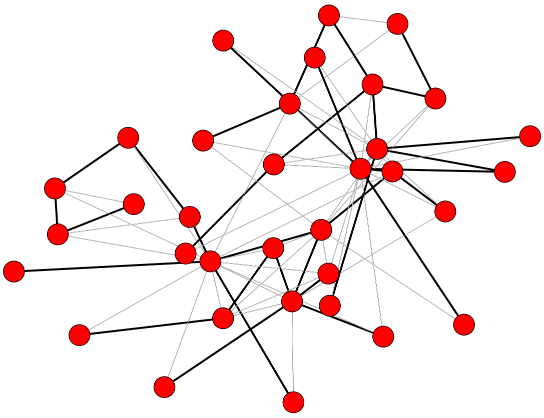


Distance Backbones of Weighted Graphs

Computing path-lengths

Miguel Bernardo Guerreiro Pereira

Thesis to obtain the Master of Science Degree in

Applied Mathematics and Computation

Supervisors: Prof. Luís Mateus Rocha
Prof. Pedro Martins Rodrigues

Examination Committee

Chairperson: Prof. Luís Barreira
Supervisor: Prof. Luís Mateus Rocha
Member of the Committee: Prof. João Xavier

December 2023

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Este trabalho não teria sido possível sem o contributo da minha família. Obrigado a todos, em especial aos meus pais e ao meu irmão, que me deram tanto para que eu conseguisse ter chegado aqui. À minha avó, obrigado pelo carinho, ao meu avô obrigado pelos livros e pela curiosidade que me transmitiste e aos meus tios obrigado pelo apoio. Aos meus amigos, obrigado pelas discussões sobre os mais variados temas (sempre tão pertinentes) e em especial à Valéria pela sua paciência e ajuda minunciosa. Sem paralelo, obrigado à Rita por ser tudo em todo o lado ao mesmo tempo, este trabalho também foi teu.

I also would like to thank every mentor that has inspired me throughout my academic path, namely Prof. Paulo Doutor, who introduced me to university level mathematics, Prof. Ludwig Krippahl for enticing me the love for programming, Prof. Pedro Ribeiro, who inspired me with his motivating teaching, even in the tough times of online classes, and Dr. Gonzalo de Polavieja who allowed me to develop my first research project under his kind guidance.

Obviously, I would like to thank everyone that was directly involved in this endeavor. Firstly, Prof. Luís Rocha, who accepted my request to develop my master thesis at his Lab and allowed me to be apart of a truly welcoming research group. For everyone at the CASCI Lab, thank you for all the help and advices. A special thanks to Felipe for the discussions and for his companionship and advice on all matters regarding this thesis and bordering subjects. Also, thanks to David Paños for the suggestions and always exciting discussions and everyone else at the Instituto Gulbenkian de Ciência that welcomed me so dearly. To Prof. Pedro Martins Rodrigues, a special thanks for enabling me to enter into this adventure and for his supportive attitude even during these difficult semesters.

Lastly, I would like to thank Fundação para a Ciência e a Tecnologia (FCT) that supported this work in the scope of the grant 2022.09122.PTDC. (DOI: 10.54499/2022.09122.PTDC) titled “Redundancy effects on spread and control in network dynamics: applications in computational biomedicine”.

Abstract

This dissertation deals with the topic of sparsification of weighted graphs. In particular, it presents in detail the method for obtaining a distance backbone of a weighted graph. The distance backbone is the subgraph that only contains the edges that belong to some shortest-path. The notion of path-length is also generalized and different functions that compute path-lengths are presented. Since each one of these functions is associated with the respective distance backbone, different backbones are compared by considering some examples of networks depicting social interactions. Lastly, it is examined how these backbones preserve characteristics of epidemiological simulations carried out on these networks using Susceptible-Infected and Susceptible-Infected-Recovered models.

Keywords

Weighted Graphs; Path-Length; Shortest-Path; Distance Backbone; Epidemiological Models

Resumo

Esta dissertação aborda o tema da esparsificação de grafos pesados. Em particular, apresenta em pormenor o método para obter um esqueleto de distância de um grafo pesado. Esse esqueleto de distância corresponde ao subgrafo que contém apenas as arestas que pertencem a algum caminho mais curto. A noção de comprimento de caminho é também generalizada e são apresentadas diferentes funções que calculam comprimentos de caminho. Como cada uma destas funções está associada ao respetivo esqueleto de distância, em seguida, comparam-se diferentes esqueletos, considerando alguns exemplos de redes de interacções sociais. Por fim, é analisado o modo como estes esqueletos preservam as características de simulações epidemiológicas efectuadas nestas redes utilizando os modelos Suscetível-Infetado e Suscetível-Infetado-Recuperado.

Palavras Chave

Grafos Pesados; Comprimento de caminho; Caminho mais curto; Esqueleto de Distância; Modelos Epidemiológicos

Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction

Contents

1.1 Structure of the thesis

This thesis is structured into 5 chapters. The first one, the introduction, lays the context in which this thesis is based and why the problem addressed is relevant. Also, it presents the main concepts and definitions that are needed to understand the following work. The second chapter concerns the bulk of the theoretical background behind the concept of distance backbone. In this chapter, a choice was made to start by the trivial concepts that give rise to each construction, spanning from the definition and examples of relations until the definition and properties of dioids. A more concise and dry description of the requirements would have been possible but, for clarity and intuitiveness purposes, the choice was made to start from the very basics and give examples along the way. The third chapter is a literature review of the most relevant works in the realm of network sparsification. Unlike in most works, this chapter is only presented after the main theoretical materials since many of the alternative techniques that are mentioned require concepts that are common to the framework of distance backbones. The fourth chapter is dedicated to the experimental results, where the method of distance backbones was implemented on real-world networks to assess its impact in the degree of sparsification of each network and the ability to preserve main features of epidemiological dynamics. Finally, the fifth chapter includes the conclusion of the thesis, where the main results are summarized and possible future work is discussed.

1.2 Motivation

In recent decades, the field of network science has emerged as a cornerstone within the study of complex systems, with seminal papers such as [5], [42] and [13] plunging it into the multidisciplinary nature that it enjoys today and having since become a subject of intense research. This field leverages the mathematical concept of graph, where nodes represent objects and edges represent interactions between them. Complex systems are characterized by their composition of numerous interacting components, and these interactions give rise to emergent behaviors not present in individual parts. They are prevalent throughout nature, spanning from microscopic cellular interactions to the macroscopic level of societal dynamics.

Given the inherent complexity of such systems, interactions are often heterogeneous, exhibiting varying intensities. Representing them effectively often requires the use of weighted graphs. Moreover, the need to analyze increasingly large and dense networks has become a prominent challenge. To facilitate the analysis of these networks and gain a deeper understanding of which interactions are pivotal in dynamic processes, network sparsification techniques have gained more and more relevance. Among various sparsification methods, the one analyzed in this thesis was proposed in [36] [35] and focuses on removing interactions that do not belong to any shortest path. In other words, if there exists an indirect path between two nodes with a shorter length than the direct path, then the direct interaction is omitted.

This concept is encapsulated by the distance backbone framework, that allows for diverse methods of calculating path lengths. The flexibility of considering different path-length calculations is promising, as it can be instrumental in the analysis of various dynamic processes taking place on networks.

The most common approach to path-length calculation in a network is the sum of the weights of interactions, which yields the metric backbone. This particular backbone has already been explored in existing literature, notably in [35] but also in [7], where it was investigated how it preserved community structure and Susceptible-Infected (SI) dynamics in social networks. However, this work endeavors to explore diverse backbones arising from alternative path-length calculation methods (some of them parametric) and assess their effectiveness in preserving specific network dynamics, such as SI and Susceptible-Infected-Recovered (SIR) epidemiological processes. This understanding can be used to predict future behaviours of these networks in similar dynamics and to design strategies to control them.

1.3 Graph Theory and Network Theory Essentials

The holy grail of concepts in this thesis is the one of *graph*. Graph theory began in the city of Königsberg in the 18^{th} century when the mathematician Leonhard Euler tackled a puzzle involving seven bridges. Euler's ease with abstract problems allowed him to represented the landmasses as vertices and the bridges as edges, forming the first mathematical graph. He solved the problem by introducing the concepts of Eulerian paths (a path which crosses every edge exactly one time) and circuits. This marked the birth of graph theory, which has since found applications in diverse fields, from computer science to social sciences, passing through biology. Some of these applications are indeed very relevant in the experimental results chapter 4 of this thesis.

While graph theory focused on abstract relationships between nodes and edges, network science expanded its horizons. In the mid-20th century, real-world networks like social networks, transportation systems, and the World Wide Web demanded more sophisticated analysis. Network science emerged as a multidisciplinary approach, combining graph theory with insights from physics, sociology, and computer science. Network science explores not only the structure but also the dynamics of networks. It studies how networks evolve, spread information, and respond to disruptions. The field has become vital in understanding complex systems, from online social interactions to disease transmissions which are a present theme throughout this dissertation. Due to this accelerated development, several books have been produced that summarise and develop the themes of network science, such as [26] and [4].

Furthermore, the field of network science has served as a primordial tool for the development of the broader area of complex systems and, in this area, the major reference in terms of modelling and computational aspects used for this thesis is [31].

Definition 1.3.1 (Graph)**.** A **simple graph** is a pair (V, E) where V is a set of nodes $\{x_1, x_2, \ldots, x_n\}$

and E is a set of edges containing elements of the form (x_i, x_j) where $x_i, x_j \in V$ and $x_i \neq x_j$. More concisely, in set-theoretic terms, V is a set and $E \subseteq V \times V$. In **undirected graphs**, for each $(x_i, x_j) \in E$ we also have $(x_j, x_i) \in E$, whereas in **directed graphs** that might not happen.

In this work, it is always assumed that both $|V|$ and $|E|$ are finite.

Definition 1.3.2 (Multigraph)**.** A **multi-graph** is a graph that may also have self loops, i.e., edges of the kind (x_i, x_i) where $x_i \in V$, and multi-edges, i.e., more than one edge with the same start and end nodes.

Definition 1.3.3 (Weighted Graph)**.** A **weighted graph** is a graph where each edge has a weight associated, i.e., for every $(x_i, x_j) \in E$, $\exists w_{ij}$ such that $(x_i, x_j) : w_{ij}$. In this work the weights are numerical values.

Definition 1.3.4 (Path)**.** A **path** in a graph is a sequence of edges such that the end node of an edge is the same as the starting node of the following edge. If $V = \{v_1, \ldots, v_n\}$ is the set of nodes then a path would be something of the kind $(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)$, assuming that every edge in this sequence is in the set E of the graph. In this case we say that the path has length $k - 1$.

Definition 1.3.5 (Pointed Circuit)**.** A **pointed circuit** in a graph is a path such that the first node of the first edge is the last node of the last edge (**circuit**), associated with a first node (**pointed**). Therefore a circuit of the kind $(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_1)$ can originate $k - 1$ different pointed circuits. A circuit can also be called a **cycle**.

Definition 1.3.6 (Connected and Disconnected Graph)**.** A a graph is said to be **connected** if, for every pair of nodes $x_i, x_j \in V$ there is a path connecting them. Thus, it is said to be **disconnected** if it is not connected or, in other words, if there are disjoint subsets of nodes V_1 and V_2 such that there is no edge linking a node from V_1 to a node in V_2.

Definition 1.3.7 (Strongly and Weakly Connected Graph)**.** A directed graph is said to be **strongly connected** if, for every $x, y \in V$, there is a directed path from x to y. Also, it is said to be **weakly connected** if, for every $x, y \in V$, there is an undirected path from x to y. An undirected path in this context is a path where one ignores the direction of the edges.

Definition 1.3.8 (Tree)**.** A **tree** is a connected graph without cycles.

Definition 1.3.9 (Degree of a node)**.** Given a graph $G = (V, E)$, the **degree** of a node $x \in V$ is the number of edges that connect x to other nodes and is denoted by $deg(x)$. More precisely,

$$deg(x) = |\{(x_i, x_j) \in E : x_i = x \text{ or } x_j = x\}|$$

In the case of directed graphs there are two types of degrees, in-degree and out-degree, which can be defined in a similar way:

$$indeg(x) = |\{(x_i, x_j) \in E : x_j = x\}|$$

$$outdeg(x) = |\{(x_i, x_j) \in E : x_i = x\}|$$

The ubiquity of graphs in the “real world” makes so that the vocabulary used to describe them and their components is very diverse. Thus, when one wants to mention nodes, one can also say vertices, points, elements or entities. On the other hand, edges can also be called connections, links or arcs. With this definition of graph, there are many nuances that can be introduced to create more or less interesting types of graph objects. Thus, it is important to introduce the most relevant variations that appear in the literature. There are a wide variety of contexts where each one of these types of graphs, or combinations of them can be useful. However, in general, the graphs of interest for this work will be **simple**, **weighted** and **directed** or **undirected**,

Graphs can be represented in different ways, depending on the type of analysis one wishes to make about them. In pedagogical contexts, but not only, they are often represented as diagrams with points and line segments. However, in computational contexts the representations are quite different, existing two main approaches, matrices and lists of lists. In the first case, every matrix entry represents the existence or the weight of the respective edge. For undirected graphs, that value entails the presence of an edge which is usually represented by a 1 whereas the non-presence by a 0. In the second case, a graph is represented by a list (or a dictionary) of nodes, each one associated with a list containing the nodes that are adjacent to it. Some examples of differences in computational costs between these two approaches can be summarized by the following table [1].

	Matrix	**Dictionary of lists**
Check existence of edge	$\mathcal{O}(1)$	$\mathcal{O}(deg(x))$
Traverse adjacent nodes	$\mathcal{O}(\|V\|)$	$\mathcal{O}(deg(x))$
Memory usage	$\mathcal{O}(\|V\|^2)$	$\mathcal{O}(\|E\| + \|V\|)$
Add and remove edges	$\mathcal{O}(1)$	$\mathcal{O}(deg(x))$

Table 1.1: Trade-Offs between computational implementations of Graphs

It’s noteworthy to mention that from the memory usage perspective, a dictionary of lists is better for representing sparse graphs but for very dense graphs (with $|E| \approx |V|^2$), the most memory efficient option is a matrix. Since adjacency matrices will be needed later on, it’s reasonable to define them now.

Definition 1.3.10 (Adjacency Matrix)**.** The **adjacency matrix** of a graph $G = (V, E)$ is a matrix A with

[1] In this table, the big-O notation is a computational complexity way of classifying procedures according to their computational cost. Let $f(x)$ be the function that outputs the cost of a procedure p according to the size x of the input. Then, $f(x)$ is said to be in $\mathcal{O}(g(x))$, if $\exists M \in \mathbb{R}^+$ and $\exists x_0 \in \mathbb{R}$ such that $\forall x \geq x_0 : f(x) \leq Mg(x)$. This notation encapsulates how the cost of a procedure evolves depending on the increase of the size of the input that it gets.

dimensions $|V| \times |V|$, such that, in the case of unweighted graphs:

$$a_{ij} = \begin{cases} 1 & \text{if } (x_i, x_j) \in E \\ 0 & \text{if } (x_i, x_j) \notin E, \end{cases}$$

and, in the case of weighted graphs:

$$a_{ij} = \begin{cases} w_{ij} & \text{if}(x_i, x_j) \in E \\ 0 & \text{if}(x_i, x_j) \notin E. \end{cases}$$

Definition 1.3.11 (Diameter of a Graph)**.** The **diameter** of a graph $G = (V, E)$ is the longest shortest-path in G, where a shortest path is the path that uses fewest edges between any pair of nodes.

Definition 1.3.12 (Bridge in a Graph)**.** A **bridge** of a graph $G = (V, E)$ is an edge $e \in E$ such that the number of connected components of $G' = (V, E \setminus \{e\})$ is larger than the number of connected components of G. The number of connected components in a graph is the number of disjoint maximal sets of edges that create connected graphs.

Definition 1.3.13 (Clustering Coefficient)**.** Given a graph $G = (X, E)$, the **clustering coefficient** of a node $x_i \in X$ is defined as $C(x_i) = \frac{2e_i}{k_i(k_i-1)}$, where e_i denotes the number of edges between the neighbor nodes of x_i and k_i is the degree of x_i. This definition has been expanded to weighted graphs in several different ways [30] and the one used in this work is defined as

$$C(u) = \frac{1}{deg(u)(deg(u)-1))} \sum_{vw} (\hat{w}_{uv} \hat{w}_{uw} \hat{w}_{vw})^{1/3},$$

where $u, v, w \in X$, w_{xy} corresponds to the weight of the edge connecting x and y and $\hat{w}$ is the normalized weight given by $\hat{w} = w / \max\limits_{xy \in E}(w_{xy})$,

Definition 1.3.14 (Density)**.** The **density** of a graph $G = (X, E)$ is given by the proportion $\delta = \frac{2|E|}{|X|(|X|-1)}$, which yields the fraction of edges that exist with respect to the edges that could exist. In the case of directed graphs, the density is given by $\delta = \frac{|E|}{|X|(|X|-1)}$ since for each pair of nodes there are 2 possible edges.

2

Theoretical Background

Contents

This chapter introduces the theoretical concepts that are the main subject in this thesis and that are used further in the experimental results. This background encompasses the theory behind computing transitive closures in weighted (fuzzy) graphs as well as their isomorphic counterpart, distance closures, which are computed over distance graphs. After discussing the parallelism between relations and graphs and how these transitive closures can be computed, several families of different types of closures are presented and analysed. Meanwhile, the concept of distance backbone emerges from the definition of distance closure and then the characteristics of such backbones are inspected as well. Moreover, given that the aim is to study and experiment how the backbones preserve dynamics on networks, there is also an explanation of two kinds of (epidemiological) spreading dynamics on networks that are used in the Experimental Results ahead in Chapter 4.

2.1 Relations and Graphs

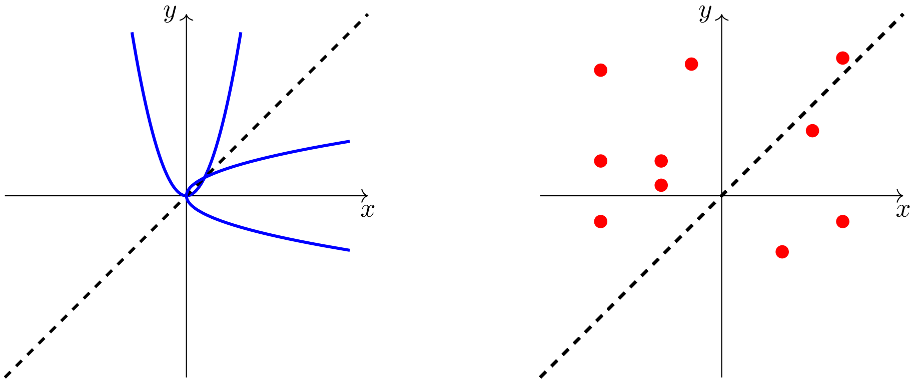
The concept of relation is very widely treated in all sorts of mathematical textbooks. However, in the context of this work, relations are regarded as the building blocks for the central topic addressed which are fuzzy graphs. For this reason the most appropriate bibliography covering these structures and their generalizations can be found in the book by George Klir and Bo Yuan [18].

Definition 2.1.1 (Relation)**.** Let X_1 and X_2 be sets. A (binary) relation, R, is a subset of $X_1 \times X_2$ such that every element of R is an ordered pair of the kind (x_1, x_2) where $x_1 \in X_1$ and $x_1 \in X_1$. If $(x_1, x_2) \in R$ or equivalently, $x_1 R x_2$, then x_1 is said to be R-related to x_2. A relation on a single set X is intended as a subset of ordered pairs from the set $X \times X$. Moreover, a relation is said to be:

- **reflexive**, if for every $a \in X$, we have $(a, a) \in R$,
- **symmetric**, if for every $(a, b) \in R$ there is also $(b, a) \in R$,
- **transitive**, if $(a, c) \in R$ whenever $(a, b) \in R$ and $(b, c) \in R$ for some $c \in X$

If a relation is **reflexive**, **symmetric** and **transitive**, then it is called an **equivalence relation**.

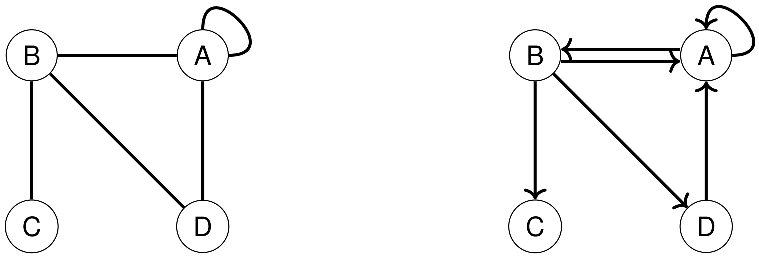
Example 2.1.1. One can take, as a common example of a relation on a set, any function $f : \mathbb{R} \rightarrow \mathbb{R}$ or even any curve or set of points in $\mathbb{R}^2$. If that set of points is symmetric with respect to the curve $y = x$ then we say that the relation is symmetric.



In these cases, the relation on the left is symmetric but the relation on the right is not. Also, neither relation is reflexive because neither one has the curve $y = x$ as a subset.

Example 2.1.2. There is also a vast number of mundane examples of relations that includes parental relations (that can be summarized in genealogy trees) where the set X is the set of people in a family and the related pairs are formed through parental relations, or even the popular game Rock-Paper-Scissors, where the set is $\{R, P, S\}$ and the relation is $\{(R, S), (S, P), (P, R)\}$ meaning that Rock beats Scissors, Scissors beats Paper and Paper beats Rock. Clearly, none of these are symmetric nor reflexive relations.

Example 2.1.3 (Graphs are Relations)**.** Another prompt general example of a relation on a set is a graph. If we consider any graph (possibly with self-loops), $G = (V, E)$, whose vertices are the elements of the set V then its edges, the set E, constitute the relation which is a subset of V^2. In this case, the relation can be interpreted as symmetric if and only if the graph is undirected.



As it is clear from this depiction, the first graph represents the symmetric relation defined by $R_1 = \{(A, A), (A, B), (B, A), (A, D), (D, A), (B, D), (D, B), (B, C), (C, B)\}$ and the second graph represents the relation $R_2 = \{(A, A), (A, B), (B, A), (D, A), (B, D), (B, C)\}$.

In the other way around, a similar parallelism can also be made. When given any relation, one can convert it to a (possibly disconnected) graph. Using the former example as an inspiration it is possible to modify the labels from one graph's nodes and create a relation over a new extended set of nodes.

Example 2.1.4 (Relations are Graphs)**.** If we consider the following symmetric relation,

$$S = \{(A,B),(A,D),(B,D),(B,C),(F,G),(F,H),(H,E)\}$$

(the symmetric pairs are omitted for the sake of simplicity), we can also represent it as a disconnected undirected graph as follows:

This example illustrates that sometimes relations can be partitioned into disjoint **blocks** when there are no related pairs between disjoint subsets of X. In this case the blocks are $\{(A,B),(A,D),(B,D),(B,C)\}$ and $\{(F,G),(F,H),(H,E)\}$. When the relation is seen as a graph, these disjoint blocks are then called connected components.

Remark 2.1.1. Binary relations concern relations between just 2 elements. However, the same notion of relation is generalizable to n elements and, in which case, it is called a n-ary relation over the sets $X_1, X_2, \ldots, X_n$. In this general relation R, a set of n elements is said to be related if:

$$(x_1, x_2, \ldots, x_n) \in R \subseteq X_1 \times X_2 \times \cdots \times X_n$$

As in Example 2.1.4, where binary relations are identified as graphs, one can make the same identification between n-ary relations and hypergraphs, where edges are arbitrary finite sets of elements (nodes).

2.1.1 Fuzziness

From its definition, a binary relation, before anything else, is a set (which is a subset of a given product set X^2). In the usual notion of set, the membership of an element is a quite strict condition that is reflected in the (also usual) way of defining its characteristic map.

Definition 2.1.2 (Characteristic Map)**.** The **characteristic map** of a set A, is a map $\chi_A : A \to \{0,1\}$, defined as:

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A. \end{cases}$$

However, a more general framework can be applied when constructing a set by allowing the membership of an element to be fuzzy instead of crisp, as discussed in [18]. This way, the characteristic map could take values in $[0,1]$ instead of $\{0,1\}$, allowing a more expressive way of characterizing sets that can, nevertheless, encapsulate the usual crisp way of defining them. Therefore, in this new setting it's natural to consider fuzzy relations as relations, interpreted as fuzzy sets.

Definition 2.1.3 (Fuzzy Relation)**.** A (binary) fuzzy relation is a map $R : X_1 \times X_2 \to [0,1]$. Naturally, the domain of R can be any subset of $X_1 \times X_2$, just like in a crisp relation. With this definition it is clear how fuzzy relations are generalizations of the characteristic map of crisp relations. It is also implicit in this definition that, for a fuzzy relation R, $(a,b) \in R$ iff $R(a,b) > 0$.

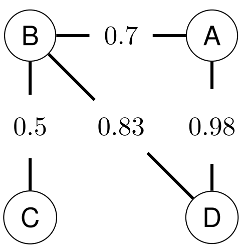
In the context of this work the most important relations are the ones where the universe sets are the same, $X_1 = X_2$. In this setting, a fuzzy relation is said to be:

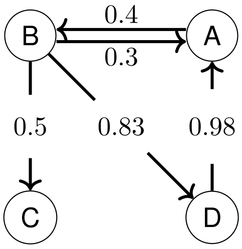
- **reflexive** if $R(x,x) = 1$ for all $x \in X$ and
- **symmetric** if $R(x,y) = R(y,x)$ for all $x, y \in X$.

The property of **transitivity**, however, doesn't have such a straightforward definition because it entails some arbitrariness on the way it can be defined, which is the central topic being explored further in this thesis.

Nevertheless we can, once again, represent a binary fuzzy relation over a single set as a graph which we call **fuzzy graph**.

Example 2.1.5. Any (symmetric or asymmetric) fuzzy relation can also be represented by a (undirected or directed) fuzzy graph, like the following ones.





The example on the left depicts a symmetric relation whereas the example on the right depicts an asymmetric relation.

Another way one can represent relations over a single set X is by a membership matrix.

Definition 2.1.4 (Membership Matrix of a Relation)**.** Let R be a fuzzy relation over a single set $X = \{x_1, \dots, x_n\}$ such that $|X| = n$. Then the **membership matrix** of R is

$$M_R = [m_{ij}]_{n\times n} \text{ where } m_{ij} = R(x_i, x_j)$$

Although this particular definition concerns fuzzy relations, this matrix can also represent a crisp relation by defining:

$$m_{ij} = \begin{cases} 1 & \text{if } (x_i, x_j) \in R \\ 0 & \text{if } (x_i, x_j) \notin R. \end{cases}$$

Example 2.1.6 (Membership Matrix)**.** Following on Example 2.1.5, one can create the respective membership matrices of those relations, by considering the order of nodes A, B, C, D in the respective matrix columns and rows.

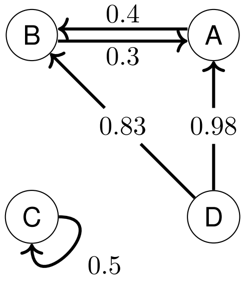
$$\begin{bmatrix} 0 & 0.7 & 0 & 0.98 \\ 0.7 & 0 & 0.5 & 0.83 \\ 0 & 0.5 & 0 & 0 \\ 0.98 & 0.83 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0.4 & 0 & 0 \\ 0.3 & 0 & 0.5 & 0.83 \\ 0 & 0 & 0 & 0 \\ 0.98 & 0 & 0 & 0 \end{bmatrix}$$

As previously mentioned in Chapter 1 Definition 1.3.10, in the context of graph theory this matrix is known as the **adjacency matrix** of a weighted graph and since any relation can be seen as a graph, we take these matrices as another way of representing graphs and binary relations. In this setting, the **reflexive** property of relations can be redefined as $(M_R)_{ii} = 1\ \forall i \in \{1, \ldots, n\}$ whereas the **symmetric** property can be stated as $(M_R)^T = M_R$.

A fuzzy graph is called a **proximity graph** if the underlying relation is reflexive and symmetric, and it's called a **similarity graph** if that relation is reflexive, symmetric and transitive.

Hence, a **similarity graph** entails an **equivalence relation** and its equivalence classes are the connected components of the graph.

Table 2.1: Summary of representations for a fuzzy relation

Fuzzy Relation	**Fuzzy Graph**	**Membership Matrix**
$X = \{A, B, C, D\}$ $R : X \times X \to [0, 1]$ $(A, B) \longmapsto 0.4$ $(B, A) \longmapsto 0.3$ $(C, C) \longmapsto 0.5$ $(D, A) \longmapsto 0.98$ $(D, B) \longmapsto 0.83$ $else \longmapsto 0$		$\begin{bmatrix} 0 & 0.4 & 0 & 0 \\ 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0.98 & 0.83 & 0 & 0 \end{bmatrix}$

2.1.2 Operations on Relations

So far we have discussed various ways to present and manipulate the abstract concept of relation. Now, we investigate ways to produce new relations from existing ones. For that purpose we present several ways to operate relations in some meaningful senses.

Definition 2.1.5 (Union of Crisp Relations)**.** Let R_1, R_2 be two crisp relations over a single set X. The union relation $R_1 \cup R_2$ is defined as

$$R_1 \cup R_2 = \{(x, y) : (x, y) \in R_1 \text{ or } (x, y) \in R_2\}.$$

Definition 2.1.6 (Intersection of Crisp Relations)**.** Let R_1, R_2 be two crisp relations over a single set X. The intersection relation $R_1 \cap R_2$ is defined as

$$R_1 \cap R_2 = \{(x, y) : (x, y) \in R_1 \text{ and } (x, y) \in R_2\}.$$

Definition 2.1.7 (Composition of Crisp Relations)**.** Let R_1, R_2 be two crisp relations over a single set X. The composition relation $R_1 \circ R_2$ is defined as

$$R_1 \circ R_2 = \{(x, y) : \exists z \in X \text{ with } (x, z) \in R_1 \text{ and } (z, y) \in R_2\}.$$

Moreover, by composing a relation with itself a finite number of times, one creates R^n, which can be recursively defined by $R^0 = id_X$ and $R^n = R^{n-1} \circ R^1$. In this context id_X denotes the identity crisp relation whose membership matrix is the identity matrix.

Lemma 2.1.1 (Neutral Element of Composition)**.** *Let R be a relation over the set X. The neutral element of $\circ : R \times R \to R$ is the relation $id_X = \{(x, x) : x \in X\}$.*

Proof. $R \circ id_X = \{(x, y) : \exists z \in X \text{ with } (x, z) \in R \text{ and } (z, y) \in id_X\}$. For (z, y) to be in id_X then $z = y$, so we obtain $\{(x, y) : (x, y) \in R \text{ and } (y, y) \in id_X\} = R$.
Also, $id_X \circ R = \{(x, y) : \exists z \in X \text{ with } (x, z) \in id_X \text{ and } (z, y) \in R\}$. For (x, z) to be in id_X then $z = x$, so we obtain $\{(x, y) : (x, x) \in id_X \text{ and } (x, y) \in R\} = R$. □

Given that, as noted before, any relation is a set and its characteristic function is enough to describe it, we can re-write the definition of self-composition of a relation by using the characteristic function ψ_R of the relation R and the Boolean logical connectives, $\vee$ (`or`) and $\wedge$ (`and`) to express the quantifier-free first order logic expression in the definition of composition.

$$R \circ R = \{(x, y) : \bigvee_{x \in X} \Big[\bigwedge(\psi_R(x, z), \psi_R(z, x))\Big]\}.$$

This perspective allows us to define composition of relations using the membership matrix. Let M_R be the membership matrix of a relation R defined over the set X with size n, then the membership matrix of $R \circ R$ is defined as follows:

$$M'_{R \circ R} = [m'_{ij}]_{n \times n} \text{ where } m'_{ij} = \bigvee_{1 \leq k \leq n} \left[\bigwedge (m_{ik}, m_{kj}) \right].$$

Thus, the **composition** of membership matrices is equivalent to the usual matrix product but the operations $\times$ and $+$ are replaced by $\wedge$ and $\vee$, respectively, and instead of matrix entries taking values from $\mathbb{R}$ or $\mathbb{C}$, they take values from $\mathbb{Z}_2 \simeq \{0, 1\}$. In this notation we can also re-write the definition of **union** and **intersection** of membership matrices like:

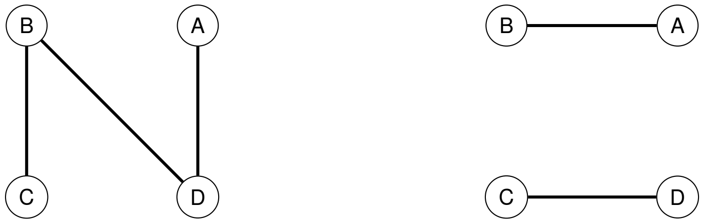
$$M'_{R \cup R} = [m'_{ij}]_{n \times n} \text{ where } m'_{ij} = \bigvee (m_{ij}, m_{ij})$$

$$M'_{R \cap R} = [m'_{ij}]_{n \times n} \text{ where } m'_{ij} = \bigwedge (m_{ij}, m_{ij}).$$

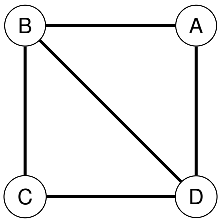
Other operations like the **converse**, **complement** and **restriction** are also easily defined over relations but are not needed in this context.

