Modulus on temporal graphs

by

Vikenty Mikheev

M.S., University of Alaska, Fairbanks, 2014

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department Of Mathematics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2020

Abstract

p-Modulus was adopted by Graph Theory from Complex Analysis and became a powerful and versatile tool for exploring the structure of graphs and networks. Clustering and community detection, the construction of a large class of graph metrics, measures of centrality, hierarchical graph decomposition, and the solution to game-theoretic models of secure network broadcast are successful applications. The flexibility of p-modulus allows it to be applied to a wide variety of graph types; the essential definitions are easily adapted for directed or undirected graphs, for weighted or unweighted graphs, for simple graphs or multigraphs, or even for hypergraphs. In this work, we study application of p-modulus to temporal networks — that is, networks whose structure can change over time.

We introduce a definition of temporal p-modulus for a temporal graph with discrete time availability moments assigned to edges. We provide theorems allowing to adjust static p-modulus algorithms to compute temporal p-modulus. We present the dual problem. For the $p = 2$ case, it has a probabilistic interpretation. Also, we give a mass-transportation interpretation of temporal Modulus for various values of $1 \leq p \leq \infty$. We consider different types of penalty function and study properties of temporal p-modulus with it. We show examples on several standard graphs modified to temporal ones. Time-respecting paths are given special attention. In order to obtain an efficient numerical algorithm to compute p-modulus we adjust the well-know Dijkstra algorithm to find optimal paths by two criteria — that is, time of arrival and static length. We compare this algorithm with another way: building a directed static graph equivalent to a given temporal one and running classic Dijkstra algorithm. Computational complexity estimates for both algorithms are the same.

Modulus on temporal graphs

by

Vikenty Mikheev

M.S., University of Alaska, Fairbanks, 2014

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department Of Mathematics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2020

Approved by:

Major Professor
Nathan Albin

Copyright

© Vikenty Mikheev 2020.

Abstract

p-Modulus was adopted by Graph Theory from Complex Analysis and became a powerful and versatile tool for exploring the structure of graphs and networks. Clustering and community detection, the construction of a large class of graph metrics, measures of centrality, hierarchical graph decomposition, and the solution to game-theoretic models of secure network broadcast are successful applications. The flexibility of p-modulus allows it to be applied to a wide variety of graph types; the essential definitions are easily adapted for directed or undirected graphs, for weighted or unweighted graphs, for simple graphs or multigraphs, or even for hypergraphs. In this work, we study application of p-modulus to temporal networks — that is, networks whose structure can change over time.

We introduce a definition of temporal p-modulus for a temporal graph with discrete time availability moments assigned to edges. We provide theorems allowing to adjust static p-modulus algorithms to compute temporal p-modulus. We present the dual problem. For the $p = 2$ case, it has a probabilistic interpretation. Also, we give a mass-transportation interpretation of temporal Modulus for various values of $1 \leq p \leq \infty$. We consider different types of penalty function and study properties of temporal p-modulus with it. We show examples on several standard graphs modified to temporal ones. Time-respecting paths are given special attention. In order to obtain an efficient numerical algorithm to compute p-modulus we adjust the well-know Dijkstra algorithm to find optimal paths by two criteria — that is, time of arrival and static length. We compare this algorithm with another way: building a directed static graph equivalent to a given temporal one and running classic Dijkstra algorithm. Computational complexity estimates for both algorithms are the same.

Table of Contents

List of Figures

Acknowledgments

First and foremost, I thank my advisor Dr. Nathan Albin. I am forever grateful for the enormous amount of time he spent meeting with me to discuss the results in this work and other projects of mine. I also thank everyone else who served on my supervisory committee. My thanks go to Dr. Pietro Poggi-Corradini, Dr. Anna Zemlyanova and Dr. Caterina Scoglio for serving on this committee and being wonderful examples to me. Thank you to Dr. Haiyan Wang for serving as the outside chairperson of my supervisory committee. Special thanks to Dr. Andrew Bennett for being the Mathematics Department head and Dr. Sarah Reznikoff for being the graduate student director. Thank you for all of the knowledge and the opportunities that you have provided for graduate students. Lastly, I thank all of my friends and family who supported me through my time in the graduate school.

Chapter 1

Background. Modulus on Static Graphs.

1.1 Introduction

Discrete p-modulus (see, e.g., [2, 3, 7, 8]) has proven to be a powerful and versatile tool for exploring the structure of graphs and networks. Applications discovered thus far include clustering and community detection [15], the construction of a large class of graph metrics [5], measures of centrality [14], hierarchical graph decomposition [4], and the solution to game-theoretic models of secure network broadcast [6]. The flexibility of p-modulus allows it to be applied to a wide variety of graph types; the essential definitions are easily adapted for directed or undirected graphs, for weighted or unweighted graphs, for simple graphs or multigraphs, or even for hypergraphs. In this dissertation, we begin the exploration of p-modulus applied to temporal networks—that is, networks whose structure can change over time.

1.2 Static Graph

Consider a *finite weighted (static) graph* $G = (V, E, \sigma)$ with vertex set V, edge set E, and positive edge weights $\sigma : E \to (0, \infty)$. To keep notation compact, we shall assume that G is simple and undirected; extensions to more complicated graph structures are typically straightforward. The case $\sigma \equiv 1$ can be interpreted as the case of an unweighted graph. In what follows, it will frequently be convenient to adopt linear algebra notation. To this end, we shall also think of σ as a positive abstract vector indexed by the edge set E, and write $\sigma \in \mathbb{R}^E_{>0}$. In order to distinguish this case from that of temporal graphs discussed in the next section, we will sometimes refer to $G = (V, E, \sigma)$ as a *static graph*.

1.3 Objects, families, usage

By a *family of objects* on G, we shall mean a collection, Γ, of abstract objects defined through some relationship with G. For example, each of the following is a possible choice for Γ.

- The family of spanning trees of G.
- The family of simple cycles in G.
- The family of all paths connecting two distinct vertices s and t in G.
- The family of all walks that traverse at least k edges (where k is some positive integer).

In order to keep the narrative simple, we shall assume that Γ is a finite set. A theory can also be established for infinite families, but more care must be taken (see, e.g., [8, Sec. 7]).

A *usage matrix*, $\mathcal{N} \in \mathbb{R}^{\Gamma \times E}_{\geq 0}$, is an abstract matrix assigning a nonnegative value $\mathcal{N}(\gamma, e)$ to each pair $(\gamma, e) \in \Gamma \times E$. There is no prescribed rule for making this definition, though the "natural" definition often suffices in most cases. For example, if the objects in Γ can be identified with subsets of edges (as in the case of spanning trees, cycles and paths), then a

natural definition for $\mathcal{N}$ is

$$\mathcal{N}(\gamma, e) = \begin{cases} 1 & \text{if } e \in \gamma, \\ 0 & \text{if } e \notin \gamma. \end{cases} \tag{1.1}$$

For walks, it is often useful to store a traversal count instead. For some sets of objects, it also proves useful to allow $\mathcal{N}$ to take non-integer values.

1.3.1 Densities, admissibility and modulus

By a *density* on G, we mean a non-negative vector on the edges, $\rho \in \mathbb{R}^E_{\geq 0}$. Each density provides a way to assess a type of cost on the objects in Γ. It is helpful to think of $\rho(e)$ as a cost per unit usage for edge e. The *total usage costs* incurred by an object $\gamma \in \Gamma$, often referred to as the ρ-*length* of γ, is defined as

$$\ell_\rho(\gamma) := \sum_{e \in E} \mathcal{N}(\gamma, e)\rho(e) = (\mathcal{N}\rho)(\gamma).$$

In words, the ρ-length of an object is the accumulation of its edge usage costs.

In the theory of p-modulus, special attention is paid to the densities that assess a minimum unit cost to all objects in the family of interest. Such densities are called *admissible* for the family Γ. The collection of all admissible densities, $\mathrm{Adm}(\Gamma)$, is defined as

$$\mathrm{Adm}(\Gamma) := \{\rho \in \mathbb{R}^E_{\geq 0} : \ell_\rho(\gamma) \geq 1 \ \forall \gamma \in \Gamma\}.$$

We will often express this set of lengths inequalities in linear algebra notation as $\mathcal{N}\rho \geq \mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^\Gamma$ is the vector of all ones. To each density ρ is assigned a p-*energy* (parametrized by $p \in [1, \infty]$ as well as the edge weights σ), which is defined as

$$\mathcal{E}_{p,\sigma}(\rho) = \begin{cases} \sum\limits_{e \in E} \sigma(e)\rho(e)^p & \text{if } 1 \leq p < \infty, \\ \sup\limits_{e \in E} \sigma(e)\rho(e) & \text{if } p = \infty. \end{cases} \tag{1.2}$$

Finally, the *p-modulus* of the family Γ is defined as the value of an optimization problem:

$$\mathrm{Mod}_{p,\sigma}(\Gamma) = \min_{\rho \in \mathrm{Adm}(\Gamma)} \mathcal{E}_{p,\sigma}(\rho). \tag{1.3}$$

Geometrically, p-modulus is simply the distance in $\mathbb{R}^E$ (measured by a weighted p-norm) between the origin and the closed, convex set $\mathrm{Adm}(\Gamma)$. From this, it is clear that the minimum is attained. A density $\rho^* \in \mathrm{Adm}(\Gamma)$ whose energy coincides with the p-modulus is called an *optimal density*. When $p \in (1, \infty)$, the strict convexity of the p-norm implies that there is a unique optimal ρ^*.

1.3.2 An example

The following is a helpful example found, e.g., in [2]. Consider a graph composed of k parallel paths, each with ℓ hops, connecting a vertex s to another vertex t. For simplicity, let $\sigma \equiv 1$. For the family Γ, we choose $\Gamma = \Gamma(s,t)$, the family of paths connecting from s to t (so $|\Gamma| = k$). From symmetry arguments, it is easy to guess that an optimal density is $\rho^* = \frac{1}{\ell}$, regardless the value of p and, therefore, that

$$\mathrm{Mod}_{p,1}(\Gamma(s,t)) = \mathcal{E}_{p,1}(\rho^*) = k\ell \left(\frac{1}{\ell}\right)^p = \frac{k}{\ell^{p-1}}.$$

This demonstrates a general heuristic that modulus is made large by a combination of "diversity" (in this case, more paths from s to t) and "shortness" (in this case, shorter paths from s to t). The parameter p adjusts the relative sensitivity of modulus to these two quantities: smaller p for more sensitivity to diversity, larger p for more sensitivity to shortness.

1.3.3 Interpretations of modulus

In practice, both the value $\mathrm{Mod}_{p,\sigma}(\Gamma)$ as well as the value of the optimal density ρ^* contain useful information about the underlying graph G. Some intuition about the meaning of modulus can be developed by considering specific examples. In the next section, we will use

the intuition gathered here to propose method for adapting p-modulus to families of objects on temporal graphs.

Modulus as dissipated power

For $p \in (1, \infty)$, it is possible to relate p-modulus to the dissipated power in a (generally nonlinear) resistor network. In this interpretation, we view the edges of the graph as nonlinear resistors and the vertices as junctions at which these resistors are joined. Along each resistor, we assume a relationship between the potential drop and current flow of the form

$$\text{current} = \sigma \times \text{voltage}^{p-1}.$$

(In the case $p = 2$, this is simply Ohm's law for a resistor with conductance σ.)

Now consider distinct vertices $s, t \in V$ and suppose a unit voltage potential drop is induced between these two junctions. From Dirichlet's minimum power principle, one would expect a steady-state response determined by a potential $\phi \in \mathbb{R}^V$ (the voltage at each junction) which minimizes the total dissipated power, leading to the following optimization problem.

$$\begin{aligned} \underset{\phi \in \mathbb{R}^V}{\text{minimize}} \quad & \sum_{\{x,y\} \in E} \sigma(\{x,y\}) |\phi(x) - \phi(y)|^p \\ \text{subject to} \quad & \phi(s) = 0, \ \phi(t) = 1. \end{aligned} \tag{1.4}$$

Notice that, for each edge $e = \{x, y\} \in E$, we are associating $\sigma(e)$ with the conductance of the resistor assigned to that edge. The relationship to p-modulus is expressed in the following Theorem [2, Thm. 4.2].

Theorem 1. *Let $p \in (1, \infty)$, let $s, t \in V$ be distinct vertices, and let $\Gamma = \Gamma(s, t)$ be the family of paths connecting s to t in G with $\mathcal{N}$ defined as in (1.1). Let ρ^* be the unique optimal density for p-modulus, and let ϕ^* be the unique optimal potential for (1.4). Then*

$$\rho^*(e) = |\phi^*(x) - \phi^*(y)| \quad \textit{for all } e = \{x, y\} \in E.$$

Consequently,

$$\mathrm{Mod}_{p,\sigma}(\Gamma) = \mathcal{E}_{p,\sigma}(\rho^*) = \sum_{\{x,y\}\in E} \sigma(\{x,y\})|\phi^*(x) - \phi^*(y)|^p.$$

In words, the p-modulus for the *family of connecting paths* $\Gamma = \Gamma(s,t)$ coincides with the minimum dissipated power of the nonlinear resistor network. Moreover, the optimal density $\rho^*(e)$ coincides with the absolute potential difference across the resistor associated with edge e.

Modulus as a metric

A related way of understanding modulus is through its connection to graph metrics. As described in [3], for any $p \in (1,\infty)$ with conjugate Hölder exponent $q = p/(p-1)$, the function $\delta_p : V \times V \to \mathbb{R}$ defined as

$$\delta_p(x,y) = \begin{cases} 0 & \text{if } x = y, \\ \mathrm{Mod}_{p,\sigma}(\Gamma(x,y))^{-\frac{q}{p}} & \text{if } x \neq y \end{cases}$$

is a metric on the vertices. In the special case $p = q = 2$, this metric coincides with the effective resistance metric due to the connection established above. Moreover, it was shown that when $p = 1$, $\mathrm{Mod}_{1,\sigma}(\Gamma(x,y))^{-1}$ coincides with the minimum cut metric (with edge capacities provided by σ) and when $p = \infty$, $\mathrm{Mod}_{\infty,\sigma}(\Gamma(x,y))^{-1}$ coincides with the shortest-path metric with edge lengths provided by σ^{-1}.

The probabilistic interpretation of modulus and expected overlap

Finally, we review briefly the probabilistic interpretation of p-modulus developed in [3, 7]. Adopting the notation of the references, we fix a family Γ and consider a random object $\underline{\gamma} \in \Gamma$. (The underline notation is used to distinguish the random object $\underline{\gamma}$ from its possible instantiations γ.) This random object is modeled through its distribution, or law, which takes the form of a probability mass function (pmf) $\mu : \Gamma \to \mathbb{R}$ that assigns to each object

in Γ its probability of selection. That is, we think of μ as the probability

$$\mu(\gamma) = \mathbb{P}_\mu(\underline{\gamma} = \gamma),$$

with the subscripted μ indicating that the probability depends on the choice of distribution. We denote by $\mathcal{P}(\Gamma)$ the set of all possible pmfs on Γ. That is,

$$\mathcal{P}(\Gamma) = \left\{\mu \in \mathbb{R}^\Gamma_{\geq 0} : \mu^T \mathbf{1} = 1\right\}.$$

We shall use the notation $\underline{\gamma} \sim \mu$ to indicate the relationship between a random variable and its underlying pmf.

For a random object $\underline{\gamma} \sim \mu$ and a fixed edge $e \in E$, the quantity $\mathcal{N}(\underline{\gamma}, e)$ is a real-valued random variable. Its expectation, called the *expected edge usage* for edge e plays a special role in this interpretation of modulus, and is identified by the symbol $\eta(e)$. That is,

$$\eta(e) = \mathbb{E}_\mu(\mathcal{N}(\underline{\gamma}, e)) = \sum_{\gamma \in \Gamma} \mathcal{N}(\gamma, e)\mu(\gamma) = (N^T\mu)(e).$$

In the special case that $\mathcal{N}$ takes the form (1.1), $\eta(e)$ can be understood as a marginal probability:

$$\eta(e) = \sum_{\gamma \in \Gamma} \mathbb{1}_{e \in \gamma}\mu(\gamma) = \mathbb{P}_\mu(e \in \underline{\gamma}).$$

The probabilistic interpretation of modulus developed in [3, 7] is summarized in the following theorem.

