Eötvös Loránd University
Faculty of Science
Institute of Mathematics

András Mihálykó

Augmentation problems in count matroids and globally rigid graphs

Ph.D. dissertation

Supervisor: Tibor Jordán
Professor, Doctor of the Hungarian Academy of Sciences, Eötvös Loránd University

Doctoral School of Mathematics
Director: Tibor Jordán, Professor, Doctor of the Hungarian Academy of Sciences

Doctoral Program: Applied Mathematics
Director: János Karátson, Professor, Doctor of the Hungarian Academy of Sciences

DOI: 10.15476/ELTE.2022.277

Department of Operations Research, Eötvös Loránd University

Budapest, 2022

To the memory of my brother.

Revisions Following Dissertation Defense

This revised edition of my dissertation incorporates corrections and improvements based on feedback received during the review process of this dissertation. The changes were necessary to make my argument complete.

The main differences arise with Theorem 3.9. In the original version, this was presented as one theorem for every (m, ℓ) case, without requiring the condition $|V| > 2c$, currently present. The theorem in that form is not true. Instead, we need to separate the theorem to three different cases, each adding their own conditions. These are Theorems 3.9, 3.10 and 3.11. Like this, we could prove the theorems and use these results in the rest of the dissertation without compromising any further results. Most of Subsection 3.3.1 consists of these revised results, that is, the correct proofs of Theorems 3.9, 3.10 and 3.11.

In addition to this, I reconsidered the use of c. In this revised version, condition $|V| \geq c^2 + 2$ appears frequently. This is a change from the original that does not make the theoretical results weaker but (hopefully) makes the results easier to follow, as it simplifies some steps of the proofs.

I also incorporated some other observations and remarks from the reviewers to improve the quality of this work. This affected mainly Section 3.1, Section 3.3 (even besides Subsection 3.3.1), Section 3.4 and Section 6.2. Nonetheless, I made small changes in many places throughout the dissertation.

I hope that these changes help to improve the readability and quality of this dissertation.

I am grateful to my eagle-eyed reviewers, especially to Zoltán Szigeti, who made a really conscientious review, significantly improving the quality of this work. He pointed out the incorrectness of the old Theorem 3.9. Without it, this work would not be complete. I am also grateful to my friend Máté Gyarmati, who helped me fix a problem arisen during the review process.

Revised version completed in November 2023, this explanation page added in August 2025.

Acknowledgements

Writing this PhD dissertation was not an easy task and it would not have been possible without the help of many great people. I would like to express my gratitude to all of them.

First of all, I wish to thank my supervisor, Tibor Jordán. He introduced me to the fascinating topic of rigidity theory in his amazing lectures. He presented me with the first paper, which inspired the start of this research back in my master's studies. He gave the opportunity for my first results to appear in an article. I would like to thank him for having accepted me as a PhD student. During my PhD studies, he always supported me and provided me with the freedom to work on the questions that interested me the most. I'm also grateful that he made it possible for me to participate in numerous conferences which I enjoyed immensely.

I would like to express my gratitude to Csaba Király, who is the co-author of most of my papers. His concentration and thoroughness made sure to cover every special case and not to leave any mistakes in the final versions. If I was stuck, I could ask him at any unmanly hour, and he gave a prompt response. I'm also grateful that any time I had a crazy idea, he took it seriously (and if they were incorrect, he provided a counter-example, before I could really believe in them).

I would also like to thank the people from the Department of Operations Research for the inspiring and uplifting atmosphere that I could enjoy mainly the times before the covid. I would like to thank especially Péter Madarasi for helping me confirm several of my conjectures and checking my programming results. I would like to thank our secretary Klaudia Szalay for taking so much administrative work off our shoulders.

Last but not least, I thank my family for helping me in this difficult journey. I am especially grateful of my wife, Zsófi, who tolerated my late working hours and encouraged me in all of my challenges.

Contents

1 Introduction **9**

1.1 Motivation . . . 9

1.2 Outline . . . 10

1.3 Notation and definitions . . . 12

2 Preliminaries **15**

2.1 Introduction to rigidity theory . . . 15

2.2 Count matroids . . . 18

2.3 Preliminary lemmas . . . 21

3 (k, ℓ)-redundant augmentation **23**

3.1 The reduction of the general problem . . . 26

3.2 Preprocessing . . . 29

3.3 The min-max theorem for the reduced problem . . . 30

3.3.1 Proofs of Theorems 3.9, 3.10 and 3.11 . . . 39

3.4 Complexity results . . . 48

4 (k, ℓ)-M-connected hypergraphs **55**

5 Globally rigid augmentation of rigid graphs **61**

5.1 Preliminaries . . . 62

5.2 Connectivity augmentation . . . 64

5.3 The min-max theorem . . . 65

5.3.1 Proof of Theorem 5.6 for $k < \ell \leq \frac{3}{2}k$ . . . 67

5.3.2 Proof sketch of Theorem 5.6 for $\ell \leq k$ . . . 77

6 Minimal cost globally rigid subgraph **79**

6.1 Minimal size globally rigid spanning subgraph . . . 80

6.2 Minimal cost globally rigid spanning subgraph . . . 81

6.2.1 Conjecture of approximation ratio . . . 84

6.2.2 Euclidean cost functions . . . 84

6.3 Complexity results . . . 85

7 Additional results **87**

7.1 Pinning problems . . . 87

7.1.1 Rigidity pinning problem . . . 88

7.1.2 Redundant rigidity pinning . . . 89

7.1.3 Global rigidity pinning . . . 90

7.1.4 Cost of pinning . . . 92

7.2 Non-rigid inputs . . . 93

7.2.1 The redundant augmentation problem on non-rigid graphs . . . 93

7.2.2 Globally rigid augmentation problem on non-rigid graphs . . . 94

7.2.3 Pinning problems on non-rigid graphs . . . 94

7.3 Characterization of tight graphs . . . 95

7.4 Simple graphs . . . 96

7.5 Adding multiple edges . . . 97

7.6 Open problems . . . 98

7.6.1 Redundant augmentation of (k, ℓ)-sparse graphs . . . 98

7.6.2 Globally rigid augmentation of general graphs . . . 99

7.6.3 Minimum cost globally rigid subgraph problem . . . 99

7.6.4 Augmentation with complete graphs of constant size . . . 100

7.6.5 Co-tight sets in higher dimensional rigidity . . . 100

7.6.6 Common framework for rigidity and connectivity . . . 101

7.6.7 $O(|V|^2)$ algorithm for minimum cost (k, ℓ)-tight spanning subgraph . . . 101

8 Algorithmic aspects **103**
8.1 Testing (k, ℓ)-sparsity for graphs 103
8.2 Algorithmic construction of the M-component hypergraph 105
8.3 Making (k, ℓ)-M-component hypergraphs (k, ℓ)-redundant 111
8.3.1 Finding a transversal of the MCT sets 112
8.3.2 Optimal augmenting edge set 118
8.4 Algorithm for global rigidity augmentation 119
8.4.1 Efficient connectivity structures 120
8.4.2 Finding a transversal of atoms 121
8.4.3 Globally rigid augmentation algorithm for $0 < \ell \leq k$ 122
8.4.4 Global rigidity augmentation algorithm for $k < \ell \leq \frac{3}{2}k$ 125
8.5 Algorithmic solutions of pinning problems 126
8.6 Approximation of minimum cost globally rigid subgraphs 128

Bibliography **131**

Appendix: Program **139**

Summary **143**

Összefoglalás **145**

Chapter 1

Introduction

1.1 Motivation

Rigidity theory is an important area with lots of industrial applications ranging from biotechnology and (bio)chemistry through architecture to sensor localization problems. It lies at the crossroads of combinatorics and distance geometry. Its main focus is on determining whether a given d-dimensional framework consisting of joints connected with solid bars is rigid or not. Moreover, if the framework is rigid, is there any other framework in a different position that can be built from the same bars between the same joints? The answer to the first question can determine the rigidity of a structure. Answering the second question may help in localizing the objects present in the framework.

For example, given some sensors with known distances between some of them, one may consider the following question. At least how many sensor-locations do we need to measure exactly to be able to reconstruct the exact location of each sensor? Sometimes measuring the exact sensor-locations is too expensive or even impossible. Instead, one may ask at least how many new distances need to be measured so that the distances – possibly with some already known positions of some sensors – uniquely determine the positions of all the sensors (or at least up to isometry, if it is impossible to determine exactly). A similar question can occur if the sensors are able to move (say they are attached to drones). How many new distances do we need to measure so that they can maintain their relative position to the others?

This last question asks for the minimum number of new distances so that the sensors and the distances form a rigid framework as joints and bars, respectively. This problem is well studied and, in $\mathbb{R}^2$ with the assumption that the sensors are in generic positions, it can be solved efficiently. On the other hand, the problem of finding minimum number of new distances so that the framework becomes uniquely localizable is open even in $\mathbb{R}^2$, despite its usefulness in the localization of sensor networks. The main results in this dissertation will contribute to unique localizability problems in $\mathbb{R}^2$. We answer several questions regarding the unique localizability of frameworks and give efficient algorithms for multiple answered questions.

Finally, once a network is uniquely localizable, the next step may be to determine the exact positions. We have to note that reconstructing the positions of the sensors is a challenging task, even if they are uniquely determined. There exist some advanced heuristics and methods to tackle this challenge, however, in this dissertation we do not address this problem.

1.2 Outline

In this section we give a short outline of the dissertation. We finish Chapter 1 with a brief summary of the basic definitions and notions used throughout this dissertation.

In Chapter 2 we collect the concepts, definitions and lemmas that we shall use extensively throughout the whole dissertation. As the problems we mentioned in the Motivation section are strongly related to (combinatorial) rigidity, we start with a brief introduction to the rigidity of graphs in Section 2.1. In this section we collect the basic definitions of rigidity theory that are necessary to understand the results presented. For a detailed and more excessive introduction to rigidity theory, the reader is referred to [24, 40]. In Section 2.2 we briefly introduce the so-called count matroids. They play a crucial role, as most of the results are presented with them in this dissertation. For a more detailed introduction to them, the reader is referred to [17, Section 13.5]. We finish Chapter 2 with some fundamental lemmas presented in Section 2.3 that will be used several times in this dissertation.

The new results start with Chapter 3, where we consider the problem of adding a minimal edge set to a hypergraph to make it (k, ℓ)-redundant. We show that this problem is optimally solvable if the input is (k, ℓ)-tight or if the input is (k, ℓ)-rigid and $\ell \leq k$, while the problem is NP-hard in case of (k, ℓ)-rigid input and $\ell > k$. Similar results were previously known for some special cases, e.g. $(k, \ell) = (1, 1)$ [14] or $(k, \ell) = (2, 3)$ [22]. Besides giving solutions with general (k, ℓ) values, we also contribute to these special cases with new structural results and new min-max theorems. This chapter is based on [51], which is my joint work with Csaba Király.

In Chapter 4 we present the connections between (k, ℓ)-redundancy, (k, ℓ)-M-connectedness and the (k, ℓ)-M-component hypergraph. Most of these results were known for $(k, \ell) = (2, 3)$ [32]. We generalize them here to every $\ell \leq \frac{3}{2}k$. This chapter is based on [50, Section 3], which is my joint work with Csaba Király.

In Chapter 5 we investigate the problem of adding edges to a (k, ℓ)-rigid graph to make it (k, ℓ)-redundant and $(c_{k,\ell} + 1)$-connected where $c_{k,\ell} = \max\left\{\left\lceil \frac{\ell}{k} \right\rceil, 0\right\}$. This includes the problem of adding minimum number of edges to a rigid graph in $\mathbb{R}^2$ to make it globally rigid. The best known results to this particular problem were constant factor approximations [16], only for the $(k, \ell) = (2, 3)$ case. We give optimal solutions to every $\ell \leq \frac{3}{2}k$. This chapter is based on [50], which is my joint work with Csaba Király.

In Chapter 6 we investigate the problem of finding a minimum cost (k, ℓ)-redundant and $(c_{k,\ell} + 1)$-connected subgraph of a weighted graph (with $\ell > 0$ hence $c_{k,\ell} = \left\lceil \frac{\ell}{k} \right\rceil$). This problem incorporates finding a minimum cost globally rigid subgraph in $\mathbb{R}^2$. We show that the general problem is NP-hard and we show approximation algorithms for two restricted versions of it (which are still NP-hard). This chapter is based on [42], my joint work with Tibor Jordán.

Chapter 7 collects some additional results that are mainly applications of theorems proved in Chapters 3 and 5. These include approximation algorithms for the problems presented in Sections 3 and 5 with different inputs as well as exact and approximation results for the so-called pinning or anchoring problems. These are in close connection with the motivating problems presented in the introduction. We finish Chapter 7 with some open problems that are closely related to the topics discussed in this dissertation. The results of this chapter are partially present in [50], which is my joint work with Csaba Király, and partially my unpublished results.

Finally, in Chapter 8 we collect the algorithmic aspects of several of the previously presented results. We show efficient algorithms to many of them. This section is based on [49], which is my joint work with Csaba Király, with some extra ideas from the algorithmic results of [42], [51] and [50].

In the Appendix, we present the code for Sections 8.2 and 8.3. This code is an efficient C++ implementation of the corresponding algorithms by me and it is freely available at: `https://github.com/mihalykoandras/rigidityAugmentations`.

1.3 Notation and definitions

In this section we introduce the main notation and definitions, which shall be used throughout the dissertation.

We take basic concepts, such as graph, hypergraph, digraph, vertex, edge, hyperedge or matroid for granted.

Throughout the definitions, we suppose that $G = (V, E)$ is a graph and $\mathcal{H} = (V, \mathcal{E})$ is a hypergraph. We will use the following notions in this dissertation without further definition.

- $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_+$ denotes the set of non-negative real numbers. $\mathbb{Z}$ denotes the set of integers while $\mathbb{Z}_+$ denotes the set of non-negative integers.
- If G_1 and G_2 are graphs, then $\boldsymbol{G_1 \subseteq G_2}$ denotes that G_1 is a subgraph of G_2. If $\mathcal{H}_1$ and $\mathcal{H}_2$ are hypergraphs, then $\boldsymbol{\mathcal{H}_1 \subseteq \mathcal{H}_2}$ denotes that $\mathcal{H}_1$ is a subhypergraph of $\mathcal{H}_2$.
- A **loop** of G is an edge e, such that there exists a $v \in V$ for which both of the end-vertices of e is v. A **loop** of $\mathcal{H}$ is a hyperedge that contains exactly one vertex.
- Two edges of G or $\mathcal{H}$ are **parallel** if their vertex sets are identical.
- A graph G or hypergraph $\mathcal{H}$ is **simple**, if it does not contain any loops or parallel (hyper)edges.
- Let $\boldsymbol{G[X]}$ be the subgraph of G induced by X and let $\boldsymbol{\mathcal{H}[X]}$ denote the subhypergraph of $\mathcal{H}$ induced by X for $X \subseteq V$.
- $\boldsymbol{i_G(X)}$ denotes the number of edges of G induced by the set $X \subseteq V$. Similarly $\boldsymbol{i_{\mathcal{H}}(X)}$ denotes the number of hyperedges of $\mathcal{H}$ induced by X.
- $\boldsymbol{d_G(X, Y)}$ denotes the number of edges of G between $X - Y$ and $Y - X$ for $X, Y \subseteq V$. $\boldsymbol{d_G(X)} := d_G(X, V - X)$. $\boldsymbol{d_{\mathcal{H}}(X, Y)}$ denotes the number of hyperedges that are induced by $X \cup Y$ but induced by neither X nor Y. $\boldsymbol{d_{\mathcal{H}}(X)} := d_{\mathcal{H}}(X, V - X)$.
- Let $\boldsymbol{d_G(v)}$ denote the number of edges of G incident to $v \in V$. Note that our definition implies that $d_G(v) \neq d_G(\{v\})$ if there is a loop incident with v. Also note that, in the usual definition of the degree, loop edges count twice for the degree of a vertex, however, we only count them once. $\boldsymbol{d_{\mathcal{H}}(v)}$ denotes the number of hyperedges that contain $v \in V$.

This way of definition makes the results consistent, even if loops are present, no matter if we consider a graph G as a graph or as a (special) hypergraph.

- Let $\boldsymbol{\widehat{E}_G(X)}$ denote the set of edges of G that have at least one end-vertex in $X \subseteq V$. Let $\boldsymbol{e_G(X)}= |\widehat{E}_G(X)|$. Note that $e_G(X) = i_G(X) + d_G(X)$. Similarly, let $\boldsymbol{\widehat{E}_{\mathcal{H}}(X)}$ denote the set of hyperedges of $\mathcal{H}$ of which the vertex set intersects $X \subseteq V$. Let $\boldsymbol{e_{\mathcal{H}}(X)}= |\widehat{E}_{\mathcal{H}}(X)|$. Hence $e_{\mathcal{H}}(X) = i_{\mathcal{H}}(X) + d_{\mathcal{H}}(X)$.

- We use $\boldsymbol{N_G(X)}$ to denote the neighbor set of $X \subseteq V$ in G, that is, $N_G(X) := \{v \in V - X : d_G(\{v\}, X) \geq 1\}$. The neighbor set of X in $\mathcal{H}$ is $\boldsymbol{N_{\mathcal{H}}(X)}= \{v \in V - X :$ there is $x \in X$ and $e \in \mathcal{E}$ such that $\{v, x\} \subseteq e\}$.

- Let $\boldsymbol{G/X}$ denote the graph arising from G by contracting X into a single vertex and deleting all the edges spanned by X. We can similarly define $\boldsymbol{D/X}$ for a directed graph D. $\boldsymbol{\mathcal{H}/X}$ denotes the graph arising from $\mathcal{H}$ by contracting X into a single vertex and deleting all the hyperedges spanned by X.