By analysing the definition of composition, one can notice that the relation R^n corresponds to the reachability of R after n steps. That is, in the relation R^n, an element is related to another if that other can be reached by using n related pairs from the original relation R.

Example 2.1.7 (Composition of Crisp Relations)**.** Omitting the symmetric pairs, let's consider the relation on the left, $R = \{(A, D), (B, D), (B, C)\}$. Then, we can produce the relation on the right which is, $R \circ R = R^2 = \{(A, B), (C, D)\}$.



And now, we can unite these two relations and create the relation that expresses the reachability by at most two arcs of the relation R, which is $R \cup R^2$. It is now clear that the only edge missing from this new relation is (A, C) because in R it takes 3 steps to go from A to C and vice-versa.



Using the membership matrix notation, the previous composition yields:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

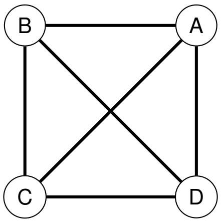
And the following union produces

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cup \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

2.1.3 Transitivity in Graphs

Until now we have seen that relations and graphs can represent each other and thus we've been using the term relation several times to mean graph. From now on we discard the term relation, unless needed, and focus on the term graph that can be more easily used as a proxy for the applications coming ahead that use the also equivalent term of network. Beyond **reflexivity** and **symmetry** there is another important property that a graph can attain which is **transitivity**. Transitive connections in networks reflect indirect associations between variables (or nodes) that can influence the behaviour of the network in various dynamical processes. These indirect associations need to be accounted for when studying these dynamics on a network since in many cases they are more important pathways than direct routes. As seen before, in a crisp relation, the definition of a transitivity emerges quite naturally as $(a, c) \in R$ whenever $(a, b) \in R$ and $(b, c) \in R$, which in plain terms can be expressed as "a friend of my friend is also my friend".

Example 2.1.8 (Symmetric Transitive Unweighted Graph)**.** Consider, for example, a transitive unweighted graph $G = (V, E)$ where $V = \{A, B, C, D\}$ and $E = \{(A, B), (B, A), (A, C), (C, A), (A, D), (D, A), (B, C), (C, B), (B, D), (D, B), (C, D), (D, C)\}$ depicted as follows:



In this example, the set of nodes has 4 elements and the graph is the complete graph, K_4, which is not a coincidence. In fact, this is *the* undirected transitive graph with 4 nodes.

Until now, most of the graphs given as examples were *connected*. This is the same as saying that the relations so far had only one block. In many relations this is not the case and there are several disjoint blocks but even in those scenarios, it is possible to interpret the relation as a graph which is a union of connected components. Bearing this in mind, we can state what it means for a graph to be transitive in terms of which edges it must have.

Lemma 2.1.2. *An (undirected) transitive graph is equivalent to a union of complete (undirected) graphs.*

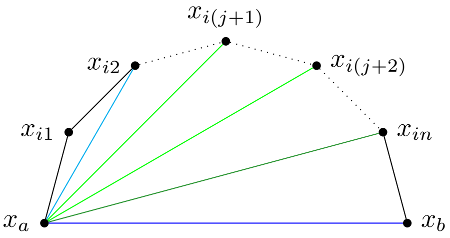
Proof. $(\Rightarrow)$ Let G be a transitive graph. Let k be the number of disjoint connected components of G, with $G_i = (V_i, E_i)$ being the i-th connected component. Since the graph G is transitive, in particular, each subgraph G_i is also transitive. Now, assume that G_i is not complete. Then there are nodes $x_a, x_b \in V_i$ such that $(x_a, x_b) \notin G_i$. But since G_i is connected, there is a path from x_a to x_b in G_i. Let that path be $(x_a, x_{i1}), \dots, (x_{in}, x_b)$ with length $n + 1$. Now we prove that:

$$\forall j \in \{1, \dots, n\} : (x_a, x_{i(j+1)}) \in G_i.$$

In this context the number j represents the number of intermediate nodes in the path. We proceed by induction on j:

$[j = 1]$ Since $(x_a, x_{i1}), (x_{i1}, x_{i2}) \in G_i$ then $(x_a, x_{i2}) \in G_i$ because G_i is transitive.

$[j \implies j + 1]$ If $(x_a, x_{i(j+1)}) \in G_i$ then $(x_a, x_{i(j+2)}) \in G_i$ because $(x_{i(j+1)}, x_{i(j+2)}) \in G_i$ and G_i is transitive.



Therefore, given that the condition holds, we have $(x_a, x_{i(n+1)}) \in G_i$. But, in this case, $x_{i(n+1)}$ is x_b, and so we conclude that $(x_a, x_b) \in G_i$. In summary, we assumed that an arbitrary edge was missing from G_i but we were forced to conclude that it is in fact not missing, so we conclude that G_i must be complete.

$(\Leftarrow)$ Let $G = (V, E)$ be a graph such that $\{G_i = (V_i, E_i)\}_{i \in I}$ is the finite family of connected components of G and G_i is a complete graph for every $i \in I$. Given that G_i is complete, it is also transitive because for every pair of edges $(x_{i1}, x_{i2}), (x_{i2}, x_{i3}) \in E$, we also have $(x_{i1}, x_{i3}) \in E$ given that G_i has all possible edges over V_i and, in particular, (x_{i1}, x_{i3}). Therefore every G_i is transitive and we conclude that G must be transitive since there are no edges connecting nodes from different components. □

Lemma 2.1.3 (Equivalent Condition for Transitivity). *A graph is transitive if and only if $G \circ G \subseteq G$.*

Proof. $(\Rightarrow)$ Let $G = (V, E)$ be a transitive graph. Consider an arbitrary edge $(a, b) \in G \circ G$, then it must exist $c \in V$ such that $(a, c), (c, b) \in G$. Then, given that $(a, c), (c, b) \in G$ and G is transitive we conclude that $(a, b) \in G$. Thus, $G \circ G \subseteq G$.

$(\Leftarrow)$ Assume that $G \circ G \subseteq G$. Now, consider any pair $(a, c), (c, b) \in G$, then by definition of composition, $(a, b) \in G \circ G$. But given that we assumed $G \circ G \subseteq G$, we can conclude that $(a, b) \in G$. Thus G is transitive. □

2.2 Transitive Closures

The closure of a graph (relation) associated with a given property is always the smallest graph that contains the original graph and also obeys to that. In particular, we can consider the reflexive closure, the symmetric closure and also the transitive closure which is the most relevant closure in the scope of this work. In the simplest case of unweighted graphs the concept of transitivity is unambiguous and therefore the definition of transitive closure comes naturally.

Definition 2.2.1 (Transitive Closure). The transitive closure of a graph G is the smallest transitive graph that contains G and is denoted as G^∞.

As was noted before, the n-th self-composition of an unweighted graph, G^n, defines the exact reachability from every node in G by paths of length n. Therefore, in particular, G^2 contains exactly the edges that complete non-transitive pairs in G. Intuitively this fact can be used to compute the transitive closure of G because, if $G \cup G^2$ is already transitive, the closure is found. Otherwise, we can compute $(G \cup G^2)^2$ and then check if $(G \cup G^2) \cup (G \cup G^2)^2$ is transitive. This process can, then, be repeated until we reach a graph that doesn't add any more edges because it is already transitive, making use of Lemma 2.1.3.

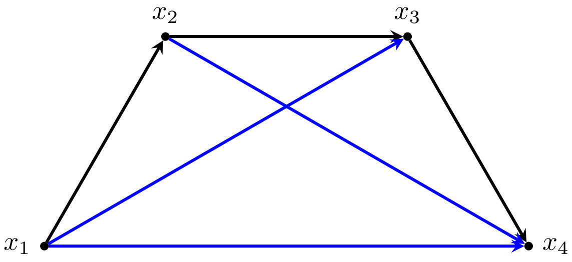
Algorithm 2.1 Transitive Closure of an Unweighted Graph

Input: Adjacency Matrix R of an Unweighted Graph G

```
1. $R' \leftarrow R \cup (R \circ R)$
2. IF $R' \neq R$:
3.     $R \leftarrow R'$
4.     GO TO Step 1
5. ELSE:
6.     $R^{\infty} \leftarrow R'$
7.     RETURN $R^{\infty}$
```

Output: Transitive Closure R^{∞}

Remark 2.2.1. In the case of directed graphs the previous statement of Lemma 2.1.2 is not necessarily true, since transitivity is inherently directional and not bidirectional.



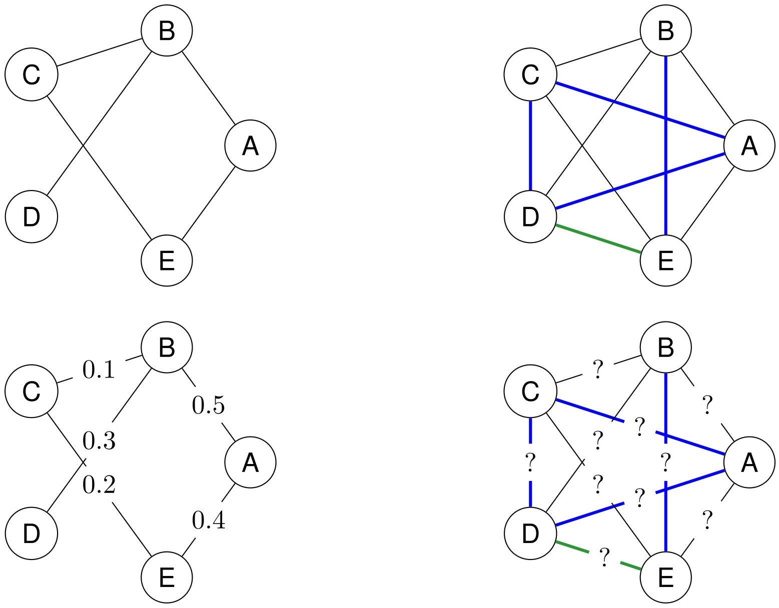
The above figure, inspired by the one in Lemma 2.1.2, illustrates this fact. Consider the graph G with the set of edges $E = \{(x_1, x_2), (x_2, x_3), (x_3, x_4)\}$. Its transitive closure is $G^{\infty} = \{(x_1, x_2), (x_2, x_3), (x_3, x_4),$ $(x_1, x_3), (x_1, x_4),$ $(x_2, x_4)\}$ and as can be visualized in this figure, this graph is not complete because, being a directed graph, it lacks an edge for every pair of nodes.

Fuzzy graphs don't have such a straightforward definition for their transitivity property. First of all, when dealing with unweighted graphs if $(x, y_1), (y_1, z) \in E$ then it does not matter that $(x, y_2), (y_2, z) \in E$ to conclude that $(x, z) \in E$ if the graph is transitive. However, given that in fuzzy graphs the edges are weighted, we need to impose certain criteria to obtain a coherent definition of transitivity. It is clear that the fuzzy transitive closure of a fuzzy graph should have the same edges as the transitive closure of the correspondent unweighted graph; however, it is not clear what weights should be added to the possible new edges and also how to update current weights according to the criteria for transitivity that we define.

Let's take an example that illustrates this challenge.

Example 2.2.1 (Crisp vs Fuzzy: Transitive Closure)**.** In this example one can observe the transitive closures (*right*) of both an unweighted (*top*) and a weighted (*bottom*) graph.

In Example 2.2.1, the edges (B, E), (C, A), (A, D) and (D, C) must be in the transitive closure because, respectively, $\{(B, A), (A, E)\} \subset E$, $\{(C, B), (B, A)\} \subset E$, $\{(A, B), (B, D)\} \subset E$ and



$\{(D, B), (B, C)\} \subset E$. However, only adding these edges does not create a transitive graph because now $(D, C), (C, E) \in E$ but $(D, E) \notin E$, so we must also add (D, E).

As can be observed in this example, when computing the transitive closure of the weighted graph, it is not obvious which values to insert in the edges. This is because when computing transitivity in crisp relations one only cares about connectivity, and if two nodes are connected by some path then, by transitivity, there must be an edge connecting them, no matter if there are other different paths between them. However, in fuzzy relations (also known as fuzzy graphs) where each connection has a strength associated, not all paths have the same aggregated strength. This fact yields two decisions that need to be made in this context: how to aggregate the strengths of edges in a path and how to combine the aggregated strengths of different paths. At this point, taking into account the algorithm for computing the transitive closure of crisp relations as well as the definition of union and composition of membership matrices, one can generalize these constructions by using operations that generalize logical values from $\{0, 1\}$ to $[0, 1]$. This way, we can take into account the strengths associated with each edge in fuzzy graphs. Operations of this kind are known in the literature as Triangular Norms and Triangular Conorms and are introduced next.

2.2.1 Triangular Norms and Conorms

The idea of Triangular Norm was first presented by Karl Menger in 1942 in [24] when he introduced the concept of Statistical Metric. This concept generalized the notion of metric so that every two elements in a statistical metric space had associated a probability function instead of a distance value. In that context Triangular Norms were used to generalize the Triangular Inequality for probability functions instead of distances. At that time, the definition of Triangular Norm was much looser than today (not requiring associativity, for instance) and also encompassed what is now known as Triangular Conorm. The current definition was presented in [34], which followed the work in [32], and also renamed the field from statistical metric spaces to probabilistic metric spaces.

Later, in 1959, Lofti Zadeh suggested the use of T-Norms and T-Conorms in the realm of set theory [43]. In this context T-Norms are interpreted as fuzzy set intersections and T-Conorms as fuzzy set unions and allowed for the development of the field of fuzzy set theory which has been found to have a lot of applicabilities, for example in the field of control theory [17]. Beyond these areas, the use of Triangular Norms and Conorms has been mostly associated with Fuzzy Logic in the sense that these operations can also generalize the logical connectives of conjunction and disjunction, respectively [44] [18], which is a useful interpretation for our purposes.

The application of T-Norms and T-Conorms in Graph Theory and Network Science, namely in the computation of Transitive Closures, has been mostly developed in [36] and [35], which are the main references for this work.

Definition 2.2.2 (T-Norm)**.** A Triangular Norm, **T-Norm**, is a binary operation, $\wedge : [0,1] \times [0,1] \rightarrow [0,1]$, with the following properties:

(*commutativity*) $\wedge(x,y) = \wedge(y,x)$
(*associativity*) $\wedge(x,\wedge(y,z)) = \wedge(\wedge(x,y),z)$
(*monotonicity*) $x \leq y \implies \wedge(x,z) \leq \wedge(y,z)\ \forall z \in [0,1]$
(*neutral element*) $\wedge(x,1) = x$

Two prominent examples of T-Norms, that will be shown later in more detail, are $\wedge(x,y) = xy$ and $\wedge(x,y) = \min(x,y)$.

Definition 2.2.3 (T-Conorm)**.** A Triangular Conorm, **T-Conorm**, is a binary operation, $\vee : [0,1] \times [0,1] \rightarrow [0,1]$, with the following properties:

(*commutativity*) $\vee(x,y) = \vee(y,x)$
(*associativity*) $\vee(x,\vee(y,z)) = \vee(\vee(x,y),z)$
(*monotonicity*) $x \leq y \implies \vee(x,z) \leq \vee(y,z)\ \forall z \in [0,1]$
(*neutral element*) $\vee(x,0) = x$

As examples of T-Conorms we have $\vee(x,y) = x + y - xy$ and $\vee(x,y) = \max(x,y)$.

In purely algebraic terms, both $([0,1],\wedge)$ and $([0,1],\vee)$ are commutative monoids with the monotonicity property. Furthermore, interpreting T-Norms as (fuzzy) logical operators, their associative and commutative properties come naturally and the neutral element is the distinguishing feature, in the sense that 1 is the neutral element for T-Norms, that generalize conjunction, and 0 is the neutral element for

T-Conorms, that generalize disjunction, as is stated below. On the other hand, the monotonicity property only makes sense in the fuzzy realm because it ensures that the truth value outputted by these operations is non-decreasing in both variables.

With these generalizations at hand we can now state the definitions for composition, intersection and union of adjacency matrices of fuzzy graphs.

Definition 2.2.4 (Operations on Fuzzy Graphs)**.** Let $P_1, P_2 \in \mathcal{M}_n([0,1])$ be adjacency matrices of fuzzy graphs G_1 and G_2. Then, we define **intersection**, **union** and **composition** of fuzzy adjacency matrices as:

$$\begin{aligned} P_1 \cap P_2 &= [p'_{ij}]_{n\times n} \text{ where } p'_{ij} = \bigwedge(p_{1ij}, p_{2ij}) \\ P_1 \cup P_2 &= [p'_{ij}]_{n\times n} \text{ where } p'_{ij} = \bigvee(p_{1ij}, p_{2ij}) \\ P_1 \circ P_2 &= [p'_{ij}]_{n\times n} \text{ where } p'_{ij} = \bigvee_{1\leq k\leq n} \left[\bigwedge(p_{1ik}, p_{2kj})\right], \end{aligned}$$

where $\wedge$ is a T-Norm and $\vee$ is a T-Conorm.

Moreover, for an arbitrary proximity graph matrix P, we define P^n recursively as $P^0 = id_P$ and $P^n = P^{n-1} \circ P^1$, where id_P denotes identity proximity matrix, I_n.

It is noteworthy to mention that, despite T-Conorms (and T-Norms) taking only two input variables, the definition of composition of fuzzy graphs implies that we can input an arbitrary number of variables, n. This is due to the fact that these operations are associative and, thus, can be applied consecutively any finite number of times without the order affecting the final result.

Taking into account the Lemma 2.1.3, one can redefine transitivity for fuzzy graphs using the notions of T-Norm and T-Conorm. Given that, for an unweighted graph G, we have G transitive $\Leftrightarrow G \circ G \subseteq G$, in the notation of adjacency matrices this condition becomes G transitive $\Leftrightarrow (P \circ P)_{ij} \leq P_{ij}$ and thus, we can extend this definition to fuzzy graphs in terms of their weighted adjacency matrices:

Definition 2.2.5. A fuzzy graph $G = (V, E)$ with adjacency matrix P with dimensions $|V| \times |V|$ is said to be **transitive** whenever

$$p_{ij} \geq \bigvee_{1\leq k\leq n} \left[\bigwedge(p_{ik}, p_{kj})\right] \forall i, j \in \{1, \ldots, |V|\}.$$

2.2.2 Fuzzy Transitive Closure Algorithm

Although we can use algorithm 2.1 to compute the transitive closure of a fuzzy graph by using the union and composition of fuzzy graphs, there is an alternative algorithm based on the consecutive power compositions of the graph given by the respective T-Norm and T-Conorm. Remembering that in the composition of membership matrices of relations the matrix R^n corresponds to the reachability of the relation R using n steps, then, it is easy to extrapolate that for proximity (or distance) graphs, their $n^{th}-$composition (i.e., the composition of their adjacency matrices) yields the reachability of the original graph after n steps but with the weight of each edge being computed through the composition of the T-Norm/T-Conorm or TD-Norm/TD-Conorm pairs. Having this in mind, we can notice that the transitive closure of a graph is precisely the graph that, for every pair of nodes, assigns the aggregation (using T-Conorm) of the path-weights (computed using T-Norm and T-Conorm) of every number of steps/edges between those same nodes. Therefore we can compute this closure using the following algorithm.

Algorithm 2.2 Transitive Closure of a Proximity Graph

Input: Adjacency Matrix of a Proximity Graph, R

```
1. $R_1 \leftarrow R$
2. $R_2 \leftarrow R \circ R$
3. $\kappa \leftarrow 1$
4. WHILE $R_1 \neq (R_1 \cup R_2)$:
5.     $R_1 \leftarrow (R_1 \cup R_2)$
6.     $R_2 \leftarrow (R_2 \circ R)$
7.     $\kappa \leftarrow \kappa + 1$
8. $R^\infty \leftarrow R_1$
9. RETURN $R^\infty$, $\kappa$
```

Output: Transitive Closure R^∞, Last Power κ

Making use of the intuition given before and the formulation of the algorithm, we can now enunciate a precise definition for the transitive closure of fuzzy graphs.

Definition 2.2.6 (Fuzzy Transitive Closure)**.** Given a proximity graph G with weighted adjacency matrix R, the Transitive Closure R^∞ of G is defined as

$$R^\infty = \bigcup_{n=1}^{\kappa} R^n$$

In general, $\kappa \to +\infty$, but under some conditions explained ahead this value might become finite.

Bearing in mind this definition of transitive closure, it is clear that it depends on the transitivity criteria we wish to impose on the given graph since we need to choose in advance which T-Norm and T-Conorm pair to use in the computation of the closure. However, the most standard definition of fuzzy transitive closure uses the pair $\langle \max, \min \rangle$ and a transitive graph in that context is said to be $\max$-$\min$-transitive [36].

2.2.3 Examples of T-Norms and T-Conorms

Before presenting some of the most relevant examples of T-Norms [17], it is pertinent to establish the boundaries which are common amongst every T-Norm. The boundary values of an arbitrary T-Norm can be obtained by manipulating the axiomatic properties of these operations in the following way.

[——] By the neutral element property, $\wedge(x,1) = x \, \forall x \in [0,1]$, and by commutativity, $\wedge(1,y) = y \, \forall y \in [0,1]$.

[——] Moreover, $\wedge(x,0) = 0 \, \forall x \in [0,1]$ because, by monotonicity, for an arbitrary $x \in [0,1]$ we have $x \leq 1 \Rightarrow \wedge(x,z) \leq \wedge(1,z) = z \, \forall z \in [0,1]$, thus $\wedge(x,z) \leq z$ and, in particular, $\wedge(x,0) \leq 0$, which implies that $\wedge(x,0) = 0$. Since x is arbitrary we get $\wedge(x,0) = 0 \, \forall x \in [0,1]$ and by commutativity $\wedge(0,y) = 0 \, \forall y \in [0,1]$.

[- - -] Also, as before, for an arbitrary $x \in [0,1]$ we have $x \leq 1 \Rightarrow \wedge(x,z) \leq \wedge(1,z) = z \forall z \in [0,1]$, thus $\wedge(x,z) \leq z$ and by commutativity, $\wedge(x,z) \leq x \, \forall x,z \in [0,1]$ and we can conclude that $\wedge(x,x) \leq x \, \forall x \in [0,1]$

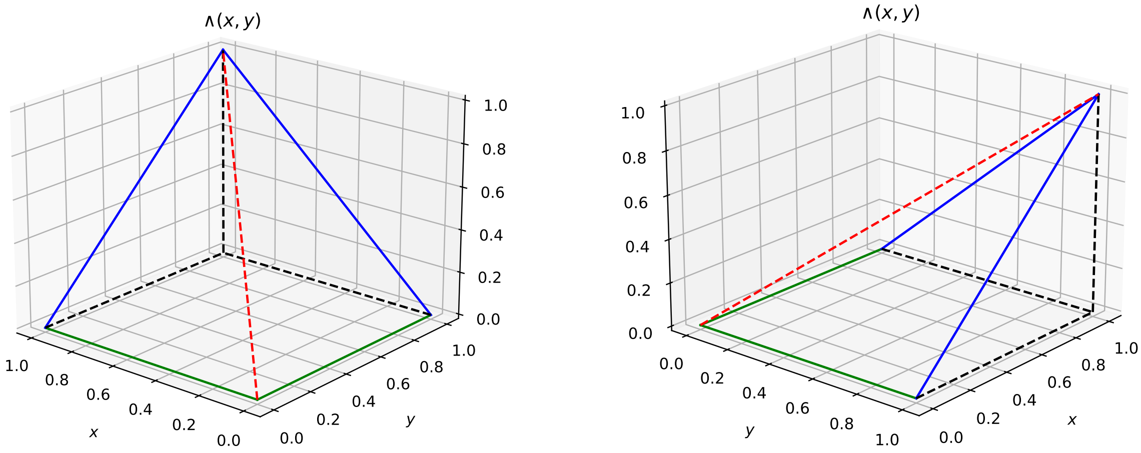


Figure 2.1: Boundary of T-Norms

Next, we present some examples of the most well-known T-Norms in the literature [17]. Many of these examples stand out by their simplicity like the Drastic T-Norm and the Minimum T-Norm, which later will be known as the bounds for every T-Norm. Beyond these, we present the standard product, which has intuitive basis in the composition of probabilistic events and the Hamacher product that mimics the Jaccard Similary of sets and also the probability of $P(X$ and $Y|X$ or $Y)$ where $P(X) = x$ and $P(Y) = y$. The other T-Norm presented is the Łukasiewicz T-Norm which is named after the Polish logician and philosopher Jan Łukasiewicz [37], who created this T-Norm while developing the field of Many-Valued Logics.

Drastic $\wedge_{\text{Drastic}}(x,y) = \begin{cases} x & \text{if } y=1 \\ y & \text{if } x=1 \\ 0 & \text{otherwise} \end{cases}$

Product $\wedge_{\text{Prod}}(x,y) = xy$

Hamacher Product $\wedge_{\text{HamProd}}(x,y) = \frac{xy}{x+y-xy}$

Łukasiewicz $\wedge_{\text{Łuka}}(x,y) = \max(0, x+y-1)$

Minimum $\wedge_{\text{Min}}(x,y) = \min(x,y)$

Before moving on to T-Conorms it's important to mention that a way to produce them is by starting from a T-Norm and construct its dual T-Conorm. In addition, that process can be done in the other way around by starting with a T-Conorm and obtaining its dual T-Norm. The concept of duality in this setting is canonically represented by the fuzzy negation $1-x$, however there are other negation operators that can be considered as well as the corresponding dual pairs of T-Norms and Conorms.

Proposition 2.2.1 (Dual Pairs). *If* $\wedge_1 : [0,1] \times [0,1] \to [0,1]$ *is a T-Norm then*

$$\vee_1(x,y) = 1 - \wedge_1(1-x, 1-y)$$

is a T-Conorm. Also, if $\vee_2 : [0,1] \times [0,1] \to [0,1]$ *is a T-Conorm then*

$$\wedge_2(x,y) = 1 - \vee_2(1-x, 1-y)$$

is a T-Norm. In both these cases we say that $\langle \vee_1, \wedge_1 \rangle$ *and* $\langle \vee_2, \wedge_2 \rangle$ *are* ***dual*** *pairs of T-Conorms and T-Norms.*

Proof. Given that $\wedge_1$ is a T-Norm $(*)$, then $\vee_1$ is a T-Conorm since,

(*commutativity*) $\vee_1(x,y) = 1 - \wedge_1(1-x, 1-y) \overset{(*)}{=} 1 - \wedge_1(1-y, 1-x) = \vee_1(y,x)$

(*associativity*) $\vee_1(x, \vee_1(y,z)) = 1 - \wedge_1(1-x, 1-(1-\wedge_1(1-y, 1-z))) = 1 - \wedge_1(1-x, \wedge_1(1-y, 1-z)) \overset{(*)}{=} 1 - \wedge_1(\wedge_1(1-x, 1-y), 1-z) = 1 - \wedge_1(1-(1-\wedge_1(1-x,1-y)), 1-z) = \vee_1(\vee_1(x,y), z)$

(*monotonicity*) $x \leq y \Rightarrow 1-y \leq 1-x \overset{(*)}{\Rightarrow} \wedge_1(1-y, z) \leq \wedge_1(1-x, z) \Rightarrow -\wedge_1(1-x,z) \leq -\wedge_1(1-y,z) \Rightarrow 1 - \wedge_1(1-x,z) \leq 1 - \wedge_1(1-y,z) \Rightarrow \vee_1(x,z) \leq \vee_1(y,z)$

(*neutral element*) $\vee_1(x,0) = 1 - \wedge_1(1-x, 1-0) = 1 - \wedge_1(1-x, 1) \overset{(*)}{=} 1 - 1 = 0$

In an analogous way, we obtain the *commutativity*, *associativity* and *monotonicity* properties of $\wedge_2$ given that $\vee_2$ is a T-Norm $(*)$. The *neutral element* property can also be obtained by:

(*neutral element*) $\wedge_2(x,1) = 1 - \vee_2(1-x, 1-1) = 1 - \vee_2(1-x, 0) \overset{(*)}{=} 1 - 0 = 1$

Thus, $\vee_2$ is a T-Norm. □

Analogously to T-Norms, the boundary values of T-Conorms can also be obtained as follows.

[——] By the neutral element property, $\vee(x,0) = x \; \forall x \in [0,1]$, and by commutativity, $\vee(0,y) = y \; \forall y \in [0,1]$.

[——] Moreover, $\vee(x,1) = 1 \; \forall x \in [0,1]$ because, by monotonicity, for an arbitrary $x \in [0,1]$ we have $0 \leq x \Rightarrow \vee(0,z) \leq \vee(x,z) \; \forall z \in [0,1]$, thus $z \leq \vee(x,z)$ and, in particular, $1 \leq \vee(x,1)$, which implies that $\vee(x,1) = 1$. Since x is arbitrary we get $\vee(x,1) = 1 \; \forall x \in [0,1]$ and by commutativity $\wedge(1,y) = 1 \; \forall y \in [0,1]$.

[- - -] Again, as before, for an arbitrary $x \in [0,1]$ we have $0 \leq x \Rightarrow \vee(0,z) \leq \vee(x,z) \; \forall z \in [0,1]$, thus $z \leq \vee(x,z)$ and by commutativity, $x \leq \vee(x,z) \; \forall x,z \in [0,1]$

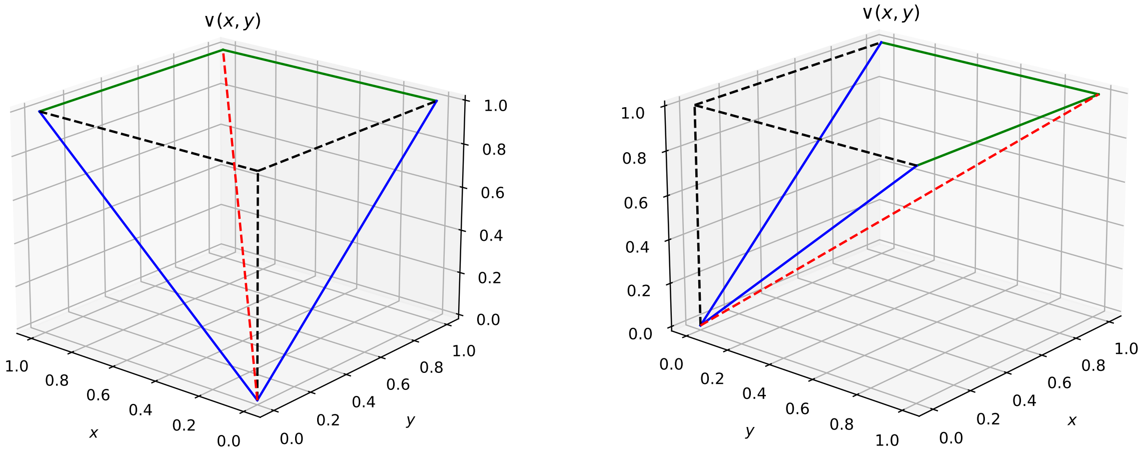


Figure 2.2: Boundary of T-Norms

The following T-Conorms examples are the dual T-Conorms for the respective basic T-Norms presented before. In the literature covering these topics it is often common to call T-Conorms, *Sums*, and T-Norms, *Products*, in analogy to the usual operations over $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ and $\mathbb{C}$. Therefore, some of the following T-Conorms have the same first name as their dual but adopt the suffix *Sum* instead.

Drastic Sum $\vee_{\text{DrasticSum}}(x,y) = \begin{cases} x & \text{if } x = 0 \\ y & \text{if } y = 0 \\ 1 & \text{otherwise} \end{cases}$

Probabilistic Sum $\vee_{\text{ProbSum}}(x,y) = x + y - xy$

Hamacher Sum $\vee_{\text{HamSum}}(x,y) = \frac{x+y-2xy}{1-xy}$

Łukasiewicz $\vee_{\text{Łuka}}(x,y) = \min(x+y,1)$

Maximum $\vee_{\text{Max}}(x,y) = \max(x,y)$

2.2.4 Properties of T-Norms and T-Conorms

Because these two kinds of operations are dual, we choose to present most of the properties concerning only T-Norms since the corresponding properties for T-Conorms can be formulated analogously. Moreover, in the following chapters it will become clear that our main focus will be on T-Norms since we will fix a particular T-Conorm and study the transitive closures associated with it.

Proposition 2.2.2. *Every T-Norm, $\wedge(x, y)$ is pointwisely bounded by the Drastic and the Minimum T-Norms,*

$$\forall(x, y) \in [0, 1]^2 : \wedge_{\textit{Drastic}}(x, y) \leq \wedge(x, y) \leq \wedge_{\textit{Min}}(x, y).$$

Proof. Since we previously checked that every T-Norm coincides on $[0, 1]^2 \backslash]0, 1[^2$, we need to prove that

$$\forall(x, y) \in]0, 1[^2 \colon 0 \leq \wedge(x, y) \leq \min(x, y).$$

We have already checked that $0 \leq \wedge(x, y)$. Moreover, by monotonicity,

$$x \leq 1 \Rightarrow \wedge(x, y) \leq \wedge(1, y) \Rightarrow \wedge(x, y) \leq y$$
$$y \leq 1 \Rightarrow \wedge(x, y) \leq \wedge(x, 1) \Rightarrow \wedge(x, y) \leq x$$

Thus, we obtain that $\wedge(x, y) \leq \min(x, y)$. □

Therefore, the previous proposition guarantees that the Drastic T-Norm is the smallest pointwise T-Norm whilst the Minimum T-Norm is the largest.