Theorem 2. *Let Γ be a family of objects on G and let $p \in (1, \infty)$ and $q = p/(p-1)$. Define the conjugate edge weights $\hat{\sigma}(e) = \sigma(e)^{-\frac{q}{p}}$. Then,*

$$\mathrm{Mod}_{p,\sigma}(\Gamma) = \left(\min_{\substack{\mu \in \mathcal{P}(\Gamma) \\ \eta = \mathcal{N}^T\mu}} \sum_{e \in E} \hat{\sigma}(e)\eta(e)^q\right)^{-\frac{p}{q}} = \left(\min_{\substack{\mu \in \mathcal{P}(\Gamma) \\ \eta = \mathcal{N}^T\mu}} \mathcal{E}_{q,\hat{\sigma}}(\eta)\right)^{-\frac{p}{q}}.$$

Moreover, while there may be infinitely many optimal pmfs μ^* for the right-hand side, there

is a unique optimal $\eta^* = \mathcal{N}^T\mu^*$ *that is related to the optimal* ρ^* *from p-modulus as follows.*

$$\frac{\sigma(e)\rho^*(e)^p}{\mathcal{E}_{p,\sigma}(\rho^*)} = \frac{\hat{\sigma}(e)\eta^*(e)^q}{\mathcal{E}_{q,\hat{\sigma}}(\eta^*)}.$$

When $\sigma \equiv 1$, $p = 2$ and $\mathcal{N}$ has the form (1.1), this probabilistic interpretation has a particularly intuitive meaning. Given any pmf μ, we may consider two independent, identically distributed random objects $\underline{\gamma}, \underline{\gamma}' \sim \mu$. If we think of these two objects as subsets of E, then it can be seen that

$$\sum_{e\in E} \eta(e)^2 = \mathbb{E}_\mu(|\underline{\gamma} \cap \underline{\gamma}'|),$$

the expected size of the intersection between the two objects. This is often called the *expected overlap* of the objects. In this setting, the p-modulus problem is equivalent to choosing a pmf μ so that this expected overlap is minimized.

There are also versions of Theorem 2 for the cases $p \in \{1, \infty\}$. These are necessarily weaker due to lack of uniqueness.

Theorem 3. *Let* Γ *be a family of objects on* G*, then*

$$\mathrm{Mod}_{1,\sigma}(\Gamma)^{-1} = \min_{\substack{\mu\in\mathcal{P}(\Gamma)\\ \eta=\mathcal{N}^T\mu}} \mathcal{E}_{\infty,\sigma^{-1}}(\eta) \qquad \textit{and} \qquad \mathrm{Mod}_{\infty,\sigma}(\Gamma)^{-1} = \min_{\substack{\mu\in\mathcal{P}(\Gamma)\\ \eta=\mathcal{N}^T\mu}} \mathcal{E}_{1,\sigma^{-1}}(\eta).$$

Using the probabilistic interpretation, we can gain more insight into the earlier observation that modulus measures a balance between diversity and shortness. To see this, consider an arbitrary family Γ and note that the constant density $\rho = (\min \mathcal{N}\mathbf{1})^{-1}\mathbf{1}$ (the reciprocal of the minimum row sum of $\mathcal{N}$) is admissible. For $p \in (1, \infty)$, this implies that

$$\mathrm{Mod}_{p,\sigma}(\Gamma) \leq \frac{\sigma(E)}{(\min \mathcal{N}\mathbf{1})^p},$$

where

$$\sigma(E) = \sum_{e\in E} \sigma(e).$$

When $\mathcal{N}$ has large row sums (corresponding to "long" or "heavy" objects), this forces the modulus to be small.

On the other hand, the uniform distribution $\mu = |\Gamma|^{-1}\mathbf{1}$ is one possible pmf for Theorem 2, which provides a lower bound on modulus. With this choice,

$$\eta(e) = |\Gamma|^{-1}(\mathcal{N}^T\mathbf{1})(e) \leq |\Gamma|^{-1} \max \mathcal{N}^T\mathbf{1},$$

where $\max \mathcal{N}^T\mathbf{1}$ is the maximum column sum of $\mathcal{N}$. Thus,

$$\mathrm{Mod}_{p,\sigma}(\Gamma)^{-1} \leq \frac{\hat{\sigma}(E)^{\frac{p}{q}}}{|\Gamma|^p}(\max \mathcal{N}^T\mathbf{1})^p.$$

When $\mathcal{N}$ has small column sums (corresponding to edges that are not used by many objects), this forces the modulus to be large.

Chapter 2

Temporal p-Modulus on graphs

Having reviewed the definition of modulus and some of its interpretations, we now focus our attention on extending the p-modulus framework to objects on temporal graphs.

2.0.1 A model for temporal networks

In order to demonstrate the use of p-modulus on temporal networks, we must first choose a suitable network model. For this dissertation, we have chosen the "contact sequence" graph described in [12, Sec. 3]. Conceptually, this type of graph can be represented as a quadruple $G = (V, E, \sigma, T)$, where the added property T encodes the temporal information. For this dissertation, we will consider T as a function $T : E \to 2^{\mathbb{R}}$; to each edge, $e \in E$, $T(e)$ assigns a discrete, positive set of times $t \in \mathbb{R}_{>0}$ during which e is available for use. To simplify the discussion, we shall assume that $T(e)$ is finite for each e. The associated graph $\check{G} = (V, E, \sigma)$ with temporal information stripped away is often called the *static graph* or *aggregated graph*.

2.0.2 Objects on temporal networks

The next step in extending p-modulus to temporal networks is to select a family of objects, Γ. Intuitively, each object should be somehow associated with the graph G and should take into account the temporal information. As in the non-temporal setting, it is evident that there is a great deal of flexibility in defining an object. A fairly broad class of families can

be obtained by considering subsets of edge-time pairs. A concrete example of such a family is the family of *time respecting paths* between s and t as described in [12, Sec. 4.2]. These are sets of the form $\{(e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k)\}$ where the edges $e_1 e_2 \cdots e_k$ form a path (traversed in order) from s to t, and the times satisfy $t_1 < t_2 < \cdots < t_k$ and $t_i \in T(e_i)$ for all i, k is the number of hops in this path. In this example, one sees the connection between the underlying graph structure (the edges must trace a path from s to t) and the temporal data (the edges must be traversed in a prescribed temporal order).

2.0.3 Density and energy

There are two relatively natural ways to define the concept of density on a temporal graph. The first is to assign a value $\rho(e)$ to each edge $e \in E$. The second is to assign a value $\rho(e, t)$ to each edge-time pair. A practical way to accommodate either option, or even a mixture of both, is to allow G to have parallel edges. In other words, G (and, therefore, $\check{G}$) is a multigraph with edge set E. Each edge e has a set $T(e)$ of one or more availability times and is assigned a value $\rho(e)$ by a density $\rho \in \mathbb{R}^E_{\geq 0}$. In this way, the definition of energy (1.2) remains unchanged in the temporal case. All that remains in order to define a temporal version of modulus, then, is to find a suitable definition for the usage matrix $\mathcal{N} \in \mathbb{R}^{\Gamma \times E}$.

2.0.4 Usage for temporal objects

Here, we describe possible strategies for defining a temporal version of usage, assuming a particular structure on the family of temporal objects. Our goal is not to develop a single generic approach for all families of temporal objects, but instead to provide the basis for a temporal p-modulus framework by way of example.

Specifically, we shall consider a family Γ whose objects can be identified with sets of the form

$$\gamma = \{(e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k)\} \quad \text{with} \quad t_i \in T(e_i) \text{ for } i = 1, 2, \ldots, k.$$

In words, the objects we consider will be sets of edge-time pairs with the requirement that

the time occurring in the pair is one of the available times of the corresponding edge. To simplify the discussion, we shall also assume that no edge appears more than once in a given object (i.e., that the edges e_i are unique in γ). This class of family is a natural generalization of the time respecting paths.

Associated with each object $\gamma \in \Gamma$ is a natural projection $\check{\gamma} \in 2^E$ defined as

$$\check{\gamma} = \{e_1, e_2, \ldots, e_k\}.$$

Since we assumed that γ cannot have repeated edges, it follows that $|\check{\gamma}| = |\gamma|$. In this way, we can think of $\check{\gamma}$ as a projected object on the aggregated graph $\check{G}$. The collection of all projections of temporal objects in Γ forms a family of objects on $\check{G}$, which we denote $\check{\Gamma}$. That is,

$$\check{\Gamma} = \{\check{\gamma} : \gamma \in \Gamma\}.$$

Since each $\check{\gamma} \in \check{\Gamma}$ is a subset of E, a natural choice of usage matrix for this aggregated family is $\check{\mathcal{N}}$ of the form (1.1). We seek to define a usage matrix $\mathcal{N}$ on Γ by incorporating temporal information into $\check{\mathcal{N}}$. We shall do this by means of a strictly monotonic *temporal penalty function* (or *time-penalty function*) $\varphi : \mathbb{R} \to \mathbb{R}_{\geq 0}$. In general, we will think of φ as monotonically increasing, though it is possible to envision applications wherein a monotonically decreasing φ would be of interest.

Per-edge time penalization

In order to find a suitable usage matrix, one might consider modifying the concept of ρ-length by incorporating temporal information independently on each edge. For example, given a density ρ and an object $\gamma \in \Gamma$, one could define

$$\ell_\rho(\gamma) = \sum_{i=1}^{k} f(\rho(e_i), \varphi(t_i)).$$

In order to apply the established p-modulus apparatus to the family of temporal objects, we need to find a usage matrix $\mathcal{N} \in \mathbb{R}_{\geq 0}^{\Gamma \times E}$ with the property that

$$\ell_\rho(\gamma) \geq 1 \quad \iff \quad (\mathcal{N}\rho)(\gamma) \geq 1.$$

There is no reason to expect such a matrix to exist in general, but for certain choices of f it does.

For example, if f is the multiplication operator, then

$$\ell_\rho(\gamma) = \sum_{i=1}^{k} \rho(e_k)\varphi(t_k) = \sum_{e \in E} \mathcal{N}(\gamma, e)\rho(e),$$

where

$$\mathcal{N}(\gamma, e) = \begin{cases} \varphi(t) & \text{if } (e,t) \in \gamma, \\ 0 & \text{if } e \notin \check{\gamma}. \end{cases} \tag{2.1}$$

If f instead is the addition operator then

$$\ell_\rho(\gamma) = \ell_\rho(\check{\gamma}) + \sum_{i=1}^{k} \varphi(t_k).$$

Requiring that $\ell_\rho(\gamma) \geq 1$ is equivalent to requiring that

$$\ell_\rho(\check{\gamma}) \geq 1 - \sum_{i=1}^{k} \varphi(t_k).$$

Since ρ and φ are non-negative, we may assume without loss of generality (by possibly removing objects from Γ) that the right-hand side of the inequality is strictly positive for all $\check{\gamma} \in \check{\Gamma}$. If we define

$$\varphi(\gamma) = \sum_{i=1}^{k} \varphi(t_k),$$

then the minimum ρ-length constraint can be written as

$$1 \leq \frac{\ell_\rho(\check{\gamma})}{1-\varphi(\gamma)} = \sum_{e\in E} \frac{\check{\mathcal{N}}(\check{\gamma}, e)}{1-\varphi(\gamma)}\rho(e),$$

which provides the temporal usage matrix

$$\mathcal{N}(\gamma, e) = \frac{\check{\mathcal{N}}(\check{\gamma}, e)}{1-\varphi(\gamma)}.$$

Per-object time penalization

Alternatively, one might assign a time penalty to the object as a whole and combine this with ρ-length on the aggregated graph. As an example, we could define a length of the form

$$\ell_\rho(\gamma) = f(\ell_\rho(\check{\gamma}), \varphi(\gamma)), \qquad \text{where} \qquad \varphi(\gamma) = \max_i \varphi(t_i).$$

Natural assumptions on f would be non-negativity and monotonicity in each argument—larger ρ-length in the aggregated graph and larger time penalties both induce larger temporal ρ-length. Moreover, in order to avoid the trivial minimizer $\rho^* \equiv 0$ and to ensure that the set of admissible densities is non-empty, we will assume the following.

1. For all $\gamma \in \Gamma$, $f(0, \varphi(\gamma)) < 1$
2. For all $\gamma \in \Gamma$, $f(\ell, \varphi(\gamma)) \geq 1$ for sufficiently large ℓ.

If we further require that f be right continuous in its first argument, then to every $\gamma \in \Gamma$ we may assign a critical length $\ell^*(\gamma) > 0$ so that

$$\ell_\rho(\gamma) \geq 1 \quad \iff \quad \ell_\rho(\check{\gamma}) \geq \ell^*(\gamma).$$

In this case, the temporal usage matrix, $\mathcal{N}$, that we seek can be found from the minimum

length inequalities, since $\ell_\rho(\gamma) \geq 1$ if and only if

$$1 \leq \frac{\ell_\rho(\check{\gamma})}{\ell^*(\gamma)} = \sum_{e \in E} \frac{\check{\mathcal{N}}(\check{\gamma}, e)}{\ell^*(\gamma)} \rho(e),$$

so we should choose

$$\mathcal{N}(\gamma, e) = \frac{\check{\mathcal{N}}(\check{\gamma}, e)}{\ell^*(\gamma)}.$$

In the particular case that f is the multiplication operator, we obtain

$$\mathcal{N}(\gamma, e) = \varphi(\gamma) \check{\mathcal{N}}(\check{\gamma}, e). \tag{2.2}$$

If f is the addition operator, then the usage matrix is

$$\mathcal{N}(\gamma, e) = \frac{\check{\mathcal{N}}(\check{\gamma}, e)}{1 - \varphi(\gamma)},$$

under the additional assumption that $\varphi(\gamma) < 1$ for all $\gamma \in \Gamma$.

Some general assumptions

All of the above possibilities can be realized by choosing $\mathcal{N}$ of the form

$$\mathcal{N}(\gamma, e) = \psi(\gamma, e) \check{\mathcal{N}}(\check{\gamma}, e), \tag{2.3}$$

where ψ performs some type of "temporal weighting." For the remainder of this d, we shall consider this as a general form of temporal usage matrix.

By making some basic assumptions on this function ψ, it is possible to translate a large portion of the modulus theory developed for non-temporal graphs to the temporal setting. For the remainder of this dissertation, we shall make the following assumptions about ψ.

Assumption 1. The function ψ is *universal* in the sense that it is defined for any set γ of

the form

$$\gamma = \{(e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k)\} \quad \text{with} \quad e_i \in E \text{ and } t_i > 0 \text{ for } i = 1, 2, \ldots, k,$$

and any $e \in E$. This allows us to consider ψ separately from the particular choice of temporal object family Γ.

Assumption 2. The function ψ is monotonic with respect to inclusions. If $\gamma' \subseteq \gamma$, then $\psi(\gamma', e) \leq \psi(\gamma, e)$.

Assumption 3. The function ψ is monotonic with respect to time. If $\gamma = \{(e_1, t_1), \cdots, (e_k, t_k)\}$ and $\gamma' = \{(e_1, t'_1), \cdots, (e_k, t'_k)\}$ with $t'_i \leq t_i$ for all i, then $\psi(\gamma', e) \leq \psi(\gamma, e)$.

Assumption 4. If $\gamma^\lambda = \{(e_1, \lambda t_1), \cdots, (e_k, \lambda t_k)\}$ (parametrized by $\lambda > 0$) then the function $\lambda \mapsto \psi(\gamma^\lambda, e)$ is continuous for each e and

$$\lim_{\lambda \to 0^+} \psi(\gamma^\lambda, e) = 1.$$

For the usage examples described previously, these assumptions can be satisfied by suitable assumptions on the function φ. In the following, we consider a generic set $\gamma = \{(e_1, t_1), \ldots, (e_k, t_k)\}$.

Multiplicative per-edge penalization. In this case,

$$\psi(\gamma, e) = \begin{cases} \varphi(t) & \text{if } (e, t) \in \gamma, \\ 1 & \text{if } e \notin \check{\gamma}. \end{cases} \tag{2.4}$$

The assumptions are met if φ is a continuous, increasing function satisfying $\varphi(0) = 1$.

Additive per-edge penalization. In this case,

$$\psi(\gamma, e) = \begin{cases} \left(1 - \sum_{i=1}^{k} \varphi(t_k)\right)^{-1} & \text{if } e \in \check{\gamma}, \\ 1 & \text{if } e \notin \check{\gamma}. \end{cases}$$

(Special care must be taken if the sum equals or exceeds 1.) The assumptions are met if φ is a continuous, increasing function satisfying $\varphi(0) = 0$.