- $\boldsymbol{K_X}$ denotes the complete graph on the vertex set X.

Chapter 2

Preliminaries

2.1 Introduction to rigidity theory

In this section we briefly present the basics of (combinatorial) rigidity theory.

A d-dimensional *framework* is a pair (G, p), where $G = (V, E)$ is a graph and p is a map from V to $\mathbb{R}^d$. We call (G, p) a *realization* of G in $\mathbb{R}^d$. Intuitively, we can think of a framework (G, p) as a collection of bars and joints where each vertex v of G corresponds to a joint located at $p(v)$ and each edge corresponds to a rigid (that is, fixed length) bar joining its end-points. Two realizations (G, p) and (G, q) are *equivalent* if $||p(u) - p(v)|| = ||q(u) - q(v)||$ holds for all pairs u, v with $uv \in E$, where $||.||$ denotes the Euclidean norm in $\mathbb{R}^d$. The frameworks (G, p) and (G, q) are *congruent* if $||p(u) - p(v)|| = ||q(u) - q(v)||$ holds for all pairs u, v with $u, v \in V$. This is the same as saying that (G, q) can be obtained from (G, p) by an isometry of $\mathbb{R}^d$. A framework (G, p) is called *rigid*, if there exists an $\varepsilon > 0$, such that for every framework (G, q), which is equivalent to (G, p) and for which $||p(u) - q(u)|| < \varepsilon$ for all $v \in V$, (G, q) is congruent to (G, p). Intuitively this means that every continuous motion of a rigid framework preserves the distances between every pair of vertices, hence the name. We say that (G, p) is *globally rigid* in $\mathbb{R}^d$, if every d-dimensional realization (G, q) of G, which is equivalent to (G, p) is congruent to (G, p). In other words, the framework is globally rigid if its edge lengths uniquely determine all pairwise distances. This property makes the notion of global rigidity a fundamental concept in problems where we are given partial information on the pairwise distances between pairs of a finite point set and our goal is to determine the configuration of the points, up to trivial transformations.

Determining whether a given framework is rigid (or globally rigid, respectively) is NP-hard in the plane (or even on the line) [1, 67]. The analysis gets more tractable, if we consider **generic frameworks** where the set of coordinates of the points is algebraically independent over the rationals [3, 23]. In this case, the rigidity and the global rigidity of a framework in $\mathbb{R}^d$ depend only on the underlying graph G (see [10, 23]). Hence we can call a graph G **rigid** (or **globally rigid**, respectively) in $\mathbb{R}^d$ if each (or equivalently, some) of its generic realizations in $\mathbb{R}^d$ is rigid (or globally rigid, respectively). The characterization of rigid and globally rigid graphs is known for $d = 1, 2$ and it is a major open problem of rigidity theory for $d \geq 3$. ([32, 53, 65])

It is an easy folklore result that a graph in $\mathbb{R}^1$ is rigid if and only if it is connected (for proof see e. g. [40]). The two-dimensional characterization of rigidity requires a bit more work.

A graph $G = (V, E)$ is called **sparse** if $i_G(X) \leq 2|X| - 3$ holds for all $X \subseteq V$ when $|X| \geq 2$. G is **tight** if it is sparse and $|E| = 2|V| - 3$. The fundamental results of Pollaczek-Geiringer [65] and Laman [53] show that a generic realization of a graph G in $\mathbb{R}^2$ is minimally rigid if and only if G is tight. In other words:

Theorem 2.1 ([53, 65]) *G is rigid in $\mathbb{R}^2$ if and only if G contains a spanning tight subgraph.* □

Global rigidity requires some more conditions. This is also true in $\mathbb{R}^1$. For example, if there exists a cutting vertex in a graph, then no 1-dimensional realization with non-zero length edges can be globally rigid, as we may just reflect one part of the graph on the cutting vertex. This can be generalized also to higher dimensions. Thus we shall introduce the concepts of connectivity here. A graph $G = (V, E)$ is called ***k*-connected** if $|V| > k$ and $G - X$ is connected for any vertex set $X \subset V$ of cardinality at most $k - 1$. For the sake of convenience, a graph which is not necessarily connected will be called **0-connected** in this work. We note that in this sense every graph is 0-connected.

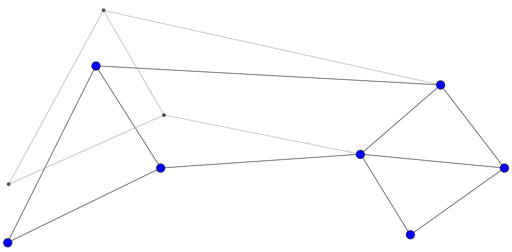
As we saw, 2-connectivity is necessary for global rigidity in $\mathbb{R}^1$, however, a folklore result shows that the converse is also true.

Theorem 2.2 *A graph $G = (V, E)$ on at least 3 vertices is globally rigid in $\mathbb{R}^1$ if and only if G is 2-connected.* □

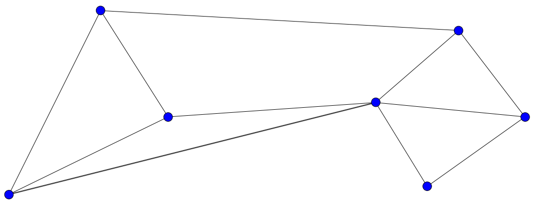
For the characterization of globally rigid graphs in the plane, we shall introduce the concept of redundant rigidity. A graph $G = (V, E)$ is **redundantly rigid** in $\mathbb{R}^2$, if $G - e$ is rigid in $\mathbb{R}^2$ for each $e \in E$. That is, the graph remains rigid in $\mathbb{R}^2$ after deleting any of its edges. A famous result of Jackson and Jordán characterizes global rigidity in $\mathbb{R}^2$.

Theorem 2.3 ([32]) *Let G be a simple graph on at least four vertices. Then G is globally rigid in $\mathbb{R}^2$, if and only if G is redundantly rigid in $\mathbb{R}^2$ and 3-connected.* $\square$

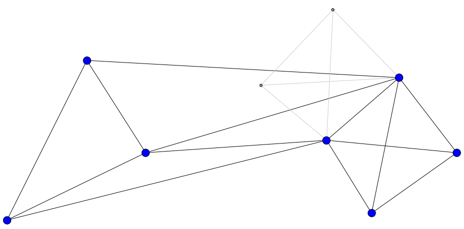
As an example, let us demonstrate the various rigidity properties in $\mathbb{R}^2$ in Figure 2.1. One can see that in Figure 2.1 a) there exists a continuous motion of the vertices preserving all edge lengths between the two shown arrangements of the frameworks, while no such motion is possible in Figure 2.1 b). Figure 2.1 c) shows a redundantly rigid graph, which is not globally rigid, as it is not 3-connected. Another possible equivalent realization of the same framework is also presented. Figure 2.1 d) shows a globally rigid graph. This is the unique framework up to isometries that the distance constraints of these bars determine.



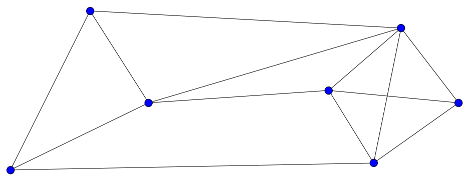
(a) A non-rigid graph in $\mathbb{R}^2$ with a possible movement of its framework.



(b) A rigid graph in $\mathbb{R}^2$.



(c) A redundantly rigid, but not globally rigid graph in $\mathbb{R}^2$ with two equivalent realizations of its framework.



(d) A globally rigid graph in $\mathbb{R}^2$.

Figure 2.1: Illustration of various rigidity properties in $\mathbb{R}^2$.

For the sake of completeness, we shall mention three more results concerning the connections between rigidity and connectivity. First, we shall mention that redundant rigidity and high connectivity are necessary for higher dimensional global rigidity (similarly to Theorem 2.3). On the other hand, in contrast to Theorem 2.3, they are not sufficient in higher dimensions.

Theorem 2.4 *[27] If G is globally rigid in $\mathbb{R}^d$, then either G is a complete graph with at most $d+1$ vertices, or G is $(d+1)$-connected and redundantly rigid in $\mathbb{R}^d$.* $\square$

Another important fact is that rigidity implies some connectivity properties. This is a folklore result, the proof of which can be found in e. g. [40].

Proposition 2.5 *If $G = (V, E)$ is a rigid graph in $\mathbb{R}^2$ for which $|V| \geq 3$, then G is 2-connected.* □

Conversely, Lovász and Yemini proved that high-enough connectivity implies rigidity, moreover, it implies global rigidity.

Theorem 2.6 *[57] Every 6-connected graph is globally rigid in $\mathbb{R}^2$.* □

There are much more results in combinatorial rigidity, even in connection with connectivity, however, this is sufficient for us for now. For more details on these topics, the reader is kindly referred to [24, 40].

2.2 Count matroids

Sparsity properties are important in rigidity theory as they can be used in the characterization of many rigidity classes. For example, as we saw, the rigid graphs in $\mathbb{R}^2$ are exactly the ones that contain a tight spanning subgraph. This sparsity condition can be generalized, as follows. This section is partially based on [60] (in Hungarian).

Let k be a positive integer and ℓ be an integer such that $\ell < 2k$. A (multi)graph $G = (V, E)$ is called **(k, ℓ)-sparse** if $i_G(X) \leq k|X| - \ell$ for all $X \subseteq V$ for which $i_G(X) > 0$. A (k, ℓ)-sparse graph is called **(k, ℓ)-tight** if $|E| = k|V| - \ell$. Due to its connections and extensive usage in rigidity theory, which we present later in this section, a graph G is called **(k, ℓ)-rigid** if G has a (k, ℓ)-tight spanning subgraph. We call an edge e of G a **(k, ℓ)-redundant edge** if $G - e$ is (k, ℓ)-rigid. A graph G is a **(k, ℓ)-redundant graph** if each edge of G is (k, ℓ)-redundant. Note here that the tight graphs of Theorem 2.1 are exactly the $(2, 3)$-tight graphs, while the redundantly rigid graphs in $\mathbb{R}^2$ are the $(2, 3)$-redundant graphs.

(Rarely – but not unprecedentedly, see for example in Chapter 4 – an edge e can be called (k, ℓ)-redundant, even if G is not rigid. In this case, it means that the number of edges in the maximal (k, ℓ)-sparse subgraph of G is equal to the number of edges in the maximal (k, ℓ)-sparse subgraph of $G-e$. Nonetheless, a graph that is not (k, ℓ)-rigid cannot be called (k, ℓ)-redundant.)

There exist some possible generalizations of (k, ℓ)-sparsity. One includes hypergraphs instead of graphs. A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is called **(k, ℓ)-sparse** if $i_{\mathcal{H}}(X) \leq k|X| - \ell$ holds for all $X \subseteq V$ for which $i_{\mathcal{H}}(X) > 0$. A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is called **(k, ℓ)-tight** if it is sparse and $|\mathcal{E}| = k|V| - \ell$. We call a hypergraph **(k, ℓ)-rigid** if it contains a spanning (k, ℓ)-tight subhypergraph. We can define redundancy of hyperedges and **(k, ℓ)-redundant hypergraphs** as a direct generalization of (k, ℓ)-redundant edges and graphs.

In another generalization, we still have the graph $G = (V, E)$, but we have a function on the vertices. Let $\ell \in \mathbb{Z}$ and $m : V \to \mathbb{Z}_+$ be a map. A graph $G = (V, E)$ is called **(m, ℓ)-sparse** if $i_G(X) \leq \widetilde{m}(X) - \ell$ holds for every $X \subseteq V$ for which $\widetilde{m}(X) - \ell \geq 0$ and $i_G(X) = 0$ otherwise, where $\widetilde{m}(X) := \sum_{v \in X} m(v)$. One may notice that any (k, ℓ)-sparse graph is (m, ℓ)-sparse for $m \equiv k$. **(m, ℓ)-tight/rigid/redundant** graphs can be defined by extending the definitions of (k, ℓ)-tight/rigid/redundant graphs.

We will also use the combinations of these two generalizations, as follows. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$, let $\ell \in \mathbb{Z}$ and $m : V \to \mathbb{Z}_+$ be a map. $\mathcal{H}$ is called **(m, ℓ)-sparse** if $i_{\mathcal{H}}(X) \leq \widetilde{m}(X) - \ell$ holds for every $X \subseteq V$ for which $\widetilde{m}(X) - \ell \geq 0$ and $i_{\mathcal{H}}(X) = 0$ otherwise. **(m, ℓ)-tight/rigid/redundant** hypergraphs can be defined by extending the definitions of (m, ℓ)-tight/rigid/redundant graphs.

It is known that the edge sets/hyperedge sets of the (m, ℓ)-sparse/(k, ℓ)-sparse subgraphs/subhypergraphs of a given graph/hypergraph correspond to the independent sets of the, so-called **(m, ℓ)-sparsity/(k, ℓ)-sparsity matroids** or **count matroids** (see [17, Section 13.5], [56], and [78, Appendix A]).

These matroids often appear in applications regarding rigidity theory. The most notable is the characterization of rigidity in $\mathbb{R}^2$ by Theorem 2.1. However, sparsity matroids are used in other similar results, as well. For example, the rigidity and global rigidity of graphs on a cylinder $C^2 \subset \mathbb{R}^3$ have been characterized by Nixon, Owen and Power [63] and Jackson and Nixon [35]. Rigidity in any space, specifically in any surface can be defined similarly to the rigidity in $\mathbb{R}^d$, only the movement of the framework is constrained to the given space (surface). For exact definitions of rigidity on $C^2 \subset \mathbb{R}^3$, see [63].

Theorem 2.7 ([63]) *A simple graph is rigid on the cylinder $C \subset \mathbb{R}^3$ if and only if it is $(2, 2)$-rigid.* □

Theorem 2.8 ([35]) *A simple graph is globally rigid on the cylinder $C \subset \mathbb{R}^3$ if and only if it is $(2, 2)$-redundant and 2-connected.* □

In these cases the characterizations use *simple* $(2, 2)$-rigid (and $(2, 2)$-redundant) graphs. Note that without the simplicity condition a $(2, 2)$-tight graph may have parallel edges (which could be meaningless from a rigidity point of view).

We note that the generic rigidity (and generic global rigidity, respectively) of body-bar and body-hinge frameworks in $\mathbb{R}^d$ have been characterized by $(\binom{d+1}{2}, \binom{d+1}{2})$-rigidity (and $(\binom{d+1}{2}, \binom{d+1}{2})$-redundancy, respectively) of a corresponding graph in [34, 41, 71, 73, 77]. The exact definitions of these rigidity properties can be found in [41, 77].

Besides the applications in rigidity, there are several other problems that may be characterized by using count matroids. The most well-known special case is the spanning tree. Given a graph G the $(1,1)$-sparse edge sets are exactly the edge sets of forests in G thus G is $(1,1)$-tight if and only if it is a spanning tree. Therefore, the graphic matroid is exactly the $(1,1)$-sparsity matroid. Another famous example is the (k,k)-sparsity matroid. A graph G is (k,k)-rigid if and only if G contains k edge-disjoint spanning trees by the fundamental result of Nash-Williams [62]. With the count matroids in hypergraphs, we can also describe the transversal matroid. Given a bipartite graph $G = (S, T, E)$, we call an $I \subseteq S$ *matchable* if there exists a matching in G that matches I. Consider the matroid with base set S and matchable sets as independent sets. This is exactly the $(1,0)$-sparsity matroid on the hypergraph that we get on the vertex set S with the hyperedges formed by the neighbors of every $t \in T$ [17].

Let us now list some basic definitions and properties concerning the sparsity matroid without proofs. We refer to [40, 78] for more details.

The spanning (m,ℓ)-tight subhypergraphs of an (m,ℓ)-rigid hypergraph form the bases of the corresponding sparsity matroid, while a subhypergraph which forms a circuit in this matroid is called an **(m,ℓ)-M-circuit**. In other words, a hypergraph C is an (m,ℓ)-M-circuit if it is not (m,ℓ)-sparse and $C - e$ is (m,ℓ)-sparse for every single hyperedge e of C. In particular, if $\mathcal{H}$ is (m,ℓ)-tight and $e = ij$ is a new (graph) edge, then $\mathcal{H} + e$ has a unique (m,ℓ)-M-circuit, denoted by $\boldsymbol{\mathcal{C}_{\mathcal{H}}^{(m,\ell)}(ij)}$ or $\boldsymbol{\mathcal{C}_{\mathcal{H}}^{(m,\ell)}(e)}$. This circuit contains e. $(V(\mathcal{C}_{\mathcal{H}}^{(m,\ell)}(e)), \mathcal{E}(\mathcal{C}_{\mathcal{H}}^{(m,\ell)}(e)) - e)$ forms an (m,ℓ)-tight subhypergraph of $\mathcal{H}$, that we call $\boldsymbol{\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(e)}$ or $\boldsymbol{\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(ij)}$. (Note that this definition may also be extended to the case where we add a new hyperedge of any size to an (m,ℓ)-tight hypergraph, however, in this dissertation we only consider additional graph edges or hyperedges containing exactly 2 vertices.) For the sake of convenience, we do not distinguish a hypergraph from its edge set, that is, $\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(e) = \mathcal{E}(\mathcal{C}_{\mathcal{H}}^{(m,\ell)}(e)) - e$. (Note that $C_{\mathcal{H}}^{(m,\ell)}(ij)$ consists of the single edge ij when $m(i) = m(j) = \ell = 0$, and hence $\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(ij)$ consists of only two isolated vertices i and j in this case. Nonetheless, this subhypergraph of $\mathcal{H}$ is (m,ℓ)-tight.) For every hyperedge e' of $C_{\mathcal{H}}^{(m,\ell)}(ij)$, $\mathcal{H}' = \mathcal{H} + ij - e'$ is also (m,ℓ)-sparse and the unique (m,ℓ)-M-circuit of $\mathcal{H}' + e'$ is again $C_{\mathcal{H}}^{(m,\ell)}(ij)$. Moreover, if $e'' \notin C_{\mathcal{H}}^{(m,\ell)}(ij)$, then $\mathcal{H}' + ij - e''$ is not

(m,ℓ)-sparse.