Proposition 2.2.3. *The Product T-Norm $\wedge_{\textit{Prod}}(x, y)$ is pointwisely smaller than the Hamacher Product T-Norm $\wedge_{\textit{Ham}}(x, y)$*

$$\forall(x, y) \in [0, 1]^2 : \wedge_{\textit{Prod}}(x, y) \leq \wedge_{\textit{Ham}}(x, y)$$

Proof. Once again, we just need to prove that

$$\forall(x, y) \in]0, 1[^2 \colon xy \leq \frac{xy}{x + y - xy}$$

First, let us evaluate when $x + y + xy \geq 1$:

$$x + y - xy \geq 1 \Rightarrow y - xy \geq 1 - x \Rightarrow y(1 - x) \geq 1 - x \Rightarrow y \geq \frac{1 - x}{1 - x} = 1 \Rightarrow y \geq 1$$

And analogously,

$$x + y - xy \geq 1 \Rightarrow x - xy \geq 1 - y \Rightarrow x(1 - y) \geq 1 - y \Rightarrow x \geq \frac{1 - y}{1 - y} = 1 \Rightarrow x \geq 1$$

But $(x,y) \in]0,1[^2$ so $x+y-xy<1$ for every $(x,y)\in]0,1[^2$. This way, $\frac{1}{x+y-xy}>1$ and we can conclude

$$xy < xy \cdot \frac{1}{x+y-xy} \ \forall (x,y) \in]0,1[^2.$$

□

Proposition 2.2.4. *In the realm of T-Norms, the following facts hold.*

(a) The only T-Norm $\wedge$ satisfying $\wedge(x,x) = x \ \forall x \in [0,1]$ is $\min(x,y)$.
(b) The only T-Norm $\wedge$ satisfying $\wedge(x,y) = 0 \ \forall x \in [0,1[$ is $\wedge_{\textit{Drastic}}(x,y)$.

Proof. (a) If $\wedge(x,x) = x$ for every $x \in [0,1]$ then for all $y \le x$ we have, by monotonicity and Proposition 2.2.2, $y = \wedge(y,y) \le \wedge(x,y) \le \min(x,y) = y$ so $y \le \wedge(x,y) \le y$, which means that $\wedge(x,y)$ must be $y = \min(x,y)$. Since $\wedge$ is commutative the same holds if we assumed $x \le y$. (b) Similarly to (a), assume that $\wedge(x,x) = 0$ for every $x \in [0,1[$ then for every $y \le x$ we have, by monotonicity and Proposition 2.2.2, $0 \le \wedge(x,y) \le \wedge(x,x) = 0$. Thus, $0 \le \wedge(x,y) \le 0$ and, again by commutativity, we must have $\wedge(x,y) = 0 \ \forall (x,y) \in [0,1[^2$. □

Definition 2.2.7 (Continuous T-Norm)**.** A T-Norm is said to be **continuous** if it is a continuous function in both variables. That is, if for all convergent sequences $(x_n)_{n\in\mathbb{N}}, (y_n)_{n\in\mathbb{N}} \in [0,1]^{\mathbb{N}}$ it is true that:

$$\bigwedge \left(\lim_{n\to+\infty} x_n, \lim_{n\to+\infty} y_n \right) = \lim_{n\to+\infty} \bigwedge (x_n, y_n)$$

Definition 2.2.8 (Strictly Monotone T-Norm)**.** A T-Norm is said to be **strictly monotone** if the monotonicity property holds strictly, i.e., $x < y \implies \wedge(x,z) < \wedge(y,z) \ \forall z \in]0,1]$.

Definition 2.2.9 (Strict T-Norm)**.** A T-Norm is said to be **strict** if it is **continuous** and **strictly monotone**.

Example 2.2.2. The Drastic T-Norm is not **strictly monotone** nor **continuous**.

Definition 2.2.10 (Archimedean T-Norm)**.** An **Archimedean T-Norm** is a T-Norm, $\wedge$, such that for every pair $(x,y) \in]0,1[^2$ there exists a number $n \in \mathbb{N}$ for which

$$\bigwedge_n x = \underbrace{x \wedge \cdots \wedge x}_{\text{n times}} < y$$

Definition 2.2.11. A T-Norm is said to hold the **limit property** if $\forall x \in]0,1[$: $\lim_{n\to+\infty} \bigwedge_n x = \underbrace{x \wedge \cdots \wedge x}_{\text{n times}} = 0$

Theorem 2.2.5. *For any T-Norm, $\wedge$, the following statements are equivalent:*

(i) $\wedge$ *is **archimedean***
(ii) $\wedge$ *holds the **limit property***

(iii) $\wedge$ *only has trivial idempotent elements* ($a \in [0,1]$*:* $\wedge(a,a) = a$) *and, whenever for some* $x_0 \in]0,1[$: $\lim\limits_{x \to x_0} \wedge(x,x) = x_0$ *then there exists* $y_0 \in]x_0,1[$ *such that* $\wedge(y_0,y_0) = x_0$

Proof.

$(i) \Rightarrow (iii)$ Assume $\wedge$ is archimedean. Then, if it exists $a \in]0,1[$ such that $\wedge(a,a) = a$, then $\underset{n}{\wedge} a = a$ for all $n \in \mathbb{N}$. But since $\wedge$ is archimedean that cannot happen, so $a \notin]0,1[$ and thus must be trivial. Also, if the second condition of (i) doesn't hold, i.e. there is no y_0 in those conditions, then, for all $y \in]x_0,1[$ we have $\wedge(y_0,y_0) > x_0$ and so by induction we also have $\underset{n}{\wedge} y_0 > x_0$ for all $n \in \mathbb{N}$ and $y \in]x_0,1[$, which contradicts the fact that $\wedge$ is archimedean. Therefore $\wedge$ being archimedean implies that the second condition holds.

$(iii) \Rightarrow (ii)$ Assume that $\wedge$ satisfies (iii) and put $\lim\limits_{n \to +\infty} \bigwedge\limits_{n} x = x_0$. Then, by the monotonicity of $\wedge$, we must also have $\lim\limits_{y \to x_0} \bigwedge\limits_{n} y = x_0$. If $x_0 > 0$, then there is some $y_0 \in]x_0,1[$ with $\wedge(y_0,y_0) = x_0$ and also $\bigwedge\limits_{n} x < y_0$ for some $n_y \in \mathbb{N}$ which implies that $\bigwedge\limits_{2n} x = x_0$ for all $n > n_y$, leading to $\wedge(x_0,x_0) = \bigwedge\limits_{4n} x = x_0$ which is a contradiction. Hence, we conclude that x_0 must be 0. Since $x \in]0,1[$ was chosen arbitrarily, we conclude that $\wedge$ holds the limit property.

$(ii) \Rightarrow (i)$ Assume that $\wedge$ holds the limit property, then choose $x, y \in]0,1[$. Since $\lim\limits_{n \to +\infty} \bigwedge\limits_{n} x = 0$, then there exists $n \in \mathbb{N}$ such that $\bigwedge\limits_{n} x < y$, which is to say that $\wedge$ is archimedean.

□

Moreover, in both [18] and [3] there is another theorem that summarizes the archimedean property in the context of continuous T-Norms. Since most of the relevant T-Norms we will be using further are continuous this theorem reveals to be specially relevant.

Theorem 2.2.6. *A continuous T-Norm is* ***archimedean*** *if and only if it doesn't have any interior idempotents, i.e.* $\forall x \in]0,1[$: $\wedge(x,x) < x$ *(also called subidempotency).*

Proof. Using Theorem 2.2.5, we can use the limit property as the archimedean property. ($\Rightarrow$) Assume that $\wedge(x,x) = x$ for some $x \in]0,1[$ then $\bigwedge\limits_{n} x = x \; \forall n \in \mathbb{N}$, which contradicts the limit property. ($\Leftarrow$) Moreover, by continuity, note that $\wedge \left(\lim\limits_{n \to +\infty} \bigwedge\limits_{n} x, \lim\limits_{n \to +\infty} \bigwedge\limits_{n} x \right) = \lim\limits_{n \to +\infty} \wedge \left(\bigwedge\limits_{n} x, \bigwedge\limits_{n} x \right)$ for all $x \in [0,1]$, and let $\lim\limits_{n \to +\infty} \bigwedge\limits_{n} x = x_0$. If $x_0 > 0$ then, by assuming that $\forall x \in]0,1[$: $\wedge(a,a) < a$, we obtain $\wedge(x_0,x_0) = \lim\limits_{n \to +\infty} \wedge \left(\bigwedge\limits_{n} x, \bigwedge\limits_{n} x \right) = \lim\limits_{n \to +\infty} \bigwedge\limits_{2n} x < x_0$. But $\lim\limits_{n \to +\infty} \bigwedge\limits_{2n} x = \lim\limits_{n \to +\infty} \bigwedge\limits_{n} x = x_0$ which yields a contradiction. Therefore x_0 must be 0 (it cannot be 1 because T-Norms are monotonous and $x < 1$) and we conclude that $\lim\limits_{n \to +\infty} \bigwedge\limits_{n} x = 0$, i.e. $\wedge$ is archimedean. □

Next we introduce the fundamental theorems that allow for the construction of T-Norms based on a specific type of function, called generator. A subtype of this kind of function, called distance functions, will later be introduced and will allow for the construction of the main concept in this thesis, distance closures.

Definition 2.2.12. A **decreasing generator** is a function $\varphi : [0,1] \to \mathbb{R}$ that is continuous, strictly decreasing and such that $\varphi(1) = 0$.

Definition 2.2.13. The **pseudo-inverse of a decreasing generator** φ, denoted by $\varphi^{(-1)}$ is the function

$$\varphi^{(-1)}(u) = \begin{cases} 1 & \text{, if } u \in]-\infty, 0[\\ \varphi^{-1}(u) & \text{, if } u \in [0, \varphi(0)] \\ 0 & \text{, if } u \in]\varphi(0), +\infty[. \end{cases}$$

Theorem 2.2.7 (Characterization Theorem of T-Norms)**.** *Let* $\wedge : [0,1]^2 \to [0,1]$ *be a function. Then* $\wedge$ *is an Archimedean T-Norm if and only if there exists a decreasing generator* φ *such that* $\forall x, y \in [0,1]$*:*

$$\wedge(x,y) = \varphi^{(-1)}(\varphi(x) + \varphi(y)),$$

in which case we say that φ *is the* ***additive decreasing generator*** *of* $\wedge$*.*

Proof. One of the directions of this theorem is proved next in Lemma 2.2.8 (i). The rest of the proof is however much longer than what would be reasonable to present in this thesis, therefore, for completeness we refer to its formulation by the original authors. Two proofs of this theorem can, thus, be found in the article [21] by Cho-Hsin Ling, where it is labeled as *(3.5) Dual of the Main Theorem*. The first proof is constructed by utilizing a series of previously established lemmas related to associative functions and the second proof relies on well-known results concerning topological semigroups. There is an additional proof in Schweizer [33], where this theorem is a corollary of the theorem concerning multiplicative generators of T-Norms. □

It is, thus, clear that, because of the definition of $\varphi^{(-1)}$, an archimedean T-Norm is continuous if and only if its additive generator is continuous.

Theorem 2.2.8. *If a T-Norm,* $\wedge$*, has an additive generator* $\varphi : [0,1] \to [0, +\infty]$ *then:*

(i) $\wedge$ *is necessarily* ***archimedean****.*

(ii) $\wedge$ *is* ***strictly monotone*** *if and only if* $\varphi(0) = +\infty$

(iii) *Each element of* $]0,1[$ *is a* ***nilpotent*** *element of* $\wedge$ *if and only if* $\varphi(0) < +\infty$

Proof.

(i) Consider arbitrary $x, y \in]0,1[$, then we have $\varphi(x) > 0$ and $\varphi(\frac{y}{2}) < +\infty$. Consequently there is $n \in \mathbb{N}$ such that $n\varphi(x) > \varphi(\frac{y}{2})$ and this implies that $\bigwedge_n x = \varphi^{-1}(n\varphi(x)) < \varphi^{-1}(\varphi(\frac{y}{2})) = \frac{y}{2} < y$. Thus, $\wedge$ is archimedean.

(ii) If $\varphi(0) = +\infty$ then $Range(\varphi) \cup [\varphi(0), +\infty] = Range(\varphi)$ and for all $x > 0$ and $y, z \in [0,1]$ with $y < z$ we get that $\varphi(x) + \varphi(y) \in Range(\varphi)$ and $\varphi(x) + \varphi(z) \in Range(\varphi)$ and $\varphi(x) + \varphi(y) > \varphi(x) + \varphi(z)$. Since $\varphi^{-1}|_{Range(\varphi)}$ is strictly decreasing then $\wedge(x, y) = \varphi^{-1}(\varphi(x) + \varphi(y)) < \varphi^{-1}(\varphi(x) + \varphi(z)) = \wedge(x, z)$, which means that $\wedge$ is strictly monotone.

(iii) If $\varphi(0) < +\infty$ then for each $x \in]0,1[$ we have $\varphi(x) > 0$ and consequently $n\varphi(x) \geq \varphi(0)$ for some $n \in \mathbb{N}$. This means that $\bigwedge_n x = 0$, that is, x is a nilpotent element of $\wedge$ and then $\wedge$ cannot be strictly monotone.

□

Moreover, the same construction of additive generators can also be applied to T-Conorms, however in that context they will not be needed for the development of this work. Nevertheless, in order to better convey the sense of duality between these two operations, we choose to also present them.

Definition 2.2.14. A **increasing generator** is a function $\varphi : [0,1] \to \mathbb{R}$ that is continuous, strictly increasing and such that $\varphi(0) = 0$.

Definition 2.2.15. The **pseudo-inverse of an increasing generator** θ, denoted by $\theta^{(-1)}$ is the function

$$\theta^{(-1)}(u) = \begin{cases} 0 & \text{if } u \in]-\infty, 0[\\ \theta^{-1}(u) & \text{if } u \in [0, \theta(1)] \\ 1 & \text{if } u \in]\theta(1), +\infty[\end{cases}$$

Theorem 2.2.9 (Characterization Theorem of T-Conorms)**.** *Let* $\vee : [0,1]^2 \to [0,1]$ *be a function. Then* $\vee$ *is an Archimedean T-Conorm if and only if there exists an increasing generator* θ *such that* $\forall x, y \in [0,1]$*:*

$$\vee(x, y) = \theta^{(-1)}(\theta(x) + \theta(y))$$

In which case we say that θ *is the* ***additive increasing generator*** *of* $\vee$*.*

Proof. As in the proof of Theorem 2.2.7, this proof can be found in [21] where this theorem is labeled as *(3.3) Main Theorem*. Also, in [21], there is another theorem that states that the theorem 2.2.7 implies this one (and conversely) which could also be used to prove this one. □

2.3 Distance Closures

2.3.1 Proximity-Distance Isomorphism

Definition 2.3.1 (Graph Isomorphism)**.** Two weighted graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are said to be isomorphic if there is a map $\varphi : E_1 \to E_2$ such that:

$$\forall v_1, v_2 \in V : (v_1, v_2) \in E_1 \Leftrightarrow \varphi((v_1, v_2)) \in E_2$$

Alternatively, it is usual to define G_1 and G_2 as isomorphic if there is a map $\varphi : V_1 \to V_2$ such that:

$$\forall v_1, v_2 \in V : (v_1, v_2) \in E_1 \Leftrightarrow (\varphi(v_1), \varphi(v_2)) \in E_2$$

However, for our purposes, it is more appropriate to define the isomorphism over the edges rather than the nodes.

Definition 2.3.2 (Proximity-Distance Map)**.** Let P be the adjacency matrix of a proximity graph $G = (V, E)$ such that $|V| = n$. Let $\varphi : [0,1] \to [0, +\infty]$ and $\Phi : \mathcal{M}_n([0,1]) \to \mathcal{M}_n([0, +\infty])$ be maps, such that:

(a) $\forall a, b \in [0,1] : a > b \Rightarrow \varphi(a) < \varphi(b)$ (strictly monotonic decreasing)

(b) φ is continuous and $\varphi(0) = +\infty$ and $\varphi(1) = 0$

(c) $\Phi(P) = [\varphi(p_{ij})]_{n \times n} \forall x_i, x_j \in V$

Then φ is called a **proximity-distance map** or a **distance function** and Φ is its extension for matrices.

Remark 2.3.1. Note that the name **distance function** in this context aims to encapsulate the purpose of converting proximities into distances and shall not be misunderstood by the concept of **distance function** in metric spaces, which are functions of the kind $d : X \times X \to [0, +\infty[$. Moreover, this last case is a type of function which will be referenced ahead by the name of **distance attribution function**.

Theorem 2.3.1. *Let P be the adjacency matrix of a proximity graph (i.e. a reflexive and symmetric graph) and Φ the matrix extension of a distance function. Then $D = \Phi(P)$ is anti-reflexive and symmetric.*

Proof. P is reflexive so we have $\forall x_i \in V : p_{ii} = 1$ and we know that $D = \Phi(P) = [\varphi(p_{ij})]$. By construction, $\varphi(p_{ij}) = 0$ if $p_{ij} = 1$ thus $\forall x_i \in V : d_{ii} = 0$ and, thus, D is anti-reflexive. P is symmetric so $\forall x_i, x_j \in V : p_{ij} = p_{ji}$ and so, $d_{ij} = \varphi(p_{ij}) = \varphi(p_{ji}) = d_{ji}$ which means that D is also symmetric. □

From now on, one should interpret φ as a function in two different, but connected, domains since it takes as input an edge (x_i, x_j) and the respective proximity weight p_{ij} and returns the same edge but with a distance weight d_{ij}. Thus, this function is an *isomorphism* of graphs from the proximity space into the distance space.

Proposition 2.3.2. *Every distance function is a bijection from* $[0,1]$ *to* $[0,+\infty]$.

Proof. Let $a, b \in [0,1]$ with $a \neq b$. Without loss of generality, assume $a > b$. Then $\varphi(a) < \varphi(b)$ and so φ is injective. Now, since $\varphi(0) = +\infty$ and $\varphi(1) = 0$, by the Mean Value Theorem, $\forall u \in]\varphi(1), \varphi(0)[\ \exists x \in]0,1[: \varphi(x) = u$, and so φ is surjective. Thus, φ is bijective. □

Using the language of T-Norms, it's clear that **distance functions** take the role of **additive decreasing generators**, and given that in a distance function we have $\varphi(0) = +\infty$, we can use Definition 2.2.13 to note that $\varphi^{(-1)}|_{[0,+\infty]}(u) = \varphi^{-1}(u)$. Since we will be dealing with the distance space, $[0,+\infty]$, and thus, will not be considering negative values, it is sufficient to consider the inverse of additive generators instead of the pseudo-inverse. Moreover, by Lemma 2.2.8 and Definition 2.2.9 one concludes that distance functions generate **strict** and **continuous archimedean** T-Norms.

2.3.2 Triangular Distance Norms and Conorms

As seen before, T-Norms and T-Conorms endowed the structure of commutative monotone monoid in the proximity space $[0,1]$. However, this structure can be endowed in other sets, such as $[0,+\infty]$. At this point, then, it is also convenient to define new operations that also endow the structure of commutative monotone monoid in the distance space $[0,+\infty]$, with the appropriate neutral elements.

Definition 2.3.3 (TD-Norm)**.** A Triangular Distance Norm, **TD-Norm**, is a binary operation, $g : [0,+\infty] \times [0,+\infty] \to [0,+\infty]$, that obeys the following properties for all $x, y, z \in [0,+\infty]$:

- (commutativity) $g(x,y) = g(y,x)$
- (associativity) $g(x, g(y,z)) = g(g(x,y), z)$
- (monotonicity) $x \leq y \implies g(x,z) \leq g(y,z)$
- (Boundary Condition) $g(x,0) = x$

Definition 2.3.4 (TD-Conorm)**.** A Triangular Distance Conorm, **TD-Conorm**, is a binary operation, $f : [0,+\infty] \times [0,+\infty] \to [0,+\infty]$, that obeys the following properties for all $x, y, z \in [0,+\infty]$:

- (commutativity) $f(x,y) = f(y,x)$
- (associativity) $f(x, f(y,z)) = f(f(x,y), z)$
- (monotonicity) $x \leq y \implies f(x,z) \leq f(y,z)$

- (Boundary Condition) $f(x, +\infty) = x$, given $u < +\infty$

Theorem 2.3.3. *If φ is a distance function, then:*

(a) *For every pair of T-Norm and T-Conorm $\langle \wedge, \vee \rangle$, there exists a φ-isomorphic pair of TD-Norm and TD-Conorm $\langle g, f \rangle$ given by $g = \varphi \circ \wedge \circ \varphi^{-1}$ and $f = \varphi \circ \vee \circ \varphi^{-1}$.*

(b) *For every pair of TD-Norm and TD-Conorm $\langle g, f \rangle$, there exists a φ-isomorphic pair of T-Norm and T-Conorm $\langle \wedge, \vee \rangle$ given by $\wedge = \varphi^{-1} \circ g \circ \varphi$ and $\vee = \varphi^{-1} \circ f \circ \varphi$.*

In both cases these pairs obey the constraints $\varphi(\wedge(a, b)) = g(\varphi(a), \varphi(b))$ and $\varphi(\vee(a, b)) = f(\varphi(a), \varphi(b))\ \forall a, b \in [0, 1]$.

Proof. Since φ is bijective it's clear that φ^{-1} exists and is well-defined.

(a) We need to prove that $g = \varphi \circ \wedge \circ \varphi^{-1}$ is a TD-Norm and $f = \varphi \circ \vee \circ \varphi^{-1}$ is a TD-Conorm. Assuming $\wedge$ is a T-Norm $(*)$, then g is a TD-Norm because,

(*commutative*) $\varphi \circ \wedge \circ \varphi^{-1}(a, b) = \varphi(\wedge(\varphi^{-1}(a), \varphi^{-1}(b))) \overset{(*)}{=} \varphi(\wedge(\varphi^{-1}(b), \varphi^{-1}(a))) = \varphi \circ \wedge \circ \varphi^{-1}(b, a)$

(*associative*) $\varphi \circ \wedge \circ \varphi^{-1}(a, \varphi \circ \wedge \circ \varphi^{-1}(b, c)) = \varphi \circ \wedge(\varphi^{-1}(a), \varphi^{-1}(\varphi \circ \wedge \circ \varphi^{-1}(b, c))) =$
$\varphi \circ \wedge(\varphi^{-1}(a), \wedge(\varphi^{-1}(b), \varphi^{-1}(c))) \overset{(*)}{=} \underline{\varphi \circ \wedge(\wedge(\varphi^{-1}(a), \varphi^{-1}(b)), \varphi^{-1}(c))}$ and also,
$\varphi \circ \wedge \circ \varphi^{-1}(\varphi \circ \wedge \circ \varphi^{-1}(a, b), c) = \varphi \circ \wedge(\varphi^{-1}(\varphi \circ \wedge \circ \varphi^{-1}(a, b)), \varphi^{-1}(c)) =$
$\varphi \circ \wedge(\wedge \circ \varphi^{-1}(a, b), \varphi^{-1}(c)) = \underline{\varphi \circ \wedge(\wedge(\varphi^{-1}(a), \varphi^{-1}(b)), \varphi^{-1}(c))}$

(*monotonic*) $a \leq b \Rightarrow \varphi^{-1}(a) \geq \varphi^{-1}(b) \overset{(*)}{\Rightarrow} \wedge(\varphi^{-1}(a), \varphi^{-1}(c)) \geq \wedge(\varphi^{-1}(b), \varphi^{-1}(c)) \Rightarrow$
$\varphi(\wedge(\varphi^{-1}(a), \varphi^{-1}(c))) \leq \varphi(\wedge(\varphi^{-1}(b), \varphi^{-1}(c))) \Leftrightarrow \varphi \circ \wedge \circ \varphi^{-1}(a, c) \leq \varphi \circ \wedge \circ \varphi^{-1}(b, c)\ \forall c \in [0, +\infty]$

(*neutral element*) $\varphi \circ \wedge \circ \varphi^{-1}(a, 0) = \varphi(\wedge(\varphi^{-1}(a), \varphi^{-1}(0))) = \varphi(\wedge(\varphi^{-1}(a), 1)) = \varphi(\varphi^{-1}(a)) = a$

Similarly to g, we can prove that $f = \varphi \circ \vee \circ \varphi^{-1}$ enjoys every property of a TD-Conorm, if we assume that $\vee$ is a T-Conorm $(*)$. Moreover, it enjoys the appropriate neutral element,

(*neutral element*) $\varphi \circ \vee \circ \varphi^{-1}(a, +\infty) = \varphi(\vee(\varphi^{-1}(a), \varphi^{-1}(+\infty))) = \varphi(\vee(\varphi^{-1}(a), 0)) \overset{(*)}{=} \varphi(\varphi^{-1}(a)) = a$

(b) We need to prove that $\wedge = \varphi^{-1} \circ g \circ \varphi$ is a T-Norm and $\vee = \varphi^{-1} \circ f \circ \varphi$ is a TD-Norm. Assuming g is a TD-Norm $(*)$, then $\wedge$ is a T-Norm because,

(*commutative*) $\varphi^{-1} \circ g \circ \varphi(a, b) = \varphi^{-1}(g(\varphi(a), \varphi(b))) \overset{(*)}{=} \varphi^{-1}(g(\varphi(b), \varphi(a))) = \varphi^{-1} \circ g \circ \varphi(b, a)$

(*associative*) $\varphi^{-1} \circ g \circ \varphi(a, \varphi^{-1} \circ g \circ \varphi(b, c)) = \varphi^{-1} \circ g(\varphi(a), \varphi(\varphi^{-1} \circ g \circ \varphi(b, c))) =$
$\varphi^{-1} \circ g(\varphi(a), g(\varphi(b), \varphi(c))) \overset{(*)}{=} \underline{\varphi^{-1} \circ g(g(\varphi(a), \varphi(b)), \varphi(c))}$ and also,
$\varphi^{-1} \circ g \circ \varphi(\varphi^{-1} \circ g \circ \varphi(a, b), c) = \varphi^{-1} \circ g(\varphi(\varphi^{-1} \circ g \circ \varphi(a, b)), \varphi(c)) =$
$\varphi^{-1} \circ g(g \circ \varphi(a, b), \varphi(c)) = \underline{\varphi^{-1} \circ g(g(\varphi(a), \varphi(b)), \varphi(c))}$

(*monotonic*) $a \leq b \Rightarrow \varphi(a) \geq \varphi(b) \overset{(*)}{\Rightarrow} g(\varphi(a), \varphi(c)) \geq g(\varphi(b), \varphi(c)) \Rightarrow \varphi^{-1}(g(\varphi(a), \varphi(c))) \leq \varphi^{-1}(g(\varphi(b), \varphi(c))) \Leftrightarrow$
$\varphi^{-1} \circ g \circ \varphi(a, c) \leq \varphi^{-1} \circ g \circ \varphi(b, c)\ \forall c \in [0, +\infty]$

(*neutral element*) $\varphi^{-1} \circ g \circ \varphi(a, 1) = \varphi^{-1}(g(\varphi(a), \varphi(1))) = \varphi^{-1}(g(\varphi(a), 0)) = \varphi^{-1}(\varphi(a)) = a$

Similarly to $\wedge$, we can prove that $\vee = \varphi^{-1} \circ f \circ \varphi$ enjoys every property of a T-Conorm, if we assume that f is a TD-Conorm $(*)$. Moreover, it enjoys the appropriate neutral element,

$$(\textit{neutral element}) \quad \varphi^{-1} \circ f \circ \varphi(a, 0) = \varphi^{-1}(f(\varphi(a), \varphi(0))) = \varphi^{-1}(f(\varphi(a), +\infty)) \overset{(*)}{=} \varphi^{-1}(\varphi(a)) = a$$

Therefore, since we proved (a) and (b), we get $g = \varphi \circ \wedge \circ \varphi^{-1}$ and $f = \varphi \circ \vee \circ \varphi^{-1}$ which, by composing φ on the right in both equations, gives $g \circ \varphi = \varphi \circ \wedge$ and $f \circ \varphi = \varphi \circ \vee$, and these are exactly the constraints we wished for. □

Example 2.3.1. Expanding on the T-Norms examples given in Chapter 2.2.3, we can create the corresponding TD-Norms which will be utilized later on by using the previous theorem.

$$g_{\text{Drastic}}(x, y) = \varphi_1^{\mathsf{D}} \circ \wedge_{\text{Drastic}} \circ (\varphi_1^{\mathsf{D}})^{-1} = \begin{cases} x & \text{if } y = \varphi_1^{\mathsf{D}}(1) = 0 \\ y & \text{if } x = \varphi_1^{\mathsf{D}}(1) = 0 \\ \varphi_1^{\mathsf{D}} \circ \wedge_{\text{Drastic}} \circ (\varphi_1^{\mathsf{D}})^{-1}(0) = +\infty & \text{otherwise} \end{cases}$$

$$g_{\text{Prod}}(x, y) = \varphi_1^{\mathsf{D}} \circ \wedge_{\text{Prod}} \circ (\varphi_1^{\mathsf{D}})^{-1} = \varphi_1^{\mathsf{D}} \circ xy \circ (\varphi_1^{\mathsf{D}})^{-1} = \frac{1}{(\frac{1}{x+1})(\frac{1}{y+1})} - 1 = (x+1)(y+1) - 1$$

$$\begin{aligned} g_{\text{Sum}}(x, y) &= \varphi_1^{\mathsf{D}} \circ \wedge_{\text{HamProd}} \circ (\varphi_1^{\mathsf{D}})^{-1} = \varphi_1^{\mathsf{D}} \circ \frac{xy}{x + y - xy} \circ (\varphi_1^{\mathsf{D}})^{-1} = \frac{1}{\frac{(\frac{1}{x+1})(\frac{1}{y+1})}{\frac{1}{x+1} + \frac{1}{y+1} - (\frac{1}{x+1})(\frac{1}{y+1})}} - 1 \\ &= \frac{1}{\frac{\frac{1}{(x+1)(y+1)}}{\frac{(y+1)+(x+1)-1}{(x+1)(y+1)}}} - 1 = \frac{1}{\frac{\frac{1}{(x+1)(y+1)}}{\frac{y+x+1}{(x+1)(y+1)}}} - 1 = \frac{1}{\frac{1}{x+y+1}} - 1 = x + y + 1 - 1 = x + y \end{aligned}$$

$$\begin{aligned} g_{\text{Max}}(x, y) &= \varphi_1^{\mathsf{D}} \circ \wedge_{\text{Min}} \circ (\varphi_1^{\mathsf{D}})^{-1} = \varphi_1^{\mathsf{D}} \circ \min(x, y) \circ (\varphi_1^{\mathsf{D}})^{-1} = \frac{1}{\min(\frac{1}{x+1}, \frac{1}{y+1})} - 1 \\ &= \begin{cases} x & \text{if } \frac{1}{x+1} \leq \frac{1}{y+1} \\ y & \text{otherwise} \end{cases} = \begin{cases} x & \text{if } x \geq y \\ y & \text{otherwise} \end{cases} = \max(x, y). \end{aligned}$$

In order to compare different g's we must make sure that the distance spaces are all the same. For that reason, we use the most algebraically simple distance function isomorphism, $\varphi_1^{\mathsf{D}} = \frac{1}{x} - 1$.

Analogously to Proximity Graphs, we can now define the usual operations on Distance Graphs as well as their Distance Closure, using the notation of TD-Norms and TD-Conorms.

Definition 2.3.5 (Operations on Distance Graphs)**.** Let $D_1, D_2 \in \mathcal{M}_n([0, +\infty])$ be adjacency matrices of distance graphs G_1 and G_2. Then, we define **intersection**, **union** and **composition** of distance graphs as:

$$D_1 \overset{\bullet}{\cap} D_2 = [d'_{ij}]_{n\times n} \text{ where } d'_{ij} = g(d_{1ij}, d_{2ij})$$

$$D_1 \overset{\bullet}{\cup} D_2 = [d'_{ij}]_{n\times n} \text{ where } d'_{ij} = f(d_{1ij}, d_{2ij})$$

$$D_1 \circ D_2 = [d'_{ij}]_{n\times n} \text{ where } d'_{ij} = \underset{1\leq k\leq n}{f} g(d_{1ik}, d_{2kj}),$$

where g is a TD-Norm and f is a TD-Conorm.

Moreover, for an arbitrary distance graph matrix D, we define D^n recursively as $D^0 = id_D$ and $D^n = D^{n-1} \circ D^1$, where id_D denotes identity distance matrix defined by $id_D = \Phi(id_P)$.

Definition 2.3.6 (Distance Closure)**.** Given a distance graph G with weighted adjacency matrix D, the Distance Closure D^∞ of G is defined as

$$D^\infty = \mathop{\overset{\kappa}{\overset{\bullet}{\bigcup}}}_{n=1} D^n$$

In general, $\kappa \to +\infty$, but, like the isomorphic transitive closures, under some conditions this value may be finite.

Remark 2.3.2. If A and B are two matrices in $\mathcal{M}_n(\mathbb{R})$, we define $A \overset{\bullet}{\supseteq} B$ as $\forall i, j \in \{1, \ldots, n\} : a_{ij} \geq b_{ij}$.

