbers of common neighbours, both successors and predecessors, whereas nodes that are distant in space or time will have few neighbours in common. So in the visualisations we expect to see nodes of the same time coordinate and close in their space coordinates, to be placed in the same antichain community. However, as the links are placed with a stochastic mechanism, we will sometimes see nodes from neighbouring layers grouped together. That is the antichain structure in a given realisation of our model is not a perfect match to the natural layers defined by the time coordinate. This simulates what one finds in real data sets. For instance, two papers written on a similar topic would be represented by nodes with a similar x coordinate in this model. If they were written independently at the same time then they could not be connected but if published at slightly different times it might be possible

for the earlier paper to be cited by the later paper.

B Price Model with Subject Fields

The Price model for citation networks [38] produces a DAG with a fat-tailed (power-law) distribution for the out-degree of nodes in our conventions which represents the citation count of papers. We modify the Price model by assigning each paper to a ‘field’ and the edges, citations between papers, are biased so they are usually between papers in the same field. While an unrealistic model of citation networks in many ways, it contains three key features of real citation networks: the order of papers imposed by time, the fat-tailed citation count distribution, and the preference of most papers to cite papers within a similar field. We use it as a DAG with a planted partition to enable us to make controlled comparisons between the different community detection approaches discussed.

The model defines a sequence of networks $\mathcal{G}(t)$ where t is a positive integer playing the role of time and which gives us an order to the nodes in our networks. Each graph $\mathcal{G}(t)$ has t nodes with vertex set $\mathcal{V}(t)$ and edge set $\mathcal{E}(t)$. In our notation, the node $u(s)$ is always the node added at step s in the process, so $u(s) \in \mathcal{V}(t)$ provided $0 < s \leq t$.

The nodes in these networks are also partitioned into different fields, that is each node $u(t)$ is in one of F fields. The fields will be labelled by integers between 0 and $(F - 1)$ with $f(t) \in \{0, 1, \dots, (F - 1)\}$ denoting the field of node $u(t)$. This creates a sequence of partitions $\mathfrak{F}(t)$ of our nodes where $\mathfrak{F}(t) = \{\mathcal{F}_f(t) | f \in \{0, 1, \dots, (F - 1)\}\}$ and $\mathcal{F}_f(t) \subseteq \mathcal{V}(t)$. A node $u(s)$, for $0 < s \leq t$, belongs to a element $\mathcal{F}_{f(s)}(t) \in \mathfrak{F}(t)$, the set of papers at time t in the same field $f(s)$ as the paper published at time s .

To create the next graph in the sequence, $\mathcal{G}(t + 1)$, we first add a new node

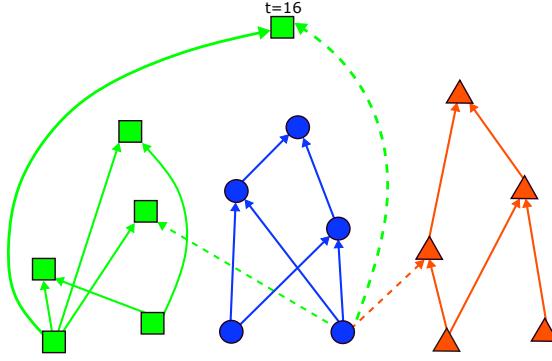


Figure B1: An illustration of the Price model with two edges added per node ($m = 2$) and three fields ($F = 3$) as indicated by the three types of node. Time increases as you move up the diagram so the vertical ordering of the nodes represents a total order in the DAG. Solid (dashed) lines represent citations between papers of identical (different) fields. A new node labelled $16 = (t + 1)$ in field 0 (green squares) is added an existing network of fifteen nodes. Here we suggest the two edges added to node 16 are to the two nodes of highest degree (the cumulative advantage bias) with one in the same field (continuous edge) and one in a different field (dashed edge). Figure reproduced from [113].

$u(t+1)$ to the vertex set, so $\mathcal{V}(t+1) = \mathcal{V}(t) \cup \{u_{t+1}\}$. This new node is assigned to f chosen uniformly at random from the set of F possible field labels.