2.3 Preliminary lemmas

In this section we aim to collect the fundamental lemmas and results that are extensively used in our work. We show their most general form, so that we can use them in various circumstances.

It follows from the definition that an (m,ℓ)-tight subhypergraph of an (m,ℓ)-sparse hypergraph is always an induced subhypergraph. Therefore, if $\mathcal{T}_1 = (V_1, \mathcal{E}_1)$ and $\mathcal{T}_2 = (V_2, \mathcal{E}_2)$ are both (m,ℓ)-tight subhypergraphs of an (m,ℓ)-sparse hypergraph $\mathcal{H}$, then $\mathcal{T}_1 \cap \mathcal{T}_2 = (V_1 \cap V_2, \mathcal{E}_1 \cap \mathcal{E}_2)$ is also an induced subhypergraph of $\mathcal{H}$. Moreover, with standard submodular techniques, we can prove the following.

Lemma 2.9 *Let $\mathcal{H} = (V, \mathcal{E})$ be an (m,ℓ)-sparse hypergraph, and let $\mathcal{T}_1 = (V_1, \mathcal{E}_1)$ and $\mathcal{T}_2 = (V_2, \mathcal{E}_2)$ be (m,ℓ)-tight subhypergraphs of $\mathcal{H}$. If $\widetilde{m}(V_1 \cap V_2) \geq \ell$, then $\mathcal{T}_1 \cup \mathcal{T}_2$ is an (m,ℓ)-tight hypergraph and $d_{\mathcal{H}}(V_1, V_2) = 0$. Furthermore, if $|V_1 \cap V_2| \geq 1$, then $\mathcal{T}_1 \cap \mathcal{T}_2$ is also (m,ℓ)-tight.*

Observation 2.10 *If $m(u) + m(v) \geq \ell$ for each pair $u, v \in V$, then $\widetilde{m}(V_1 \cap V_2) \geq \ell$ always holds when $|V_1 \cap V_2| \geq 2$, ensuring the (m,ℓ)-tightness of $\mathcal{T}_1 \cup \mathcal{T}_2$.*

With the help of Observation 2.10 we will often apply Lemma 2.9 to cases when $|V_1 \cap V_2| \geq 2$. Also, if $\ell \leq 0$, then $\widetilde{m}(V_1 \cap V_2) \geq \ell$ for arbitrary $V_1, V_2 \subseteq V$, since $m \geq 0$.

Proof (of Lemma 2.9) As $\mathcal{T}_1$ and $\mathcal{T}_2$ are (m,ℓ)-tight,

$$\begin{aligned} i_{\mathcal{H}}(V_1 \cup V_2) + i_{\mathcal{H}}(V_1 \cap V_2) &= i_{\mathcal{H}}(V_1) + i_{\mathcal{H}}(V_2) + d_{\mathcal{H}}(V_1, V_2) \geq i_{\mathcal{H}}(V_1) + i_{\mathcal{H}}(V_2) \\ &= \widetilde{m}(V_1) - \ell + \widetilde{m}(V_2) - \ell = \widetilde{m}(V_1 \cup V_2) - \ell + \widetilde{m}(V_1 \cap V_2) - \ell. \end{aligned}$$

As $\widetilde{m}(V_1 \cap V_2) \geq \ell$ and $\mathcal{H}$ is an (m,ℓ)-sparse hypergraph,

$$i_{\mathcal{H}}(V_1 \cup V_2) + i_{\mathcal{H}}(V_1 \cap V_2) \leq \widetilde{m}(V_1 \cup V_2) - \ell + \widetilde{m}(V_1 \cap V_2) - \ell$$

holds. Thus equality stands in all the above inequalities. This proves both statements. □

The main property of $\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(ij)$ is the following.

Lemma 2.11 *Let $\mathcal{H} = (V, \mathcal{E})$ be an (m,ℓ)-sparse hypergraph and let $i, j \in V$. Assume that $\mathcal{H} + ij$ is not (m,ℓ)-sparse. If $\mathcal{H}' = (V', \mathcal{E}')$ is an (m,ℓ)-tight subhypergraph of $\mathcal{H}$ with $i, j \in V'$, then*

$\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(ij) \subseteq \mathcal{H}'$. *Hence* $\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(ij) = \bigcap\{\mathcal{T}_h : \mathcal{T}_h$ *is an* (m,ℓ)*-tight subhypergraph of* $\mathcal{H}$ *including* i *and* $j\}$.

Proof Suppose that $i, j \in V'$. As $\mathcal{H}' + ij$ cannot be (m,ℓ)-sparse, there exists an (m,ℓ)-M-circuit C' in $\mathcal{H}' + ij$ such that $i, j \in V(C')$. On the other hand, C' is an (m,ℓ)-M-circuit in $\mathcal{H}$, as well, thus $\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(ij) = C' - ij$ by definition, therefore $\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(ij) \subseteq \mathcal{H}'$. Hence $\mathcal{T}_{\mathcal{H}}^{(m,\ell)}(ij)$ is the unique smallest (m,ℓ)-tight set containing i and j. $\square$

Let $\boldsymbol{R_{\mathcal{H}}^{(m,\ell)}(i_1j_1,\dots,i_rj_r)}$ denote the subhypergraph of $\mathcal{H} = (V, \mathcal{E})$ induced by the hyperedges of $\mathcal{E}$ which are (m,ℓ)-redundant in $\mathcal{H} \cup \{i_1j_1,\dots,i_rj_r\}$ where $i_1,\dots,i_r, j_1,\dots,j_r \in V$. For the sake of simplicity, when the hypergraph $\mathcal{H}$ or (m,ℓ) is clear from the context, we will omit the subscript $\mathcal{H}$ or superscript (m,ℓ), respectively. Note that $R(ij) = \mathcal{T}(ij)$ for any $i, j \in V$. The following lemma extends this simple fact by generalizing [22, Lemma 4].

Lemma 2.12 *If* $\mathcal{H}$ *is an* (m,ℓ)*-tight hypergraph then*

$$R(i_1j_1,\dots,i_rj_r) = \mathcal{T}(i_1j_1) \cup \dots \cup \mathcal{T}(i_rj_r).$$

Proof Since $R(ij) = \mathcal{T}(ij)$, $\mathcal{T}(i_1j_1) \cup \dots \cup \mathcal{T}(i_rj_r) \subseteq R(i_1j_1,\dots,i_rj_r)$. For the other direction, let $e \in R(i_1j_1,\dots,i_rj_r)$ be an arbitrary hyperedge. Now, $\mathcal{H} - e$ is (m,ℓ)-sparse and $|\mathcal{E} - e| = \widetilde{m}(V) - \ell - 1$. $\mathcal{H} \cup \{i_1j_1,\dots,i_rj_r\} - e$ is (m,ℓ)-rigid, hence $\mathcal{E} \cup \{i_1j_1,\dots,i_rj_r\} - e$ has rank of $\widetilde{m}(V) - \ell$ in the (m,ℓ)-sparsity matroid. Thus there is a hyperedge f in $\{i_1j_1,\dots,i_rj_r\}$ for which $\mathcal{E} - e + f$ is a base of the (m,ℓ)-sparsity matroid. Since $\mathcal{E} - e + f$ is independent in the (m,ℓ)-sparsity matroid, we must have $e \in \mathcal{T}(f)$. $\square$

Lemma 2.13 *Let* $\mathcal{H} = (V, \mathcal{E})$ *be an* (m,ℓ)*-tight hypergraph with* $|V| \geq 3$ *and let* $v \in V$ *with* $0 < m(v) < \ell$. *Then no loop is incident with* v *and* $d(v) \geq m(v)$.

Proof Note that v cannot induce any loop by the (m,ℓ)-sparsity condition. Hence the (m,ℓ)-tightness of $\mathcal{H}$ and the (m,ℓ)-sparsity of $V - v$ imply that $\widetilde{m}(V) - \ell = i(V) = d(v) + i(V - v) \leq d(v) + \widetilde{m}(V - v) - \ell = \widetilde{m}(V) - \ell - m(v) + d(v)$. Thus $d(v) \geq m(v)$. $\square$

Notice that Lemma 2.13 also means that v is connected to $V - v$.

Chapter 3

(k, ℓ)-redundant augmentation

In some applications regarding (k, ℓ)-rigid graphs it may be required to have redundancy in the system. Hence, it is interesting to consider what is the minimum number of edges to add to a graph so that it becomes (k, ℓ)-redundant. Moreover, this question gains lots of its importance due to its connection with the global rigidity augmentation problem that we shall investigate in detail in Chapter 5. In a nutshell, in Chapter 5 we aim to make a rigid graph redundantly rigid and 3-connected. As we shall see in Chapter 4, redundant rigidity together with vertex connectivity has deep connections with hypergraphs. Therefore, as a preparation for later results, we shall consider the "(k, ℓ)-redundant augmentation problem" for hypergraphs instead of graphs in this chapter. Obviously, this includes the case of the input being a graph, as a special case.

Throughout this chapter we consider the following problem that we call the **general $(\boldsymbol{k}, \boldsymbol{\ell})$-redundant augmentation problem**.

Problem 1 *Let k and ℓ be integers with $k > 0$ and $\ell < 2k$ and let $\mathcal{H} = (V, \mathcal{E})$ be a (k, ℓ)-rigid hypergraph. Find a graph $H = (V, F)$ on the same vertex set with minimum number of edges, such that $\mathcal{H} \cup H = (V, \mathcal{E} \cup F)$ is (k, ℓ)-redundant.*

We call the special case of this problem, where the input hypergraph $\mathcal{H}$ is (k, ℓ)-tight, the **reduced $(\boldsymbol{k}, \boldsymbol{\ell})$-redundant augmentation problem**. In this chapter, we give a min-max theorem for the reduced problem for each pair of k and ℓ (where $\ell < 2k$, as usual). We also show how this method can be extended to solve the general problem when $\ell \leq k$. In contrast, we show that the general problem is NP-hard, whenever $\ell > k$. In Section 8.3 we give an $O(|V|^2)$

time algorithm for reaching the optimal solution in arguably the most important special case of the presented problem.

There are several special pairs of k and ℓ for which these problems were already investigated. The general problem for $(1,1)$-rigid *graphs* is exactly the well-studied 2-edge-connectivity augmentation problem solved by Eswaran and Tarjan [14]. The general problem for *graphs* where $k=\ell$ was solved by Frank and Király [19] who gave a polynomial algorithm to augment an arbitrary graph to an h-times (k,k)-redundant graph using polyhedral techniques (where h-times (k,k)-redundant means that it remains (k,k)-rigid after deleting any set of its edges of cardinality h). García and Tejel [22] showed that the general problem is NP-hard for $(2,3)$-rigid graphs but can be solved in polynomial time for $(2,3)$-tight graphs. We use similar techniques to [22], however, our method is based on a new min-max theorem for the reduced problem, as follows.

For a (k,ℓ)-tight hypergraph $\mathcal{H}=(V,E)$, we call a set of vertices $\emptyset \neq C \subsetneq V$ **(k,ℓ)-co-tight** in $\mathcal{H}$ if the complement of C induces a (k,ℓ)-tight subhypergraph of $\mathcal{H}$. We will see that it is possible that there is no (k,ℓ)-co-tight set in $\mathcal{H}$ in which case $\mathcal{H}+uv$ is (k,ℓ)-redundant for any $u,v\in V$. The main result of this chapter is the following min-max theorem which is illustrated by Figure 3.1.

Theorem 3.1 *Let $\mathcal{H}=(V,\mathcal{E})$ be a (k,ℓ)-tight hypergraph on at least k^2+2 vertices. If there exists any (k,ℓ)-co-tight set in $\mathcal{H}$, then*

$$\begin{aligned}&\min\{|F| : H=(V,F) \textit{ is a graph for which } \mathcal{H}\cup H \textit{ is } (k,\ell)\textit{-redundant}\}\\ =\ &\max\left\{\left\lceil\frac{|\mathcal{C}|}{2}\right\rceil : \mathcal{C} \textit{ is a family of pairwise disjoint } (k,\ell)\textit{-co-tight sets in } \mathcal{H}\right\}.\end{aligned}$$

To obtain the solution for the general problem by using the reduced problem when $k\geq\ell$, we need some extra work. The idea of our method for this case comes from Jackson and Jordán [33] who proved that the (k,k)-redundant edges of a (k,k)-rigid graph $\bar{G}$ form induced subgraphs of $\bar{G}$ with disjoint vertex sets. If we contract these subgraphs into single vertices it is proven [33] that the resulting graph is (k,k)-tight for which we can use the algorithm of the reduced problem. The problem is not that simple though if we consider $k>\ell$.

We will need the concept of (m,ℓ)-rigidity introduced in Section 2.2, and a little extra. First remember that given $\ell\in\mathbb{Z}$ and $m: V\to\mathbb{Z}_+$ a hypergraph $\mathcal{H}=(V,\mathcal{E})$ is called (m,ℓ)-sparse if $i_{\mathcal{H}}(X)\leq\widetilde{m}(X)-\ell$ holds for every $X\subseteq V$ for which $\widetilde{m}(X)-\ell\geq 0$ and $i_{\mathcal{H}}(X)=0$ otherwise. To ensure that the condition $\widetilde{m}(X)-\ell\geq 0$ holds (usually with strict inequality) for each set

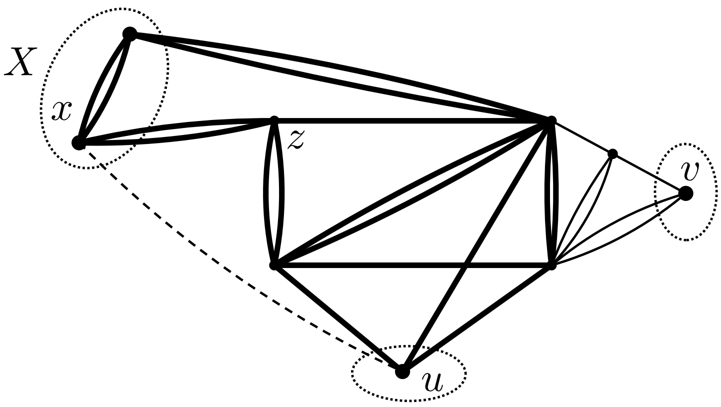


Figure 3.1: The solid edges form a $(3,4)$-tight graph G. The three highlighted sets are pairwise disjoint $(3,4)$-co-tight sets in G, hence no single edge augments G to a $(3,4)$-redundant graph by Theorem 3.1. However, it is easy to check that the addition of the two edges uv and ux makes all edges of G $(3,4)$-redundant, and hence this gives rise to an optimal solution of the reduced problem on G. For example, after the addition of the dashed edge ux we get a $(3,4)$-rigid graph where ux and the bold edges are the only $(3,4)$-redundant edges.

$X \subseteq V$ with $|X| \geq 2$, we have the following assumption throughout this chapter.

(A0) $\ell \in \mathbb{Z}$ and $m : V \to \mathbb{Z}_+$ such that, for each pair $u, v \in V$ with $u \neq v$, either $m(u)+m(v) > \ell$ or $m(u) = m(v) = \ell = 0$ holds.

Now we can refer to Observation 2.10 claiming that following (A0), in Lemma 2.9 $\mathcal{T}_1 \cup \mathcal{T}_2$ is an (m,ℓ)-tight hypergraph if $|V_1 \cap V_2| \geq 2$.

Note that, when $m \equiv k$ (thus an (m,ℓ)-sparse hypergraph is (k,ℓ)-sparse) (A0) holds by our assumption that $\ell < 2k$. **(m,ℓ)-co-tight** sets can be defined by extending the definitions of (k,ℓ)-co-tight sets. For simplicity, we call a set $X \subseteq V$ **(m,ℓ)-tight** (**(k,ℓ)-tight**, respectively) **in $\mathcal{H}$** if $\mathcal{H}[X]$ is (m,ℓ)-tight ((k,ℓ)-tight, respectively). Note that (A0) implies that each (m,ℓ)-tight set X in $\mathcal{H}$ of cardinality at least two induces at least one hyperedge in $\mathcal{H}$, except when $\ell = 0$ and $m(x) = 0$ for each $x \in X$.

We will see in Section 3.1 that, after using the contraction idea that we presented for (k,ℓ)-rigid hypergraphs with $k > \ell$, the resulting hypergraph $\mathcal{H}' = (V', \mathcal{E}')$ on which we need to solve the reduced problem is (m',ℓ')-tight for some $m' : V' \to \mathbb{Z}_+$ and $\ell' \in \mathbb{Z}$ for which (A0) holds. Hence we will prove an extension of Theorem 3.1 for (m,ℓ)-tight hypergraphs in Section 3.3 and we will give our algorithm for the reduced problem in Section 8.3 for (k,ℓ)-tight inputs.

The first ideas of this chapter including co-tight sets and a min-max theorem were already presented in the conference and proceedings of HJ2019 [61], which was followed by a technical report on a generalized version of the problem [45]. A simplified version of these results was presented in the IPCO20 conference [46], and finally, the full paper is published in Mathematical Programming B in 2021 [51]. This chapter is based on our results published in [51]. We note that in this chapter we consider more general settings, namely we consider the problem on hypergraphs. This possibility was already mentioned but not yet presented in detail in [51].

Notation For all the hypergraphs in this chapter we allow parallel hyperedges and loops. For simplicity, we do not distinguish hypergraphs and their hyperedge sets when it is clear from the context. When the pair (m, ℓ) ((k, ℓ), respectively) is clear from the context we may omit the prefix (m, ℓ) ((k, ℓ), respectively) from the notions above. Also, when it is clear from the context, we omit the subscript $\mathcal{H}$ from the notation.

