Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Συνεκτικότητα σε Χρονικά Μεταβαλλόμενα Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΥΡΙΑΚΟΣ ΑΞΙΩΤΗΣ

Επιβλέπων : Δημήτρης Φωτάκης
Επ. Καθηγητής Ε.Μ.Π.

Αθήνα, Αύγουστος 2016

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Συνεκτικότητα σε Χρονικά Μεταβαλλόμενα Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΥΡΙΑΚΟΣ ΑΞΙΩΤΗΣ

Επιβλέπων : Δημήτρης Φωτάκης
Επ. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26η Αυγούστου 2016.

....	....	....
Δημήτρης Φωτάκης Επ. Καθηγητής Ε.Μ.Π.	Άρης Παγουρτζής Αν. Καθηγητής Ε.Μ.Π.	Νικόλαος Σ. Παπασπύρου Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Αύγουστος 2016

....

Κυριάκος Αξιώτης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στα παραδοσιακά γραφήματα, το ελάχιστο πιστοποιητικό συνεκτικότητας ονομάζεται συνδετικό δένδρο. Θεωρούμε μία γενίκευση της έννοιας της συνεκτικότητας για την πιο γενική κλάση των Χρονικά Μεταβαλλόμενων Γραφημάτων με διακριτές χρονικές ετικέτες, όπου το πιστοποιητικό συνεκτικότητας με ελάχιστο μέγεθος ονομάζεται Ελάχιστο Χρονικά Συνεκτικό Υπογράφημα ή Minimum Temporally Connected Subgraph (MTCS).

Αρχικά, παρουσιάζουμε διάφορες εφαρμογές οι οποίες μοντελοποιούνται εγγενώς σαν χρονικά μεταβαλλόμενα δίκτυα, καθώς και κάνουμε μια επισκόπηση αποτελεσμάτων στη βιβλιογραφία που σχετίζονται με χρονική συνεκτικότητα, όπως μία γενίκευση του θεωρήματος Menger για χρονικά μεταβαλλόμενα γραφήματα.

Στη συνέχεια, παρουσιάζουμε μία νέα κατασκευή, η οποία αποκαθιστά το μέγεθος χειρότερης περίπτωσης ενός ελάχιστου πιστοποιητικού συνεκτικότητας, δείχνοντας ουσιαστικά ότι τα πιστοποιητικά χρονικής συνεκτικότητας μπορεί να είναι ”πολύ πυκνά” ως γραφήματα.

Στο τελευταίο μέρος θεωρούμε τα προβλήματα βελτιστοποίησης της εύρεσης ενός πιστοποιητικού χρονικής συνεκτικότητας ελάχιστου κόστους, για δύο παραλλαγές, την single-source (r-MTC) και την all-pairs (MTC). Βρίσκουμε συνδέσεις μεταξύ του r-MTC και του Directed Steiner Tree, καθώς και μεταξύ του MTC και του Directed Steiner Forest. Μέσω διάφορων αναγωγών, αποδεικνύουμε τόσο άνω, όσο και κάτω φράγματα στην προσεγγισιμότητα για αυτά τα προβλήματα βελτιστοποίησης. Τέλος, μελετάμε ειδικές κλάσεις γραφημάτων στις οποίες τα παραπάνω προβλήματα έχουν αποδοτικότερους αλγορίθμους. Συγκεκριμένα, αναπτύσσουμε έναν αλγόριθμο πολυωνυμικού χρόνου για το r-MTC όταν το βασικό (underlying) γράφημα έχει φραγμένο δενδροπλάτος, για το MTC όταν το βασικό γράφημα είναι δένδρο, και έναν 2-προσεγγιστικό αλγόριθμο για το MTC όταν το βασικό γράφημα είναι κύκλος.

Τα βασικά αποτελέσματα αυτής της εργασίας παρουσιάστηκαν στο διεθνές συνέδριο International Colloquium on Automata, Languages, and Programming (ICALP - Track C) τον Ιούλιο του 2016. [Axio16]

Λέξεις κλειδιά

Αλγόριθμοι, Πολυπλοκότητα, Θεωρία γραφημάτων, Γραφήματα, Προσεγγιστικοί αλγόριθμοι, Χρονική συνεκτικότητα, Χρονικά μεταβαλλόμενα γραφήματα.

Abstract

In traditional graphs, a minimal connectivity certificate is called a spanning tree. We consider a generalization of the notion of connectivity for the more general class of temporal graphs with discrete time labels, where the minimum-size certificate is called a Minimum Temporally Connected Subgraph (MTCS).

In the first part, we present various applications which can be inherently modeled as temporal graphs and we survey results concerning temporal connectivity, like a generalization of Menger's theorem for temporal graphs.

Then, we present a novel construction which settles the worst-case size of a minimal connectivity certificate, basically showing that temporal connectivity certificates can be a ”very dense” graph.

In the last part we consider the optimization problems of determining the minimum-cost temporal connectivity certificate for both a single source (r-MTC) and an all-pairs (MTC) variant. We establish connections between r-MTC and Directed Steiner Tree, and between MTC and Directed Steiner Forest. Through various reductions, we establish both inapproximability and approximation results for the respective optimization problems. Finally, we study special graph classes in which these problems are easier. Specifically, we give a polynomial-time algorithm for r-MTC if the underlying graph has bounded treewidth, for MTC if the underlying graph is a tree, and a 2-approximation algorithm for MTC if the underlying graph is a cycle.

The basic results of this work were presented in the International Colloquium on Automata, Languages, and Programming (ICALP - Track C) in July 2016. [Axio16]

Key words

Algorithms, Complexity, Graph theory, Graphs, Approximation algorithms, Temporal connectivity, Temporal graphs.

Ευχαριστίες

Ευχαριστώ τον επιβλέποντα καθηγητή Δημήτρη Φωτάκη, που με καθοδήγησε κατά τη διάρκεια συγγραφής αυτής της εργασίας, καθώς και για τη γενικότερη βοήθειά του και συνεργασία σε ερευνητικά θέματα καθ’ όλη τη διάρκεια των σπουδών μου.
Ευχαριστώ επίσης όλα τα μέλη του Corelab για το φιλικό περιβάλλον που έχουν δημιουργήσει, το οποίο ενθαρρύνει την ενασχόληση με τη θεωρητική πληροφορική.
Τέλος, ευχαριστώ θερμά την οικογένειά μου για τη διαρκή υποστήριξη που μου προσφέρει όλα αυτά τα χρόνια.

Κυριάκος Αξιώτης,

Αθήνα, 26η Αυγούστου 2016

Contents

Chapter 1

Introduction

1.1 Motivation

Graphs and networks have always been a central and consistently studied subject in Computer Science, as they are able to model a lot of real-world systems in an abstract way. This abstraction has allowed computer scientists to focus out of the particular application at hand and treat graphs as abstract mathematical objects, thus being able to reason about their properties in a mathematically consistent way. These abstract results are then being tailored and used in specific applications. In fact, most of the interesting practical problems that we are facing today have an inherent graph structure and are thus naturally represented as graphs.

As an example, in computer networks, one may need to find a route (going through intermediate routers, bridges, firewalls, etc) through which to send a data packet from one router to another, so that the time it takes to reach its destination is minimized. This computational problem can be easily modeled as a graph: Computers and intermediate nodes translate to *vertices* of the graph, and links between these nodes, together with the delay that is associated to them, translate to *weighted edges* of the graph. The problem of finding the best route then simply translates to solving the well-studied *Shortest path problem* on the respective graph. In another example, consider a delivery company that needs to deliver its shipments to a number of clients across the city, using a single delivery truck. Of course, doing that in a naive way costs extra money and time. What needs to be as small as possible here is the total distance traveled to deliver all goods. This can again be easily modeled as a graph in which client locations translate to vertices, and pairs of clients, together with the associated distance between them, translate to weighted edges. Then, the problem one needs to solve in order to minimize the cost is the famous *Traveling salesman problem*. Other applications (among hundreds of them) include communication networks, evolving processes like epidemics and information spreading, social, chemical, and biological networks.

As computer science progresses and tries to solve real world problems in increasingly fast and accurate ways, sometimes new models must be invented so that they can adapt better to the form of the process that is being modeled. A simplistic example of this development of new models is the discovery of *directed* graphs to effectively capture scenarios where links between nodes can only be traversed in one direction; scenarios that traditional undirected graphs usually fail to capture.

One way in which traditional graph theoretic models are seriously limited with respect to adapting to real world scenarios is changes over time. In fact, traditional models assume that the networks studied are not subject to time-dependent variations. In other words, they are *static*: They are based on the simplification that the structure of the network, as well as the strength of interaction between its members is time-invariant. However, as we intend to describe here, this is very often not the case. In fact in many scenarios traditional graph models provide merely simplifications of the situation and one needs to search for more sophisticated models in order to achieve the desired expressiveness.

As observed in e.g., [Berm96, Kemp02], in many applications of graph theoretic models, the availability (edge existence) as well as the strength of interaction (edge weights) is actually time-dependent. Applications where this behavior is observed include transportation networks and route planning (e.g. [Berm96, Fosc14]), information spreading and distributed computation in dynamic networks (e.g. [Cast12, Dutt13, Kemp02, Kuhn10, Hede88, Ravi94]) mobile adhoc and sensor networks (e.g. [ODel05, Angl06]), epidemics, biological and ecological networks (e.g. [Holm12, Kemp02, Mich11]), influence systems and coevolutionary opinion formation (e.g. [Bhaw13, Chaz12]). As a result of the significant applicability, dynamic networks in general have attracted attention over the past few years (see for example [Avin08, Bui 03, Cast12, Dutt13, Mich14, Barj14, Whit12]). In the following, we argue about the time dependence of some of these applications.

Transportation Networks

In transportation networks and route planning (e.g. airline schedules, road traffic) routes are not always available. This could be due to maintenance or construction operations, planned strikes, and of course even scheduled changes depending on the time and date. Going a step further, even if there are not changes that render a connection unusable, its usefulness can vary greatly, depending on a lot of factors like traffic congestion, constructions, and delays because of weather. These factors don't necessarily alter the structure of the network, but change the weight of links between nodes, a change that we would like to be reflected on our model.

As an example, consider an airline network for Europe. An edge between two airports signifies the existence of a flight between these two airports. However, its mere existence does not imply that it is possible to traverse it at any time moment. In fact, flights are *scheduled* to happen at specific time moments, according to the airline's timetable. For example, if one wants to fly from Athens to Rome and then take another flight to Madrid in the same day, this may not be possible if all flights from Athens to Rome are scheduled after all flights from Rome to Madrid. This is maybe the simplest practical example that traditional graphs fail to model. Connectivity arguments like ”Starting from Athens, one can fly to reach any destination in Europe in the same day (directly or indirectly)” no longer make sense to be stated as a traditional graph connectivity question because of the impact of time.

Information Spreading, Distributed Computation

In information spreading, the entities that share information with each other are usually not broadcasting information at all times. Rather, they communicate at specific time moments (or time intervals) with other entities and exchange information during that time. The flow of information then is time-dependent, since some entity could end its information gathering before some other entity has the opportunity to broadcast its information. Therefore, again, traditional connectivity does not apply here.

An example of non-constant communication between entities sharing information is social networks, where people can start and end their relationships at any time, thus making the flow of information through the network time-dependent. A question of interest here is, supposing that every entity initial possesses some piece of information, is it possible for everyone to have complete information (everyone knows everything) after some time?

Mobile Adhoc and Sensor Networks

In mobile ad hoc and sensor networks, there is no wired infrastracture like routers, or managed wireless infrastracture like access points through which messages can be delivered. Instead, the routing is done in an ad hoc and evolving manner, going through various intermediate nodes. The routing itself, therefore, depends on the availability of the nodes as well as on the network connectivity strength between them. In practice, these parameters are anything but static, since nodes can suffer downtime due to maintenance or other causes and the strength of the signal depends on time-dependent factors like battery usage, network bandwidth, fluctuation of demand according to the time period, and the location of the node. This makes routing through these networks a fully time-dependent process and in order to study algorithms for routing one has to employ a model that takes these factors into account. Questions with respect to this application include finding the fastest route for a network packet if it exists, studying the number of nodes that have to fail so that it is no longer possible to transport a message to a specific target, and so on.

Epidemics, Biological and Ecological Networks

The *study of epidemics*, if considered in a fine-grained manner, is closely tied to the study of physical proximity. In order to study the evolving dynamics of a contagious disease, one needs to model physical interactions between the subject of study (humans, cattle, etc) that happen during a specified time period. This situation is inherently dynamic since the subjects are not in permanent contact, but rather communicate in restricted periods. For example, for studying the dynamics of a disease outbreak, one can obtain proximity information from hospitals. Patients that were hospitalized in the same hospital room (and, to a lesser extent, in the same hospital building) are more likely to have transmitted the disease to one another. The times, therefore, during which physical contact was being made, give useful information on determining the possible causes of a disease or even eliminating it by appropriately reducing the physical interactions. The impact of time information makes this scenario not a good fit for modeling by traditional graph structures.

Consider *biological applications* like the study of the brain or the dynamics of inter-molecule reactions. One can model different parts of the brain as vertices and add edges between them which model that the corresponding areas are in phase. Of course the existence of these edges depends on the activity exercised by the individual, as for example physical exercise activates different links between brain areas than mental exercise. Another scenario is the influence of electric signals and perturbations of magnetic fields on the brain network. It is possible that if we can model these processes in a more convenient and exact way, we will make way to the better understanding of elementary functions of the human brain.

Ecological networks are the ones that can be observed in any ecosystem. One discipline that could possibly benefit from the study of abstract time-changing models is the process of evolution. The evolution of species is primarily influenced by interactions between different populations, as well as weather conditions. Food scarcity can occur due to a change of the climate and result in the dramatic reduction or even extinction of various species. Both the interactions between populations and outer factors like climate, food scarcity, disease are inherently dynamic and cannot be effectively modeled by static models.

1.2 Models that take Time into Account

In order to overcome the inherent limitations of the traditional model that as we have now seen arise in a vast number of disciplines and applications, several variants of time-dependent graphs have been suggested as abstractions of such settings and computational problems (see e.g., [Cast12] and the references therein). No matter the particular variant, the main research questions are usually related either to optimizing or exploiting temporal connectivity or to computing short time-respecting paths (see e.g., [Akri15, Akri16, Berm96, Erle15, Fosc14, Kemp02, Mert13]). Other questions include time-dependent flow computations (see e.g. [Flei07, Flei98]).

[Berm96] first introduced the concept of *scheduled networks* in order to account for scheduled changes in the edge availability of a directed multigraph. In order to signify at which time an edge is available, they associated every edge with an interval of real numbers: The time interval during which the edge is available.

Afterwards, [Kemp02] introduced the slightly different model of *temporal graphs* with discrete time labels. Their model can be thought of as a sequence of traditional undirected graphs, each individual graph denoting the structure (edge availability) of the network during that time moment. More specifically, it can be considered as a usual undirected graph in which each edge is coupled with a time label: The time moment at which that edge is available. A significant advantage of this model is that it directly generalizes traditional graphs as temporal graphs that use the same single time label in all edges (i.e. all edges are available in exactly one time moment, that is the same among all edges).

[Mert13] then proceeded into a further generalization of the model of temporal graphs, allowing the introduction of multiple labels per edge. This is very natural since an edge may be available at more than one time moments. It can be easily seen that this modified model can easily express the model of [Berm96] and in fact is a more convenient model to deal with. One reason for that is that the notion of direction can very easily be expressed in this model as a sequence of two edges, each one available at a different time moment, so that one can only move in one direction. Additionally, there is no need to use real numbers instead of integers as labels, as the relationship between them is merely relative. Also continuous intervals can be modeled equivalently as a number of consecutive integer labels and interactions that also encure some delay can be modeled as length-2 paths, where the first edge of the path is available on the time moment of the beginning of the interaction, and the second edge is available at the end of the interaction. Overall, it seems that this model is both widely applicable and technically satisfying.

To argue in favor of applicability, we will outline the way that it can be used in some specific situations mentioned in Section 1.1.

For example for the airline schedule scenario, vertices of the temporal graph correspond to airports and edges correspond to flights between pairs of airports. The labels on each edge are exactly the time moments at which specific flights are available, if we assume that flights are instantaneous. If they are not, the option of length-2 paths mentioned above can be used instead. For the information spreading scenario, edges can signify the time moment at which communication between the two respective entities occured. For mobile adhoc networks, the edges are also equipped with nonnegative weights at each time moment, which signify the delay incurred along the edge, if it is used at that specific time moment.

In this thesis, we adopt the simple and natural model of temporal graphs with discrete time labels [Kemp02] (and its extension with multiple labels per edge [Mert13]).

1.3 Problems Concerning Temporal Graphs

Being the first to define a model closely related to temporal graphs, [Berm96] concerned themselves with connectivity questions arising from the study of temporal networks. All the questions that they studied revolved about generalizations of Menger's theorem. The usefulness of this study stems from the need for fault tolerance in networks. For example, in mobile ad hoc networks, the more alternative routes there exist between a pair of nodes, the less vulnerable the network to failures and downtime is. Broadly speaking Menger's theorem characterizes the relationship between the maximum number of alternative routes between a pair of vertices and the minimum number of failures to separate these two vertices, stating that they are equal. [Berm96] provides a polynomial-time algorithm for measuring the fault tolerance of temporal graphs: Given a number k, the algorithm computes the time period during which the deletion of k edges does not affect connectivity between a pair of vertices. However, they note that Menger's theorem does not generalize for temporal graphs, either when referring to vertex or edge failures, and that, in fact, the problem of finding alternative routes is, in fact, an NP-hard problem.

[Kemp02] continued the study of generalizations of Menger's theorem with a complete classification of networks in which Menger's theorem is true. In fact, in a pretty nice result, they proved that this class of graphs has a specific excluded minor. In other words, the generalization of Menger's theorem works in all labelings of a graph if and only if that graph does not contain a specific graph as a minor. Moreover, they provided an algorithm that computes the maximum number of alternative paths between two vertices, provided that the aforementioned condition holds for the graph. In the second part of their work, they considered *Inference problems*, in which one is given incomplete information about the time availability of edges and seeks to compute consistent time availability information for the whole graph such that certain conditions are met. They provide an algorithm that, given an incomplete labeling, tries to complete the missing labels in a valid way so that connectivity holds.

Trying to find a workaround that makes the generalized Menger's theorem hold, [Mert13] provide a different formulation that also takes the temporal nature of the graph into account and, in fact, prove its correctness. Instead of concerning vertex or edge failures, it concerns failure to exit a vertex at a specific time moment. Their proof is immediately yields an algorithm that can efficiently compute the maximum number of routes to follow, no two of which share a single point of failure (exit at the same node at the same time). They also give a simple algorithm for finding short time-respecting paths. Finally, dealing with the design of temporal networks, they propose parameters that potentially characterize ”good” networks and should be minimized. They study the optimization of these parameters subject to connectivity constraints. The first parameter is the maximum number of time moments at which an edge is available, and the second is the total number of labels used.

Departing from connectivity related questions, [Erle15] consider the problem of *temporal exploration*, in which one needs to explore all vertices of the temporal graph as soon as possible. Considering temporal graphs that are connected at each time moment, they prove that even the crudest approximation for minimizing the cost of the exploration is an NP-hard problem. However they show that if one restricts the problem to specific graph families, one can find efficient algorithms. Finally, they show that not all temporal graphs can be explored using few edges, by constructing an infinite family of temporal graphs for which this is the case.

Studying the problem of *designing* temporal networks that are *connected*, [Akri15] consider the design of such networks that at the same time have low *cost* (availability instances used). If there are no more constraints, they show that there is an algorithm that achieves nearly optimal cost. However, if one introduces the extra constraint of having to pick availability times from a specified set, the problem of removing the maximum number of time labels is not susceptible of arbitrarily close approximation. Finally, inspired by (static) works concerning stochastic and survivable network design (e.g. [Gupt12, Lau09, Lau13]), they consider temporal graphs with *random* edge availabilities and show that almost surely a lot of time labels can be removed, thus random temporal graphs that preserve connectivity can almost always be considered sparse.

[Huan15] consider single-source temporal connectivity (enforces routes from one source to every other vertex) in two optimization variants: Minimizing the maximum time needed, or minimizing the total cost, if edges are coupled with nonnegative costs at each time moment that they are available. They show that the first problem can be solved in linear time, but the second one is much harder. They show that it is hard by establishing a connection to a notoriously hard problem (Directed Steiner Tree).

Considering temporal graphs with random edge availability, [Akri16] define and study a generalization of graph diameter and characterize temporal graphs as *slow* or *fast*, according to their expected diameter size.

As is elucidated by the direction of previous work, one of the most important type of questions that have to do with temporal networks is related to connectivity. This is because connectivity is an elementary property that is very important in practice. It manifests that it is possible to reach the whole network, an attribute that usually is in direct correlation with the overall quality of the network that ones seeks to design, improve, or test. In this thesis, we delve into temporal connectivity questions.

One of the most basic questions that we answer here concerns the *size of temporal connectivity certificates*. Given a temporal network that is connected, it is natural for one to seek to sparsify the network while still preserving the desirable connectivity property. This is important because in a lot of applications (with infrastructure networks like the internet or phone lines being the most notable), a large number of links between nodes incur a great cost in the design and usage of the network. Therefore, it is natural for one to seek to design networks that are both cheap (have few links) and connected (an essential property). The question to ask, therefore, is whether it is true that for every temporal network, we can preserve connectivity, while keeping only few of the links and deleting all others? This question was first posed by [Kemp02]. We will see in Chapter 3 that the answer to

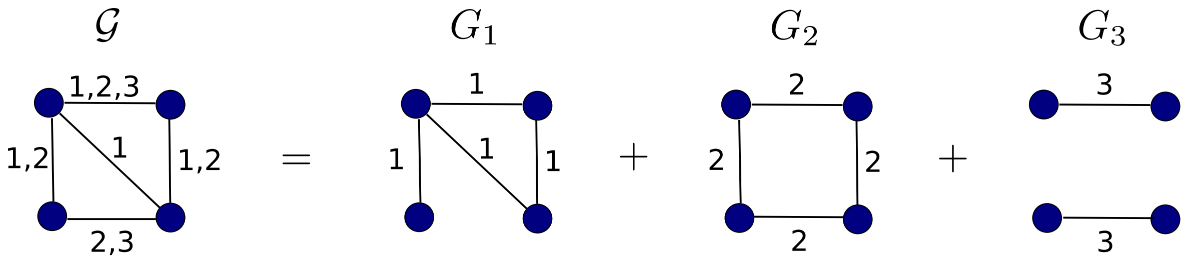


Figure 1.1: A temporal graph $\mathcal{G}$ with lifetime 3 breaks down into three usual undirected graphs G_1, G_2, G_3, one for each time label in [3]

this question is negative. This means that there exist temporal networks in which one should keep close to all links in order to preserve connectivity. This fact goes in contrast with our usual intuition about traditional (directed and undirected) graphs, in which only few links are needed. We conclude, therefore, that connectivity information in temporal networks is much more involved and complex than in traditional models.

For the same reasons that we mentioned above, the question of computing close to optimal connectivity certificates for temporal graphs is relevant and interesting. In this context, we consider two different kinds of connectivity that lead to two optimization problems: *Single-source* connectivity, in which every vertex should be reachable from a signified source, and *All-pairs* connectivity, in which every vertex should be reachable from every other vertex. We show that, for the general case with weighted edges, there exist approximation algorithm yet with bad approximation guarantees to the optimal certificates. To justify this, we show that achieving a significantly better approximation for these problems is computationally infeasible. Therefore we restrict the network structure in order to be able to obtain fast algorithms that approximate the optimal solution better. We consider, therefore, various cases of special graphs and introduce algorithms that either solve the problem exactly, or achieve a good approximation guarantee. The conclusion is that for general graphs the problem of designing low-cost temporal networks subject to connectivity constraints is a hard problem, but if the given network has a specific structure one can sometimes achieve close to optimal results.

1.4 Basic Definitions

1.4.1 Temporal Graphs

A *temporal graph* is defined on a time-invariant set of n vertices. Each (undirected) edge e is associated with a set of discrete time labels denoting when e is available. If every edge is associated with a single time label, as in [Kemp02], the temporal graph is *simple*. An edge e available at time t comprises a temporal edge (e, t) and there is a positive weight $w(e, t)$ associated with it.

Definition 1.4.1 (Temporal Graphs)**.** An (edge weighted) *temporal graph* $\mathcal{G}(V, E, L)$ with vertex set V, edge set E and lifetime L is a sequence of (undirected edge-weighted) graphs $(G_t(V, E_t, w_t))_{t\in[L]}$, where $E_t \subseteq E$ is the set of edges available at time t and $w_t(e)$ (or $w(e, t)$) is the nonnegative weight of each edge $e \in E_t$ at time t. We often write $\mathcal{G}$ or $\mathcal{G}(V, E)$, for brevity. A temporal graph $\mathcal{G}$ is *unweighted* if $w(e, t) = 1$ for all $e \in E_t$ and all $t \in [L]$. For each edge $e \in E_t$, we say that (e, t) is a *temporal edge* of $\mathcal{G}$. For each edge $e \in E$, $L_e = \{t \in [L] : e \in E_t\}$ denotes the set of time units (or *time labels*) when e is available. A temporal graph is *simple* if $|L_e| = 1$ for all edges $e \in E$. See also Figure 1.1 for the example of a temporal graph with lifetime 3, as well as the 3 usual graphs that compose it.

The temporal analogue of a usual path is called a *temporal path*. A temporal path is like a usual path, but additionally respects the time availability constraints of the edges. It is a usual path whose edges are equipped with non-decreasing time labels along the path.

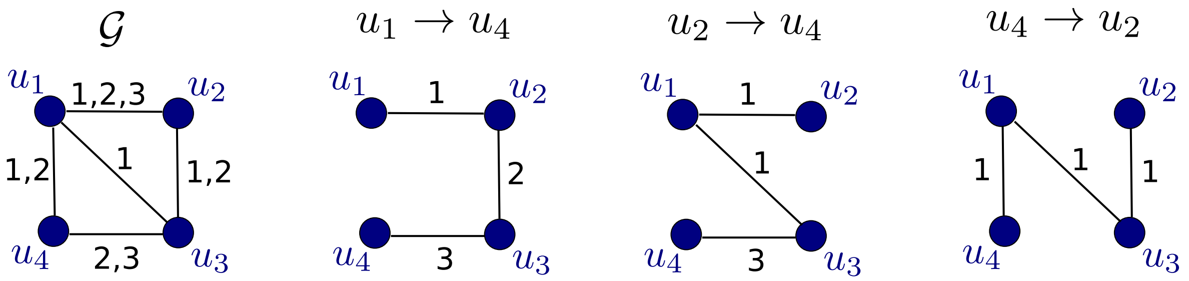


Figure 1.2: A temporal graph $\mathcal{G}$ and three temporal paths on it: One strict temporal path from u_1 to u_4 and two non-strict from u_2 to u_4 and from u_4 to u_2 respectively

Definition 1.4.2 (Temporal Paths)**.** A *temporal* (or *time-respecting*) path is an alternating sequence $(v_1, (e_1, t_1), v_2, (e_2, t_2), \dots, v_k, (e_k, t_k), v_{k+1})$ of vertices v_i and temporal edges (e_i, t_i), such that $e_i \equiv \{v_i, v_{i+1}\} \in E_{t_i}$, for all $i \in [k]$, and $1 \leq t_1 \leq t_2 \leq \cdots \leq t_k$. A temporal path is *strict* if $t_1 < t_2 < \cdots < t_k$. Such a temporal path is from v_1 to v_{k+1} (or a temporal $v_1 - v_{k+1}$ path). See also Figure 1.2 for the example of a temporal graph together with three temporal paths on it.

We usually let n denote the number of vertices and $M = \sum_e |L_e|$ denote the number of temporal edges of $\mathcal{G}$. For temporal connectivity problems, we can assume wlog. that at least one temporal edge is available in each time unit, and thus, $L \leq M$.

Definition 1.4.3 (Underlying Graphs)**.** The (static) graph $G(V, E)$ is the *underlying graph* of $\mathcal{G}(V, E, w)$. We say that $\mathcal{G}$ has some (non-temporal) graph theoretic property (e.g., is a tree, a cycle, a clique, has bounded treewidth) if the underlying graph G has this property. If the underlying graph of $\mathcal{G}$ is in a specific graph family X, we will say that $\mathcal{G}$ is a temporal X. For example, if the underlying graph of $\mathcal{G}$ is a tree, then $\mathcal{G}$ is a temporal tree. For a vertex set S, $G[S]$ (resp. $\mathcal{G}[S]$) is the underlying (resp. temporal) graph induced by S.

Definition 1.4.4 (Spanning Subgraphs)**.** A *(temporal) spanning subgraph* $\mathcal{G}'$ of a temporal graph $\mathcal{G} \equiv (G_t(V, E_t, w_t))_{t \in [L]}$ is a sequence of graphs $(G'_t(V, E'_t, w_t))_{t \in [L]}$ such that $E'_t \subseteq E_t$. The total weight of $\mathcal{G}'$ is $\sum_{t \in [L]} \sum_{e \in E'_t} w(e, t)$.

Definition 1.4.5 (MTCS)**.** A *Minimum Temporally Connected Subgraph* (MTCS) of a temporal graph $\mathcal{G}$ is a temporally connected spanning subgraph of $\mathcal{G}$ that has the minimum number of edges. Equivalently, it is also called a *Minimum Connectivity Certificate*.

1.4.2 Temporal Connectivity

Given a source vertex r, a temporal graph is (temporally) *r-connected* if there is a temporal path from r to any other vertex. A temporal graph is (temporally) *connected* if there exists a temporal path from any vertex to any other vertex.