Theorem 2.3.4. *Let P be the adjacency matrix of a proximity graph, $G = (V, E)$ with $|V| = s$, and D the distance adjacency matrix obtained by $D = \Phi(P)$, where Φ is the matrix extension of a distance function. Then the following two statements hold:*

(a) $\Phi(P) \overset{\bullet}{\supseteq} \Phi(P^2) \overset{\bullet}{\supseteq} \ldots \overset{\bullet}{\supseteq} \Phi(P^\infty)$

(b) $D \overset{\bullet}{\supseteq} D^2 \overset{\bullet}{\supseteq} \ldots \overset{\bullet}{\supseteq} D^\infty$

Proof. Let p^n_{ij} denote the proximity value of the P^n matrix in the i-th row and j-th column and d^n_{ij} denote the distance value of the D^n matrix in the i-th row and j-th column.

(a) We wish to prove that $\varphi(p^n_{ij}) \geq \varphi(p^{n+1}_{ij})\ \forall x_i, x_j \in V$. Since φ is strictly monotonic decreasing, it is sufficient to prove that $p^n_{ij} \leq p^{n+1}_{ij}\ \forall x_i, x_j \in V$.

Assume, without loss of generality that $1 \leq i \leq j \leq s$, then

$$\begin{aligned}
p_{ij}^{n+1} &= \bigvee_{1 \leq k \leq s} \left(\wedge(p_{ik}^n, p_{kj}^n) \right) \\
&= \bigvee \left(\wedge(p_{i1}^n, p_{1j}^n), \ldots, \wedge(p_{ii}^n, p_{ij}^n), \ldots, \wedge(p_{ij}^n, p_{jj}^n), \ldots, \wedge(p_{is}^n, p_{sj}^n) \right) \\
&= \bigvee \left(\wedge(p_{i1}^n, p_{1j}^n), \ldots, p_{ij}^n, \ldots, p_{ij}^n, \ldots, \wedge(p_{is}^n, p_{sj}^n) \right) \\
&\geq \max \left(\wedge(p_{i1}^n, p_{1j}^n), \ldots, p_{ij}^n, \ldots, p_{ij}^n, \ldots, \wedge(p_{is}^n, p_{sj}^n) \right) \\
&\geq p_{ij}^n .
\end{aligned}$$

(b) We need to prove that $d_{ij}^n \geq d_{ij}^{n+1}\ \forall x_i, x_j \in V$.

Assume, without loss of generality that $1 \leq i \leq j \leq s$, then

$$\begin{aligned}
d_{ij}^{n+1} &= \underset{1 \leq k \leq s}{f} \left(g(d_{ik}^n, d_{kj}^n) \right) \\
&= f\left(g(d_{i1}^n, d_{1j}^n), \ldots, g(d_{ii}^n, d_{ij}^n), \ldots, g(d_{ij}^n, d_{jj}^n), \ldots, g(d_{is}^n, d_{sj}^n) \right) \\
&= f\left(g(d_{i1}^n, d_{1j}^n), \ldots, d_{ij}^n, \ldots, d_{ij}^n, \ldots, g(d_{is}^n, d_{sj}^n) \right) \\
&\leq \min\left(g(d_{i1}^n, d_{1j}^n), \ldots, d_{ij}^n, \ldots, d_{ij}^n, \ldots, g(d_{is}^n, d_{sj}^n) \right) \\
&\leq d_{ij}^n .
\end{aligned}$$

□

Theorem 2.3.5. *Let $G = (V, E)$ be a graph with $|V| = s$. Let P be a proximity matrix and D a distance matrix over G. Also, let φ be a distance function and φ^{-1} its inverse (which exists since φ is bijective). Then,*

(a) If $\langle \wedge, \vee \rangle$ is a T-Norm/Conorm pair and φ an isomorphism such that the isomorphic TD-Norm/Conorm pair $\langle g, f \rangle$ obeys the condition

$$\forall x_i, x_j, x_k \in V : \underset{x_k \in V}{f} (g(\varphi(p_{ik}), \varphi(p_{kj})) = \varphi(\bigvee_{x_k \in V} (\bigwedge(p_{ik}, p_{kj}))) \tag{2.1}$$

then the closures resulting from both pairs are isomorphic, i.e. $\Phi(P^\infty) = D^\infty$.

(b) If $\langle g, f \rangle$ is a TD-Norm/Conorm pair and φ an isomorphism such that the isomorphic T-Norm/Conorm pair $\langle \wedge, \vee \rangle$ obeys the condition

$$\forall x_i, x_j, x_k \in V : \bigvee_{x_k \in V} (\bigwedge(\varphi^{-1}(p_{ik}), \varphi^{-1}(p_{kj})) = \varphi^{-1}(\underset{x_k \in V}{f} (g(p_{ik}, p_{kj}))) \tag{2.2}$$

then the closures resulting from both pairs are isomorphic, i.e. $\Phi^{-1}(D^\infty) = P^\infty$.

Proof. By Theorem 2.3.3, we can already guarantee the existence of the isomorphic pairs. Next, we need to prove that isomorphic pairs lead to isomorphic closures.

(a) Assume that P^{k_1} is the transitive closure of P and D^{k_2} is the distance closure of D where $k_1, k_2 \in \mathbb{N} \cup \{+\infty\}$. Taking $k = \max\{k_1, k_2\}$ let us, then, prove that $\Phi(P^k) = D^k$ for every $k \in \mathbb{N}$. As already defined, $D = \Phi(P)$, thus $D^k = (\Phi(P))^k$, or more concisely, $D^k = \Phi^k(P)$. Hence, we want to prove that $\Phi(P^k) = \Phi^k(P)$. Proceeding by induction, the basis case is for $k = 2$, since in $k = 1$ there is no composition of matrices. Therefore, we need to check that $\Phi(P^2) = \Phi^2(P)$, that is,

$$\Phi(P \circ P) = \Phi(P) \circ \Phi(P) \Leftrightarrow \varphi(\bigvee_{x_w \in V} \wedge (p_{iw}, p_{wj})) = \mathop{f}_{x_w \in V} g(\varphi(p_{iw}), \varphi(p_{wj}))\ \forall x_i, x_j, x_w \in V \tag{2.3}$$

which is precisely the condition in Equation 2.1. Then, in the step of the induction we want to show that $\Phi(P^k) \overset{(*)}{=} \Phi^k(P) \Rightarrow \Phi(P^{k+1}) = \Phi^{k+1}(P)$. Indeed, $\Phi^{k+1}(P) = \Phi^k(P) \circ \Phi(P) \overset{(*)}{=} \Phi(P^k) \circ \Phi(P) = \Phi(P^{k+1})$

(b) Follows from the fact that Φ is bijective and so Φ^{-1} exists and is well-defined. Thus, $\Phi(P^k) = D^k \Rightarrow \Phi^{-1}(\Phi(P^k)) = \Phi^{-1}(D^k) \Leftrightarrow P^k = \Phi^{-1}(D^k)$

□

With this theorem we finalize the framework that allows for a complete correspondence between Proximity Space and Distance Space (and their corresponding Transitive Closures and Distance Closures) through the bijection of distance functions. This correspondence can be summarized in the following diagram.

$$\begin{tikzcd}
\mathrm{P} \arrow[r, shift left, "\varphi"] \arrow[d, "\vee \circ \wedge"'] & \mathrm{D} \arrow[l, shift left, "\varphi^{-1}"] \arrow[d, "f \circ g"] \\
\mathrm{P}^{\infty} \arrow[r, shift left, "\varphi"] & \mathrm{D}^{\infty} \arrow[l, shift left, "\varphi^{-1}"]
\end{tikzcd}$$

Figure 2.3: Isomorphism between Transitive and Distance Closures

2.3.3 Algebraic Structures

Although the definition of T-Norms and T-Conorms is restricted to the set $[0, 1]$, their axiomatic properties hint that these operations underlie an algebraic structure on an arbitrary set. Moreover, taking into account that the properties of TD-Norms and TD-Conorms resemble so much the ones of T-Norms and T-Conorms, differing only on the neutral elements, one concludes that there must be some unifying structure that includes both of these pairs of operations. That structure, as mentioned before, is a commutative monoid that enjoys the monotonicity property. Therefore it is pertinent to define what is a monoid. Here, we privilege infix notation for its more natural suitability in this context.

Definition 2.3.7 (Semigroup)**.** A **semigroup** is a pair $(S, \oplus)$ where S is a set and $\oplus$ is an operation $\oplus : S \times S \to S$ that enjoys commutativity:

(*commutativity*) $\forall a, b, c \in S : a \oplus (b \oplus c) = (a \oplus b) \oplus c$

Definition 2.3.8 (Monoid)**.** A **monoid** is a pair $(S, \oplus)$ where S is a set and $\oplus$ is an operation $\oplus : S \times S \to S$ that enjoys the following properties:

(*commutativity*) $\forall a, b, c \in S : a \oplus (b \oplus c) = (a \oplus b) \oplus c$
(*neutral element*) $\exists e \in S \; \forall a \in S : a \oplus e = a = e \oplus a$

Definition 2.3.9 (Selective Monoid)**.** A monoid $(S, \oplus)$ is said to be *selective* if $a \oplus b = a$ or $b \; \forall a, b \in S$.

Hence, both Triangular and Triangular Distance Norms and Conorms induce a commutative and monotone monoid structure on $[0, 1]$ and $[0, +\infty]$, respectively and by Theorem 2.3.3 we get that, in particular, φ is indeed an isomorphism of monoids ($([0, 1], \wedge) \overset{\varphi}{\simeq} ([0, +\infty], g)$ and $([0, 1], \vee) \overset{\varphi}{\simeq} ([0, +\infty], f)$).

Furthermore, as alluded in the definition of composition for fuzzy graphs, in order to make computations over graphs to determine some property like transitivity, there is the need for applying both these operations on the same set of elements, in these cases, $[0, 1]$ and $[0, +\infty]$. Therefore, the set must have a richer structure that encompasses two operations. This structure is called a semiring which is the natural generalization of semigroup for a structure with two operations. The description of these structures is mostly available in [36] but is treated with much more depth in [14].

Definition 2.3.10 (Semiring)**.** A **semiring** is a triple $(S, \oplus, \otimes)$ where S is a set and $\oplus$, $\otimes$ are operations defined as $\oplus : S \times S \to S$ and $\otimes : S \times S \to S$ such that:

(a) (*associativity* of $\oplus$) $\forall a, b, c \in S : a \oplus (b \oplus c) = (a \oplus b) \oplus c$
(b) (*commutativity* of $\oplus$) $\forall a, b, c \in S : a \oplus b = b \oplus a$
(c) (*neutral element* of $\oplus$) $\exists \varepsilon \in S \; \forall a \in S : a \oplus \varepsilon = a = \varepsilon \oplus a$
(d) (*associativity* of $\otimes$) $\forall a, b, c \in S : a \oplus (b \oplus c) = (a \oplus b) \oplus c$
(e) (*neutral element* of $\otimes$) $\exists e \in S \; \forall a \in S : a \oplus e = a = e \oplus a$
(g) (*distributivity* of $\otimes$ over $\oplus$) $\forall a, b, c \in S : a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$
(h) (*absorbency* of ε for $\otimes$) $\forall a \in S : a \otimes \varepsilon = \varepsilon = \varepsilon \otimes a$

Using the definitions from before we could summarize properties (a), (b) and (c) as $(S, \oplus)$ being a commutative monoid and properties (d) and (e) as $(S, \otimes)$ being a monoid.

Definition 2.3.11 (Canonical Order)**.** Let $(S, \oplus)$ be a monoid. A **canonical preorder** is a binary relation $\leq$ defined as $a \leq b \Leftrightarrow \exists c \in S : b = a \oplus c$. This relation is reflexive and transitive since:

(*reflexive*) Trivially, $\exists \varepsilon \in S : a = a \oplus \varepsilon \Leftrightarrow a \leq a$

(*transitivity*) $[a \leq b$ and $b \leq c] \Leftrightarrow [\exists c_1 \in S : b = a \oplus c_1$ and $\exists c_2 \in S : c = b \oplus c_2] \Rightarrow c = b \oplus c_2 = a \oplus c_1 \oplus c_2 = a \oplus c_3 \Rightarrow a \leq c$

A monoid is said to be **canonically ordered** if the canonical preorder is an order, that is, if $\leq$ is antisymmetric ($a \leq b$ and $b \leq a \Leftrightarrow a = b$).

Definition 2.3.12 (Dioid)**.** A semiring $(S, \oplus, \otimes)$ is called a **diod** when the monoid $(S, \oplus)$ is **canonically ordered**.

Definition 2.3.13 (Selective Dioid)**.** A dioid $(S, \oplus, \otimes)$ is said to be *selective* if $(S, \oplus)$ is *commutative* and *selective*.

Proposition 2.3.6. *If $(S, \oplus)$ is commutative and idempotent then the canonical preorder relation $\leq$ is an order relation.*

Proof. We already checked that $\leq$ is *reflexive* and *transitive*. Thus, we need to check it is also *antisymmetric*.

$$a \leq b \Rightarrow \exists c_1 : b = a \oplus c_1 \text{ and } b \leq a \Rightarrow \exists c_2 : a = b \oplus c_2$$

Therefore $a = a \oplus c_1 \oplus c_2$ and $b = a \oplus c_1 = a \oplus c_1 \oplus c_2 \oplus c_1 = a \oplus c_1 \oplus c_2 = a$ which proves antisymmetry. □

Proposition 2.3.7. *If $(S, \oplus)$ is selective and commutative then $\leq$ is a total order relation.*

Proof. Selectivity implies idempotency ($\forall x \in S\, s \oplus s = s$), therefore $\leq$ is an order relation. Furthermore, $a \oplus b = a$ or b implies that for every $a, b \in S$ we have either $a \leq b$ or $b \leq a$, which proves that $\leq$ is a total order. □

In the notation of T-Norms/T-Conorms pairs and TD-Norms/TD-Conorms pairs, we obtain the following correspondence:

S	$\oplus$	neutral element ε	$\otimes$	neutral element e
$[0, 1]$	$\vee$	0	$\wedge$	1
$[0, +\infty]$	f	$+\infty$	g	0

Table 2.2: Correspondence between semirings notation and T-Norms/T-Conorms and TD-Norms/TD-Conorms pairs

Thus, $([0, 1], \vee, \wedge)$ is a semiring whenever $\wedge$ is distributive over $\vee$ and 0 is absorbent for $\wedge$. We already checked that $\wedge(a, 0) = 0\ \forall a \in [0, 1]$, thus it is only necessary to obtain the distributivity in order to get a semiring. The same reasoning applies to $([0, +\infty], f, g)$ as $g(a, +\infty) = +\infty\ \forall a \in [0, +\infty]$ and we just require that g is distributive over f. However, in general, these operations don't distribute unless for a very specific pair of T-Norm and T-Conorm which we discuss now.

Definition 2.3.14. Let $\wedge$ be a T-Norm and $\vee$ be a T-Conorm. We say that

$\wedge$ is distributive over $\vee$ if for all $x, y, z \in [0,1] : \wedge(x, \vee(y, z)) = \vee(\wedge(x, y), \wedge(x, z))$

$\vee$ is distributive over $\wedge$ if for all $x, y, z \in [0,1] : \vee(x, \wedge(y, z)) = \wedge(\vee(x, y), \vee(x, z))$

Or, in infix notation,

$\wedge$ is distributive over $\vee$ if for all $x, y, z \in [0,1] : x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

$\vee$ is distributive over $\wedge$ if for all $x, y, z \in [0,1] : x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$

In the case that $\wedge$ is distributive over $\vee$ and $\vee$ is distributive over $\wedge$, then $\langle \wedge, \vee \rangle$ is called a **distributive pair** of T-Norms and T-Conorms.

Proposition 2.3.8. *Let $\wedge$ be a T-Norm and $\vee$ a T-Conorm, then we have*

(i) $\vee$ *is distributive over* $\wedge$ *if and only if* $\wedge = \min$

(ii) $\wedge$ *is distributive over* $\vee$ *if and only if* $\vee = \max$

(iii) $\langle \wedge, \vee \rangle$ *is a distributive pair if and only if* $\wedge = \min$ *and* $\vee = \max$*.*

Proof.

(i) $(\Leftarrow)$ If $\wedge = \min$, by monotonicity of T-Norms, we have $\forall x \in [0,1] : y \leq z \Rightarrow \wedge(x, y) \leq \wedge(x, z)$. Then, $\min(\wedge(x, y), \wedge(x, z)) = \wedge(x, y)$. Also, $\wedge(x, \min(y, z)) = \wedge(x, y)$. Therefore we conclude that $\wedge(x, \min(y, z)) = \wedge(x, y) = \min(\wedge(x, y), \wedge(x, z))$. $(\Rightarrow)$ If $\vee$ is distributive over $\wedge$ then, for all $x \in [0,1]$ we have $x = \vee(x, \wedge(0,0)) = \wedge(\vee(x, 0), \vee(x, 0)) = \wedge(x, x)$ and by Proposition 2.2.4, we conclude that $\vee = \max$.

(ii) Similar to (i).

(iii) Given that (i) and (ii), we can conclude (iii).

□

Hence, by (ii) of Proposition 2.3.8, in the context of T-Norms and T-Conorms, the only semirings are $([0,1], \max, \wedge)$, and by the isomorphism in Theorem 2.3.5, in the context of TD-Norms and TD-Conorms, the only semirings are $([0,+\infty], \min, \wedge)$. Moreover, these *semirings* are in fact *dioids*, since the corresponding canonical preorders are orders.

Lemma 2.3.9. *The monoids* $([0,+\infty], \min)$ *and* $([0,1], \max)$ *are canonically ordered.*

Proof. We need to prove that the canonical preorder relations $a \leq b \Leftrightarrow \exists c \in [0,+\infty] : b = \min(a, c)$ and $a \leq b \Leftrightarrow \exists c \in [0,1] : b = \max(a, c)$ are antisymmetric, i.e. that $a \leq b$ and $b \leq a \Rightarrow a = b$.

For the monoid $([0,+\infty],\min)$, note that $(a \leq b$ and $b \leq a) \Leftrightarrow (\exists c_1 \in [0,+\infty] : b = \min(a,c_1)$ and $\exists c_2 \in [0,+\infty] : a = \min(b,c_2))$. If $b = \min(a,c_1)$ then either $b = a$ (in which case the proof is done) or $b = c_1$. At the same time, since $a = \min(b,c_2)$ we get that either $a = b$ (in which case the proof is done) or $a = c_2$. Thus, we just need to check the case where $b = c_1$ and $a = c_2$. In that case we can write, $b = \min(a,b)$ and $a = \min(b,a)$ so $a = \min(b,a) = b$ and the proof is completed.

Consider now, for the monoid $([0,1],\max)$, that $(\exists c_1 \in [0,1] : b = \max(a,c_1)$ and $\exists c_2 \in [0,1] : a = \max(b,c_2))$. Similarly to the previous case, we just need to check when $b = c_1$ and $a = c_2$. Again, in that case, $b = \max(a,b)$ and $a = \max(b,a)$ so $a = \max(b,a) = b$ and the proof is completed.

□

Since $([0,+\infty],\min)$ and $([0,1],\max)$ are both selective (and commutative) monoids, we could also use Proposition 2.3.7, to conclude that the canonical preorder $\leq$ is an order. From Lemma 2.3.9 and the definition of *dioid*, we have the following result.

Corollary 2.3.9.1. *Any semiring of the kind* $([0,+\infty],\min,g)$*, where the* g *is a TD-Norm, or* $([0,1],\max,\wedge)$*, where the* $\wedge$ *is a T-Norm, is a dioid.*

Definition 2.3.15. Let $(S,\oplus,\otimes)$ be a dioid and $a \in S$, then we define a^k as

$$a^k = \underbrace{a \otimes \cdots \otimes a}_{k \text{ times}} = \bigotimes_k a$$

and $a^{(k)}$ as

$$a^{(k)} = e \oplus a \oplus a^2 \oplus \cdots \oplus a^k = \bigoplus_{j=0}^{k} a^k$$

where e is the neutral element for $\otimes$.

Definition 2.3.16. Given a dioid $(S,\oplus,\otimes)$, we say that an element $a \in S$ is $p-$**stable** if for some $p \in \mathbb{N}_0$ we have $a^{(p+1)} = a^{(p)}$. This also implies that $a^{(p+2)} = e \oplus a \otimes a^{(p+1)} = e \oplus a \oplus a^{(p)} = a^{(p+1)}$ and, by induction, $a^{(p+r)} = a^{(p)}\ \forall r \in \mathbb{N}_0$.
Moreover, for each $p-$**stable** element $a \in S$, we conclude that exists an element $a^* = \lim\limits_{k\to+\infty} a^{(k)} = a^{(p)}$, which is called the **quasi-inverse** of a and satisfies $a^* = a \otimes a^* \oplus e = a^* \otimes a \oplus e$

Extending Definitions 2.2.4 and 2.3.5 to semiring operations we can define the following operations.

Definition 2.3.17 (Operations on Matrices over Semirings)**.** Let $A_1, A_2 \in \mathcal{M}_n(S)$, where $(S,\oplus,\otimes)$ is a semiring. Then, we define $A_1 \otimes A_2 = [a'_{ij}]_{n\times n}$ where $a'_{ij} = \bigoplus\limits_{1\leq k\leq n} \left(a_{1ik} \otimes a_{2kj}\right)$ and $A_1 \oplus A_2 = [a'_{ij}]_{n\times n}$ where $a'_{ij} = a_{1ij} \oplus a_{2ij}$.

Definition 2.3.18. Let $A \in \mathcal{M}_n(S)$ where S is the set underliyng the semiring $(S, \oplus, \otimes)$, i.e. A is a matrix with elements over a semiring. Then, we define A^k as being

$$A^k = \underbrace{A \otimes \cdots \otimes A}_{k \text{ times}}$$

also, $A^{(k)}$ as being

$$A^{(k)} = I \oplus A \oplus \cdots \oplus A^k$$

where

$$I = \begin{bmatrix} e & \varepsilon & \cdot & \varepsilon \\ \varepsilon & \cdot & \varepsilon & \cdot \\ \cdot & \varepsilon & \cdot & \varepsilon \\ \varepsilon & \cdot & \varepsilon & e \end{bmatrix}$$

is the identity matrix over $\mathcal{M}_n(S)$ and the quasi-inverse of A is defined as being the limit

$$A^* = \lim_{k \to +\infty} A^{(k)}$$

if it exists. Moreover the computation of A can be performed using Algorithm 2.2 by replacing the T-Norms and T-Conorms by the appropriate semiring operations. However the convergence of that algorithm is bounded by some constraints, one of them being detailed further ahead.

2.3.4 Convergence of Closures Algorithm

Remark 2.3.3. As was noted before, semirings can generalize, in the presence of distributivity, the properties of T-Norms/Conorms and TD-Norms/Conorms. Thus, it is plausible to consider an abstract adjacency matrix for a graph that can either represent proximities (like P) or distances (like D), and thus have elements from an abstract semiring. With this in mind, in the following results we assume that A is an abstract adjacency matrix of a graph G.

Definition 2.3.19. If $A \in \mathcal{M}_n(S)$ is the adjacency matrix of a graph G, we say that G has no $p-$absorbing circuit if the weight of each *pointed circuit* in G is a $p-$stable element of the semiring S.

Theorem 2.3.10. *In semirings of the kind* $([0, +\infty], \min, g)$ *or* $([0, 1], \max, \wedge)$*, every pointed circuit has a* $0-$*stable weight, and so any graph whose adjacency matrix* A *is over these semirings, has no* $0-$*absorbing circuit.*

Proof. Let A be the adjacency matrix of a graph $G = (X, E)$ and $\gamma = x_1, x_2, \ldots, x_{(k-1)}, x_1$ be a pointed circuit in that graph. Let also $a = g(a_{12}, a_{23}, \ldots, a_{(k-1)1}) \in [0, +\infty]$ be the weight of γ over $([0, +\infty], \min, g)$. Then, we have $a^{(k)} = \min(0, a, g(a, a), \ldots, \underset{k}{g}(a)) = 0 \; \forall k \in \mathbb{N}_0$. Analogously, if $a =$

$\wedge(a_{12}, a_{23}, \ldots, a_{(k-1)1}) \in [0,1]$ is the weight of γ over $([0,1], \max, \wedge)$ then $a^{(k)} = \max(1, a, \wedge(a,a), \ldots, \underset{k}{\wedge}(a)) = 1 \; \forall k \in \mathbb{N}_0$. Hence, we conclude that a is $p-$stable for both of these semirings. □

Theorem 2.3.11. *If G does not have any $0-$absorbing circuit then the sequence of matrices $A^{(k)}$ has a limit A^* when $k \to +\infty$ and this limit is reached for $k \leq n-1$, where n is the dimension of A, that is, the number of nodes in G.*

Proof. The rigorous proof of this theorem is available in [14]. However, we can sketch it here. Note that $A^{(k)}$ accounts for the sum (using $\oplus$) of the weights of the paths with at most k steps and also that, if G has no $0-$absorbing circuit then the weight, a, of any pointed circuit will not be accounted for the weights between any pair of nodes because it will be absorbed by e ($a \oplus e = e$), because a is $0-$stable. Thus, it is sufficient to finish the computation at $A^{(n-1)}$ because at that point we know that we have reached every node from every original node, because the diameter of a graph can be at most $|X| - 1$, and we didn't account any additional circuit weight to the final weights. □

Theorem 2.3.12. *Let P be the proximity adjacency matrix of a graph $G = (X, E)$ and $\langle \wedge, \vee \rangle$ a T-Norm/Conorm pair. If $([0,1], \wedge, \vee)$ is a dioid, the transitive closure of P can be computed in finite κ using Algorithm 2.2.*

Proof. If $\wedge$ is a T-Norm, $\vee$ is a TD-Norm and $([0,1], \vee, \wedge)$ is a dioid then $\vee$ must be $\max$ by Proposition 2.3.8. Also, in a dioid $([0,1], \max, \wedge)$, G has no $0-$absorbing circuit by Theorem 2.3.10. Lastly, by Theorem 2.3.11 we get that the Algorithm 2.2 converges in $\kappa \leq |X| - 1$ steps. □

Theorem 2.3.13. *Let $D(X)$ be a distance matrix and $([0, +\infty], f, g)$ an algebraic structure with $\langle f, g \rangle$ being a TD-Conorm/TD-Norm pair. If $([0, +\infty], f, g)$ is a dioid, then the distance closure can be computed via the transitive closure of the isomorphic graph with adjacency matrix P, where $P = \Phi^{-1}(D)$, and using the algebraic structure $([0,1], \vee, \wedge)$. In other words, if $([0, +\infty], f, g)$ is a dioid, we obtain an algebraic structure $([0,1], \vee, \wedge)$ which is also a dioid, via an isomorphism satisfying Theorem 2.3.5.*

Proof. The proof follows from the Theorems 2.3.4, 2.3.5 and 2.3.12. □

There are other constraints on the graph G, or more specifically on its adjacency matrix, A, that guarantee other finite bounds for the convergence of the quasi-inverse matrix (or transitive closure, or distance closure). However those constraints are not as relevant as this one for this work and are left out but can be found in [14]. In fact, the closures involving the semirings $([0,1], \max, \wedge)$ and $([0, +\infty], \min, g)$, are the most helpful in our context because the most important family of closures that we will deal with is based on them and is called the family of Shortest-Path Distance Closures.

2.3.5 Shortest-Paths Distance Closures

Within the category of Distance Closures, there are a few particular closures of more interest and applicability. Two of them are the Diffusion Closure [36] and Shortest-Paths Distance Closures. The first is a particular closure where the T-Norm/T-Conorm pair is the dual pair of Hamacher Product and Hamacher Sum, and has some applicabilities in modelling diffusion processes on networks. The second is the family we will be mainly focused on and concerns Distance Closures where the TD-Conorm, f, is fixed as $\min$, which is isomorphic to choosing the T-Conorm, $\vee$, as $\max$. Therefore, in these closures the only choice we have relies on the way we aggregate distances to compute path-lengths (or, isomorphically, how we aggregate proximities), because the way we choose the best path over all paths between two nodes is fixed. Therefore, in the case of Shortest-Paths Distance Closures the diagram 2.3 becomes:

$$\begin{tikzcd}
\mathsf{P} \arrow[r, shift left, "\varphi"] \arrow[d, "\max\circ\wedge"'] & \mathsf{D} \arrow[l, shift left, "\varphi^{-1}"] \arrow[d, "\min\circ g"] \\
\mathsf{P}^{\infty} \arrow[r, shift left, "\varphi"] & \mathsf{D}^{\infty} \arrow[l, shift left, "\varphi^{-1}"]
\end{tikzcd}$$

Figure 2.4: Isomorphism between Proximity and Shortest-Path Distance Closures

This type of closures is specially relevant when modelling phenomena where the most relevant property is the proximity between nodes, or entities. One of these phenomena is the propagation of infectious diseases on a network of human individuals. Hence, the impact of these closures in this kind of dynamic process is analysed further in this work.

2.3.6 (Shortest-Paths) Distance Backbones

In the computation of Shortest-Paths Distance Closures there are edges whose weights are invariant because they correspond to the shortest path between those two nodes. For this reason there is a useful construction, based solely on these edges, that sparsifies the original graph by only considering the edges that belong to the shortest paths [35].

Definition 2.3.20. The **distance backbone** of a distance graph $G = (X, E)$ with distance matrix $D(X)$ and distance closure $D^{\infty,g}$ computed with $\langle f \equiv \min, g\rangle$ is the graph B^g defined by its adjacency matrix, $B^g(X)$ in the following way:

$$b_{ij}^{g} = \begin{cases} b_{ij} & \text{if } d_{ij} = d_{ij}^{\infty,g} \\ +\infty & \text{if } d_{ij} > d_{ij}^{\infty,g} \end{cases}$$

where $b_{ij}^g = +\infty$ means that the edge (x_i, x_j) doesn't exist in B.

The inclusion, or not, of an edge in the distance backbone relies uniquely on the inequality given by the computation of distance closures:

$$d_{ij} \leq \min_k g(d_{ik}, d_{kj})$$

In this formulation, d_{ij} must be less than or equal to the minimum of all first order paths between two nodes. Hence, it must also be less than or equal to each of those paths, as can be described by the following *generalized triangular inequality*.

$$d_{ij} \leq \; g(d_{ik}, d_{kj}) \, \forall x_i, x_j, x_k \in X$$

And, as was previously described, this can be generalized to any n-th order path $(x_i, x_{k_1}), \ldots, (x_{k_n}, x_j)$:

$$d_{ij} \leq \; g(d_{ik_1}, \ldots, d_{k_n j}).$$

Moreover, the edges that obey this *generalized triangular inequality* are called **triangular** and the ones that don't are called **semi-triangular**, analogously to *semi-metrics* that relax the usual *triangular inequality* in *metrics*.

The construction of distance backbones can reveal to be very useful in some contexts, however, when applying this technique to unweighted graphs one cannot retrieve any additional information.

Theorem 2.3.14 (Distance Backbone of Unweighted Graphs)**.** *If $D(X)$ is an unweighted graph, then its distance backbone, $B^g(X)$ is the entire graph, that is $D(X) \equiv B^g(X)$.*

This theorem is a consequence of some results proved before. One, is the fact that the transitive closure of an unweighted graph is the complete graph over the same set of nodes, and also, the fact that the backbone is the set of edges in the closure whose weight stays the same as in the original graph. Converting any transitive closure to an isomorphic distance closure we obtain exactly the same graph (topologically) and, thus, the distance closure of an unweighted is also the complete graph. Also, since in unweighted graphs the edge weights are 1 if they exist and 0 if they do not, it is clear that exactly all the edges stay the same in the distance closure and so are all in the backbone.

Theorem 2.3.15 (Backbone Sufficiency)**.** *Let $D(X)$ be a distance graph and $B^g(X)$ its distance backbone given by a TD-Norm g. Then, the shortest-paths distance closure associated with g of $D(X)$ and $B^g(X)$ are the same, $D^{\infty,g}(X) \equiv B^{\infty,g}(X)$*

Proof. We just need to prove that the edges in the backbone are sufficient to compute the shortest paths between every pair of nodes in $D(X)$ since, once those edges are present, the shortest path distance closure will yield the same result because it only chooses the minimum value.

Assume $d_{ij} > g(d^{\infty}_{ik}, d^{\infty}_{kj})\, \forall x_i, x_k, x_j \in X$. Therefore $b^g_{ij} = +\infty$ and there must exist an indirect path between x_i and x_j that uses the intermediate nodes from $K \subset X$ such that $l_{ij} = g(d_{ik_1}, \ldots, d_{k_n j}) < d_{ij}$. Since the shortest path distance closure has $f \equiv \min$, then $d^{\infty}_{ij} = \min_K(l_{ij})$ where K denotes an arbitrary sequence of nodes that induce a path from x_i to x_j. Then, it is clear that the edge (x_i, x_j) is not used for the computation of d^{∞}_{ij} nor the length of any shortest path that goes through x_i and x_j, because every path in those conditions should go through the other indirect shortest path.

Also, if $d_{ij} \leq g(d^{\infty}_{ik}, d^{\infty}_{kj})\, \forall x_i, x_k, x_j \in X$, then it is an edge in $B^g(x)$ and it is sufficient to compute the shortest path between x_i, x_j since there is no other shortest path. In this case, $b_{ij} = d_{ij} = d^{\infty}_{ij}$. □

Corollary 2.3.15.1 (Preservation of connectedness)**.** *Given a connected distance graph $D(X)$ then, for every TD-Norm g, its distance backbone $B^g(X)$ is also connected.*

Proof. If $D(X)$ is a connected graph, we've see that its distance closure, $D^{\infty,g}$ is a complete graph for every TD-Norm g. Then, by Theorem 2.3.15, we conclude that $B^g(X)$ must also be connected, otherwise it would not produce the same complete graph as the distance closure but a disjoint union of complete graphs. □

Corollary 2.3.15.2 (Preservation of bridges)**.** *Given a distance graph $D(X)$ then, for every TD-Norm g, all bridges of $D(X)$ are contained in the distance backbone $B^g(X)$.*

Proof. Since the removal of a bridge implies that the graph gets disconnected and we've seen that the backbone preserves connectedness, it is clear that is must also preserve all bridges. □

The importance of these corollaries highlights a very significant advantage that distance backbones have when compared to other graph reduction techniques that do not necessarily preserve connectedness such as thresholding edge weights or other types of backbones.