3.1 The reduction of the general problem

García and Tejel [22] showed that the general augmentation problem is NP-hard for $(2, 3)$-rigid graphs by reducing it to the set cover problem. Based on their method we will prove in Section 3.4 the NP-hardness of the general problem on graphs whenever $k < \ell$. In this section, we show that in any other case (that is, if $\ell \leq k$), there exists a reduction from the general problem to the reduced problem which we solve in Section 3.3. Moreover, we give our reduction for all (m, ℓ)-rigid hypergraphs for which $m(v) \geq \ell$ for all $v \in V$ (or $m \geq \ell$ for the sake of brevity). The main advantage of (m, ℓ)-tight hypergraphs over (k, ℓ)-tight hypergraphs is that an (m, ℓ)-tight hypergraph remains (m', ℓ')-tight after contracting an (m, ℓ)-tight subhypergraph for some pair (m', ℓ'), as follows.

Lemma 3.2 *Let $\mathcal{H} = (V, \mathcal{E})$ be an (m, ℓ)-tight hypergraph with $m \geq \ell$. Suppose that $T \subsetneq V$ is an (m, ℓ)-tight set in $\mathcal{H}$. Let t' be the new vertex that arises after contracting T in $\mathcal{H}$. Let $\ell' = \max(\ell, 0)$ and let $m' : V(\mathcal{H}/T) \to \mathbb{Z}_+$ be a map such that $m'(v) = m(v)$ when $v \in V(\mathcal{H}/T) \cap V$ and $m'(t') = \ell'$.*

(a) A set $S \subseteq V(\mathcal{H}/T)$ containing t' induces an (m', ℓ')-tight subhypergraph of $\mathcal{H}/T$ if and only if $(S - t') \cup T$ is (m, ℓ)-tight in $\mathcal{H}$. In particular, $\mathcal{H}/T$ is (m', ℓ')-tight.

(b) A set $S \subseteq V(\mathcal{H}/T) - t'$ is (m', ℓ')-tight in $\mathcal{H}/T$ if and only if either $\ell \geq 0$ and S is (m, ℓ)-tight in $\mathcal{H}$, or $\ell < 0$ and $S \cup T$ is (m, ℓ)-tight in $\mathcal{H}$ such that $d(S, T) = 0$.

Proof First, it is easy to see that, if the pair (m, ℓ) satisfies (A0), then so does (m', ℓ'), as the values of m' are equal with the values of m, unless this latter is ℓ'.

Let us show that $\mathcal{H}/T$ is (m', ℓ')-sparse. Let $X \subseteq V(\mathcal{H}/T)$. Assume first that $t' \in X$. Then $i_{\mathcal{H}/T}(X) = i_{\mathcal{H}}((X - t') \cup T) - i_{\mathcal{H}}(T) \leq \widetilde{m}((X - t') \cup T) - \ell - (\widetilde{m}(T) - \ell) = \widetilde{m}(X - t') = \widetilde{m}'(X) - m'(t') = \widetilde{m}'(X) - \ell'$ as T is (m, ℓ)-tight. If $t' \notin X$ and $\ell \geq 0$, then $i_{\mathcal{H}/T}(X) = i_{\mathcal{H}}(X) \leq \max(\widetilde{m}(X) - \ell, 0) = \max(\widetilde{m}'(X) - \ell', 0)$. If $t' \notin X$ and $\ell < 0$, then $i_{\mathcal{H}/T}(X) = i_{\mathcal{H}}(X) \leq i_{\mathcal{H}}(X \cup T) - i_{\mathcal{H}}(T) \leq \widetilde{m}(X \cup T) - \ell - (\widetilde{m}(T) - \ell) = \widetilde{m}(X) = \widetilde{m}'(X) - \ell'$. Hence $\mathcal{H}/T$ is indeed (m', ℓ')-sparse.

(a) Assume that $t' \in S \subseteq V(\mathcal{H}/T)$. The tightness of S in $\mathcal{H}/T$ means that $i_{\mathcal{H}/T}(S) = \widetilde{m}'(S) - \ell' = \widetilde{m}(S - t')$. Since $i_{\mathcal{H}}(T) = \widetilde{m}(T) - \ell$, replacing t' with T results $i_{\mathcal{H}}((S - t') \cup T) = \widetilde{m}((S - t') \cup T) - \ell$. The proof of the other direction follows from the same equations the other way around.

(b) Here first assume that $\ell \geq 0$ and $t' \notin S \subseteq V(\mathcal{H}/T)$. Suppose that S is (m', ℓ')-tight. Then $i_{\mathcal{H}/T}(S) = \widetilde{m}'(S) - \ell' = \widetilde{m}(S) - \ell$. On the other hand, the (m, ℓ)-tightness of S means that $i_{\mathcal{H}}(S) = \widetilde{m}(S) - \ell$. Since $i_{\mathcal{H}}(S) = i_{\mathcal{H}/T}(S)$, these two are equivalent.

Now consider the case of $\ell < 0$ and $t' \notin S \subseteq V(\mathcal{H}/T)$. Suppose that S is (m', ℓ')-tight. Then $i_{\mathcal{H}/T}(S) = \widetilde{m}'(S) - \ell' = \widetilde{m}(S) - 0$ (remember, $\ell' = \max(\ell, 0)$). Since T is (m, ℓ)-tight, $i_{\mathcal{H}}(S) = i_{\mathcal{H}/T}(S)$ and $\ell < 0$; $\widetilde{m}(S) + \widetilde{m}(T) - \ell \geq i_{\mathcal{H}}(S \cup T) = i_{\mathcal{H}}(S) + i_{\mathcal{H}}(T) + d_{\mathcal{H}}(S, T) \geq (\widetilde{m}(S) - 0) + (\widetilde{m}(T) - \ell) + 0$ follows, and hence equality must hold throughout, that is, $i_{\mathcal{H}}(S \cup T)$ is (m, ℓ)-tight and $d_{\mathcal{H}}(S, T) = 0$. The proof of the other direction is similar. □

Throughout this section, let $\boldsymbol{\mathcal{H} = (V, \mathcal{E})}$ denote an (m, ℓ)-rigid hypergraph with $m \geq \ell$ and $\boldsymbol{\mathcal{H}^* = (V, \mathcal{E}^*)}$ denote an (m, ℓ)-tight spanning subhypergraph of $\mathcal{H}$. Obviously, every hyperedge in $\mathcal{E} - \mathcal{E}^*$ is (m, ℓ)-redundant in $\mathcal{H}$. By Lemma 2.12, the (m, ℓ)-redundant hyperedges of $\mathcal{H}^*$ in $\mathcal{H}$ are the edges of $R_G(\mathcal{E} - \mathcal{E}^*) = \bigcup_{e \in \mathcal{E} - \mathcal{E}^*} \mathcal{T}_{\mathcal{H}^*}(e)$. Since we want to reduce the general augmentation problem to the reduced problem, we can assume that $\mathcal{E} - \mathcal{E}^* \neq \emptyset$. As $m \geq \ell$, these edges induce vertex disjoint (m, ℓ)-tight subhypergraphs in $\mathcal{H}^*$ by Lemmas 2.9 and 2.12. (Note that we have only one such subhypergraph which may be disconnected when $\ell \leq 0$, and these subhypergraphs are exactly the connected components (that is, the maximal connected subhypergraphs) of $R^{\mathcal{H}^*}(\mathcal{E} - \mathcal{E}^*)$ when $\ell > 0$.) By contracting each of these subhypergraphs to a single vertex and by defining $\boldsymbol{\ell'}$ as $\max(\ell, 0)$ and $\boldsymbol{m'}$ to be ℓ' on each of the contracted vertices and to be $m(v)$ on each non-contracted vertex v, we get the contracted hypergraph $\mathcal{H}' = (V', \mathcal{E}')$ along with the map $m' : V' \to \mathbb{Z}_+$. Notice that this contraction yields a natural surjective map,

$s : V \to V'$. By using Lemma 3.2(a) for each contracted tight subhypergraph of $\mathcal{H}^*$ sequentially it is easy to see that $\mathcal{H}'$ is (m', ℓ')-tight. We can state even more.

Proposition 3.3 *Let $X' \subseteq V'$ induce an (m', ℓ')-tight subhypergraph of $\mathcal{H}'$. Then $s^{-1}(X')$ induces an (m, ℓ)-tight subhypergraph in V, if $\ell \geq 0$. When $\ell < 0$, $s^{-1}(X') \cup V'$ induces an (m, ℓ)-tight subhypergraph (that is the union of $s^{-1}(X')$ with the vertices of the sole contracted (m, ℓ)-tight subhypergraph of $\mathcal{H}^*$).*

Conversely, if $X \subseteq V$ induces an (m, ℓ)-tight subhypergraph of $\mathcal{H}^*$, then $s(X)$ induces an (m', ℓ')-tight subhypergraph in V'. Moreover, if $\ell < 0$, then it contains the only contracted vertex (that is, the single element of $V' - V$).

Proof The first statement follows directly by using Lemma 3.2 (b) for each contracted tight subhypergraph of $\mathcal{H}^*$ sequentially. For the second statement, let $\bar{X}$ be the union of X and all the contracted components V_i's for which $X \cap V_i \neq \emptyset$. Then $\bar{X}$ induces an (m, ℓ)-tight subhypergraph of $\mathcal{H}^*$ by Lemma 2.9 and $s(\bar{X}) = s(X)$ is (m', ℓ)-tight by Lemma 3.2. (Note that if $\ell < 0$, then X must intersect the single contracted component V_1 since otherwise $X \cup V_1$ violates the (m, ℓ)-sparsity condition as $i_{\mathcal{H}^*}(X \cup V_1) \geq i_{\mathcal{H}^*}(X) + i_{\mathcal{H}^*}(V_1) = \widetilde{m}(X) - \ell + \widetilde{m}(V_1) - \ell > \widetilde{m}(X \cup V_1) - \ell$.) $\square$

If the union of some (m, ℓ)-tight subhypergraphs of $\mathcal{H}^*$ contains every hyperedge of $\mathcal{H}$ which is not (m, ℓ)-redundant, then by Proposition 3.3 the union of the corresponding (m', ℓ')-tight subhypergraphs of $\mathcal{H}'$ is $\mathcal{H}'$ itself. Hence the minimum number of edges that we need to make $\mathcal{H}$ (m, ℓ)-redundant is at least the minimum number of edges that we need to make G' (m', ℓ')-redundant. The following statement shows that these two values are equal.

Proposition 3.4 *Let F' denote an edge set of minimum cardinality on V' for which $\mathcal{H}' + F'$ is (m', ℓ')-redundant. Let F be an arbitrary edge set on V for which $\{s(u)s(v) : uv \in F\} = F'$. Then $\mathcal{H} + F$ is (m, ℓ)-redundant.*

Proof Given $u, v \in V$, $s(\mathcal{T}_{\mathcal{H}^*}^{(m,\ell)}(uv))$ is an (m', ℓ)-tight subhypergraph of $\mathcal{H}'$ that contains both $s(u)$ and $s(v)$ by Proposition 3.3. Thus it includes $\mathcal{T}_{\mathcal{H}'}^{(m',\ell')}(s(u)s(v))$ by Lemma 2.11. Since the image of each non-(m, ℓ)-redundant hyperedge of $\mathcal{H}$ is in $\mathcal{H}'$ and the subhypergraphs $\{\mathcal{T}_{\mathcal{H}'}^{(m',\ell)}(s(u)s(v)) : s(u)s(v) \in F'\}$ cover the hyperedge set of $\mathcal{H}'$, the subhypergraphs $\{\mathcal{T}_{\mathcal{H}^*}^{(m,\ell)}(uv) : uv \in F\}$ cover every non-(m, ℓ)-redundant hyperedge of $\mathcal{H}$. Hence $\mathcal{H} + F$ is (m, ℓ)-redundant by Lemma 2.12. $\square$

With Proposition 3.4, we have reduced the problem of augmenting an (m, ℓ)-rigid hypergraph to an (m, ℓ)-redundant hypergraph to the problem of augmenting an (m', ℓ')-tight hypergraph to an (m', ℓ')-redundant hypergraph. It is easy to check that (A0) still holds after the reduction. Note that when we get a solution for the arisen reduced problem, we can get back a solution to the original problem in linear time.

3.2 Preprocessing

Our goal is to provide a theoretic, as well as an algorithmic solution for the reduced problem for all (m, ℓ)-tight hypergraphs. The algorithmic solution of a variation on the restricted problem is shown in Chapter 8 in Section 8.3. The version presented there shows the algorithm on a special hypergraph class (the so-called (k, ℓ)-M-component hypergraphs). The complete solution covering every case in detail can be found in [51, Section 6]. Nonetheless, we need some extra conditions regarding the input for all cases. The first one constrains the function m.

(*) There exists a (universal) constant $c > 0$ such that $m(v) \leq c$ for every $v \in V$ for every hypergraph $\mathcal{H} = (V, \mathcal{E})$. We may also suppose that $|\ell| \leq c$.

We note that in the reduced problem we always deal with (k, ℓ)-tight hypergraphs or (m, ℓ)-tight hypergraphs that arise from a (k, ℓ)-rigid hypergraphs via contraction. In both cases k and ℓ are fixed constants thus $c = \max(k, \ell)$ fulfills (*) (remember that $m'(v) = k$ or $m'(v) = \max(\ell, 0)$ after the contraction steps). Observe that (*) implies that an (m, ℓ)-sparse graph on V has $O(|V|)$ hyperedges. This will be useful when we give the running time of our algorithms in Section 8.3.

Assuming (*) is also practical in the following condition. This new condition will help us present our results in a simpler form. Let us denote $V(\mathcal{T}(uv))$ by $\boldsymbol{V(uv)}$ throughout this whole chapter.

(A) Assuming (A0) and (*) for m and ℓ, $\mathcal{H} = (V, \mathcal{E})$ is an (m, ℓ)-tight hypergraph on at least four vertices such that either $\ell \leq 0$ or all of the following three conditions hold.

(A1) There exists no $v \in V$ such that $m(v) = 0$ with no hyperedge on v. (That is, $\mathcal{H}$ has no isolated vertex.)

(A2) There exist $u, v \in V$ such that $V(uv) \neq \{u, v\}$.

(A3) There exists no $v \in V$ such that $V(uv) = \{u, v\}$ for all $u \in V - v$ and $V - v$ induces an (m, ℓ)-tight hypergraph.

We note that the conditions of (A) automatically hold for (k, ℓ)-tight hypergraphs with sufficiently many vertices. First, obviously, (A1) is always true as $m(v) \neq 0$. (A2) holds, as there are $O(|V|)$ hyperedges in $\mathcal{H}$ and $O(|V|^2)$ vertex pairs, hence with sufficiently many vertices there exists at least one vertex-pair, say $u, v \in V$, so that $V(uv) \neq \{u, v\}$. Contradicting (A3) would mean that $i(V - v) = k(|V| - 1) - \ell$ thus as $i(V) - i(V - v) \geq |V - 1|$, $|V| - 1 \leq k$. To show the exact value of the sufficiently many vertices, let us generalize these ideas.

In fact, if we consider (m, ℓ)-tight hypergraphs, that satisfy (A0) and (*) with $\ell > 0$, we can show that (A2) and (A3) hold if $|V| \geq c^2 + 2$, where c is the constant from (*). Indeed, (A0) and $\ell > 0$ imply that if $\{u, v\}$ is an (m, ℓ)-tight set in $\mathcal{H}$, then it induces at least one hyperedge. Hence if (A2) does not hold, then $\mathcal{H}$ has at least $\frac{|V|(|V|-1)}{2}$ (hyper)edges. By the (m, ℓ)-tightness and (*) though, $\mathcal{H}$ has at most $c|V| - \ell \leq c|V| - 1$ hyperedges. By this, with a little computation follows that $|V| \leq c^2 + 1$. Now if (A3) does not hold, then by the (m, ℓ)-tightness of $\mathcal{H}[V - v]$ and $\mathcal{H}$, we get that $|V| - 1 \leq c$ (similarly to the (k, ℓ)-tight case). Hence if $|V| \geq c^2 + 2$, (A2) and (A3) both hold. Note that since c is considered to be a constant in our algorithms, the solution of the reduced problem on less than $c^2 + 2$ vertices can be provided in constant time. We also note that in some of our previous works we set this threshold to $c^2 + 3$, which is equally good for our purpose. Nonetheless, in this dissertation we stick to the value of $c^2 + 2$.

When $\mathcal{H}$ violates (A1), we may delete its isolated vertices with $m(v) = 0$ and solve the reduced problem for the arising hypergraph. The solution of our augmentation problem on this reduced input is a solution of our original problem, as $\mathcal{T}_{\mathcal{H}}(e) = \mathcal{T}_{\mathcal{H}-v}(e)$ is true for every edge e, since the sparsity matroids that arise in the two cases are identical.

As this reduction can be done in $O(|V|^2)$ running time, the assumption of (A) does not restrict the usability of our results for all (m, ℓ)-tight hypergraphs as we presented it in [51, Section 6].

3.3 The min-max theorem for the reduced problem

In this section we prove Theorem 3.1 and its extension for (m, ℓ)-tight hypergraphs for which, throughout this whole section, **we assume (A)**, including that $\mathcal{H} = (V, \mathcal{E})$ is an (m, ℓ)-tight hypergraph on at least four vertices. The generalization of Theorem 3.1 for (m,ℓ)-tight hypergraphs is the following.