Definition 1.4.6 (Temporal connectivity)**.** A temporal graph $\mathcal{G}$ is (temporally) *r-connected*, for a given source $r \in V$, if for all $u \in V$ there exists a temporal path from r to u. A temporal graph $\mathcal{G}$ is (temporally) *connected*, if for all ordered pairs $(u, v) \in V \times V$ there exists a temporal path from u to v. If all temporal paths are strict, $\mathcal{G}$ is strictly connected (or strictly r-connected). A(n) *(r-)connectivity certificate* of $\mathcal{G}$ is any spanning subgraph of $\mathcal{G}$ that is also (r-)connected.

We study the existence of dense minimally connected temporal graphs and the optimization problems of computing a minimum weight subset of temporal edges that preserve either r-connectivity or connectivity. We refer to these optimization problems as (Minimum) Single-Source Temporal Connectivity (or r-MTC, in short) and (Minimum) All-Pairs Temporal Connectivity (or MTC, in short). They arise as natural generalizations of Minimum Spanning Tree and Minimum Arborescence in temporal networks.

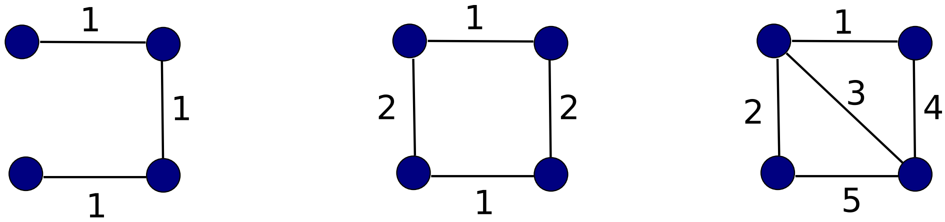


Figure 1.3: Three temporal graphs on 4 vertices, which have minimum connectivity certificate sizes 3, 4, and 5 respectively.

Definition 1.4.7 (Single-Source Connectivity)**.** Given a temporal graph $\mathcal{G}$ and a source vertex r, the problem of (Minimum) *Single-Source Temporal Connectivity* (r-MTC) is to compute a temporally r-connected spanning subgraph of $\mathcal{G}$ with minimum total weight. The optimal solution to r-MTC is a simple temporal graph whose underlying graph is a tree (see [Kemp02, Section 6] and Lemma 3.1.1).

While for r-MTC the optimal solution is always a tree, as we will prove in Section 3.1, for MTC this is far from being true. In fact, the size of the connectivity certificate is not a function of n but also depends on the structures of the temporal graph itself. This is illustrated in Figure 1.3, where we present three simple and minimally connected temporal graphs with 4 edges, each of which has a different number of edges, so a different connectivity certificate size.

Definition 1.4.8 ((All-Pairs) Temporal Connectivity)**.** Given a temporal graph $\mathcal{G}$, the problem of (Minimum) *All-Pairs Temporal Connectivity* (MTC) is to compute a temporally connected spanning subgraph of $\mathcal{G}$ with minimum total weight.

1.4.3 Approximation Algorithms

An approximation algorithm is an algorithm whose output is guaranteed to approximate the optimal answer to the problem by a certain multiplicative factor. This factor is called the *approximation ratio* of the algorithm.

Definition 1.4.9 (Approximation Ratio)**.** An algorithm A has *approximation ratio* $\rho \geq 1$ for a minimization problem if for any instance I, the cost of A on I is at most ρ times I's optimal cost.

1.5 Contribution

In this work, we first consider the size of connectivity certificates for temporal graphs. As we have mentioned before, this is a parameter that is very important to keep small when designing temporal networks. In order to evaluate the worst possible case for general temporal graphs, we seek to bound the maximum possible connectivity certificate size over all temporal graphs. This will give us the worst that we can expect of temporal graphs in general. For r-connectivity, it is trivial to see that the size is always $n-1$, since as we will prove in Section 3.1 an r-connectivity certificate is always a tree. For all-pairs connectivity, however, things are much more complicated. As [Kemp02] first noted, the size of the connectivity certificate is not a function of the number of vertices n, but also depends on the structure of the temporal graph. They asked, therefore, what is the tightest function $c(n)$ such that all simple temporal graphs on n vertices have connectivity certificates of size at most $c(n)$? In Section 3.3 we will give their proof that shows that $c(n) = \Omega(n \log n)$. [Akri15] show that this lower bound holds also even if we are restricted to discrete time labels. Moreover, it is easy to see that $c(n) = O(n^2)$, which we will prove in Section 3.2. Note that for connectivity by strict temporal paths the answer is trivially $\Theta(n^2)$, since we can take the complete graph with the same time label in every edge. In this thesis we close the large gap between $\Omega(n \log n)$ and $O(n^2)$ by showing that $g(n) = \Theta(n^2)$ in Section 3.4. To do that, we construct an infinite family of temporally connected graphs and prove that for this family keeping less than quadratic number of edges results in a temporal graph that is no longer temporally connected. It is not hard to tweak our construction to work with discrete time labels, so our result basically improves all the previous results and settles (up to constant multiplicative factors) that $c(n) = \Theta(n^2)$. Moreover, our construction is also tight with respect to the lifetime, since the temporal graphs constructed have linear lifetime, and it is easy to see that with sublinear lifetime one can only achieve subquadratic connectivity certificate size. Furthermore, our construction can be adapted to show the more general result that for any lifetime $L \leq n$, simple temporal graphs with lifetime at most L have $c(n) = \Theta(nL)$.

In the rest part of this work we derive upper and lower bounds on the approximability of Single-Source and All-Pairs Temporal Connectivity. Given the huge gap on the size of temporal connectivity certificates, it is natural to ask about the complexity and the approximability of Single-Source and All-Pairs Temporal Connectivity. Previous work shows that we can decide if a temporal graph is connected in polynomial time (see e.g., [Akri15, Berm96, Kemp02]) and that Single-Source Temporal Connectivity can be solved in polynomial time in the unweighted case. Another interesting observation is that if we use the time-expanded version of a temporal graph for Minimum Temporal Connectivity, the resulting optimization problems are quite similar to Group Steiner Tree problems. In fact, this observation serves as the main intuition behind several of our results.

In Chapter 4, we consider the optimization problem r-MTC. The main thing we show is that this problem has strong ties with a classical graph problem, Directed Steiner Tree. Specifically, exploiting similarities between edge direction and pairs of temporal edges, we present a reduction from r-MTC to Directed Steiner Tree (Theorem 4.1.1) we show that r-MTC cannot be approximated within a ratio of $O(\log^{2-\varepsilon} n)$, for any constant $\varepsilon > 0$, unless NP $\subseteq$ ZTIME($n^{\text{poly}\log n}$). Our transformation also implies that any $o(n^{\varepsilon})$-approximation for r-MTC would improve the best known approximation ratio of the well studied Directed Steiner Tree. On the positive side, using a transformation from r-MTC to Directed Steiner Tree and the algorithm of [Char99], we obtain a polynomial-time $O(n^{\varepsilon})$-approximation, for any constant $\varepsilon > 0$, and a quasipolynomial-time $O(\log^3 n)$-approximation for r-MTC (Theorem 4.2.1). We also show that r-MTC can be solved in polynomial time if the underlying graph has bounded treewidth (Theorem 4.3.5). Independently, [Huan15] also presented an algorithm for rTC as well as approximation hardness results for it, yet significantly weaker than ours. Specifically, they present a reduction from r-MTC to Directed Steiner Tree that is similar to ours and achieves the same approximation guarantees for r-MTC. Moreover, they present reductions from NP-complete problems to r-MTC, thus establishing that r-MTC is APX-hard (has no polynomial-time approximation scheme unless $P = NP$). However, the reductions that they employ (using problems different than Directed Steiner Tree) turn out to be much weaker than ours, as confirmed by our reduction from Directed Steiner Tree that shows polylogarithmic inapproximability.

In Chapter 5, we consider the approximability of All-Pairs Temporal Connectivity (MTC). Theorem 4.2.1 implies an $O(n^{1+\varepsilon})$-approximation for MTC (Corollary 5.1.1). An approximation-preserving reduction to Directed Steiner Forest and [Feld12, Theorem 1.1] imply a polynomial-time $O((\Delta M)^{2/3+\varepsilon})$-approximation for MTC, , where M is the number of temporal edges and Δ is the maximum degree of the underlying graph (Theorem 5.1.2). If M is quasilinear and Δ is polylogarithmic, we obtain an $O(n^{2/3+\varepsilon})$-approximation. On the negative side, a reduction from Symmetric Label Cover implies that MTC cannot be approximated within a factor of $O(2^{\log^{1-\varepsilon} n})$ unless NP $\subseteq$ DTIME($n^{\text{poly}\log n}$) (Theorem 5.2.1, see also [Dodi99, Section 4]). We also show that the unweighted version of MTC is APX-hard (Theorem 5.3.1). A somewhat similar result to this is present in [Akri15]. Specifically, the authors reverse the objective by studying the problem of *deleting* the maximum number of temporal edges, while still preserving temporal connectivity. The only significant difference between this problem and unweighted r-MTC is that the objective is to remove the maximum number of temporal edges, instead of keeping the minimum number of them. The authors show that this ”reversed” problem is APX-hard, by employing a reduction from the *Monotone Max-XOR(3)* problem (which essentially encodes the *Max-Cut* problem on 3-regular graphs). However, it turns out that the only implication that this result has on r-MTC is its NP-hardness, while we present a stronger APX-hardness result. The reason is that the temporal graphs that serve as hard instances in their reduction are easy instances for our problem. Specifically, they have $\frac{17}{4}n^2 + 28n + 1$ temporal labels, so they are pretty dense, and if k is the optimal value of the Monotone Max-XOR(3) instance, then the maximum number of labels that can be deleted is $9n + k$. However, for our minimization problem, this translates to keeping $\frac{17}{4}n^2 + 19n + 1 - k$ labels, where $k \leq \frac{3}{2}n$. Achieving a PTAS for our problem accounts to, for given $\varepsilon > 0$, computing a solution that keeps at most $(1+\varepsilon)(\frac{17}{4}n^2 + 19n + 1 - k)$ temporal edges. But as we will see in Section 3.2, it is easy to find a solution with n^2 temporal edges, and $(1+\varepsilon)(\frac{17}{4}n^2 + 19n + 1 - k) \geq (1+\varepsilon)(\frac{17}{4}n^2) \geq n^2$, so we can always easily achieve a PTAS for these instances.

In Chapter 6, we show that MTC can be solved optimally, in polynomial time, if the underlying graph is a tree (Theorem 6.1.1), and that MTC is 2-approximable if the underlying graph is a cycle (Theorem 6.2.1, but it is open whether MTC remains NP-hard for cycles).

For clarity, we focus on connectivity by (non-strict) temporal paths. However, all our results can be extended (with small changes in the proofs and with the same approximation guarantees and running times) to the case of connectivity by strict temporal paths.

Chapter 2

Connectivity in Temporal Graphs

The model of simple temporal graphs with discrete time labels was introduced in [Kemp02]. It is essentially equivalent to the model of scheduled networks [Berm96], where each edge is available in a time interval. [Berm96, Kemp02] investigated how time availability restrictions on the edges affect certain graph properties. Berman [Berm96] presented an algorithm for reachability by temporal paths and proved that an analogue of the max-flow-min-cut theorem holds for temporal graphs. Kempe et al. [Kemp02] focused on vertex-disjoint temporal paths and showed that Menger's theorem does not generalize to temporal graphs. They also identified a simple forbidden topological minor for Menger's theorem in temporal graphs. Mertzios at al. [Mert13] introduced multiple labels per edge and studied the number of temporal edges required for a temporal design to guarantee certain graph properties. Interestingly, they proved that a variant of Menger's theorem, which also takes time into account, holds in all temporal graphs. A key technical tool in [Berm96, Kemp02, Mert13] is the time-expanded version of a temporal graph, which reduces reachability, edge-disjoint path and vertex-disjoint path questions in temporal graphs to similar questions in standard directed graphs.

[Akri15] shows that computing the maximum number of edges that are redundant for temporal connectivity is APX-hard.

Erlebach et al. [Erle15] study the problem of computing a shortest exploration schedule of a temporal graph, i.e., a shortest strict temporal walk that visits all vertices. They prove that it is NP-hard to approximate the shortest exploration schedule within a factor of $O(n^{1-\varepsilon})$, for any $\varepsilon > 0$, and construct temporal graphs whose exploration requires $\Theta(n^2)$ steps. Since the notion of exploration schedules is much stronger than (r-)connectivity, their results do not have any immediate implications for r-MTC and MTC (e.g., the $\Theta(n^2)$-explorable graphs of [Erle15, Lemma 4] admit a temporally connected subgraph with $O(n)$ edges).

2.1 Time Expansion

Here we describe the *Time expansion* of a temporal graph, an associated usual directed graph, as described in [Mert13]. Some reachability questions on temporal graphs can therefore be equivalently asked on directed graphs.

Definition 2.1.1 (Time Expansion)**.** Given a temporal graph $\mathcal{G}$ with labels in $[L]$, the *time expansion* of $\mathcal{G}$ is a directed graph H constructed as follows:

For each vertex-time pair $(u, t) \in V(\mathcal{G}) \times [L + 1]$, H has a vertex $h_{u,t}$. Moreover, for each temporal edge (e, t) of $\mathcal{G}$ with endpoints u and v, H has a directed edge from $h_{u,t}$ to $h_{v,t+1}$ and a directed edge from $h_{v,t}$ to $h_{u,t+1}$. Finally, for every $u \in V(\mathcal{G})$ and $t \in [L]$, H has an edge from $h_{u,t}$ to $h_{u,t+1}$.

Intuitively, $h_{u,t}$ keeps not only the information of on which vertex we are on, but also at which time, which ensures that all directed paths in H correspond to time-respecting paths in $\mathcal{G}$. Note that edges of the form $(h_{u,t}, h_{u,t+1})$ encode the fact that an agent moving on temporal paths can choose to wait on some vertex, before traversing the next temporal edge.

The above construction can be used when working with *strict* temporal paths (or when all time labels are distinct), but similar constructions can be employed to also allow non-strict temporal paths. We will see such variations in our reductions in Chapter 4 and Chapter 5.

2.2 Checking Temporal Connectivity

In contrast to finding temporally connected subgraphs of small cost, checking that a temporal graph is temporally (r-)connected can easily be done in polynomial time. Here we will present algorithms that, given a temporal graph $\mathcal{G}$ decide whether it is temporally (r-)connected or not.

***Theorem* 2.2.1.** *Given a temporal graph $\mathcal{G}(V, E)$ with lifetime L and M temporal edges, there exists an algorithm that runs in time $O(M)$ and decides whether $\mathcal{G}$ is temporally r-connected. For dense graphs, the time complexity is $O(n^2 L)$.*

Proof. Let r be the source vertex. We will process each time label $t \in [L]$ one by one in increasing order. Let S_t denote the subset of $V(\mathcal{G})$, that are vertices reachable from the source using only labels in $[t]$. Note that $S_0 = \{r\}$ and $S_L = V(\mathcal{G}) \Leftrightarrow \mathcal{G}$ r-connected.

When processing edges with label t, we use DFS to identify the connected components of G_t, say $C_1, \ldots, C_k$. Note that the components that have no vertex in common to S_{t-1} cannot be connected yet, while those that do have, can. Therefore, we have that $S_t = S_{t-1} \cup \bigcup\{C_i \mid i \in [k],\ C_i \cap S_{t-1} \neq \emptyset\}$. Both the DFS and the last part take linear time in the size of G_t, so the total complexity is linear, $O(M)$. □

***Corollary* 2.2.2.** *Given a temporal graph $\mathcal{G}(V, E)$ with lifetime L and M temporal edges, there exists an algorithm that runs in time $O(nM)$ and decides whether $\mathcal{G}$ is temporally connected. For dense graphs, the time complexity is $O(n^3 L)$.*

Proof. We simply run the r-connectivity algorithm for each vertex as source, and only answer 'yes' if every instance of that algorithm answers 'yes'. □

Note: Given the time expansion construction of Section 2.1, it is even easier to describe an alternative algorithm that checks whether a temporal graph $\mathcal{G}$ is r-connected. Let H be the time expansion of $\mathcal{G}$. (For *non-strict* temporal paths, we have to use a slightly different time expansion. For example, for each temporal edge (e, t) with $e \equiv \{u, v\}$, instead of adding directed edges $(h_{u,t}, h_{v,t+1})$ and $(h_{v,t}, h_{u,t+1})$, we add directed edges $(h_{u,t}, h_{v,t})$ and $(h_{v,t}, h_{u,t})$. Even though this tweak does not work for most other cases, it works for reachability)
For r-connectivity with source r, it suffices now to check whether for all $u \in V(\mathcal{G})$ there is a directed path from r to $h_{u,L+1}$ in H. This is one example of the importance of the time expansion construction, as we have reduced a question on temporal graphs to a well-known one in usual directed graphs with minimal effort.

2.3 A Generalization of Menger's Theorem

In this section we present an interesting temporal analogue of Menger's theorem, first described by [Mert13]. It follows easily by a direct application of the Max flow Min cut theorem on a graph similar to the time expansion of the temporal graph, as described in Section 2.1. Before we present this result, we introduce some terminology.

Definition 2.3.1. Two temporal paths p_1 and p_2 are called *out-disjoint* (resp. *in-disjoint*) if they do not leave from (resp. arrive to) some vertex at the same time. In other words, two temporal paths p_1 and p_2 are *not* out-disjoint iff there is some vertex u and some time label t, such that both p_1 and p_2 depart from u at time t.

Now it is time to state the theorem.

***Theorem* 2.3.2** (Temporal Menger's Theorem)**.** *Let $\mathcal{G}$ be a temporal graph and a, b two of its vertices. Then, the maximum number of (mutually) out-disjoint strict temporal paths from a to b is equal to the minimum number of "vertex-departure time" pairs that should be forbidden so that there is no strict temporal path from a to b.*

A temporal path with forbidden "vertex-departure time" pair (u, t) is any temporal path that does not depart from vertex u at time t.

Proof. For convenience, in this proof we will be always referring to *strict* temporal paths, even if not explicitly using the word "strict".

Let the directed graph H be the time expansion of $\mathcal{G}$, with the following difference: Instead of having edges of the form $(h_{u,t}, h_{v,t+1})$ with $u \neq v$, we introduce new vertices of the form $w_{u,t}$ and add edges from $h_{u,t}$ to $w_{u,t}$ and from $w_{u,t}$ to $h_{v,t+1}$ for all appropriate v.

Intuitively, any path in H corresponds to a temporal path in $\mathcal{G}$ and vice versa. Moreover, forbidding a single "vertex-departure time" pair (u, t) in a temporal path in $\mathcal{G}$ is equivalent to forbidding an edge of the form $(h_{u,t}, w_{u,t})$ in a path in H. Finally, two out-disjoint temporal paths in $\mathcal{G}$ correspond to two edge-disjoint paths in H. From the two last sentences the purpose of the new vertices $w_{u,t}$ that we introduced becomes clear: It is so that no multiple edges of the form $(h_{u,t}, h_{v,t+1})$ for fixed u, t exist, which ensures out-disjoint temporal paths.

Moreover, in order to use the Max flow Min cut theorem on H we need to add capacities to the edges. For each edge of the form $(h_{u,t}, h_{u,t+1})$ we use capacity $+\infty$, and for all other edges we use capacity 1. Intuitively, each normal (non-waiting) edge should be visited at most once (hence the capacity 1), while each waiting edge can be used by any number of paths (but the flow will be at most L).

In order to continue, we define the following quantities:
A: The maximum number of out-disjoint temporal paths from a to b in $\mathcal{G}$
B: The minimum number of "vertex-departure time" pairs that should be forbidden so that there is no temporal path from a to b in $\mathcal{G}$
C: The maximum flow from $h_{a,1}$ to $h_{b,L+1}$ in H
D: The minimum cut from $h_{a,1}$ to $h_{b,L+1}$ in H.

The Max flow Min cut theorem immediately implies that $C = D$. We will prove that (i) $A = C$ and (ii) $B = D$, therefore it will follow that $A = B$.
(i): It is easy to see that each temporal path $u_1, (e_1, t_1), \ldots, (e_k, t_k), u_k$ in $\mathcal{G}$ corresponds to a directed path from $h_{u_1,1}$ to $h_{u_k,L+1}$ in H: If we are at $h_{u,t}$ and the next temporal edge in the temporal path is (e, t') with $e \equiv \{u, v\}$ and $t' > t$, then the directed path follows the sequence of vertices
$h_{u,t+1}, \ldots, h_{u,t'}, h_{v,t'+1}$. With a similar transformation, to each directed path in H from $h_{u_1,1}$ to $h_{u,k,L+1}$ corresponds a temporal path from u_1 to u_k in $\mathcal{G}$: Each transition from $h_{u,t}$ $w_{u,t}$ and then to $h_{v,t+1}$ corresponds to simply using the edge $e \equiv \{u, v\}$ at time t to move from u to v. Note that the maximum flow from $h_{a,1}$ to $h_{b,L+1}$ in H is also equal to the maximum number of edge-disjoint paths from $h_{a,1}$ to $h_{b,L+1}$. Furthermore, using the above transformation, a set of edge-disjoint (except for edges $(h_{u,t}, h_{u,t+1})$) paths from $h_{a,1}$ to $h_{b,L+1}$ translates to the same number of out-disjoint paths from a to b, since edge-disjointness implies that there will be no more than one paths visiting $w_{u,t}$ for any u, t, and therefore no more than one temporal paths departing from u at time t. The converse is also true, since similarly out-disjointness also implies edge-disjointness. From the above we get that $A \geq c \Leftrightarrow C \geq c$, therefore $A = C$.
(ii): First of all, note that some minimum cut of H separating a and b will only contain edges of the form $(h_{u,t}, w_{u,t})$. To see this, note that if it cannot contain an edge of infinite capacity and that if it contained any edge of the form $(w_{u,t}, h_{v,t+1})$, then one would be able to cut $(h_{u,t}, w_{u,t})$ instead, as this will neither increase the cut or introduce a path from $w_{u,t}$ to $h_{v,t+1}$.

With this in mind, note that if for every edge $(h_{u,t}, w_{u,t})$ of this minimum cut one forbids departing from vertex u at time t in $\mathcal{G}$, then it is no longer possible to find a path from a to b in $\mathcal{G}$. If it were, then it would contradict the assumption that would have a cut. Conversely, forbidding any subset of "vertex-departure time" pairs from $\mathcal{G}$ to isolate b from a can be turned into cutting the same number of directed edges from H, so that there is no directed path from $h_{a,1}$ to $h_{b,L+1}$. If it existed, then a temporal path from a to b in $\mathcal{G}$ would also exist. Therefore, we have that $B = D$.

From the above we have that $A = C = D = B$, so the temporal analogue of Menger's theorem is proven. □

***Corollary* 2.3.3.** *By symmetry, it follows that for every temporal graph $\mathcal{G}$, the maximum number of in-disjoint temporal paths from some $a \in V(\mathcal{G})$ to some $b \in V(\mathcal{G})$ is equal to the minimum number of "vertex-arrival time" pairs that should be forbidden so that there is no temporal path from a to b in $\mathcal{G}$.*

We note here that there is also an alternative way to prove the temporal analogue of Menger's theorem, without the use of the Max flow Min cut theorem, but with direct application of Menger's theorem in a directed graph associated with the temporal graph. The same construction is used in Section 4.2 as a way to reduce r-MTC to Directed Steiner Tree.

Chapter 3

The Size of Minimum Temporally Connected Subgraphs (MTCS)

As we have already mentioned, the study of temporal graphs aims to fill the gap that traditional (static) graph theory leaves in the expressibility of real-world problems that are innately temporal. To this end, we have described some generalizations of graph-theoretic notions in temporal graphs, that make it much easier to express and reason about temporal applications.

As an example of the expressibility of temporal graphs, consider the following setting: n people hold meetings for a number of days and each day a certain set of (unordered) pairs of people denotes the pairs of people that have a meeting with each other (suppose that the meetings held in the same day are held simultaneously). This setting can be easily expressed by a temporal graph whose vertices correspond to people and temporal edges to 2-person meetings, each of whose label is the day that the meeting is being held. What does temporal r-connectivity translate to? Say that some person r has some piece of information and that during any meeting, the people attending it exchange all the information they have. Then the temporal graph being temporally r-connected is equivalent to r's information having been successfully broadcast to everyone after the last meeting. What does temporal connectivity translate to? Everyone has some piece of information initially, and after the last meeting every piece of information is known to every person.

Since we have defined the temporal analogue of spanning trees (Minimum Temporally Connected Subgraphs), a natural question to ask now concerns the size of MTCS. Being the first to study this, Kempe, Kleinberg, and Kumar ([Kemp02]) noted that unlike in spanning trees, MTCS size does not only depend on the number of nodes n, but also on the structure of the temporal graph itself. For example for $n = 4$, there are temporal graphs for any MTCS size in the set $\{3, 4, 5\}$.

The natural question to ask is, therefore, what is the largest possible MTCS size? Specifically, as asked in [Kemp02], what is the tightest function $c(n)$ for which every temporal graph with n vertices has MTCS size at most $c(n)$? The important core of this question is whether for all temporal graphs there exists a sparse connectivity certificate, since that would greatly improve the worst-case space needed to store connectivity information about a temporal graph.

As we will see, the answer to the previous question is negative, even for simple temporal graphs with linear lifetime. In fact, we prove that $c(n) = \Theta(n^2)$. First we will prove the easy $O(n^2)$ upper bound that follows by taking n time-respecting arborescences, each rooted at a different vertex and results in a temporally connected graph. Then, we will present the simple $\Omega(n \log n)$ lower bound of [Kemp02] by a labeling of the hypercube, and finally our more involved $\Omega(n^2)$ lower bound, that settles the worst case MTCS size. As a side note, if we consider connectivity by *strict* temporal paths then it is trivial to show that $c(n) = \Omega(n^2)$, but it does not imply anything for the non-strict setting.

The above claims mean that we cannot hope to have better than quadratic size temporal connectivity certificates for all temporal graphs. This underlines a vast difference between connectivity certificates in temporal graphs and usual (undirected and directed) graphs: Note that in undirected graphs $n - 1$ edges are enough and in directed graphs $2(n - 1)$ edges are enough for the certificate.

In Section 3.1 we prove a lemma that states that the optimal solution to r-MTC can always be considered to be a simple temporal tree.

In Section 3.2 we show that $c(n) = O(\min\{n^2, nL\})$, i.e that the Minimum Temporal Connectivity Certificate of any temporal graph with n vertices has $O(n^2)$ and $O(nL)$ temporal edges. Note that this upper bound works for all (not only simple) temporal graphs.

In Section 3.4 we first show a simple construction that shows $c(n) = \Omega(n \log n)$ and then build a significantly stronger one that shows $c(n) = \Omega(n^2)$ and $O(n)$ lifetime. Specifically, we construct a family of simple temporal graphs with $3n$ vertices and roughly $n(n+9)/2$ edges which are almost minimally temporally connected, in the sense that the removal of any subset of $5n$ edges results in a disconnected temporal graph [1] (Theorem 3.4.2). Hence, we show that $c(n) = \Theta(n^2)$ (i.e., there are graphs with dense minimum temporal connectivity certificates), thus resolving the open question of [Kemp02]. Our construction is essentially best possible with respect to the number of edges as well as the lifetime, and can be easily extended to connectivity by strict temporal paths (with distinct time labels on the edges). An interesting feature of our construction (and an indication of its tightness) is that slightly increasing the time label of a single temporal edge results in a temporal connectivity certificate with $O(n)$ edges!

3.1 The Structure of Solutions to Single-Source Temporal Connectivity

We will now show an essential result concerning the structure of solutions to *Single-Source* Temporal Connectivity (solution to r-MTC). The following lemma is implicit in [Kemp02, Section 6] and shows that any feasible solution to r-MTC can be transformed to a simple temporal tree without increasing its total weight. Thus, we can always assume that the optimal solution to r-MTC is a simple temporal tree.

***Lemma* 3.1.1.** *Given a feasible solution $\mathcal{T}'$ for r-MTC, we can obtain a feasible solution $\mathcal{T}$ such that (i) $\mathcal{T}$ is a simple temporal graph, (ii) the total weight of $\mathcal{T}$ does not exceed the total weight of $\mathcal{T}'$, and (iii) the set of edges in $\mathcal{T}$ form a tree in the underlying graph.*

Proof. Suppose that the proposed transformation is not possible. Then, let $\mathcal{G}$ be a temporally r-connected subgraph of $\mathcal{T}'$ with the minimum number of temporal edges. Now, for each vertex u of $\mathcal{G}$, let t_u be the minimum time moment, at which it is possible to reach u starting from the source r. In other words, t_u is the minimum integer t such that there is a temporal path from r to u ending at time t.

First, we will prove that for all vertices $u \in V(\mathcal{G})$, there exists a temporal path from r to u such that for all temporal edges (e, t) of the path, where e goes from x to y, we have that $t = t_y$. We will apply induction on time labels. Now, define $S_t = \{u \in V(\mathcal{G}) \mid t_u \leq t\}$. The inductive hypothesis is that for all vertices $u \in S_t$ the statement holds. For S_0 this is trivially true. Now, suppose that it holds for S_{t-1}. Let $C_1, C_2, \ldots, C_k$ be the connected components of the subgraph of G induced by vertices u with $t_u = t$. Note that for all $i \in [k]$ there should be a temporal edge (e, t), with $e \equiv \{x, y\}$ and $x \in S_{t-1}$, $y \in C_i$, or otherwise t_u for $u \in C_i$ would have to be more than t. Now, since C_i is connected using only edges with label t, for any vertex $u \in C_i$, there is a temporal path from x to u using only temporal edges with time label t. Therefore, the statement is true for u and the induction is complete. So by definition of $\mathcal{G}$, for all temporal edges (e, t) with $e \equiv \{x, y\}$, we have that either $t_x \leq t_y = t$, or $t_y \leq t_x = t$, or otherwise (e, t) is redundant.