Definition 2.3.21. Given a distance graph $D(X)$ and its distance backbone $B^g(X)$ associated with TD-Norm g, we define these two proportions:

- *Proportion of semi-triangular edges*:

  $$\sigma^g(D) = \frac{|\{d_{ij} : d_{ij} > d^{\infty,g}_{ij}\}|}{|\{d_{ij}\}|}\ \forall x_i, x_j \in X \text{ with } i > j$$

  This proportion is a measure of edge redundancy because it translates the proportion of edges that are redundant for the computation of shortest paths.

- *Proportion of triangular edges*:

  $$\tau^g(D) = \frac{|\{d_{ij} : d_{ij} = d^{\infty,g}_{ij}\}|}{|\{d_{ij}\}|} = \frac{|\{b^g_{ij}\}|}{|\{d_{ij}\}|}\ \forall x_i, x_j \in X \text{ with } i > j$$

In this case, the value represents the proportion of edges that are kept in the distance backbone with respect to the original graph.

These definitions focus on undirected graphs and so require $i > j$. However, for directed graphs this condition is relaxed since we want to count every edge no matter its direction. Also, for computational purposes, we require that the adjacency matrices are reflexive ($d_{ii} = 0$ or $p_{ii} = 1 \, \forall x_i \in X$) and this would mean that the graph had a loop over every node, however, this is only a computational requirement and these edges do not really need to exist. Therefore we also do not tally them in these proportions. Since both numerators are complementary over the set of edges in the graph, we can conclude that these proportions are related by $\tau^g(D) = 1 - \sigma^g(D)$.

Definition 2.3.22. Given a distance graph $D(X)$ and its distance closure $D^{\infty,g}(X)$ associated with TD-Norm g, we define the measure of *semi-triangular edge distortion* as:

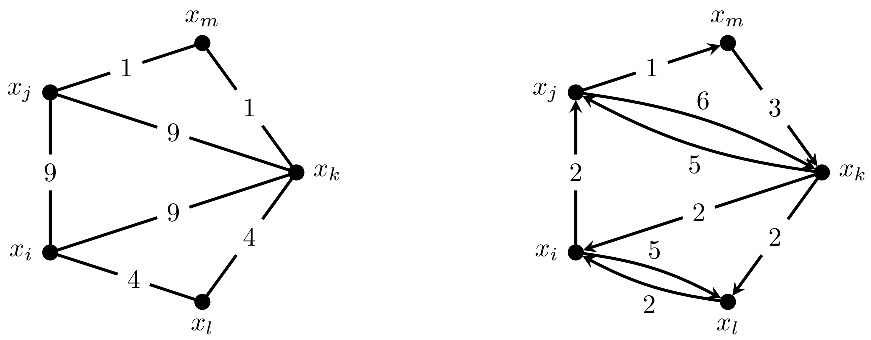
$$s^g_{ij} = \frac{d_{ij}}{d^{\infty,g}_{ij}} \; \forall x_i, x_j \in X \text{ with } i \neq j$$

Hence, it is clear that triangular edges have *semi-triangular distortion* $s^g_{ij} = 1$ and semi-triangular edges have $s^g_{ij} > 1$.

This measure allows to distinguish between edges that break the *generalized triangle inequality* by a small margin and other that break it by large amounts. The distinction between these types of edges reveals itself to be important when sparsifying graphs by the distortion value of the edges.

As highlighted several times before, there is an infinite number of possible TD-Norms g that allow for the computation of distance backbones. Nonetheless, there are a few special cases that entail backbones with well-known properties. To better visualize and interpret these special backbones let us start by considering a simple example of a weighted graph inspired by the one presented in [35].

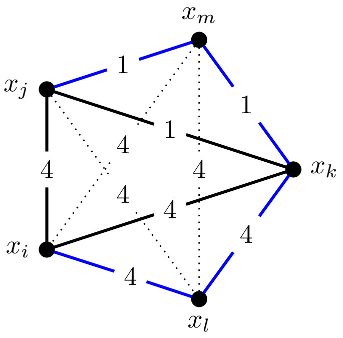
Example 2.3.2. Undirected and Directed Unweighted Graph



2.3.6.A Ultra-Metric Backbone

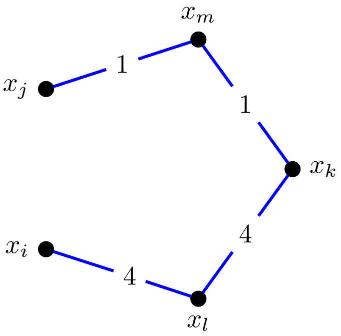
The Ultra-Metric Backbone, $B^u(X)$, is the backbone computed through the distance closure associated with the pair $\langle f \equiv \min, g_{um} \equiv \max\rangle$. This is to say that an edge d_{ij} belongs to this backbone if it obeys the *ultra-metric triangle inequality* $d_{ij} \leq \max(d_{ik}, d_{kj})\ \forall x_i, x_j, x_k \in X$, in which case the edge is said to be *ultra-metric*, instead of the general term, *triangular*. In case an edge doesn't belong to this backbone it is said to be *semi-ultra-metric.* Because of the employed TD-Norm and TD-Conorm names in this backbone, this choice of path length computation is often called the *minimax-path*.

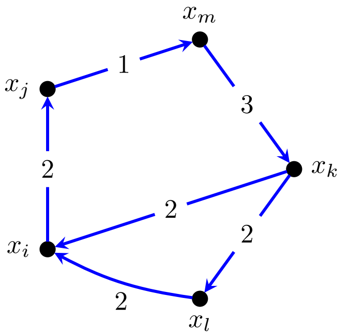
Example 2.3.3. Ultra-Metric Closure of the Undirected Graph in Example 2.3.2



The ultra-metric closure of the directed graph is a also a complete (directed) graph, because the original graph is strongly connected, but since it has twice as many edges, the figure would be too cumbersome to be intuitive.

Example 2.3.4. Ultra-Metric Backbone of the Graphs in Example 2.3.2





The distance space resulting from the ultra-metric distance closure is called an ultra-metric space (X, d), where X is the set of nodes and d is the distance attribution function induced by the distance edge weights. This is because, by construction, distance graphs are already in the conditions of ultra-metric spaces apart from the *ultra-metric triangle inequality*, $d_{ij} \leq \max_k\{d_{ik}, d_{kj}\}$, which is now fulfilled.

In the case of the directed graphs, this is not necessarily true since the distance function might not be symmetric.

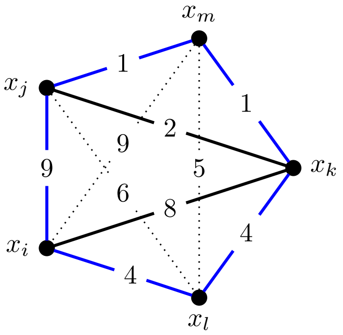
2.3.6.B Metric Backbone

The framework of distance backbones is based on a notion of *generalized triangle inequality* that is formulated with an arbitrary TD-Norm g. However, this abstraction was actually formulated with the purpose of generalizing the usual well-known inequality with $g_m \equiv +$, the *triangle inequality*. Therefore, the Metric Backbone, $B^m(X)$, is computed with the pair $\langle f \equiv \min, g \equiv + \rangle$, and thus selects the edges that obey the condition

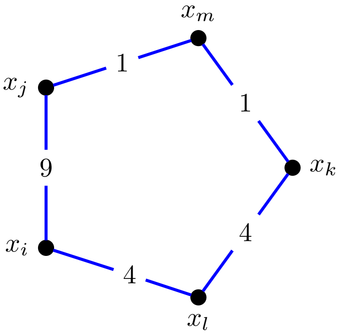
$$d_{ij} \leq d_{ik} + d_{kj} \; \forall x_k \in X$$

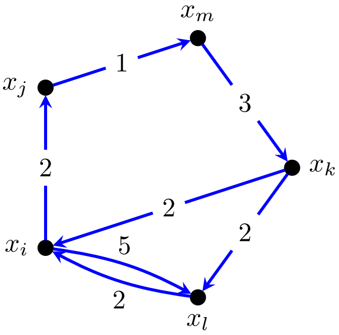
The name *metric* in this context is used because the induced distance function on this closure ends up accomplishing every requirement of a *metric*.

Example 2.3.5. Metric Closure of the Undirected Graph in Example 2.3.2



Example 2.3.6. Metric Backbone of the Graphs in Example 2.3.2



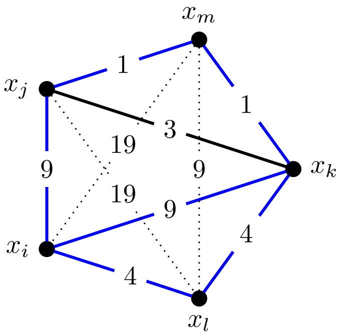


Given a graph $G = (X, E)$, the metric closure produces a metric space (X, d) where d is given by the distance attribution function $d : E \to [0, +\infty]$. Analogously to the ultra-metric closure, this space is metric because distance graphs fulfill every condition of metric spaces except the *triangular inequality*, which is obtained by the closure. In the case of directed graphs the symmetry condition might be broken, and in that case the distance attribution function d is said to be a *quasimetric* instead of a *metric* [38].

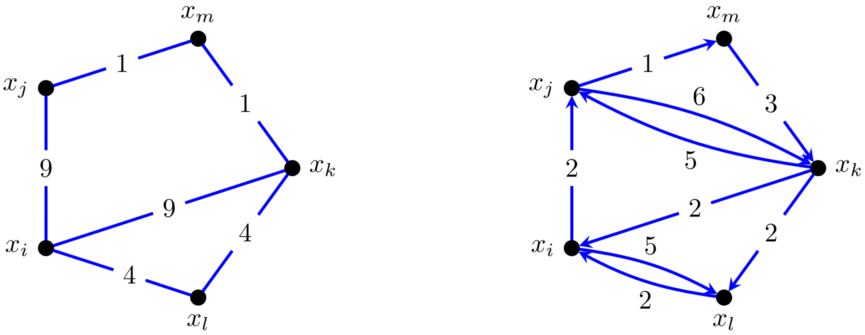
2.3.6.C Product Backbone

The Product Backbone, $B^p(X)$, is obtained by computing the distance closure with the pair $\langle f \equiv \min, g_p \equiv (u+1)(v+1)-1 \rangle$. Thus, this closure computes path-lengths by using an adjusted product of distances. Moreover, this TD-Norm is isomorphic to the exact product T-Norm $\wedge \equiv xy$ through the isomorphism $\varphi \equiv \frac{1}{x} - 1$, which is detailed later in Section 2.5.

Example 2.3.7. Product Closure of the Undirected Graph in Example 2.3.2



Example 2.3.8. Product Backbone of the Graphs in Example 2.3.2



For the Product Closure there isn't any well-known topological metric space that incorporates all its properties, however this space makes sense in several contexts where the compound paths over proximity edges have some parallel with composition/product of probabilities.

2.3.7 Comparing important Backbones

Another intuitive backbone that can be computed is the Euclidean Backbone $B^e(X)$, that uses $g_e \equiv \sqrt{x^2+y^2}$. In the previous example this backbone wouldn't be different from the backbones previously shown in terms of edges that it kept and for that reason it's not presented. Nevertheless, there is a result that compares all these relevant backbones in terms of their sizes:

Proposition 2.3.16. *Let the set $B(X)$ denote the set of edges of the backbone B over a weighted graph with set of nodes X. Then, it is true that:*

$$B^{um}(X) \subseteq B^e(X) \subseteq B^m(X) \subseteq B^p(X) \tag{2.4}$$

Proof. For a given edge (i,j) to be in a backbone obtained using g, it is equivalent that $d_{ij} \leq g(d_{ik}, d_{kj}) \forall k \in X$. Thus we need to prove that

$$g_{um}(x,y) \leq g_e(x,y) \leq g_m(x,y) \leq g_p(x,y) \; \forall x,y \in [0,+\infty]$$
$$\Leftrightarrow \max(x,y) \leq \sqrt{x^2+y^2} \leq x+y \leq (x+1)(y+1)-1 \; \forall x,y \in [0,+\infty]$$

It is clear that $(\max(x,y))^2 \leq x^2+y^2 \Rightarrow \max(x,y) \leq \sqrt{x^2+y^2}$. Also, $x^2+y^2 \leq x^2+2xy+y^2 \Rightarrow x^2+y^2 \leq (x+y)^2 \Rightarrow \sqrt{x^2+y^2} \leq x+y$. Lastly, $x+y \leq x+y+xy = (x+1)(y+1)-1$. □

The Ultra-Metric Backbone operates under the assumption that the only relevant part of a path is the segment with the greatest distance. This assumption is particularly meaningful in scenarios where clusters of nodes are widely separated from each other, with large distance gaps in-between them. In contrast, the Euclidean Backbone assumes that the graph is embedded in a Euclidean space, and each edge traversal corresponds to movement along a dimension in that space. Additionally, the Metric Backbone, which is perhaps the most intuitive approach to computing path-lengths, calculates the sum of the costs associated with each traversed edge, similarly to planning a route over a road network. Lastly, the Product Backbone represents the distance isomorphic solution to the Maximum Reliability Path Problem, as outlined in [14]. When dealing with a network containing independent probabilities associated with its edges, the maximum reliability path concerns the identification of the most reliable path, characterized by higher probability, between two nodes. The transitive closure computed using the Product T-Norm provides the desired solution to this problem by aggregating probabilities through multiplication.

2.4 All Pairs Shortest Paths Problem

Recalling the definition for the metric closure of a distance graph with TD-Norm/Conorm pair $\langle f \equiv \min, g \equiv +\rangle$ and its respective backbone, $B^m(X)$, it is easy to note that this is a new formulation of a very well-known problem in Graph Theory and Computer Science called the All Pairs Shortest Paths (*APSP*) Problem. This problem aims at finding the shortest path between any pair of nodes in a weighted graph and is, thus, related with the Single Source Shortest Paths (*SSSP*) Problem which also concerns the computation of shortest paths but with a fixed source node. Briefly, given a weighted graph $G = (V, E)$, directed or undirected, the APSP Problem concerns the finding of the value $m_{ij} = \min_{K \subset X} w_{ik_1} + \cdots + w_{k_n j}$ for every pair $x_i, x_j \in X$, of unordered or ordered nodes, respectively. It is quite obvious that the Algorithm 2.2 solves this problem when $\langle \wedge \equiv \max, \vee \equiv \frac{xy}{x+y-xy}\rangle$ and the weights are $w_{ij} \in [0,1]$, because this is isomorphic to choosing $\langle f \equiv \min, g \equiv +\rangle$ using the weights given by $\varphi(w_{ij}) \equiv \frac{1}{w_{ij}} - 1$, which is to say using Definition 2.3.6.

The *SSSP* Problem can also be framed as a linear system over a semiring (i.e. where the adjacency matrix of a graph has entries in a semiring) and the *APSP* problem as the quasi-inversion of that same matrix, A [14]. Assuming that the time it takes to execute $\oplus$ is k_1 and $\otimes$ is k_2, then, given that the standard algorithm for matrix multiplication has computational complexity $\mathcal{O}(n^3)$, we can deduce the computational complexity of creating the quasi-inverse matrix by considering Algorithm 2.2 and Definition 2.3.17. Thus this algorithm has time complexity of $\mathcal{O}((k_1 + k_2)n^3 + (n-1)[k_1 n^2 + (k_1 + k_2)n^3]) = \mathcal{O}(k_1(n^4 + n^3 - n^2) + k_2 n^4)$, where $n = |V|$.

Although there are recent developments about faster algorithms for matrix multiplication, this procedure still requires substantially more space (and time) then other approaches that opt for navigating the graph structure, essentially reproducing the comparison of Table 1.1. These alternative approaches tend to be even more efficient if the graphs are relatively sparse ($|E| < |X|^2$). There is a wide variety of different algorithms that tackle this problem in such a way. One of them is known as the **Floyd-Warshall Algorithm**, because it was presented in [12] by Robert Floyd, who based its algorithm in a previous one by Stephen Warshall [41]. Actually, the original formulation of the algorithm by Warshall is exactly the same as the one of Algorithm 2.2 for the Transitive Closure of a graph, although it only considers unweighted graphs because it only concerns boolean matrices. Nevertheless, the current formulation of this algorithm is based on the idea of dynamic programming, where problems are solved by incrementally and recursively updating our final output considering sub-problems of the initial one, and it has computational time complexity of $\mathcal{O}(|V|^3)$ [22]. Another relevant algorithm is known as **Bellman–Ford Algorithm** and, unlike **Floyd-Warshall Algorithm**, this one can detect negative cycles in a graph that allows negative weights. However, the complexity of this algorithm is $\mathcal{O}(|V||E|)$ and in our case the graphs have non-negative weights so this ends up not being the best choice. Another algorithm, which is one of the most well-known in Computer Science, is the **Dijkstra Algorithm** which was presented by Edsger W. Dijkstra in the seminal paper [10] of 1959. This algorithm, in practice, solves the *SSSP* Problem but can be used to solve the *APSP* by iterating the same algorithm over every node. The complexity of **Dijkstra Algorithm**, which is the best among the solutions presented, is $\mathcal{O}(|E| + |V| \log |V|)$ in the case where the queue Q is implemented with a Fibonacci Heap. For this reason, we present in Appendix B the outline of this algorithm, although, for clarity, with a naive implementation of the queue Q.

It is worth mentioning that the Dijkstra Algorithm B.1 computes the shortest-path lengths for nodes that are reachable from the source node and, therefore, leaves any other path-length assigned with $+\infty$. Thus, we can now create the expanded algorithm that solves the *APSP* Problem.

Algorithm 2.3 APSP Dijkstra Algorithm
Input: Weighted Distance Graph $G = (V, E)$

```
dists ← EmptyMatrix($(|V| \times |V|)$)
FOR EACH $v \in V$ :
    dist, prev ← Dijkstra($(G, v)$)
    #Store the current dist list as the $v$-th row of dists
    dists[$v$] ← dist
RETURN dists
```

Output: Shortest-Paths Distances Matrix `dists`
Description: `dist` stores the minimum distance between every pair of nodes

In the case of Algorithm 2.3, the **FOR** loop has to be done over every node v of the graph and every shortest-path starting in that node must be computed because the graph might be directed and, thus, the shortest-paths lengths might not be symmetric. In the case of undirected graphs, one could run Dijkstra from every node but only initially adding to Q the nodes that have not been source nodes in the Dijkstra's computed before, and this way only compute half of the shortest-paths, since the rest are symmetric.

Given the computational advantage of using Dijkstra Algorithm to compute shortest paths in $\langle \min, + \rangle$ and that we established the existence of isomorphisms from $\langle \min, g \rangle$ to $\langle \max, \wedge \rangle$ by Theorem 2.3.5, we can create a way of converting any (shortest-path) distance closure $\langle \min, g \rangle$ into a shortest-path problem in $\langle \min, + \rangle$. This is done using the isomorphisms in the next diagram. Here, $\varphi_1^D = \frac{1}{x} - 1$ and $(\varphi_1^D)^{-1} = \frac{1}{x+1}$ because, algebraically, this is the simplest isomorphism possible. As can be seen in the diagram 2.5 we can choose any path-length aggregation function g, or TD-Norm, and with that we fix the φ_1^D-isomorphic T-Norm, $\wedge$, and consequently we obtain the unique isomorphism φ that should be used to convert the space into a *Dijkstra* distance space $\langle \min, + \rangle$.

$$\begin{tikzcd}
D \arrow[d, "\min\circ g"'] \arrow[r, shift right, "(\varphi_1^D)^{-1}"'] & P \arrow[l, shift right, "\varphi_1^D"'] \arrow[d, "\max\circ\wedge"'] \arrow[r, shift left, "\varphi"] & D \arrow[l, shift left, "\varphi^{-1}"] \arrow[d, "\min\circ +"] \\
D^{\infty} \arrow[r, shift right, "(\varphi_1^D)^{-1}"'] & P^{\infty} \arrow[l, shift right, "\varphi_1^D"'] \arrow[r, shift left, "\varphi"] & D^{\infty} \arrow[l, shift left, "\varphi^{-1}"]
\end{tikzcd}$$

Figure 2.5: Isomorphism between any shortest path distance space $\langle \min, g\rangle$ and *Dijkstra* distance space $\langle \min, +\rangle$

In this framework, if one can starts with a proximity matrix, P, then the choice of the desired φ is in fact the choice of how would path-lengths be computed (g) if we converted them to distance using φ_1^D. On the other hand, if one starts with a distance matrix D, then one needs to convert the distance space into the *Dijkstra* distance space using $\varphi \circ (\varphi_1^D)^{-1}$ and in that case, as before, the φ isomorphism leads to the fixation of what would be the path-length function g in the original distance space. Another option would be to compute Dijkstra using the function g instead of the sum but, since the most computationally efficient g is $+$, this would not be as fast as converting all distances to Dijkstra space $\langle \min, +\rangle$ using $\varphi \circ (\varphi_1^D)^{-1}$ in $\mathcal{O}(|E|)$ steps and then computing Dijkstra Algorithm, because the algorithm would require more than $\mathcal{O}(|E|)$ computations of g.

Remark 2.4.1. Although all the solutions before required the computation of the closure in order to compute the backbone, and the closure requires the solution of the *APSP* Problem, there is a method for computing the metric backbone that does not require the solution to the *APSP* Problem since it computes the backbone in a constructive manner. That algorithm is available in [16] and its time-complexity is dependent on the number of metric edges, with the worst case being equivalent to the APSP. However, in this work this algorithm is not used since in many cases we also wish to study the edges that don't belong to the backbone.

2.5 Parametric Families of Distance Backbones

Some features of the metric backbone, such as the preservation of community structure and SI epidemic dynamics, have already been studied in [7]. Moreover, in [35] it was already hinted that other backbones such as the Product backbone ($g \equiv (x+1)(y+1)-1$) or the Euclidean backbone ($g \equiv \sqrt{x^2+y^2}$) might also be of interest in the study of dynamics on complex networks. Bearing that in mind, and given the construction from Diagram 2.5, we can now consider an infinite range of backbones constructed from different families of T-Norms and their generators. The families presented here are a subset of well-known families that are described in [18]. This subset includes the T-Norms whose generators are distance functions as in Definition 2.3.2, however, there are other families that also have distance functions as generators but are not present in [18]. The construction of these families is settled on the following result, which allows for the creation of a T-Norm starting from the generator of another T-Norm.

Proposition 2.5.1. *Let* $\wedge$ *be an archimedean T-Norm with additive generator* $\varphi : [0,1] \to [0,+\infty]$*. If* φ *is continuous and* $\psi : [0,+\infty] \to [0,+\infty]$ *is a strictly increasing bijection, then the function* $\psi \circ \varphi : [0,1] \to [0,+\infty]$ *is an additive generator for some continuous archimedean T-Norm.*

Proof. Let's prove that $\psi\circ\varphi$ is an additive generator. Given that ψ is a stricly increasing bijection, it follows that $\psi(0) = 0$ and $\psi(+\infty) = +\infty$ and that ψ must be continuous. Thus, $\psi \circ \varphi(1) = \psi(\varphi(1)) = \psi(0) = 0$. Also, since φ is continuous, we obtain $\psi \circ \varphi$ is continuous. Lastly, $x > y \Rightarrow \varphi(x) < \varphi(y) \Rightarrow \psi(\varphi(x)) < \psi(\varphi(y))$ and thus ψ is strictly decreasing. Clearly, $\psi \circ \varphi(x) + \psi \circ \varphi(y) \in [0,+\infty]$, so $\psi \circ \varphi$ is an additive generator for some T-Norm. □

2.5.1 Dombi

In [36] it was already made clear that the metric backbone used a T-Norm that is a special case of a broader family of T-Norms called the Dombi family, named after the hungarian computer scientist who first presented this family [11].

$$\begin{aligned}
\textbf{Additive Decreasing Generator: } \varphi^{\mathsf{D}}_{\lambda}(p) &= \left(\frac{1}{p} - 1\right)^{\lambda} \\
\textbf{T-Norm: } \wedge^{\mathsf{D}}_{\lambda}(x,y) &= (\varphi^{\mathsf{D}}_{\lambda})^{-1} \circ + \circ \varphi^{\mathsf{D}}_{\lambda} \\
&= (\varphi^{\mathsf{D}}_{\lambda})^{-1}(\varphi^{\mathsf{D}}_{\lambda}(x) + \varphi^{\mathsf{D}}_{\lambda}(y)) \\
&= \frac{1}{\left[\left(\frac{1}{x} - 1\right)^{\lambda} + \left(\frac{1}{y} - 1\right)^{\lambda}\right]^{\frac{1}{\lambda}} + 1} \\
\textbf{T-Norm Limits: } \wedge^{\mathsf{D}}_{0}(x,y) &= \lim_{\lambda\to 0} \wedge^{\mathsf{D}}_{\lambda}(x,y) = \wedge_{\textsf{Drastic}}(x,y) \\
\wedge^{\mathsf{D}}_{1}(x,y) &= \lim_{\lambda\to 1} \wedge^{\mathsf{D}}_{\lambda}(x,y) = \frac{xy}{x+y-xy} \\
\wedge^{\mathsf{D}}_{+\infty}(x,y) &= \lim_{\lambda\to +\infty} \wedge^{\mathsf{D}}_{\lambda}(x,y) = \min(x,y) \\
\textbf{TD-Norm: } g^{\mathsf{D}}_{\lambda}(x,y) &= \varphi^{\mathsf{D}}_{1} \circ \wedge^{\mathsf{D}}_{\lambda} \circ (\varphi^{\mathsf{D}}_{1})^{-1} \\
&= \varphi^{\mathsf{D}}_{1}(\wedge^{\mathsf{D}}_{\lambda}((\varphi^{\mathsf{D}}_{1})^{-1}(x), (\varphi^{\mathsf{D}}_{1})^{-1}(y))) \\
&= \left(x^{\lambda} + y^{\lambda}\right)^{\frac{1}{\lambda}}
\end{aligned}$$

2.5.2 Aczél-Alsina

This family is described in [18] with the name *Schweizer & Sklar 3* because the first version of the general T-Norm formula was presented in [33] by Schweizer and Sklar, although that version did not have a parameter λ, and thus, was simply defined for $\lambda = 1$. Nevertheless, this family was more thoroughly described by János Aczél and Claudi Alsina in [1], and for that reason, like in other references, we choose to adopt the name of these authors. The main relevance of this family lays in the fact that it generalizes the Product T-Norm, that is its simplified T-Norm when the parameter $\lambda = 1$. From Proposition 2.5.1, it follows that in the case of the Dombi and Aczél-Alsina families, the ψ function is $\psi(x) = x^{\lambda}$.

$$\textbf{Additive Decreasing Generator: } \varphi_{\lambda}^{\mathsf{AA}}(p) = (-\log(p))^{\lambda}$$

$$\begin{aligned}
\textbf{T-Norm: } \wedge_{\lambda}^{\mathsf{AA}}(x,y) &= (\varphi_{\lambda}^{\mathsf{AA}})^{-1} \circ + \circ \varphi_{\lambda}^{\mathsf{AA}} \\
&= (\varphi_{\lambda}^{\mathsf{AA}})^{-1}(\varphi_{\lambda}^{\mathsf{AA}}(x) + \varphi_{\lambda}^{\mathsf{AA}}(y)) \\
&= e^{-[(-\log(x))^{\lambda} + (-\log(y))^{\lambda}]^{\frac{1}{\lambda}}} \\
\textbf{T-Norm Limits: } \wedge_{0}^{\mathsf{AA}}(x,y) &= \lim_{\lambda\to 0} \wedge_{\lambda}^{\mathsf{AA}}(x,y) = \wedge_{\mathsf{Drastic}}(x,y) \\
\wedge_{1}^{\mathsf{AA}}(x,y) &= \lim_{\lambda\to 1} \wedge_{\lambda}^{\mathsf{AA}}(x,y) = xy \\
\wedge_{+\infty}^{\mathsf{AA}}(x,y) &= \lim_{\lambda\to +\infty} \wedge_{\lambda}^{\mathsf{AA}}(x,y) = \min(x,y) \\
\textbf{TD-Norm: } g_{\lambda}^{\mathsf{AA}}(x,y) &= \varphi_{1}^{\mathsf{D}} \circ \wedge_{\lambda}^{\mathsf{AA}} \circ (\varphi_{1}^{\mathsf{D}})^{-1} \\
&= \varphi_{1}^{\mathsf{D}}(\wedge_{\lambda}^{\mathsf{AA}}((\varphi_{1}^{\mathsf{D}})^{-1}(x), (\varphi_{1}^{\mathsf{D}})^{-1}(y)) \\
&= e^{\left[\left(-\log\left(\frac{1}{x+1}\right)\right)^{\lambda} + \left(-\log\left(\frac{1}{y+1}\right)\right)^{\lambda}\right]^{\frac{1}{\lambda}}} - 1
\end{aligned}$$

2.5.3 Frank

In the case of Frank T-Norms, these were created when M. J. Frank was characterizing all continuous associative T-Norms that obeyed some particular conditions. Of that work the Frank T-Norms resulted as the building blocks that described those particular associative functions [17].

$$\textbf{Additive Decreasing Generator: } \varphi_{\lambda}^{\mathsf{F}}(p) = -\log\left(\frac{\lambda^{p}-1}{\lambda-1}\right)$$

$$\begin{aligned}
\textbf{T-Norm: } \wedge_{\lambda}^{\mathsf{F}}(x,y) &= (\varphi_{\lambda}^{\mathsf{F}})^{-1} \circ + \circ \varphi_{\lambda}^{\mathsf{F}} \\
&= (\varphi_{\lambda}^{\mathsf{F}})^{-1}(\varphi_{\lambda}^{\mathsf{F}}(x) + \varphi_{\lambda}^{\mathsf{F}}(y)) \\
&= \frac{\log[1 + \frac{(\lambda^{x}-1)(\lambda^{y}-1)}{\lambda-1}]}{\log(\lambda)}
\end{aligned}$$

$$\begin{aligned}
\textbf{T-Norm Limits: } \wedge^{\mathsf{F}}_{0}(x,y) &= \lim_{\lambda\to 0} \wedge^{\mathsf{F}}_{\lambda}(x,y) = \min(x,y)\\
\wedge^{\mathsf{F}}_{1}(x,y) &= \lim_{\lambda\to 1} \wedge^{\mathsf{F}}_{\lambda}(x,y) = xy\\
\wedge^{\mathsf{F}}_{+\infty}(x,y) &= \lim_{\lambda\to +\infty} \wedge^{\mathsf{F}}_{\lambda}(x,y) = \max(0, x+y-1)\\
\textbf{TD-Norm: } g^{\mathsf{F}}_{\lambda}(x,y) &= \varphi^{\mathsf{D}}_{1} \circ \wedge^{\mathsf{F}}_{\lambda} \circ (\varphi^{\mathsf{D}}_{1})^{-1}\\
&= \varphi^{\mathsf{D}}_{1}(\wedge^{\mathsf{F}}_{\lambda}((\varphi^{\mathsf{D}}_{1})^{-1}(x), (\varphi^{\mathsf{D}}_{1})^{-1}(y)))\\
&= \frac{\log(\lambda)}{\log\left(\frac{\lambda+\lambda^{\frac{1}{x+1}}\lambda^{\frac{1}{y+1}}-\lambda^{\frac{1}{x+1}}-\lambda^{\frac{1}{y+1}}}{\lambda-1}\right)} - 1
\end{aligned}$$

In the Frank family, the additive generator corresponding to $\lambda = 1$ does not exist since

$$\varphi^{\mathsf{F}}_{1}(p) = -\log\left(\frac{1^p-1}{1-1}\right) = -\log\left(\frac{0}{0}\right).$$

However, by fixing p as a constant and using the L'Hôpital Rule, we can obtain the following limit,

$$\lim_{\lambda\to 1}\frac{\lambda^p-1}{\lambda-1} = \lim_{\lambda\to 1}\frac{\frac{d}{d\lambda}(\lambda^p-1)}{\frac{d}{d\lambda}(\lambda-1)} = \lim_{\lambda\to 1}\frac{p\lambda^{p-1}}{1} = \lim_{\lambda\to 1} p\lambda^{p-1} = p1^{p-1} = p.$$

And now we can fix $\varphi^{\mathsf{F}}_{1}(p)$ as being the corresponding limit

$$\lim_{\lambda\to 1}\varphi^{\mathsf{F}}_{\lambda}(p) = \lim_{\lambda\to 1} -\log\left(\frac{\lambda^p-1}{p-1}\right) = -\log(p).$$

2.5.4 Hamacher

Like in the Frank family, the Hamacher family builds upon the product T-Norm to form a family of T-Norms using a ψ function not so intuitive as the one of Dombi and Aczél-Alsina T-Norms. This family was developed by Hamacher in 1975 and 1978 when he was trying to characterize continuous T-Norms that are rational functions (quotients of polynomials). Indeed, his main result about that ended up being that any continuous T-Norm is a rational function if and only if it belongs to the Hamacher family [17].