Multiplicative per-object penalization. In this case,

$$\psi(\gamma, e) = \begin{cases} \max \varphi(t_i) & \text{if } e \in \check{\gamma}, \\ 1 & \text{if } e \notin \check{\gamma}. \end{cases} \tag{2.5}$$

The assumptions are met if φ is a continuous, increasing function satisfying $\varphi(0) = 1$.

Additive per-object penalization. In this case,

$$\psi(\gamma, e) = \begin{cases} (1 - \max \varphi(t_i))^{-1} & \text{if } e \in \check{\gamma}, \\ 1 & \text{if } e \notin \check{\gamma}. \end{cases}$$

(Special care must be taken if the maximum equals or exceeds 1.) The assumptions are met if φ is a continuous, increasing function satisfying $\varphi(0) = 0$.

2.0.5 Modulus of a family of temporal objects

Using the procedure outlined above, we have now developed the concepts of density and energy on a temporal graph G and the concept of usage on a temporal object $\gamma \in \Gamma$. With these definition in hand, the p-modulus of Γ can be defined exactly as in (1.3). In this way, we have incorporated all needed temporal information into the usage matrix $\mathcal{N}$ and, when it is convenient, we may think of Γ as a family of objects on the aggregated graph $\check{G}$. This

allows properties of modulus to translate more or less directly from the non-temporal to the temporal case.

2.1 A mass transport interpretation of modulus

To help provide some intuition about what temporal modulus measures, we now describe a mass transport interpretation based on the probabilistic interpretation of modulus described in Section 1.3.3. For this discussion, we focus on the family $\Gamma = \Gamma^{\mathrm{trp}}(s,t)$ of time respecting paths passing from a vertex s to another vertex t in a temporal graph G. Note that, in this case, the projected family $\check{\Gamma}$ is a subset of $\Gamma(s,t)$, the family of paths connecting s to t in $\check{G}$. This inclusion is strict if there are paths $\check{\gamma} \in \check{\Gamma}$ that cannot be traced while respecting the temporal constraint. Moreover, $|\check{\Gamma}| \leq |\Gamma|$, and this inequality is strict if there are multiple ways to trace some path while respecting time. For a temporal usage matrix, we adopt an $\mathcal{N}$ of the form (2.3), with $\check{\mathcal{N}}$ as in (1.1).

Every pmf $\mu \in \mathcal{P}(\Gamma)$ can be thought of as a *plan* for transporting one unit of mass along the time respecting paths from s to t. For any path $\gamma \in \Gamma$, $\mu(\gamma)$ provides the fraction of mass that should be sent along γ. In this way, we can view temporal modulus within the class of optimal mass transport problems, with p, σ and ψ defining the fitness function as follows.

First, every plan incurs a *per edge usage cost*

$$\eta(e) = \sum_{\gamma \in \Gamma} \mathcal{N}(\gamma, e)\mu(\gamma) = \sum_{\gamma \in \Gamma} \psi(\gamma, e)\check{\mathcal{N}}(\gamma, e)\mu(\gamma) = \sum_{\gamma \in \Gamma_e} \psi(\gamma, e)\mu(\gamma),$$

where $\Gamma_e = \{\gamma \in \Gamma : e \in \check{\gamma}\}$ is the subfamily of paths that traverse edge e. The quantity $\psi(\gamma, e)$ can be thought of as an amount of some resource per unit mass consumed when transporting this unit mass along edge e of γ. Examples of ψ can be found in Section 2.0.4. Since the temporal penalty function φ is assumed to be increasing, paths that cross edges a later times tend to consume more resource per unit mass than paths traversing edges at earlier times. The quantity $\eta(e)$, then, is the accumulated resource expenditure on edge e

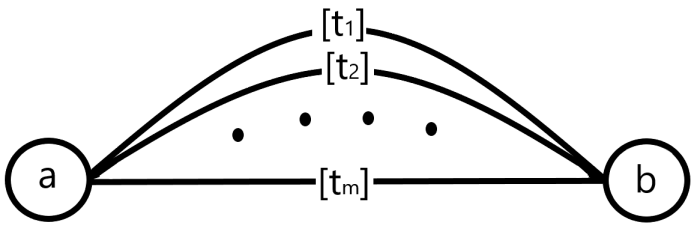

Figure 2.1: *Example 2.1. Multi edge simple graph.*

required to enact the plan μ.

The values p and σ then define the global fitness function $\mathcal{E}_{q,\hat{\sigma}}(\eta)$ that we seek to minimize by choosing the plan μ. For example, if we simply wish to minimize the total resource expenditure (i.e., the sum of $\eta(e)$ across all edges), this corresponds to setting $p = \infty$ and $\sigma \equiv 1$. On the other hand, if we wish to minimize the maximum resource expenditure across all edges, then we would set $p = 1$ and $\sigma \equiv 1$. More generally, $\sigma(e)$ allows us to adjust the importance of edge e within the fitness function and p provides us a way of interpolating between the extremes of minimizing the sum of $\sigma^{-1}(e)\eta(e)$ and minimizing the maximum of $\sigma^{-1}(e)\eta(e)$. In particular, 2-modulus corresponds to minimizing the sum of $\sigma^{-1}(e)\eta(e)^2$ across all edges and bears a striking resemblance to effective resistance if we think of $\eta(e)$ as a generalized "current" flowing across edge e.

2.2 Examples

The following examples demonstrate the ability for p-modulus to incorporate temporal information. In the examples, we will consider the two types of temporal usage matrix, those corresponding to the the per-edge multiplicative penalty (2.4) and the multiplicative per-object penalty (2.5).

Example 1. Consider a graph G with two vertices, a and b, and a collection of parallel edges $\{e_1, e_2, \ldots, e_m\}$, each with a single available time $T(e_i) = \{t_i\}$ (see Figure 2.1). In this

case, (2.4) and (2.5) give the same usage matrix. The length constraints on ρ are

$$\varphi(t_i)\rho(e_i) \geq 1 \quad i = 1, 2, \ldots, m.$$

Since we seek to minimize $\mathcal{E}_{p,\sigma}(\rho)$, it is straightforward to check that $\rho^*(e_i) = \varphi(t_i)^{-1}$ is an optimal density, so

$$\mathrm{Mod}_{p,\sigma}(\Gamma^{\mathrm{trp}}(a,b)) = \begin{cases} \sum_{i=1}^{m} \frac{\sigma(e_i)}{\varphi(t_i)^p} & \text{if } p < \infty, \\ \max_i \frac{\sigma(e_i)}{\varphi(t_i)} & \text{if } p = \infty. \end{cases}$$

Each object in this example is a single edge-time pair, $\gamma_i = \{(e_i, t_i)\}$ for $i = 1, 2, \ldots, m$. From Theorem 2 one can see that the optimal mass transport plan when $p \in (1, \infty)$ assigns γ_i a mass proportional to $\sigma(e_i)\varphi(t_i)^{-\frac{p}{q}}$. For $p \approx 1$, this mass is primarily determined by the relative sizes of the $\sigma(e_i)$—larger capacity edges will be assigned more mass regardless the availability time. For $p \gg 1$, it is primarily determined by the relative sizes of the $\varphi(t_i)$—edges that are available at early times will be assigned more mass regardless their capacity.

Example 2. Consider a path graph P_n with n vertices and $\ell = n - 1$ edges $\{e_1, e_2, \ldots, e_\ell\}$ which, when traversed in order, connect the vertex a to b. Assume that each edge has a single available time $T(e_i) = \{t_i\}$ and that $t_1 < t_2 < \cdots < t_\ell$ so that $\Gamma^{\mathrm{trp}}(a,b)$ contains a single time respecting path $\gamma = \{(e_1, t_1), (e_2, t_2), \ldots, (e_\ell, t_\ell)\}$. Since there is a single path, the optimal mass transport plan is to assign all mass to this path. Thus, the optimal expected edge usage is $\eta^*(e) = \mathcal{N}(\gamma, e) = \psi(\gamma, e)$. For $p \in (1, \infty)$, Theorem 2 shows that

$$\mathrm{Mod}_{p,\sigma}(\Gamma^{\mathrm{trp}}(a,b)) = \left(\sum_{i=1}^{\ell} \hat{\sigma}(e_i)\psi(\gamma, e_i)^q \right)^{-\frac{p}{q}}.$$

If we choose per-edge temporal penalization, then

$$\mathrm{Mod}_{p,\sigma}(\Gamma^{\mathrm{trp}}(a,b)) = \left(\sum_{i=1}^{\ell} \hat{\sigma}(e_i)\varphi(t_i)^q \right)^{-\frac{p}{q}}.$$

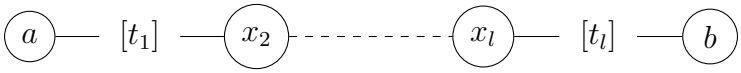

Figure 2.2: *Example 2. Consecutive connection graph.*

On the other hand, if we choose per-object temporal penalization, then

$$\mathrm{Mod}_{p,\sigma}(\Gamma^{\mathrm{trp}}(a,b)) = \varphi(t_\ell)^{-p}\left(\sum_{i=1}^{\ell}\hat{\sigma}(e_i)\right)^{-\frac{p}{q}}.$$

Modulus decreases as the available times of the edges in the path increase.

Example 3 (Parallel Paths Graph). As a temporal version of the example in Section 1.3.2 (see Figure 2.3), consider k identical copies of the path graph described in the previous example, all glued together at common end vertices a and b. From the symmetry of the configuration, it is evident that each of the k paths will be assigned a mass of $1/k$ by the optimal transport plan, so for $e \in \gamma$, $\eta^*(e) = \psi(\gamma, e)/k$. Applying Theorem 2 once again shows that

$$\mathrm{Mod}_{p,\sigma}(\Gamma^{\mathrm{trp}}(a,b)) = \left(k\sum_{i=1}^{r}\hat{\sigma}(e_i)\frac{\psi(\gamma,e_i)^q}{k^q}\right)^{-\frac{p}{q}}.$$

If $\sigma \equiv 1$, then per-edge penalization and per-object penalization respectively yield

$$\mathrm{Mod}_{p,1}(\Gamma^{\mathrm{trp}}(a,b)) = k\left(\sum_{i=1}^{r}\varphi(t_i)^q\right)^{-\frac{p}{q}} \quad \text{and} \quad \mathrm{Mod}_{p,1}(\Gamma^{\mathrm{trp}}(a,b)) = \frac{k}{r^{p-1}}\varphi(t_r)^{-p},$$

showing the triple interaction among the number of paths, the (graph) lengths of the paths, and the time required to traverse the paths on the temporal graph.

From [2] we also know that p-modulus is a continuous function w.r.t. p.

Since φ is non-decreasing, 1-modulus for both per-edge penalization and per-object one is

$$\mathrm{Mod}_{1,1}(\Gamma^{\mathrm{trp}}(a,b)) \overset{Cont.}{=\!=} \lim_{p\to 1}\mathrm{Mod}_{p,1}(\Gamma^{\mathrm{trp}}(a,b)) = k\varphi(\tau_r)^{-1},$$

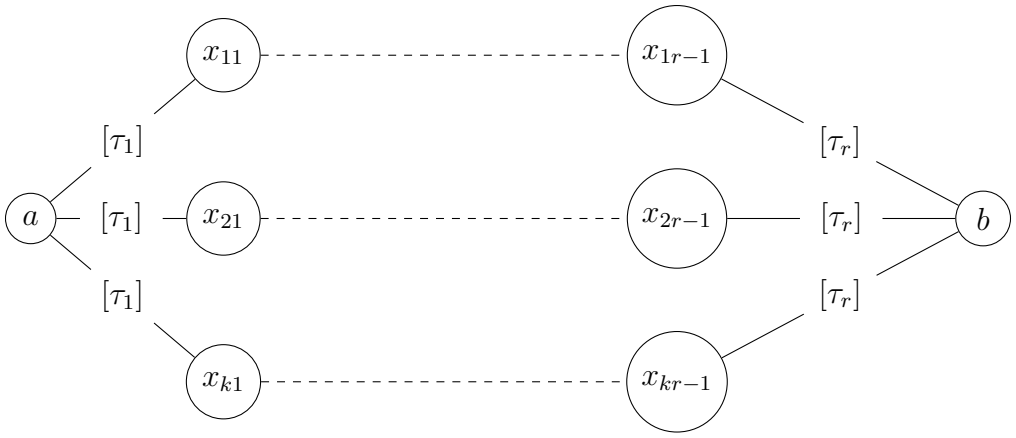

Figure 2.3: *Example 3. Parallel paths graph.*

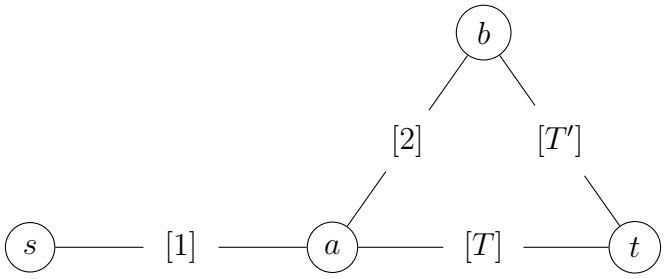

Figure 2.4: *Example 4. Paw graph with* $T' > 2,\ \ T > 1$

The ∞-modulus for per-edge penalization is

$$\mathrm{Mod}_{\infty,1}(\Gamma^{\mathrm{trp}}(a,b)) \overset{Cont.}{=\!=} \lim_{p\to\infty} \mathrm{Mod}_{p,1}(\Gamma^{\mathrm{trp}}(a,b))^{\frac{1}{p}} = \left(\sum_{i=1}^{r} \varphi(\tau_i)\right)^{-1}.$$

Since $k, r \geq 1$, ∞-Modulus for object-edge penalization is

$$\mathrm{Mod}_{\infty,1}(\Gamma^{\mathrm{trp}}(a,b)) \overset{Cont.}{=\!=} \lim_{p\to\infty} \mathrm{Mod}_{p,1}(\Gamma^{\mathrm{trp}}(a,b))^{\frac{1}{p}} = \frac{1}{r}\varphi(\tau_r)^{-1}.$$

We also can conclude that optimal maximizer for the dual problem is the vector $\mu^* = \frac{1}{k}\mathbf{1}$.

Example 4 (Paw Graph). Let's compute 2-Modulus on a set of time-respecting paths Γ from s to t in the graph on Figure 2.4, where all weights on the edges are assumed to be 1, and $\varphi(t)$ is a non-decreasing positive per-object penalization.

For convenience we denote $\varphi(T) =: \alpha,\ \ \varphi(T') =: \beta$. First, we find the usage matrix $\mathcal{N}$ of

the temporal graph. The time-dependent distance of a path ω is

$$l^{\tau}_{\rho}(\omega) = \varphi(\tau(\omega))l_{\rho}(\omega) = \varphi(\tau(\omega))\sum_{e}\check{\mathcal{N}}\rho(e)$$

The essential [7] set of paths consists only of two objects: $\Gamma^{\text{tmp}} = \{s \to a \to t,\ s \to a \to b \to t\}$. And so the temporal usage matrix can be found through the usage matrix of the static part of the graph as

$$\sum_{e}\mathcal{N}(\omega, e)\rho(e) = \sum_{e}\varphi(\tau(\omega))\check{\mathcal{N}}(\omega, e)\rho(e)$$

Thus,

$$\mathcal{N} = \begin{bmatrix} \alpha & \alpha & 0 & 0 \\ \beta & 0 & \beta & \beta \end{bmatrix}$$

Let $\mu = \begin{bmatrix} \delta \\ 1-\delta \end{bmatrix}$, where $0 \leq \delta \leq 1$. Then

$$\text{Mod}_{2,1}(\Gamma^{\text{trp}}(s,t)) = \left(\min_{\mu\in\mathcal{P}(\Gamma)}(\mu^T\mathcal{N}\left(\mathcal{N}\right)^T\mu)\right)^{-1}$$

$$= \left(\min_{\mu\in\mathcal{P}(\Gamma)}(\mu^T\begin{bmatrix} 2\alpha^2 & \alpha\beta \\ \alpha\beta & 3\beta^2 \end{bmatrix}\mu)\right)^{-1} = \left(\min_{\delta\in[0,1]}(2\alpha^2\delta^2 + 2\alpha\beta(1-\delta)\delta + (1-\delta)^2 3\beta^2\right)^{-1}. \quad (2.6)$$

In order to minimize this value by δ we take the derivative with respect to δ and set it equal to zero:

$$2(2\alpha^2 + 3\beta^2 - 2\alpha\beta)\delta + 2\alpha\beta - 6\beta^2 = 0.$$

Solving this now for δ we get

$$\delta^* = \frac{\beta(3\beta - \alpha)}{2\alpha^2 - 2\alpha\beta + 3\beta^2}.$$

Notice that if $3\beta < \alpha$ then $\delta^* = 0$ and if $\beta(3\beta - \alpha) > 2\alpha^2 - 2\alpha\beta + 3\beta^2$, which is $2\alpha < \beta$

then $\delta^* = 1$.