Obviously, the underlying graph of $\mathcal{G}$ has at least one cycle. Now, fix any cycle C which consists of the vertices $u_1, u_2, \ldots, u_k$ in this order. Let $t = \max\{t_{u_i} \mid i \in [k]\}$ and wlog $u_1, u_2, \ldots, u_m$ be a maximal path of C such that $u_1 = u_2 = \cdots = u_m = t$. Obviously, $u_{m+1}, u_k < t$, and so by the previous claim the edges $e_1 \equiv \{u_m, u_{m+1}\}$, $e_2 \equiv \{u_1, u_k\}$ both have time label t. By definition, both

[1] Based on Theorem 3.4.2, we can easily obtain a family of minimally connected temporal graphs with $\Omega(n^2)$ edges (e.g., we remove temporal edges from the graph, as long as connectivity is preserved). For simplicity and clarity, we avoid presenting a tight (but more complicated) construction of dense minimally connected temporal graphs, and stick to almost minimal graphs in the proof of Theorem 3.4.2.

of these temporal edges are essential. This means that there is some vertex u such that all temporal paths from r to u go through (e_1, t). This temporal path has the form $r, \ldots, u_{m+1}, (e_1, t), u_m, \ldots, u$. However, note the existence of the path $r, \ldots, u_k, (e_2, t), u_1, u_2, \ldots, u_m, \ldots, u$, which contradicts the hypothesis that (e_1, t) is essential. Also, in the case that $e_1 \equiv e_2$ the same holds, since we can substitute any temporal path $r, \ldots, u_1, (e_1, t), u_k, \ldots, u$ with the temporal path $r, \ldots, u_1, u_2, \ldots, u_k, \ldots, u$. Therefore G has no cycles and $\mathcal{G}$ is a temporal tree. □

3.2 An Upper Bound to the Worst Case MTCS Size

***Theorem* 3.2.1.** *Given a temporal graph $\mathcal{G}$ with n vertices and lifetime L, there is a Minimum Temporally Connected Subgraph with $O(\min\{n^2, nL\})$ temporal edges.*

Proof. Suppose that we have an MTCS $\mathcal{H}$ of $\mathcal{G}$ with minimum number of temporal edges. Define the temporal subgraph $\mathcal{H}'$ of $\mathcal{H}$ as the union of r-MTC solutions for each node $u \in V(\mathcal{H})$ as source. As shown in Lemma 3.1.1, each r-MTC solution can be assumed to have $n - 1$ edges and therefore we will have $O(n^2)$ temporal edges in total. Moreover, $\mathcal{H}'$ is a temporally connected subgraph of $\mathcal{H}$: For every pair of vertices u and v, there is a temporal path from u to v in the r-MTC solution with u as a source and this path also exists in $\mathcal{H}'$. Since $\mathcal{H}$ has minimum number of temporal edges, $\mathcal{H}' \equiv \mathcal{H}$ and so $\mathcal{H}$ will also have $O(n^2)$ temporal edges.

Now, let $H_1, \ldots, H_L$ denote the sequence of usual graphs that correspond to $\mathcal{H}$. If for some i, H_i contains a cycle, then obviously $\mathcal{H}$ does not have the minimum number of temporal edges, since deleting any edge of that cycle does not break temporal connectivity. Therefore, $\mathcal{H}$ has $O(nL)$ temporal edges. □

3.3 A Simple Lower Bound to the Worst Case MTCS Size

As a warm-up, we present the lower bound of [Kemp02], which establishes an $\Theta(n \log n)$ lower bound on the MTCS size.

***Theorem* 3.3.1.** *There exists an infinite family of temporal graphs, the MTCS size of each of which is $n\frac{\log_2 n}{2}$.*

Proof. Let $\mathcal{G}$ be a temporal graph with n vertices, where n is a power of two. Name vertices with unique binary strings of length $\log_2 n$ (Say b_u is the string corresponding to vertex u). Add a edge between two vertices u and v if the Hamming distance of b_u and b_v is exactly one, and use the index of the bit position that b_u and b_v differ as the time label on the edge (1-indexed).

$\mathcal{G}$ has $n\frac{\log_2 n}{2}$ (temporal) edges (its underlying graph is the hypercube) and is temporally connected: A temporal path from u to v, where c_u, c_v differ in bit positions $i_1 < i_2 < \cdots < i_k$, is one that visits edges with labels $i_1, i_2, \ldots, i_k$, in this order.

So what is left is to prove that $\mathcal{G}$ is minimally temporally connected. Suppose that an edge with label i and endpoints u, v is deleted. We will show that now there is no temporal path between u and v, so the graph is no longer temporally connected. b_u and b_v differ precisely at the i-th bit. By definition, a temporal path from u to v corresponds to a change of bits (starting from c_u), where bits are changed by increasing index: One cannot change bit i, then bit j, and then again bit i because of time-respect. Suppose b is the bit string that corresponds to the vertex immediately before fixing the i-th bit. Since the edge between u and v does not exist, b differs from b_u (and b_v) in at least one bit with index $< i$. After fixing the i-th bit, one can no longer fix bits with indices less than i. Therefore, one cannot hope to reach v. □

3.4 A Tight Lower Bound to the Worst Case MTCS Size

In this section we will concern ourselves with the construction of a temporal graph family and bound the size of its Minimum Temporally Connected Subgraphs, ultimately showing that the size is quadratic for this family. We will work on simple temporal graphs, as none of the results require multiple labels per edge.

In Section 3.4.1 we prove an essential lemma for the construction that concerns packing paths with linear length in a linear number of nodes. Using this, we introduce the main claims and intuition in Section 3.4.2. Detailed proof of individual claims is given in the remaining sections.

3.4.1 Partitioning a Complete Graph into Hamiltonian Paths

We first show that for any even $n \geq 2$, the edges of the complete graph K_n can be partitioned into $n/2$ Hamiltonian paths, each visiting the vertices of K_n in a different order.

***Lemma* 3.4.1.** *For any even $n \geq 2$, the edges of the complete graph K_n can be partitioned into $n/2$ Hamiltonian paths.*

Proof. Let $a_0, a_1, \dots, a_{n-1}$ be the vertices[2] of K_n. For every $i \in \{0, 1, ..., (n-2)/2\}$, we consider the Hamiltonian path $p_i = (a_i, a_{i-1}, a_{i+1}, a_{i-2}, a_{i+2}, \dots, a_{i-n/2+1}, a_{i-n/2-1}, a_{i-n/2})$, where all indices are taken modulo n.

We claim that for any $j \neq i$, the paths p_j and p_i do not have any edges in common. To this end, we observe that for any $k \in [n-1]$, the absolute difference between the indices of the k-th vertex and $(k+1)$-th vertex in any such path is k, for $1 \leq k \leq \frac{n}{2}$, and $n-k$, for $\frac{n}{2}+1 \leq k \leq n-1$. Therefore, if any two paths p_i and p_j with $i \neq j$ shared the same edge (a_x, a_y) with $|x-y| = k$, then it would either be in the same position in both paths (both in position k or both in position $n-k$), or in position k in one path and in position $n-k$ in the other path. In the former case, p_i and p_j would have been identical, which is a contradiction since each p_i starts at a different vertex. In the latter case p_j would be p_i reversed, which is again a contradiction since the first vertex of p_j would have to be $a_{i-n/2} \equiv a_{i+n/2} \equiv a_j$, but $i + n/2 = j \leq n/2$.

Using this observation, we conclude that for any $j \neq i$, the edge sets of the paths p_j and p_i are disjoint. □

3.4.2 The Construction

It is now time to describe the final construction in detail.

***Theorem* 3.4.2.** *For any even $n \geq 2$, there is a simple connected temporal graph with $3n$ vertices, $n(n+9)/2-3$ edges and lifetime at most $7n/2$, so that the removal of any subset of $5n$ edges results in a disconnected temporal graph.*

Proof sketch. For any even n, we construct a simple connected temporal graph $\mathcal{G}$ with $\Theta(n)$ vertices and $\Theta(n^2)$ edges so that virtually any edge is essential for temporal connectivity.

The Construction. For any even n, $\mathcal{G}$ consists of $3n$ vertices partitioned into three sets $A = \{a_1, \dots, a_n\}$, $H = \{h_1, \dots, h_n\}$ and $M = \{m_1, \dots, m_n\}$, with n vertices each.

The underlying graph $G[A]$ is the complete graph K_n and comprises the *dense* part of the construction with $\Theta(n^2)$ edges. The edges of $G[A]$ are partitioned into $n/2$ edge-disjoint paths $p_1, \dots, p_{n/2}$. Each path p_i has length $n-1$ and spans all vertices in A (see Figure 3.1 and Lemma 3.4.1). All edges of each path p_i have time label i.

[2] For convenience, we index the vertices of K_n from 0 to $n-1$, instead of indexing them from 1 to n, as in the proof of Theorem 3.4.2. After partitioning the vertices into $n/2$ Hamiltonian paths, we can increase all indices by one and become consistent with the notation in the construction of Theorem 3.4.2.

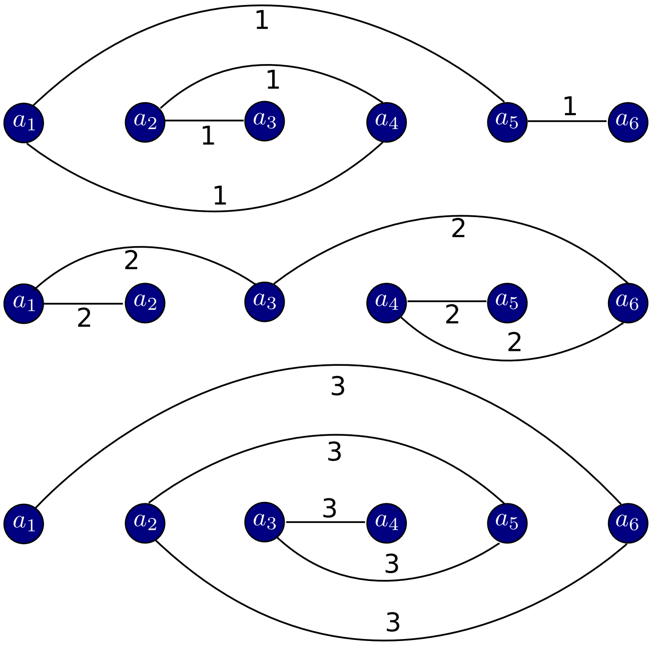


Figure 3.1: Partition of the complete graph with 6 vertices into 3 Hamiltonian paths.

The vertices of H comprise the *intermediate* part of the construction. There are no edges with both endpoints in H. For every $i \in [n/2]$, one endpoint of the path p_i is connected to h_{2i-1} and the other endpoint is connected to h_{2i}. Both edges have time label i.

The vertices of M form the *interconnecting* part of the construction. For each $i \in [n/2]$, we refer to m_{2i-1} (resp. m_{2i}) as the *entry vertex* (resp. the *exit vertex*) for the vertices h_{2i-1} and h_{2i}. There are two edges connecting m_{2i-1} to h_{2i-1} and h_{2i} with labels $n/2+2i-1$ and $n/2+2i$, respectively, and two edges connecting m_{2i} to h_{2i-1} and h_{2i} with labels $(n/2+2i-1)\epsilon$ and $(n/2+2i)\epsilon$, respectively, for some fixed $\epsilon \in (0, 1/(4n))$. We also connect the vertices of M to each other. For every $i \in [n/2-2]$, there are edges connecting m_{2i-1} to m_{2i+2} and to m_n, and a single edge connecting m_{n-3} to m_n. To allocate time labels to these edges, we order them in decreasing order of their endpoint with higher index, breaking ties by ordering them in increasing order of their endpoint with lower index, i.e., the order is $\{m_1, m_n\}$, $\{m_3, m_n\}, \dots, \{m_{n-3}, m_n\}$, $\{m_{n-5}, m_{n-2}\}$, $\{m_{n-7}, m_{n-4}\}, \dots, \{m_1, m_4\}$. The time label of the k-th edge in this order is $1-(k-1)\epsilon$. Finally, for every $i \in [n/2]$, there are an edge with time label ϵ connecting the vertex m_{2i-1} to the vertex a_{2i-1} in A and an edge with time label $n+1$ connecting the vertex m_{2i} to the vertex a_{2i} in A (see also Figure 3.2).

The total number of edges is $n(n+9)/2-3$, the number of different time labels is at most $7n/2$, and each edge has a single label (see also Section 3.4.3).

Intuition and Main Claims. The construction is based on the collection $p_1, \dots, p_{n/2}$ of $n/2$ edge-disjoint paths, where all edges in each path p_i have label i. Extending each path p_i to vertices h_{2i-1} and h_{2i}, we get a path that connects h_{2i} to h_{2i-1} (and vice versa) and to all vertices in A at time i. Moreover, different time labels make these paths essentially independent of each other, in the sense that if a temporal walk begins and ends at time i, it can use only edges with label i (i.e., only edges of this path) to connect h_{2i} to h_{2i-1}. In Section 3.4.5, we formalize this intuition and show that the unique temporal path from h_{2i} to h_{2i-1} is through path p_i. Therefore, all edges of $G[A]$ must be present in any temporally connected spanning subgraph of $\mathcal{G}$. To achieve a dense underlying graph $G[A]$, we observe that the collection of $n/2$ edge-disjoint paths can be defined so that they go through the same n vertices, in a different order each (see Figure 3.1 and Lemma 3.4.1). This describes the main intuition behind our construction and explains how the dense and the intermediate parts work. The only problem now is that H-vertices with high indices, e.g., h_n, cannot reach H-vertices with

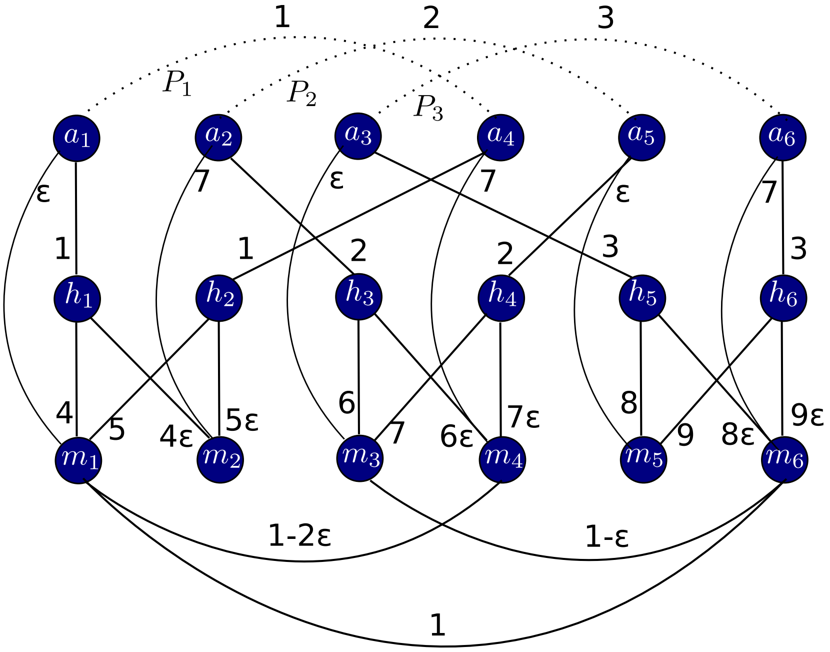


Figure 3.2: Putting the 3 parts together we obtain the final construction for $n = 6$.

low indices, e.g., h_1. The vertices in the interconnecting part M serve to carefully connect each h_j to each h_i, with $j > i + 1$, without destroying the property that the only temporal path from h_{2i} to h_{2i-1} is through path p_i.

For every vertex pair $h_{2i-1}, h_{2i} \in H$, we introduce a vertex pair $m_{2i-1}, m_{2i} \in M$. As an entry vertex, m_{2i-1} is connected to h_{2i-1} and h_{2i} with "large" labels (larger than $n/2$). Hence, starting from the rest of $\mathcal{G}$, we can reach h_{2i-1} and h_{2i} through m_{2i-1}, but we cannot continue to the edges of p_i (with label $i \leq n/2$). As an exit vertex, m_{2i} is connected to h_{2i-1} and h_{2i} with "very small" labels (at most $1/4$). Thus, starting from m_{2i}, we can reach first h_{2i-1} and h_{2i}, and then all vertices in A and any vertex h_j with index $j > 2i$. Moreover, to avoid creating a temporal path from h_{2i} to h_{2i-1}, the label of the edge $\{h_{2i}, m_{2i-1}\}$ (resp. $\{h_{2i}, m_{2i}\}$) is larger than the label of the edge $\{h_{2i-1}, m_{2i-1}\}$ (resp. $\{h_{2i-1}, m_{2i}\}$).

It remains now to connect the M-vertices to each other, without creating any alternative temporal paths from h_{2i} to h_{2i-1}, for any $i \in [n/2]$. For each $i \in [n/2]$, the edges between M-vertices should create temporal paths from m_{2i-1} and m_{2i} to any vertex m_j with index $j < 2i-1$. On the other hand, they should not create any temporal $m_{2i} - m_{2i-1}$ paths, since then we would have a new temporal $h_{2i} - h_{2i-1}$ path. We introduce roughly n edges between M-vertices and carefully select their "small" labels in $[3/4, 1]$. Furthermore, to achieve temporal connectivity between all vertex pairs, we introduce an edge $\{m_{2i-1}, a_{2i-1}\}$ with the minimum time label ϵ and an edge $\{m_{2i}, a_{2i}\}$ with label $n+1$, for each $i \in [n/2]$.

To complete the proof, in Section 3.4.4, we consider all possible types of ordered vertex pairs and show that the temporal graph $\mathcal{G}$ is indeed connected. Moreover, any subset of at least $5n$ edges includes some edges of $G[A]$. In Section 3.4.5, we show that the removal of any edge from $G[A]$ with label i destroys the unique temporal path from h_{2i} to h_{2i-1}. □

We should highlight that increasing the label of edge $\{a_1, m_1\}$ from ϵ to 1, in the graph of Theorem 3.4.2, results in a temporal graph that admits a connectivity certificate of size $\Theta(n)$ (see Lemma 3.4.3). Moreover, it is not hard to modify the construction of Theorem 3.4.2 so that all time labels of the edges are distinct, the temporal graph $\mathcal{G}$ is connected by strict temporal paths, and the removal of any subset of $5n$ edges results in a disconnected temporal graph. Therefore, the quadratic lower bound of Theorem 3.4.2 also applies to connectivity by strict temporal paths and improves on the lower bound of $\Omega(n \log n)$ in [Akri15, Theorem 3].

3.4.3 The Number of Edges and the Lifetime of $\mathcal{G}$

The total number of edges in G is $n(n+9)/2-3$. Specifically, we have $n(n-1)/2$ edges between vertices in A, $2n$ edges connecting vertices in $H \cup M$ to vertices in A, $2n$ edges connecting vertices in M to vertices in H, and $2(n/2-2)+1$ edges between M-vertices.

The total number of different time labels is at most $7n/2$ and each edge has a single label (so the temporal graph is simple). Specifically, we use $n/2$ labels for the edges in $G[A]$ and for the edges connecting vertices in H to vertices in A, $2n$ labels for the edges connecting vertices in M to vertices in H, 2 labels for the edges connecting vertices in M to vertices in A, and at most $n-2$ labels for the edges in $G[M]$. We also note that the use of non-integral labels in our construction is just for simplicity and without loss of generality.

3.4.4 The Temporal Graph $\mathcal{G}$ is Connected

We proceed to show that the temporal graph $\mathcal{G}$ constructed in the proof of Theorem 3.4.2 is indeed connected. To this end, we consider all possible types of ordered vertex pairs (u, v) and show that $\mathcal{G}$ contains a temporal path from u to v. We need to distinguish between several different cases:

$u, v \in A$: We move along the path p_1 of $\mathcal{G}[A]$, where all edges have label 1.

$u \in A$, $v = h_i \in H$: We move from u to a_i using the path p_1, where all edges have label 1. Then, we follow the edge with label $\lfloor i/2 \rfloor$ from a_i to h_i.

$u \in A$, $v = m_{2i-1} \in M$: We first move to h_{2i-1}, as in the previous case, and then follow the edge with label $n/2+2i-1$ from h_{2i-1} to m_{2i-1}.

$u \in A$, $v = m_{2i} \in M$: We move from u to a_{2i} using the path p_1, where all edges have label 1. Then, we follow the edge with label $n+1$ from a_{2i} to m_{2i}.

$u = h_i \in H$, $v \in A$: We move from h_i to a_i using the edge with label $\lceil i/2 \rceil$ and proceed to v along the path $p_{n/2}$, where all edges have label $n/2$.

$u = h_i \in H$, $v = h_j \in H$ and $\lceil i/2 \rceil \leq \lceil j/2 \rceil$: We first move from h_i to a_i, using the edge with label $\lceil i/2 \rceil$, and then on the path $p_{\lceil i/2 \rceil}$ to a_j. Then, we move to h_j using the edge with label $\lceil j/2 \rceil$.

$u = h_i \in H$, $v = h_j \in H$ and $\lceil i/2 \rceil > \lceil j/2 \rceil$: We let $\ell_i = 2\lceil i/2 \rceil$ and $\ell_j = 2\lceil j/2 \rceil$. We first move from h_i to m_{ℓ_i} using the edge with "very small" label $(n/2+i)\epsilon$. We then move to m_{ℓ_i-3}, to m_n and to m_{ℓ_j-1} using the edges between M-vertices with "small" labels. Finally, we move from m_{ℓ_j-1} to h_j using the edge with "large" label $n/2+j$. We note that the labels of the temporal edges between M-vertices are increasing due to the particular ordering that we use.

$u = h_i \in H$, $v = m_{2j-1} \in M$: If $\lceil i/2 \rceil \leq j$, we move from h_i to h_{2j} as in the corresponding case, using edges with labels at most $n/2$. Otherwise, we move from h_i to h_{2j} as in the previous case, using edges with labels at most $n/2+2j$. Finally, we move from h_{2j} to m_{2j-1} using the edge with label $n/2+2j$.

$u = h_i \in H$, $v = m_{2j} \in M$: We move from h_i to a_{2j} as in the corresponding case, using labels at most $n/2$, and from a_{2j} to m_{2j} using the edge with label $n+1$.

$u = m_{2i-1} \in M$ to any other vertex v: We move from m_{2i-1} to a_{2i-1} using the edge with ϵ and from a_{2i-1} to any other vertex, as in the first four cases, since ϵ is the minimum time label.

$u = m_{2i} \in M$, $v \in A$: We move from m_{2i} to h_{2i}, using the edge with label $(n/2+2i)\epsilon$, next to the second endpoint of path p_i, using the edge with label i, and next to any vertex in A, along the path p_i, where all edges have label i.

$u = m_{2i} \in M$, $v = m_{2j} \in M$: We move from m_{2i} to a_{2j} as in the previous case, using labels at most j, and next to m_{2j}, using the edge with label $n+1$.

$u = m_{2i} \in M$, $v = h_j \in H$: If $i \leq \lceil j/2 \rceil$, we move from m_{2i} to h_{2i} using the edge with label $(n/2+2i)\epsilon$, next to the second endpoint of path p_i, using the edge with label i, next to vertex a_j along the path p_i, where where all edges have label i, and finally to h_j using the edge with label $\lceil j/2 \rceil$. Otherwise, we move from m_{2i} to m_{2i-3}, to m_n and to $m_{2\lceil j/2 \rceil -1}$, using the edges between M-vertices, with "small" labels, and finally from $m_{2\lceil j/2 \rceil -1}$ to h_j, using the edge with label $n/2+j$.

$u = m_{2i} \in M$, $v = m_{2j-1} \in M$: If $i \leq j$, we move to h_{2j-1}, as in the first walk of the previous case, using labels at most j, and next to m_{2j-1} using the edge with label $n/2+2j-1$. Otherwise, we follow the second walk of the previous case up to m_{2j-1}. □

3.4.5 Removing a Linear Number of Edges Disconnects $\mathcal{G}$

Since the total number of edges in $\mathcal{G}$ is $n(n+9)/2-3$ and the dense part $\mathcal{G}[A]$ includes $n(n-1)/2$ of them, removing any subset of at least $5n$ edges results in the removal of at least one edge e between A-vertices. Let us assume that e has label i. We next show that the removal of any edge with label i results in a temporal graph $\mathcal{G}'$ with no temporal path from h_{2i} to h_{2i-1}. Hence, the removal of any set of at least $5n$ edges results in a disconnected temporal graph.

To reach a contradiction, let us assume that there is a temporal path p in $\mathcal{G}'$ from h_{2i} to h_{2i-1}. Since an edge of label i has been removed, the path p_i is not present in $\mathcal{G}'$. Therefore, p should use some edges with labels other than i. We distinguish between two cases:

The first edge of p has label i and the last edge has label larger than i. The only edge incident to h_{2i-1} that can be used as the last edge of p has label $n/2+2i-1$. We note that any edges with labels at most 1 cannot be used by p. The same is true for the edges with label $n+1$, which connect vertices m_{2j} and vertices a_{2j}, for every $j \in [n/2]$, since all other edges incident to vertices m_{2j} have labels at most 1 and all other edges incident to a_{2j} have labels at most $n/2$. If we ignore all edges with labels at most 1 and $n+1$, we have no edges between A-vertices and M-vertices and no edges between M-vertices. Therefore, p has to visit vertex a_{2i} first and then to move through the vertices of A. At some point, a vertex $m_j \in M$ must be visited, before p finally reaches h_{2i-1}. But this leads to a contradiction, since there are no usable edges between A-vertices and M-vertices. Hence, no such path p from h_{2i} to h_{2i-1} exists.

Otherwise. The only alternative for p is to begin with label $(n/2+2i)\epsilon$ and to end with either label i or label $n/2+2i-1$. Then, the first edge brings p to m_{2i}. At that point, any edge with the minimum label ϵ cannot be used by p. As before, the same is true for the edges with label $n+1$, which connect vertices m_{2j} and vertices a_{2j}, for every $j \in [n/2]$, since all other edges incident to vertices m_{2j} have labels at most 1 and all other edges incident to a_{2j} have labels at most $n/2$. So, we ignore all edges with labels either ε or $n+1$. Then, p can only visit vertices m_{2j-1}, with $j < i$, or the vertex m_n (through some edge with a 'small" label in $[3/4, 1]$). From m_n, p cannot proceed any further, since all edges that connect m_n to some vertex in H have "very small" labels in $(0, 1/4]$. Otherwise, if the last M-vertex visited by p is m_{2j-1}, for some $j < i$, the next edge of p must have either label $n/2+2j-1$ (vertex h_{2j-1}) or label $n/2+2j$ (vertex h_{2j}), and then, p cannot proceed any further. Again, we reach a contradiction. Hence, no such path p from h_{2i} to h_{2i-1} exists. □

3.4.6 Tightness of the Construction: Increasing a Single Label

We next show that if in the temporal graph $\mathcal{G}$ constructed in Theorem 3.4.2, we increase the label of edge $\{a_1, m_1\}$ from ϵ to 1, the resulting temporal graph $\mathcal{G}'$ admits a connectivity certificate with $\Theta(n)$ edges.

***Lemma* 3.4.3.** *Let $\mathcal{G}$ be the temporal network constructed in Theorem 3.4.2 and let $\mathcal{G}'$ be the temporal network obtained from $\mathcal{G}$ if we increase the label of edge $\{a_1, m_1\}$ from ϵ to 1. Then, $\mathcal{G}'$ admits a temporal connectivity certificate with $\Theta(n)$ edges.*

Proof. We show that the temporal graph $\mathcal{G}'$ remains connected if we remove the edges of all paths p_i with $i > 1$. Clearly, this results in a temporal graph with $\Theta(n)$ edges. The proof shows how to modify all temporal $u - v$ paths that are described in Section 3.4.4 and use some edges of a path p_i, with $i > 1$. Again, we need to consider several cases for the initial vertex u and the final vertex v:

$u = h_i \in H$, $v \in A$: We move from h_i to m_{2i}, to m_{2i-3}, to m_n, and to m_1, using only edges with labels at most 1. Using the edge $\{m_1, a_1\}$, with label 1, we can visit any vertex in A through the path p_1, where all edges have label 1.

$u = h_i \in H$, $v = h_j \in H$: As in the first case, we move from h_i to the endpoint of the path $p_{\lceil j/2 \rceil}$ which is connected to h_j. This part uses labels at most 1. For the last step, we use the edge from the endpoint of $p_{\lceil j/2 \rceil}$ to h_j, which has label $\lceil j/2 \rceil \geq 1$.

$u = m_{2i} \in M$, $v \in A$: Similarly to the first case, we move from m_{2i} to m_{2i-3}, to m_n, and to m_1, using only edges with labels at most 1. Using the edge $\{m_1, a_1\}$, with label 1, we can visit any vertex in A through the path p_1, where all edges have label 1.

$u = m_{2i} \in M$, $v = h_j \in H$: As in the previous case, we from m_{2i} to the endpoint of the path $p_{\lceil j/2 \rceil}$ which is connected to h_j. This part uses labels at most 1. For the last step, we use the edge from the endpoint of $p_{\lceil j/2 \rceil}$ to h_j, which has label $\lceil j/2 \rceil \geq 1$.