Theorem 3.5 *Let $|V| \geq 2c$, where c is the universal constant from the condition (*).*

If there exists any (m,ℓ)-co-tight set in $\mathcal{H}$, then

$$\begin{aligned}&\min\{|F| : H=(V,F) \text{ is a graph for which } \mathcal{H}\cup H \text{ is } (m,\ell)\text{-redundant}\}\\ &= \max\left\{\left\lceil\frac{|\mathcal{C}|}{2}\right\rceil : \mathcal{C} \text{ is a family of pairwise disjoint } (m,\ell)\text{-co-tight sets}\right\}.\end{aligned}$$

Otherwise, $\mathcal{H}+uv$ is (m,ℓ)-redundant for every pair $u,v \in V$.

We shall note here that $c^2+2 > 2c$ always holds, as it is equivalent to $(c^2-1)+1 > 0$. Hence Theorem 3.5 is in fact a generalization of Theorem 3.1.

Recall that we call a set of vertices $\emptyset \neq C \subsetneq V$ (m,ℓ)-co-tight in $\mathcal{H}$ if $i_{\mathcal{H}}(V-C) = \widetilde{m}(V-C)-\ell$, that is, if the complement of C is an (m,ℓ)-tight set in $\mathcal{H}$. Equivalently (by $\widetilde{m}(V)-\ell = |\mathcal{E}| = e_{\mathcal{H}}(C)+i_{\mathcal{H}}(V-C)$), C is (m,ℓ)-co-tight if $e_{\mathcal{H}}(C) = \widetilde{m}(C)$. Note that for every $X \subset V$ for which $\widetilde{m}(V-X) \geq \ell$, $e_{\mathcal{H}}(X) \geq \widetilde{m}(X)$ holds by the sparsity of $V-X$. Inclusion-wise minimal (m,ℓ)-co-tight sets will be called **(m,ℓ)-MCT** sets. (See Figure 3.1 where the three highlighted sets $\{u\},\{v\}$, and X are the $(3,4)$-MCT sets of the graph G formed by the solid edges.) A useful observation on the MCT sets is the following.

Lemma 3.6 *Assume that $\ell > 0$. Let C be an (m,ℓ)-MCT set and let $v \in C$. Then $m(v) \neq 0$.*

Proof Suppose that $m(v)=0$. When $|C|=1$, the co-tightness of C implies that $d(v)=0$, contradicting (A1). When $|C| \geq 2$, then $i(V-(C-v)) \geq i(V-C) = \widetilde{m}(V-C)-\ell = \widetilde{m}(V-(C-v))-\ell$ follows by the co-tightness of C, hence $V-(C-v)$ is also tight in $\mathcal{H}$, contradicting the minimality of C. □

Observation 3.7 *Let $\mathcal{H}=(V,\mathcal{E})$ be an (m,ℓ)-tight hypergraph and let $C \subset V$ be an (m,ℓ)-co-tight set. If $\mathcal{H}\cup H$ is (m,ℓ)-redundant for a graph $H=(V,F)$, then there exists an edge $uv \in F$ such that $u \in C$ or $v \in C$.*

Proof If an edge e is not incident with at least one vertex in a co-tight set C, then $V(e) \subseteq V-C$ by Lemma 2.11. Thus no edge set that avoids C can augment $\mathcal{H}$ to (m,ℓ)-redundant by Lemma 2.12. □

This observation immediately implies that min $\geq$ max in Theorem 3.5 since each co-tight set in $\mathcal{C}$ must contain an end-vertex of an edge of H.

We say that a set U **covers** a set family $\mathcal{C}$, if $|U \cap C| \geq 1$ for every $C \in \mathcal{C}$. Let us denote the family of all (m, ℓ)-MCT sets of $\mathcal{H}$ by $\boldsymbol{\mathcal{C}^*}$ from now on.

Lemma 3.8 *Suppose that $U \subseteq V$ is a set that covers $\mathcal{C}^*$. If $V' \subseteq V$ is a set of vertices such that $U \subseteq V'$ and V' induces an (m, ℓ)-tight subhypergraph in $\mathcal{H}$, then $V' = V$. In particular, for two vertices $u, v \in V$, the set $\{u, v\}$ covers $\mathcal{C}^*$ if and only if $\mathcal{H} + uv$ is (m, ℓ)-redundant.*

Proof Let us suppose that there exists a tight set $V' \subsetneq V$ in $\mathcal{H}$ for which $U \subseteq V'$. Then $V - V'$ is co-tight by definition, and hence there exists an MCT set $C \in \mathcal{C}^*$ such that $C \subseteq V - V'$. However, as $U \subset V'$, this contradicts the assumption that $|U \cap C| \geq 1$ for every $C \in \mathcal{C}^*$. The 'only if' part of the second statement follows by the first one and the tightness of $V(uv) \ni u, v$. The converse direction follows by Observation 3.7. $\square$

Note that it is possible that there are no co-tight sets in a hypergraph $\mathcal{H}$. For example, the graph $K_6 - e$ is $(3, 4)$-tight and there are no $(3, 4)$-co-tight sets in it. By Lemma 3.8, if there are no MCT sets in $\mathcal{H}$ then $\mathcal{H} + uv$ is (m, ℓ)-redundant for any pair $u, v \in V$ which proves the last part of Theorem 3.5. Thus we may suppose that $\mathcal{H}$ contains at least one MCT set. We shall show that in this case min $\leq$ max also holds. This statement is obvious when the minimum is one since we assumed the existence of an MCT set in $\mathcal{H}$. Hence we may assume that the minimum is at least two. We will show that, in this case, the maximum in Theorem 3.5 is obtained by $\mathcal{C}^*$. The following statement on the structure of MCT sets shows that the members of $\mathcal{C}^*$ are 'usually' pairwise disjoint. It is an extension of a result of Jordán [40, Theorem 3.9.13] that states the same for $(2, 3)$-MCT sets.

Theorem 3.9 *Let $\mathcal{H} = (V, \mathcal{E})$ be an (m, ℓ)-tight hypergraph. Let $|V| \geq 2c$, where c is the universal constant from the condition (*). Then the members of $\mathcal{C}^*$ are either pairwise disjoint and $|\mathcal{C}^*| \geq 3$ or there exists a pair $u, v \in V$ such that $\mathcal{T}(uv) = \mathcal{H}$, that is, $\mathcal{H} + uv$ is (m, ℓ)-redundant.*

Let us list here two other variants of Theorem 3.9, where restricting some of its conditions can result in relaxing some other conditions. Different variants will be used in different context in this dissertation later. Hence we shall prove all of these theorems.

Theorem 3.10 *Let $\mathcal{H} = (V, \mathcal{E})$ be a (k, ℓ)-tight hypergraph. Let $|V| \geq 4$, and let $\ell \leq \frac{3}{2}k$. Then the members of $\mathcal{C}^*$ are either pairwise disjoint and $|\mathcal{C}^*| \geq 3$ or there exists a pair $u, v \in V$ such that $\mathcal{T}(uv) = \mathcal{H}$, that is, $\mathcal{H} + uv$ is (k, ℓ)-redundant.*

Theorem 3.11 *Let $\mathcal{H} = (V, \mathcal{E})$ be an (m, ℓ)-tight graph, that is, a hypergraph containing only hyperedges of size 2. Let $|V| \geq 4$. Then the members of $\mathcal{C}^*$ are either pairwise disjoint and $|\mathcal{C}^*| \geq 3$ or there exists a pair $u, v \in V$ such that $\mathcal{T}(uv) = \mathcal{H}$, that is, $\mathcal{H} + uv$ is (m, ℓ)-redundant.*

Note that in the most general case, when $\mathcal{H}$ is an (m, ℓ)-tight hypergraph and $|V| < 2c$, no similar theorem can hold, see a counterexample in Figure 3.2.

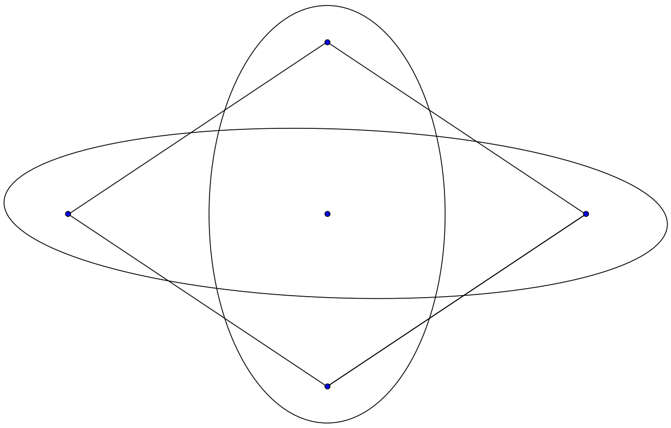


Figure 3.2: Let $\mathcal{H} = (V, \mathcal{E})$ be the (k, ℓ)-tight hypergraph above, with $k = 5$ and $\ell = 9$. Both ellipses represent 6 parallel hyperedges, all on the corresponding 3 vertices. All segments represent one single edge. One can see that $\mathcal{H}$ is (k, ℓ)-sparse and $|\mathcal{E}| = 16 = 5|V| - 9$. Any two vertices are in a tight set, every two vertices connected by an edge represent a maximal tight set and hence the MCT sets are intersecting.

The proofs of Theorem 3.9, 3.10 and 3.11 require more involved thoughts, hence we prove them separately in Subsection 3.3.1. With the help of Theorem 3.9 we can finish the proof of Theorem 3.5. For that we need to show that $\mathcal{H}$ can be augmented to an (m, ℓ)-redundant hypergraph by using a set of $\left\lceil \frac{|\mathcal{C}^*|}{2} \right\rceil$ edges when at least two edges are needed for the augmentation. Our plan is to add these new edges between the members of $\mathcal{C}^*$. By Theorem 3.9, we may suppose from now on that all the MCT sets of $\mathcal{H}$ are pairwise disjoint and $|\mathcal{C}^*| \geq 3$. In this case, the following stronger result also holds.

Lemma 3.12 *Let $\mathcal{H} = (V, \mathcal{E})$ be an (m, ℓ)-tight hypergraph and let C, K be two disjoint (m, ℓ)-MCT sets of $\mathcal{H}$. If $\widetilde{m}(V - (C \cup K)) \geq \ell$, then $\widehat{E}_{\mathcal{H}}(C) \cap \widehat{E}_{\mathcal{H}}(K) = \emptyset$.*

Proof By counting the hyperedges induced by $V-(C\cup K)$, we get that

$$i_{\mathcal{H}}(V-(C\cup K)) \le \widetilde{m}(V-(C\cup K)) - \ell = \widetilde{m}(V) - |\widehat{E}_{\mathcal{H}}(C)| - |\widehat{E}_{\mathcal{H}}(K)| - \ell$$

where the first inequality comes from the sparsity of $\mathcal{H}$ and the property that $\widetilde{m}(V-(C\cup K)) \ge \ell$, while the equalities hold because C and K are disjoint (m,ℓ)-MCT sets.

Counting the same hyperedges with their complements implies

$$i_{\mathcal{H}}(V-(C\cup K)) = |\mathcal{E}| - |\widehat{E}_{\mathcal{H}}(C) \cup \widehat{E}_{\mathcal{H}}(K)| \ge \widetilde{m}(V) - \ell - |\widehat{E}_{\mathcal{H}}(C)| - |\widehat{E}_{\mathcal{H}}(K)|.$$

Thus equality must hold throughout. This is only possible if $\widehat{E}_{\mathcal{H}}(C) \cap \widehat{E}_{\mathcal{H}}(K) = \emptyset$. □

If all the MCT sets of $\mathcal{H}$ are pairwise disjoint and $|\mathcal{C}^*| \ge 3$, then this is the case as the following lemma shows.

Lemma 3.13 *Suppose that the members of $\mathcal{C}^*$ are pairwise disjoint and $|\mathcal{C}^*| \ge 3$. Let $C, K \in \mathcal{C}^*$. Then $N(C) \cap K = \emptyset$.*

Proof First, if $\widetilde{m}(V-(C\cup K)) \ge \ell$ (e.g. $\ell \le 0$), then the statement follows by Lemma 3.12 immediately.

Suppose now that $\widetilde{m}(V-(C\cup K)) < \ell$ (and $\ell > 0$). Now $|V-(C\cup K)| \le 1$ holds by (A0). $V \ne C\cup K$ since there exist at least three disjoint MCT sets. Therefore, for some $v \in V$, $V-(C\cup K) = v$ holds and v is a co-tight set on its own with $m(v) < \ell$. (A1) implies that $m(v) > 0$ also holds. Since $\{v\}$ is co-tight in $\mathcal{H}$, $d_{\mathcal{H}}(v) = m(v)$. By Lemma 2.13, there is no loop in $\mathcal{H}$ incident with v. Note that $C\cup\{v\} = V-K$ and $K\cup\{v\} = V-C$ are tight sets in $\mathcal{H}$ each of which must induce at least one edge incident with v by Lemma 2.13 and the fact that any vertex with zero degree in an (m,ℓ)-tight graph has also $m(v)=0$. Since $V = C\cup K\cup\{v\}$ and $|V| \ge 4$, at least one of C and K, say C, must contain at least 2 vertices. Hence $d_{\mathcal{H}[C\cup\{v\}]}(v) \ge m(v)$ also follows by Lemma 2.13. But now the disjointedness of C and K implies that $d_{\mathcal{H}}(v) > m(v)$, a contradiction. □

As we have noted before, we plan to connect the members of $\mathcal{C}^*$ by the new edges. Based on the above result, the following statement provides a useful property of the arising redundant subhypergraphs. (It can be illustrated again by Figure 3.1, where the highlighted sets are $(3,4)$-MCT sets in G and $\mathcal{T}(ux)$ is the subgraph formed by the bold edges.)

Lemma 3.14 *Let C be an (m,ℓ)-MCT set in $\mathcal{H}$ and let $u \in C$ and $v \in V-(C \cup N(C))$ such that $m(v) \neq 0$. Then $C \cup N(C) \subset V(uv)$.*

Proof As $v \notin N(u)$ and $m(v) \neq 0$, condition (A0) implies that $\{u,v\}$ is not tight in $\mathcal{H}$.

First, we show that $C \subset V(uv)$. Note that $V-C$ is maximal amongst all the eligible tight subsets of V, which implies that either $V(uv) \cup (V-C) = V$, and hence $C \subset V(uv)$ or $V(uv) \cup (V-C)$ is not tight in $\mathcal{H}$. In the latter case, $V(uv) \cap (V-C) = v$ and $m(v) < \ell$ both follow by Lemma 2.9. Thus, since v is not connected to C, the tightness of $\mathcal{T}(uv)$ and the sparsity of $\mathcal{H}$ imply that $\widetilde{m}(V(uv)) - \ell = i(V(uv)) = i(v) + i(V(uv)-v) \leq 0 + \widetilde{m}(V(uv)-v) - \ell = \widetilde{m}(V(uv)) - \ell - m(v)$ contradicting $m(v) > 0$. Here $i(v) = 0$ because $m(v) - \ell < 0$.

Now the (m,ℓ)-tightness of $V(uv)$, the (m,ℓ)-sparsity of $\mathcal{H}$, and the co-tightness of C imply that $\widetilde{m}(V(uv)) - \ell = i_{\mathcal{H}}(V(uv)) = i_{\mathcal{H}}(V(uv)-C) + e_{\mathcal{T}(uv)}(C) \leq i_{\mathcal{H}}(V(uv)-C) + e_{\mathcal{H}}(C) \leq \widetilde{m}(V(uv)-C) - \ell + \widetilde{m}(C)$. Hence equality must hold throughout, in particular, $e_{\mathcal{T}(uv)}(C) = e_{\mathcal{H}}(C)$ implying $N(C) \subset V(uv)$. $\square$

A set P is called a **transversal** of a family $\mathcal{S}$ if $|P \cap S| = 1$ for each $S \in \mathcal{S}$. Note that a transversal of $\mathcal{C}^*$ can be easily constructed by taking one arbitrary element of each member of $\mathcal{C}^*$ when the members of $\mathcal{C}^*$ are pairwise disjoint. Next, we show that any connected graph on a transversal of $\mathcal{C}^*$ augments $\mathcal{H}$ to an (m,ℓ)-redundant hypergraph, that is, we can make the augmentation with $|\mathcal{C}^*| - 1$ edges.

Lemma 3.15 *Let $\mathcal{H} = (V, \mathcal{E})$ be an (m,ℓ)-tight hypergraph. Let P be a set of vertices, so that $|P \cap C| \geq 1$ for each $C \in \mathcal{C}^*$. Suppose moreover, that for each $p \in P$, there exists a set D_p such that $D_p \subset V(\mathcal{T}(pq))$ with $\widetilde{m}(D_p) \geq \ell$ for all $q \in P-p$. Let F be an edge set of a connected graph on $P' \subseteq P$. Then $R_{\mathcal{H}}(F)$ is the minimal (m,ℓ)-tight subhypergraph inducing all elements of P'. In particular, if F is the edge set of a star $K_{1,|P|-1}$ on the vertex set P, then $\mathcal{H}+F$ is (m,ℓ)-redundant.*

Proof Recall that $R(F)$ denotes the set of (m,ℓ)-redundant edges of $\mathcal{H}$ in $\mathcal{H}+F$, and Lemma 2.12 claims that $R(F) = \bigcup_{f \in F} T(f)$. Let us use induction on $|F|$. If $F = \{ij\}$, then $R(F) = T(ij)$ which is the minimal (m,ℓ)-tight subgraph of $\mathcal{H}$ containing both of i and j by Lemma 2.11.