We now add m directed edges to this new node $v(t+1)$ from existing nodes $u(s)$ where $s < t$. To encode the ‘‘cumulative advantage’’ principle of Price, that is the higher the current citation count of a paper the more likely it is to be cited, we can chose nodes $u(s)$ from the existing nodes $\mathcal{V}(t)$ in the network $\mathcal{G}(t)$ with probability $\Pi^{(\text{CA})}(t, s)$ defined as¹

$$\Pi^{(\text{CA})}(t, s) = \frac{k^{(\text{out})}(t, s) + 1}{|\mathcal{V}(t)| + |\mathcal{E}(t)|}. \quad (2)$$

Here $k^{(\text{out})}(t, s)$ is the number of outgoing edges from node $u(s)$ in $\mathcal{G}(t)$, the network at time t . In our conventions, these edges represent citations from later papers to the paper published at time s .

¹Other forms linear in $k^{(\text{out})}(t, s)$ are also easy to work with (for example see [4]), but these variations are not our focus here. We chose to follow the same form as used in Price’s original paper.

C Numerical Reverse Greedy Path Algorithm

Numerically, the reverse greedy path may be found as follows

1. Set initial graph \mathbf{G} with t nodes.
2. Set the reverse greedy path length for the initial t nodes, `greedy_length[s]` for $s = 1, 2, \dots, t$.
3. Set the nearest neighbour along this reverse greedy path length for the initial t nodes, `greedy_neighbour[s]` for $s = 1, 2, \dots, t$.
4. Increment t .
5. Add a new node index t to \mathbf{G} .
6. Update graph \mathbf{G} by adding m new edges, from vertices $\{s\}$ to new node t , which then define the past neighbour set of vertex t , i.e. $\mathcal{N}^{\text{pre}} = \{s\}$.
7. Find the past neighbour with the largest time $s_{\text{gr}} = \max(\mathcal{N}^{\text{pre}}(t))$.
8. Record this closest past neighbour as the last step on the reverse greedy path to t ,
i.e. set `greedy_neighbour[t] = s_{\text{gr}}`.
9. Set the length of the reverse greedy path of new node t to be one more than the length of the reverse greedy path to this nearest past neighbour s_{gr} , so set $\ell(t) = \ell(s_{\text{gr}}) + 1$. Numerically we just need to set
`greedy_length[t] = greedy_length[greedy_neighbour[t]] + 1`.
10. If adding more nodes, return to step 4.

The length of the reverse greedy path from the first node $t = 1$ to a given node t is now stored for each node. The vertices on the unique reverse greedy path can

be found by iterating back though the `greedy_neighbour` property of the nodes.

If the specific path used is not required we need not record this information.

We also note that the length of the longest path, the longest path from the initial node $t = 1$ to any node t , may be tracked numerically in a similar way. Such a longest path always exists but is not unique so we could only ever record one example longest path using this approach.

Glossary

Minkowski space-time flat $1+d$ -dimensional pseudo-Riemannian manifold with a metric $ds^2 = (cdx_0)^2 - \sum_{i=1}^d (dx_i)^2$, where dx_i ($i \in [2, d]$) are components of an infinitesimal coordinate displacement vector coordinates, c is the speed of light and x_0 is the temporal coordinate. Page 32

antichain partition a set of antichains such that each node of a graph is in one antichain. Page 92

antichain a set of nodes from the graph, such that there is no path between any two nodes. Page 21

bibliographic coupling in scientometrics, the measure of similarity of two documents, quantified by the size of the overlap of their bibliographies. Page 97

centrality network measure to quantify importance of nodes Page 61, 65

co-citation in scientometrics, the measure of similarity of two documents, quantified by the size of the overlap of the documents citing them. Page 97

depth of a node is the length of the longest path from it to any node with zero in-degree. Page 21

directed acyclic graph directed network with no directed cycles. Page 17

directed neighbourhood inclusion preorder neighbourhoodinclusion for DAGs.