□

Chapter 4

The Approximability of Single-Source Temporal Connectivity

In Section 4, we show that the polynomial-time approximability of Single-Source Temporal Connectivity (r-MTC) is closely related to the approximability of the classical Directed Steiner Tree problem. Using a transformation from Directed Steiner Tree to r-MTC (Theorem 4.1.1) and [Halp03, Theorem 1.2], we show that r-MTC cannot be approximated within a ratio of $O(\log^{2-\varepsilon} n)$, for any constant $\varepsilon > 0$, unless NP $\subseteq$ ZTIME($n^{\text{poly}\log n}$). Our transformation also implies that any $o(n^{\varepsilon})$-approximation for r-MTC would improve the best known approximation ratio of Directed Steiner Tree. On the positive side, using a transformation from r-MTC to Directed Steiner Tree and the algorithm of [Char99], we obtain a polynomial-time $O(n^{\varepsilon})$-approximation, for any constant $\varepsilon > 0$, and a quasipolynomial-time $O(\log^3 n)$-approximation for r-MTC (Theorem 4.2.1). We also show that r-MTC can be solved in polynomial time if the underlying graph has bounded treewidth (Theorem 4.3.5).

To understand the approximability of r-MTC, we use reductions from and to Directed Steiner Tree, so we will define the problem formally.

Definition 4.0.4 (Directed Steiner Tree)**.** Given a directed edge-weighted graph $G(V, E)$ with n vertices, a source $r \in V$ and a set of k terminals $S \subseteq V$, the Directed Steiner Tree (DST) problem asks for a subgraph of G that includes a directed path from r to any vertex in S and has minimum total weight. The best known algorithm for DST is due to Charikar et al. [Char99] and achieves an approximation ratio of $O(k^{\varepsilon})$, for any constant $\varepsilon > 0$, in polynomial time, and of $O(\log^3 k)$ in quasipolynomial time. On the negative side, [Halp03, Theorem 1.2] shows that DST cannot be approximated within a factor $O(\log^{2-\varepsilon} n)$, for any constant $\varepsilon > 0$, unless NP $\subseteq$ ZTIME($n^{\text{poly}\log n}$).

4.1 A Lower Bound on the Approximability of r-MTC

We start with an approximation-preserving transformation from DST to r-MTC. The intuition is that we can use strict temporal paths to "simulate" the directed edges of DST.

***Theorem* 4.1.1.** *Any polynomial-time $\rho(n)$-approximation algorithm for* r-MTC *on simple temporal graphs implies a polynomial-time $\rho(n^2)$-approximation algorithm for* DST.

Proof. We present an approximation-preserving transformation from the DST to r-MTC. Given an instance $I = (G(V, E, w), S, r)$ of DST with $|V| = n$, we construct a temporal graph $\mathcal{G}'$ with n^2 vertices so that (i) any Steiner tree connecting r to S in G can be mapped to an r-connected subgraph of $\mathcal{G}'$ with no larger cost; and (ii) given any r-connected subgraph of $\mathcal{G}'$, we can efficiently compute a feasible Steiner tree for I with no larger cost.

Each vertex $u \in V$ corresponds to a vertex u in the temporal network $\mathcal{G}'$ of I'. For every directed edge $e = (u, v)$ of G, we create $n - 1$ strict temporal $u - v$ paths of length 2. Specifically, for every vertex $u \in V$, $\mathcal{G}'$ contains auxiliary vertices z_i^u, for all $i \in [n-1]$, and temporal edges $\{u, z_i^u\}$ with time label i and weight 0. For every edge $e = (u, v) \in E$, $\mathcal{G}'$ contains temporal edges $\{z_i^u, v\}$ with time label $i+1$ and weight $w(e)$, for all $i \in [n-1]$. Let $Z = \{z_i^u\}_{u \in V, i \in [n-1]}$ be the set of all auxiliary vertices. For every vertex $x \in Z \cup (V \setminus S)$, $x \neq r$, $\mathcal{G}'$ contains a temporal edge $\{r, x\}$ with time label

$n + 1$ and weight 0. These edges ensure that r is connected to all non-terminal and auxiliary vertices at no additional cost.

The temporal graph $\mathcal{G}'$ has $\Theta(n^2)$ vertices. What is left is to prove claims (i) and (ii), which follow from the construction of $\mathcal{G}'$.

We have to prove that (i) any feasible solution T of the DST instance $I = (G(V, E, w), S, r)$ can be mapped to a temporally r-connected subgraph T' of $\mathcal{G}'$ with no larger cost; and (ii) given a temporally r-connected subgraph T' of $\mathcal{G}'$, we can efficiently compute a directed Steiner tree T which connects r to all terminal vertices in S and has no larger cost than T'.

We first show (i). Let T be any feasible solution to DST. For every vertex u, let $d(u)$ be the number of edges on the $r - u$ path in T (we consider the $r - u$ path with the smallest number of edges, if there are many). To transform T into an r-connected subgraph T' of $\mathcal{G}'$, we first include in T' all 0-cost edges incident to r. Next, for each directed edge (u, v) in T, we include in T' the temporal path $u - z^u_{d(u)} - v$, with time labels $d(u)$ and $d(u) + 1$ and weight $w(e)$. Clearly, the total weight of T' does not exceed the total weight of T. Moreover, there is a temporal path from r to any vertex u of $\mathcal{G}'$. If $u \in Z \cup (V \setminus S)$, we use the direct edge $\{r, u\}$ with time label $n + 1$. If $u \in S$, there is a directed $r - u$ path in T. For the i-th edge of this path, $i \leq n - 1$, we have added a temporal path of length 2 with labels i and $i + 1$. The union of these paths defines a temporal path from r to u.

We next show (ii). Let T' be any r-connected subgraph of $\mathcal{G}'$. Without increasing the cost of T', we can assume that any edge $\{z^u_i, v\}$ with positive weight $w(u, v)$ in T' belongs to a temporal $u - v$ path of length 2, where $u, v \in V$. To transform T' into a feasible solution T of the DST instance I, we include in T the edge $(u, v) \in E$ of any temporal path $u - z^u_i - v$ in T'. Clearly, the total weight of T does not exceed the total weight of T'. To show that T is a feasible solution, we observe that all edges incident to a terminal vertex $u \in S$ have time labels at most n. Therefore, the maximum label in any temporal $r - u$ path is at most n. Hence, for any $u \in S$, T' includes a temporal $r - u$ path consisting of temporal paths of length 2, i.e., of paths of the form $v_k - z^{v_k}_i - v_{k+1}$. Each of these paths becomes a directed edge (v_k, v_{k+1}) in T. These directed edges together form a directed $r - u$ path in T. □

Directed Steiner Tree cannot be approximated within a ratio of $O(\log^{2-\varepsilon} n)$, for any constant $\varepsilon > 0$, unless NP $\subseteq$ ZTIME($n^{\text{poly}\log n}$) [Halp03, Theorem 1.2]. Theorem 4.1.1 implies that this inapproximability result carries over to r-MTC. Moreover, any polynomial-time $o(n^\varepsilon)$-approximation algorithm for r-MTC would immediately improve the best known approximation ratio of the notoriously difficult DST problem.

4.2 An Approximation Algorithm for r-MTC

In this Section we present an approximation-preserving reduction from r-MTC to DST, thus proving Theorem 4.2.1 (see also the more general proof of Theorem 5.1.2). Then, we can use the algorithm of [Char99] and approximate r-MTC within a ratio of $O(n^\varepsilon)$, for any constant $\varepsilon > 0$, in polynomial time, and within a ratio of $O(\log^3 n)$ in quasipolynomial time. The reduction of Theorem 4.2.1 can be easily extended to r-connectivity by strict temporal paths.

***Theorem* 4.2.1.** *Any polynomial-time $\rho(k)$-approximation algorithm for* DST *implies a polynomial-time $\rho(n)$-approximation algorithm for r-*MTC *on general temporal graphs*

Proof. Let I be an instance of r-MTC consisting of an underlying graph $G(V, E)$, a source $r \in V$, a finite set of time labels L_e for each edge e, and a cost $w(e, t)$ for any temporal edge (e, t). We show how to transform I into an instance $I' = (H, S, r')$ of DST so that (i) any feasible solution of I can be mapped to a feasible solution of I' with no larger cost; and (ii) given a feasible solution of I', we can compute a feasible solution of I with no larger cost.

The directed graph H of the DST instance contains a non-terminal vertex for each temporal edge (e, t) of $\mathcal{G}$. For every (ordered) pair of temporal edges (e_1, t_1), (e_2, t_2) of $\mathcal{G}$, such that $e_1 \neq e_2$, $t_1 \leq t_2$ (or $t_1 < t_2$, for strict r-connectivity), and e_1 and e_2 share a common endpoint, there is a directed edge

$((e_1, t_1), (e_2, t_2))$ with weight $w(e_2, t_2)$ in H. H also contains a root vertex r' corresponding to the source r. For every temporal edge (e, t) incident to r in $\mathcal{G}$, there is a directed edge $(r', (e, t))$ with weight $w(e, t)$ in H. For every vertex $u \in V \setminus \{r\}$, H contains a terminal vertex u'. The terminal set S of I' consists of all these terminal vertices u'. Moreover, for every temporal edge (e, t) incident to u in $\mathcal{G}$, H contains a directed edge $((e, t), u')$ with weight 0. The directed graph H has $O(M)$ vertices, $n - 1$ of which are terminal vertices.

We first show (i). Let T be any feasible solution of the r-MTC instance I. By Lemma 3.1.1, we can assume that T is a simple temporal graph and that its underlying graph is a tree (otherwise, we can obtain such a solution from T without increasing its total weight). Thus, T contains a unique temporal path from r to any vertex $u \in V \setminus \{r\}$. To obtain a feasible solution T' for the DST instance I', we do the following, starting from r, in a BFS order: For every pair of vertices $u, v \in V$ such that u is the immediate predecessor of v on the unique temporal $r - v$ path in T, u is reached through the temporal edge (e_1, t_1), and v is reached from u through the temporal edge (e_2, t_2), we add the directed edge $((e_1, t_1), (e_2, t_2))$ to T' (this edge exists in H, because both e_1 and e_2 have u as one of their endpoints and $t_1 \leq t_2$). For every vertex $v \in V$ reachable in T directly from r through the temporal edge (e, t), we add the directed edge $(r', (e, t))$ to T'. Moreover, we add all directed edges of weight 0 to T'. Clearly, the cost of T' does not exceed the cost of T. As for the feasibility of T', we observe that any temporal $r - u$ path in T, which consists of a temporal edge sequence $((e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k))$, with e_1 starting at r and e_k ending at u, corresponds to a directed $r' - u'$ path $(r', (e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k), u')$ in T'

We next show (ii). Let T' be any feasible solution of the DST instance I'. Thus, T' includes a directed path from r' to any vertex $u' \in S$. For each Steiner vertex (e, t) used by some path in T', we add the corresponding temporal edge to the solution T of the r-MTC instance I. Clearly, the total cost of T is no more than the total weight of T'. We also need to show that T is indeed a feasible solution for the r-MTC instance I. For every vertex $u \in V \setminus \{r\}$, there is a directed path $(r', (e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k), u')$ in T'. By the construction of I', $t_1 \leq t_2 \leq \cdots \leq t_k$ and $(e_1, e_2, \ldots, e_k)$ form an $r - u$ path in the underlying graph of $\mathcal{G}$. Therefore, since the temporal edges $(e_1, t_1), (e_1, t_2), \ldots, (e_k, t_k)$ are present in T, T includes a temporal $r - u$ path for every vertex $u \in V \setminus \{r\}$. □

4.3 A Polynomial-Time Algorithm for Graphs with Bounded Treewidth

In this section we prove that r-MTC can be solved in polynomial time, by dynamic programming, if the underlying graph has bounded treewidth. First, we will present some well-known definitions and lemmas about tree decompositions that will be needed for the algorithm of this section. We refer to vertices of the tree decomposition as *nodes* and to vertices of the original graph as *vertices*.

Definition 4.3.1 (Tree Decompositions)**.** A *tree decomposition* of a graph G is a tree T, each node of which is also equipped with a subset of $V(G)$. For the node i of T, we will denote the corresponding subset of $V(G)$ as T_i. The nodes of T are called *bags* of the tree decomposition. A tree decomposition also has to obey the following properties:

- The union of all T_i is $V(G)$
- For all edges $e \equiv (u, v) \in E(G)$, there exists a bag i such that $u, v \in T_i$
- For all vertices $u \in V(G)$, all pairs of bags i, j such that $u \in T_i$ and $u \in T_j$, and all bags k that lie on the path from i to j in T, we have that $u \in T_k$. This intuitively means that the set of bags that contain any given vertex u form a connected region in T.

The *width* of the tree decomposition is the maximum $|T_i|$ over all bags i of the tree decomposition, minus one.

Definition 4.3.2 (Treewidth)**.** The *treewidth* of a graph G is the minimum width over all tree decompositions of G.

Definition 4.3.3 (Nice Tree Decompositions)**.** A *Nice tree decomposition* of a graph G is a tree decomposition T of G, where T is rooted and binary, in which for for all bags i, i has one of the following properties:

- i is a leaf and $|T_i| = 1$ (*leaf node*)
- i has two children $c_1(i)$, $c_2(i)$, and $T_i = T_{c_1(i)} = T_{c_2(i)}$ (*join node*)
- i has one child $c_1(i)$, and $T_i \supset T_{c_1(i)}$, $|T_i| = |T_{c_1(i)}| + 1$ (*introduce node*)
- i has one child $c_1(i)$, and $T_i \subset T_{c_1(i)}$, $|T_i| = |T_{c_1(i)}| - 1$ (*forget node*)

We will use the following well-known theorem to only work on nice tree decompositions:

***Theorem* 4.3.4.** *Let T be a tree decomposition of a graph G ($n = |V(G)|$) with width k. Then, there is an algorithm that runs in time $O(nk^2)$ and outputs a nice tree decomposition of G with width k and $O(nk)$ bags.*

***Theorem* 4.3.5.** *Let $\mathcal{G}$ be a temporal graph on n vertices with lifetime L, source vertex r and treewidth at most k. Then, there is a dynamic programming algorithm which given a nice tree decomposition of G, computes an optimal solution to r-*MTC *in time $O(nk^2 3^k (L+k)^{k+1})$.*

Proof. We present an exact polynomial-time algorithm for r-MTC on temporal graphs with treewidth at most k. The algorithm is based on dynamic programming and its time complexity is $O(nk^2 3^k (L+k)^{k+1})$. To this end, we consider a temporal graph $\mathcal{G}(V, E)$ on n vertices with lifetime L, source vertex let r and treewidth k. Below, we use the standard notation and terminology employed in dynamic programming algorithms for graph theoretic problems on graphs of bounded treewidth (see e.g., [Down13, Chapter 10]).

We assume a *nice tree decomposition* of $\mathcal{G}$ with $O(nk)$ bags, where the root bag consists only of the vertex r. Also, for each bag i, consider an enumeration $v_1^i, v_2^i, \ldots, v_p^i$ of the vertices in bag i, where $p \leq k+1$. For each bag i, we define $f(i, a_1, t_1, \ldots, a_{k+1}, t_{k+1})$, where $a_j \in \{0, 1\}, t_j \in [L+k+1]$, as follows:

- Let H be the subgraph of $\mathcal{G}$ induced on the vertices of bag i and all of its descendants in the tree decomposition.
- Let T be a minimum cost temporal subgraph of H such that for every vertex v of H, there exists a vertex v_j^i with $a_j = 1$ and a temporal path from v_j^i to v in T, starting no sooner than t_j.
- Moreover, if $v = v_l^i$ for some l, then this temporal path ends at time t_l.

Note that H is always a temporal forest. Intuitively, $a_j = 1$ means that the j-th vertex of the bag is currently connected to the root, while $a_j = 0$ means that it is not connected yet, and we should connect it via some temporal path in H. Note that some of the arguments of f can be undefined, as some bags may have less than $k+1$ vertices. The cost of the optimal solution to r-MTC is then $f(1, 1, 1)$, since we start at bag 1 (which only contains the source r) and we are at r at time 1. To break the symmetry introduced by same-label components, for each set of vertices in bag i with the same time label t, we introduce an *ordering* between these vertices. This is done to avoid cyclic dependencies between parent links, which would lead to an infeasible solution. To express such an ordering $x_1, \ldots, x_k$, suppose that the vertices $v_{x_1}^i, \ldots, v_{x_k}^i$ all share the same time label t (the temporal path from r to each of them ends at time t). Then, we set $t_{x_1} = t, t_{x_2} = L + x_1, \ldots, t_{x_k} = L + x_{k-1}$. Thus, we express the required information with space $(L+k)^{k+1}$.

Below, we let $c_1(i)$ and $c_2(i)$ denote the left and right children of bag i respectively. The recurrence relation for computing f is the following:

Case 1: i is an introduce node. Suppose wlog that v_p^i is introduced. If $a_p = 0$, then there must exist a temporal path from some vertex v_j^i with $a_j = 1$ to v_p^i in T, the second-to-last vertex of the path being in bag i. This is because v_p^i exists in bag i but none of its descendants, so this is the last chance to connect it to vertices with $a_j = 1$. It is also the last chance to connect vertices with $a_j = 0$ directly to v_p^i (as its children). To this end, we try all possible vertices as parents of v_p^i, and also all possible subsets of vertices with $a_j = 0$ as the set of children of v_p^i. Of course, we adhere to the time labels of the vertices, as well as the ordering introduced above. Thus, we have that

$$\begin{aligned} f(i, a_1, t_1, \ldots, 0, t_p) = \min_{j \neq i} \Big\{ & f(c_1(i), a_1, t_1, \ldots, 1, t_p) + w_{\text{time}(p)}(v_j^i, v_p^i) \ : \\ & \text{time}(j) < \text{time}(p) \lor \\ & (\text{time}(j) = \text{time}(p) \land j \text{ is before } p \text{ in the ordering}) \Big\} \end{aligned}$$

where time(j) is the time label that the temporal path from r to v_j^i ends. (If $t_j \leq L$, then time$(j) = t_j$, otherwise time$(j) = $ time$(t_j - L)$, where $t_j - L$ is the index of the predecessor of v_j^i in the ordering of vertices with time$(k) = $ time(j)). Similarly,

$$\begin{aligned} f(i, a_1, t_1, \ldots, 1, t_p) = \min_{S} \Bigg\{ & f(c_1(i), a_1^*, t_1, \ldots, a_{p-1}^*, t_{p-1}) + \sum_{v_j^i \in S} w_{\text{time}(j)}(v_p^i, v_j^i) \ : \\ & S \subseteq \{v_j^i \in \{v_1^i, \ldots, v_{p-1}^i\} : a_j = 0 \land \\ & (\text{time}(p) < \text{time}(j) \lor \\ & (\text{time}(p) = \text{time}(j) \land p \text{ is before } j \text{ in the ordering}))\} \Bigg\} \end{aligned}$$

where $a_j^* = 0$ if $a_j = 0$ and $v_j^i \notin S$, and $a_j^* = 1$ otherwise. Therefore, the complexity of computing all values of f that fall to this case is $O(nk^2 3^k (L+k)^{k+1})$. This breaks down into $O(nk)$ for the size of the tree, $O(k)$ for iterating over vertices of S and computing time(j), $O(3^{k+1})$ for choices over vertices with $a_j = 1$ as well as vertices with $a_j = 0$ and $v_j^i \in S$, and $(L+k)^{k+1}$ for choices of t_j.

Case 2: i is a forget node. Suppose wlog that $v_{p+1}^{c_1(i)}$ is forgotten, and iterate over all possible values t_{p+1}. v_{p+1}^i is either assigned a new time label, or is inserted in the ordering between vertices with the same time label. We have that

$$\begin{aligned} f(i, a_1, t_1, \ldots, a_p, t_p) = \min_{t_{p+1} \in [L+k]} \Big\{ & f(c_1(i), a_1, t_1, \ldots, a_{j'}, L+p+1, \ldots, a_p, t_p, 0, t_{p+1}) \ : \\ & t_{p+1} \leq L \lor t_{p+1} - L = j \in [p] \Big\} \end{aligned}$$

where $v_{j'}^i$ was the successor of v_j^i (if it had one) in the ordering, which now becomes the successor of v_{p+1}^i in the case that $t_{p+1} - L = j$ (v_{p+1}^i is inserted between v_j^i and $v_{j'}^i$ in the ordering). So, in these cases, the total time complexity is $O(nk^2 2^k (L+k)^{k+1})$.

Case 3: i is a join node. Then some vertices with $a_j = 0$ will be connected in $c_1(i)$ and some others in $c_2(i)$. Suppose that $P = \{v_j^i \in \{v_1^i, \ldots, v_p^i\} : a_j = 0\}$. Therefore, trying every possible partition of the vertices with $a_j = 0$, we have that

$$\begin{aligned} f(i, a_1, t_1, \ldots, a_p, t_p) = \min_{S_1, S_2} \Big\{ & f(c_1(i), a_1^*, t_1, \ldots, a_p^*, t_p) + f(c_2(i), a_1^{**}, t_1, \ldots, a_p^{**}, t_p) \ : \\ & S_1 \cup S_2 = P \text{ and } S_1 \cap S_2 = \emptyset \Big\} \end{aligned}$$

Here we have that $a_j^* = 0$ and $a_j^{**} = 1$, if $a_j = 0$ and $v_j^i \in S_2$, that $a_j^* = 1$ and $a_j^{**} = 0$, if $a_j = 0$ and $v_j^i \in S_1$, and that $a_j^* = a_j^{**} = 1$, if $a_j = 1$.

Here, the total time complexity is $O(nk^2 3^k (L + k)^{k+1})$. We observe that the vertex order is introduced in the recurrence due to Case 3, since without such an ordering, it would not be possible to solve the two subproblems independently.

Overall we can compute f, and the corresponding minimum cost r-connected temporal subgraph, in time $O(nk^2 3^k (L + k)^{k+1})$. □

Chapter 5

The Approximability of All-Pairs Minimum Temporal Connectivity

In this section, we study the approximability of the all-pairs version of Minimum Temporal Connectivity in general temporal graphs. Reducing MTC to r-MTC and to Directed Steiner Forest, we obtain polynomial-time approximation algorithms for MTC, albeit with not so strong guarantees (Corollary 5.1.1 and Theorem 5.1.2). Specifically, we get an $O(\min\{n^{1+\varepsilon}, (\Delta M)^{2/3+\varepsilon}\})$-approximation for MTC, which also implies an $O(n^{2/3+\varepsilon})$-approximation if M is quasilinear and Δ polylogarithmic. To justify the poor approximation ratios, we reduce Symmetric Label Cover (SLC) to MTC and show that any $\rho(n)$-approximation for MTC implies a $\rho(n^2)$-approximation for SLC (Theorem 5.2.1), which implies that MTC cannot be approximated within a factor of $O(2^{\log^{1-\varepsilon}})$ for any $\varepsilon > 0$ unless NP $\subseteq$ DTIME$(n^{\text{poly}\log n})$ (see also [Dodi99, Section 4]). Moreover, using an approximation-preserving reduction from the Steiner Tree problem, we show that the unweighted version of MTC is APX-hard (Theorem 5.3.1).

To understand the approximability of MTC, we use a reduction to Directed Steiner Forest, so we define the problem formally.

Definition 5.0.6 (Directed Steiner Forest)**.** Given a directed edge-weighted graph $G(V, E)$ with n vertices and m edges, and a collection $D \subseteq V \times V$ of k ordered vertex pairs, the Directed Steiner Forest (DSF) problem asks for a subgraph of G that contains an $s - t$ path for each $(s, t) \in D$ and has minimum total weight. [Feld12] presents a polynomial-time $O(n^{\varepsilon} \min\{n^{4/5}, m^{2/3}\})$ -approximation for DSF, for any constant $\varepsilon > 0$. An $O(2^{\log^{1-\varepsilon} n})$-appproximation algorithm for this problem for some $\varepsilon > 0$ would imply that NP $\subseteq$ DTIME$(n^{\text{poly}\log n})$, as proven in [Dodi99, Section 4].

5.1 Approximation Algorithms for MTC

Using every vertex of the temporal graph as a source vertex and taking the union of the solutions obtained by the algorithm of Theorem 4.2.1 for r-MTC, we obtain the following.

***Corollary* 5.1.1.** *For any constant $\varepsilon > 0$, there is a polynomial-time $O(n^{1+\varepsilon})$-approximation algorithm for* MTC *on temporal graphs with n vertices.*

Next, we present a reduction from MTC to Directed Steiner Forest (DSF) that leads to a different algorithm. Although the approximation ratio may be worse than $O(n^{1+\varepsilon})$ in general, this algorithm gives significantly better guarantees if the total number of temporal edges is quasilinear (and if the maximum degree of the underlying graph is polylogarithmic).

***Theorem* 5.1.2.** *Let $\mathcal{G}$ be a temporal graph with n vertices and M temporal edges such that the underlying graph has maximum degree Δ. Then, for any constant $\varepsilon > 0$, there is a polynomial-time $O(M^{\varepsilon} \min\{M^{4/5}, (\Delta M)^{2/3}\})$-approximation algorithm for* MTC *on $\mathcal{G}$. If $M = O(n \,\text{poly}\log n)$, we obtain an approximation ratio of $O(n^{4/5+\varepsilon})$. If both $M = O(n \,\text{poly}\log n)$ and $\Delta = O(\text{poly}\log n)$, we obtain an approximation ratio of $O(n^{2/3+\varepsilon})$.*

Proof. The reduction from MTC to DSF is a generalized and refined version of the reduction in Theorem 4.2.1. Let I be an instance of MTC consisting of an underlying graph $G(V, E)$, a finite

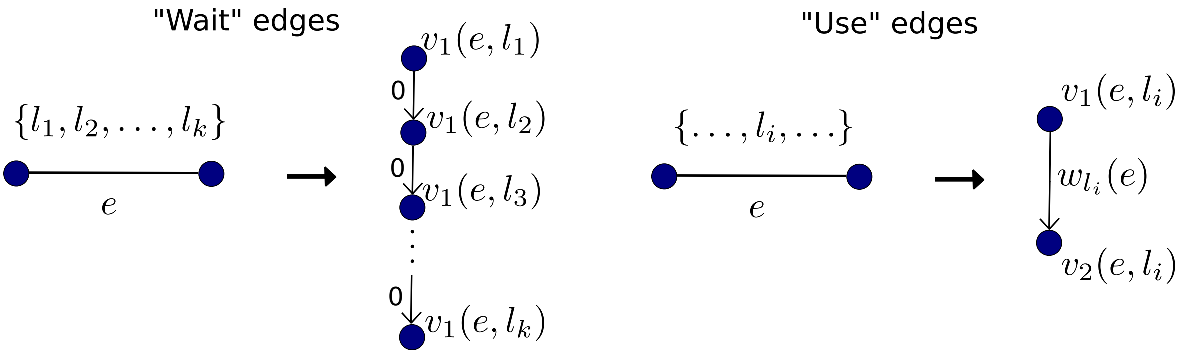


Figure 5.1: Gadgets used for the MTC $\rightarrow$ DSF transformation: ”Wait” and ”Use” edges

set of time labels L_e for each edge e, and a weight $w(e,t)$ for any temporal edge (e,t). We show how to transform I into an instance I' of DSF so that (i) any feasible solution of I can be mapped to a feasible solution of I' with no larger total weight; and (ii) given a feasible solution of I', we can compute a feasible solution of I with no larger total weight.

For convenience, we denote H the edge-weighted directed graph of the DSF instance I'. For every temporal edge (e,t) of $\mathcal{G}$, H contains two vertices $h^1_{(e,t)}$ and $h^2_{(e,t)}$. Intuitively, $h^1_{(e,t)}$ indicates that we may use (e,t) and $h^2_{(e,t)}$ indicates that we actually use (e,t). For each edge $e \in E$, let $l_1(e) < l_2(e) < \ldots < l_k(e)$ be the time labels in L_e. For every $i \in [k-1]$, H contains a directed edge $(h^1_{(e,l_i(e))}, h^1_{(e,l_{i+1}(e))})$ with weight 0. Intuitively, these edges indicate that we can wait and use e at some later time up to $l_k(e)$. Moreover, for every $i \in [k]$, H contains a directed edge $(h^1_{(e,l_i(e))}, h^2_{(e,l_i(e))})$ with weight $w(e, l_i(e))$. This edge indicates that we actually use the temporal edge $(e, l_i(e))$ and incur the corresponding cost. See an example of these two gadgets in Figure 5.1.

For every ordered pair of temporal edges $(e_1, t_1), (e_2, t_2)$ of $\mathcal{G}$, such that $e_1 \neq e_2$, t_2 is the smallest time label in L_{e_2} such that $t_2 \geq t_1$ ($t_2 > t_1$, for strict connectivity), and e_1 and e_2 share a common endpoint, H contains a directed edge $(h^2_{(e_1,t_1)}, h^1_{(e_2,t_2)})$ with weight 0.

For every vertex $v_i \in V$, $i \in [n]$, H contains a pair of terminal vertices s_i and t_i. For every temporal edge (e,t) incident to v_i, H contains a directed edge $(s_i, h^1_{(e,t)})$ with weight 0 and a directed edge $(h^2_{(e,t)}, t_i)$ with weight 0. The set of connection requirements of the DSF instance I' consists of all pairs (s_i, t_j) for all $i, j \in [n]$ with $i \neq j$. An example of the above process that transforms an instance of the MTC problem to an instance of the DSF problem is presented in Figure 5.2.