Let $ij \in F$ such that $F - ij$ is connected. By induction, $R(F-ij)$ is a tight subhypergraph of $\mathcal{H}$ which induces each element of $V(R(F-ij))$, in particular, we may assume (by possibly switching the role of i and j) that $i \in V(R(F-ij))$. If $j \in V(R(F-ij))$ also holds, then

$\mathcal{T}(ij) \subseteq R(F - ij)$ by Lemma 2.11. Hence we may assume that $j \notin V(R(F - ij))$. The connectivity of $F - ij$ implies that there exists an edge $ij' \in F - ij$. Note that $\mathcal{T}(ij') \subseteq R(F - ij)$ by Lemma 2.12. Hence $D_i \subset V(\mathcal{T}(ij')) \subseteq V(R(F - ij))$ while $D_i \subset V(\mathcal{T}(ij))$ by our assumption. Thus we may use Lemmas 2.9 and 2.12 to conclude that $R(F) = R(F - ij) \cup \mathcal{T}(ij)$ is tight.

Let now T be the minimal tight subhypergraph of $\mathcal{H}$ which induces all elements of P'. Lemma 2.11 implies that $\mathcal{T}(f) \subseteq T$ for each $f \in F$. Hence it follows by Lemma 2.12 that $R(F) = \bigcup_{f \in F} \mathcal{T}(f) \subseteq T$, that is, $R(F) = T$.

Finally, if $P' = P$, then $P \subset V(R(F))$ and thus $R(F) = \mathcal{H}$ by Lemma 3.8 since P intersects every MCT set. □

Lemma 3.16 *Let $\mathcal{H}$ be an (m, ℓ)-tight hypergraph. Suppose that the members of $\mathcal{C}^*$ are pairwise disjoint and $|\mathcal{C}^*| \geq 3$. If P is a transversal of $\mathcal{C}^*$ and F is the star $K_{1,|P|-1}$ on the vertex set P, then $\mathcal{H} + F$ is (m, ℓ)-redundant.*

Proof By Lemmas 3.13 and 3.14 it is easy to check that the conditions of Lemma 3.15 hold thus we can conclude that $\mathcal{H} + F$ is redundant. □

The cardinality of the edge set provided by Lemma 3.16 can be decreased by iteratively using the following statement.

Lemma 3.17 *Suppose that the members of $\mathcal{C}^*$ are pairwise disjoint and $|\mathcal{C}^*| \geq 4$. Let y, x_1, x_2, x_3 be elements of distinct members of $\mathcal{C}^*$. Let $T = \mathcal{T}(yx_1) \cup \mathcal{T}(yx_2) \cup \mathcal{T}(yx_3)$. Then $T = \mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3)$ or $T = \mathcal{T}(yx_3) \cup \mathcal{T}(x_1x_2)$ holds.*

Proof Let us suppose that $T \neq \mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3)$. Thus there exists a hyperedge e for which $e \in T$ and $e \notin \mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3)$. Note that e cannot be a loop, as that would contradict (for example) the tightness of $\mathcal{T}(yx_1)$.

Lemmas 2.12 and 3.16 imply that T is the minimal tight subhypergraph of $\mathcal{H}$ inducing all of y, x_1, x_2, and x_3. However, they similarly imply that this statement also holds for $\mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3) \cup \mathcal{T}(yx_3)$ and $\mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3) \cup \mathcal{T}(x_1x_2)$, that is, these two hypergraphs are both equal to T. Since $e \in T$ and $e \notin \mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3)$, we get $e \in \mathcal{T}(yx_3)$ and $e \in \mathcal{T}(x_1x_2)$. (See Figure 3.3.)

Lemma 2.9 with (A0) implies now that $\mathcal{T}(yx_3) \cup \mathcal{T}(x_1x_2)$ is a tight subhypergraph of $\mathcal{H}$ inducing all of y, x_1, x_2, and x_3, hence it must be equal to T. (We note that if $T \neq \mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3)$, then $T = \mathcal{T}(yx_2) \cup \mathcal{T}(x_1x_3)$ also follows by a similar proof.) □

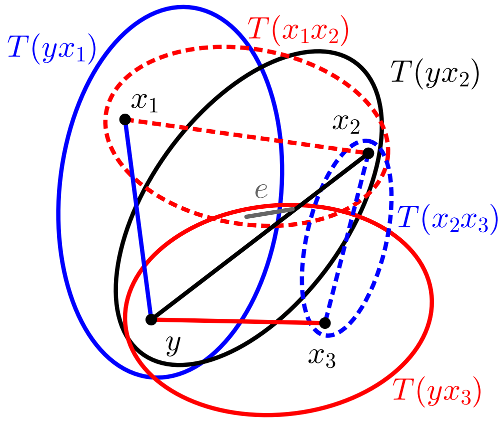


Figure 3.3: Illustrations of Lemma 3.17.

Observation 3.18 *Let $x_1, x_2, x_3, y \in V$ and let T be the minimal tight subgraph containing x_1, x_2, x_3, y. If $T = \mathcal{T}(yx_1) \cup \mathcal{T}(yx_2) \cup \mathcal{T}(yx_3)$, $T = \mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3) \cup \mathcal{T}(yx_3)$ and $T = \mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3) \cup \mathcal{T}(x_1x_2)$, then $T = \mathcal{T}(yx_1) \cup \mathcal{T}(x_2x_3)$ or $T = \mathcal{T}(yx_3) \cup \mathcal{T}(x_1x_3)$ holds.*

This finally allows us to finish the proof of Theorem 3.5.

Proof of Theorem 3.5 As mentioned before, we only need to prove that max $\geq$ min holds. This is obvious when min $= 1$. Hence we may assume that at least two edges are needed for the augmentation. In this case, the members of $\mathcal{C}^*$ are pairwise disjoint by Theorem 3.9 and we need to show that the augmentation can be done by using $\left\lceil \frac{|\mathcal{C}^*|}{2} \right\rceil$ edges. Let us denote a transversal of $\mathcal{C}^*$ by P. For an arbitrary $q \in P$, let the edge set $F_0 := \{pq : p \in P - q\}$, and let $H_0 = (V, F_0)$. By Lemma 3.16 $\mathcal{H} \cup H_0$ is (m, ℓ)-redundant. Let us decrease the number of edges in H_i (that is H_0 at the beginning). While there are at least three edges in H_i that are incident with q, we can decrease the number of edges in H_i and also the edges incident with q by Lemma 3.17, so that the arising graph $H_{i+1} = (V, F_i)$ still augments $\mathcal{H}$ to an (m, ℓ)-redundant hypergraph. We can repeat this until the degree of q is at most two and the degree of every other vertex is at most one in the final graph $H = (V, F)$. Thus $|F| = \left\lceil \frac{|P|}{2} \right\rceil = \left\lceil \frac{|\mathcal{C}^*|}{2} \right\rceil$ and $\mathcal{H} \cup H$ is (m, ℓ)-redundant. $\square$

The method above gives a polynomial time algorithm for solving the reduced augmentation problem and also the general problem if $\ell \leq k$. In Section 8.3 of Chapter 8 we present a rather complex algorithm that solves the reduced problem in $O(|V|^2)$ running time, in case the input is a (k, ℓ)-M-component hypergraph. We mention that there exists an efficient algorithm for every polynomially solvable case [51].

Before we prove Theorem 3.9, we show the connection between the results of this section and the paper of García and Tejel [22]. This also shows how we managed to extend their results. We also use this result in the proof of the NP-hardness result in Section 3.4. As in [22] the authors used graphs and not hypergraphs we shall use graphs, as well, to remain consistent. Let $G = (V, E)$ be an (m, ℓ)-tight graph. Let us call an (m, ℓ)-tight subgraph G' of G **generated** if there are $u, v \in V$ such that $T(uv) = G'$.

Lemma 3.19 *Assume that there is no edge uv that augments G to an (m, ℓ)-redundant graph. Then $\mathcal{T}_G(uv)$ is inclusion-wise maximal amongst all the generated subgraphs of G if and only if $u, v \in V$ are elements of two distinct (m, ℓ)-MCT sets. Moreover, two inclusion-wise maximal generated subgraphs $\mathcal{T}_G(uv_1)$ and $\mathcal{T}_G(uv_2)$ are equal if and only if v_1, v_2 are in the same (m, ℓ)-MCT set.*

Note that this result implies that 'classes of extreme vertices' defined in [22] are exactly the $(2, 3)$-MCT sets when no edge uv augments G to a $(2, 3)$-redundant graph.

Proof By Theorem 3.11, all the MCT sets of G are pairwise disjoint. Assume that $\mathcal{T}(uv)$ is an inclusion-wise maximal generated subgraph of G for some $u, v \in V$. Let $\{x_1, \dots, x_t\}$ be a transversal of $\mathcal{C}^*$. By Lemma 3.16, $G = \mathcal{T}(x_1x_2) \cup \dots \cup \mathcal{T}(x_1x_t)$, hence we can assume that $u \in V(x_1x_2)$ and $v \in V(x_1x_2) \cup V(x_1x_3)$. On the other hand, $\mathcal{T}(x_1x_2) \cup \mathcal{T}(x_1x_3) = \mathcal{T}(x_1x_2) \cup \mathcal{T}(x_2x_3) = \mathcal{T}(x_1x_3) \cup \mathcal{T}(x_2x_3)$ by Lemmas 2.12 and 3.16. Hence one of the above three generated tight subgraphs, say $\mathcal{T}(x_1x_2)$, contains both u and v thus $V(uv) \subseteq V(x_1x_2)$ by Lemma 2.11. Since $\mathcal{T}(uv)$ is inclusion-wise maximal equality must hold.

Let $C_1, C_2 \in \mathcal{C}^*$ such that $x_1 \in C_1$ and $x_2 \in C_2$. Suppose that $\{u, v\} \cap C_1 = \emptyset$. Note that $V - C_1$ is a tight set in G, and hence $V(uv)$ is disjoint from C_1, in particular $x_1 \notin V(uv)$, contradicting $V(uv) = V(x_1x_2)$. Therefore, with the same argument for C_2, either $u \in C_1$ and $v \in C_2$, or $u \in C_2$ and $v \in C_1$.

It is clear that taking any element x of C_1 and any element y of C_2 the generated tight set $\mathcal{T}(xy)$ is the same by Lemmas 3.14, 3.13 and 2.11. On the other hand, if there are $a, b \in V$ such that $\mathcal{T}(xy) \subseteq \mathcal{T}(ab)$, then $a \in C_1$ and $b \in C_2$ or vice-versa by Observation 3.7. This proves that $\mathcal{T}(xy)$ is an inclusion-wise maximal generated subgraph of G. $\square$

3.3.1 Proofs of Theorems 3.9, 3.10 and 3.11

Recall that we assume (A) for our (m,ℓ)-tight hypergraph $\mathcal{H}$ and $\mathcal{C}^*$ denotes the family of all (m,ℓ)-MCT sets of $\mathcal{H}$.

Lemma 3.20 *If X and Y are two (m,ℓ)-MCT sets in $\mathcal{H}$, such that $X\cap Y\neq\emptyset$, then $\widetilde{m}(V-(X\cup Y))<\ell$. In particular, $|X\cup Y|\geq|V|-1$.*

Proof For the sake of contradiction, let us suppose that $\widetilde{m}(V-(X\cup Y))\geq\ell$. $V-X$ and $V-Y$ are tight sets in $\mathcal{H}$ by the co-tightness of X and Y. $(V-X)\cap(V-Y)=V-(X\cup Y)$, hence $V-(X\cap Y)$ is also tight by Lemma 2.9. Thus $X\cap Y$ is co-tight, contradicting the minimality of X and Y. Finally, $|X\cup Y|\geq|V|-1$ follows by (A0) and $\widetilde{m}(V-(X\cup Y))<\ell$. □

Note that Lemma 3.20 implies that all the MCT sets of $\mathcal{H}$ are pairwise disjoint, when $\ell\leq 0$. Hence we only need to prove Theorems 3.9, 3.10 and 3.11 for the case where $\ell>0$. In this case, we have two possibilities for intersecting MCT sets: their union may be equal to V or be of cardinality $|V|-1$.

The first case is quite easy to solve, with the idea coming from [40, Theorem 3.9.13]

Lemma 3.21 *Suppose that for any two (m,ℓ)-MCT sets X and Y for which $X\cap Y\neq\emptyset$, $X\cup Y=V$. Then there exist two vertices $u_1,u_2\in V$, so that $\mathcal{T}_{\mathcal{H}}(u_1u_2)=\mathcal{H}$.*

Proof Let us consider a vertex $v\in Y-X$. Suppose that there exists an MCT set Z that is different from X and Y. The minimality of Y implies that Z intersects X and hence by Lemma 3.20 (with the assumption that the union of any two intersecting MCT sets equals V) $Y-X\subseteq Z$ and $v\in Z$. Thus v belongs to all MCT sets except for X. Now with any $w\in X$, $\mathcal{T}(vw)=\mathcal{H}$ by Lemma 3.8. □

Let us now focus on the more difficult case, that is, if there exist two MCT sets, that intersect and the cardinality of their union is $|V|-1$. This case proves to be significantly more difficult.

For a vertex $v\in V$, let $\boldsymbol{\mathcal{C}^*(v)}:=\{C\in\mathcal{C}^*:v\notin C\}$ denote the set of MCT sets that don't contain v. Given a ground set S, a family of its subsets is called a **co-partition** of S, if their complements form a partition of S.

Lemma 3.22 *Suppose that $\ell>0$. Assume that there exist two (m,ℓ)-MCT sets $X,Y\in\mathcal{C}^*$ such that $X\cap Y\neq\emptyset$ and $X\cup Y=V-v$ for some $v\in V$. Then $\mathcal{C}^*(v)$ is a co-partition of $V-v$ with $|\mathcal{C}^*(v)|\geq 3$ or there exists a vertex $u\in V-v$ such that $\mathcal{T}_{\mathcal{H}}(uv)=\mathcal{H}$.*

Proof Assume that there exists no vertex $u \in V - v$ such that $\mathcal{T}(uv) = \mathcal{H}$. Then $\mathcal{C}^*(v) \neq \{X, Y\}$, since otherwise the set formed by a vertex $u \in X \cap Y$ and v would cover $\mathcal{C}^*$ contradicting Lemma 3.8 and $\mathcal{T}(uv) \neq \mathcal{H}$. Hence $|\mathcal{C}^*(v)| \geq 3$ as $X, Y \in \mathcal{C}^*(v)$. Let $Z \in \mathcal{C}^*(v) - \{X, Y\}$. As Z must intersect X or Y, $X \cup Z$ or $Y \cup Z$ is equal to $V - v = X \cup Y$ according to Lemma 3.20. However, then both $X \cup Z$ and $Y \cup Z$ are equal to $V - v$ by the same reasoning. Thus $Z \supseteq (X - Y) \cup (Y - X)$. This implies also that every two members of $\mathcal{C}^*(v)$ are intersecting, and hence for every three members W_1, W_2, and W_3 of $\mathcal{C}^*(v)$, $W_3 \supseteq (W_1 - W_2) \cup (W_2 - W_1)$ holds. Therefore, every vertex in $V - v$ is avoided by at most one member of $\mathcal{C}^*(v)$. If there exists a vertex u that is contained in every member of $\mathcal{C}^*(v)$, then $\{u, v\}$ covers $\mathcal{C}^*$ contradicting Lemma 3.8 and $\mathcal{T}(uv) \neq \mathcal{H}$. Therefore, every vertex in $V - v$ is avoided by exactly one member of $\mathcal{C}^*(v)$, that is, $\mathcal{C}^*(v)$ is a co-partition of $V - v$. □

This is the point where our proofs get more involved than that of [40, Theorem 3.9.13].

For a vertex $v \in V$ and a set $W \subseteq V - v$, let $\boldsymbol{\widetilde{W}^v} := V - v - W$.

We can assume that there are at least two (m, ℓ)-MCT sets in $\mathcal{H}$ that are intersecting and by Lemma 3.21 we might also assume that there exists a vertex v that is not in either of these MCT sets. Now by Lemmas 3.20 and 3.22 we can assume that $|\mathcal{C}^*(v)|$ is a co-partition of $V - v$ and $|\mathcal{C}^*(v)| \geq 3$.

Lemma 3.23 *Suppose that $\ell > 0$. Let $v \in V$ be a vertex for which the family $\mathcal{C}^*(v)$ is a co-partition of $V - v$ with $|\mathcal{C}^*(v)| \geq 3$. Let $T' \subsetneq V$ induce an (m, ℓ)-tight set in $\mathcal{H}$. Suppose that there exists an MCT set $Z \in \mathcal{C}^*(v)$ for which $T' \not\subseteq (V - Z) = (\widetilde{Z}^v \cup v)$ and $\widetilde{m}(T' \cap (V - Z)) \geq \ell$ for the maximal tight set $(V - Z)$. Then there exist two vertices $u_1, u_2 \in V$ for which $\mathcal{T}_{\mathcal{H}}(u_1 u_2) = \mathcal{H}$. In particular, if $|T' \cap (V - Z)| \geq 2$, then there exist two vertices $u_1, u_2 \in V$ for which $\mathcal{T}_{\mathcal{H}}(u_1 u_2) = \mathcal{H}$.*

Proof By Lemma 2.9 $T' \cup (V - Z) = T' \cup (\widetilde{Z}^v \cup v)$ is a tight set in $\mathcal{H}$. Since $\widetilde{Z}^v \cup v$ is a maximal tight set in $\mathcal{H}$, $T' \cup (V - Z) = V$ must hold. Lemma 2.9 also states that $d(T', V - Z) = 0$.

Suppose first that $v \in T'$. Since $T' \cup (V - Z) = V$, $Z \subset T'$, hence, for an MCT set $W \neq Z$, we have $(\widetilde{W}^v \cup v) \subseteq T'$ contradicting the maximality of $\widetilde{W}^v \cup v$.