So, the value of $\mathrm{Mod}_2(\Gamma^{\mathrm{trp}}(\mathrm{s},\mathrm{t}))$ can be found by substitution of this optimal δ^* to (2.6):

$$\mathrm{Mod}_{2,1}(\Gamma^{\mathrm{trp}}(s,t)) = \begin{cases} (3\beta^2)^{-1}, \text{ if } \delta^* = 0 \ (3\beta < \alpha) \\ (2\alpha^2)^{-1}, \text{ if } \delta^* = 1 \ (3\alpha < \beta) \\ \frac{(2\alpha^2 - 2\alpha\beta + 3\beta^2)^2}{2\alpha^2\beta^2[(2\beta-\alpha)^2 + (2\beta-\alpha)(2\alpha-\beta) + (3/2)(2\alpha-\beta)^2]}, \text{ if } 0 < \delta^* < 1 \ (\beta/2 \le \alpha \le 3\beta) \end{cases}$$

The optimal density by Theorem 2 is

$$\rho^* = \mathrm{Mod}_{2,1}(\Gamma^{\mathrm{trp}}(s,t))(\mathcal{N})^T \mu^*$$

Case $p = 1$:

Let's compute temporal 1-Modulus.

$$\mathrm{Mod}_{1,1}(\Gamma^{\mathrm{trp}}(s,t))^{-1} = \min_{\mu \in \mathcal{P}(\Gamma^{\mathrm{trp}}(s,t))} \max_{e \in E} (\mathcal{N}(\cdot, e))^T \mu$$

$$= \min_{\mu \in \mathcal{P}(\Gamma^{\mathrm{trp}}(s,t))} \max_{e \in E} \{\alpha\delta + \beta(1-\delta), \ \alpha\delta, \ \beta(1-\delta), \ \beta(1-\delta)\} = \min_{\delta \in [0,1]} \{\alpha\delta + \beta(1-\delta)\}$$

Thus,

$$\mathrm{Mod}_{1,1}(\Gamma^{\mathrm{trp}}(s,t))^{-1} = \begin{cases} \beta, \text{ if } \alpha > \beta \text{ and } \ \delta = 0 \\ \alpha, \text{ if } \alpha \le \beta \text{ and } \ \delta = 1 \end{cases}$$

The minimizer ρ^* is not unique. For example, it can be $\rho^* = (1/\min(\alpha,\beta), 0, 0, 0)^T$.

Case $p = \infty$:

$$\mathrm{Mod}_{\infty,1}(\Gamma^{\mathrm{trp}}(s,t))^{-1} = \min_{\mu \in \mathcal{P}(\Gamma^{\mathrm{trp}}(s,t))} \sum_{e \in E} (\mathcal{N}(\cdot, e))^T \mu = \min_{\delta \in [0,1]} \{2\alpha\delta + 3\beta(1-\delta)\}$$

$$= \begin{cases} 3\beta, \text{ if } \ 2\alpha > 3\beta \text{ and } \ \delta = 0 \\ 2\alpha, \text{ if } \ 2\alpha \le 3\beta \text{ and } \ \delta = 1 \end{cases}$$

One of optimal densities is $\rho^* = (1/\min(2\alpha, 3\beta), 1/2\alpha, 1/3\beta, 1/3\beta)^T$.

General p case:

$$\mathrm{Mod}_{p,\sigma}(\Gamma^{\mathrm{trp}}(s,t))^{-\frac{q}{p}} = \min_{\mu\in\mathcal{P}(\Gamma^{\mathrm{trp}}(s,t))} \sum_{e\in E} \eta^q = \min_{\mu\in\mathcal{P}(\Gamma^{\mathrm{trp}}(s,t))} \sum_{e\in E} \left((\mathcal{N}(\cdot,e))^T\mu\right)^q$$

$$= \min_{\delta\in[0,1]} \{[\alpha\delta+\beta(1-\delta)]^q + [\ \alpha\delta]^q + \ 2[\beta(1-\delta)]^q\}.$$

Applying Calculus, we have that the optimal δ is the solution of the equation:

$$(\alpha-\beta)[(\alpha-\beta)\delta+\beta]^{q-1} + [\alpha\beta]^{q-1} - 2\beta(\beta(1-\delta))^{q-1} = 0.$$

On the Figure 4 , one can see the set of the solutions. Observe that there is one blue horizontal line $\alpha/\beta = 3/2$ corresponds to the case $p = \infty$ and red curves converge to a horizontal line $\alpha/\beta = 1$, which corresponds to the case $p = 1$.

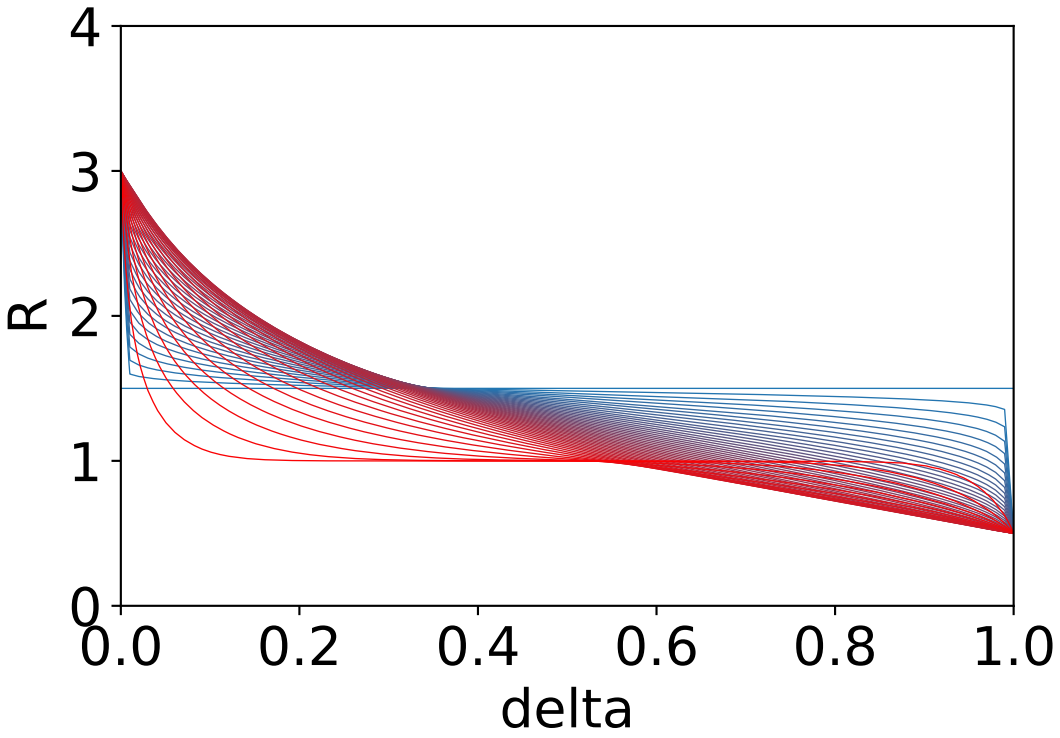

Figure 2.5: *Optimal δ for $R = \alpha/\beta$.*

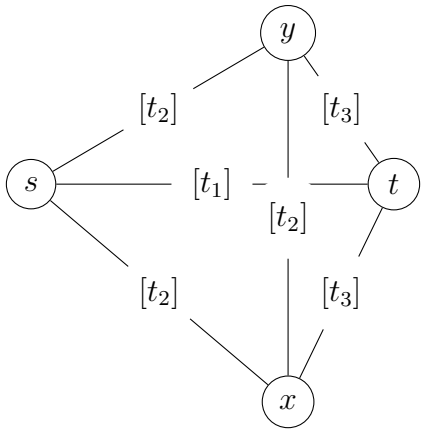

Figure 2.6: *Example 5. Complete graph.*

By Theorem 2, the optimal density can be found from:

$$(\rho_e^*)^{\frac{q}{p}} = \mathrm{Mod}_{p,1}(\Gamma^{\mathrm{trp}}(s,t))\mathcal{N}(\cdot,e)^T\mu^*$$

Example 5 (Complete Graph). Consider a complete graph K_n with n nodes (Figure 2.6). Choose some pair of nodes s and t to be source and sink, respectively. Assign the same weights on all edges $\sigma = 1$ and set of time windows T on edges in the following way:

1)$\tau^{(s,t)} = t_1$.

2) $\tau^{(x,y)} = t_2$, if the node $x \neq t$, node $y \neq t$ and $y \neq s$.

3) $\tau^{(x,t)} = t_3$, if the node $x \neq s$ and $x \neq t$.

Let's compute p-modulus for K_n on Γ, the set of time-respecting paths from s to t.

Only two types of paths $s \rightsquigarrow t$ can be optimal (best either by static length or arrival time):

1) The straight one-hop path from s to t with time t_1,

2) Any two-hop path from s to any node different from s and t with the time window t_2 then from this node to t with the time window t_3 (There exist $n-2$ of them.).

Any other path won't bring us to t faster than t_1 or t_3 and the travel cost will be strictly more than 2, i.e. it is dominated by previously described paths and so cannot be in the optimal set.

p-modulus is the solution of the minimization problem

$$\underset{\rho_1,\rho_2,\rho_3}{\text{minimize}}\ \rho_1^p + (n-2)(\rho_2^p + \rho_3^p),$$

where ρ_1 is the density on the edge with the assigned time window t_1, the density ρ_2 is on the edges going from s to any node, which is not s or t. The density ρ_3 belongs to any edge going to t from any node except s and t.

The constraints for the modulus problem with **per-edge penalization** $\varphi(t)$ are:

$$\rho_1\varphi(t_1) \geq 1,$$

$$\rho_2\varphi(t_2) + \rho_3\varphi(t_3) \geq 1.$$

If $t_2 \geq t_3$ then 2-hop (or more) temporal paths from s to t are impossible. Consequently, $\rho_2 = \rho_3 = 0$ and by Beurling (Theorem [8, Theorem 4.2]) we have only one condition

$$\rho_1 = \frac{1}{\varphi(t_1)}.$$

If $t_2 < t_3$, we apply KKT conditions:

$$p\rho_i^{p-1} = \lambda\varphi(t_i),\ i = 1, 2.$$

Since $p - 1 = \dfrac{q}{p}$, we obtain $\rho_i = \tilde{\lambda}\varphi^{\frac{q}{p}}(t_i),\ i = 1, 2$. Plugging it back into the constraint, we have $1 = \tilde{\lambda}(\varphi(t_2)^q + \varphi(t_3)^q)$. Thus,

$$\rho_i = \frac{\varphi(t_i)^{\frac{p}{p}}}{\varphi(t_2)^q + \varphi(t_3)^q},\ i = 1, 2.$$

The constraints for the modulus problem with **per-object penalization** $\varphi(t)$ are:

$$\rho_1\varphi(t_1) \geq 1,$$

$$(\rho_2 + \rho_3)\varphi(t_3) \geq 1.$$

If $t_2 > t_3$,

$$\rho_1 = \frac{1}{\varphi(t-1)}.$$

If $t_2 < t_3$ then by symmetry

$$\rho_2 = \rho_3 = \frac{1}{2\varphi(t_3)}.$$

Therefore, for both **per-edge** and **per-object penalization**, if $t_2 \geq t_3$

$$\mathrm{Mod}_{p,1}(\Gamma) = \frac{1}{\varphi(t_1)^p}, \quad \text{for } 1 \leq p < \infty,$$

$$\mathrm{Mod}_{\infty,1}(\Gamma) = \frac{1}{\varphi(t_1)}.$$

If $t_2 < t_3$ then for **per-edge penalization** case, we have

$$\mathrm{Mod}_{p,1}(\Gamma) = \frac{1}{\varphi(t_1)^p} + (n-2)\left(\varphi(t_2)^q + \varphi(t_3)^q\right)^{-\frac{p}{q}}, \quad \text{for } 1 < p < \infty,$$

$$\lim_{p\to 1} \mathrm{Mod}_{p,1}(\Gamma) = \frac{1}{\varphi(t_1)} + \frac{1}{\max\{\ \varphi(t_2) + \varphi(t_3)\}}.$$

$$\mathrm{Mod}_{\infty,1}(\Gamma) = \max\left\{\frac{1}{\varphi(t_1)}, \frac{1}{\varphi(t_2) + \varphi(t_3)}\right\}.$$

And with **per-object penalization**, we have

$$\mathrm{Mod}_{p,1}(\Gamma) = \frac{1}{\varphi(t_1)^p} + \frac{2(n-2)}{2^p\varphi(t_3)^p}, \quad \text{for } 1 \leq p < \infty,$$

$$\mathrm{Mod}_{\infty,1}(\Gamma) = \max\left\{\frac{1}{\varphi(t_1)}, \frac{1}{2\varphi(t_3)}\right\}.$$

Now let's compute the same multiplicative temporal p-modulus with per-object penal-

ization through the duel problem. Choose $\mu = \begin{bmatrix} \Theta \\ 1-\Theta \end{bmatrix}$: With probability Θ go straight from s to t. With probability $1-\Theta$ choose uniformly among all 2-hop paths form s to t.

By Theorem 2,

$$\mathrm{Mod}_{p,\sigma}(\Gamma)^{-\frac{q}{p}} = \min_{\mu\in\mathcal{P}(\Gamma)} \sum_{e\in E} \hat{\sigma}(e)\eta(e)^q,$$

where $\eta = (\mathcal{N})^T\mu$.

Dual densities:

$$\eta_1 = \Theta\varphi(t_1) \quad \text{and} \quad \eta_2 = \eta_3 = \frac{1-\Theta}{n-2}\varphi(t_3). \tag{2.7}$$

The dual energy is

$$\mathcal{E}_{p,\sigma}(\mu) = (\Theta\varphi(t_1))^p + 2(n-2)\left(\frac{1-\Theta}{n-2}\varphi(t_2)\right)^p.$$

Let's consider the ratio:

$$\frac{\eta_1}{\eta_2} = \left(\frac{\rho_1}{\rho_2}\right)^{\frac{q}{p}} = \left(\frac{2\varphi(t_3)}{\varphi(t_1)}\right)^{\frac{q}{p}}$$

Also,

$$\frac{\eta_1}{\eta_2} = \frac{\Theta(n-2)}{1-\Theta}\frac{\varphi(t_1)}{\varphi(t_3)}.$$

Combining last two equations, we have

$$(n-2)\frac{\eta_1}{\eta_2}\Theta = (1-\Theta)\left(\frac{2\varphi(t_3)}{\varphi(t_1)}\right)^{\frac{q}{p}}$$

$$\Theta = \frac{\left(\frac{2\varphi(t_3)}{\varphi(t_1)}\right)^{\frac{q}{p}}}{(n-2)\frac{\varphi(t_1)}{\varphi(t_3)} + \left(\frac{2\varphi(t_3)}{\varphi(t_1)}\right)^{\frac{q}{p}}}.$$

Plugging in this optimal Θ in (2.7) we obtain optimal dual densities and they, being used in the formula of dual energy, give optimal dual energy.

For simplicity of formulas, we show this process for $p = 2$:

$$\mathcal{E}(\eta) = \frac{4\varphi(t_1)^2\varphi(t_3)^4}{[(n-2)\varphi(t_1)^2 + 2\varphi(t_3)^2]^2} + \frac{2(n-2)\varphi(t_1)^4\varphi(t_3)^2}{[(n-2)\varphi(t_1)^2 + 2\varphi(t_3)^2]^2}$$

$$= \frac{2\varphi(t_1)^2\varphi(t_3)^2}{(n-2)\varphi(t_1)^2 + 2\varphi(t_3)^2} = \varphi(t_1)^2\Theta.$$

Therefore,

$$\mathrm{Mod}_{2,1}(\Gamma) \geq \frac{n-2}{2\varphi(\omega)^2} + \frac{1}{\varphi(t_1)^2} = \frac{1}{\varphi(t_1)^2\Theta}.$$

Now we consider the case $t_3 > t_2$.