Page 74

future centrality Centrality, related to paths in a given node's future—its descendants. Page 66

height of a node is the length of the longest path to it from any node with zero in-degree. Page 21

index of a node u is i if it is in the i^{th} place in the topological order. Page 21, 72

induced graph of an original graph is a graph in which each node represents a single community and an edge between two nodes of induced graph is given a weight equal to the sum of the weights of all the edges pointing from any node of one antichain to any of the nodes in the another antichain. Page 102

neighbourhood inclusion preorder a preorder relation between nodes \preceq_N , such that when a node u is a neighbour of a node v and u 's neighbourhood is a subset of neighbours of v , $u \leq v$. Page 65

past centrality Centrality, related to paths from a given node's past—its ancestors—to the node. Page 66

path a sequence of nodes, where an edge from each vertex to the next vertex in the sequence exists, with no repeated nodes. Page 15

reduced degree degree (either $k^{(\text{in})}$ or $k^{(\text{out})}$) of a node in a DAG after transitive reduction. Page 70

reverse greedy path suppose a DAG where edges point in the direction of increasing time. Then for a node t the greedy path follows to an earlier node s (against edge direction), which is the most recent predecessor. This process is continued until the source node is found. Page 47

shortest-path-based centrality Centrality, based on shortest paths related to a node: paths that either originate, terminate or both there Page 66

siblinarity quality function of the antichain partition \mathfrak{A} . Page 98

topological order linear sort of nodes in a DAG such that u comes before v if $u \prec v$. Page 21

total centrality centrality based on both, future and past sets of a node. Page 67

transitive reduction edge removal procedure acted upon a DAG, during which all edges that can be removed, are removed, given that the connectivity in the network is preserved. Page 20

Acronyms

DAG DIRECTED ACYCLIC GRAPH Page 17

PPP POISSON POINT PROCESS Page 34

RGG RANDOM GEOMETRIC GRAPH Page 33

TR TRANSITIVE REDUCTION Page 20, 70

List of symbols

\mathfrak{A} antichain partition. Page 92

\mathbf{A} adjacency matrix. $A_{i_u j_v} = 1$ if there is an edge between u, v . Page 13

\mathcal{A} antichain. Page 21, 91

$\tilde{\mathbf{A}}$ similarity matrix used for siblinarity optimisation, depicts neighbourhood overlap between nodes. Page 99

C centrality function. Page 65

$C^{(\text{future})}$ function of future centrality. Page 66

$C^{(\text{past})}$ function of past centrality. Page 66

c speed of light. Page 32

$d(u)$ depth of node u . Page 21

D dimension of space. Page 30

\mathcal{E} a set of edges in the network. Page 12

E number of edges. Page 12

$\mathcal{F}(u)$ future of u . Page 22

\mathcal{G} network, a pair of nodes and edges. Page 12

$\tilde{\mathcal{G}}$ network with adjacency matrix $\tilde{\mathbf{A}}$. Page 99

\mathcal{H} induced graph Page 102

$h(u)$ height of node u . Page 21

$k^{(\text{in})}$ in-degree. Page 14

κ_u strength of edges attached to a node u in a graph $\tilde{\mathcal{G}}$. Page 99

$k^{(\text{out})}$ out-degree. Page 14

λ resolution parameter in siblinarity function. Page 99

\mathcal{L} a path. Page 15

L length of the longest path. Page 21

\mathcal{L}^{LP} the longest path in a DAG. Page 21

\mathcal{N} a set of node's neighbours. Page 14

N number of nodes. Page 12

$\mathcal{P}(v)$ past of v . Page 22

\mathcal{N}^{pre} a set of node's predecessors. Page 14

$S(\mathfrak{A})$ siblinarity. Page 98

\mathcal{N}^{suc} a set of node's successors. Page 14

\mathcal{V} a set of nodes in the network. Page 12

W total strength of edges in $\tilde{\mathcal{G}}$. Page 99