Suppose now that $v \notin T'$. Then $d(T', V - Z) = 0$ implies $d(v, Z) = 0$. Hence $d(v, \widetilde{W}^v) = 0$ for each $W \in \mathcal{C}^*(v) - \{Z\}$. Note that $m(v) < \ell$ by Lemma 3.20 (since any two members of $\mathcal{C}^*(v)$ are intersecting and their union avoids v). If $m(v) > 0$, then Lemma 2.13 on the tight hypergraph $\mathcal{H}[V - W]$ implies that $d(v, \widetilde{W}^v) > 0$ for each $W \in \mathcal{C}^*(v) - \{Z\}$, contradicting $d(v, Z) = 0$ as

$V - W - v = \widetilde{W}^v \subset Z$. However, if $m(v) = 0$, then v is not in any MCT sets by Lemma 3.6. Thus $\{w_1, w_2\}$ covers all MCT sets for $w_1 \in \widetilde{W}_1^v$ and $w_2 \in \widetilde{W}_2^v$ where $W_1, W_2 \in \mathcal{C}^*(v)$, meaning that $V(w_1w_2) = V$ by Lemma 3.8. □

The immediate consequence of Lemma 3.23 is that we can assume that $0 < m(u) < \ell$ for every $u \in V$:

Lemma 3.24 *Suppose that $\ell > 0$. Let $v \in V$ be a vertex for which the family $\mathcal{C}^*(v)$ is a co-partition of $V - v$ with $|\mathcal{C}^*(v)| \geq 3$. If there is a vertex $u \in V$ for which $m(u) \geq \ell$ or $m(u) = 0$, then there exist two vertices $u_1, u_2 \in V$ for which $\mathcal{T}_{\mathcal{H}}(u_1u_2) = \mathcal{H}$.*

Proof By Lemma 3.20 we can assume that $m(v) < \ell$, since any two members of $\mathcal{C}^*(v)$ are intersecting and their union avoids v. Hence by (A0) and $\ell > 0$ we have $m(u) > 0$. Let now Z be an MCT set, so that $u \in \widetilde{Z}^v$. Let us take a vertex $z \in Z$ (that is not in $\widetilde{Z}^v$). We either have $\mathcal{T}(uz) = \mathcal{H}$ or we can use Lemma 3.23 with $\mathcal{T}(uz)$ concluding that there exist $u_1, u_2 \in V$ for which $\mathcal{T}(u_1u_2) = \mathcal{H}$. □

If $m(u) < \ell$ for all $u \in V$, there cannot be any loops (or hyperedges of size 1) in $\mathcal{H}$ by Lemma 2.13.

Lemma 3.25 *Suppose that $\ell > 0$. Let $v \in V$ be a vertex for which the family $\mathcal{C}^*(v)$ is a co-partition of $V - v$ with $|\mathcal{C}^*(v)| \geq 3$. Let us assume that there exist no two vertices $u_1, u_2 \in V$ for which $\mathcal{T}_{\mathcal{H}}(u_1u_2) = \mathcal{H}$. Let $\mathcal{S}$ denote the family of inclusion-wise maximal (m, ℓ)-tight sets (which are not the complete $\mathcal{H}$). For any two vertices $u_1, u_2 \in V$, there exists a set $T_{u_1u_2} \in \mathcal{S}$ for which $u_1, u_2 \in T_{u_1u_2}$, while for any two $T_1, T_2 \in \mathcal{S}$, $|T_1 \cap T_2| \leq 1$.*

Proof As we know that $\mathcal{T}(u_1u_2)$ is a tight set for any two $u_1, u_2 \in V$ and we assumed that $\mathcal{T}(u_1u_2) \neq V$, this ensures the existence of a set suitable to be $T_{u_1u_2}$.

Let us suppose now for a contradiction that $T_1, T_2 \in \mathcal{S}$ and $|T_1 \cap T_2| > 1$. By Lemma 2.9 $T_1 \cup T_2$ is a tight set. If $T_1 \cup T_2 \neq V$, then $T_1 \cup T_2$ is a tight set containing T_1 hence contradicting the maximality of T_1. On the other hand, if $T_1 \cup T_2 = V$, one of them must be a maximal tight set containing v. Hence we can use Lemma 3.23 to contradict the condition of not having $u_1, u_2 \in V$ for which $\mathcal{T}(u_1u_2) = \mathcal{H}$. This also shows the uniqueness of $T_{u_1u_2}$. □

At this point the proofs of Theorem 3.9, 3.10 and 3.11 diverge. We can prove, nonetheless, three very similar lemmas for them, namely Lemmas 3.26, 3.29 and 3.32. With the help of these lemmas we can finish the proofs of the theorems parallel, using the same steps. However, the proofs of these lemmas require their own distinct and often technical solutions.

For example, we can state the following lemma for (k,ℓ)-tight hypergraphs if $0 < \ell \leq \frac{3}{2}k$.

Lemma 3.26 *Let $\mathcal{H}$ be a (k,ℓ)-tight hypergraph with $|V| \geq 4$, and $0 < \ell \leq \frac{3}{2}k$. Let $v \in V$ be a vertex for which the family $\mathcal{C}^*(v)$ is a co-partition of $V - v$ with $|\mathcal{C}^*(v)| \geq 3$. Then there exist two vertices $u_1, u_2 \in V$ for which $\mathcal{T}_{\mathcal{H}}(u_1u_2) = \mathcal{H}$.*

Proof If $k \geq \ell$, Lemma 3.24 shows the existence of such $u_1, u_2 \in V$ for which $\mathcal{T}(u_1u_2) = \mathcal{H}$. Hence we can assume that $k < \ell$. Then $\mathcal{H}$ contains no loop by Lemma 2.13.

Claim 3.27 *Let $0 < \ell \leq \frac{3}{2}k$ and let $\mathcal{H}$ contain no loop. Let $T_1 = (V_1, \mathcal{E}_1)$, $T_2 = (V_2, \mathcal{E}_2)$ and $T_3 = (V_3, \mathcal{E}_3)$ be three (k,ℓ)-tight subhypergraphs of $\mathcal{H}$. If $|V_i \cap V_j| = 1$ for $(i,j) \in \{(1,2),(2,3),(3,1)\}$, then $T_1 \cup T_2 \cup T_3$ is (k,ℓ)-tight, and $d_{\mathcal{H}}(V_i, V_j) = 0$ for any $(i,j) \in \{(1,2),(2,3),(3,1)\}$.*

Proof $|\mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3| = |\mathcal{E}_1| + |\mathcal{E}_2| + |\mathcal{E}_3| = k|V_1| + k|V_2| + k|V_3| - 3\ell = k|V_1 \cup V_2 \cup V_3| + 3k - 3\ell$. (Here the first equation holds because $\mathcal{H}$ contains no loop.) By this and the sparsity of $\mathcal{H}$ we have $k|V_1 \cup V_2 \cup V_3| + 3k - 3\ell \leq k|V_1 \cup V_2 \cup V_3| - \ell$. That is, $3k \leq 2\ell$, which can be true if and only if $\frac{3}{2}k = \ell$ by the condition $\ell \leq \frac{3}{2}k$. However, then equations must hold, which proves both statements of the claim. $\square$

By Claim 3.27 on $V - X$, $V - Y$ and $\mathcal{T}(xy)$ for arbitrary $x \in \widetilde{X}^v$ and $y \in \widetilde{Y}^v$, we can state that $(V - X) \cup (V - Y) \cup V(\mathcal{T}(xy))$ spans a tight subhypergraph. By Lemma 3.8 and the fact that x, y and v covers all MCT sets by Lemma 3.22 we can conclude that $V = (V - X) \cup (V - Y) \cup V(\mathcal{T}(xy))$. Let $Z \in \mathcal{C}^*(v) - \{X, Y\}$. Obviously $\widetilde{X}^v \subset V(\mathcal{T}(xy))$ and hence $|\widetilde{X}^v| = 1$ by Lemma 3.25. However, by Claim 3.27 we conclude that $d_{\mathcal{H}}(\{v\}, \widetilde{Z}^v) = 0$. The same reasoning holds for example to X and Z, hence we can conclude that $d_{\mathcal{H}}(\{v\}, \widetilde{Y}^v) = 0$ and later similarly $d_{\mathcal{H}}(\{v\}, \widetilde{X}^v) = 0$. In other words (since $\mathcal{H}$ contains no loop), $d_{\mathcal{H}}(v) = 0$. This, however, contradicts the (k,ℓ)-sparsity of $V - v$, since we could assume $m(v) > 0$ by Lemma 3.24.

$\square$

A similar lemma can be stated that will help us prove Theorem 3.9. We prove it via a technical lemma that we phrase in a more general manner. Once the lemma is stated we will point out the connections with the (k,ℓ)-rigidity setup in order to make it easier to follow.

Lemma 3.28 *Let k, ℓ be two positive integers so that $k < \ell < 2k$. Let V be a set of n elements, where $n \geq 2k$.*

Let $\mathcal{B}$ be a family of sets on V with $|\mathcal{B}| \geq 2$, where $|S| \geq 2$ for every $S \in \mathcal{B}$. Assume that for every two elements $a, b \in V$ there exists a set $S \in \mathcal{B}$ for which $a, b \in S$. Suppose moreover that for any two sets $S, R \in \mathcal{B}$, $|S \cap R| \leq 1$.

Let us have one marked element $e \in V$. Let us define a cost function $f : \mathcal{B} \to \mathbb{Z}_+$ as follows: for any $S \in \mathcal{B}$ let $f(S) = (k|S| - \ell)$ if $e \notin S$ and $f(S) = ((k-1)|S| + h - \ell)$, if $e \in S$ for a constant $0 < h \leq k$ integer for which $k + h > \ell$.

Then $\sum\limits_{S \in \mathcal{B}} f(S) > k(n-1) + h - \ell$.

Before the proof, let us shed some light on the parallels of the phrasing of Lemma 3.28 and the (k, ℓ)-rigidity setup. The set of the n elements (V) stands for the vertices of $\mathcal{H}$. $\mathcal{B}$ represents the maximal tight sets that follow such an arrangement by Lemma 3.25. The function f gives the number of induced hyperedges in the given (m^*, ℓ)-tight subhypergraphs of $\mathcal{H}$, where m^* is almost constant k on every vertex, except for the marked $e \in V$, for which $m^*(e) = h$. The lemma states that if $|\mathcal{B}| \geq 2$, $\mathcal{H}$ cannot be (m^*, ℓ)-tight. We use this more general phrasing to simplify the notations of the proof.

Proof By $|\mathcal{B}| \geq 2$, $|S| \geq 2$ and the condition that for any two sets $S, R \in \mathcal{B}$, $|S \cap R| \leq 1$, we have $|S| < n$ for any $S \in \mathcal{B}$. Let us denote the number of sets in $\mathcal{B}$ that have size 2,3,…,$n-1$ by d_2,d_3, …,d_{n-1}, respectively. There are $\frac{n(n-1)}{2}$ pairs of elements in V. By the conditions every two elements are in exactly one set in $\mathcal{B}$ thus we have $\frac{n(n-1)}{2} = \sum\limits_{i=2}^{n-1} d_i \frac{(i-1)i}{2}$. Hence $n(n-1) = \sum\limits_{i=2}^{n-1} d_i (i-1)i$.

Let us now calculate a lower bound for $\sum\limits_{S \in \mathcal{B}} f(S)$. A set $S \in \mathcal{B}$ for which $|S| \geq 2$, contributes to this sum with at least 1 by $\ell < k + h$ and $h \leq k$. Hence a set $S \in \mathcal{B}$ with $|S| = i$ contributes to this sum with at least $(i-2)k + 1$. Thus $\sum\limits_{i=2}^{n-1} d_i((i-2)k + 1) \leq \sum\limits_{S \in \mathcal{S}} f(S)$.

For the sake of contradiction let us suppose that $\sum\limits_{S \in \mathcal{S}} f(S) \leq k(n-1) + h - \ell < k(n-1)$, where the second inequality comes from $h \leq k < \ell$. Therefore $\sum\limits_{i=2}^{n-1} d_i((i-2)k + 1) < k(n-1)$.

Let us multiply both sides by $n(n-1) = \sum_{i=2}^{n-1} d_i(i-1)i$. The following calculation gives us

$$\begin{aligned}
n(n-1)\sum_{i=2}^{n-1} d_i((i-2)k+1) &< k(n-1)\sum_{i=2}^{n-1} d_i(i-1)i \\
n\sum_{i=2}^{n-1} d_i(i-2)k + n\sum_{i=2}^{n-1} d_i &< k\sum_{i=2}^{n-1} d_i(i-1)i \\
n\sum_{i=2}^{n-1} d_i(i-1)k - n\sum_{i=2}^{n-1} d_i(k-1) &< k\sum_{i=2}^{n-1} d_i(i-1)i \\
n\sum_{i=2}^{n-1} d_i(i-1)k - k\sum_{i=2}^{n-1} d_i(i-1)i &< n\sum_{i=2}^{n-1} d_i(k-1) \\
k\sum_{i=2}^{n-1} d_i(i-1)(n-i) &< n(k-1)\sum_{i=2}^{n-1} d_i
\end{aligned}$$

As $2 \leq i \leq n-1$, we can see that $n-i$ and $i-1$ are both positive integers. In such a case we can use the fact that $n-2 \leq (n-i)(i-1)$. Therefore, from the previous statement we get that

$$k\sum_{i=2}^{n-1} d_i(n-2) < n(k-1)\sum_{i=2}^{n-1} d_i$$

which is a contradiction, since

$$k(n-2) \geq n(k-1)$$

by the condition $n \geq 2k$. Hence, in fact, $\sum_{S\in\mathcal{S}} f(S) \leq k(n-1)+h-\ell$ thus finishing the proof. $\square$

Now we can use this lemma to prove a statement similar to Lemma 3.26.

Lemma 3.29 *Suppose that $\ell > 0$. Let $|V| \geq 2c$, where c is the universal constant from the condition (*). Let $v \in V$ be a vertex for which the family $\mathcal{C}^*(v)$ is a co-partition of $V-v$ with $|\mathcal{C}^*(v)| \geq 3$. Then there exist two vertices $u_1, u_2 \in V$ for which $\mathcal{T}_{\mathcal{H}}(u_1u_2) = \mathcal{H}$.*

Proof Suppose for contradiction that there exist no such two vertices $u_1, u_2 \in V$ for which $\mathcal{T}(u_1u_2) = \mathcal{H}$. Since by Lemma 3.24 we might assume that $m < \ell$, we may suppose that we have a family of (m,ℓ)-tight sets $\mathcal{S}$, which contains no tight sets of size 1 by Lemma 3.25. In other words, $\mathcal{H}$ contains no loop by Lemma 2.13. Let $q \in V$ be a vertex for which $m(q)$ is the minimum value of m. Let $w \in V-q$ be a vertex that has minimum m value in $V-q$ and let us denote $m(w)$ by g. By (A0) and $0 < \ell$, we have $\ell < m(q)+m(w) = m(q)+g \leq 2g$. Obviously $g \leq c$, therefore $|V| \geq 2g$. By $m < \ell$, we have $g < \ell$.

Let us define a new m' function on V as $m'(q) = m(q)$ and $m'(j) = g$, if $j \in V - q$. Therefore, $m'(i) \le m(i)$ for all $i \in V$. Thus, by Lemma 3.25, $\mathcal{S}$ satisfies the conditions of Lemma 3.28 with $g = k$, q being the marked element and $\widetilde{m}'(S) - \ell = f(S)$. Hence Lemma 3.28 gives $\sum_{S\in\mathcal{S}} (\widetilde{m}'(S) - \ell) > g(|V| - 1) + m(q) - \ell = \widetilde{m}'(V) - \ell$. Since $\mathcal{S}$ covers V and $m(i) \ge m'(i)$ for all $i \in V$, $\sum_{S\in\mathcal{S}} (\widetilde{m}(S) - \widetilde{m}'(S)) \ge \widetilde{m}(V) - \widetilde{m}'(V)$. Now summing these two inequalities we get $\sum_{S\in\mathcal{S}} (\widetilde{m}(S) - \ell) > \widetilde{m}(V) - \ell$. However, as $\mathcal{H}$ contains no loop, $\sum_{S\in\mathcal{S}} (\widetilde{m}(S) - \ell)$ is a lower bound of the number of hyperedges in $\mathcal{H}$, which is a contradiction, as $\mathcal{H}$ is (m, ℓ)-tight and thus it has exactly $\widetilde{m}(V) - \ell$ hyperedges. Therefore, we can conclude that there exist such two vertices $u_1, u_2 \in V$ for which $\mathcal{T}(u_1u_2) = \mathcal{H}$. $\square$

Now let us prove a very similar lemma to Lemmas 3.26 and 3.29, only this time restricted to graphs instead of hypergraphs. This will be Lemma 3.32, however, this is the most complicated case, hence we need two additional lemmas before.

Lemma 3.30 *Suppose that $\ell > 0$. Let $\mathcal{H}$ be a graph, so that $|V| \ge 4$. Let $v \in V$ be a vertex for which the family $\mathcal{C}^*(v)$ is a co-partition of $V - v$, so that $|\mathcal{C}^*(v)| \ge 2$. Suppose that $m(v) > \frac{\ell}{2}$. Let $W_1, W_2 \in \mathcal{C}^*(v)$ and let $w_1 \in \widetilde{W}_1^v$ and $w_2 \in \widetilde{W}_2^v$. Suppose that V' is an (m, ℓ)-tight set in $\mathcal{H}$ with $w_1, w_2 \in V'$. Then either $V' = V$ or $V' = \{w_1, w_2\}$. In particular, either $V(w_1w_2) = V$ (and $\mathcal{T}(w_1w_2) = \mathcal{H}$) or $V(w_1w_2) = \{w_1, w_2\}$.*

Proof By Lemma 3.23 we may suppose that $|V' \cap (\widetilde{Z}^v \cup v)| \le 1$ for every $Z \in \mathcal{C}^*(v)$. Thus $v \notin V'$. Hence $|V' \cap \widetilde{Z}^v| \le 1$. Now if $|\mathcal{C}^*(v)| = 2$, this finishes the proof by Lemma 3.22.