For upper bound guess on edge with t_1: $\rho_1 = \Theta\varphi(t_1)\dfrac{1}{\Theta\varphi(t_1)^2} = \dfrac{1}{\varphi(t_1)}$. And on the edges with t_2, t_3: $\rho_1 = \rho_2 = \frac{1-\Theta}{n-2}\varphi(t_3)\frac{1}{\Theta\varphi(t_1)^2}$.

Note that it is admissible because the direct path $s \to t$ has length $\frac{\varphi(t_1)}{\varphi(t_1)} = 1$ and all other paths with two hops have length $\varphi(t_3)\left[\frac{1}{2\varphi(t_3)} + \frac{1}{2\varphi(t_3)}\right] = 1$.

Thus,

$$\mathrm{Mod}_{2,1}(\Gamma) = \frac{1}{\varphi(t_3)^2} + 2(n-2)\left(\frac{1}{\varphi(t_3)^2}\right) = \frac{1}{\varphi(t_1)^2} + \frac{n-2}{2\varphi(t_3)^2}.$$

In the case when $t_3 \leq t_2$, the optimal densities are $\rho_1 = \frac{1}{\varphi(t_1)}$ and all other are zero. Hence

$$\mathrm{Mod}_{2,1}(\Gamma) = \frac{1}{\varphi(t_1)^2}.$$

So, we have obtained the same Modulus.

2.3 Conclusion of the chapter

We have defined temporal p-modulus in the form, where the time penalty function can be conveniently included and reflects temporal property of a particular application. We have showed that the Dual Theorem stays for temporal p-modulus. We have demonstrated some properties of temporal modulus on simple graph examples.

Chapter 3

Temporal p-Modulus and time-respecting paths

In Chapter 2 we introduced Modulus on general temporal families of objects on graphs. The analytical computations of Modulus we showed in examples become quite problematic on bigger graphs. For the latter ones we need numerical methods. For very big graphs [13], for which the size cannot be even computed precisely, only estimated, we might need a special approach. In this work, we talk only about "reasonably big" graphs. To compute p-Modulus on static graphs there exists an efficient numerical algorithm, presented in [8]. This chapter is about adjusting this algorithm to time-respecting paths on temporal graphs.

First, we reintroduce some notions from Chapter 2 in a more formal way.

3.1 Background. Time-respecting paths.

Definition 1 (**Temporal Graph**). Let *temporal graph* be a quadruple $G = (V, E, T, \sigma)$, where V is the set of vertices (nodes), E is the set of directed edges with assigned weights, i.e. there exists $\sigma : E \to \mathbb{R}_{>0}$, and $T : E \to 2^{\mathbb{R}}$ such that $T(e) = (t_1^e, t_2^e, ...)$ is a finite increasing sequence of *time moments* (instant *time windows*), when edge $e \in E$ is available.

In this dissertation, we always assume that for each considered graph there is some

enumeration of the edges. So, a directed edge connecting two vertices $a, b \in V$ is denoted by ab or just by a natural number, corresponding to the number of that edge.

In some literature [16], [11] one can meet temporal graph under the names *dynamic graph, evolving graph* etc.

Definition 2 (**Temporal Object).** Let $G = (V, E, T, \sigma)$ be a temporal graph. A *time-respecting object* or *temporal object* $\widehat{\gamma}$ is a sequence of pairs $((\gamma_1, \tau_1), ..., (\gamma_{|E|}, \tau_{|E|}))$, where γ_i is a non-negative real number representing some information about i-th element of E and τ_i is one of the time moments of i-th edge in E chosen in some way from $T(e_i)$.

Only finite graphs and finite objects will be considered in this dissertation.

Next definitions use the notion of *density* $\rho \in \mathbb{R}_{\geq 0}^{|E|}$ of the edges in a graph. One way to think of it is that *density* $\rho \in \mathbb{R}_{\geq 0}^{|E|}$ is the usage ("importance") the edges by (for) all given objects on a graph together .

Definition 3 (p**-Energy).** We call p*-energy* on a temporal graph $G = (V, E, T, \sigma)$ for density $\rho \in \mathbb{R}_{\geq 0}^{|E|}$ the function

$$\mathcal{E}_{p,\sigma}(\rho) = \begin{cases} \sum\limits_{e \in E} \sigma(e)\rho_e^p, & \text{if } 1 \leq p < \infty \\ \sup\limits_{e \in E} \sigma(e)\rho_e, & \text{if } p = \infty \end{cases}.$$

Definition 4 (**Undirected edge).** *Undirected edge* between a and b is a pair of directed edges $\{ab, ba\}$ with $\sigma(ab) = \sigma(ba)$ and $T(ab) = T(ba)$ if there exists only one element in the vector ρ corresponding to $\{ab, ba\}$.

For linguistic convenience, we will omit the words "directed" and "undirected" when it is clear from the context.

Notice that if σ is assigned to a graph once and does not change ever, ρ is a variable on a graph under our control.

Definition 5 (**Temporal p-Modulus).** Let Γ^{tmp} be a set of temporal objects on a temporal graph $G = (V, E, T, \sigma)$, *max-function* be a map $S : \Gamma^{\text{tmp}} \to \mathbb{R}$ such that $S(\widehat{\gamma}) = \max_{(\gamma_i, \tau_i) \in \widehat{\gamma}}(\tau_i)$ for any $\widehat{\gamma} = ((\gamma_1, \tau_1), ..., (\gamma_n, \tau_n)) \in \Gamma^{\text{tmp}}$, $n = |E|$. Let *wrap function* f be

a non-negative function of two arguments with the properties: non-decreasing by both arguments, continuous from the right on first argument and for any $\widehat{\gamma} \in \Gamma^{\text{tmp}}$ there exists $\rho \geq 0$ such that $f(l_\rho(\widehat{\gamma}), S(\widehat{\gamma})) \geq 1$, where $l_\rho(\widehat{\gamma}) := \sum_{i=1}^{n} \gamma_i \rho_i$ is called *static total usage cost*. *Temporal p-Modulus*, for $1 \leq p \leq \infty$ on a set of objects Γ^{tmp} of a temporal graph G is

$$\text{Mod}_{p,\sigma}^{tmp}(\Gamma^{\text{tmp}}) = \min_{\rho \geq 0,\ l_\rho^{\text{tmp}}(\widehat{\gamma}) \geq 1\ \forall \widehat{\gamma} \in \Gamma^{\text{tmp}}} \mathcal{E}_{p,\sigma}(\rho),$$

where $l_\rho^{\text{tmp}}(\widehat{\gamma}) := f[l_\rho(\widehat{\gamma}), S(\widehat{\gamma})]$ is called *total usage cost*. The vector minimizer ρ^* for the problem above is called *optimal density*.

□

Definition 6 (Projection and Usage Matrix). Let *projection* of temporal object to a static object be a map $\pi : \Gamma^{\text{tmp}} \to \Gamma$ such that $\pi(\widehat{\gamma}) = (\gamma_1, ...\gamma_{|E|})$ for all $\widehat{\gamma} \in \Gamma^{\text{tmp}}$.

Then $\mathcal{N}(\Gamma) = \mathcal{N}(\pi(\Gamma^{\text{tmp}})) = \begin{bmatrix} \pi(\widehat{\gamma}^1) \\ \dots \\ \pi(\widehat{\gamma}^{|\Gamma^{\text{tmp}}|}) \end{bmatrix}$, where $\widehat{\gamma}^i \in \Gamma^{\text{tmp}}$. We denote a column of $\mathcal{N}$ corresponding to an edge $e \in E$ as $\mathcal{N}_e$ and an element of $\mathcal{N}$ in i-th row and j-th column as $\mathcal{N}(i, j)$ and so we will do for temporal usage matrix.

Let function $\alpha : \Gamma^{\text{tmp}} \to \mathbb{R}$ be define by

$$\alpha(\widehat{\gamma}) := \inf_{\rho} \{x : f[l_\rho(\widehat{\gamma}), S(\widehat{\gamma})] \geq 1 \quad \wedge \quad x = l_\rho(\widehat{\gamma})\} \quad \forall\ \widehat{\gamma} \in \Gamma^{\text{tmp}}. \tag{3.1}$$

The *temporal usage matrix* of Γ^{tmp} is obtained from $\mathcal{N}$ by

$$\mathcal{N}^{\text{tmp}}(\widehat{\gamma}, e) = \frac{\mathcal{N}(\widehat{\gamma}, e)}{\alpha(\widehat{\gamma})}\ \forall\ e \in E,\ \forall\ \widehat{\gamma} \in \Gamma^{\text{tmp}}. \tag{3.2}$$

The rows of $\mathcal{N}^{\text{tmp}}$ are called *reduced temporal objects*.

Theorem 4. *Let $\mathcal{N}^{\text{tmp}}$ be a temporal usage matrix of Γ^{tmp}.*

Then temporal p-Modulus in Definition 5 for $1 \le p \le \infty$ can be computed as

$$\mathrm{Mod}^{\mathrm{tmp}}_{p,\sigma}(\Gamma^{\mathrm{tmp}}) = \min_{\rho\in\mathbb{R}^{|E|}_{\ge 0} \ \wedge \ \mathcal{N}^{\mathrm{tmp}}\rho\ge 1} \mathcal{E}_{p,\sigma}(\rho).$$

Proof. Observe that if $f(l_\rho(\gamma), S(\widehat{\gamma})) \ge 1$ then $\alpha(\widehat{\gamma}) \le l_\rho(\gamma)$.

On the other hand, $\alpha(\widehat{\gamma}) > 0$ because $l_\rho(\gamma) > 0$ for all ρ. Therefore

$$f(l_\rho(\gamma), S(\widehat{\gamma})) \ge 1 \implies l_\rho(\gamma)/\alpha(\widehat{\gamma}) \ge 1.$$

According to Definition 5 and to (3.1) value $\alpha(\widehat{\gamma})$ exists for all $\widehat{\gamma} \in \Gamma^{\mathrm{tmp}}$. So, if $f(l_\rho(\gamma), S(\gamma)) < 1$ then due to non-decreasing of f inequality $\alpha(\widehat{\gamma}) > l_\rho(\gamma)$ is valid. That yields

$$f(l_\rho(\gamma), S(\widehat{\gamma})) < 1 \implies l_\rho(\gamma)/\alpha(\widehat{\gamma}) < 1.$$

And all together

$$f(l_\rho(\gamma), S(\widehat{\gamma})) \stackrel{?}{\ge} 1 \iff l_\rho(\gamma)/\alpha(\widehat{\gamma}) \ge 1.$$

Remembering that $l^{tmp}_\rho(\widehat{\gamma}) = f(l_\rho(\gamma), S(\gamma))$ and $l_\rho(\gamma) = \sum_{e\in E} \mathcal{N}(\gamma, e)\rho_e$ we have

$$l^{tmp}_\rho(\widehat{\gamma}) \ge 1 \iff \sum_{e\in E} \frac{\mathcal{N}(\gamma, e)}{\alpha(\widehat{\gamma})}\rho_e \ge 1 \stackrel{(3.2)}{\iff} \sum_{e\in E} \mathcal{N}^{\mathrm{tmp}}(\gamma, e)\rho_e \ge 1.$$

Or

$$\big((\forall\ \widehat{\gamma} \in \Gamma^{\mathrm{tmp}})\, l^{tmp}_\rho(\widehat{\gamma}) \ge 1\big) \iff \mathcal{N}^{\mathrm{tmp}}\rho \ge 1. \tag{3.3}$$

□

Remark 1. Theorem 4 says that, for computation of Modulus, it's enough to have only information contained in $\mathcal{N}^{\mathrm{tmp}}$ instead of knowledge of function f and numbers $\{S(\hat{\gamma})\}_{\hat{\gamma}\in\Gamma^{\mathrm{tmp}}}$. From now, unless it causes confusion, we will call reduced temporal objects simply by *objects*.

Definition 5 gives us big variety of wrap function f. So far we consider only two versions of f. Let's present them.

Definition 7 (**Time-Penalty and Additive Temporal p-Modulus**). Let *time-penalty function* be a non-decreasing function $\varphi : \mathbb{R} \to \mathbb{R}_{\geq 0}$. Let wrap function in Definition 5 be $f(x, y) = x + \varphi(y)$. Then the temporal p-Modulus with such f for $1 \leq p \leq \infty$ on Γ^{tmp} we call *Additive Temporal p-Modulus*.

Lemma 1. Let additive temporal p-modulus be defined on a family of temporal objects Γ^{tmp} and time-penalty function φ satisfy $\varphi(x) < 1$ for all x. Then temporal usage matrix is provided by

$$\mathcal{N}^{\text{tmp}}(\widehat{\gamma}, e) = \frac{\mathcal{N}(\widehat{\gamma}, e)}{1 - \varphi(S(\widehat{\gamma}))}, \quad e \in E, \ \widehat{\gamma} \in \Gamma^{\text{tmp}}.$$

The notations are taken from Definition 5 and Theorem 4.

Proof. Since

$$l_{\rho}^{\text{tmp}}(\widehat{\gamma}) = l_{\rho}(\pi(\widehat{\gamma})) + \varphi(S(\widehat{\gamma})) \geq 1 \tag{3.4}$$

condition $\varphi(S(\widehat{\gamma})) < 1$ provides existence of such density ρ that the non-strict inequality (3.4) becomes equality and we have $\alpha(\widehat{\gamma}) = 1 - \varphi(S(\widehat{\gamma}))$. Apply Theorem 4. □

It appears that in applications, the intuitive choice of time-penalty function φ is easier when wrap function is a product of its arguments.

Definition 8 (**Multiplicative Temporal p-Modulus**). Let φ be time-penalty function and wrap function in Definition 5 be $f(x, y) = x\varphi(y)$. Then the temporal p-Modulus with such f for $1 \leq p \leq \infty$ on Γ^{tmp} we call *Multiplicative Temporal p-Modulus*.

Lemma 2. Let multiplicative temporal p-modulus be defined on a family of temporal objects Γ^{tmp} and time-penalty function φ satisfy $\varphi(x) > 0$ for all x. Then temporal usage matrix is provided by

$$\mathcal{N}^{\text{tmp}}(\widehat{\gamma}, e) = \mathcal{N}(\widehat{\gamma}, e)\varphi(S(\widehat{\gamma})), \quad e \in E, \ \widehat{\gamma} \in \Gamma^{\text{tmp}}.$$

The notations are taken from Definition 5 and Theorem 4.

Proof. Since

$$l_\rho^{\mathrm{tmp}}(\widehat{\gamma}) = l_\rho(\pi(\widehat{\gamma}))\varphi(S(\widehat{\gamma})) \geq 1 \tag{3.5}$$

condition $\varphi(S(\widehat{\gamma})) > 0$ provides existence of such density ρ that the non-strict inequality (3.5) becomes equality and we have $\alpha(\widehat{\gamma}) = 1/\varphi(S(\widehat{\gamma}))$. Apply Theorem 4. □

3.2 Algorithm for Modulus.

Let's define a function *shortest* as $\gamma^* = \mathrm{shortest}(\rho) \implies \forall \gamma \in \Gamma : l_\rho(\gamma^*) \leq l_\rho(\gamma)$. Algorithmically, the function **s**hortest can be implemented by means of Dijkstras algorithm [1, Theorem 7.15]. Let $A(\Gamma')$ denote the admissible set of density ρ on the family Γ'. To approximate $\mathrm{Mod}(\Gamma)$ with an error tolerance of $0 < \epsilon_{tol} < 1$ in [8] there is the following

Algorithm of Modulus approximation.

0. $\rho \longleftarrow 0$
1. $\Gamma' \longleftarrow \emptyset$
2. **loop**
3. $\quad \gamma \longleftarrow \mathrm{shortest}(\rho)$
4. $\quad$ **if** $l_\rho(\gamma)^p \geq 1 - \epsilon_{tol}$ then
5. $\quad\quad$ stop
6. $\quad$ **end if**
7. $\quad \Gamma' \longleftarrow \Gamma' \cup \gamma$
8. $\quad \rho \longleftarrow \mathrm{argmin}\{\mathcal{E}_p(\rho) : \rho \in A(\Gamma')\}$
10. **end loop**

It is easy to see that to adjust the algorithm above to temporal modulus, all we need is just to change l_ρ to the temporal total usage cost function ℓ_ρ^{tmp} and provide **s**hortest for temporal or *time-respecting paths*.

Figure 3.1: *Example of a simple temporal graph.*

3.3 Temporal modulus on time-respecting paths

Let's introduce one important special case of temporal object.