By construction, any temporal $v_i - v_j$ path, which consists of a temporal edge sequence $((e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k))$, with e_1 starting at v_i and e_k ending at v_j, corresponds to a directed $s_i - t_j$ path in H of the form

$$(s_i, h^1_{(e_1,t_1)}), (h^1_{(e_1,t_1)}, h^2_{(e_1,t_1)}), (h^2_{(e_1,t_1)}, h^1_{(e_2,t_2'')}), (h^1_{(e_2,t_2'')}, h^1_{(e_2,t_2')}),$$

$$(h^1_{(e_2,t_2')}, h^1_{(e_2,t_2)}), (h^1_{(e_2,t_2)}, h^2_{(e_2,t_2)}), \ldots, (h^1_{(e_k,t_k)}, h^2_{(e_k,t_k)}), (h^2_{(e_k,t_k)}, t_j)$$

with the same weight and vice versa. Using this observation, we can now establish claims (i) and (ii). Specifically, to show (i), we construct a feasible solution to I' that includes all directed edges of weight 0 and the directed edges $(h^1_{(e,t)}, h^2_{(e,t)})$ corresponding to the temporal edges (e,t) used in the feasible solution to I. Clearly, the two solutions have the same total weight and any temporal $v_i - v_j$ path in the solution to I corresponds to an $s_i - t_j$ path in the solution to I'. To show (ii), we first observe that any directed path from some s_i to some t_j should include some directed edges of the form $(h^1_{(e,t)}, h^2_{(e,t)})$ with weight $w(e,t)$. So, we construct a feasible solution to I that includes the temporal edges (e,t) corresponding to the positive-weight directed edges $(h^1_{(e,t)}, h^2_{(e,t)})$ included in the feasible solution to I'.

In the resulting DSF instance I', the total number of vertices is $O(n + M) = O(M)$ and the number of connection requirements is $O(n^2)$. If the maximum degree of the underlying graph is Δ,

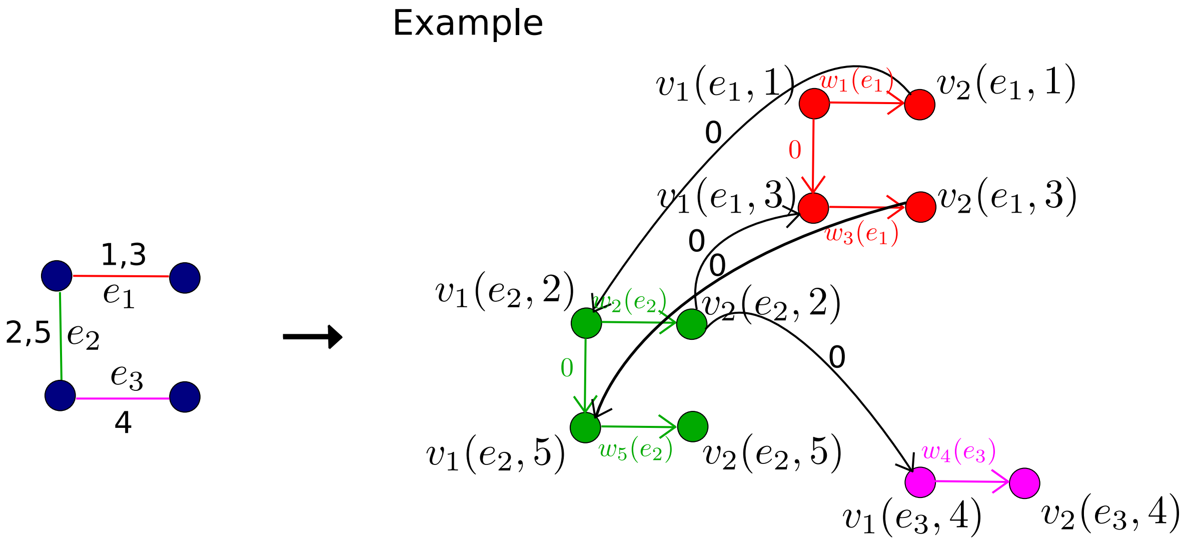


Figure 5.2: Example of a transformation of a MTC instance to a DSF instance

the total number of edges is dominated by the edges of the form $(h^2_{(e_1,t_1)}, h^1_{(e_2,t_2)})$, which are $O(\Delta M)$. Applying the approximation algorithm of [Feld12, Theorem 1.1] to the DSF instance I', we obtain a polynomial-time $O(M^{\varepsilon} \min\{M^{4/5}, (\Delta M)^{2/3}\})$-approximation algorithm, for any constant $\varepsilon > 0$. In the special case where the number of temporal edges is $M = O(n \text{poly} \log n)$, we obtain an $O(n^{4/5+\varepsilon})$-approximation, for any constant $\varepsilon > 0$. If both $M = O(n \text{poly} \log n)$ and the maximum degree of the underlying graph $\Delta = O(\text{poly} \log n)$, we obtain a polynomial-time $O(n^{2/3+\varepsilon})$-approximation algorithm for any constant $\varepsilon > 0$. □

5.2 A Lower Bound on the Approximability of MTC

In this section, we present an approximation-preserving reduction from Symmetric Label Cover to MTC. Our reduction along with standard inapproximability results for Symmetric Label Cover indicate that MTC in general temporal networks is hard to approximate.

***Theorem* 5.2.1.** MTC *on temporal graphs with n vertices cannot be approximated within a factor of $O(2^{\log^{1-\varepsilon} n})$, for any constant $\varepsilon > 0$, unless* NP $\subseteq$ DTIME$(n^{\text{poly} \log n})$.

Proof. We present a polynomial-time approximation-preserving reduction from the Symmetric Label Cover (SLC) problem to MTC. In SLC (see e.g., [Dodi99, Definition 4.1]), we are given a complete bipartite graph $H(U, W)$, with $|U| = |W|$, a finite set of colors C and a binary relation $R(u, w) \subseteq C \times C$ for every vertex pair $(u, w) \in U \times W$. We seek to assign a color subset $\sigma(u) \subseteq C$ to each vertex $u \in U \cup W$ so that for every vertex pair $(u, w) \in U \times U$, there are colors $a \in \sigma(u)$ and $b \in \sigma(w)$ with $(a, b) \in R(u, w)$ and $\sum_{u \in U \cup W} |\sigma(u)|$, i.e., the total number of colors used, is minimized.

Given an instance of SLC, we create a temporal graph $\mathcal{G}$ whose vertex set V is partitioned into six sets V_U, $V_{C(U)}$, V_W, $V_{C(W)}$, V_X and $\{p, q\}$. There is a correspondence between the vertices of the bipartite graph H and the vertices of $\mathcal{G}$ in the sets V_U and V_W. The vertex sets $V_{C(U)} = V_U \times C$ and $V_{C(W)} = V_W \times C$ serve to encode the color assignment to the vertices of U and W in the SLC instance. Moreover, V_X contains a vertex (u, w, a, b) for every vertex pair $(u, w) \in U \times W$ and every allowable color pair $(a, b) \in R(u, w)$. Intuitively, the vertices of V_X serve to ensure that the color assignment is consistent. Finally, the vertices p and q ensure that the temporal graph $\mathcal{G}$ is connected.

For every $u \in V_U$ and every $(u, a) \in V_{C(U)}$, $\mathcal{G}$ contains a temporal edge $\{u, (u, a)\}$ with time label 1 and weight 1. Similarly, for every $w \in V_W$ and every $(w, b) \in V_{C(W)}$, $\mathcal{G}$ contains a temporal edge $\{w, (w, b)\}$ with time label 4 and weight 1. For every vertex $(u, w, a, b) \in V_X$, $\mathcal{G}$ contains the temporal edges $\{(u, a), (u, w, a, b)\}$ with time label 2 and weight 0 and $\{(u, w, a, b), (w, b)\}$ with label

3 and weight 0. The temporal graph $\mathcal{G}$ contains temporal edges with label 5 and weight 0 between p and every vertex in $V_U \cup V_{C(U)} \cup V_{C(W)} \cup V_X$ and between q and every vertex in V_U. Moreover, $\mathcal{G}$ contains temporal edges with label 0 and weight 0 between p and every vertex in V_W and between q and every vertex in $V_W \cup V_{C(U)} \cup V_{C(W)} \cup V_X$. We note that the temporal graph $\mathcal{G}$ is connected and has $O(k^2c^2)$ vertices, where $k = |U| = |W|$ and $c = |C|$ (in fact, the number of vertices of $\mathcal{G}$ is of the same order as the total size of all binary relations $R(u, w)$).

We next show that this reduction is approximation-preserving. We first show that any feasible solution to the SLC instance can be mapped to a temporally connected subgraph $\mathcal{G}'$ of $\mathcal{G}$ with at most the same weight. Let us fix any assignment σ of a color set to each vertex of H that is feasible for the SLC instance. We first include in $\mathcal{G}'$ all temporal edges of weight 0. For every vertex $u \in U$ with assigned colors $\sigma(u)$, we include in $\mathcal{G}'$ the temporal edges $\{u, (u, a)\}$, for all $a \in \sigma(u)$. The total weight of these edges is $|\sigma(u)|$. Similarly, for every vertex $w \in W$, we include in $\mathcal{G}'$ the temporal edges $\{w, (w, b)\}$, for all $b \in \sigma(w)$. The total weight of these edges is $|\sigma(w)|$. Therefore, the total weight of the temporal subgraph $\mathcal{G}'$ is equal to the cost of the solution σ for the SLC problem.

It remains to show that $\mathcal{G}'$ is temporally connected. All vertices in $V_U \cup V_{C(U)} \cup V_{C(W)} \cup V_X \cup \{p\}$ are connected with each other (through p) by temporal edges with time label 5. There are also temporal $p - q$ and $q - p$ paths consisting of edges with time label 5 through the vertices of V_U. Similarly, all vertices in $V_W \cup V_{C(U)} \cup V_{C(W)} \cup V_X \cup \{q\}$ are connected with each other (through q) by temporal edges with time label 0. Moreover, there is a temporal path (using edges with time labels 0 and 5) from every vertex in $V_W \cup V_{C(U)} \cup V_{C(W)} \cup V_X \cup \{q\}$ to every vertex in V_U. Also, p is connected to every vertex in V_W with temporal edges of time label 0 and vice versa. All these vertex pairs are connected through temporal paths entirely consisting of 0-weight edges. The really interesting case concerns vertex pairs $(u, w) \in V_U \times V_W$. By the feasibility of the solution σ, for every vertex pair $(u, w) \in U \times W$, there are colors $a \in \sigma(u)$ and $b \in \sigma(w)$ such that $(a, b) \in R(u, w)$. Therefore, the temporal $u - w$ path $(u, (u, a), (u, w, a, b), (w, b), b)$ is included in $\mathcal{G}'$. Hence, $\mathcal{G}'$ is temporally connected.

We also need to show that given a temporally connected subgraph $\mathcal{G}'$ of $\mathcal{G}$, we can efficiently compute an assignment σ of a color set to each vertex in $U \cup W$ that is feasible for the SLC instance and has total cost no larger than the total weight of $\mathcal{G}'$. For every $u \in V_U$ and every temporal edge of the form $(\{u, (u, a)\}, 1)$ included in $\mathcal{G}'$, we include the color a in $\sigma(u)$. Similarly, for every $w \in V_W$ and every temporal edge of the form $(\{w, (w, b)\}, 4)$ included in $\mathcal{G}'$, we include the color b in $\sigma(w)$. Since these are the only edges of $\mathcal{G}$ (and $\mathcal{G}'$) with positive weight, the total cost of σ is equal to the total weight of $\mathcal{G}'$.

It remains to show that σ is a feasible solution to the SLC instance. Let $(u, w) \in U \times W$ of the bipartite graph H in the SLC instance. The crucial observation is that the only way to connect $u \in V_U$ to $w \in V_W$ in $\mathcal{G}'$ is through a temporal path consisting of the temporal edge sequence $(\{u, (u, a)\}, 1)$, $(\{(u, a), (u, w, a, b)\}, 2)$, $(\{(w, b), (u, w, a, b)\}, 3)$, $(\{w, (w, b)\}, 4)$, for some colors a, b such that $(a, b) \in R(u, w)$. This claim immediately implies the feasibility of the assignment σ. To prove this claim, we observe that a temporal $u - w$ path cannot use any temporal edge incident to p or q, since all edges between V_U and $\{p, q\}$ have time label 5 and all edges between V_W and $\{p, q\}$ have time label 0. So, any temporal $u - w$ path in $\mathcal{G}'$ has to move from u to some vertex $(u, a) \in V_{C(U)}$. Such a vertex $(u, a) \in V_{C(U)}$ does not have any neighbors in V_U other than u. Hence, the next vertex of any temporal $u - w$ path in $\mathcal{G}'$ must be to visit some $(u, w, a, b) \in V_X$, where $w \in W$ and $(a, b) \in R(u, w)$. Similarly, since such a vertex $(u, w, a, b) \in V_X$ does not have any neighbors in $V_{C(U)}$ other than (u, a) and any neighbors in $V_{C(W)}$ other than (w, b), we conclude that the next step of any temporal $u - w$ path in $\mathcal{G}'$ must be to the vertex $(w, b) \in V_{C(W)}$. But now, the last temporal edge used has time label 3, which implies that the $u - w$ path cannot use any edges with time labels 1 and 2 anymore. Thus, the path cannot return to $V_U \cup V_{C(U)}$. The only choice now is that the path moves to w through the temporal edge $(\{w, (w, b)\}, 4)$, which establishes the claim about the structure of any temporal $u - v$ path in $\mathcal{G}'$.

The discussion above establishes the correctness of the reduction from SLC to MTC. Using the fact that the number of vertices of the temporal graph $\mathcal{G}$ is quadratic in the number of vertices of the bipartite graph H and standard inapproximability results for SLC (e.g., that used in [Dodi99]), we conclude the proof of the theorem. □

Adjusting the proof of Theorem 5.2.1, we can get an approximation-preserving reduction from the MinRep problem, which is considered in [Char11] and is similar to SLC, to MTC. Thus, we obtain that any polynomial-time $\rho(n)$-approximation algorithm for MTC on simple temporal graphs implies a polynomial-time $\rho(n^2)$-approximation algorithm for MinRep. Since the best known approximation ratio for MinRep is $O(n^{1/3}\log^{2/3} n)$ [Char11, Section 2], any $O(n^{1/6})$-approximation to MTC would imply an improved approximation ratio for MinRep.

5.3 Inapproximability of Unweighted MTC

In this section we present an approximation-preserving reduction from the Steiner Tree problem on undirected graphs with edge weights either 1 or 2 to MTC on unweighted temporal graphs, where all temporal edges have weight equal to 1. Since this version of the Steiner Tree problem is known to be APX-hard [Bern89], we obtain the following.

***Theorem* 5.3.1.** MTC *on unweighted temporal graphs is* APX*-hard, and thus it does not admit a PTAS, unless* P $=$ NP.

We use an approximation-preserving reduction from Steiner Tree on undirected graphs with edge weights 1 and 2 to unweighted MTC.

Given an undirected edge-weighted graph $G(V, E)$ and a set of terminals $S \subseteq V$, the Steiner Tree (ST) problem asks for a connected subgraph of G that spans S and has minimum total weight. It is easy to see that the optimal solution is always a tree. If the weight of each edge is either 1 or 2, we get the Steiner Tree problem with weights 1 and 2 $(\mathrm{ST}(1, 2))$.

In [Bern89], $\mathrm{ST}(1, 2)$ is shown APX-hard by a reduction from Vertex Cover on graphs with bounded maximum degree. If n and m are the number of vertices and edges of the Vertex Cover instance respectively, then the $\mathrm{ST}(1, 2)$ instance has m terminals and n non-terminals. Furthermore, all the terminals have degree 2 and the graph is bipartite, the one part consists of the terminal vertices and the other part consists of the non-terminal vertices. This means that the total number of edges is at most $2m$. In order to use uniform weights for the edges of the temporal graph, we substitute each edge of weight 2 with a 2-edge path, where each edge has weight 1 and the newly added vertex is a non-terminal vertex. Since at most one non-terminal vertex per edge is added, the resulting number of non-terminal vertices is at most $n + 2m \leq 3m$, assuming that the initial graph is connected and not a tree. So we obtain an instance of the Steiner Tree problem with uniform edge-weights, a set S of m terminal vertices and a set T of at most $3m$ non-terminal vertices.

To reduce this special case of the Steiner Tree problem to MTC with uniform temporal edge weights, we create a temporal variant of the above graph, where each existing edge has label 3. To ensure temporal connectivity, we add some new elements to our temporal graph. We add new vertices p and q and connect every non-terminal vertex in T to p with time label 4 and to q with time label 2. These edges ensure that there is a temporal path from every vertex in $S \cup T$ (terminal and non-terminal vertices, respectively) to every other vertex in the same set.

It remains to ensure temporal connectivity between vertices p and q and to the rest of the temporal graph. Suppose we enumerate the elements of S as $u_1, u_2, \ldots, u_m$. We add m new vertices $a_1, a_2, \ldots, a_m$, composing vertex set A. To create a path from p to any vertex of S, add temporal edges with label 1 from p to every vertex in A, and for each $i \in [m]$, add a temporal edge between a_i and u_i with label 5. Similarly, to connect S to q, add a new vertex set B consisting of vertices $b_1, b_2, \ldots, b_m$, for every i add a temporal edge between u_i and b_i with label 1, and add an edge between q and every vertex in B with label 5. Finally, to create a temporal path from p to q, add a new vertex x, which is connected to p with label 1 and to q with label 5. Then, we solve MTC on the resulting temporal graph $\mathcal{G}$.

We are ready to show the following. Note that Theorem 5.3.1 follows from Lemma 5.3.2 below and the fact that Steiner Tree with edge weights 1 and 2 is APX-hard.

***Lemma* 5.3.2.** *An* $(1+\varepsilon)$*-approximate solution to* MTC *implies an* $(1+12\varepsilon)$*-approximate solution to the Steiner Tree instance with uniform weights.*

Proof. The proof consists of two main claims. The first claim is that any feasible solution to MTC on the resulting temporal graph $\mathcal{G}$ uses all the edges that are not in the Steiner Tree instance and that none of them can be part of some temporal path between vertices of S. The second claim is that taking all the edges not in the Steiner Tree instance together with the set of temporal edges corresponding to a feasible solution to the Steiner Tree instance results in a temporally connected subgraph of $\mathcal{G}$.

Having established these claims and assuming that the number of the additional (not in the Steiner Tree instance) edges is $k \leq 4m + 6m + 2 \leq 11m \leq 11\,\text{OPT}$, where OPT is the optimal solution to the Steiner Tree instance, we obtain that the MTC solution has cost at most $(1+\varepsilon)(\text{OPT}+k)$. Deleting these k edges, we have a feasible solution to Steiner Tree of cost at most $(1+\varepsilon)\text{OPT}+\varepsilon k \leq (1+12\varepsilon)\text{OPT}$.

Let us first show that any feasible solution to MTC on the temporal graph $\mathcal{G}$ resulting by the reduction above uses all the edges that are not in the Steiner Tree instance. We examine several cases:

Edge between u_i and a_i is missing for some i: Then there is no temporal path from u_i to a_i, since a_i is only adjacent to a temporal edge with label 1, and there is no 1-labeled path from u_i to a_i.

Edge between u_i and b_i is missing for some i: Then there is no temporal path from b_i to u_i, since b_i is only adjacent to a temporal edge with label 5, and there is no 5-labeled path from b_i to u_i.

Edge between a_i and p is missing for some i: Then there is no temporal path from a_i to p, since a_i is only adjacent to a temporal edge with label 5, and there is no 5-labeled path from a_i to p.

Edge between b_i and q is missing for some i: Then there is no temporal path from q to b_i, since b_i is only adjacent to a temporal edge with label 1, and there is no 1-labeled path from q to b_i.

Edge between p and $t \in T$ is missing: Then there is no temporal path from p to t: Going from p to A or x doesn't help, since we then have to use label 5, which is a dead-end. Using another edge with label 4 is also a dead-end.

Edge between q and $t \in T$ is missing: Then there is no temporal path from t to q: Reasons similar as above.

Edge between p and x or edge between q and x is missing: Then there is no temporal path from x to p, or from q to x respectively.

So, we have shown that all temporal edges not present in the Steiner Tree instance are necessary for a feasible solution to the MTC instance. Furthermore, every temporal path between two vertices of S can consist only of temporal edges with label 3: If an edge between S and A is traversed, then we can no longer proceed, since it has label 5. If an edge between S and B is traversed, then an edge between B and q must be traversed, but then we can no longer proceed to S, since we have traversed an edge with label 5. This means that the edges with label 3 in some feasible solution define a feasible solution to the Steiner Tree instance.

Let us now show that taking all the edges not in the Steiner Tree instance together with the set of temporal edges corresponding to a feasible solution to the Steiner Tree instance results in a temporally connected subgraph of $\mathcal{G}$. For this, it suffices to prove that the vertices in $S \cup T \cup \{p\} \cup \{q\}$ are connected to each other with temporal paths, since the rest of the vertices are connected to them with both minimum and maximum labels (1 and 5). Hence, temporal connectivity is implied. We already have that there is a 3-labeled path between each pair of vertices in S. Consider the following cases:

From $u_i \in S$ **to** $t \in T$**:** Move from u_i to some $t' \in T$ with label 3, and then to t with label 4 via p. (This also connects u_i to p, and t' to t).

From $u_i \in S$ **to** q**:** (u_i, b_i, q)

From $t \in T$ **to** $u_i \in S$**:** Move from t to some t' (with label 2 via q) that is connected to $u_j \in S$ with label 3, and then move on the 3-labeled path to u_i. (This also connects t to q, q to t and q to u_i).

From $t \in T$ **to** p**:** Move on edge with label 4. (This also connects p to t).

From p **to** $u_i \in S$**:** (p, a_i, u_i)

From p **to** q**:** (p, x, q)

From q **to** p**:** (q, t, p), for some $t \in T$.

Since we have shown all the required facts, the claim is proven. □

Chapter 6

All-Pairs Temporal Connectivity on Trees and Cycles

In Section 6, we show that MTC can be solved much more efficiently if we restrict the family of the underlying graph. Specifically, in Section 6.1, Lemma 6.1.2, we show that if the underlying graph is a tree, there is an optimal solution to the MTC problem that uses each edge with at most two time labels. Using this structural property, we can show that MTC can be solved efficiently by dynamic programming if the underlying graph is a tree (see Section 6.1 for the details). Also, we show that MTC is 2-approximable if the underlying graph is a cycle (Theorem 6.2.1).

6.1 Temporal Connectivity on Trees: A Polynomial-Time Algorithm

***Theorem* 6.1.1.** *Let $\mathcal{G}$ be a temporal tree on n vertices with lifetime L. There is a dynamic programming algorithm that computes an optimal solution to* MTC *on $\mathcal{G}$ in time $O(nL^4)$.*

***Lemma* 6.1.2.** *Let $\mathcal{G}$ be a temporal tree. Then, there is an optimal solution to* MTC *on $\mathcal{G}$ that uses at most two time labels of each edge.*

Proof. Let T be the underlying tree of $\mathcal{G}$. For some edge e, denote by T_A, T_B the connected components in which the T is partitioned after the deletion of e. For some fixed node u in T_A and every node v in T_B, there is a temporal path in the optimal solution that connects u to v. Among all these temporal paths, let $(u, (e_1, t_1), (e_2, t_2), \ldots, (e, t), \ldots, v)$ be the path with the minimum traversal time for edge e. Note that for all nodes v' of T_B other than v, there exists a path $(u, (e'_1, t'_1), (e'_2, t'_2), \ldots, (e, t'), p', v')$ from u to v'. Since $t' \geq t$, the path $(u, (e_1, t_1), (e_2, t_2), \ldots, (e, t), p', v')$ is also a valid temporal $u - v'$ path. This means that in some optimal solution to MTC, one time label $t(e, u)$ on edge e can be used by all paths from u to vertices in T_B. If we keep as the time label of e the maximum time label $t(e, v)$, over all vertices $v \in T_A$, this can be used by any temporal path from some vertex in T_A to some vertex in T_B. Hence, for all paths from T_A to T_B, one label suffices for e. By symmetry, the same holds for all paths from T_B to T_A. Overall, there is an optimal solution to MTC where at most two time labels are used for any edge e. □

Now, MTC on temporal trees can be solved as follows: We root the tree arbitrarily at some vertex r. For some vertex u, let $f(u, t_i, t_o)$ denote the minimum cost of MTC on the temporal tree T_u rooted at u, with the additional constraint that for every node v in T_u, there must exist a temporal path from u to v using only temporal edges with time label at least t_i and a temporal path from v to u using only temporal edges with label at most t_o. Denote by c_j, for $i \in [1, x_u]$ the children of u in the tree, where x_u is the number of children. Also, let $g(u, j, t_i, t_o)$ denote the minimum cost for MTC on the temporal tree rooted at u but with no children other than $c_1, \ldots, c_j$, with the constraints concerning t_i, t_o as stated above, but restricted to the subtrees of the first j children. Obviously, $g(u, 0, t_i, t_o) = 0$ and $g(u, x_u, t_i, t_o) = f(u, t_i, t_o)$. We now observe that

$$g(u, j, t_i, t_o) = \min_{t'_i, t'_o \in \mathbb{N}} \big\{ g(u, j-1, \max(t_i, t'_o), \min(t_o, t'_i)) + f(c_j, t'_i, t'_o) + q(u, c_j, t'_i, t'_o) \ : \ t'_i \geq t_i \text{ and } t'_o \leq t_o \big\}$$

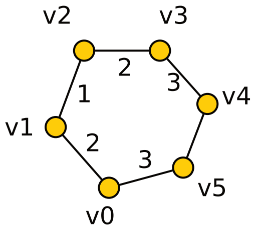


Figure 6.1: *
(a) v_1 is connected to any other vertex through temporal paths v_1, v_2, v_3, v_4 and v_1, v_0, v_5.

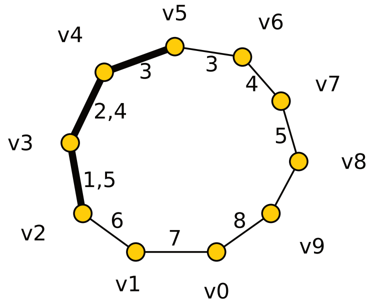


Figure 6.2: *
(b) The sector v_2, v_3, v_4, v_5. Its increasing and decreasing paths are $v_2, v_3, \dots, v_8$ (using labels $1, 2, 3, 3, 4, 5$) and $v_5, v_4, \dots, v_9$ (using labels $3, 4, 5, 6, 7, 8$) respectively.

Figure 6.3: Connectivity in temporal cycles.

In the equation above, t'_i denotes the time that the edge $\{u, c_j\}$ is traversed from u to c_j and t'_o denotes the time that $\{u, c_j\}$ is traversed from c_j to u. Moreover,

$$q(u, c_j, t'_i, t'_o) = \begin{cases} w((u, c_j), t'_i) + w((u, c_j), t'_o) & \text{if } t'_i \neq t'_o \\ w((u, c_j), t'_i) & \text{otherwise} \end{cases}$$

If L is the total number of temporal edges, then f and g can be computed in time $O(nL^4)$ using dynamic programming. The optimal value of MTC is then $f(r, 0, \infty)$. In order to get the actual MTC solution, we just follow the parent links between states of the recurrence relation.

6.2 Temporal Connectivity on Cycles: 2-Approximation Algorithm

In this section, we observe that if the underlying graph is a cycle $C_n = (v_0, v_1, \dots, v_{n-1}, v_0)$, any temporally connected subgraph $\mathcal{G}'$ can be partitioned into *sectors*. A sector is a connected part $(v_i, v_{i+1}, \dots, v_k)$ of the cycle for which there is a vertex $v_j \notin \{v_i, \dots, v_{k-1}\}$ such that the temporal paths $p_{\text{incr}} = (v_i, v_{i+1}, \dots, v_j)$ and $p_{\text{decr}} = (v_k, v_{k-1}, \dots, v_{j+1})$ are present in $\mathcal{G}'$ (the vertex indices along C_n are taken modulo n). Intuitively, any vertex in the sector $(v_i, v_{i+1}, \dots, v_k)$ can reach every vertex in C_n through the paths p_{incr} and p_{decr}. Then, in Lemma 6.2.4, we show that there is an optimal solution to the MTC problem on C_n where each edge is shared by at most two different sectors. Then, ignoring edges shared by different sectors and using dynamic programming to determine a near optimal partitioning of C_n into sectors, we obtain the following.

***Theorem* 6.2.1.** *There is a polynomial-time* 2*-approximation algorithm for the* MTC *problem on any temporal cycle C_n.*

When the underlying graph is a cycle, we have a 2-approximation algorithm for MTC. In order for the temporal graph to be connected, for every vertex in the cycle there must exist temporal paths to every other vertex. Specifically, if we mark with $v_0, v_1, v_2, \dots, v_{n-1}, v_n \equiv v_0$ the nodes of the cycle, then for a solution to be feasible, for every node v_i there must exist a node v_j, such that there are temporal paths $v_i, v_{i+1}, \dots, v_{j-1}, v_j$ and $v_i, v_{i-1}, \dots, v_{j+2}, v_{j+1}$ in the solution (vertex indices are taken modulo n). See also Figure 6.3.a.

***Lemma* 6.2.2.** *Suppose that in some temporally connected subgraph $\mathcal{G}'$ of the temporal cycle, there is a temporal path $v_i, v_{i+1}, \dots, v_j$ and for some $k \in \{i, i+1, \dots, j\}$ there is also a temporal path*

$v_k, v_{k-1}, \dots, v_{j+2}, v_{j+1}$. *Then every vertex $v_p \in \{i, i+1, \dots, k\}$ is connected to every other vertex in $\mathcal{G}'$.*

Proof. Since the temporal path $v_i, v_{i+1}, \dots, v_j$ exists in $\mathcal{G}'$, so does the path $v_p, v_{p+1}, \dots, v_j$. Also, since the path $v_k, v_{k-1}, \dots, v_{j+2}, v_{j+1}$ exists in $\mathcal{G}'$, so does the path $v_p, v_{p-1}, \dots, v_{j+2}, v_{j+1}$. So there are paths from v_p to every other vertex in $\mathcal{G}'$. □

This leads us to the following definition (see also Figure 6.3.b for an example).