$$\begin{aligned}
\textbf{Additive Decreasing Generator: } \varphi^{\mathsf{H}}_{\lambda}(p) &= -\log\left(\frac{p}{\lambda+(1-\lambda)p}\right)\\
\textbf{T-Norm: } \wedge^{\mathsf{H}}_{\lambda}(x,y) &= (\varphi^{\mathsf{H}}_{\lambda})^{-1} \circ + \circ \varphi^{\mathsf{H}}_{\lambda}\\
&= (\varphi^{\mathsf{H}}_{\lambda})^{-1}(\varphi^{\mathsf{H}}_{\lambda}(x) + \varphi^{\mathsf{H}}_{\lambda}(y))\\
&= \frac{xy}{\lambda+(1-\lambda)(x+y-xy)}
\end{aligned}$$

$$\begin{aligned}
\textbf{T-Norm Limits: } \wedge^{\mathsf{H}}_{0}(x,y) &= \lim_{\lambda\to 0} \wedge^{\mathsf{H}}_{\lambda}(x,y) = \frac{xy}{x+y-xy} \\
\wedge^{\mathsf{H}}_{1}(x,y) &= \lim_{\lambda\to 1} \wedge^{\mathsf{H}}_{\lambda}(x,y) = xy \\
\wedge^{\mathsf{H}}_{+\infty}(x,y) &= \lim_{\lambda\to +\infty} \wedge^{\mathsf{H}}_{\lambda}(x,y) = \wedge_{\mathsf{Drastic}}(x,y) \\
\textbf{TD-Norm: } g^{\mathsf{H}}_{\lambda}(x,y) &= \varphi^{\mathsf{D}}_{1} \circ \wedge^{\mathsf{H}}_{\lambda} \circ (\varphi^{\mathsf{D}}_{1})^{-1} \\
&= \varphi^{\mathsf{D}}_{1}(\wedge^{\mathsf{H}}_{\lambda}((\varphi^{\mathsf{D}}_{1})^{-1}(x), (\varphi^{\mathsf{D}}_{1})^{-1}(y))) \\
&= x + y + \lambda xy
\end{aligned}$$

2.5.5 Schweiser & Sklar 4

This family of T-Norms has the particularity of coinciding with the Dombi Family when $\lambda = 1$. This, however, is not a coincidence since the generator of these families is essentially the same but with the parameter λ being an exponent for the proximity value instead of the whole expression. This slight difference translates into a very different family of generator functions as can be attested by the Figure A.12. In particular, one can state that the family SS4 is dense over the function space $[0,+\infty]^{[0,1]}$, in the sense that for every $(a,b) \in]0,1[\times]0,+\infty[$ there exists a $\lambda \in]0,+\infty[$ such that $\varphi^{\mathsf{SS4}}_{\lambda}(a) = b$, which is not true for the Dombi Family.

$$\begin{aligned}
\textbf{Additive Decreasing Generator: } \varphi^{\mathsf{SS4}}_{\lambda}(p) &= \frac{1}{p^{\lambda}} - 1 \\
\textbf{T-Norm: } \wedge^{\mathsf{SS4}}_{\lambda}(x,y) &= (\varphi^{\mathsf{SS4}}_{\lambda})^{-1} \circ + \circ \varphi^{\mathsf{SS4}}_{\lambda} \\
&= (\varphi^{\mathsf{SS4}}_{\lambda})^{-1}(\varphi^{\mathsf{SS4}}_{\lambda}(x) + \varphi^{\mathsf{SS4}}_{\lambda}(y)) \\
&= \frac{xy}{(x^{\lambda} + y^{\lambda} - x^{\lambda}y^{\lambda})^{\frac{1}{\lambda}}} \\
\textbf{T-Norm Limits: } \wedge^{\mathsf{SS4}}_{0}(x,y) &= \lim_{\lambda\to 0} \wedge^{\mathsf{SS4}}_{\lambda}(x,y) = xy \\
\wedge^{\mathsf{SS4}}_{1}(x,y) &= \lim_{\lambda\to 1} \wedge^{\mathsf{SS4}}_{\lambda}(x,y) = \frac{xy}{x+y-xy} \\
\wedge^{\mathsf{SS4}}_{+\infty}(x,y) &= \lim_{\lambda\to +\infty} \wedge^{\mathsf{SS4}}_{\lambda}(x,y) = \min(x,y) \\
\textbf{TD-Norm: } g^{\mathsf{SS4}}_{\lambda}(x,y) &= \varphi^{\mathsf{D}}_{1} \circ \wedge^{\mathsf{SS4}}_{\lambda} \circ (\varphi^{\mathsf{D}}_{1})^{-1} \\
&= \varphi^{\mathsf{D}}_{1}(\wedge^{\mathsf{SS4}}_{\lambda}((\varphi^{\mathsf{D}}_{1})^{-1}(x), (\varphi^{\mathsf{D}}_{1})^{-1}(y))) \\
&= \left(\frac{1}{\left(\frac{1}{x+1}\right)^{-\lambda} + \left(\frac{1}{y+1}\right)^{-\lambda} - 1} \right)^{-\frac{1}{\lambda}} - 1 \\
&= \left[(x+1)^{\lambda} + (y+1)^{\lambda} - 1\right]^{\frac{1}{\lambda}} - 1
\end{aligned}$$

Remark 2.5.1. Another important trait, specially relevant for these families of T-Norms, is the (pointwise) comparability of T-Norms. In that sense, we say that the T-Norms $\wedge_1$ and $\wedge_2$ are (pointwisely) comparable if $\wedge_1(x,y) \leq \wedge_2(x,y)$ for all $(x,y) \in [0,1]^2$ or $\wedge_2(x,y) \leq \wedge_1(x,y)$ for all $(x,y) \in [0,1]^2$. In [17], several conditions are presented that assure the comparability of two continuous archimedean T-Norms based on their additive generator. Also, it is stated that for every $\lambda_1, \lambda_2 \in]0, +\infty[$ with $\lambda_1 < \lambda_2$:

- The **Dombi family is strictly increasing**, $\wedge^{\mathsf{D}}_{\lambda_1}(x, y) < \wedge^{\mathsf{D}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, 1]^2$
- The **Aczél-Alsina family is strictly increasing**, $\wedge^{\mathsf{AA}}_{\lambda_1}(x, y) < \wedge^{\mathsf{AA}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, 1]^2$
- The **Frank family is strictly decreasing**, $\wedge^{\mathsf{F}}_{\lambda_1}(x, y) > \wedge^{\mathsf{F}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, 1]^2$
- The **Hamacher family is strictly decreasing**, $\wedge^{\mathsf{H}}_{\lambda_1}(x, y) > \wedge^{\mathsf{H}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, 1]^2$

Furthermore, given that the distance function isomorphism is strictly decreasing, we get that for $\lambda_1, \lambda_2 \in]0, +\infty[$ with $\lambda_1 < \lambda_2$:

- $g^{\mathsf{D}}_{\lambda_1}(x, y) > g^{\mathsf{D}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, +\infty]^2$
- $g^{\mathsf{AA}}_{\lambda_1}(x, y) > g^{\mathsf{AA}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, +\infty]^2$
- $g^{\mathsf{F}}_{\lambda_1}(x, y) < g^{\mathsf{F}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, +\infty]^2$
- $g^{\mathsf{H}}_{\lambda_1}(x, y) < g^{\mathsf{H}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, +\infty]^2$.

This allows to conclude that the backbones of each family form a chain of sets. Indeed, for every $\lambda_1, \lambda_2 \in]0, +\infty[$ with $\lambda_1 < \lambda_2$ we have:

- $B^{g_{\mathsf{Drastic}}} = B^{g^{\mathsf{D}}_0} \supseteq \cdots \supseteq B^{g^{\mathsf{D}}_{\lambda_1}} \supseteq \cdots \supseteq B^{g^{\mathsf{D}}_{\lambda_2}} \supseteq \cdots \supseteq B^{g^{\mathsf{D}}_{+\infty}} = B^{um}$
- $B^{g_{\mathsf{Drastic}}} = B^{g^{\mathsf{AA}}_0} \supseteq \cdots \supseteq B^{g^{\mathsf{AA}}_{\lambda_1}} \supseteq \cdots \supseteq B^{g^{\mathsf{AA}}_{\lambda_2}} \supseteq \cdots \supseteq B^{g^{\mathsf{AA}}_{+\infty}} = B^{um}$
- $B^{um} = B^{g^{\mathsf{F}}_0} \subseteq \cdots \subseteq B^{g^{\mathsf{F}}_{\lambda_1}} \subseteq \cdots \subseteq B^{g^{\mathsf{F}}_{\lambda_2}} \subseteq \cdots \subseteq B^{g^{\mathsf{F}}_{+\infty}}$
- $B^{m} = B^{g^{\mathsf{H}}_0} \subseteq \cdots \subseteq B^{g^{\mathsf{H}}_{\lambda_1}} \subseteq \cdots \subseteq B^{g^{\mathsf{H}}_{\lambda_2}} \subseteq \cdots \subseteq B^{g^{\mathsf{H}}_{+\infty}} = B^{g_{\mathsf{Drastic}}}$.

Moreover, in [17] there is another result which gives a sufficient condition for the comparability of two T-Norms and is stated as follows.

Proposition 2.5.2. *Let $\wedge_1$ and $\wedge_2$ be continuous archimedean T-Norms with differentiable generators $\varphi_1, \varphi_2 : [0, 1] \to [0, +\infty]$, respectively. If the function*

$$\frac{(\varphi_1)'}{(\varphi_2)'} :]0, 1[\to [0, +\infty[$$

is non-decreasing, then we have $\wedge_1(x, y) \leq \wedge_2(x, y)\ \forall (x, y) \in [0, 1]^2$.

Given this result, we can prove the monotonicity of the family Schweizer & Sklar 4, since:

$$\frac{(\varphi^{\mathsf{SS4}}_{\lambda_1})'}{(\varphi^{\mathsf{SS4}}_{\lambda_2})'} = \frac{(\frac{1}{x^{\lambda_1}} - 1)'}{(\frac{1}{x^{\lambda_2}} - 1)'} = \frac{-\lambda_1 x^{-\lambda_1 - 1}}{-\lambda_2 x^{-\lambda_2 - 1}} = \frac{\lambda_1}{\lambda_2} x^{-\lambda_1 - 1 + \lambda_2 + 1} = \frac{\lambda_1}{\lambda_2} x^{\lambda_2 - \lambda_1}$$

and this function is clearly non-decreasing whenever $\lambda_1 < \lambda_2$. Hence, the **Schweizer & Sklar 4 family is increasing**, $\wedge^{\mathsf{SS4}}_{\lambda_1}(x, y) \leq \wedge^{\mathsf{SS4}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, 1]^2$ if $\lambda_1 < \lambda_2$. Then, we extend this inequality to the TD-Norms and the backbones as $g^{\mathsf{SS4}}_{\lambda_1}(x, y) \geq g^{\mathsf{SS4}}_{\lambda_2}(x, y)\ \forall (x, y) \in [0, +\infty]^2$ and $B^{g_{\mathsf{Prod}}} = B^{g^{\mathsf{SS4}}_0} \supseteq \cdots \supseteq B^{g^{\mathsf{SS4}}_{\lambda_1}} \supseteq \cdots \supseteq B^{g^{\mathsf{SS4}}_{\lambda_2}} \supseteq \cdots \supseteq B^{g^{\mathsf{SS4}}_{+\infty}} = B^{um}$.

2.6 Spreading Dynamics on Networks

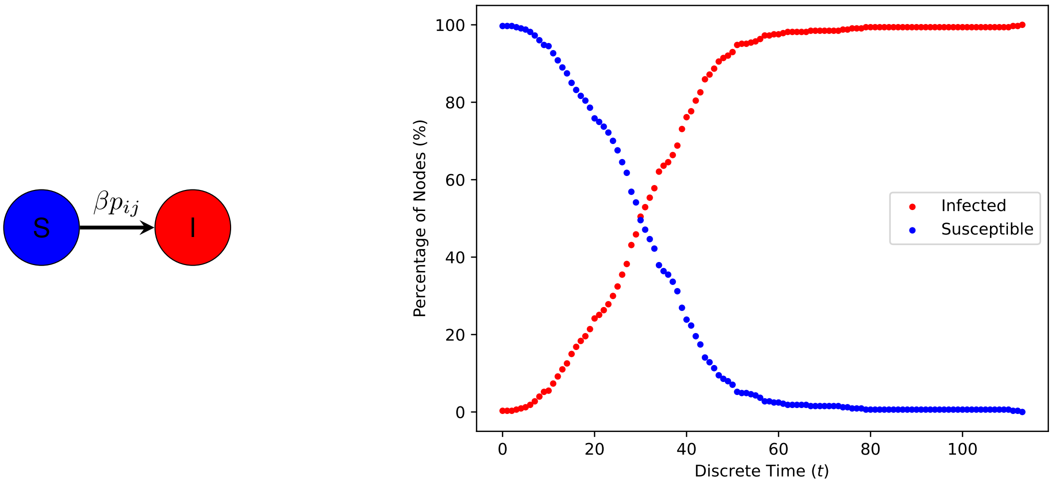
There are several types of dynamics over networks that one can study. In this work we focus in dynamics on networks, which is intended as some sort of dynamic property of the nodes in a network that changes over time, as opposed to dynamics of networks, which is related to connectivity changes in a network over time. More specifically, we deal with spreading dynamics that concern the spreading of a node state through a network, taking into account the connectivity and connection weights in that network and also some global parameters that regulate that spreading. The most well-known spreading dynamics models that encompass all these characteristics come from the field of Epidemiology and are known as Compartmental Models. Although there are many possible models in this category [27], this work will focus primarily on two: Susceptible-Infected (SI) and on Susceptible-Infected-Recovered (SIR) models.

2.6.1 Susceptible-Infected (SI)

In Susceptible-Infected Models, it is assumed that nodes can be either in *susceptible* state or in *infected* state and once a node turns infected, it does not go back to susceptible, unlike in Susceptible-Infected-Susceptible models. For that reason, the infected state is said to be a *sink* state. This model works by starting with an infected seed node and iterating at each timestep t by infecting the neighbors of already infected nodes with probability βp_{ij} where p_{ij} is the proximity between two given nodes x_i, x_j and $\beta \in [0, +\infty[$ is a global parameter that contracts or dilates the speed at which the epidemic spreads over time. If $\beta = 0$ there is no epidemic because the only infected node will be the initial one, and if $\beta = 1$ then the proximities in each edge dictate the exact probability of contagion between those two nodes. Moreover, if $\beta > 1$, we produce an accelerated epidemic in the sense that more nodes will get infected earlier.

Given that, if t is allowed to be large enough, SI models always produce a full population of infected nodes, a good pair of global metrics that measure its evolution is the pair $\langle t_{half}, t_{all} \rangle$ where t_{half} denotes the first timestep at which at least half of the nodes were infected and t_{all} denotes the first timestep where all nodes were infected. This fact elucidates the construction of Algorithm B.2, used in this work, where the output is the percentages of nodes infected at each step t. That information is sufficient for the computation of t_{half} and t_{all} and also provides a way to visualize the evolution of a given simulation of the epidemic over time, since with this information we can also infer the percentage of nodes that are still *susceptible* at each step t by considering $\texttt{PercSusceptible}[t] = 100 - \texttt{PercInfected}[t]$.

The example simulation presented next, in particular, illustrates a case where the end of the epidemic happens in an almost plateau growth. This might be due to the border nodes in the network taking more time to get infected which can be caused by the edges connecting those nodes to the rest of the network being relatively weak, i.e. having low proximities.



(a) State Transition

(b) Simulation with $\beta = 1$

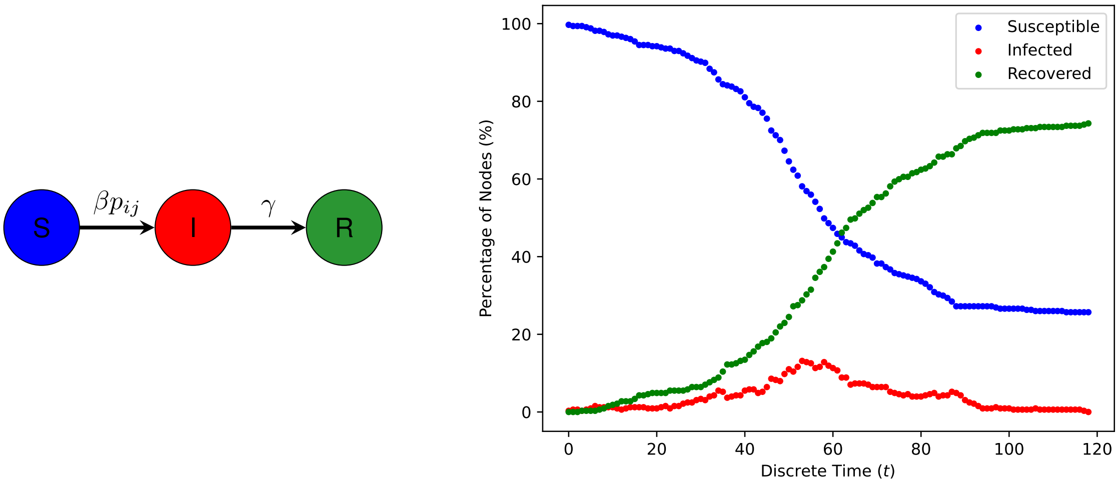
Figure 2.6: Local and global description of SI Dynamics

2.6.2 Susceptible-Infected-Recovered (SIR)

In Susceptible-Infected-Recovered Models, the infection procedure occurs in a similar way to SI but at the end of every iteration/timestep t an infected node might recover from the infection with probability γ. Thus, $\gamma \in [0, 1]$ is a global parameter, that is, it is the same for every node of the network. Thus, the parameter β has the same function as in the SI model and the γ controls the rate at which any node recovers after being *infected*. In this particular model it is assumed that, once a node is *recovered*, it no longer changes its state, and so the recovered state acts as a *sink* state.

In the case of the SIR Algorithm B.3 available in Appendix B, the outputs are similar to the SI Algorithm B.2. However, in this case, in order to construct the graph in Figure 2.7(b), one also needs to save the information of the percentage of recovered nodes at each time t and then calculate the percentage of susceptible nodes at each time t analogously (assuming no deaths in the population).

Nevertheless, these outputs are only useful for the visualization of the spreading evolution because, in the case of SIR, for sufficiently lower values of γ compared to β, not all nodes will have been infected at the end of each simulation.



(a) State Transitions

(b) Simulation with $\beta = 1$ and $\gamma = 1/6$

Figure 2.7: Local and global description of SIR Dynamics

3

Related Work

Contents

Given the growing need to analyse large graphs, the development of methods to reduce the size of these graphs while preserving their main properties is of great importance. In this work, we have proposed a framework to extract the backbone of a network based on the concept of *distance* between nodes and in that framework it is guaranteed that the shortest-paths between nodes are preserved. However in many other cases the shortest-paths are not the most relevant feature in a network, and therefore, there are other methods that aim at reducing the size of graphs while preserving different properties.

3.1 Sampling Subgraphs

One broad class of sparsification methods is the one of **sampling subgraphs**. These methods choose a subset from the original graph, whether by picking nodes, edges, or both. As described in [20], in general, when sparsifying a graph by sampling methods, there are a few questions that need to be addressed such as what sample size is best and how to measure the quality of a sample and the quality of a sampling method in terms of preserving original properties of the graph. However, many of these questions do not have straightforward answers and depend on the specific application of the sampling method. One of these sampling methods, which is used in this work, is **random sampling**. Amongst sampling methods this method is perhaps the simplest one, since it does not depend on any parameter besides the size of the sample. In this method, the sampling universe is usually the set of edges but it can also be the set of nodes. Consequently, in the case of sampling nodes, the outcome graph will be disconnected and in the case of edges, depending on the sample size, the sampled graph might also not be connected since we may remove bridge edges. Thus, in order to study spreading phenomena, for example, it is advisable to consider several samples of a given size and then average the results.

3.2 Thresholding Edge Weights

Another sparsification method is the **thresholding of edge weights**. In this method, the edges are sorted by their weight and then the strongest weight edges are chosen until a certain threshold (or subgraph size) has been reached. Unlike distance backbones, this method only takes into consideration local information of the network, since the only thing that matters for an edge to be in the sparsification is its original weight. This reduction, however, might not preserve the connectivity of the original graph, and it certainly does not preserve the distribution of edge weights, nor the shortest-paths between nodes. Nevertheless, in many scenarios, there are graphs with very weak edges that do not account for much in a particular dynamic or process and can be ignored when making analysis of that graph. However, as will be demonstrated in the experimental results, in the case of contact social networks, many times there are weak edges between nodes that play an important role in the spreading of infections. Therefore, thresholding edge weights is not a practical method for reducing the complexity of contact social networks in the context of epidemiology because it could remove essential edges that play a critical role in the transmission of infections.

3.3 Transitive Reduction

In Graph Theory, the **transitive reduction** of a graph G is the smallest possible graph that has the same transitive closure as G. In other words, the transitive reduction graph is the smallest graph that has the same reachability of the original graph. This method has a close connection with the one of Transitive Closure, however this one does not always achieve a unique subgraph unlike the Transitive Closure that allows for the construction of a unique backbone. When the graph G has cycles, its transitive reduction might not be unique and might even contain edges that do not belong to the original graph. This marks a difference between the transitive reduction and another concept which is the **minimum equivalent graph**, whereas the minimum equivalent subgraph also is a graph with the same reachability of G but it must also be a subgraph of G. Beyond that, the backbone itself might not be the smallest subgraph with the same transitive closure since it might contain edges between two nodes with the same weight as another path connecting the same nodes. Although mostly developed for directed unweighted graphs [2], the transitive reduction has been expanded for weighted graphs as well as general fuzzy graphs [8]. In the case of directed unweighted graphs it has been shown that the best algorithm to compute a *canonical* transitive reduction has the same time complexity than computing the transitive closure [2], however, as noted in Remark 2.4.1, the best time-complexity to compute the distance backbone can be lower than the transitive closure.

3.4 Minimum Spanning Tree (MST)

The minimum spanning tree of a graph G is the tree with the smallest weight that spans the graph G, where the weight of a tree is the sum of the weights in the edges of the tree. Each minimum spanning tree has exactly $n - 1$ edges, where n is the number of nodes in G. Moreover, it is possible for a graph to have multiple MSTs, but in the case that all the edge weights are different it is guaranteed to exist only one MST. The concept of MST is connected to the distance backbone framework because the Ultra-Metric Backbone is the union of all Minimum Spanning Trees of a weighted graph [29]. Although this kind of sparsification is useful in many real-world scenarios, it might fall short in contexts where the reduction in the number of edges should not be so drastic. Also, unlike in distance closures, this method does not distinguish different edges that are not in the MST, whereas in the framework of distance closures, the semi-triangular edges can be compared through the semi-triangular distortion.

3.5 Multiscale Backbone

A more recent approach to the sparsification of complex networks was presented in [45] and is called the **multiscale backbone**. This method assumes that there is a null model to which a weighted graph can be compared. In that null-model, given a node with degree k, the probability density function of a normalized edge weight from that node taking value $x \in [0, 1]$ is given by $F_k(x) = \int_0^x (k-1)(1-x)^{k-2} dx$. This assumption builds up to the creation of the **disparity filter** that chooses the edges that obey the inequality $\alpha_{ij} = 1-(k-1) \int_0^{p_{ij}} (1-x)^{k-2} < \alpha \Leftrightarrow \alpha_{ij} = 1-F_k(p_{ij}) < \alpha$. In other words, an edge is chosen if the probability of it having a larger normalized weight than it has in the null-model is less than the significance level α. The multiscale backboone is then constructed by picking the edges that satisfy that criterion for at least one of the nodes. In the case of nodes with degree 1 connected with nodes of degree bigger than 1, the edge is kept only if it satisfies the criterion for the largest degree node. Unlike distance backbones, this method relies on a null model and a significance level, which makes its sparsification less versatile than the distance backbone that is a purely algebraic construction transversely applicable on every weighted graph and with no assumptions. Moreover, this method allows for the removal of all the edges that connect a single node, and thus, can create islands which, in practise, means that it can erase nodes.

3.6 Effective Resistance Thresholding

Another graph sparsification scheme that has been often used is the **effective resistance thresholding**. This scheme relies on the analogy that graphs can be viewed as electrical networks and the resistance between each pair of nodes can be calculated with respect to the network topology and weights. The effective resistance of any edge in a weighted graph can be computed by $R^e_{ij} = (e_i - e_j)^\top L^\dagger (e_i - e_j)$ where L is the Laplacian matrix of the graph and e_i, e_j are elements of the canonical basis [28]. The Laplacian matrix is defined as $D - A$, where D is the diagonal matrix with the node degrees and A is the adjacency matrix. Thus, the effective resistance between any two nodes, retains information about all the possible paths connecting them, which is a feature not always shared by other sparsification methods. In the end, the sparsification can be done by either thresholding directly the effective resistances of the edges or assigning a probability of picking a given edge, p^R_{ij} , such that $p^R_{ij} \sim p_{ij} R^e_{ij}$. and then sample edges according to those probabilities [28]. This scheme requires the computation of the Moore-Penrose pseudo-inverse of the Laplacian matrix, $L^\dagger$, which is often a very difficult task especially for very large networks. Thus, some approximation algorithm must be used [25].

4

Experimental Results

Contents

4.1 Description of Networks Studied

The networks analysed in this chapter come from publicly available datasets in the SocioPatterns database (`http://www.sociopatterns.org/datasets`). These are social contact networks from 3 different environments. The datasets contain information about how many times two people have been in close contact in a predetermined time-frame, which allows for the construction of a network with adjacency matrix $R(X)$ where r_{ij} denotes the number of close contacts that nodes x_i, x_j had in that timeframe. The notion of close contact and the way that data is collected differs slightly from network to network but, in general, a close contact corresponds to an event where two nodes shared the same space for a period of $20s$. After these contact networks are constructed, the weights in the edges are converted to proximities using the Jaccard Measure, such that $p_{ij} = \frac{r_{ij}}{r_{ii}+r_{jj}-r_{ij}}$. These particular networks have been studied in previous published research articles where more information about them can be found. Nevertheless, we leave a brief summary of each network.

- Fr-HS [23] was collected in a High School from Marseille, Paris during a time-frame of 1 week in December 2013.
- Ir-Ex [15] was collected during an exhibit in Dublin, Ireland during a time-frame of 1 day. This dataset contained a network for each day that the exhibit was open, from April 17th to July 17th of 2009, but the network analysed here only concerns the day which had the largest number of attendees and contacts.
- Fr-Wo [6] was collected at the French Health Observatory Workplace in Paris, France during a time-frame of 2 weeks in 2015.

All these networks have also been studied in [7] where it was analysed whether their metric backbone maintained community structure and was a primary transmission subgraph for SI Dynamics.

Network	$\lvert X\rvert$	$\lvert E\rvert$	$\langle C(x_i)\rangle_{x_i\in X}$	δ	**Average Degree**	**Diameter**
French High-School	327	5818	0.0038	0.1092	35.5841	4
Exhibit	410	2765	0.0233	0.0330	13.4878	9
Workplace	217	4274	0.0100	0.1824	39.3917	5

Table 4.1: Summary of properties of the networks studied

Network	$\lvert B^{um}(X)\rvert(\tau^{um})$	$\lvert B^{e}(X)\rvert(\tau^{e})$	$\lvert B^{m}(X)\rvert(\tau^{m})$	$\lvert B^{p}(X)\rvert(\tau^{p})$
French High-School	$326(5.6\%)$	$426(7.32\%)$	$603(10.36\%)$	$4369(75.09\%)$
Exhibit	$413(14.94\%)$	$639(23.11\%)$	$1088(39.35\%)$	$2661(96.24\%)$
Workplace	$216(5.05\%)$	$376(8.80\%)$	$745(17.43\%)$	$4164(97.43\%)$

Table 4.2: Backbones sizes of the networks studied

4.2 Parametric Families and Distortion Analysis

As mentioned in Definition 2.3.22, when computing a backbone using a TD-Norm g, each edge is associated with a distortion value s^g_{ij} that measures how different is that edge distance from the shortest-path distance between those same two nodes. Therefore, there is a natural way of ordering edges according to how semi-triangular they are, by using the order of the distortion values. Moreover, in Remark 2.5.1 we establish the theoretical inclusions of the backbones from each family. Thus, it is expected that the respective backbones have growing or decreasing sizes. Therefore, the size of each backbone associated with parameter λ was computed for each family.

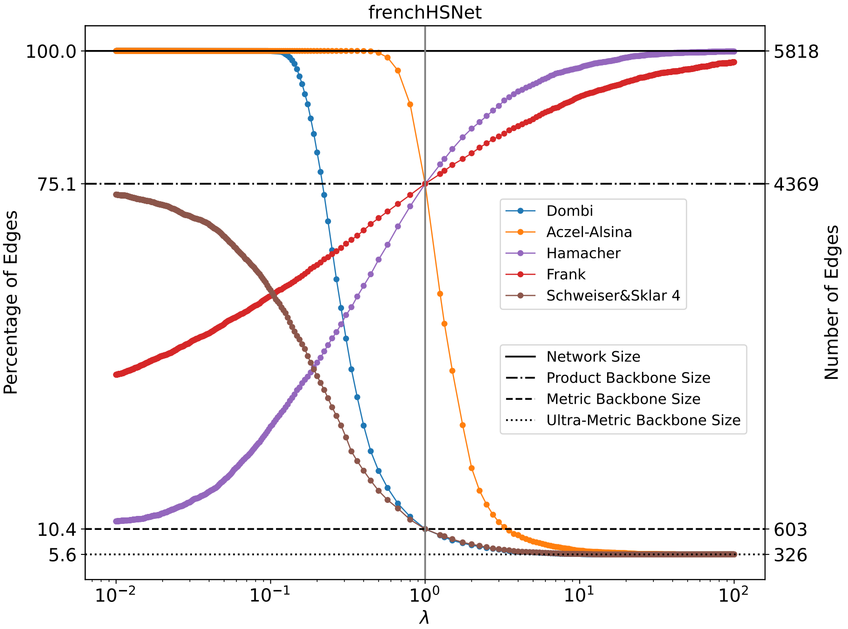


Figure 4.1: Families Backbones Sizes of the French High-School Network. Each point corresponds to the size of the backbone for the respective family and λ value. Since the computation of many backbones is computational expensive, the backbones are computed for $\lambda \in \{\frac{1}{n}, \frac{1}{n+0.25}, \frac{1}{n+0.5}, \frac{1}{n+0.75}, n, n+0.25, n+0.5, n+0.75 : n \in \mathbb{N}\} \cap [\frac{1}{100}, 100]$. Although the domain of λ is $[0, +\infty]$, the largest variation in the sizes occurs in $[\frac{1}{20}, 20]$ and so the figure only presents that subinterval.

Clearly, the backbones sizes of this network[1] obey the natural size order given by the backbones inclusions. Moreover, it is clear in this figure that the families Aczél-Alsina, Frank and Hamacher coincide in the product backbone when $\lambda = 1$ and the families Dombi and Schweizer & Sklar 4 coincide in the metric backbone when $\lambda = 1$. Given the inclusions in the backbones from each family, we can create an alternative ordering in the edges by considering the first λ at which each edge enters the backbone of the respective family. This order, however, is much more expensive to compute since we need to compute a very large amount of backbones to be able to define it. For this reason it is pertinent to compare

[1] The same plot for the other networks can be found in Appendix B

this order to other sortings of the edges that could hint at a simpler way to obtain it. In the case of the Dombi and Aczél-Alsina families, the backbones sizes span from the smallest ($|B^{um}|$) to the largest ($|B^{g_{\text{Drastic}}}|$) so we can compare the order provided by λ with the order provided by the distortion from the ultra-metric backbone and also with the order provided by the distortions from the λ_* closure, where the λ_* corresponds to the first λ that produces the ultra-metric backbone in the correspondent family. It's worthy to mention here that, although there are families that produce the ultra-metric backbone when λ is sufficiently larger (or smaller), the distortion values from those closures are not the same as the ones from the ultra-metric closure produced with $g \equiv \max(x, y)$.