Definition 9 (**Time-respecting Path).** Let $G = (V, E, T, \sigma)$ be a temporal graph with some enumeration of edges. A *time-respecting path* (TRP) of length $m \in \mathbb{N} \cup \infty$ on a G is a sequence of pairs $((e_{k_1}, t_{k_1}), (e_{k_2}, t_{k_2}), ..., (e_{k_m}, t_{k_m}))$, such that e_{k_1},..., e_{k_m} are sequentially connected edges of G, i.e. $e_{k_i} = ab$ and $e_{k_{i+1}} = cd$ implies $b = c$ for $i = \overline{1, m-1}$. The time moments t_{k_j} satisfy $t_{k_1} < t_{k_2} < ... < t_{k_m}$, where $t_{k_j} \in T(e_{k_j})$ for all j. If TRP has no loops, then we call it *simple* TRP.

TRP is also known in some literature [16] as *time-dependent path*.

Problem 1.

Compute temporal p-Modulus on $\Gamma_{a \rightsquigarrow b}$, a set of TRPs from a node a (*source*) to the node b (*sink*).

To solve that problem we need to form a temporal usage matrix for $\Gamma_{a \rightsquigarrow b}$. First, let's observe that all TRP with loops can be disregarded since their total usage cost is always greater than the same TRP with the loop cut off. For the TRP with no loops, we can easily build a usage matrix. For example, on Figure 3.1 there are two significant (meaningful as constraints) TRPs: $((ac, 1), (cb, 3))$ and $((ac, 2), (cb, 3))$. A temporal usage matrix would be

$$\mathcal{N}_{a \rightsquigarrow b} = \begin{bmatrix} \varphi(3) & \varphi(3) & 0 \\ \varphi(4) & \varphi(4) & 0 \end{bmatrix}$$

The existing algorithm for computing static p-Modulus on a set Γ of paths uses a shortest path algorithm as an inside tool. Thus, to adjust Modulus for temporal graphs we have to

define some equivalent to shortest paths for time-respecting paths and invent an algorithm to find it.

To find shortest path on a static graph, there is well-known Dijkstra algorithm with Min Heap [9] with complexity $O(|E|\log|V|)$ and with Fibonacci Heap [10] with complexity $O(|E|+|V|\log|V|)$.

3.3.1 Pareto optimal set of paths on a temporal graph

On temporal graphs the concept of best path becomes more complicated than on static ones. Now we have two criteria of quality: time of arrival to the destination point and cost to cross all the edges of the path. Before using any idea to combine these two into one, it makes sense to find the Pareto set of TRP in this two-criteria space.

Definition 10. Let Ξ, Λ be ordered sets and $\xi_1, \xi_2 \in \Xi$ and $\lambda_1, \lambda_2 \in \Lambda$. We say that a pair (ξ_1, λ_1) *dominates* (ξ_2, λ_2) and write $(\xi_1, \lambda_1) \succ (\xi_2, \lambda_2)$ if either $\xi_1 > \xi_2$ and $\lambda_1 \geq \lambda_2$ or $\xi_1 \geq \xi_2$ and $\lambda_1 > \lambda_2$.

Definition 11. Let binary relation "$\succ$" be defined on $\Xi \times \Lambda$ as in definition above. A set of pairs $P \subseteq D \subseteq \Xi \times \Lambda$ is called *Pareto set* of D and denote $P(D)$ if

$$P = \{y' \in D : \{y'' \in D : y' \succ y''\} = \emptyset\}. \tag{3.6}$$

For a node $u \in V$, we call a pair of a node and time (v, t) a *virtual node* of u if $uv \in E$ and $t \in T(uv)$. Let a be the source node. We'll say a virtual node (v, t) is *reachable* if there exists TRP $\widehat{\gamma}$ from a to v (denote $\gamma : a \rightsquigarrow v$) such that $S(\widehat{\gamma}) = t$.

Given ρ, φ, S as in Definition 5 we have got $l_\rho(\widehat{\gamma})$ and $S(\widehat{\gamma})$ as two criteria of optimality of TRP. We'll call them in this subsection *length* and *arrival time*, respectively. The pair $(l_\rho(\pi(\widehat{\gamma})), S(\widehat{\gamma}))$ is called *value-pair* of TRP $\widehat{\gamma}$.

In fact, the algorithms we present in this subsection work if the length $l : \pi(\Gamma^{\text{tmp}}) \to \mathbb{R}_{\geq 0}$

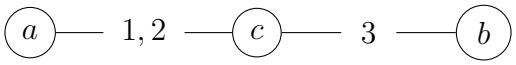

Figure 3.2: *Example of multiplicity of TRP $a \rightsquigarrow b$.*

is additive, i.e. for any TRP $u \rightsquigarrow v \in \Gamma^{\text{tmp}}$ and for any node c on $u \rightsquigarrow v$

$$l(\pi(u \rightsquigarrow v)) = l(\pi(u \rightsquigarrow c)) + l(\pi(c \rightsquigarrow v)).$$

For convenience of the narration we introduce the following

Definition 12. Let $\Gamma_{a \rightsquigarrow b}$ be a set of all TRP from a to b. The function $F : \Gamma_{a \rightsquigarrow b} \to \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ such that $F(\widehat{\gamma}) = (l_\rho(\widehat{\gamma}), S(\widehat{\gamma}))$ for any $\widehat{\gamma} \in \Gamma_{a \rightsquigarrow b}$ is called *TRP-meter*.

Definition 13. A TRP $\widehat{\gamma} : a \rightsquigarrow b$ is *Pareto optimal* if, for any other TRP $\widehat{\gamma}' : a \rightsquigarrow b$, the value of TRP-meter $F(\widehat{\gamma}') = (l(\widehat{\gamma}'), S(\widehat{\gamma}'))$ is not dominated by $F(\widehat{\gamma}) = (l(\widehat{\gamma}), S(\widehat{\gamma}))$, i.e. $F(\widehat{\gamma}) \not\succ F(\widehat{\gamma}')$.

It is not hard to see that the set value-pairs of all Pareto optimal TRP $a \rightsquigarrow b$ is a Pareto set of value-pairs of all TRP from $a \rightsquigarrow b$.

In the view of future computations of Modulus on a set of TRP $a \rightsquigarrow b$ (where we just have to satisfy the constrains on all TRP), we don't really need to find all optimal TRP but we want the set of TRP forming the Pareto set of value-pairs of all TRP from $a \rightsquigarrow b$. What is the difference? Two different TRP may have the same length and arrival time. First, they may go through different nodes. Second, there might be a multiplicity if they go through the same nodes but in different time moments. On the Figure 3.2), there are two TRP $a \rightsquigarrow b$. One goes from a to c either at the time moment 1 or 2, the value-pair will be the same because the length is the same since they go through the same edges and the arrival time is 3.

The following is a very inefficient but algorithmically simple way of finding Pareto set $P(F(\Gamma_{a \rightsquigarrow b}))$ and the set of all Pareto optimal TRP.

Algorithm 0.

Let $\Gamma_{a\rightsquigarrow b}$ be the set of all TRP from node a to node b on a temporal graph G. Let $D = F(\Gamma_{a\rightsquigarrow b})$. Now we form a Pareto set $P(D)$ by removing from D consequently in arbitrary order all dominating elements. So, as a result we obtain (3.6). It is easy to see that $F^{-1}(P(D))$ is the set of all Pareto optimal TRP.

In practice, for graphs with just n nodes, $|\Gamma_{a\rightsquigarrow b}|$ may be greater than $(n-2)!$. Thus, Algorithm 0 is unrealistic.

Problem 1.1. For nodes $a, b \in V$, find $P(F(\Gamma_{a\rightsquigarrow b}))$ and some subset $R \subseteq F^{-1}(P(F(\Gamma_{a\rightsquigarrow b})))$ such that each value-pair from $P(F(\Gamma_{a\rightsquigarrow b}))$ has at least one TRP in R.

For the upcoming Algorithm 1 we will build TRP satisfying the following

Principle of Optimality in Forward Direction: Let $p_{a\rightsquigarrow b}$ be a Pareto optimal TRP from a to b. Then for each intermediate node c of $p_{a\rightsquigarrow b}$, the sub-path $p_{a\rightsquigarrow c}$ is also a Pareto optimal TRP a to c.

The idea of Two-dimensional Dijkstra Algorithm.

In order to understand how to search efficiently for Pareto optimal TRP, we imagine an associated static graph. In this graph, each node $u \in V$ is split into a set, $\cup_\tau(u, \tau)$, of virtual nodes indicating the node and arrival time. We will study the computational complexity of running Dijkstra on this graph as an upper bound for the Two-dimensional Dijkstra in the upcoming Section 3.3.3. This algorithm can be made more efficient by using the following observations.

Consider the step of visiting a given virtual node (u, τ). When exploring the neighbors, we do not need to consider all virtual neighbors, rather, for each neighbor v of u, we need only consider the virtual neighbor (v, q) such that

$$q = \arg\min\{r \in T(uv) : r > \tau\}, \tag{3.7}$$

as long as such a time exists. Moreover, once a virtual node (u, τ) has been visited, no virtual node of the form (u, q) with $q \geq \tau$ need to be explored. To see this, suppose that $\gamma \in \Gamma_{a\rightsquigarrow u}$ is the first TRP to reach node u (and does so at time τ). Due to the nature of

Dijkstra's algorithm,

$$\ell_\rho(\pi(\gamma)) = \min_{\gamma' : a \rightsquigarrow u} \ell_\rho(\pi(\gamma')).$$

These observations suggest the following modified Dijkstra algorithm.

Variables and Functions:

For the convenience of the code, nodes are enumerated as $\{1, 2, ..., |V|\}$ and called by these numbers. The set Q holds the quadruples (l, τ, u, p) describing the frontier of the explored part of the graph by paths going from a, where u is the node of the frontier, τ is the arrival time to u, l is the path length from a to u, and p the predecessor node. Q is implemented as a min-heap ordered first by l and then by τ. In fact, in our Python code, min-heap uses lexicographical order for all four entries. So, the ties by first two arguments will be resolved by the numerical names of nodes.

The function **heappop** returns four elements of the minimum quadruple from Q in sense of lexicographical order and simultaneously delete this quadruple from Q and resorting the rest of Q.

The function **heappush** adds to Q a new quadruple and then sorts Q.

The function **rho(u,v)** returns the density on the edge uv.

The list minTau contains the time moments we have arrived at each node or ∞ if the node not visited yet.

The function **TemporalNeighbors(u, t)** returns the earliest reachable virtual neighbors of (u, t), i.e. the set $\{(v, q_v)\}$, such that each time moment q_v is the minimum when the neighbor v is reachable after time moment t (see (3.7)).

Input: Temporal graph $G = (V, E, T)$ with assigned σ and a starting point $a \in V$ and a final point $b \in V$.

Output: The set Dist holds the tuples that have been computed in order which allows to build easily the whole set of Pareto optimal TRP from a to b.

In fact, the set Dist can be imaged as tree with the root at a. We can grow all the branches (i.e. optimal TRP) as following. Extract next quadruple from Q. Find in the ends of a growing branch a node (which is in 4-th position of the quadruple) and continue

the branch by 3-d position of the quadruple. If the search in the ends of the branches is unsuccessful, that means that we reached the start node a.

The final result after decoding Dist as above is the collection of singly connected lists, which are the set of TRP from a to all other nodes.

The Python pseudo-code of the ideas above:

Algorithm 1. (Two-dimensional Dijkstra)

```
0: Q = {(0, 0, a, None)}
1: Dist = {}
2: for u in V :
3:     minTau[u] = ∞
4: while Q :
5:     l, t, u, (p,τ) = heappop(Q)
6:     if  t < minTau[u] :
7:         Dist[u][t]= ( l, (p,τ))
8:         minTau[u] = t
9:         for  (v, q)  in  TemporalNeighbors (u, t):
10:            if minTau[b] > t:
11:              heappush(Q, (l + rho ( uv ), q , v , (u, t)))
```

To retrace a path from Dist for an optimal value pair (l^*, t^*), one can apply a simple

Algorithm 2.

```
0: cur = ( b, t*)
1: path = [ ]
2: while cur != None:
3:     path.append(cur[0])
4:     cur = Dist[cur[0]][cur[1]][1]
```

Theorem 5. *For $a, b \in V$ Algorithms 1,2 find $P(\Gamma_{a\rightsquigarrow b})$ and $R \subset F^{-1}(P(\Gamma_{a\rightsquigarrow b}))$ such that each value-pair from $P(F(\Gamma_{a\rightsquigarrow b}))$ has at least one TRP in R.*

Proof. Let's fix $b \in V$ and show that every TRP $a \rightsquigarrow b$ obtained by Algorithms 1, 2 is Pareto optimal.

If a quadruple (l, t, b, p) has bubbled to the top of the heap Q, the associated virtual node (b, t) is closer in sense of length to the source node than any other node in the heap, because heap sorts (by heappop and heappush) quadruples in lexicographic order. Now if t is smaller than minTau$[b]$, then the value-pair (l, t) should belong to Pareto set. since all other value-pairs with greater arrival time have already been discovered. On the other hand, if t is no smaller than minTau$[b]$, then a dominating pair has already been discovered and (l, t) is not in Pareto set.

Now we show that if there exists a Pareto Optimal TRP $\gamma : a \rightsquigarrow b$ for $b \in V$ then it will be found by Algorithms 1, 2. Assume that γ with the values (l, t) at b has not been found by Algorithms 1, 2. There are two options: either Algorithm 1 got to the predecessor $pred(b)$ of b or it stopped γ somewhere else before. The first is possible only if in the line 6 of Algorithm 1 the condition $t <$ minTau$[u]$ was not satisfied. That means that (l, t) dominates pairs obtained by some other paths, which reached b. Thus, γ is simply not a Pareto Optimal TRP $a \rightsquigarrow b$, which contradicts the assumptions. Hence, γ stopped at $pred(pred(b))$ or somewhere else before. Repeating previous steps recurrently we follow path γ backwards and come all the way to the node a. Here we have to conclude that γ never left the starting point a and hence doesn't exist. The source of contradiction is in the assumption. Therefore, every Pareto Optimal TRP has been found by Algorithms 1, 2. □

Proposition 1. Let U be the set virtual nodes passed by TRPs from $\Gamma_{a \rightsquigarrow b}$. Then a virtual node (v, t) is added to Q in Algorithm 1 if and only if (v, t) is a temporal neighbor of some point in U.

Proof. By Theorem 5, the virtual nodes, which temporal neighbors are getting added to Q, lie on Pareto paths to virtual nodes U. Thus, by Algorithm 1 (v, t) can be added to Q if and only if this (v, t) is a temporal neighbor of some virtual node from U. □

Theorem 6. *Let M be an upper bound on the number of all time moments of a temporal graph $G = (V, E, T)$. The complexity of Two-Dimensional Dijkstra on a temporal graph is*

$O(M|E|\log(M|V|))$.

Proof. Let's denote $R := \sum_{(v,t)\in P} \deg((v,t))$, where the degree deg of a virtual node we understand as the number of reachable nodes from v.

It is known that the most computationally expensive operation in Dijkstra algorithm with Fibonacci heap is sorting of Q, which happens in the loop twice: for deletions and additions. Same must happen in Algorithm 1. We run the loop for Q no more than R times. By Proposition 1, each time we either do nothing or remove from Q. That is done by Fibonacci heap and takes $O(\log R)$ operations. Also we add to Q. There is no need to count additions on each step since the final amount of additions (all together from the loop) must be equal to the amount of deletions.

Thus, the upper bound on computational complexity of Algorithm 1 is $O(R\log(R))$.

Now let's see that

$$O(R\log(R)) = O\left(\sum_{(v,t)\in P} \deg((v,t)) \log\left(\sum_{(v,t)\in P} \deg((v,t))\right)\right)$$

$$\leq O\left(\sum_{v\in V} M\deg(v)\right) \log\left(\sum_{v\in V} M\deg(v)\right)$$

$$= O\left(M2|E|\log\left(M2|E|\right)\right) \qquad \text{(By Handshaking Lemma)}$$

$$= O\left(M|E|\log\left(M|V|^2\right)\right) = O\left(M|E|\log\left(M|V|\right)\right).$$

□

3.3.2 Computing total usage cost on Pareto set of value-pairs

After we obtained Pareto set of value-pairs $P(F(\Gamma_{a\rightsquigarrow}b))$ by algorithm 1 we can easily compute the total usage cost $l(\widehat{\gamma})^{\text{tmp}} = l(\widehat{\gamma})\varphi(S(\widehat{\gamma}))$ and find the minimum one. This is a Python code for that:

Algorithm 3.

```
0:  options = [( dist[b][t][0], t) for t in dist[b]]
```

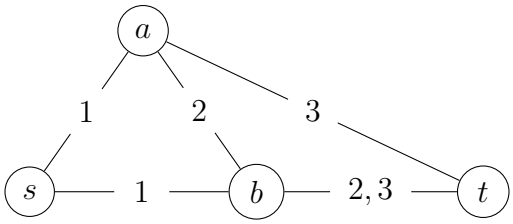

Figure 3.3: *A simple temporal graph.*

```
1: costs = [l * TimePenalty(t) for (l, t) in options]
2: ind = np.argmin(costs)
3: l, t = options[ind]
```

Thus, now Problem 1 can be solved by an algorithm in [2], where we satisfy iteratively the most violated constraint.