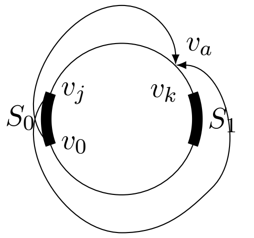
Definition 6.2.3. Consider a temporal cycle $\mathcal{G}$ and a temporally connected subgraph $\mathcal{G}'$ of $\mathcal{G}$. A *sector* of $\mathcal{G}$ with respect to $\mathcal{G}'$ is a contiguous sequence of vertices $(v_i, v_{i+1}, \dots, v_j)$ (it may be $i = j$) such that there exists a vertex $v_k \notin \{i, i+1, \dots, j-1\}$ and the temporal paths $p_{\text{incr}} = (v_i, v_{i+1}, \dots, v_k)$ and $p_{\text{decr}} = (v_j, v_{j-1}, \dots, v_{k+1})$ are present in $\mathcal{G}'$. We refer to p_{incr} as the *increasing path* and to p_{decr} as the *decreasing path* of the sector $(v_i, v_{i+1}, \dots, v_j)$. The *cost* of a sector $S = (v_i, v_{i+1}, \dots, v_j)$ is the minimum, over all choices of increasing and decreasing paths pairs for S (in an increasing and decreasing path pair, if the increasing path ends at k, the decreasing ends at $k+1$), of the total cost of all temporal edges present in these increasing and decreasing paths.

Note that the vertices of a temporal cycle $\mathcal{G}$ with a temporally connected subgraph $\mathcal{G}'$ can always be partitioned into sectors with respect to $\mathcal{G}'$.

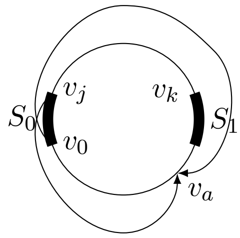
***Lemma* 6.2.4.** *For a temporal cycle $\mathcal{G}$, there exists an optimal* MTC *solution $\mathcal{G}'$, such that no two sectors' increasing (resp. decreasing) paths use the same temporal edge (the same edge at the same time).*

Proof. Suppose that $\mathcal{G}'$ is partitioned to s disjoint sectors and that s is minimum among all optimal solutions. Furthermore, suppose that there are two (disjoint) sectors, $S_0 : v_0, v_1, \dots, v_j$ and $S_1 : v_k, v_{k+1}, \dots, v_l$, defined by this partition, whose increasing paths contain a common temporal edge. Note that disjointness implies $j < k$ and $l < n(\mathcal{G})$. We will show that this contradicts the minimality of s. Suppose the increasing path of sector S_0 is $v_0, v_1, \dots, v_a$ and the increasing path of sector S_1 is $v_k, v_{k+1}, \dots, v_b$. We distinguish between two cases:

$a \in [j, k-1]$**:** In order for the two increasing paths to have a common temporal edge, the increasing path of S_1 must be of the form $v_k, v_{k+1}, \dots, v_z, v_{z+1}, \dots, v_m$, where $z \in \{0, 1, \dots, a-1\}$ and the edge between v_z and v_{z+1} is visited at the same time by the two increasing paths. This means that we can merge these two paths, creating a path $v_k, v_{k+1}, \dots, v_a$. Combining this with the decreasing path of S_0, which is $v_j, v_{j-1}, \dots, v_{a+1}$, this means that we can replace all sectors that contain vertices $v_k, v_{k+1} \dots, v_j$ with a single sector consisting of exactly those vertices. Additionally, we have not added extra edges so the cost of the solution will not increase. Also note that the case $b \in [l, n(\mathcal{G}) - 1]$ is symmetric to the above case.



$a \in [k, n(\mathcal{G}) - 1]$ **and** $b \in [0, k-1]$**:** Let the common temporal edge of the two paths be connecting v_z and v_{z+1}. If $z \in \{0, 1, \dots, k-1\}$, then merging the two increasing paths we obtain a new path $v_k, v_{k+1}, \dots, v_z, v_{z+1}, \dots, v_k, \dots, v_a$, which means that we can replace all sectors with a single sector with no extra cost. As for the case $z \in \{k, k+1, \dots, a-1\}$, we can combine temporal paths $v_0, v_1, \dots, v_a$ with $v_l, v_{l-1}, \dots, v_{a+1}$ to merge vertices $v_0, v_1, \dots, v_l$ into a single sector with no extra cost.



□

The above lemma implies that in some MTC solution, every temporal edge is used at most twice by different sectors. Thus, ignoring the mutual dependence between sectors we can get a 2-approximate solution to our problem.

***Theorem* 6.2.5.** *There is an algorithm that runs in polynomial time and outputs a feasible* MTC *solution of a temporal cycle $\mathcal{G}$, with cost at most* 2 *times the optimal.*

Proof. Denote the vertices of the cycle as $v_0, v_1, \ldots, v_n \equiv v_0$. First, we can try all possible vertices as the start of a sector. Let this be vertex v_0 of the cycle. Let $p(i)$ denote the partition of $v_0, v_1, \ldots, v_i$ into sectors with minimum cost (the cost of a set of disjoint sectors is defined as the sum of costs of the individual sectors) and $c(i)$ its cost.

Furthermore, define as inc_path(i, j) the minimum cost of a temporal path that visits $v_i, v_{i+1}, \ldots, v_j$ in this order and dec_path(i, j) the minimum cost of a temporal path that visits $v_i, v_{i-1}, \ldots, v_j$ in this order. inc_path and dec_path can be precomputed in time $O(nM)$ and space $O(n^2 + M)$, where M is the number of temporal edges.

Also, define sector_cost(i, j) to be the cost of the sector $v_i, v_{i+1}, \ldots, v_j$. This can be precomputed in time $O(n^3)$ and space $O(n^2)$, as

$$\text{sector\_cost}(i,j) = \min_{k \in \{0,\ldots,n-1\} \setminus \{i,i+1,\ldots,j-1\}} \Big\{ \text{inc\_path}(i,k) + \text{dec\_path}(j,k+1) \Big\}$$

Then, we have that $c(0) = \text{sector\_cost}(0, 0)$ and that for $i > 0$,

$$c(i) = \min\left\{ \text{sector\_cost}(0,i), \min_{j \in \{0,\ldots,i-1\}} \{c(j) + \text{sector\_cost}(j+1,i)\} \right\}$$

$p(i)$ can also be easily computed this way.

Finally, for the choice of v_0 that achieves the minimum $c(n-1)$, the algorithm outputs the union of the optimal increasing and decreasing paths of the sectors defined in $p(n-1)$.

The temporal graph that the algorithm produces is a feasible solution, because every vertex belongs to some sector and is thus connected to every other vertex.

To show that it is 2-approximate of the optimal solution, suppose that we double every temporal edge of the graph. The optimal cost OPT will not change. Yet, we can transform an optimal solution in which every temporal edge is used by at most 2 sectors to a solution in which every temporal edge is used by at most 1 sector in the doubled graph, with cost at most $2 \cdot OPT$. Either this or a smaller cost solution will be considered and produced by our algorithm. Overall, our algorithm uses $O(n^3 + n \cdot M)$ time and $O(n^2 + M)$ space. □

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή

Τα γραφήματα και τα δίκτυα πάντα ήταν ένα κεντρικό ζήτημα της Επιστήμης Υπολογιστών, διότι με αυτά μπορεί κανείς να μοντελοποιήσει πολλές πραγματικές καταστάσεις με αφηρημένο τρόπο. Αυτό μας έχει επιτρέψει να θεωρούμε τα δίκτυα σαν αφηρημένα μαθηματικά αντικείμενα, αγνοώντας λεπτομέρειες που αφορούν την εκάστοτε εφαρμογή. Τα περισσότερα πρακτικά προβλήματα που αντιμετωπίζουμε σήμερα έχουν εγγενή δομή γραφήματος, και έτσι είναι φυσικό να τα αναπαριστούμε ως τέτοια.

Παρ' όλα αυτά, τα κλασικά γραφοθεωρητικά μοντέλα που χρησιμοποιούνται είναι περιορισμένα όσον αφορά κάποιες εφαρμογές. Ένας τέτοιος περιορισμός αφορά τις μεταβολές με το χρόνο. Τα γραφήματα που χρησιμοποιούνται ευρέως στη θεωρία είναι *στατικά* και δεν μπορούν να αναπαραστήσουν με καλό τρόπο αλλαγές με το χρόνο.

Όπως παρατηρήθηκε πχ στο [Berm96, Kemp02], σε πολλές εφαρμογές γραφοθεωρητικων μοντέλων, η διαθεσιμότητα (ύπαρξη ακμής), όπως και η ισχύς της διαδραστικότητας (βάρη ακμών) δεν είναι σταθερές με το πέρασμα του χρόνου. Εφαρμογές στις οποίες παρατηρείται αυτή η συμπεριφορά περιλαμβάνουν δίκτυα μετακίνησης και σχεδιασμό διαδρομών (πχ [Berm96, Fosc14]), διάδοση πληροφοριών και κατανεμημένο υπολογισμό σε δυναμικά δίκτυα (πχ [Cast12, Dutt13, Kemp02, Kuhn10, Hede88, Ravi94]) κινητά adhoc και δίκτυα αισθητήρων (πχ [ODel05, Angl06]), βιολογικά και οικολογικά δίκτυα (πχ [Holm12, Kemp02, Mich11]), συστήματα επιρροής και σχηματισμός απόψεων (πχ [Bhaw13, Chaz12]). Σαν αποτέλεσμα της σημαντικής εφαρμοστικότητάς τους, τα τελευταία χρόνια τα δυναμικά δίκτυα έχουν τραβήξει αρκετά την προσοχή (για παράδειγμα [Avin08, Bui 03, Cast12, Dutt13, Mich14, Barj14, Whit12]).

1.2 Μοντέλα που λαμβάνουν υπόψιν τους το χρόνο

Για να ξεπεραστούν οι δυσκολίες που περιγράφηκαν παραπάνω, διάφορα μοντέλα έχουν προταθεί για την αναπαράσταση χρονικά εξαρτημένων δικτύων (δείτε πχ το [Cast12] και τις αναφορές του). Ανεξάρτητα από τη συγκεκριμένη παραλλαγή, τα κύρια ερευνητικά ερωτήματα σχετίζονται συνήθως με τη βελτιστοποίηση ή την εκμετάλλευση της συνεκτικότητας χρονικά μεταβαλλόμενων δικτύων, ή με τον υπολογισμό σύντομων μονοπατιούν που τηρούν τους χρονικούς περιορισμούς. (δείτε πχ [Akri15, Akri16, Berm96, Erle15, Fosc14, Kemp02, Mert13]). Άλλες ερωτήσεις αφορούν υπολογισμούς χρονικά εξαρτημένων ροών (δείτε πχ [Flei07, Flei98]).

Στο [Berm96] πρωτο-εισήχθη η έννοια των *προγραμματισμένων δικτύων* έτσι ώστε να λάβουν υπόψιν προγραμματισμένες αλλαγές στη διαθεσιμότητα των ακμών ενός κατευθυνόμενου πολυγραφήματος. Για να κωδικοποιήσουν τις χρονικές στιγμές κατά τις οποίες μια ακμή είναι διαθέσιμη, αντιστοίχισαν σε κάθε ακμή ένα διάστημα πραγματικών αριθμών: Το χρονικό διάστημα κατά τη διάρκεια του οποίου η ακμή είναι διαθέσιμη.

Στη συνέχεια, στο [Kemp02] ορίστηκε το ελαφρά διαφορετικό μοντέλο των *χρονικά μεταβαλλόμενων γραφημάτων* με διακριτές ετικέτες. Μπορεί κανείς να σκεφτεί αυτό το μοντέλο ως μια ακολουθία κλασικών μη κατευθυνόμενων γραφημάτων, κάθε ένα από τα οποία αντανακλά τη δομή (διαθεσιμότητα ακμών) του δικτύου, κατά τη διάρκεια μίας χρονικής στιγμής. Πιο συγκεκριμένα, μπορεί να θεωρηθεί ως ένα συνήθες μη κατευθυνόμενο γράφημα, στο οποίο σε κάθε ακμή αντιστοιχεί μία χρονική ετικέτα: Η στιμή κατά την οποία αυτή η ακμή είναι διαθέσιμη. Ένα σημαντικό πλεονέκτημα αυτού του μοντέλου είναι ότι γενικεύει άμεσα τα παραδοσιακά γραφήματα ως χρονικά μεταβαλλόμενα δίκτυα των οποίων όλες οι ακμήες έχουν την ίδια χρονική ετικέτα (δηλ. όλες οι ακμές είναι διαθέσιμες σε μία μοναδική χρονική στιγμή, η οποία είναι η ίδια για όλες τις ακμές).

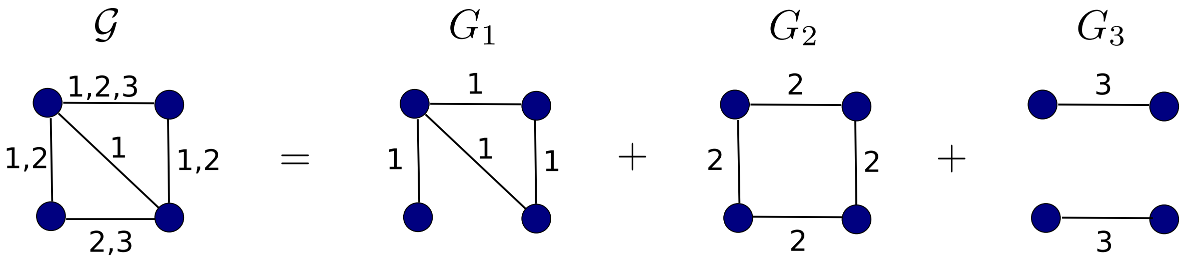
Στο [Mert13] τότε πραγματοποιείται μια περαιτέρω γενίκευση του μοντέλου των χρονικά μεταβαλλόμενων γραφημάτων, επιτρέποντας την εισαγωγή πολλαπλών ετικετών για κάθε ακμή. Αυτό είναι πολύ φυσικό καθώς μία ακμή μπορεί να είναι διαθέσιμη σε περισσότερες από μία χρονικές στιγμές. Είναι εύκολο να δει κανείς ότι αυτό το μοντέλο μπορεί εύκολα να εκφράσει το μοντέλο του [Berm96]. Για παράδειγμα, η έννοια της κατεύθυνσης μπορεί εύκολα να εκφραστεί σε αυτό το μοντέλο ως ένα μονοπάτι μήκους 2, στο οποίο κάθε ακμή έχει διαφορετική ετικέτα, έτσι ώστε να μπορεί να προσπελαστεί μόνο προς τη μία κατεύθυνση. Τα συνεχή διαστήματα μπορούν να αναπαρασταθούν ως σύνολα ακεραίων, καθώς αρκεί να κινούμαστε σε ακέραιες χρονικές στιγμές.

Στο παράδειγμα με το πρόγραμμα των αεροπλάνων, οι κορυφές του χρονικά μεταβαλλόμενου γραφήματος αντιστοιχούν σε αεροδρόμια και οι ακμές σε προγραμματισμένες πτήσεις μεταξύ ζευγαριών αεροδρομίων. Οι ετικέτες σε κάθε ακμή είναι ακριβώς οι χρονικές στιγμές κατά τις οποίες συγκεκριμένες πτήσεις είναι διαθέσιμες. Για το παράδειγμα της διάδοσης πληροφοριών, οι ακμές αντιστοιχούν στη στιγμή κατά την οποία λαμβάνει χώρα η επικοινωνία μεταξύ των δύο οντοτήτων.

Σε αυτή τη διπλωματική εργασία, θα υιοθετήσουμε το απλό και φυσικό μοντέλο των χρονικά μεταβαλλόμενων γραφημάτων με διακριτές χρονικές ετικέτες [**?**] (και την επέκτασή του με πολλαπλές ετικέτες ανά ακμή [Mert13]).

Μία από τις βασικότερες ερωτήσεις που απαντούμε εδώ αφορά το *μέγεθος των χρονικά συνεκτικών πιστοποιητικών*. Δοθέντος ενός συνεκτικού χρονικά μεταβαλλόμενου γραφήματος. είναι φυσικό κανείς να θέλει να αραιώσει αυτό το δίκτυο, διατηρώντας τη χρονική συνεκτικότητα. Αυτό είναι σημαντικό διότι σε πολλές εφαρμογές, το μεγάλο πλήθος συνδέσεων συνεπάγεται μεγάλο κόστος στο σχεδιασμό και τη χρήση του δικτύου. Συνεπώς είναι φυσικό να αναζητούμε δίκτυα τα οποία είναι φθηνά (έχουν λίγες συνδέσεις) και συνεκτικά. Η ερώτηση λοιπόν, είναι, μπορούμε για κάθε χρονικά μεταβαλλόμενο δίκτυο να διατηρήσουμε τη συνεκτικότητα, κρατώντας μόνο λίγες συνδέσεις και διαγράφοντας όλες τις άλλες; Αυτή η ερώτηση τέθηκε πρώτη φορά στο [Kemp02]. Στο Κεφάλαιο 2 θα δούμε ότι η απάντηση σε αυτή την ερώτηση είναι αρνητική. Αυτό σημαίνει ότι υπάρχουν χρονικά μεταβαλλόμενα δίκτυα στα οποία πρέπει κάποιος να διατηρήσει σχεδόν όλες τις συνδέσεις για να διατηρήσει τη συνεκτικότητα. Αυτό το γεγονός έρχεται σε αντίθεση με τη συνήθη διαίσθησή μας για τα παραδοσιακά (κατευθυνόμενα και μη) γραφήματα, στα οποία μόνο λίγες συνδέσεις είναι αναγκαίες. Συμπεραίνουμε λοιπόν ότι οι πληροφορίες συνεκτικότητας που κωδικοποιούνται στα χρονικά μεταβαλλόμενα γραφήματα είναι πιο πολύπλοκες από τα παραδοσιακά γραφήματα.

Για τους παραπάνω λόγους, η ερώτηση του υπολογισμού σχεδόν βέλτιστων πιστοποιητικών συνεκτικότητας για χρονικά μεταβαλλόμενα γραφήματα είναι ενδιαφέρουσα. Θα θεωρήσουμε δύο διαφορετικά ήδη συνεκτικότητας τα οποία οδηγούν σε δύο διαφορετικά προβλήματα βελτιστοποίησης: Συνεκτικότητα *μοναδικής πηγής*, όπου κάθε κορυφή πρέπει να είναι προσβάσιμη από μία συγκεκριμένη πηγή, και συνεκτικότητα *όλων των ζευγαριών*, όπου κάθε κορυφή πρέπει να είναι προσβάσιμη από κάθε άλλη κορυφή. Δείχνουμε ότι για τη γενική περίπτωση με βεβαρυμένες ακμές, υπάρχει προσεγγιστικός αλγόριθμος, εν τούτοις με κακό λόγο προσέγγισης των βέλτιστων πιστοποιητικών. Για να το δικαιολογήσουμε αυτό, δείχνουμε ότι το να επιτύχουμε σημαντικά καλύτερο λόγο προσέγγισης για αυτά τα προβλήματα είναι υπολογιστικά ανέφικτο. Συνεπώς στη συνέχεια περιορίζουμε τη δομή του δικτύου έτσι ώστε να μπορούμε να αποκτήσουμε γρήγορους αλγορίθμους που προσεγγίζουν τη βέλτιστη λύση καλύτερα. Θεωρούμε, λοιπόν, διάφορες συγκεκριμένες οικογένειες γραφημάτων και παρουσιάζουμε αλγορίθμους που είτε λύνουν ακριβώς το πρόβλημα, ή πετυχαίνουν καλό λόγο προσέγγισης. Το συμπέρασμα είναι ότι για γενικά γραφήματα το πρόβλημα του σχεδιασμού χρονικά μεταβαλλόμενων δικτύων μικρού κόστους κάτω από περιορισμούς συνεκτικότητας είναι δύσκολο, αλλά αν το γράφημα έχει συγκεκριμένη δομή μπορεί κανείς μερικές φορές να πάρει αποτελέσματα



Σχήμα 1.1: Ένα χρονικά μεταβαλλόμενο γράφημα $\mathcal{G}$ με διάρκεια ζωής 3 αποτελείται από 3 συνήθη μη κατευθυνόμενα γραφήματα G_1, G_2, G_3, ένα για κάθε μία ετικέτα στο $[3]$.

κοντά στο βέλτιστο.

1.3 Βασικοί Ορισμοί

1.3.1 Χρονικά Μεταβαλλόμενα Γραφήματα

Ένα *χρονικά μεταβαλλόμενο γράφημα* ορίζεται σε ένα χρονικά αμετάβλητο σύνολο n κορυφών. Κάθε (μη κατευθυνόμενη) ακμή e συσχετίζετα με ένα σύνολο διακριτών χρονικών ετικετών κάθε μία από τις οποίες δηλώνει πότε η e είναι διαθέσιμη. Αν κάθε ακμή είναι συσχετισμένη με μία μοναδική ετικέτα, όπως στο [Kemp02], το χρονικά μεταβαλλόμενο γράφημα λέγεται *απλό*. Μία ακμή e διαθέσιμη τη στιγμή t συγκροτεί τη χρονική ακμή (e, t) και υπάρχει ένα μη αρνητικό κόστος $w(e, t)$ που συνδέεται με αυτήν.

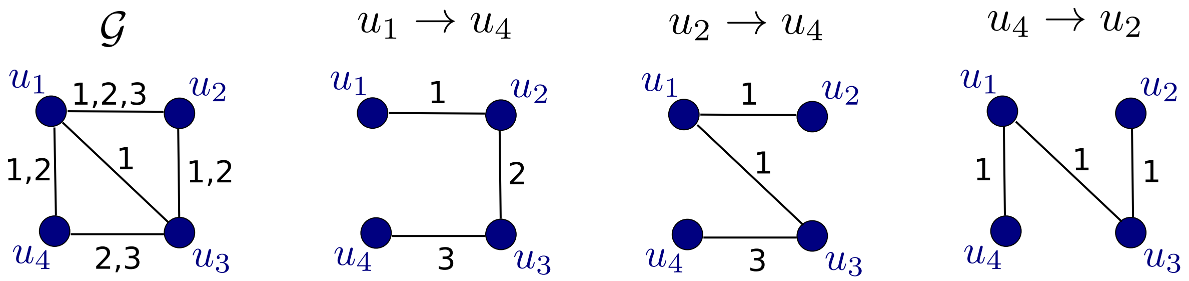
Ορισμός 1.3.1 (Χρονικά Μεταβαλλόμενα Γραφήματα)**.** Ένα *χρονικά μεταβαλλόμενο γράφημα* $\mathcal{G}(V, E, L)$ με σύνολο κορυφών V, σύνολο ακμών E και *διάρκεια ζωής* L είναι μία ακολουθία (μη κατευθυνόμενων ακμοβεβαρυμένων) γραφημάτων $(G_t(V, E_t, w_t))_{t\in[L]}$, όπου $E_t \subseteq E$ είναι το σύνολο των ακμών που είναι διαθέσιμες τη στιγμή t και $w_t(e)$ (ή $w(e, t)$) είναι το μη αρνητικό βάρος της καθεμιάς ακμής $e \in E_t$ τη χρονική στιγμή t. Συχνά γράφουμε $\mathcal{G}$ ή $\mathcal{G}(V, E)$ για συντομία. Ένα χρονικά μεταβαλλόμενο γράφημα $\mathcal{G}$ είναι *μη βεβαρυμένο*, εάν $w(e, t) = 1$ για κάθε $e \in E_t$ και για κάθε $t \in [L]$. Για κάθε ακμή $e \in E_t$, μπορούμε να πούμε ότι η (e, t) είναι μία *χρονική ακμή* του $\mathcal{G}$. Για κάθε ακμή $e \in E$, το $L_e = \{t \in [L] : e \in E_t\}$ συμβολίζει το σύνολο χρονικών στιγμών (ή *χρονικών ετικετών*) κατά τις οποίες η e είναι διαθέσιμη. Ένα χρονικά μεταβαλλόμενο γράφημα είναι *απλό* εάν $|L_e| = 1$ για κάθε ακμή $e \in E$. Δείτε επίσης το Σχήμα 1.1 για ένα παράδειγμα χρονικά μεταβαλλόμενου γραφήματος με διάρκεια ζωής 3, καθώς και τα 3 συνήθη γραφήματα από τα οποία αποτελείται.

Το χρονικό ανάλογο ενός συνηθισμένου μονοπατιού ονομάζεται *χρονικό μονοπάτι*. Ένα χρονικό μονοπάτι είναι σαν ένα συνηθισμένο μονοπάτι, αλλά επιπλέον τηρεί τους περιορισμούς χρονικής διαθεσιμότητας στις ακμές. Είναι ένα συνηθισμένο μονοπάτι του οποίου οι ακμές είναι εφοδιασμένες με αύξουσες χρονικές ετικέτες κατά μήκος του μονοπατιού.

Ορισμός 1.3.2 (Χρονικά Μονοπάτια)**.** Ένα *χρονικό* μονοπάτι είναι μία εναλλασσόμενη ακολουθία $(v_1, (e_1, t_1), v_2, (e_2, t_2), \ldots, v_k, (e_k, t_k), v_{k+1})$ από κορυφές v_i και χρονικές ακμές (e_i, t_i), έτσι ώστε $e_i \equiv \{v_i, v_{i+1}\} \in E_{t_i}$, για κάθε $i \in [k]$, και $1 \leq t_1 \leq t_2 \leq \cdots \leq t_k$. Ένα χρονικό μονοπάτι είναι *αυστηρό* εάν $t_1 < t_2 < \cdots < t_k$. Ένα τέτοιο χρονικό μονοπάτι είναι από το v_1 στο v_{k+1} (ή αλλιώς ένα χρονικό $v_1 - v_{k+1}$ μονοπάτι). Δείτε επίσης το Σχήμα 1.2 για ένα παράδειγμα χρονικά μεταβαλλόμενου γραφήματος, καθώς και τρία χρονικά μονοπάτια του.

Συνήθως θα ονομάζουμε n το πλήθος των κορυφών και $M = \sum_e |L_e|$ το πλήθος των χρονικών ακμών του $\mathcal{G}$.

Ορισμός 1.3.3 (Υποκείμενο Γράφημα)**.** Το (στατικό) γράφημα $G(V, E)$ είναι το *υποκείμενο γράφημα* του $\mathcal{G}(V, E, w)$. Θα λέμε ότι το $\mathcal{G}$ έχει κάποια (μη χρονική) γραφοθεωρητική ιδιότητα (π.χ. είναι



Σχήμα 1.2: Ένα χρονικά μεταβαλλόμενο γράφημα $\mathcal{G}$ και τρία χρονικά μονοπάτια σε αυτό: Ένα αυστηρό χρονικό μονοπάτι από το u_1 στο u_4 και δύο μη αυστηρά από το u_2 στο u_4 και από το u_4 στο u_2 αντίστοιχα.

δένδρο, κύκλος, κλίκα, έχει φραγμένο δενδροπλάτος) αν το υποκείμενο γράφημα G έχει αυτή την ιδιότητα. Αν το υποκείμενο γράφημα του $\mathcal{G}$ είναι σε μία συγκεκριμένη οικογένεια γραφημάτων X, θα λέμε ότι το $\mathcal{G}$ είναι ένα χρονικό X. Για παράδειγμα, αν το υποκείμενο γράφημα του $\mathcal{G}$ είναι ένα δένδρο, τότε το $\mathcal{G}$ είναι ένα χρονικό δενδρο. Για το σύνολο κορυφών S, το $G[S]$ (αντίστοιχα $\mathcal{G}[S]$) είναι το υποκείμενο (αντ. χρονικά μεταβαλλόμενο) γράφημα που ενάγεται από το S.

Ορισμός 1.3.4 (Συνδετικά Υπογραφήματα). Ένα *(χρονικά) συνδετικό υπογράφημα* $\mathcal{G}'$ ενός χρονικά μεταβαλλόμενου γραφήματος $\mathcal{G} \equiv (G_t(V, E_t, w_t))_{t\in[L]}$ είναι μια ακολουθία γραφημάτων $(G'_t(V, E'_t, w_t))_{t\in[L]}$ έτσι ώστε $E'_t \subseteq E_t$. Το συνολικό βάρος του $\mathcal{G}'$ είναι $\sum_{t\in[L]} \sum_{e\in E'_t} w(e, t)$.

Ορισμός 1.3.5 (MTCS). Ένα *Ελάχιστο Χρονικά Συνεκτικό Υπογράφημα* (MTCS) ενός χρονικά μεταβαλλόμενου γραφήματος $\mathcal{G}$ είναι ένα συνδετικό υπογράφημα του $\mathcal{G}$ το οποίο έχει το ελάχιστο πλήθος ακμών. Ισοδύναμα, θα το λέμε επίσης και *Ελάχιστο Πιστοποιητικό Συνεκτικότητας*.