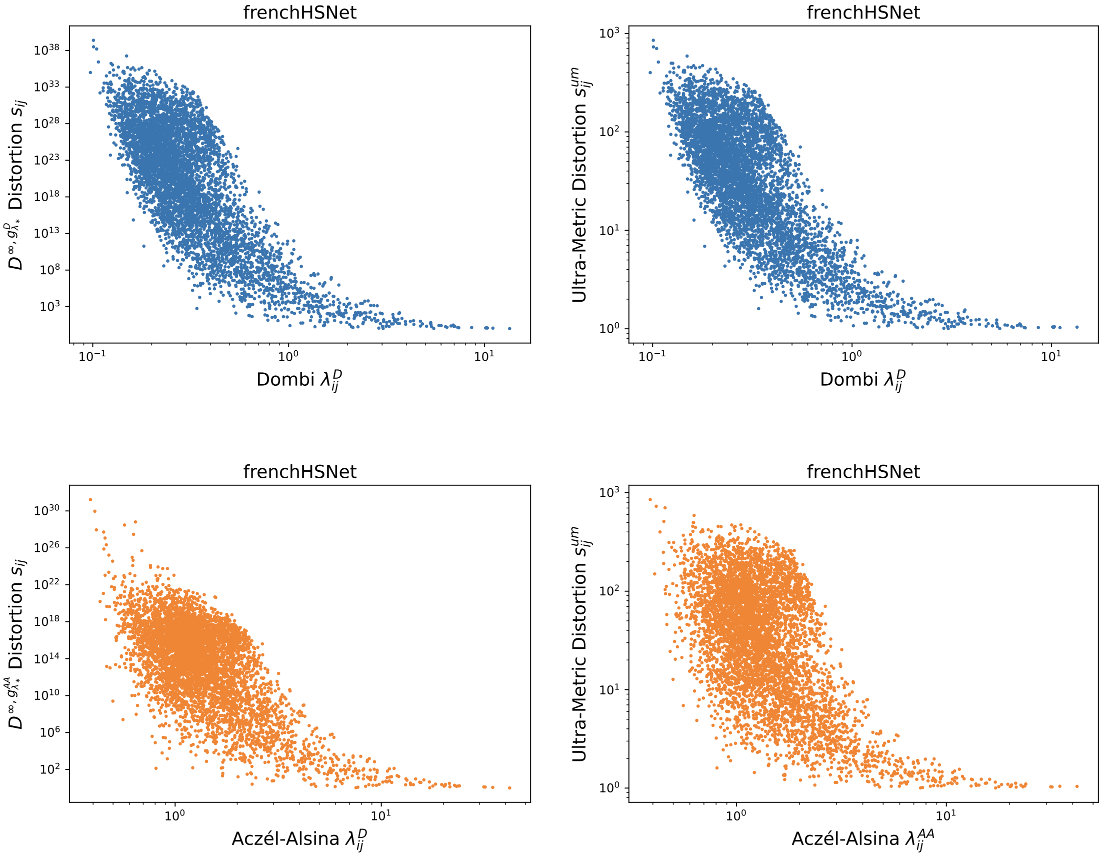


Figure 4.2: Relation between λ and distortions values in the French High-School Network. Each point in these figures corresponds to a non-ultrametric edge. The vertical axis measures the first λ at which that edge was added to the backbone in that family. The horizontal axis measures the distortion value (when it is not 1) whether it is from the closure associated with the $B^{g_{\lambda_*}}$ (left) or the B^{um} (right) backbone.

By inspection of these plots it is noticeable that none of the distortions can induce the same order of the edges as the λ since we don't have a strictly decreasing relation between them. However, it is clear that in this network[2], there is a general tendency for the larger λ backbones to add edges with higher

[2] The plots for the other networks can be found in Appendix A

distortions and the smaller λ backbones to add edges with smaller distortions. Nevertheless, there is a pronounced mixture of edges with very different distortions being added at the same λ backbones. This fact suggests a deeper investigation should be done in order to evaluate if there is some feature that can characterize different edges (and is manifested by the first λ at which they appear in the backbone) beyond the semi-triangular distortion.

4.3 SI Epidemics

Given that our main purpose is to compare how different subgraphs of a given network preserve the dynamics in SI epidemics, we need to establish how to do that comparison. Firstly, since we want to compare different sparsifications, it is easier to compare how the metrics measured in those sparsifications relate to the same metrics measured in the original unsparsified network. Thus, we want a reasonable way to compute $t/t^{D(X)}$ where t is the metric measured in the sparsified network and $t^{D(X)}$ is the metric measured in the original network. Given that some methods of sparsification require random sampling of edges, it is necessary to account for a vast set of random realizations N_R for each subgraph size. Also, since the starting seed node is important for the way the infection spreads in the network, it is also worthy to consider a set of initial seed nodes, S. Hence, for a given seed node $s \in S$ and network realization $n_r \in N_R$, we can create samples of N_S simulations with the corresponding t values (t_{half} or t_{all}):

$$\mathbf{t} = \{t_1, t_2, \ldots, t_{N_S}\} \quad \mathbf{t}^{D(X)} = \{t_1^{D(X)}, t_2^{D(X)}, \ldots, t_{N_S}^{D(X)}\}$$

Assuming $\langle \mathbf{t} \rangle$ and $\langle \mathbf{t}^{D(X)} \rangle$ are uncorrelated, we can use the error propagation formula for a quotient quantity based on the Taylor Series linearization [40] and write:

$$y_{s,n_r} = \frac{\langle \mathbf{t} \rangle}{\langle \mathbf{t}^{D(X)} \rangle} \quad \sigma^2_{y_{s,n_r}} = (y_{s,n_r})^2 \left(\frac{\sigma^2_{\mathbf{t}}}{\langle \mathbf{t} \rangle^2} + \frac{\sigma^2_{\mathbf{t}^{D(X)}}}{\langle \mathbf{t}^{D(X)} \rangle^2} \right)$$

Then, we extend these quantities by considering their values across all random network realizations and initial seed nodes:

$$\mathbf{y} = \langle y_{s,n_r} \rangle_{s \in S, n_r \in N_R} \quad \sigma_{\mathbf{y}} = \sqrt{\frac{\sum\limits_{s \in S, n_r \in N_R} \sigma^2_{y_{s,n_r}}}{|S| \times |N_R|}}$$

This way we obtain an expected value of t, $\mathbf{y}$, and its correspondent error $\sigma_{\mathbf{y}}$, for each subgraph size of a given type.

Considering again the French High-School Network[3], we can compare the sparsification of Ultra-Metric Backbone with two other sparsifications already mentioned in Chapter 3 by analysing the t_{half}

[3] The plots for the other networks can also be found in Appendix A

values and respective errors of SI dynamics along two sparsification procedures.

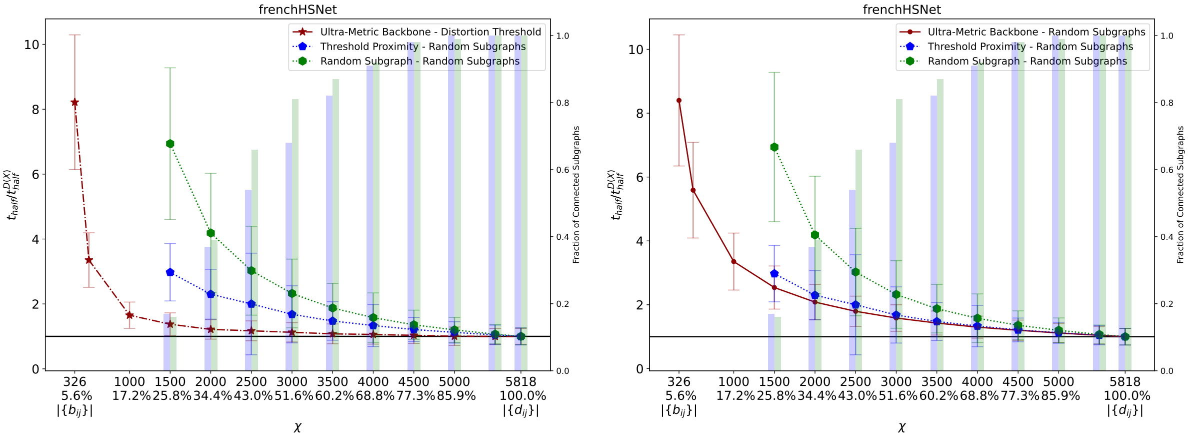


Figure 4.3: SI Dynamics t_{half} comparison between Threshold Proximity(TP), Random Subgraph(RS), and Ultra-Metric Backbone(UMB) Sparsifiers in the French High-School Network. The intermediate size subgraphs of (TP) and (RS) are constructed by adding random edges until the subgraph reaches the desired size. The ones for (UMB) are constructed by thresholding distortion (**left**) and random subgraphs (**right**). The background bars illustrate the percentage of network realizations from that size that are fully connected, respective to the initial sparsifier. These simulations were computed with $|N_S| = 10$, $|S| = 0.1|X|$, $|N_R| = 100$ and $\beta = 1$.

This comparison could also be done using the rest of the relevant backbones but the ultra-metric is the smallest possible backbone, thus, it captures the largest edge reduction possible in the distance backbones framework. However, for completeness sake, we also compare this backbone to the others.

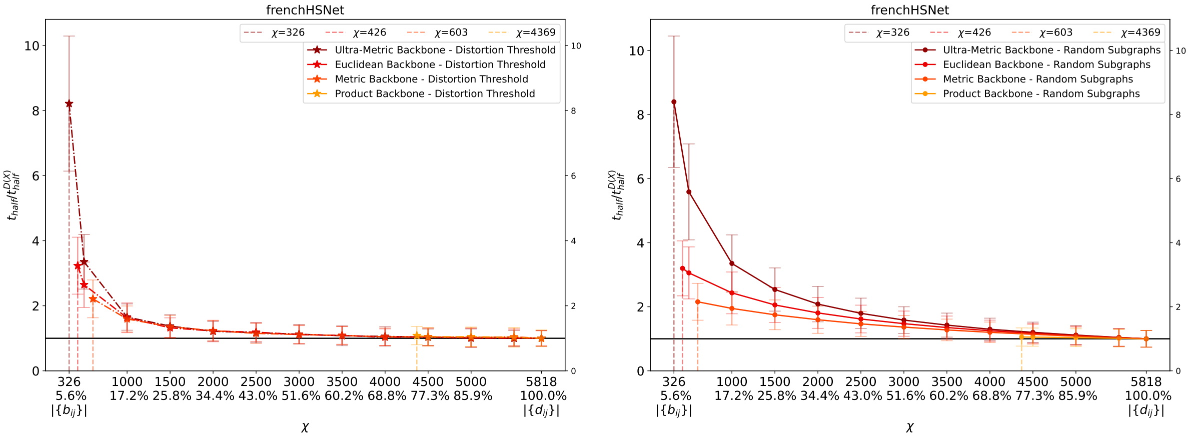


Figure 4.4: SI Dynamics t_{half} comparison between Ultra-Metric, Euclidean, Metric and Product Backbones Sparsifiers in the French High-School Network. The intermediate size subgraphs are constructed by either thresholding the respective distortions (**left**) or adding random edges until the subgraphs reach the desired size (**right**). These simulations were computed with $|N_S| = 10$, $|S| = 0.1|X|$ and $\beta = 1$.

The previous plots demonstrate that the sparsification done using distortion thresholding produces subgraphs more relevant for the SI epidemics, since the t_{half} values for these subgraphs are much closer to the original network than the random subgraphs. Also, since we have proven before that $B^{um} \subseteq B^{e} \subseteq B^{m} \subseteq B^{p}$, it is not surprising that the distortion thresholding subgraphs from these different closures produce t_{half} values within the same range for most of the subgraphs sizes. Lastly, it is also clear that the sparsification done with the product backbone represents a clear trade-off when it comes to the explainability of SI dynamics since the metrics differ very little from the original network but the sparsification is very modest and, in other networks, almost negligible. Finally, it's worthy of mention that this same analysis was also done with the t_{all} but, since this metric produces experimental data with a much higher error, it may not be as meaningful as the t_{half} and so it is not presented.

4.4 SIR Epidemics

In SIR dynamics the evaluation of sparsification techniques is not as straightforward as in SI dynamics. This has to due with the fact that, in SIR, the equilibrium/final state of the epidemic is not unique. Since there exists a recovery probability γ at the end of each timestep t for an *infected* node to recover, the epidemic can suddenly end after it begins, if $\gamma \gg \beta$ and, it can approximate an SI epidemic when $\gamma \ll \beta$. In other words, the behavior of the epidemic depends on the value of the proportion $R_0 = \beta/\gamma$, which is known in the literature [27] as the *basic reproductive number*. The study of this number in the context of contact networks usually assumes that the node degrees are homogeneous across the network and that the proximities between each pair of nodes are all 1. However, these simplifications do not reflect accurately the topology and weights distribution in the networks studied here.

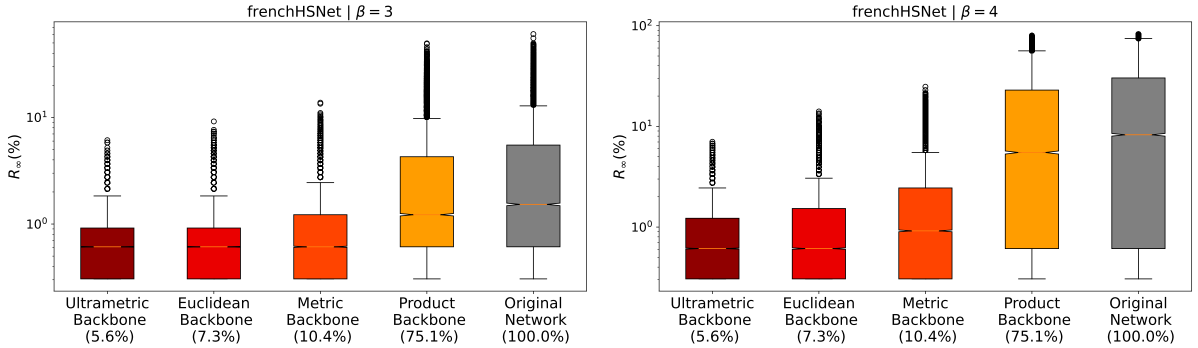


Figure 4.5: SIR Dynamics $R_\infty(\%)$ comparison between Ultra-Metric, Euclidean, Metric and Product Backbones Sparsifiers and Original Network using the French High-School Network. Each boxplot illustrates the distribution of the $R_\infty(\%)$ values obtained in the respective subgraph with $|N_S| = 10 \cdot 100$, $|S| = 0.1|X|$, $\gamma = 1$ and $\beta = 3$ (**left**) or $\beta = 4$ (**right**).

For this reason, a complete analysis of SIR dynamics in the context of network sparsifications would need a more thorough review of literature and use of diversified techniques. Thus, in Figure 4.5 we present a simpler analysis that compares SIR epidemics in the backbones with SIR epidemics in the original size network, using again the French High-School Network[4]. The metric used to compare this dynamic processes is the proportion of *recovered* individuals at the end of the epidemic, R_∞, in percentage to the number of nodes. This metric is meaningful in this context since it represents the cumulative amount of infected individuals throughout all the epidemic process, which is something one would like to control when dealing with a real-world epidemic. This figure shows that within backbone sparsifiers, the smaller is the backbone, the more reduced is the proportion of recovered individuals at the end of an SIR simulation, which might be explained by the drastic reduction in the density of the networks, that provides less opportunities for infection. Without surprise, the best backbone at preserving the distribution of $R_\infty(\%)$ is the Product Backbone, that is also the backbone with larger size. Thus, we can only conclude that, in general, these backbones don't preserve R_∞ in SIR epidemics.

[4] Like in previous plots, the corresponding figures for the other networks can be found in Appendix A

5

Conclusion

Contents

5.1 Final Remarks

The primary focus of this thesis centered on the theoretical development of distance closures and their associated backbones, employing tools from Fuzzy Logic, such as T-Norms and T-Conorms. Nevertheless, this work pursued two additional main objectives.

The first goal was to demonstrate that, building upon the groundwork laid out in [7], there exist specific distance backbones that can more reliably highlight small subsets of connections, within complex social networks, than other sparsification schemes. These subsets can be of particular interest to health specialists when dealing with a pandemic. For example, in the context of SI epidemics, it was found that edges that are metric but not ultra-metric play a crucial role in extending the time required for half of the population to become infected. This insight can be used to analyze real-world interactions and gain a better understanding of their impact on epidemiological processes.

The second major objective of this work was to explore the infinite families of backbones that can be constructed using distinct families of T-Norms. These families of T-Norms offer several generalized perspectives on the concept of path-length and, consequently, shortest-path. In parallel, these general perspectives were compared with a more profound analysis of a specific subset of backbones that embrace more intuitive notions of path-length, by means of semi-triangular distortions and λ values. The comparison between these two parameters didn't yield a a monotonous relation but it revealed a decreasing tendency between them that could hint further explorations.

In conclusion, this thesis covered the theoretical construction behind the framework of distance closures by testing it using novel sets of Triangular Norms that have been known in the literature of Fuzzy Logic for some time. This experiment involved the exploration of backbones sizes as well as the study of epidemiological processes, and allowed for the conclusion that within this framework it is possible to sparsify weighted graphs until almost any degree intended (not beyond ultrametric) and it is possible to tailor the type of sparsification to the desired path-length measure. The challenge for further applications remains to identify the most appropriate path-length measure in each context.

5.2 Future Work

While this work has made significant progress in developing each of the covered subtopics, there remain numerous adjacent areas and avenues for further research that merit exploration. One significant area of potential development involves the interpretation of T-Norms whose additive generators are functions that are not traditional distance functions. For example, using the families nomenclature from [18], the families Schweizer & Sklar 1 and Yager have additive generator functions with a signature $\varphi : [0,1] \rightarrow [0,1]$, rather than $\varphi : [0,1] \rightarrow [0,+\infty]$. These functions, specifically $1 - x^\lambda$ and $(1-x)^\lambda$, can serve as fuzzy negations and offer two distinct interpretations. One is to consider the outputs of these generators as normalized versions of distances, while the other is to regard the negation of a proximity graph as an object by itself, opening up the study of the backbones that can be derived from it.

Another promising research direction is the development of new families of T-Norms using novel distance functions. These families may yield different backbones with alternative and meaningful interpretations of path-length. An ideal characteristic for one of these families would be the ability to encompass the four primary backbones (Ultrametric, Euclidean, Metric, and Product) across its parameter range.

A more comprehensive and detailed future study could focus on investigating backbone properties, such as size and semi-triangular distortion distribution, across a diverse set of null models. These models could include examples like Barabási–Albert, Watts-Strogatz, Erdős–Rényi, Stochastic Block Model, or the Configuration Model. When combined with different weight sampling techniques, this approach could provide a deeper characterization of distance backbone properties in relation to specific, well-known network properties.

In the realm of Epidemiological Models, it may be valuable to compare the results obtained with different distance backbones to other alternative sparsification methods, such as the Multiscale Backbone or Effective Resistance Thresholding. Additionally, exploring other compartmental models like Susceptible-Infected-Susceptible (SIS) or Susceptible-Infected-Recovered-Susceptible (SIRS) could yield valuable insights. There's also room for further development of the methodology used to analyze SIR models.

Another intriguing avenue of research not discussed in this work pertains to the synchronization phenomenon in networks, whose nodes can be represented using coupled oscillators. A notable model used in this area is the Kuramoto Model [19]. This model, originally defined for homogeneous coupling, can be adapted to incorporate varying coupling strengths between pairs of nodes, reflecting proximities between individuals. While synchronization can be studied in various social interaction contexts, it has been more extensively explored in the natural sciences, such as ecology or biology, with phenomena like the synchronization of firefly blinking patterns [39] serving as a prominent example.

Another direction for future work involves exploring the Diffusion Closure framework [36], where the T-Norm/Conorm pair is $\langle \wedge_1^D, \vee_1^D \rangle$. This closure allows for the modelling of diffusion-like processes where the most relevant connection between each pair of nodes isn't a single path but rather all paths connecting them. In this case, the corresponding algebraic structure is not a semiring, since $\vee_1^D \neq \max$, so it isn't guaranteed that the Diffusion Closure Algorithm converges within a finite number of steps.

Finally, it's worth noting that the Experimental Results section did not address directionality, as all the networks studied were undirected. However, as outlined in the Theoretical Background, the methodology of Distance Backbones is applicable to directed networks [9]. Therefore, a valuable extension of this work would involve applying the same analyses to directed networks to investigate whether directionality has any impact, for example, on the proportional sizes of the backbones.

Bibliography

[1] ACZÉL, J., AND ALSINA, C. Characterizations of some classes of quasilinear functions with applications to triangular norms and to synthesizing judgments. *Aeq. Math. 25* (1982), 313–315.

[2] AHO, A. V., GAREY, M. R., AND ULLMAN, J. D. The transitive reduction of a directed graph. *SIAM Journal on Computing 1*, 2 (1972).

[3] ALSINA, C., FRANK, M. J., AND SCHWEIZER, B. *Associative Functions: Triangular Norms and Copulas*. World Scientific, February 2006.

[4] BARABÁSI, A. L. *Network Science*. Cambridge University Press, 2016.

[5] BARABÁSI, A.-L., AND ALBERT, R. Emergence of scaling in random networks. *Science 286*, 5439 (1999), 509–512.

[6] BARRAT, A., AND GÉNOIS, M. Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Science 7*, 1 (2018), 11.

[7] BRATTIG CORREIA, R., BARRAT, A., AND ROCHA, L. M. Contact networks have small metric backbones that maintain community structure and are primary transmission subgraphs. *PLoS Computational Biology 19* (2023), e1010854.

[8] CHAKRABORTY, M., AND DAS, M. Reduction of fuzzy strict order relations. *Fuzzy Sets and Systems 15* (1985), 33–44.

[9] COSTA, F., CORREIA, R., AND ROCHA, L. The distance backbone of directed networks. In *Complex Networks and Their Applications XI*, H. Cherifi, R. Mantegna, L. Rocha, C. Cherifi, and S. Micciche, Eds., vol. 1078 of *Studies in Computational Intelligence*. Springer, 2023, ch. 11.

[10] DIJKSTRA, E. W. A note on two problems in connection with graphs. *Numerical Mathematics 1* (1959), 269–271.

[11] DOMBI, J. A general class of fuzzy operators, the demorgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems 8* (1982), 149–163.

[12] FLOYD, R. W. Algorithm 97. *Communications of the ACM 5–6* (1962), 345.

[13] GIRVAN, M., AND NEWMAN, M. E. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences 99*, 12 (2002), 7821–7826.

[14] GONDRAN, M., AND MINOUX, M. *Graphs, Dioids and Semirings*. Operations Research/Computer Science Interfaces Series. Springer New York, NY, 2008. Hardcover ISBN: 978-0-387-75449-9.

[15] ISELLA, L., STEHLÉ, J., BARRAT, A., CATTUTO, C., PINTON, J.-F., AND DEN BROECK, W. V. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of Theoretical Biology 271* (2011), 166–180.

[16] KALAVRI, V., SIMAS, T., AND LOGOTHETIS, D. The shortest path is not always a straight line: Leveraging semi-metricity in graph analysis. *Proceedings of the VLDB Endowment 9* (2016), 672–683.

[17] KLEMENT, E. P., MESIAR, R., AND PAP, E. *Triangular Norms*. Trends in Logic. Springer Dordrecht, 2000. Hardcover ISBN: 978-0-7923-6416-0.

[18] KLIR, G. J., AND YUAN, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995.

[19] KURAMOTO, Y. Lecture notes in physics. In *International Symposium on Mathematical Problems in Theoretical Physics* (New York, 1975), H. Araki, Ed., vol. 39, Springer-Verlag, p. 420.

[20] LESKOVEC, J., AND FALOUTSOS, C. Sampling from large graphs. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (August 2006), pp. 631–636.

[21] LING, C. H. Representation of associative functions. *Publ. Math. Debrecen 12* (1965), 189–212.

[22] MADKOUR, A., AREF, W. G., REHMAN, F. U., RAHMAN, M. A., AND BASALAMAH, S. M. A survey of shortest-path algorithms. *Computing Research Repository* (2017).

[23] MASTRANDREA, R., FOURNET, J., AND BARRAT, A. Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLoS ONE 10*, 9 (2015), e0136497.

[24] MENGER, K. Statistical metrics. *Proceedings of the National Academy of Sciences of the United States of America 8* (1942), 535–537.

[25] MERCIER, A., SCARPINO, S., AND MOORE, C. Effective resistance against pandemics: Mobility network sparsification for high-fidelity epidemic simulations. *PLoS Comput Biol 18*, 11 (2022), e1010650.

[26] NEWMAN, M. E. *Networks: An Introduction*. Oxford University Press, 2018.

[27] PASTOR-SATORRAS, R., CASTELLANO, C., MIEGHEM, P. V., AND VESPIGNANI, A. Epidemic processes in complex networks. *Rev. Mod. Phys. 87* (August 2015), 925.

[28] PAÑOS, D. S., COSTA, F. X., AND ROCHA, L. M. Semi-metric topology characterizes epidemic spreading on complex networks, 2023.

[29] ROZUM, J. C., AND ROCHA, L. M. The ultrametric backbone is the union of all minimum spanning trees. 'In Preparation'.

[30] SARAMÄKI, J., KIVELÄ, M., ONNELA, J.-P., KASKI, K., AND KERTÉSZ, J. Generalizations of the clustering coefficient to weighted complex networks. *Phys. Rev. E 75* (February 2007), 027105.

[31] SAYAMA, H. *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNY, 2015.

[32] SCHWEIZER, B., AND SKLAR, A. Statistical metric spaces. *Pacific Journal of Mathematics 10* (1960), 313–334.

[33] SCHWEIZER, B., AND SKLAR, A. Associative functions and abstract semigroups. *Publicationes Mathematicae Debrecen 10*, 1-4 (1963), 69–81.

[34] SCHWEIZER, B., AND SKLAR, A. *Probabilistic Metric Spaces*. Dover Publications, 2011.

[35] SIMAS, T., CORREIA, R. B., AND ROCHA, L. M. The distance backbone of complex networks. *Journal of Complex Networks 2021*, 00 (2021), 1–35.

[36] SIMAS, T., AND ROCHA, L. M. Distance closures on complex networks. *Network Science 3*, 2 (2015), 227 – 268. Published online: 30 March 2015.

[37] SIMONS, P. Jan Łukasiewicz. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta and U. Nodelman, Eds., Spring 2023 ed. Metaphysics Research Lab, Stanford University, 2023.

[38] STEEN, L. A., AND J. ARTHUR SEEBACH, J. *Counterexamples in Topology*. Springer-Verlag, 1970.

[39] STROGATZ, S. H., AND STEWART, I. Coupled oscillators and biological synchronization. *Scientific American* (December 1993), 68–75.

[40] TAYLOR, J. R. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, second edition ed. University Science Books, 1997.

[41] WARSHALL, S. A theorem on boolean matrices. *Journal of the ACM 9*, 1 (1962), 11–12.

[42] WATTS, D. J., AND STROGATZ, S. H. Collective dynamics of small-world networks. *Nature 393*, 6684 (1998), 440.

[43] ZADEH, L. Fuzzy sets. *Information and Control 8*, 3 (1965), 338–353.

[44] ZADEH, L. Fuzzy logic and the calculi of fuzzy rules, fuzzy graphs, and fuzzy probabilities. *Computers & Mathematics with Applications 37*, 11-12 (1999), 35.

[45] ÁNGELES SERRANO, M., BOGUÑÁ, M., AND VESPIGNANI, A. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences 106*, 16 (2009), 6483–6488.