3.3.3 Converting temporal graph to static one

Let's think of another way to solve Problem 1. Instead of considering a family of temporal objects, we can transform the graph to a static one and translate the problem to well-known to us p-Modulus on a static graph.

A temporal graph can be converted into a static directed graph $G' = (V', E')$ by the following.

Procedure 1:

0) Create a set of pairs $\{(v, 0)\}$, where v is a node from V. That's gonna be layer $\#0$ G'.

Then repeat for all existing in G time moments $0 < n \leq max_{e \in E}(\tau(e))$ the following:

1) Create a layer $\#n$ of G' (set of pairs $\{(v, n)\}$) by copying all nodes V of G.

2) Connect all pairs of previous layer with corresponding pairs (v, n) by a directed edges with the weight 0 on it.

3) If a node y of G at time n had an edge with a weight w to it from some node x of G, connect the copy of y in layer $\#n$ with the copy of x in the previous layer by a directed edge $x- > y$ with the weight w.

Look at the example on Figures 3.4

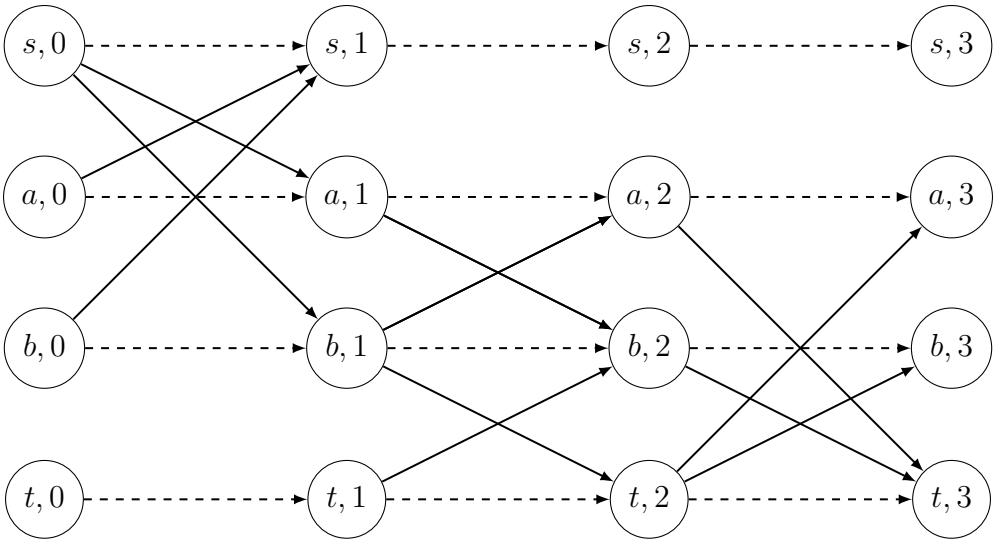

Figure 3.4: *The static representation of the temporal graph in Figure 3.3.*

Theorem 7. *The weighted temporal graph G with discrete time moments on edges is equivalent to the static graph G' (denote $G \sim G'$) in sense of minimal total usage cost path problem up to the "time" component of the weighted time-respecting distance $d_\rho(s,t)$:*

$$d_\rho(s,t) = \min_{(t,r)\in V'} f[d'_\rho((s,0),(t,r)),\varphi(r)],$$

where d'_ρ is the ρ-distance on G'. The minimum is taken only for existing paths.

Proof. Assume there exists a shortest path $a \to b = \{ae_1x_1e_2...e_nb\}$ with the time moments $\{\tau_1\tau_2...\tau_n\} \subset T$, where T is the set of all edge time moments of G . Let's show there exists a path $(a,0) \to (b,\tau_n)$ in G'. Since $e_1 \in a \to b$, according to (2) and (3) of Procedure 1, there must exist layers $\#0, \#t_1, \#t_2, ..., \#\tau_1$ of G', where $\tau_1 > t_i \in T$. Another words if there are edge time moments in graph G earlier than τ_1 then we understand this as "seating" at the node a in G before getting to x_1 at τ_1 or going through the chain of duplicates till the last of them is connected to (x_1,τ_1), i.e. $(a,0) \to (a,t_1) \to (a,t_2) \to ... \to (x_1,\tau_1)$ in G'. Then, similarly, we get to (x_2,τ_2) and so on. Thus, $(a,0)$ is path connected to (b,τ_n) in G'.

Consider now some directed shortest path $(a,\tau_1) \to (b,\tau_n)$ in G'. Any edge e on this path is either edge between duplicates, which corresponds to "seating" on the node, or $(x,\tau^*) \to (y,\tau^{**})$ for some $\tau^* < \tau^{**}$. According to Procedure 1, the last edge could appear

only if there exists an edge $x \to y$ at moment τ^{**} in G. So, all edges and nodes of a path in G' have corresponding temporal edges and nodes in G. Obviously, the partial order of layers in G' and partial order between time moments on edges of any path G are the same.

The distance $d_\rho(a \to b) = f[\sum_{i=1}^{n} w_i, \varphi(\tau_n)]$, where w_i is the weight on e_i. Meanwhile, the distance $d'_\rho((a,0) \to (b,\tau_n)) = \sum_{i=1}^{n} w_i$.

All is left to show that there is no paths $(a,0) \to (b,\tau_{n'})$ in G' such that $f[d'_\rho((a,0) \to (b,\tau_n)), \varphi(\tau_n)] > f[d'_\rho((a,0) \to (b,\tau_{n'})), \varphi(\tau_{n'})]$. Assume it is not true. Then as we have shown above for this $(a,0) \to (b,\tau_{n'})$ there must exist $a \to b$ with the finishing time moment $\tau_{n'} < \tau_n$, since f is non-decreasing by both arguments function and φ is non-decreasing as well . Hence, this last path had to be the shortest one between a and b in G. Contradiction.

□

Theorem 8. *Let $G = (V, E, T)$ be a temporal graph and M be an upper bound of the number of all time moment on the edges of G. Then the complexity of Dijkstra on a temporal graph converted to a static one is $O(M|E| \log(M|V|)$.*

Proof. Let $G' = (V', E')$ be a static graph obtained by Procedure 1 from a temporal graph $G = (c)$. Consider a problem of finding a Pareto set of TRP from node a to node b in G. In Theorem 7 we have showed that it is equivalent to family of shortest paths between nodes $(a,0)$ and $\{(b,\tilde{t})\}$, where $\tilde{t} > 0$, obtained by replication of the nodes of G for all time moments of graph existence.

As it's known the complexity of Dijkstra on a static graph is $O(|E'| + |V'| \log(V'))$. So, all we need is to estimate the number of virtual nodes $|V'|$ and virtual edges $|E'|$ in terms of $|V|, |E|$ and M.

It is easy to see that $|V'| = M|V|$. The number of edges of waiting on the nodes (meaning going from replicate to another) is $|E^{\text{wait}}| \leq |V|(M-1)$. Also we have edges between replicates of different nodes $|E^{\text{move}} \leq |E|M$. So, $|E'| = |E^{\text{wait}}| + |E^{\text{move}}|$. Thus, $|E'| \leq (|E| + |V|)M$. Then the complexity of shortest problem on G' is

$$O\left((|E| + |V|)M \log(M|V|)\right) = O(|E|M \log(M|V|)).$$

□

Thus, we proved that the computational complexity of two-dimensional (temporal) Dijkstra is the same that the computational complexity of one-dimensional (classic) Dijkstra on the converted to static graph. However, evidently, the first approach uses less memory.

3.3.4 Parallel rule of temporal p-modulus

Theorem 9 (Parallel Rule)**.** *If a temporal graph $G = (V, E, T)$ consists of two subgraphs $G = A \cup B$, wherer $A = (V_A, E_A, T_A)$ and $B = (V_B, E_B, T_B)$ such that they connected only in two nodes a and b $(A \cap B = \{a, b\})$, then temporal p-modulus of G on Γ, the set of time-respecting paths from a to b, equals to the sum of temporal p-moduli of A and B on the subsets Γ_A, Γ_B of objects Γ associated with A and B, respectively:*

$$\mathrm{Mod}_p(G, \Gamma) = \mathrm{Mod}_p(A, \Gamma_A) + \mathrm{Mod}_p(B, \Gamma_B).$$

Proof. Consider the temporal p-modulus of G on Γ:

$$\mathrm{Mod}_p(G, \Gamma) = \min_{\rho_e \in E} \left\{ \sum_{e \in E} \sigma(e) |\rho_e|^p \;\middle|\; \sum_{e \in E} \mathcal{N}_e^{\mathrm{tmp}} \rho_e \geq 1 \right\}.$$

Since Γ is the set of time-respecting paths going from a to b we can be split it into two subsets $\Gamma = \Gamma_A \cup \Gamma_B$, where Γ_A and Γ_B are the sets of time-respecting paths from a to b going through nodes of A and B, respectively. Moreover, any non-direct path from a to b of Γ_A cannot go through B and no paths of Γ_B can go through A except perhaps for the direct path (a, b) if exists, otherwise we would get self-intersections. Thus, the intersection of two subsets of objects is $\Gamma_A \cap \Gamma_B = (a, b)$. To simplify our notations, without loss of generality let's assume that if such a direct path from a to b exists it belongs only to the graph A but not B, so we can always say $\Gamma_A \cap \Gamma_B = \emptyset$.

Notice that for multiplicative modulus $\mathcal{N}^{\mathrm{tmp}}(\gamma, e) = \mathcal{N}(\gamma, e)\varphi(\gamma)$ and, for additive modulus, $\mathcal{N}^{\mathrm{tmp}}(\gamma, e) = \dfrac{\mathcal{N}(\gamma, e)}{1 - \varphi(\gamma)}$, where φ is the time-penalty function in the definition of G.

Then for both multiplicative and additive moduli if $\gamma \in A$ $\varphi(\gamma) = \varphi(\tau_A)$, where τ_A is some moment of time on E_A. Similarly, if $\gamma \in B$ $\varphi(\gamma) = \varphi(\tau_B)$, where τ_B is some moment of time on E_B. Since $E_A \cap E_B = \emptyset$ rows of the matrix $\mathcal{N}^{\text{tmp}}$ corresponding to E_A are completely independent of E_B and vice versa. Thus, we can rearrange this matrix as follows:

$$\mathcal{N}^{\text{tmp}} = \begin{bmatrix} \mathcal{N}_A^{\text{tmp}} \\ \mathcal{N}_B^{\text{tmp}} \end{bmatrix}.$$

Now observe that

$$\text{Mod}_p(G, \Gamma) = \min_{\rho_e \in (E_A \cup E_B)} \left\{ \sum_{\rho_e \in E_A} \sigma(e)|\rho_e|^p + \sum_{\rho_e \in E_B} \sigma(e)|\rho_e|^p \middle| \sum_{e \in E} \begin{bmatrix} \mathcal{N}_A^{\text{tmp}} \\ \mathcal{N}_B^{\text{tmp}} \end{bmatrix}_e \rho_e \geq 1 \right\} =$$

$$\min_{\rho_e \in E_A} \left\{ \sum_{\rho_e \in E_A} \sigma(e)|\rho_e|^p \middle| \sum_{e \in E} \mathcal{N}_A^{\text{tmp}} \geq 1 \right\} + \min_{\rho_e \in E_B} \left\{ \sum_{\rho_e \in E_B} \sigma(e)|\rho_e|^p \middle| \sum_{e \in E} \mathcal{N}_B^{\text{tmp}} \geq 1 \right\} =$$

$$\text{Mod}_p(A, \Gamma_A) + \text{Mod}_p(B, \Gamma_B)$$

□

3.4 Homogeneous modulus

It would be nice to have a property of time scaling for p-Modulus. Meaning:

$$\text{Mod}_p(\Gamma^\lambda) = \mathcal{A}(\lambda)\text{Mod}_p(\Gamma), \tag{3.8}$$

where Γ^λ is the set of time-dependent objects Γ on the same graph G but all the assigned time windows on edges are multiplied by λ. Using the Definition 5, (3.8) implies $(\rho_i^\lambda)^p = \mathcal{A}\rho_i^p$ for all i, where $\{\rho_i^\lambda\}$ is the optimal density of $\text{Mod}_p(\Gamma^\lambda)$.

Assume the the function f from Definition 5 has the following property

$$f(x, \lambda y) = g(\lambda) f(x, y), \tag{3.9}$$

where λ is the time scaling coefficient, g is some real-valued function. Then

$$l_\rho^{\text{tmp}}(\gamma) = f(l_\rho(\pi(\gamma)), S(\gamma)) = g(S(\gamma)) f(l_\rho(\gamma), 1) = g(S(\gamma)) f(l_\rho(\gamma)).$$

Assume f is continuous and strictly increasing, also $g(y) > 0$ if $y > 0$.

Let's rewrite the constraint according to these assumptions. The inequality $l_\rho^{\text{tmp}}(\gamma) \geq 1$ is equivalent to $f(l_\rho(\pi(\gamma))) \geq \dfrac{1}{g(S(\gamma))}$ or $l_\rho(\pi(\gamma)) \geq f^{-1}\left(\frac{1}{g(S(\gamma))}\right)$. Thus

$$\left[f^{-1}\left(\frac{1}{g(S(\gamma))}\right)\right]^{-1} l_\rho(\pi(\gamma)) \geq 1.$$

And so,

$$\mathcal{N}^{\text{tmp}}(\gamma, e) = \left[f^{-1}\left(\frac{1}{g(S(\gamma))}\right)\right]^{-1} \mathcal{N}(\pi(\gamma), e)$$

Observe that $\mathcal{A}(\lambda)^{\frac{1}{p}} = \left[f^{-1}\left(\frac{1}{g(\lambda)}\right)\right]^{-1}$. So, in order to have Homogeneous p-Modulus we it is sufficient to have the following property

$$\left[f^{-1}\left(\frac{1}{g(\lambda y)}\right)\right]^{-1} = \lambda^k \left[f^{-1}\left(\frac{1}{g(y)}\right)\right]^{-1}, \tag{3.10}$$

where k is some real number.

Remark 2. Notice that if both f^{-1} and g are homogeneous of degrees d_1 and d_2, respectively, such that $d_1 - d_2 = k$, then (3.10) is true. However, this is not a necessary condition. For example, two functions $y(x) = x + 5$ and $z(x) = x - 5$ are not homogeneous but $y(z(x)) = x$ is. That means that we are not limited to choose only homogeneous f^{-1} and g to get a Homogeneous Modulus.

Example of Homogeneous p-Modulus

Let $f(x) = x$, $g(y) = \varphi(y)$. Then we get $\mathcal{N}^{\text{tmp}}(\gamma, e) = \varphi(\tau(\gamma))\mathcal{N}(\pi(\gamma), e)$. If one choose $\varphi(y) = y^k$ we get Homogeneous p-Modulus.

Example of Non-Homogeneous p-Modulus

For some applications it can be reasonable to have $f(x) = x$ and $g(0) = 1$, because then

$$\lim_{\lambda \to 0+} \left[l_\rho^{\text{tmp}}(\gamma) = f(l_\rho(\pi(\gamma)), S(\gamma)) \right] = l_\rho(\pi(\gamma)) \tag{3.11}$$

and

$$\mathcal{N}^{\text{tmp}}(\gamma, e) = \varphi(\tau(\gamma))\mathcal{N}(\pi(\gamma), e),$$

where φ is continuous, increasing function function such that $\varphi(0) = 1$.

One can choose φ to be $\varphi(y) = e^{\alpha y}$ or $\varphi(y) = 1 + \alpha y^k$, where α is some time-scaling coefficient and k is some real number.

Alas, the homogeneous modulus and property (3.11) cannot go together.