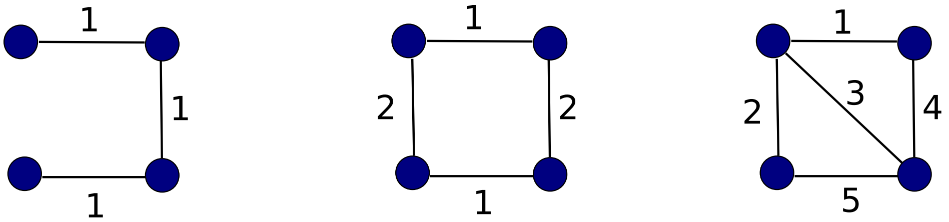
1.3.2 Χρονική Συνεκτικότητα

Δοθείσας μίας κορυφής r, ένα χρονικά μεταβαλλόμενο γράφημα είναι (χρονικά) *r-συνεκτικό* αν υπάρχει χρονικό μονοπάτι από το r προς οποιαδήποτε άλλη κορυφή. Ένα χρονικά μεταβαλλόμενο γράφημα είναι (χρονικά) *συνεκτικό* αν υπάρχει χρονικό μονοπάτι από οποιαδήποτε κορυφή προς οποιαδήποτε άλλη. Αν όλα τα χρονικά μονοπάτια είναι αυστηρά, το $\mathcal{G}$ είναι αυστηρά συνεκτικό (αντίστοιχα αυστηρά r-συνεκτικό). Ένα *πιστοποιητικό (r-)συνεκτικότητας* του $\mathcal{G}$ είναι ένα συνδετικό υπογράφημα του $\mathcal{G}$ που είναι επίσης (r-)συνεκτικό.

Μελετάμε την ύπαρξη πυκνών χρονικά μεταβαλλόμενων γραφημάτων που είναι ελαχιστικά συνεκτικά και τα προβλήματα του υπολογισμού ενός υποσυνόλου χρονικών ακμών ελαχίστου κόστους που διατηρούν είτε την r-συνεκτικότητα είτε τη συνεκτικότητα. Θα αναφερόμαστε σε αυτά τα προβλήματα βελτιστοποίησης ως (Ελάχιστη) r-συνεκτικότητα μίας πηγής (r-MTC) και (Ελάχιστη) συνεκτικότητα όλων των ζευγαριών (MTC). Εμφανίζονται ως φυσικές γενικεύσεις του Ελάχιστου Συνδετικού Δένδρου και του Ελάχιστου Συνδετικού Κατευθυνόμενου Δένδρου αντίστοιχα.

Ορισμός 1.3.6 (Συνεκτικότητα Μίας Πηγής). Δοθέντος ενός χρονικά μεταβαλλόμενου γραφήματος $\mathcal{G}$ και μίας κορυφής-πηγής r, το πρόβλημα της ελάχιστης Χρονικής Συνεκτικότητας μίας πηγής (r-MTC) είναι να υπολογιστεί ένα χρονικά r-συνεκτικό συνδετικό υπογράφημα του $\mathcal{G}$ με ελάχιστο συνολικό βάρος. Η βέλτιστη λύση στο r-MTC είναι ένα απλό χρονικά μεταβαλλόμενο γράφημα, το υποκείμενο γράφημα του οποίου είναι ένα δένδρο (δείτε [Kemp02, Section 6] και Lemma 2.1.1).

Ενώ για το r-MTC η βέλτιστη λύση είναι πάντα δένδρο, όπως θα αποδείξουμε στην Ενότητα 2.1, για το MTC αυτό δεν είναι καθόλου κοντά στην αλήθεια. Στην πραγματικότητα, το μέγεθος του ελάχιστου πιστοποιητικού συνεκτικότητας δεν είναι συνάρτηση του n, αλλά εξαρτάται επίσης και από τη δομή του χρονικά μεταβαλλόμενου γραφήματος. Αυτό φαίνεται στο Σχήμα 1.3, όπου παρουσιάζουμε



Σχήμα 1.3: Τρία χρονικά μεταβαλλόμενα γραφήματα με 4 κορυφές, τα οποία έχουν μεγέθη ελάχιστων πιστοποιητικών συνεκτικότητας 3, 4 και 5 αντίστοιχα.

τρία απλά και ελαχιστικά συνεκτικά γραφήματα με 4 ακμές, κάθε ένα από τα οποία έχει διαφορετικό πλήθος ακμών, οπότε και διαφορετικό μέγεθος πιστοποιητικού συνεκτικότητας.

Ορισμός 1.3.7 (Χρονική Συνεκτικότητα όλων των ζευγαριών)**.** Δοθέντος ενός χρονικά μεταβαλλόμενου γραφήματος $\mathcal{G}$, το πρόβλημα της (Ελάχιστης) *Χρονικής Συνεκτικότητας όλων των ζευγαριών* (MTC) είναι να υπολογιστεί ένα χρονικά συνεκτικό συνδετικό υπογράφημα του $\mathcal{G}$ με ελάχιστο βάρος.

1.3.3 Προσεγγιστικοί Αλγόριθμοι

Ένας προσεγγιστικός αλγόριθμος είναι ένας αλγόριθμος η έξοδος του οποίου είναι εγγυημένα μία προσέγγιση της βέλτιστης απάντησης στο πρόβλημα κατά ένα πολλαπλασιαστικό παράγοντα. Ο παράγοντας αυτός λέγεται *λόγος προσέγγισης* του αλγορίθμου.

Ορισμός 1.3.8 (Λόγος προσέγγισης)**.** Ένας αλγόριθμος A έχει *λόγο προσέγγισης* $\rho \geq 1$ για ένα πρόβλημα ελαχιστοποίησης αν για κάθε παράδειγμα εισόδου I, το κόστος του A στο I είναι το πολύ ρ φορές το βέλτιστο κόστος του I.

1.4 Συνεισφορά

Σε αυτή τη δουλειά, πρώτα μελετάμε το μέγεθος των πιστοποιητικών συνεκτικότητας για χρονικά μεταβαλλόμενα γραφήματα. Όπως έχουμε αναφέρει πιο πριν, αυτή είναι μία παράμετρος που είναι πολύ σημαντικό να έχει μικρή τιμή όσον αφορά το σχεδιασμό χρονικά μεταβαλλόμενων δικτύων. Για να εκτιμήσουμε τη χειρότερη δυνατή περίπτωση για γενικά χρονικά μεταβαλλόμενα γραφήματα, θέλουμε να φράξουμε το μέγιστο δυνατό πιστοποιητικό συνεκτικότητας πάνω σε όλα τα χρονικά μεταβαλλόμενα γραφήματα. Αυτό θα μας δώσει το χειρότερο που μπορούμε να αναμένουμε από τα χρονικά μεταβαλλόμενα γραφήματα γενικά. Για την r-συνεκτικότητα, είναι τετριμμένο να δει κανείς ότι το μέγεθος είναι πάντοτε $n-1$, αφού όπως θα αποδείξουμε στην Ενότητα 2.1 ένα πιστοποιητικό r-συνεκτικότητας είναι πάντοτε δένδρο. Για τη συνεκτικότητα όλων των ζευγαριών, από την άλλη, τα πράγματα είναι πολύ πιο περίπλοκα. Όπως παρατηρήθηκε πρώτα στο [Kemp02], το μέγεθος του πιστοποιητικού συνεκτικότητας δεν είναι συνάρτηση μόνο του πλήθους των κορυφών n, αλλά εξαρτάται και από τη δομή του χρονικά συνεκτικού γραφήματος. Έθεσαν, λοιπόν, το ερώτημα, ποιά είναι η πιο ”σφιχτή” συνάρτηση $c(n)$ έτσι ώστε όλα τα απλά χρονικά μεταβαλλόμενα γραφήματα με n κορυφές να έχουν πιστοποιητικό συνεκτικότητας με μέγεθος το πολύ $c(n)$; Στην Ενότητα 2.3 θα δείξουμε την απόδειξή τους, που δείχνει ότι $c(n) = \Omega(n \log n)$. Οι [Akri15] δείχνουν ότι αυτό το κάτω φράγμα ισχύει ακόμη και αν περιοριστούμε σε διαφορετικές χρονικές ετικέτες. Επιπλέον, είναι εύκολο να δούμε ότι $c(n) = O(n^2)$, το οποίο θα αποδείξουμε στην Ενότητα 2.2. Παρατηρήστε ότι για τη συνεκτικότητα με αυστηρά χρονικά μονοπάτια η απάντηση είναι τετριμμένα $\Theta(n^2)$, αφού μπορούμε να πάρουμε το πλήρες γράφημα με την ίδια χρονική ετικέτα σε κάθε ακμή. Σε αυτή τη διπλωματική κλείνουμε το μεγάλο κενό μεταξύ $\Omega(n \log n)$ και $O(n^2)$, δείχνοντας ότι $g(n) = \Theta(n^2)$ στην Ενότητα 2.4. Για να το κάνουμε αυτό, κατασκευάζουμε μία άπειρη οικογένεια χρονικά συνεκτικών γραφημάτων και αποδεικνύουμε ότι για αυτή την οικογένεια το να κρατήσει κανείς λιγότερο πλήθος ακμών από τετραγωνικό έχει ως αποτέλεσμα το να μην είναι χρονικά συνεκτικό το νέο γράφημα. Δεν είναι δύσκολο να τροποποιήσουμ την κατασκευή μας έτσι ώστε να έχει όλες τις ετικέτες διαφορετικές, οπότε το αποτέλεσμά μας βελτιώνει όλα τα προηγούμενα αποτελέσματα και αποκαθιστά ότι $c(n) = \Theta(n^2)$. Επιπλέον, η κατασκευή μας είναι επίσης βέλτιστη και όσον αφορά τη διάρκεια ζωής, αφού τα χρονικά μεταβαλλόμενα γραφήματα που κατασκευάζουμε έχουν γραμμική διάρκεια ζωής, και είναι εύκολο να δει κανείς ότι με υπογραμμική διάρκεια ζωής τότε υπάρχει υποτετραγωνικό πιστοποιητικό συνεκτικότητας. Επιπλέον, η κατασκευή μας μπορεί να προσαρμοστεί έτσι ώστε να δειχθεί ένα πιο γενικό αποτέλεσμα, ότι για κάθε διάρκεια ζωής $L \leq n$, τα απλά χρονικά μεταβαλλόμενα γραφήματα με διάρκεια ζωής το πολύ L έχουν $c(n) = \Theta(nL)$.

Στο υπόλοιπο αυτής της εργασίας παράγουμε άνω και κάτω φράγματα στην προσεγγιστικότητα του προβλήματος της Συνεκτικότητας (μίας πηγής και όλων των ζευγαριών). Δεδομένου του μεγάλου χάσματος στο μέγεθος των πιστοποιητικών χρονικής συνεκτικότητας, είναι φυσικό να αναρωτηθεί κανείς για την πολυπλοκότητα και την προσεγγισιμότητα αυτών των προβλημάτων. Προηγούμενες δουλειές δείχνουν ότι μπορούμε να αποφασίσουμε αν ένα χρονικά μεταβαλλόμενο γράφημα είναι συνεκτικό σε πολυωνυμικό χρόνο (πχ [Akri15, Berm96, Kemp02]) και επίσης ότι η Χρονική Συνεκτικότητα μίας πηγής μπορεί να λυθεί σε πολυωνυμικό χρόνο στη μη βεβαρυμένη περίπτωση. Μία άλλη ενδιαφέρουσα παρατήρηση είναι ότι αν χρησιμοποιήσουμε τη χρονικά επεκτεταμένη έκδοση ενός χρονικά μεταβαλλόμενου γραφήματος για την Ελάχιστη Χρονική Συνεκτικότητα, τα προβλήματα που προκύπτουν είναι πολύ παρόμοια με προβλήματα Group Steiner Tree. Στην πραγματικότητα, αυτή η παρατήρηση είναι η κύρια διαίσθηση πίσω από αρκετά από τα αποτελέσματά μας.

Στο Κεφάλαιο 3, θεωρούμε το πρόβλημα βελτιστοποίησης r-MTC. Το βασικό που δείχνουμε είναι ότι αυτό το πρόβλημα έχει μεγάλη σχέση με ένα κλασικό γραφοθεωρητικό πρόβλημα, το Directed Steiner Tree. Συγκεκριμένα, εκμεταλλευόμενοι ομοιότητες μεταξύ της κατεύθυνσης ακμών και ζευγαριών χρονικών ακμών, παρουσιάζουμε μία αναγωγή από το r-MTC στο Directed Steiner Tree (Θεώρημα 3.1.1) και δείχνουμε ότι το r-MTC δεν μπορεί να προσεγγιστεί με λόγο προσέγγισης $O(\log^2 2 - \varepsilon n)$, για κάθε σταθερά $\varepsilon > 0$, εκτός εάν NP $\subseteq$ ZTIME($n^{\text{poly}\log n}$). Ο μετασχηματισμός μας συνεπάγεται επίσης ότι οποιαδήποτε $o(n^\varepsilon)$ προσέγγιση για το r-MTC θα βελτίωνε τον καλύτερο γνωστό λόγο προσέγγισης του εκτενώς μελετημένου Directed Steiner Tree. Στη θετική πλευρά, δείχνοντας ένα μετασχηματισμό από το r-MTC στο Directed Steiner Tree, και σε συνδυασμό με τον αλγόριθμο του [Char99], λαμβάνουμε μία $O(n^\varepsilon)$ προσέγγιση σε πολυωνυμικό χρόνο, για κάθε σταθερά $\varepsilon > 0$, και μία $O(\log^3 n)$ προσέγγιση σε οιονεί πολυωνυμικό χρόνο ($O(2^{\text{poly}\log n})$) για το r-MTC (Θεώρημα 3.2.1). Επίσης δείχνουμε ότι το r-MTC μπορεί να λυθεί σε πολυωνυμικό χρόνο, εάν το υποκείμενο γράφημα έχει φραγμένο δενδροπλάτος (Θεώρημα 3.3.1). Ανεξάρτητα, οι [Huan15] επίσης παρουσίασαν έναν αλγόριθμο για το r-MTC όπως επίσης και αποτελέσματα δυσκολίας προσέγγισης για αυτό, αν και σημαντικά ασθενέστερα από τα δικά μας. Συγκεκριμένα, παρουσιάζουν μία αναγωγή από το r-MTC στο Directed Steiner Tree παρόμοια με τη δική μας και λαμβάνουν τις ίδιες εγγυήσεις προσέγγισης για το r-MTC. Επιπλέον, παρουσιάζουν αναγωγές από NP-complete προβλήματα στο r-MTC, και έτσι δείχνουν ότι το r-MTC είναι APX-hard (δεν έχει προσεγγιστικό σχήμα πολυωνυμικού χρόνου εκτός αν $P = NP$). Όμως, οι αναγωγές που χρησιμοποιούν (προβλήματα διαφορετικά από το Directed Steiner Tree) είναι πολύ πιο ασθενείς από τις δικές μας, όπως επιβεβαιώνεται από την αναγωγή μας από το Directed Steiner Tree που δείχνει πολυλογαριθμική μη προσεγγισιμότητα.

Στο Κεφάλαιο 4, μελετάμε την προσεγγισιμότητα του προβλήματος της Χρονικής Συνεκτικότητας όλων των ζευγαριών (MTC). Το Θεώρημα 3.2.1 έχει ως συνέπεια μία $O(n^{1+\varepsilon})$ προσέγγιση για το MTC (Πόρισμα 4.1.1). Μία αναγωγή που διατηρεί την προσεγγισιμότητα στο Directed Steiner Forest και το [Feld12, Theorem 1.1] έχουν ως συνέπεια μία $O((\Delta M)^{2/3+\varepsilon})$ προσέγγιση για το MTC, όπου M είναι το πλήθος των χρονικών ακμών και Δ είναι ο μέγιστος βαθμός του υποκείμενου γραφήματος (Θεώρημα 4.1.2). Αν το M είναι σχεδόν γραμμικό και το Δ είναι πολυλογαριθμικό, λαμβάνουμε μία $O(n^{2/3+\varepsilon})$ προσέγγιση. Στα αρνητικά αποτελέσματα, μία αναγωγή από το Symmetric Label Cover έχει ως συνέπεια ότι το MTC δεν μπορεί να προσεγγιστεί με λόγο $O(2^{\log^{1-\varepsilon}} n)$ εκτός εάν NP $\subseteq$ DTIME($n^{\text{poly}\log n}$) (Θεώρημα 4.2.1, δείτε επίσης την [Dodi99, Ενότητα 4]). Επίσης δείχνουμε ότι η μη βεβαρυμένη έκδοση του MTC είναι APX-hard (Θεώρημα **??**). Ένα αποτέλεσμα κάπως παρόμοιο με αυτό βρίσκεται στο [Akri15]. Συγκεκριμένα, οι συγγραφείς αλλάζουν την αντικειμενική συνάρτηση μελετώντας το πρόβλημα της *διαγραφής* του μέγιστου πλήθους χρονικών ακμών, διατηρώντας ταυτόχρονα τη χρονική συνεκτικότητα. Η μόνη σημαντική διαφορά μεταξύ αυτού του προβλήματος και του μη βεβαρυμένου r-MTC είναι ότι η αντικειμενική συνάρτηση μετράει πόσες ακμές διαγράφηκαν, και όχι πόσες έμειναν. Οι συγγραφείς δείχνουν ότι αυτό το αλλαγμένο πρόβλημα είναι APX-hard. Παρ' όλα αυτά, η μόνη συνέπεια που έχει αυτό το αποτέλεσμα στο r-MTC είναι το γεγονός ότι είναι NP-hard, ενώ εμείς παρουσιάζουμε ένα πιο ισχυρό αποτέλεσμα APX-hardness. Ο λόγος είναι ότι τα χρονικά μεταβαλλόμενα γραφήματα που χρησιμοποιούν ως δύσκολες περιπτώσεις στην αναγωγή τους είναι εύκολες περιπτώσεις για το δικό μας πρόβλημα.

Στο Κεφάλαιο 5, δείχνουμε ότι το MTC μπορεί να λυθεί βέλτιστα, σε πολυωνυμικό χρόνο, αν το υποκείμενο γράφημα είναι δένδρο (Θεώρημα 5.1.1), και ότι το MTC είναι 2-προσεγγίσιμο αν το υποκείμενο γράφημα είναι κύκλος (Θεώρημα 5.2.1), αλλά είναι παραμένει ανοικτό ερώτημα το αν το MTC είναι NP-hard για κύκλους.

Για λόγους σαφήνειας, θα επικεντρωθούμε σε συνεκτικότητα με μη αυστηρά χρονικά μονοπάτια. Παρ' όλα αυτά, όλα τα αποτελέσματά μας μπορούν να επεκταθούν (με μικρές αλλαγές στις αποδείξεις και με τις ίδιες εγγυήσεις προσεγγιστικότητας) στην περίπτωση της αυστηρής συνεκτικότητας.

Κεφάλαιο 2

Μέγεθος Ελάχιστων Χρονικά Συνεκτικών Υπογραφημάτων (MTCS)

Όπως έχουμε ήδη αναφέρει, η μελέτη των χρονικά μεταβαλλόμενων γραφημάτων έχει ως στόχο να συμπληρώσε το κενό που αφήνει η παραδοσιακή γραφοθεωρία στην εκφραστικότητα πραγματικών προβλημάτων που έχουν εγγενώς την παράμετρο του χρόνου. Έτσι, περιγράψαμε κάποιες γενικεύσεις γραφοθεωρητικών εννοιών σε χρονικά μεταβαλλόμενα γραφήματα.

2.1 Η δομή των λύσεων στο πρόβλημα της χρονικής συνεκτικότητας μίας πηγής

Το παρακάτω είναι ένα αποτέλεσμα που αφορά τη δομή των λύσεων του r-MTC. Το παρακάτω λήμμα υπονοείται στο [Kemp02, Section 6] και δείχνει ότι οποιαδήποτε εφικτή λύση στο r-MTC μπορεί να μετατραπεί σε ένα απλό χρονικό δένδρο χωρίς να αυξηθεί το συνολικό βάρος της λύσης. Έτσι, μπορούμε πάντα να υποθέτουμε ότι η βέλτιστη λύση στο r-MTC είναι ένα απλό χρονικό δένδρο.

***Λήμμα* 2.1.1.** *Δοθείσας μίας εφικτής λύσης $\mathcal{T}'$ για το r-MTC, μπορούμε να λάβουμε μία εφικτή λύση $\mathcal{T}$ έτσι ώστε (i) το $\mathcal{T}$ είναι ένα απλό χρονικό γράφημα, (ii) το συνολικό βάρος του $\mathcal{T}$ δεν ξεπερνά το συνολικό βάρος του $\mathcal{T}'$, και (iii) το σύνολο των ακμών στο $\mathcal{T}$ σχηματίζουν ένα δένδρο στο υποκείμενο γράφημα.*

2.2 Ένα άνω φράγμα στο μέγεθος χειρότερης περίπτωσης του MTCS

***Θεώρημα* 2.2.1.** *Δοθέντος ενός χρονικά μεταβαλλόμενου γραφήματος $\mathcal{G}$ με n κορυφές και διάρκεια ζωής L, υπάρχει ένα Χρονικά συνεκτικό υπογράφημα με $O(\min\{n^2, nL\})$ χρονικές ακμές.*

2.3 Ένα απλό κάτω φράγμα στο μέγεθος χειρότερης περίπτωσης του MTCS

Το παρακάτω θεώρημα είναι από το [Kemp02] και δείχνει ένα $\Theta(n\log n)$ κάτω φράγμα στο μέγεθος χειρότερης περίπτωσης του MTCS

***Θεώρημα* 2.3.1.** *Υπάρχει μία άπειρη οικογένεια χρονικά μεταβαλλόμενων γραφημάτων, για κάθε ένα από τα οποία το μέγεθος του ελάχιστου πιστοποιητικού είναι $n\frac{\log_2 n}{2}$.*

2.4 Ένα βέλτιστο κάτω φράγμα στο μέγεθος χειρότερης περίπτωσης του MTCS

Το αποτέλεσμα αυτής της ενότητας είναι η κατασκευή μίας οικογένειας χρονικά μεταβαλλόμενων γραφημάτων και η εύρεση του μεγέθους του MTCS του, δείχνοντας τελικά ότι για αυτή την οικογένεια είναι τετραγωνικό. Θα δουλέψουμε σε απλά χρονικά μεταβαλλόμενα γραφήματα, αφού κανένα απο τα αποτελέσματά μας δεν απαιτεί πολλαπλές ετικέτες για κάθε ακμή.

Στην Ενότητα 2.4.1 αποδεικνύουμε ένα σημαντικό λήμμα για την κατασκευή, που αφορά το πακετάρισμα μονοπατιών με γραμμικό μήκος σε ένα γραμμικό πλήθος κόμβων. Χρησιμοποιώντας αυτό, εισάγουμε τους κύριους ισχυρισμούς και την διαίσθηση στην Ενότητα 2.4.2. Λεπτομερής απόδειξη των ισχυρισμών δίνεται στις υπόλοιπες ενότητες.

2.4.1 Διαμερίζοντας ένα πλήρες γράφημα σε μονοπάτια Hamilton

Πρώτα δείχνουμε ότι για κάθε άρτιο $n \geq 2$, οι ακμές του πλήρους γραφήματος K_n μπορούν να διαμεριστούν σε $n/2$ μονοπάτια Hamilton, κάθε ένα από τα οποία επισκέπτεται τις κορυφές του K_n με διαφορετική σειρά.

***Λήμμα* 2.4.1.** *Για κάθε άρτιο $n \geq 2$, οι ακμές του πλήρους γραφήματος K_n μπορούν να διαμεριστούν σε $n/2$ μονοπάτια Hamilton.*

2.4.2 Κατασκευή

Τώρα είναι η ώρα για να περιγράψουμε την τελική κατασκευή λεπτομερώς.

***Θεώρημα* 2.4.2.** *Για κάθε άρτιο $n \geq 2$, υπάρχει ένα απλό συνεκτικό χρονικά μεταβαλλόμενο γράφημα με $3n$ κορυφές, $n(n+9)/2 - 3$ ακμές και διάρκεια ζωής το πολύ $7n/2$, έτσι ώστε η διαγραφή οποιουδήποτε υποσυνόλου $5n$ ακμών έχει ως αποτέλεσμα ένα μη συνεκτικό χρονικά μεταβαλλόμενο γράφημα.*

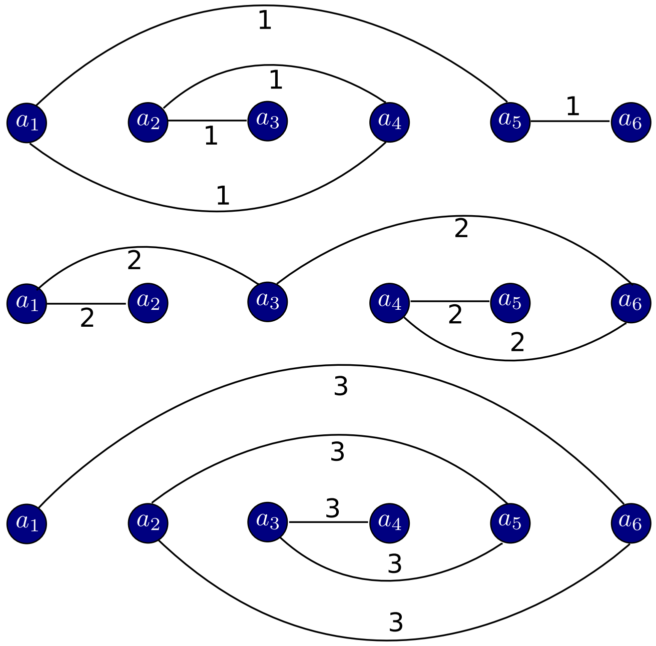
Σκιαγράφηση απόδειξης. Για κάθε άρτιο n, κατασκευάζουμε ένα απλό συνεκτικό χρονικά μεταβαλλόμενο γράφημα $\mathcal{G}$ με $\Theta(n)$ κορυφές και $\Theta(n^2)$ ακμές έτσι ώστε κάθε ακμή να είναι υποχρεωτική για την χρονική συνεκτικότητα.

Η Κατασκευή Για κάθε άρτιο n, το $\mathcal{G}$ αποτελείται από $3n$ κορυφές διαμερισμένες σε τρία σύνολα $A = \{a_1, \ldots, a_n\}$, $H = \{h_1, \ldots, h_n\}$, και $M = \{m_1, \ldots, m_n\}$, κάθε ένα από τα οποία έχει n κορυφές.

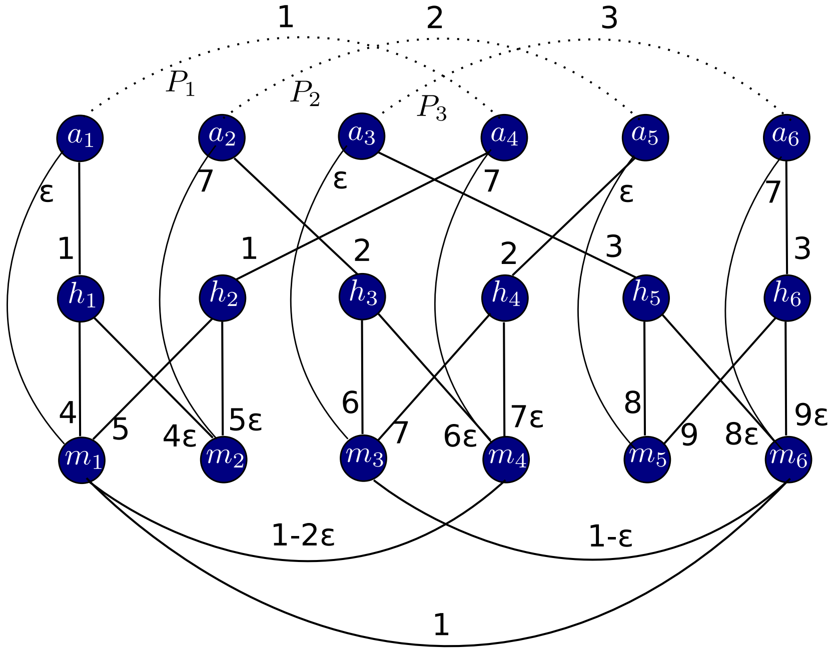
Το υποκείμενο γράφημα $G[A]$ είναι το πλήρες γράφημα K_n και αποτελεί το *πυκνό* μέρος της κατασκευής με $\Theta(n^2)$ ακμές. Οι ακμές του $G[A]$ είναι διαμερισμένες σε $n/2$ ακμοδιακεκριμένα μονοπάτια $p_1, \ldots, p_{n/2}$. Κάθε μονοπάτι p_i έχει μήκος $n-1$ και συνδέει όλες τις κορυφές του A (δείτε το Σχήμα 2.1 και το Λήμμα 2.4.1). Όλες οι ακμές του κάθε μονοπατιού p_i έχουν χρονική ετικέτα i.

Οι ακμές του H αποτελούν το *μεσαίο* κομμάτι της κατασκευής. Δεν υπάρχουν ακμές με και τα δύο άκρα τους στο H. Για κάθε $i \in [n/2]$, το ένα άκρο του μονοπατιού p_i είναι συνδεδεμένο στο h_{2i-1} και το άλλο είναι συνδεδεμένο στο h_{2i}. Και οι δύο ακμές έχουν χρονική ετικέτα i.