A

Extra Figures

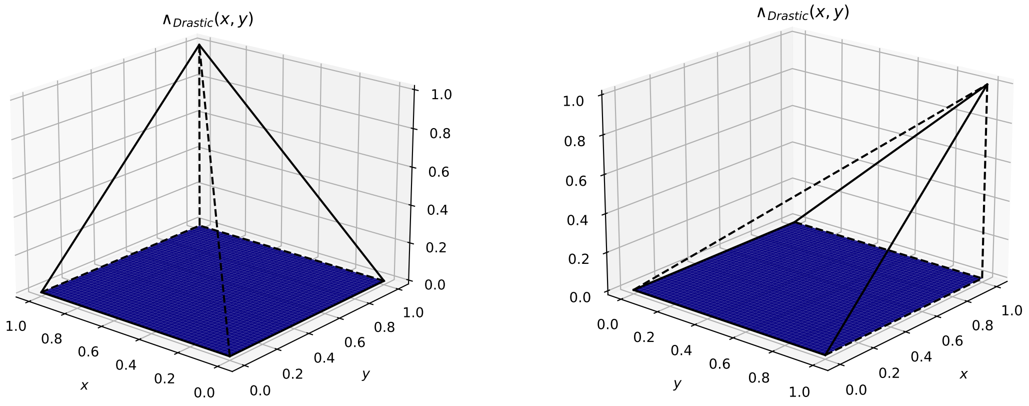


Figure A.1: Drastic T-Norm

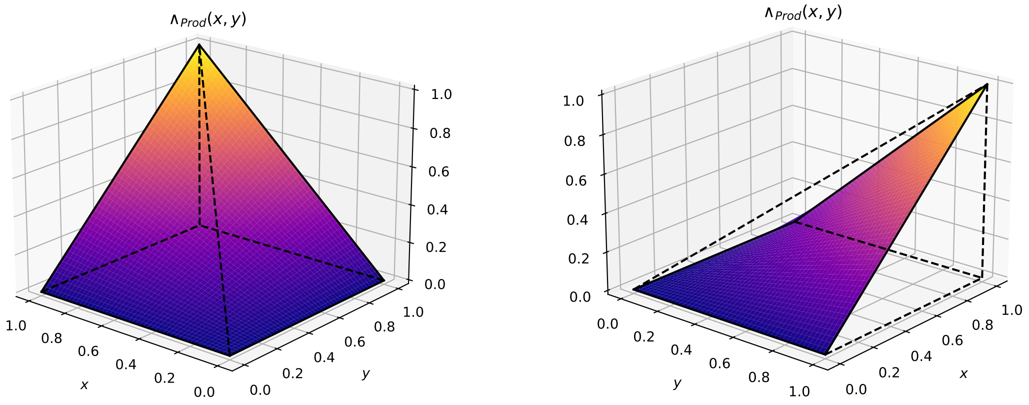


Figure A.2: Product T-Norm

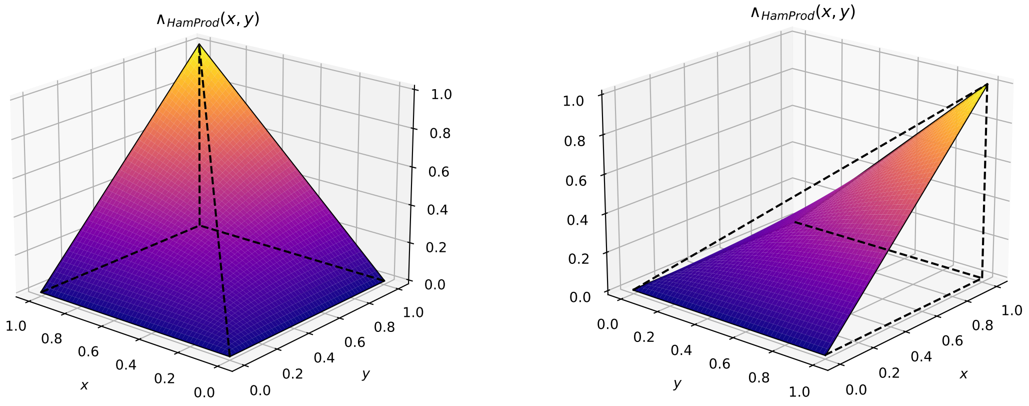


Figure A.3: Hamacher Product T-Norm

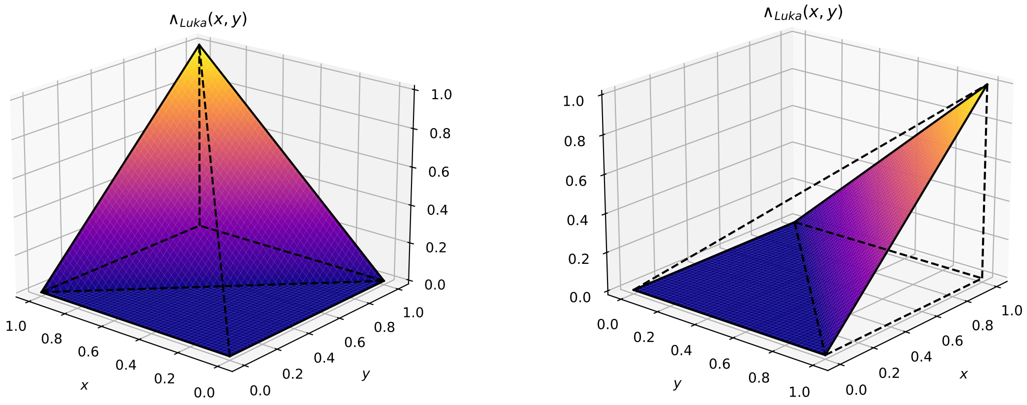


Figure A.4: Łukasiewicz T-Norm

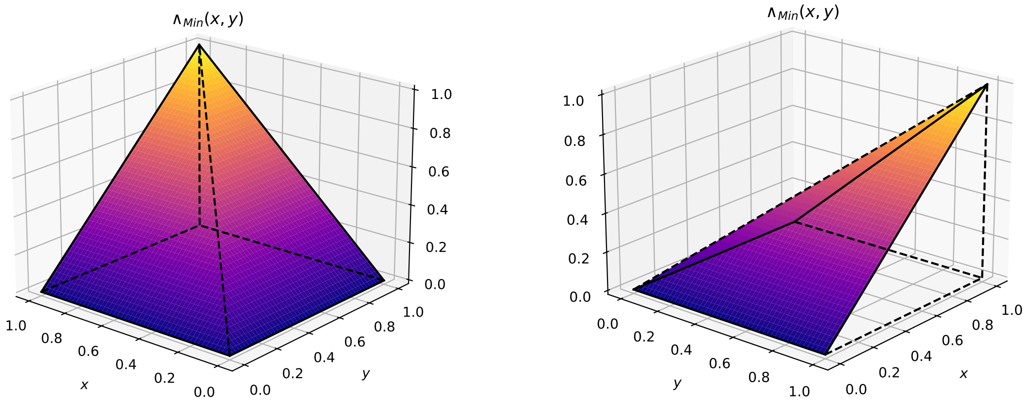


Figure A.5: Minimum T-Norm

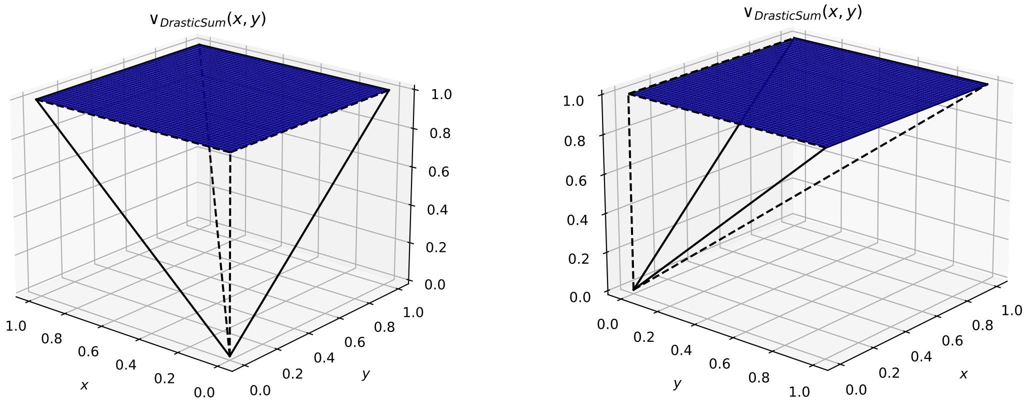


Figure A.6: Drastic Sum T-Conorm

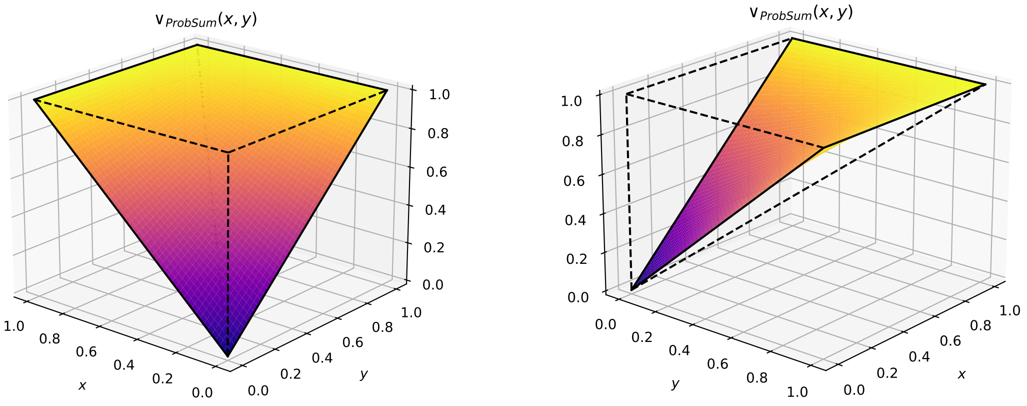


Figure A.7: Probabilistic Sum T-Conorm

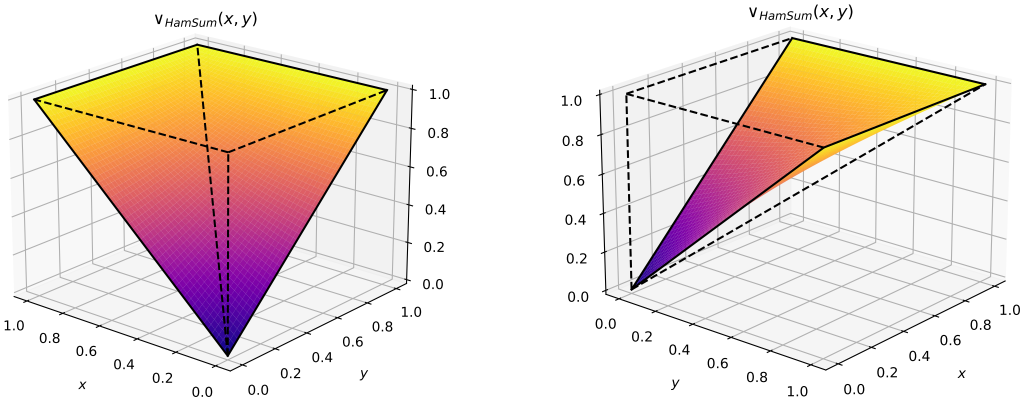


Figure A.8: Hamacher Sum T-Conorm

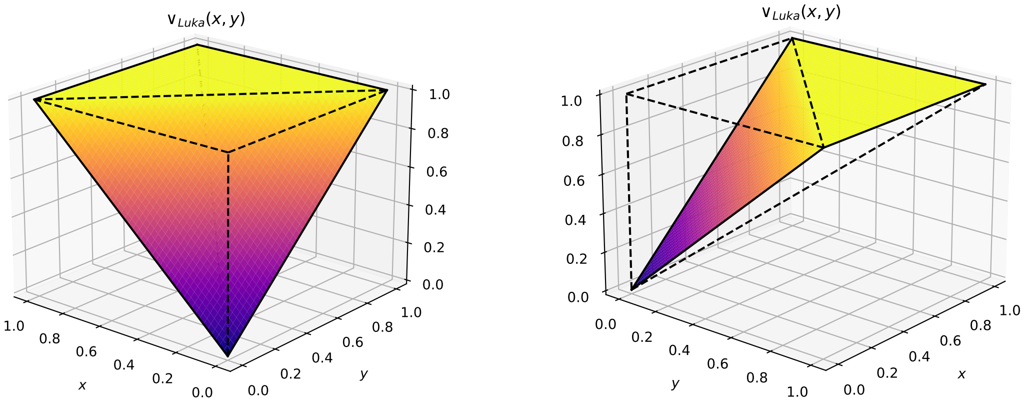


Figure A.9: Łukasiewicz T-Conorm

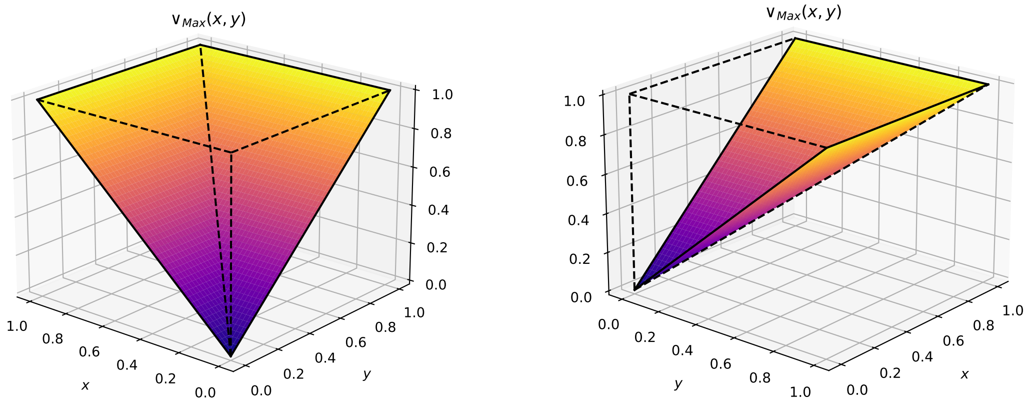


Figure A.10: Maximum T-Conorm

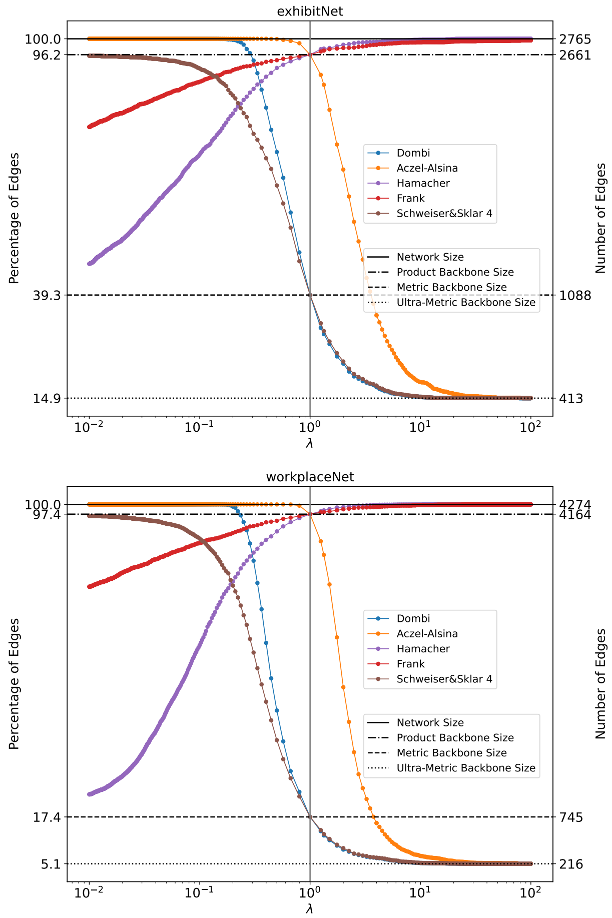
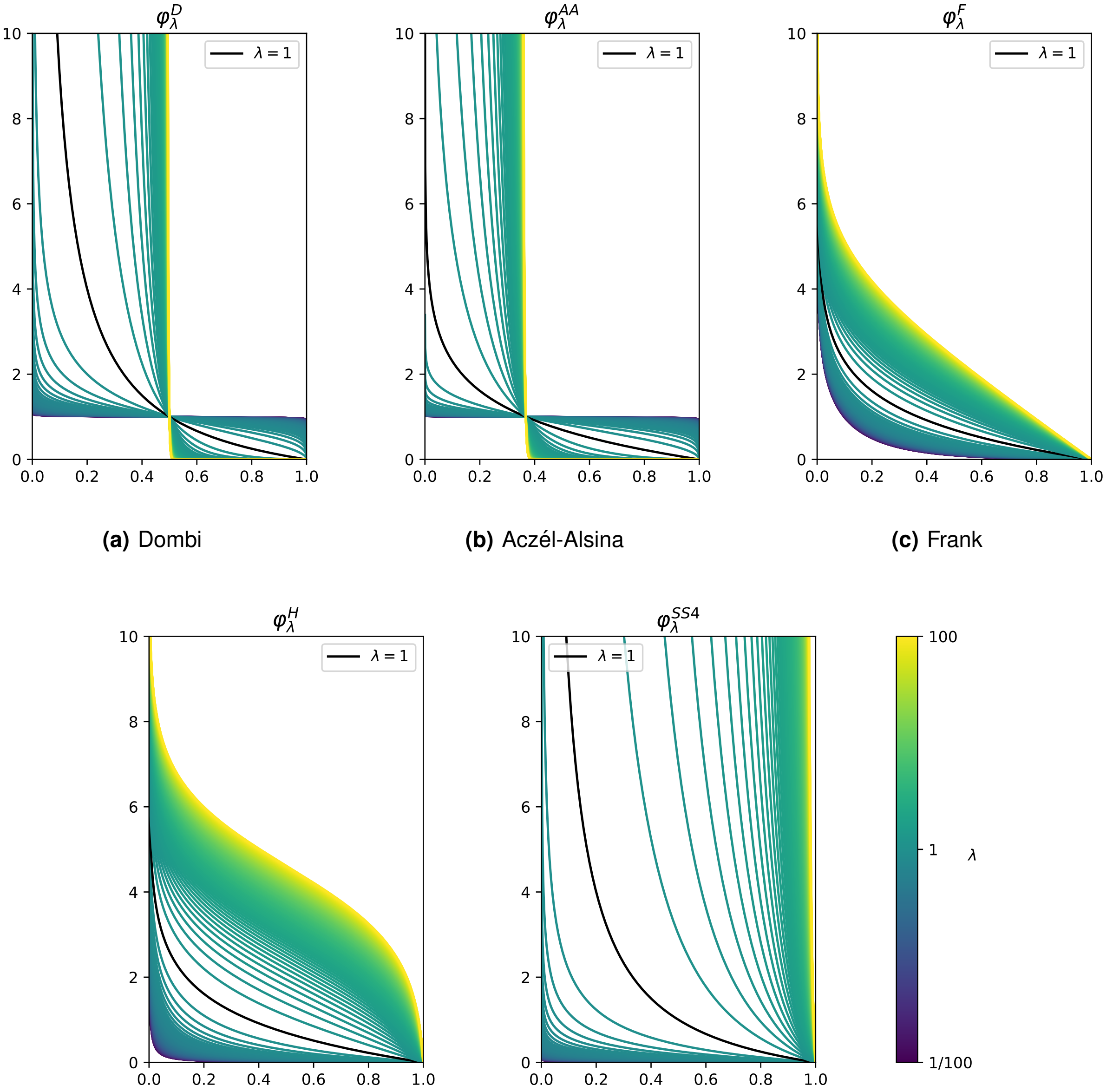


Figure A.11: Families Backbones Sizes of the Exhibit and Workplace Network. Each point corresponds to the size of the backbone for the respective family and λ value. Since the computation of many backbones is computational expensive, the backbones are computed for $\lambda \in \{\frac{1}{n}, \frac{1}{n+0.25}, \frac{1}{n+0.5}, \frac{1}{n+0.75}, n, n+0.25, n+0.5, n+0.75 : n \in \mathbb{N}\} \cap [\frac{1}{100}, 100]$. Although the domain of λ is $[0, +\infty]$, the largest variation in the sizes occurs in $[\frac{1}{20}, 20]$ and so the figure only presents that subinterval.



(a) Dombi

(b) Aczél-Alsina

(c) Frank

(d) Hamacher

(e) Schweizer & Sklar 4

(f) λ

Figure A.12: Additive decreasing generators of the different families with $\lambda \in \{\frac{1}{n} : 100 \geq n > 1\} \cup \{n : 1 \leq n \leq 100\}$

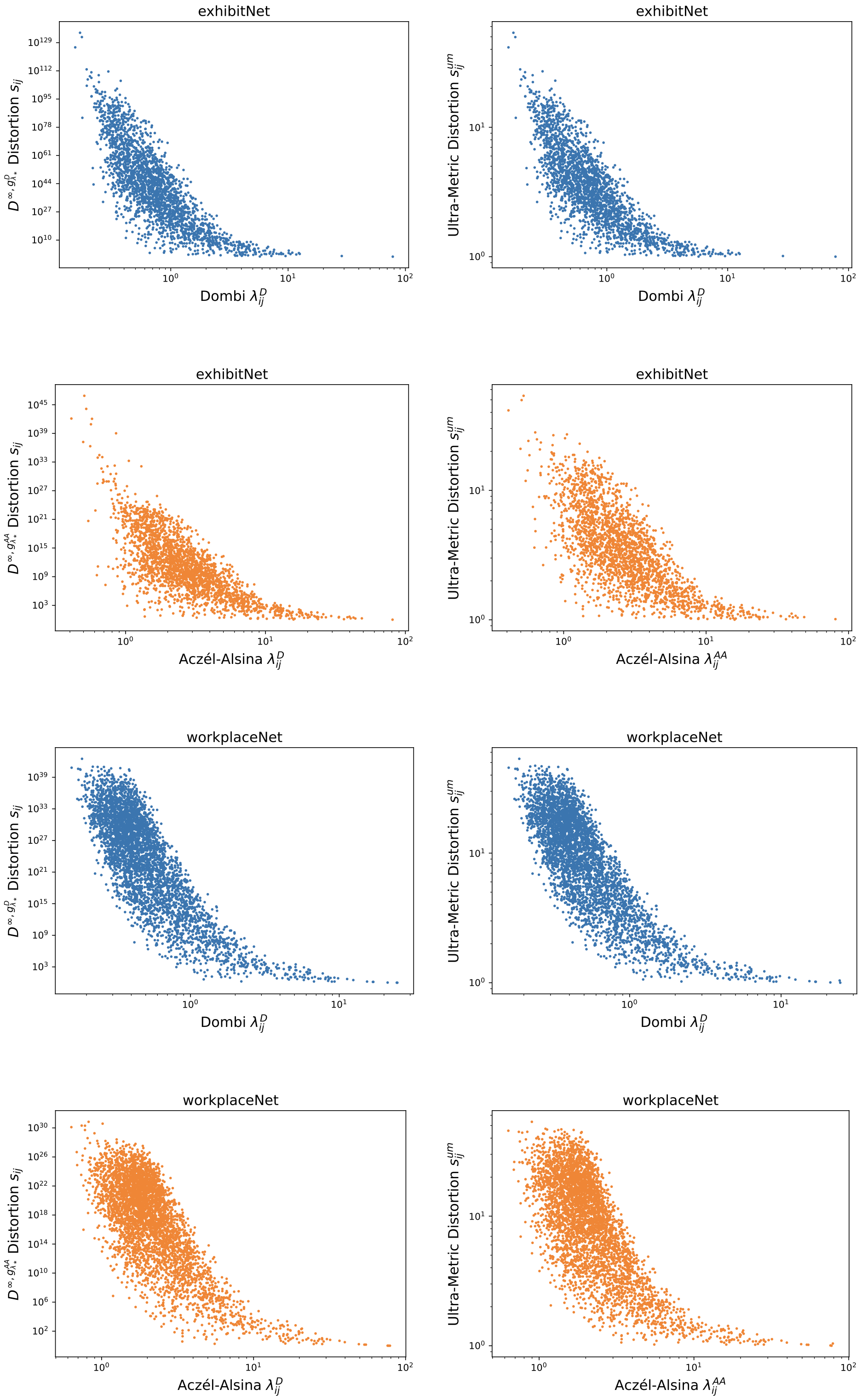


Figure A.13: Relation between λ and distortions values in the Exhibit and Workplace Networks. Each point in these figures corresponds to a non-ultrametric edge. The vertical axis measures the first λ at which that edge was added to the backbone in that family. The horizontal axis measures the distortion value (when it is not 1) whether it is from the closure associated with the $B^{g_{\lambda_*}}$ (left) or the B^{um} (right) backbone.

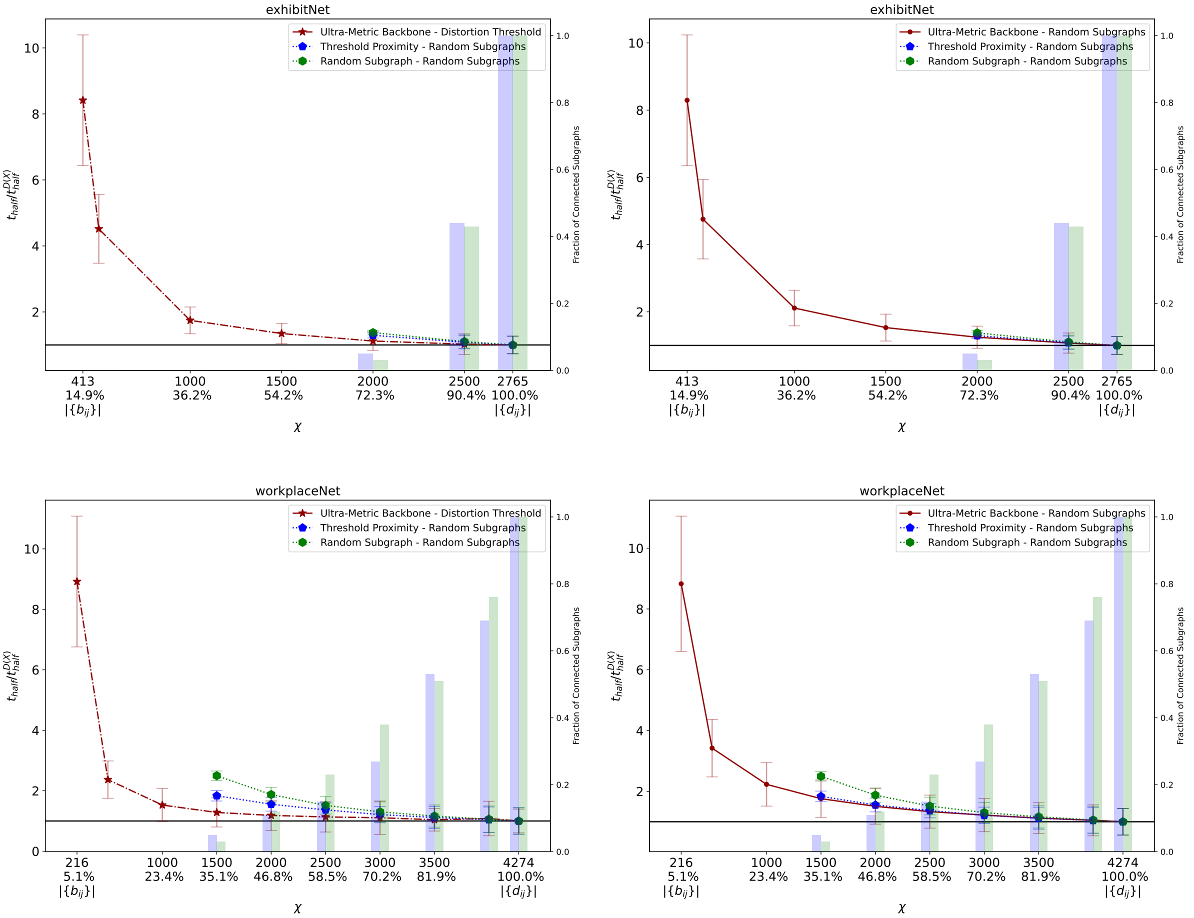


Figure A.14: SI Dynamics t_{half} comparison between Threshold Proximity(TP), Random Subgraph(RS), and Ultra-Metric Backbone(UMB) Sparsifiers in the Exhibit and Workplace Networks. The intermediate size subgraphs of (TP) and (RS) are constructed by adding random edges until the subgraph reaches the desired size. The ones for (UMB) are constructed by thresholding distortion (**left**) and random subgraphs (**right**). The background bars illustrate the percentage of network realizations from that size that are fully connected, respective to the initial sparsifier. These simulations were computed with $|N_S| = 10$, $|S| = 0.1|X|$, $|N_R| = 100$ and $\beta = 3$.

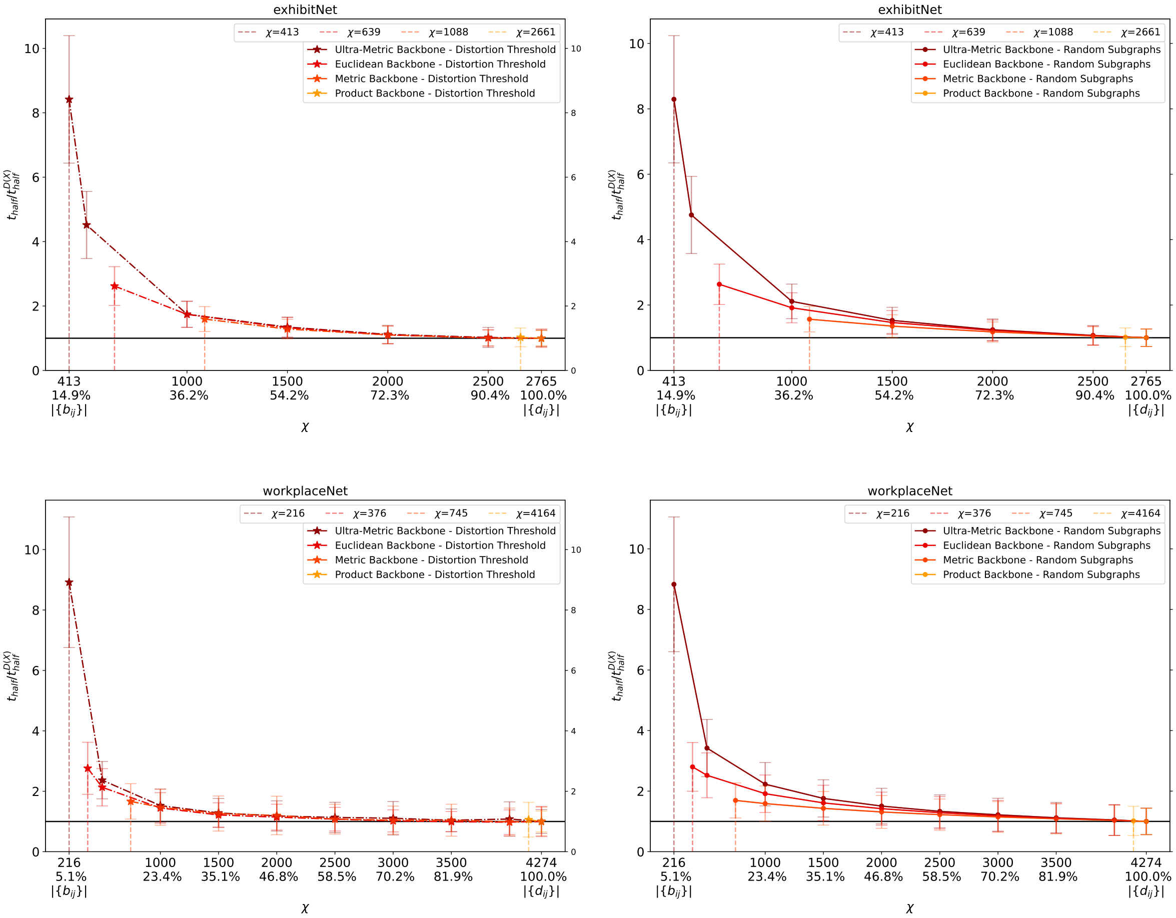


Figure A.15: SI Dynamics t_{half} comparison between Ultra-Metric, Euclidean, Metric and Product Backbones Sparsifiers in the Exhibit and Workplace Networks. The intermediate size subgraphs are constructed by either thresholding the respective distortions (**left**) or adding random edges until the subgraphs reach the desired size (**right**). These simulations were computed with $|N_S| = 10$, $|S| = 0.1|X|$ and $\beta = 3$.

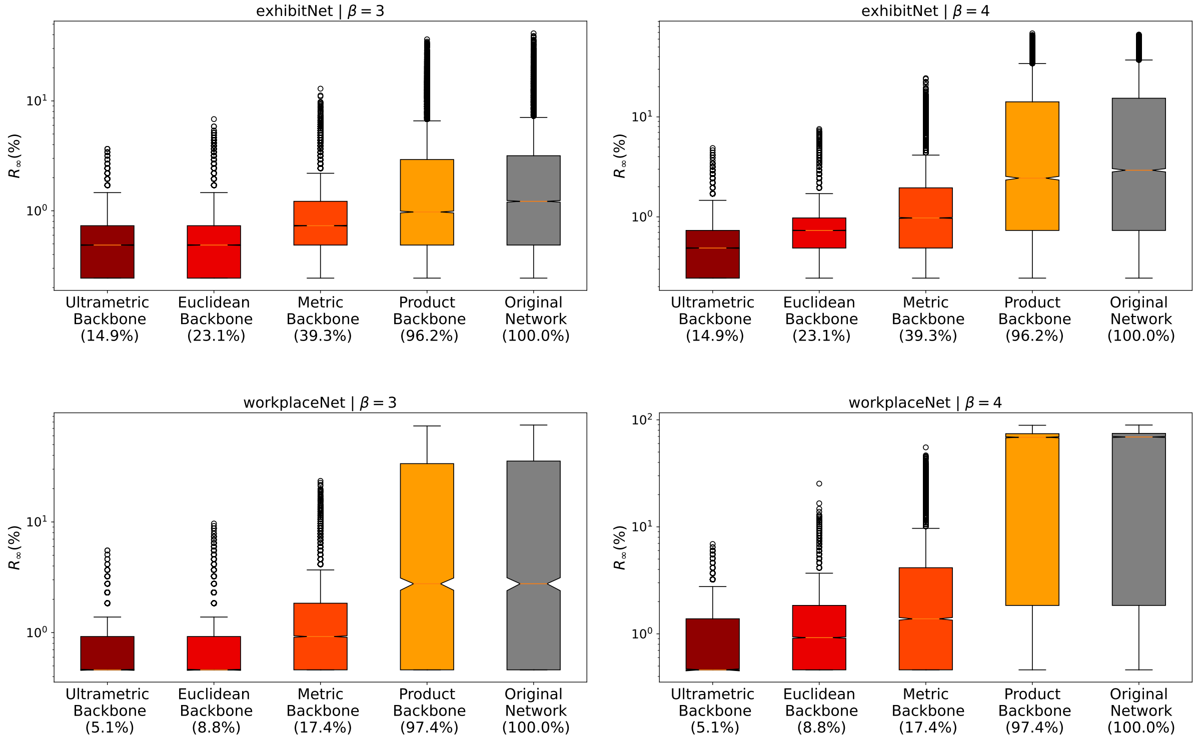


Figure A.16: SIR Dynamics $R_\infty(\%)$ comparison between Ultra-Metric, Euclidean, Metric and Product Backbones Sparsifiers and Original Network for the Exhibit and Workplace Networks. Each boxplot illustrates the distribution of the $R_\infty(\%)$ values obtained in the respective subgraph with $|N_S| = 10{\cdot}100$, $|S| = 0.1|X|$, $\gamma = 1$ and $\beta = 3$ (**left**) or $\beta = 4$ (**right**).

B

Extra Algorithms

Algorithm B.1 Dijkstra Algorithm

Input: Weighted Distance Graph $G = (V, E)$, Source Node $s \in V$

```
FOR EACH $v \in V$ :
    dist[$v$] $\leftarrow +\infty$
    prev[$v$] $\leftarrow$ None
    $Q$.add($v$)
dist[$s$] $\leftarrow 0$

WHILE $Q \neq \varnothing$:
    #Finding the node in $Q$ with minimum distance from the source node
    dist$_{min}$ $\leftarrow +\infty$
    FOR EACH $v \in Q$:
        IF dist[$v$] $\leq$ dist$_{min}$:
            dist$_{min}$ $\leftarrow$ dist[$v$]
            $u \leftarrow v$
    #Update the shortest paths distance of the neighbors of $u$
    $Q$.remove($u$)
    FOR EACH $v \in u$.neighbors:
        IF $v \in Q$ AND dist[$u$] + $(u, v)$.weight $<$ dist[$v$]:
            dist[$v$] $\leftarrow$ dist[$u$] + $(u, v)$.weight
            prev[$v$] $\leftarrow u$
RETURN dist, prev
```

Output: Shortest-Paths to source node: `dist`, `prev`

Description: `dist` stores the minimum distance from every node to the source node, `prev` stores the previous node in the shortest-path to the source node.

Algorithm B.2 SI Spreading on a Proximity Graph

Inputs: Proximity Graph $P = (V, E)$, Seed Node v_1, Infection Parameter β

```
FOR EACH $v \in V$:
    $v$.state $\leftarrow s$

N $\leftarrow |V|$
$v_1$.state $\leftarrow i$
InfectedNodes $\leftarrow \{v_1\}$
PercInfected $\leftarrow [\frac{1}{\texttt{N}}100]$

WHILE |InfectedNodes| $<$ N:
    FOR EACH $j \in$ range(0, |InfectedNodes|):
        InfNode $\leftarrow$ InfectedNodes[$j$]
        SusceptibleNeighbors $\leftarrow \{v \in$ InfNode.neighbors : $v$.state $= s\}$
        FOR EACH $v \in$ SusceptibleNeighbors:
            $r \leftarrow$ RandomNumber(0, 1)
            $p \leftarrow$ (InfNode, $v$).weight
            IF $r < \beta p$:
                InfectedNodes $\leftarrow$ InfectedNodes $\cup \{v\}$
                $v$.state $\leftarrow i$
    PercInfected.append($\frac{|\texttt{InfectedNodes}|}{\texttt{N}}100$)

RETURN PercInfected
```

Outputs: List of percentages, PercInfected, containing the percentage of infected nodes at time i.

Algorithm B.3 SIR Spreading on a Proximity Graph

Inputs: Proximity Graph $P = (V, E)$, Seed Node v_1, Infection Parameter β, Recovery Parameter γ

```
FOR EACH $v \in V$ :
    $v$.state $\leftarrow s$

N $\leftarrow |V|$
$v_1$.state $\leftarrow i$
InfectedNodes $\leftarrow \{v_1\}$
PercInfected $\leftarrow [\frac{1}{N}100]$
RecoveredNodes $\leftarrow \varnothing$
PercRecovered $\leftarrow [0]$

WHILE $|$InfectedNodes$|$ < N:
    newInfected $\leftarrow \varnothing$
    newRecovered $\leftarrow \varnothing$
    FOR EACH $j \in$ range$(0,$ |InfectedNodes|$)$:
        InfNode $\leftarrow$ InfectedNodes$[j]$
        SusceptibleNeighbors $\leftarrow \{v \in$ InfNode.neighbors $: v$.state $= s\}$
        FOR EACH $v \in$ SusceptibleNeighbors:
            $r_1 \leftarrow$ RandomNumber$([0, 1])$
            $p \leftarrow$ (InfNode, $v$).weight
            IF $r_1 < \beta p$ AND $v \notin$ newInfected:
                newInfected $\leftarrow$ newInfected $\cup \{v\}$
        $r_2 \leftarrow$ RandomNumber$([0, 1])$
        IF $r_2 < \gamma$:
            newRecovered $\leftarrow$ newRecovered $\cup$ {InfNode}
    FOR EACH $n \in$ newInfected:
        $n$.state $\leftarrow i$
        InfectedNodes.append$(n)$
    FOR EACH $n \in$ newRecovered:
        $n$.state $\leftarrow r$
        InfectedNodes.remove$(n)$
        RecoveredNodes.append$(n)$
    PercInfected.append$(\frac{|\text{InfectedNodes}|}{N}100)$
    PercRecovered.append$(\frac{|\text{RecoveredNodes}|}{N}100)$

RETURN PercInfected, PercRecovered
```

Outputs: Lists of percentages, PercInfected and PercRecovered, containing the percentages of *infected* nodes and of *recovered* nodes at time i.