3.5 Examples of numerical computation of temporal modulus

Example 6 (Complete Graphs). Consider the graph G in Example 5 of Chapter 2 with the number of nodes $n = 7$ and time windows on the edges: $t_1 = 0.8$, $t_2 = 2.775$, $t_3 = 3.8$ (see Figure 3.5). Let the per-object penalization be $\varphi(t) = e^t$ and $\sigma = 1$. Let's compute the 2-modulus of the family of time-respecting paths $s \rightsquigarrow t$ on G.

The densities are $\rho_1 = \frac{1}{\varphi(t_1)} = e^{-0.8} \approx 0.449328$ and $\rho_2 = \rho_3 = \frac{1}{2\varphi(t_3)} = 0.5e^{-3.8} \approx 0.011185$. Thus,

$$\text{Mod}_{2,1} = \frac{1}{\varphi(t_1)^2} + \frac{2(7-2)}{2^2\varphi(t_3)^2} \approx 0.202209.$$

The python code with the implemented Algorithm of Modulus approximation in Section 3.2 and Algorithm 1 in Section 3.3.1 for the problem above converges in 7 steps giving $\rho_1 = 0.449328$, $\rho_2 = \rho_3 = 0.011185$ and the value of modulus equals to 0.2031. Thus, error

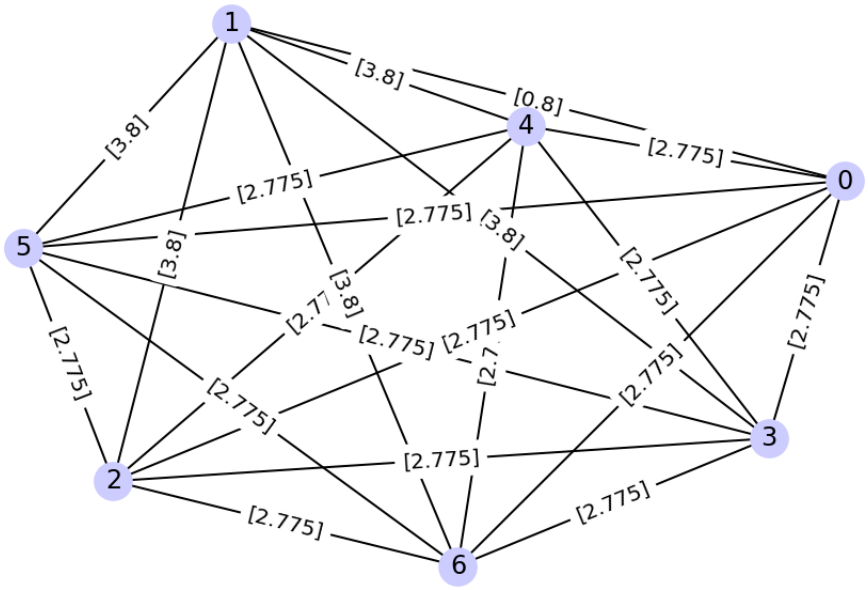

Figure 3.5: *Example 6. Complete 7-node temporal graph. Nodes "0" and "1" denote nodes s and t, respectively.*

is about 0.4% for the value of modulus and about the machine error for the densities.

Example 7 (**Trivial multi-graph).** Let's consider an example of a trivial multi-graph G, where nodes s and t are connected by two edges e_1, e_2 with single time windows 1.0 and 2.0, respectively, $\sigma = 1$ (see Figure 3.6). Running the code with the implemented Algorithm of Modulus approximation in Section 3.2 and Algorithm 1 in Section 3.3.1 on a set of time-respecting paths $s \rightsquigarrow t$ of G with the per-object penalization $\varphi(t) = e^t$. , we obtain the value of 2-modulus 0.1537 on the third iteration when the machine stopped with the error estimate equal to $1.806 \cdot 10^{-16}$. The optimal densities are $\rho_1 = 0.367879$ and $\rho_2 = 0.135335$.

Now let us compute 2-modulus on the simple graph version of G, meaning s and t are connected by a single edge e_1 with two time windows assigned $[1.0, 2.0]$, $\sigma = 1$ (see Figure 3.7). Running the same code for time-respecting paths $s \rightsquigarrow t$ of G, we receive that the value of 2-modulus equals to 0.1353 and optimal density is $\rho_1 = 0.367879$.

The fact that ρ_1 is the same in both versions of G above is expected.

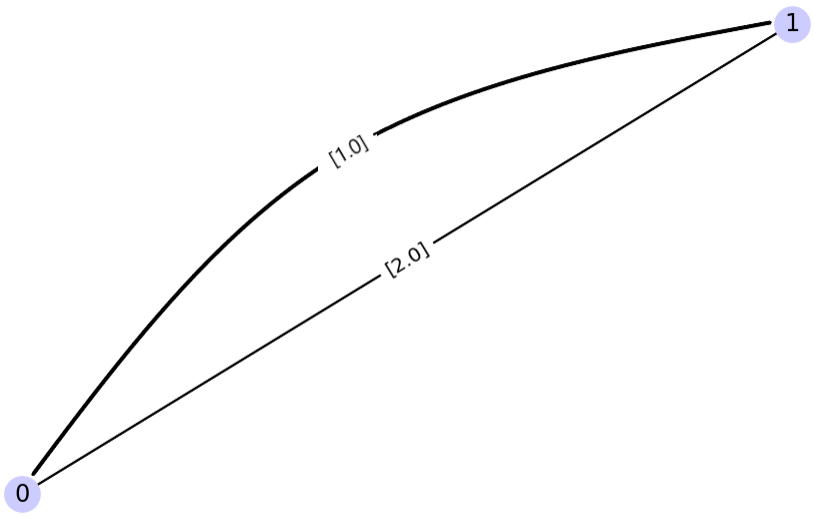

Figure 3.6: *Example 7. Trivial airplane connection graph. Nodes "0" and "1" denote nodes s and t, respectively.*

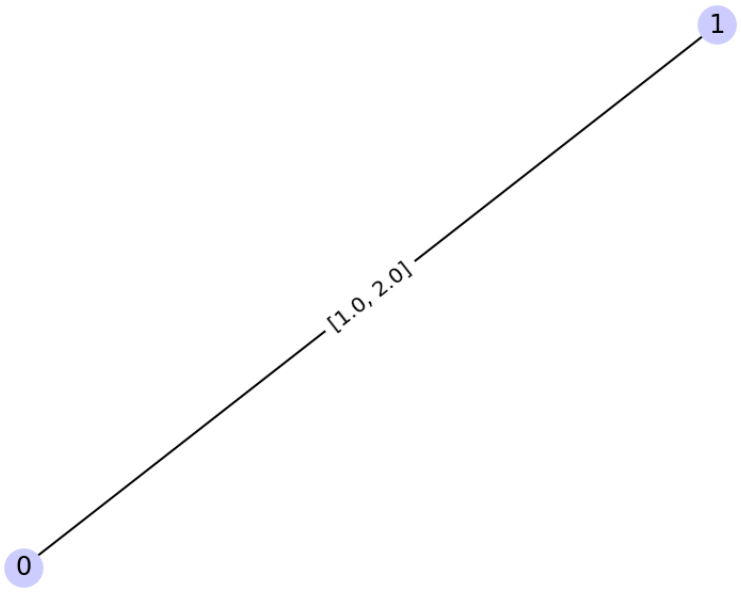

Figure 3.7: *Example 7. Trivial railroad connection. Nodes "0" and "1" denote nodes s and t, respectively.*

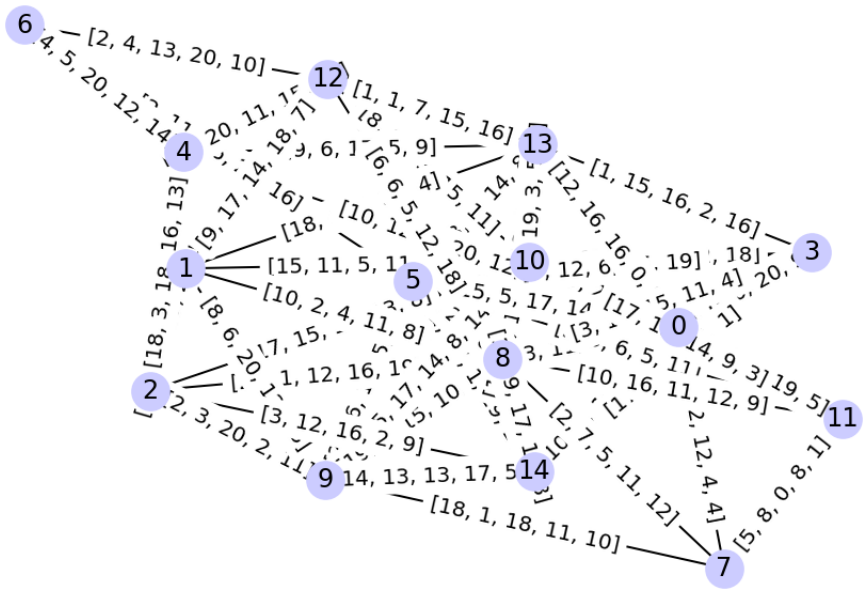

Figure 3.8: *Example 8. Random graph. Nodes "0" and "1" denote nodes s and t, respectively.*

Example 8 (Random graph). Here we build a random temporal graph. A 15-node graph G, $\sigma = 1$, with maximum 5 time moments with integer value from 0 to 20 per edge has been generated by the code in Appendix A (see Figure 3.8). Let the per-object penalization be $\varphi(t) = e^t$.

By the same code mentioned in previous examples, we compute the value of 2-modulus and optimal densities of the family of TRP $s \rightsquigarrow t$ (s and t are just two random nodes in G). Because of the random nature of G, the answer varies with every run. The histogram of these values for 100 runs is in Figure 3.9. In Figures 3.10 and 3.10, we show the optimal densities for two random seeds.

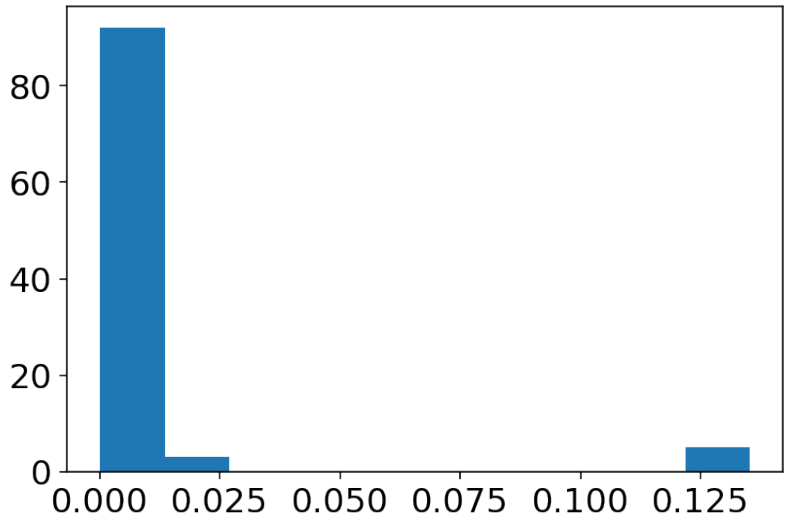

Figure 3.9: *Example 8. Histogram of values of modulus of TRP on random graph G.*

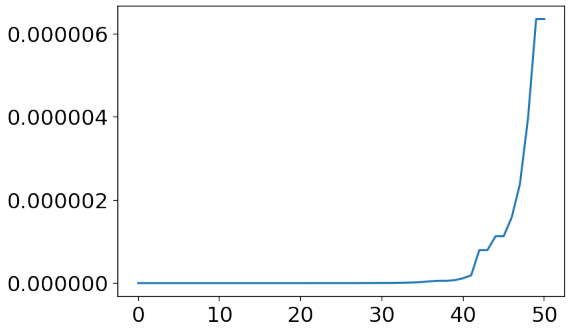

Figure 3.10: *Densities of 2-modulus in Example 8, first random seed.*

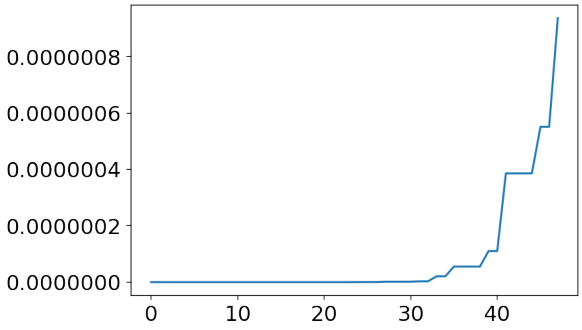

Figure 3.11: *Sorted densities of 2-modulus in Example 8, second random seed.*

Bibliography

[1] Aigner, M. *Discrete Mathematics*. American Mathematical Society, 2007.

[2] Albin, N., Brunner, M., Perez, R., Poggi-Corradini, P., and Wiens, N. Modulus on graphs as a generalization of standard graph theoretic quantities. *Conformal Geometry and Dynamics 19*, 13 (Dec. 2015), 298–317. arXiv:1504.02418.

[3] Albin, N., Clemens, J., Fernando, N., and Poggi-Corradini, P. Blocking duality for p-modulus on networks and applications. *Annali di Matematica Pura ed Applicata (1923 -) 198*, 3 (Nov. 2018), 973–999.

[4] Albin, N., Clemens, J., Hoare, D., Poggi-Corradini, P., Sit, B., and Tymochko, S. Fairest edge usage and minimum expected overlap for random spanning trees. *preprint* (2019).

[5] Albin, N., Fernando, N., and Poggi-Corradini, P. Modulus metrics on networks. *Discrete & Continuous Dynamical Systems - B 24*, 1 (2019), 1–17.

[6] Albin, N., Kottegoda, K., and Poggi-Corradini, P. Spanning tree modulus for secure broadcast games. *Networks* (Apr. 2020). arXiv:1904.03962.

[7] Albin, N., and Poggi-Corradini, P. Minimal subfamilies and the probabilistic interpretation for modulus on graphs. *The Journal of Analysis 24*, 2 (Aug. 2016), 183–208.

[8] Albin, N., Poggi-Corradini, P., Darabi Sahneh, F., and Goering, M. Modulus of families of walks on graphs. In *Proceedings of Complex Analysis and Dynamical Systems VII* (2017), M. L. Agranovsky, M. Ben-Artzi, C. Bénéteau, L. Karp, D. Khavinson, S. Reich, D. Shoikhet, G. Weinstein, and L. Zalcman, Eds., vol. 699 of *Contemporary Mathematics*, pp. 35–56. arXiv:1401.7640.

[9] Barbehenn, M. A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. *IEEE Transactions on Computers 47*, 2 (Feb. 1998), 263.

[10] Fredman, M. L., and Tarjan, R. E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM 34*, 3 (July 1987), 596–615.

[11] Grindrod, P., and Higham, D. J. Evolving graphs: dynamical models, inverse problems and propagation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 466*, 2115 (Nov. 2009), 753–770.

[12] Holme, P., and Saramäki, J. Temporal networks. *Physics Reports 519*, 3 (Oct. 2012), 97–125.

[13] Lovász, L. *Large networks and graph limits*, vol. 60. American Mathematical Society, 2012.

[14] Shakeri, H., Moradi-Jamei, B., Poggi-Corradini, P., Albin, N., and Scoglio, C. Generalization of effective conductance centrality for egonetworks. *Physica A: Statistical Mechanics and its Applications 511* (Dec. 2018), 127–138.

[15] Shakeri, H., Poggi-Corradini, P., Albin, N., and Scoglio, C. Network clustering and community detection using modulus of families of loops. *Physical Review E 95*, 1 (Jan. 2017), 012316.

[16] Zavlanos, M. M., and Pappas, G. J. Controlling connectivity of dynamic graphs. In *Proceedings of the 44th IEEE Conference on Decision and Control* (2005), IEEE, pp. 6388–6393.

Appendix A

Python code for generation of random temporal multi-graph

```
import networkx as nx
import random
from itertools import combinations
def RandTempGraphAir(n,Edmax,Tmax,nT=1):
    # n — number of nodes, Edmax — maximum number of edges between two nodes,
    # nT — number of time moments on an edge, Tmax — max value of the time moment
    G = nx.MultiGraph()
    G.add_nodes_from(range(0,n))
    Ed=[ ]
    for (u,v) in combinations( G.nodes(data=False), 2 ):
        nEd = random.randint(0, Edmax)   #uniform distribution
        for i in range(0,nEd):
            Ed.append((u,v,'sigma':1.0,'t':[random.randint(0, Tmax) for i in range(0,nT)]))
    G.add_edges_from(Ed)
    return G
```