Οι κορυφές του M αποτελούν το κομμάτι *διασύνδεσης* της κατασκευής. Για κάθε $i \in [n/2]$, καλούμε το m_{2i-1} (αντ. m_{2i}) *κορυφή εισόδου* (αντ. *κορυφή εξόδου*) για τις κορυφές h_{2i-1} και h_{2i}. Υπάρχουν δύο ακμές που συνδέεουν το m_{2i-1} με τα h_{2i-1} και h_{2i} με ετικέτες $n/2+2i-1$ και $n/2+2i$, αντίστοιχα, και δύο ακμές που συνδέουν το m_{2i} με τα h_{2i-1} και h_{2i} με ετικέτες $(n/2+2i-1)\epsilon$ και $(n/2+2i)\epsilon$, αντίστοιχα, για κάποιο $\epsilon \in (0, 1/(4n))$. Επίσης συνδέουμε τις κορυφές του M μεταξύ τους. Για κάθε $i \in [n/2-2]$, υπάρχουν ακμές που συνδέουν το m_{2i-1} με το m_{2i+2} και με το m_n, και μία ακμή που συνδέει το m_{n-3} με το m_n. Για να μοιράσουμε ετικέτες σε αυτές τις ακμές, τις ταξινομούμε με φθίνουσα σειρά του άκρου τους με το μεγαλύτερο δείκτη, λύνοντας τις ισοπαλίες ταξινομόντας τους κατά αύξουσα σειρά των άκρων με το μικρότερο δείκτη, δηλαδή η σειρά είναι $\{m_1, m_n\}, \{m_3, m_n\}, \ldots, \{m_{n-3}, m_n\}, \{m_{n-5}, m_{n-2}\}, \{m_{n-7}, m_{n-4}\}, \ldots, \{m_1, m_4\}$. Η χρονική ετικέτα της k-οστής ακμής σε αυτή τη σειρά είναι $1-(k-1)\epsilon$. Τέλος, για κάθε $i \in [n/2]$, υπάρχει μία ακμή με χρονική ετικέτα ϵ που συνδέει την ακμή m_{2i-1} με την ακμή a_{2i-1} στο A και μία ακμή με χρονική ετικέτα $n+1$ που συνδέει την κορυφή m_{2i} με την κορυφή a_{2i} στο A (δείτε επίσης το Σχήμα 2.2).



Σχήμα 2.1: Διαμέριση του πλήρους γραφήματος με 6 ακμές σε 3 μονοπάτια Hamilton.



Σχήμα 2.2: Συνδυάζοντας τα τρία κομμάτια μαζί παίρνουμε την τελική κατασκευή για $n = 6$.

Το συνολικό πλήθος ακμών είναι $n(n+9)/2-3$, το πλήθος των διαφορετικών χρονικών ετικετών είναι το πολύ $7n/2$, και κάθε ακμή έχει μία ετικέτα (δείτε επίσης την Ενότητα **??**).

Διαίσθηση και κύριοι ισχυρισμοί Η κατασκευή βασίζεται στη συλλογή $p_1, \dots, p_{n/2}$ των $n/2$ ακμοδιακεκριμένων μονοπατιών, όπου όλε ςοι ακμές σε κάθε μονοπάτι p_i έχουν ετικέτα i. Επεκτείνοντας κάθε μονοπάτι p_i στις κορυφές h_{2i-1} και h_{2i}, παίρνουμε ένα μονοπάτι που συνδέει το h_{2i} με το h_{2i-1} (και αντιστρόφως) και με όλες τις κορυφές στο A τη στιγμή i. Επιπλέον, οι διαφορετικές ετικέτες κάνουν αυτά τα μονοπάτια ουσιαστικά ανεξάρτητα μεταξύ τους, με την έννοια ότι αν ένας χρονικός περίπατος ξεκινάει και τελειώνει τη στιγμή i, μπορεί να χρησιμοποιεί μόνο ακμές με ετικέτα i (δηλ.

μόνο ακμές αυτού του μονοπατιού) για να πάει από το h_{2i} στο h_{2i-1}. Συνεπώς, όλες οι ακμές του $G[A]$ πρέπει να υπάρχουν σε ένα χρονικά συνεκτικό συνδετικό υπογράφημα του $\mathcal{G}$. Για να πετύχουμε πυκνό υποκείμενο γράφημα $G[A]$, παρατηρούμε ότι η συλλογή των $n/2$ ακμοδιακεκριμένων μονοπατιών μπορεί να οριστεί έτσι ώστε αυτά να χρησιμοποιούν n κοινές κορυφές, κάθε ένα με διαφορετική σειρά (δείτε το Σχήμα 2.1 και το Λήμμα 2.4.1). Αυτό περιγράφει την κύρια διαίσθηση πίσω από την κατασκευή μας και εξηγεί πώς λειτουργούν το πυκνό και το μεσαίο κομμάτι. Το μόνο πρόβλημα τώρα είνα ότι οι H-κορυφές με μεγάλους δείκτες, πχ το h_n, δεν μπορούν να φτάσουν H-κορυφές με μικρούς δείκτες, πχ h_1. Οι κορυφές στο κομμάτι διασύνδεσης έχουν ως σκοπό την προσεκτική σύνδεση του κάθε h_j με κάθε h_i με $j > i + 1$, χωρίς να χαλάει η ιδιότητα ότι το μόνο χρονικό μονοπάτι από το h_{2i} στο h_{2i-1} είναι μέσω του μονοπατιού p_i.

□

Πρέπει να υπογραμμίσουμε ότι αν αυξήσουμε την ετικέτα της ακμής $\{a_1, m_1\}$ από ϵ σε 1, στο γράφημα του Θεωρήματος 2.4.2, το νέο χρονικά μεταβαλλόμενο γράφημα έχει γραμμικό μέγεθος MTCS. Επιπλέον, δεν είναι δύσκολο να τροποποιήσουμε την κατασκευή του Θεωρήματος 2.4.2 έτσι ώστε όλες οι ετικέτες των ακμών να είναι διακεκριμένες, το $\mathcal{G}$ να είναι συνεκτικό με αυστηρά χρονικά μονοπάτια, και η διαγραφή οποιωνδήποτε $5n$ ακμών να έχει ως αποτέλεσμα ένα μη συνεκτικό γράφημα. Συνεπώς, το τετραγωνικό κάτω φράγμα του Θεωρήματος 2.4.2 ισχύει και για συνεκτικότητα με αυστηρά χρονικά μονοπάτια και βελτιώνει το κάτω φράγμα στο [Akri15, Theorem 3].

Κεφάλαιο 3

Η Προσεγγισιμότητα της χρονική συνεκτικότητας μίας πηγής

Στην Ενότητα 3, δείχνουμε ότι η προσεγγισιμότητα του προβλήματος της χρονικής συνεκτικότητας μίας πηγής (r-MTC) σχετίζεται με την προσεγγισιμότητα του κλασικού προβλήματος Directed Steiner Tree. Με ένα μετασχηματισμό από το Directed Steiner Tree στο r-MTC (Θεώρημα 3.1.1) δείχνουμε ότι το r-MTC δεν μπορεί να προσεγγιστεί με λόγο $O(\log^{2-\varepsilon} n)$, για κάθε σταθερά $\varepsilon > 0$, εκτός αν NP $\subseteq$ ZTIME$(n^{\text{poly} \log n})$. Ο μετασχηματισμός μας υπονοεί επίσης ότ κάθε $o(n^{\varepsilon})$-προσέγγιση για το r-MTC βελτιώνει τον καλύτερο γνωστό λόγο προσέγγισης για το Directed Steiner Tree. Στα θετικά, με ένα μετασχηματισμό από το r-MTC στο Directed Steiner Tree και τον αλγόριθμο του [Char99], λαμβάνουμε έναν $O(n^{\varepsilon})$ προσεγγιστικό αλγόριθμο. Επίσης δείχνουμε ότι το r-MTC μπροεί να λυθεί σε πολυωνυμικό χρόνο αν το υποκείμενο γράφημα έχει φραγμένο δενδροπλάτος (Θεώρημα 3.3.1).

Ορισμός 3.0.1 (Directed Steiner Tree)**.** Δοθέντος ενός κατευθυνόμενου ακμοβεβαρυμένου γραφήματος $G(V, E)$ με n κορυφές, μία πηγή $r \in V$ και ένα σύνολο k τερματικών $S \subseteq V$, το Directed Steiner Tree (DST) πρόβλημα ζητά για ένα υπογράφημα του G που περιλαμβάνει κατευθυνόμενο μονοπάτι από το r προς κάθε κορυφή του S, και έχει ελάχιστο συνολικό βάρος. Ο καλύτερος γνωστός προσεγγιστικός αλγόριθμος για αυτό το πρόβλημα έχει λόγο προσέγγισης $O(k^{\varepsilon})$.

3.1 Ένα κάτω φράγμα στην προσεγγισιμότητα του r-MTC

Ξεκινάμε με μία αναγωγή του DST στο r-MTC. Η διαίσθηση είναι ότι μπορούμε να χρησιμοποιήσουμε αυστηρά χρονικά μονοπάτια για να εξομοιώσουμε τις κατευθυνόμενες ακμές του DST.

***Θεώρημα* 3.1.1.** *Κάθε $\rho(n)$-προσεγγιστικός αλγόριθμος για το r-*MTC *σε απλά χρονικά μεταβαλλόμενα γραφήματα γεννά έναν $\rho(n^2)$-προσεγγιστικό αλγόριθμο για το* DST.

Το Directed Steiner Tree δεν μπορεί να προσεγγιστεί με λόγο από $O(\log^{2-\varepsilon} n)$, εκτός εάν NP $\subseteq$ ZTIME$(n^{\text{poly} \log n})$ [Halp03, Theorem 1.2]. Το Θεώρημα 3.1.1 δείχνει ότι αυτό το αποτέλεσμα ισχύει και για το r-MTC.

3.2 Ένας προσεγγιστικός αλγόριθμος για το r-MTC

Σε αυτή την Ενότητα παρουσιάουμε μία αναγωγή από το r-MTC στο DST, δείχνοντας έτσι το Θεώρημα 3.2.1. Τότε, χρησιμοποιούμε τον αλγόριθμο του [Char99] και προσεγγίζουμε το r-MTC σε λόγο $O(n^{\varepsilon})$, για κάθε σταθερά $\varepsilon > 0$.

***Θεώρημα* 3.2.1.** *Κάθε $\rho(k)$-προσεγγιστικός αλγόριθμος για το* DST *έχει ως αποτέλεσμα έναν $\rho(n)$-προσεγγιστικό αλγόριθμο για το r-*MTC *σε γενικά χρονικά μεταβαλλόμενα γραφήματα.*

3.3 Ένας αλγόριθμους πολυωνυμικού χρόνου για γραφήματα με φραγμένο δενδροπλάτος

Σε αυτή την ενότητα δείχνουμε ότι το r-MTC μπορεί να λυθεί σε πολυωνυμικό χρόνο, με δυναμικό προγραμματισμό, αν το υποκείμενο γράφημα έχει φραγμένο δενδροπλάτος.

***Θεώρημα* 3.3.1.** *Έστω $\mathcal{G}$ ένα χρονικά μεταβαλλόμενο γράφημα με n κορυφές και διάρκεια ζωής L, κορυφή πηγής r και δενδροπλάτος το πολύ k. Τότε, υπάχει ένας αλγόριθμος δυναμικού προγραμματισμού, ο οποίος υπολογίζει μία βέλτιστη λύση για το r-MTC σε χρόνο $O(nk^2 3^k (L+k)^{k+1})$.*

Κεφάλαιο 4

Η προσεγγισιμότητα της ελάχιστης χρονικής συνεκτικότητας όλων των ζευγαριών

Σε αυτή την ενότητα μελετάμε την προσεγγισιμότητα του προβλήματος της ελάχιστης χρονικής συνεκτικότητας όλων των ζευγαριών σε γενικά χρονικά μεταβαλλόμενα γραφήματα. Ανάγοντας το MTC στο r-MTC και στο Directed Steiner Forest, λαμβάνουμε προσεγγιστικούς αλγορίθμους πολυωνυμικού χρόνου για το MTC. Συγκεκριμένα, παρουσιάζουμε έναν $O(\min\{n^{1+\varepsilon}, (\Delta M)^{2/3+\varepsilon}\})$ προσεγγιστικό αλγόριθμο για το MTC. Επίσης παρουσιάζουμε μια αναγωγή του προβλήματος Symmetric Label Cover (SLC) στο MTC και δείχνουμε ότι οποιοσδήποτε $\rho(n)$-προσεγγιστικός αλγόριθμος για το MTC έχει ως συνέπεια έναν $\rho(n^2)$ προσεγγιστικό αλγόριθμο για το SLC. Όπως θα δούμε, αυτό με τη σειρά του σημαίνει ότι για κάθε ε δεν υπάρχει $O(2^{\log^{1-\varepsilon}})$ προσεγγιστικός αλγόριθμος για το MTC, εκτός αν NP $\subseteq$ DTIME$(n^{\text{poly}\log n})$ ([Dodi99, Section 4]). Επιπλέον, με μία αναγωγή από το Steiner Tree, δείχνουμε ότι η μη βεβαρυμένη έκδοση του MTC είναι APX-hard (Theorem **??**).

Ορισμός 4.0.1 (Directed Steiner Forest)**.** Δοθέντος ενός κατευθυνόμενου ακμοβεβαρυμένου γραφήματος $G(V, E)$ με n κορυφές και m ακμές, και μιας συλλογής $D \subseteq V \times V$ από k ζευγάρια κορυφών, το Directed Steiner Forest (DSF) ζητάει ένα υπογράφημα του G που περιέχει κάποιο $s - t$ μονοπάτι για κάθε $(s, t) \in D$, και έχει ελάχιστο συνολικό βάρος.

4.1 Προσεγγιστικοί Αλγόριθμοι για το MTC

Χρηιμοποιώντας κάθε κορυφή του χρονικά μεταβαλλόμενου γραφήματος σαν κορυφή πηγής και παίρνοντας την ένωση των λύσεων που προκύπτουν από τον αλγόριθμο του Θεωρήματος 3.2.1 για το r-MTC, παίρνουμε το παρακάτω:

***Πόρισμα* 4.1.1.** *Για κάθε $\varepsilon > 0$, υπάρχει $(n^{1+\varepsilon})$-προσεγγιστικός αλγόριθμος για το* MTC *σε χρονικα μεταβαλλόμενα γραφήματα με n κορυφές.*

Επίσης, το παρακάτω Θεώρημα προκύπτει από μία αναγωγή του MTC στο Directed Steiner Forest και δίνει ένα διαφορετικό προσεγγιστικό αλγόριθμο για το πρώτο.

***Θεώρημα* 4.1.2.** *Έστω $\mathcal{G}$ ένα χρονικά μεταβαλλόμενο γράφημα με n κορυφές και M χρονικές ακμές τέτοιο ώστε το υποκείμενο γράφημα να έχει μέγιστο βαθμό Δ. Τότε, για κάθε $\epsilon > 0$, υπάρχει πολυωνυμικός $O(M^{\varepsilon} \min\{M^{4/5}, (\Delta M)^{2/3})$-προσεγγιστικός αλγόριθμος για το* MTC *στον $\mathcal{G}$. Αν $M = O(n \operatorname{poly}\log n)$, αυτό δίνει λόγο προσέγγισης $O(n^{4/5+\varepsilon})$. Αν ταυτόχρονα $M = O(n \operatorname{poly}\log n)$ και $\Delta = O(\operatorname{poly}\log n)$, παίρνουμε λόγο προσέγγισης $O(n^{2/3+\varepsilon})$.*

4.2 Κάτω φράγμα για την προσεγγισιμότητα του MTC

Το παρακάτω θεώρημα προκύπτει από μια αναγωγή του Symmetric Label Cover στο MTC. Μαζί με γνωστά αποτελέσματα για τη μη προσεγγισιμότητα του Symmetric Label Cover, αυτό δείχνει ότι το MTC σε γενικά χρονικά μεταβαλλόμενα γραφήματα είναι δύσκολο να προσεγγιστεί.

***Θεώρημα* 4.2.1.** *Για κάθε $\varepsilon > 0$, το* MTC *σε χρονικά μεταβαλλόμενα γραφήματα με n κορυφές δεν μπορεί να προσεγγιστεί με λόγο προσέγγισης $O(2^{\log^{1-\varepsilon} n})$, εκτός εάν* NP $\subseteq$ DTIME$(n^{\text{poly} \log n})$.

Κεφάλαιο 5

Χρονική συνεκτικότητα όλων των ζευγαριών σε Δένδρα και Κύκλους

Στην Ενότητα 5, δείχνουμε ότι το MTC μπορεί να λυθεί πολύ πιο αποδοτικά για περιορισμένες οικογένειες υποκείμενων γραφημάτων. Συγκεκριμένα, στην Ενότητα 5.1 Λήμμα 5.1.2, δείχνουμε ότι αν το υποκείμενο γράφημα είναι δένδρο, υπάρχει βέλτιστη λύση για το MTC που χρησιμοποιεί κάθε ακμή το πολύ κατά τη διάρκεια δύο διαφορετικών χρονικών στιγμών. Χρησιμοποιώντας αυτή την ιδιότητα, μπορούμε να δείξουμε ότι το MTC μπορεί να λυθεί αποδοτικά με δυναμικό προγραμματισμό αν το υποκείμενο γράφημα είναι δένδρο. Επίσης, δείχνουμε ότι το MTC είναι 2-προσεγγίσιμο αν το υποκείμενο γράφημα είναι κύκλος.

5.1 Χρονική Συνεκτικότητα σε Δένδρα: Αλγόριθμος Πολυωνυμικού χρόνου

***Θεώρημα* 5.1.1.** *Έστω $\mathcal{G}$ ένα χρονικά μεταβαλλόμενο δένδρο σε n κορυφές με διάρκεια ζωής L. Υπάρχει αλγόριθμος δυναμικού προγραμματισμού που υπολογίζει μία βέλτιστη λύση για το* MTC *στον $\mathcal{G}$ σε χρονο $O(nL^4)$.*

Σημαντικό λήμμα είναι το παρακάτω:

***Λήμμα* 5.1.2.** *Έστω $\mathcal{G}$ ένα χρονικό δένδρο. Τότε υπάρχει βέλτιστη λύση του* MTC *στο $\mathcal{G}$ που χρησιμοποιεί το πολύ δύο χρονικές ετικέτες σε καθε ακμή.*

5.2 Χρονική Συνεκτικότητα σε κύκλους: 2-προσεγγιστικός Αλγόριθμος

Σε αυτή την ενότητα, παρατηρούμε ότι εάν το υποκείμενο γράφημα είναι ένας κύκλος $C_n = (v_0, v_1, \ldots, v_{n-1}, v_0)$, κάθε ελαχιστικό χρονικά συνεκτικό υπογράφημα $\mathcal{G}'$ μπορεί να διαμεριστεί σε *τομείς*. Ένας τομέας είναι ένα συνεχόμενο τμήμα $(v_i, v_{i+1}, \ldots, v_k)$ του κύκλου για το οποίο υπάρχει μία κορυφή $v_j \notin \{v_i, \ldots, v_{k-1}\}$ έτσι ώστε τα χρονικά μονοπάτια $p_{\text{incr}} = (v_i, v_{i+1}, \ldots, v_j)$ και $p_{\text{decr}} = (v_k, v_{k-1}, \ldots, v_{j+1})$ υπάρχουν στο $\mathcal{G}'$ (οι δείκτες των κορυφών κατά μήκος του C_n παίρνονται modulo). Διαισθητικά, κάθε κορυφή στον τομέα $(v_i, v_{i+1}, \ldots, v_k)$ μπορεί να φτάσει κάθε κορυφή στο C_n μέσω των μονοπατιών p_{incr} και P_{decr}. Τότε, στο Λήμμα 5.2.4, δείχνουμε ότι υπάρχει μία βέλτιστη λύση για το MTC στο C_n όπου κάθε ακμή ανήκει σε το πολύ δύο τομείς. Τότε, αγνοώντας το γεγονός ότι υπάρχουν κοινές ακμές σε διαφορετικούς τομείς και χρησιμοποιώντας δυναμικό προγραμματισμό για να βρούμε μία βέλτιστη διαμέριση του C_n σε τομείς, λαμβάνουμε το παρακάτω.

***Θεώρημα* 5.2.1.** *Υπάρχει 2-προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου για το* MTC *σε οποιονδήποτε χρονικό κύκλο C_n.*

Σημαντικό Λήμμα είναι το παρακάτω.

Λήμμα **5.2.2.** *Ας υποθέσουμε ότι σε κάποιο χρονικά συνεκτικό υπογράφημα $\mathcal{G}'$ του χρονικού κύκλου, υπάρχει ένα χρονικό μονοπάτι $v_i, v_{i+1}, \dots, v_j$ και για κάποιο $k \in \{i, i+1, \dots, j\}$ υπάρχει επίσης ένα χρονικό μονοπάτι $v_k, v_{k-1}, \dots, v_{j+2}, v_{j+1}$. Τότε κάθε κορυφή $v_p \in \{i, i+1, \dots, k\}$ είναι συνδεδεμένη σε κάθε άλλη κορυφή στο $\mathcal{G}'$.*

Αυτό μας οδηγεί στον παρακάτω ορισμό.

Ορισμός 5.2.3. Έστω $\mathcal{G}$ χρονικός κύκλος και $\mathcal{G}'$ ένα ελαχιστικό χρονικά συνεκτικό υπογράφημα του $\mathcal{G}$. Ένας *τομέας* του $\mathcal{G}$ είναι μία συνεχόμενη ακολουθία κορυφών $(v_i, v_{i+1}, \dots, v_j)$ έτσι ώστε να υπάρχει κορυφή $v_k \notin \{i, i+1, \dots, j-1\}$ και τα χρονικά μονοπάτια $p_{\text{incr}} = (v_i, v_{i+1}, \dots, v_k)$ και $p_{\text{decr}} = (v_j, v_{j-1}, \dots, v_{k+1})$ να υπάρχουν στο $\mathcal{G}'$. Θα αναφερόμαστε στο p_{incr} ως το *αύξων μονοπάτι* και στο p_{decr} ως το *φθίνον μονοπάτι* του τομέα $(v_i, v_{i+1}, \dots, v_j)$. Το *κόστος* ενός τομέα $S = (v_i, v_{i+1}, \dots, v_j)$ είναι το ελάχιστο, πάνω σε όλες τις επιλογές ζευγαριών αύξοντων και φθινόντων μονοπατιών για το S, του συνολικού κόστους όλων των χρονικών ακμών που υπάρχουν σε αυτά τα δύο μονοπάτια.

Οι κορυφές ενός χρονικά συνεκτικού χρονικού κύκλου $\mathcal{G}$ μπορούν πάντα να διαμεριστούν σε τομείς.

Λήμμα **5.2.4.** *Για ένα χρονικό κύκλο $\mathcal{G}$, υπάρχει μία βελτιστη λύση για το* MTC *έτσι ώστε για κάθε δύο τομείς τα αύξοντα (αντίστοιχα: τα φθίνοντα) μονοπάτια τους δεν έχουν κάποια κοινή χρονική ακμή.*

Το παραπάνω λήμμα σημαίνει ότι σε κάποια λύση του MTC, κάθε χρονική ακμή χρησιμοποιείται το πολύ δύο φορές από διαφορετικούς τομείς. Οπότε, αγνοώντας την αμοιβαία εξάρτηση μεταξύ τομέων παίρνουμε μία 2-προσεγγιστική λύση για το πρόβλημα.

Θεώρημα **5.2.5.** *Υπάρχει πολυωνυμικός αλγόριθμος που υπολογίζει μία εφικτή* MTC *λύση ενός χρονικού κύκλου $\mathcal{G}$, με κόστος το πολύ 2 φορές το βελτιστο κόστος.*

Bibliography

[Akri15] E.C. Akrida, L. Gąsieniec, G.B. Mertzios and P.G. Spirakis, "On Temporally Connected Graphs of Small Cost", in *Proc. of the 13th Workshop on Approximation and Online Algorithms (WAOA '15)*, vol. 9499 of *LNCS*, pp. 84–96, 2015.

[Akri16] E.C. Akrida, L. Gąsieniec, G.B. Mertzios and P.G. Spirakis, "Ephemeral networks with random availability of links: The case of fast networks", *Journal of Parallel and Distributed Computing*, vol. 87, pp. 109–120, 2016.

[Angl06] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer and R. Peralta, "Computation in networks of passively mobile finite-state sensors", *Distributed Computing*, vol. 18, no. 4, pp. 235–253, 2006.

[Avin08] C. Avin, M. Koucký and Z. Lotker, "How to Explore a Fast-Changing World (Cover Time of a Simple Random Walk on Evolving Graphs)", in *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, pp. 121–132, 2008.

[Barj14] M. Barjon, A. Casteigts, S. Chaumette, C. Johnen and Y. M. Neggaz, "Testing Temporal Connectivity in Sparse Dynamic Graphs", *CoRR*, vol. abs/1404.7634, 2014.

[Berm96] K.A. Berman, "Vulnerability of scheduled networks and a generalization of Menger's Theorem", *Networks*, vol. 28, no. 3, pp. 125–134, 1996.

[Bhaw13] K. Bhawalkar, S. Gollapudi and K. Munagala, "Coevolutionary Opinion Formation Games", in *Proc. of the 45th ACM Symposium on Theory of Computing (STOC '13)*, pp. 41–50, 2013.

[Bui 03] B. Bui-Xuan, A. Ferreira and A. Jarry, "Computing Shortest, Fastest, and Foremost Journeys in Dynamic Networks", *Int. J. Found. Comput. Sci.*, vol. 14, no. 2, pp. 267–285, 2003.

[Cast12] A. Casteigts, P. Flocchini, W. Quattrociocchi and N. Santoro, "Time-varying graphs and dynamic networks", *International Journal of Parallel, Emergent and Distributed Systems (IJPEDS)*, vol. 27, no. 5, pp. 387–408, 2012.

[Char99] M. Charikar, C. Chekuri, T.-Y. Cheung, Z. Dai, A. Goel, S. Guha and M. Li, "Approximation Algorithms for Directed Steiner Problems", *Journal of Algorithms*, vol. 33, no. 1, pp. 73–91, 1999.

[Chaz12] B. Chazelle, "The Dynamics of Influence Systems", in *Proc. of the 53rd IEEE Symposium on Foundations of Computer Science (FOCS '12)*, pp. 311–320, 2012.

[Dodi99] Y. Dodis and S. Khanna, "Design Networks with Bounded Pairwise Distance", in *Proc. of the 31st ACM Symposium on Theory of Computing (STOC '99)*, pp. 750–759, 1999.

[Dutt13] C. Dutta, G. Pandurangan, R. Rajaraman, Z. Sun and E.Viola, "On the Complexity of Information Spreading in Dynamic Networks", in *Proc. of the 24th ACM-SIAM Symposium on Discrete Algorithms (SODA '13)*, pp. 717–736, 2013.

[Erle15] T. Erlebach, M. Hoffmann and F. Kammer, "On Temporal Graph Exploration", in *Proc. of the 42nd International Colloquium on Automata, Languages and Programming (ICALP '15)*, vol. 9134 of *LNCS*, pp. 444–455, 2015.

[Feld12] M. Feldman, G. Kortsarz and Z. Nutov, "Improved approximation algorithms for Directed Steiner Forest", *Journal of Computer and System Sciences*, vol. 78, no. 1, pp. 279–292, 2012.

[Flei98] L. Fleischer and É. Tardos, "Efficient continuous-time dynamic network flow algorithms", *Oper. Res. Lett.*, vol. 23, no. 3-5, pp. 71–80, 1998.

[Flei07] L. Fleischer and M. Skutella, "Quickest Flows Over Time", *SIAM J. Comput.*, vol. 36, no. 6, pp. 1600–1630, 2007.

[Fosc14] L. Foschini, J. Hershberger and S. Suri, "On the Complexity of Time-Dependent Shortest Paths", *Algorithmica*, vol. 68, no. 4, pp. 1075–1097, 2014.

[Halp03] E. Halperin and R. Krauthgamer, "Polylogarithmic inapproximability", in *Proc. of the 35th ACM Symposium on Theory of Computing (STOC '03)*, pp. 585–594, 2003.

[Hede88] S. M. Hedetniemi, S. T. Hedetniemi and A. L. Liestman, "A survey of gossiping and broadcasting in communication networks", *Networks*, vol. 18, no. 4, pp. 319–349, 1988.

[Holm12] P. Holme and J. Saramäki, "Temporal Networks", *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.

[Huan15] S. Huang, A.W.-C. Fu and R. Liu, "Minimum Spanning Trees in Temporal Graphs", in *Proc. of the 2015 ACM International Conference on Management of Data (SIGMOD '15)*, pp. 419–430, 2015.

[Kemp02] D. Kempe, J.M. Kleinberg and A. Kumar, "Connectivity and Inference Problems for Temporal Networks", *Journal of Computer and System Sciences*, vol. 64, no. 4, pp. 820–842, 2002.

[Kuhn10] F. Kuhn, N.A. Lynch and R.Oshman, "Distributed computation in dynamic networks", in *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC '10)*, pp. 513–522, 2010.

[Mert13] G.B. Mertzios, O. Michail, I. Chatzigiannakis and P.G. Spirakis, "Temporal Network Optimization Subject to Connectivity Constraints", in *Proc. of the 40th International Colloquium on Automata, Languages and Programming (ICALP '13)*, vol. 7966 of *LNCS*, pp. 657–668, 2013.

[Mich11] O. Michail, I. Chatzigiannakis and P. G. Spirakis, "Mediated population protocols", *Theor. Comput. Sci.*, vol. 412, no. 22, pp. 2434–2450, 2011.

[Mich14] O. Michail, I. Chatzigiannakis and P. G. Spirakis, "Causality, influence, and computation in possibly disconnected synchronous dynamic networks", *J. Parallel Distrib. Comput.*, vol. 74, no. 1, pp. 2016–2026, 2014.

[ODel05] R. O'Dell and R. Wattenhofer, "Information dissemination in highly dynamic graphs", in *Proc. of the DIALM-POMC Joint Workshop on Foundations of Mobile Computing (DIALM-POMC '05)*, pp. 104–110, 2005.

[Ravi94] R. Ravi, "Rapid Rumor Ramification: Approximating the minimum broadcast time (Extended Abstract)", in *Proc. of the 35th IEEE Symposium on Foundations of Computer Science (FOCS '94)*, pp. 202–213, 1994.

[Whit12] J. Whitbeck, M. Dias de Amorim, V. Conan and J. Guillaume, "Temporal reachability graphs", in *The 18th Annual International Conference on Mobile Computing and Networking, Mobicom'12, Istanbul, Turkey, August 22-26, 2012*, pp. 377–388, 2012.