Therefore, we may assume that $|\mathcal{C}^*(v)| \ge 3$. Let us consider the complement sets of the members of $\mathcal{C}^*(v)$. V' intersects at least two of them: $\widetilde{W}_1^v \cup v$ and $\widetilde{W}_2^v \cup v$. Let us prove that V' intersects exactly these two of them.

Let us denote the family of maximal tight sets containing v by $\mathcal{S}_v = \{\widetilde{Z}^v \cup v \text{ where } Z \in \mathcal{C}^*(v)\}$. (In other words, the elements of $\mathcal{S}_v$ are exactly the complements of the elements of $\mathcal{C}^*(v)$.) Suppose that V' intersects t members of $\mathcal{S}_v$, say $V_1, \dots, V_t$. Since $|V' \cap \widetilde{Z}^v| \le 1$ for every $Z \in \mathcal{C}^*(v)$ and $v \notin V'$, $|V'| = t$. Suppose that $t \ge 3$. Let $\mathcal{E}'$ denote the set of hyperedges induced by V'.

As V' is (m, ℓ)-tight, $\widetilde{m}(V') - \ell = |\mathcal{E}'|$. Since every pair of vertices in V' induces an (m, ℓ)-sparse subgraph in $\mathcal{H}$, $|\mathcal{E}'| \le (t-1)\widetilde{m}(V') - \binom{t}{2}\ell$. This results $\widetilde{m}(V') - \ell \le (t-1)\widetilde{m}(V') - \frac{t(t-1)}{2}\ell$. Hence $\frac{t(t-1)-2}{2}\frac{1}{t-2}\ell \le \widetilde{m}(V')$, and thus $\frac{t+1}{2}\ell \le \widetilde{m}(V')$. Note that here we used the fact that $\mathcal{H}$ is a graph, that is, there exist only edges in $\mathcal{H}$.

Let $\mathcal{E}^*$ denote the union of $\mathcal{E}'$ and the set of edges induced by $V_1, \ldots, V_t$. Clearly, $\widetilde{m}(V') - \ell + \widetilde{m}(V_1) - \ell + \cdots + \widetilde{m}(V_t) - \ell = |\mathcal{E}^*|$. By the sparsity condition, $|\mathcal{E}^*| \leq \widetilde{m}(V_1 \cup \cdots \cup V_t) - \ell$. As $\widetilde{m}(V_1 \cup \cdots \cup V_t) = \widetilde{m}(V_1) + \cdots + \widetilde{m}(V_t) - (t-1)m(v)$, we get $t\ell \geq \widetilde{m}(V') + (t-1)m(v) \geq \frac{t+1}{2}\ell + (t-1)m(v)$. Thus $m(v) \leq \ell\frac{2t-t-1}{2(t-1)} = \frac{\ell}{2}$, contradicting our condition of $m(v) > \frac{\ell}{2}$. Therefore, $t = 2$. $\square$

Based on Lemma 3.30, using assumption (A2) we prove the following.

Lemma 3.31 *Suppose that $\ell > 0$. Let $\mathcal{H}$ be a graph, so that $|V| \geq 4$. Let $v \in V$ be a vertex for which the family $\mathcal{C}^*(v)$ is a co-partition of $V - v$, so that $|\mathcal{C}^*(v)| \geq 3$. Then either $m(v) \leq \frac{\ell}{2}$ holds or there exist two vertices $u_1, u_2 \in V - v$ such that $\mathcal{T}_{\mathcal{H}}(u_1u_2) = \mathcal{H}$.*

Proof Let us suppose that $m(v) > \frac{\ell}{2}$.

Suppose first that there is an MCT set $Z \in \mathcal{C}^*(v)$ for which $|\widetilde{Z}^v| \geq 2$. Let us have $z_1, z_2 \in \widetilde{Z}^v$ and let us take a vertex $x \in Z$ such that $m(x)$ is not the unique minimum of m. (Note that such x must exist as $|\mathcal{C}^*(v)| \geq 3$ and $\mathcal{C}^*(v)$ is a co-partition of $V - v$ hence $|Z| \geq 2$.) Then $m(x) > \frac{\ell}{2}$ by (A0) and $\ell > 0$. By Lemma 3.30, a tight set containing both of x and z_i is $\{x, z_i\}$ or V for $i = 1, 2$. Hence, for $i = 1, 2$, $\{x, z_i\}$ is a maximal tight set in $\mathcal{H}$ (that is $\neq V$) or $\mathcal{T}(xy) = \mathcal{H}$ holds for $y = z_i$. In other words, we can assume that the complements of $\{z_1, x\}$ and $\{z_2, x\}$ are MCT sets. Therefore, there exist at least two MCT sets avoiding x which are intersecting. Hence $\mathcal{C}^*(x)$ is a co-partition of $V - x$ or there exists a vertex $y \in V - x$ such that $\mathcal{T}(xy) = \mathcal{H}$ by Lemma 3.22. Now we may assume that $\mathcal{C}^*(x)$ is a co-partition of $V - x$. Then z_1 and z_2 are avoided by different members of $\mathcal{C}^*(x)$ since $V - \{z_1, x\}, V - \{z_2, x\} \in \mathcal{C}^*(x)$. Thus Lemma 3.30 for x assures that any tight set in $\mathcal{H}$ containing z_1 and z_2 is either $\{z_1, z_2\}$ or V, which contradicts the tightness of $\widetilde{Z}^v \cup v$.

Now suppose that $|\widetilde{Z}^v| = 1$ for each co-tight set $Z \in \mathcal{C}^*(v)$. This implies that $V(uv) = \{u, v\}$ for every $u \in V - v$. By (A2) there must exist a pair $x, y \in V - v$ for which $V(xy) \neq \{x, y\}$. Then $V(xy) = V$ and $\mathcal{T}(xy) = \mathcal{H}$ follows by Lemma 3.30. $\square$

Now we are ready to prove an analogous lemma for Lemmas 3.26 and 3.29.

Lemma 3.32 *Suppose that $\ell > 0$. Let $\mathcal{H}$ be a graph, so that $|V| \geq 4$. Let $v \in V$ be a vertex for which the family $\mathcal{C}^*(v)$ is a co-partition of $V - v$ with $|\mathcal{C}^*(v)| \geq 3$. Then there exist two vertices $u_1, u_2 \in V$ for which $\mathcal{T}(u_1u_2) = \mathcal{H}$.*

Proof Since $m(u_1) + m(u_2) > \ell$ for all $u_1, u_2 \in V$ by (A0) and $\ell > 0$, there can be at most one vertex q for which $m(q) \leq \frac{\ell}{2}$. By Lemma 3.31, we may suppose that $q = v$ is this the vertex with a unique minimal m value.

In this case, the following claim follows by Lemma 3.22 and 3.31. (Note that there exists no vertex $w \in V - v$ for which $\mathcal{T}(vw) = \mathcal{H}$ since $\mathcal{C}^*(v)$ is a co-partition.)

Claim 3.33 *Assume that there exist two intersecting MCT sets X and Y for which $X \cup Y = V - u$ for some $u \in V - v$. Then there exists a pair of vertices $u_1, u_2 \in V - v$ for which $\mathcal{T}(u_1u_2) = \mathcal{H}$.* □

We may suppose that there exists an MCT set Z containing v, since otherwise the family $\mathcal{C}^*$ is equal to $\mathcal{C}^*(v)$, which is a co-partition; hence it could be covered by a set $\{u_1, u_2\} \subseteq V - v$ implying the existance of $u_1, u_2 \in V - v$ for which $\mathcal{T}(u_1u_2) = \mathcal{H}$ by Lemma 3.8. Then, for every $X \in \mathcal{C}^*(v)$ intersecting Z, we may assume that $X \cup Z = V$ holds by Claim 3.33, and therefore $\widetilde{X} \subset Z$. Since $\widetilde{X} \subseteq W$ for all $W \in \mathcal{C}^*(v) - \{X\}$, $\mathcal{C}^*(v)$ is a co-partition of $V - v$ and $Z \neq V$, we can assume that Z intersects no member of $\mathcal{C}^*(v)$. Hence Z is the singleton $\{v\}$. Therefore, $V - v$ is tight and $d_{\mathcal{H}}(v) = m(v)$.

As $V - v$ is tight, (A3) implies that there exists a vertex $u \in V - v$ such that $V(uv) \neq \{u, v\}$, that is, $\{u, v\}$ is not tight. Let W_1 be the member of the co-partition $\mathcal{C}^*(v)$ of $V - v$ which does *not* contain this u. Since $\{u, v\}$ is not tight and W_1 is a minimal co-tight set in $\mathcal{H}$ with $u, v \notin W_1$, $|V - W_1| \geq 3$. Lemma 3.24 implies that $0 < m(v) < \ell$. Hence Lemma 2.13 implies $d_{\mathcal{H}}(v, \widetilde{W_1}^v) = d_{\mathcal{H}[V-W_1]}(v) \geq m(v)$. Let W_2 be another member of $\mathcal{C}^*(v)$. Now the tightness of $V - W_2$ in $\mathcal{H}$ and Lemma 2.13 imply that $d_{\mathcal{H}}(v, \widetilde{W_2}^v) = d_{\mathcal{H}[V-W_2]}(v) > 0$. However, since $\mathcal{C}^*(v)$ is a co-partition of $V - v$, $\widetilde{W_1}^v$ and $\widetilde{W_2}^v$ are disjoint, hence $m(v) = d_{\mathcal{H}}(v) \geq d_{\mathcal{H}}(v, \widetilde{W_1}^v) + d_{\mathcal{H}}(v, \widetilde{W_2}^v) > m(v)$, a contradiction. □

Now we are ready to finish the proofs of Theorems 3.9, 3.10 and 3.11. They can be proved in an identical way hence we provide their proofs together. When applying Lemmas 3.26, 3.29 and 3.32 we may assume the corresponding set of conditions to apply.

Proofs of Theorems 3.10, 3.9 and 3.11. We have already seen that Lemma 3.20 implies the disjointedness of the (m, ℓ)-MCT sets of $\mathcal{H}$ when $\ell \leq 0$. Hence it is enough to prove the statement for $\ell > 0$.

Let us suppose that there exist $X, Y \in \mathcal{C}^*$ such that $X \cap Y \neq \emptyset$. By Lemma 3.20, $|X \cup Y| \geq |V| - 1$ holds. By Lemma 3.21 we might even assume that $|X \cup Y| = |V| - 1$.

If there exist two MCT sets X, Y, so that $X \cup Y = V - v$ for a vertex $v \in V$, then according to Lemma 3.22 we may assume that $|\mathcal{C}^*(v)| \geq 3$ and $\mathcal{C}^*(v)$ forms a co-partition of $V - v$. However then by Lemma 3.26, 3.29 and 3.32, respectively, there exists a pair $u_1, u_2 \in V$ such that $\mathcal{T}(u_1u_2) = \mathcal{H}$. □

3.4 Complexity results

In this section we prove that the general augmentation problem is NP-hard for every (k, ℓ) pair, when $\ell > k$. Moreover, our method also implies that there exists no polynomial time constant factor approximation algorithm for this problem if P≠NP. García and Tejel showed in [22] that the general augmentation problem for graphs is NP-hard when $k = 2$ and $\ell = 3$. Our construction is based on their idea.

We use graphs instead of hypergraphs in this section, and we use (k, ℓ)-tightness instead of (m, ℓ)-tightness, as we can prove the NP-hardness even with these restrictions.

First, we show the NP-hardness of the following problem, called the **Colored Tight Augmentation** problem or **CTA** problem.

Problem 2 *Let $G = (V, E)$ be a (k, ℓ)-tight graph such that the edges in E are colored red or black. Find a graph H on the same vertex set with the minimum number of edges, such that each black edge of G is (k, ℓ)-redundant in $G \cup H$.*

The CTA problem is an extension of the general problem: given an instance $\bar{G} = (V, \bar{E})$ of the general problem, one can get an instance $G = (V, E_r \cup E_b)$ of the CTA problem by taking a spanning (k, ℓ)-tight subgraph of $\bar{G}$ and coloring each edge, which is already redundant in $\bar{G}$ to red and each edge not redundant in $\bar{G}$ to black. Now, for a graph $H = (V, F)$, $\bar{G} \cup H$ is (k, ℓ)-redundant if and only if all the black edges of G are (k, ℓ)-redundant in $G \cup H$ since the set of the redundant edges of $\bar{G} \cup H$ is equal to $(\bar{E} - (E_r \cup E_b)) \cup R_G(\bar{E} - (E_r \cup E_b) \cup F) = (\bar{E} - (E_r \cup E_b)) \cup (\bigcup_{e \in \bar{E} - (E_r \cup E_b)} \mathcal{T}_G(e)) \cup (\bigcup_{f \in F} \mathcal{T}_G(f)) = (\bar{E} - (E_r \cup E_b)) \cup E_r \cup R_G(F) = (\bar{E} - E_b) \cup R_G(F)$ by Lemma 2.12 and the definition of E_r. Thus augmenting all the black edges to redundant solves the general problem.

By extending the work of García and Tejel [22], we shall prove that the CTA problem is NP-hard for every (k, ℓ) where $k > 1$, that is, also for $0 < \ell \leq k$, in which case the general augmentation problem is solvable in polynomial time. On the other hand, for the $(k, \ell) = (2, 3)$

case the CTA problem is equivalent to the general augmentation problem: adding parallel edges to the red ones reduces the solution of the CTA problem to the solution of the general problem. (We note that the same construction works whenever $\ell = 2k-1$. Also, our general construction results in the very same graph for $\ell = 2k-1$.)

Our NP-hardness proofs for both the CTA problem and the general problem are based on the following statement which extends the well-known fact that the addition of a single vertex of degree k maintains the (k,ℓ)-tightness (or the (k,ℓ)-rigidity of a graph).

Lemma 3.34 *Let $G = (V, E)$ be a (k,ℓ)-tight graph on at least 4 vertices, let $G' = (V' \cup S, E')$ be a connected (k,ℓ)-sparse graph where $V' \neq \emptyset$, $V \cap V' = \emptyset$ and $\emptyset \neq S \subseteq V$. Let $G^* = (V \cup V', E \cup E')$. Suppose that $e_{G'}(X) > k|X|$ holds for each $\emptyset \neq X \subsetneq V'$, and $e_{G'}(V') = |E'| = k|V'|$. Then*

(a) G^* is (k,ℓ)-tight and V' is a (k,ℓ)-MCT set in G^*.

(b) Assume that there are at least two disjoint (k,ℓ)-MCT sets C_1 and C_2 in G which do not intersect S. Then the (k,ℓ)-MCT sets of G^* are exactly V' and those (k,ℓ)-MCT sets of G that do not intersect S.

(c) Moreover, with the assumption of (b), if X, Y are (k,ℓ)-MCT sets of G with $X \cap S \neq \emptyset$ while $Y \cap S = \emptyset$, then for each $v' \in V'$, $x \in X$ and $y \in Y$, $\mathcal{T}_G(xy) \subset \mathcal{T}_{G^*}(v'y)$ holds.

Lemma 3.34 is illustrated by Figure 3.1 where the graph $\mathbf{G}$ - formed by the solid edges - can be considered as the result of a method described in Lemma 3.34, where the role of G from the lemma is taken by $\mathbf{G} - X$, $S = N(X)$ and V' is X. Observe that the MCT sets in $\mathbf{G} - X$ are the singletons $\{u\}$, $\{v\}$ and $\{z\}$ and the MCT sets in $\mathbf{G}$ are $\{u\}$, $\{v\}$ and X, that is, z is removed from the family of $(3,4)$-MCT sets as claimed by Lemma 3.34(b). $\mathcal{T}_{\mathbf{G}}(ux)$ is the graph formed by the bold edges, thus $z \in V_{\mathbf{G}}(ux)$. This implies that $\mathcal{T}_{\mathbf{G}-X}(uz) \subset \mathcal{T}_{\mathbf{G}}(ux)$ holds by Lemma 2.11, as claimed by Lemma 3.34(c).

Proof (a) It follows by the construction that G^* induces $k|V \cup V'| - \ell$ edges. Hence we need to prove the (k,ℓ)-sparsity of G^*. Let $X \subseteq V' \cup V$. Now, $i_{G^*}(X) = i_G(X \cap V) + |E'| - e_{G'}(V' - X) \leq k|X \cap V| - \ell + k|V'| - k|V' - X| = k|X \cap V| - \ell + k|X \cap V'| = k|X| - \ell$ by the (k,ℓ)-sparsity of G and the assumption on $e_{G'}$. If G^* is (k,ℓ)-tight, then it follows by the assumption on $e_{G'}$ that V' is MCT in G^*.

(b) V' is an MCT set in G^* by part (a). Observe that $e_{G^*}(Z) = e_G(Z)$ holds for each $Z \subseteq V - S$. This implies that every MCT set of G which does not intersect V' is an MCT set also in G^* (besides V'). Now the disjointedness of the sets C_1, C_2 and V' implies that the MCT sets in G^* are pairwise disjoint by Theorem 3.11. However, then, by Lemma 3.13, there is no MCT set $Z \neq V'$ in G^* which intersects $S \cup V'$.

(c) By Lemmas 3.13 and 3.14, $V' \cup N_{G^*}(V') = V' \cup S \subset V_{G^*}(v'y)$. Hence $\mathcal{T}_{G^*}(x'y) \subset \mathcal{T}_{G^*}(v'y)$ holds for each $x' \in X \cap S$ by Lemma 2.11. On the other hand, $\mathcal{T}_{G^*}(x'y) = \mathcal{T}_G(x'y)$ since $x', y \in V$ and V' is co-tight in G^*. Furthermore, $\mathcal{T}_G(x'y) = \mathcal{T}_G(xy)$ follows by Lemma 3.19, completing the proof. □

We shall use the **set cover** problem to show the NP-hardness of the CTA problem. Given a ground set X and a family $\mathcal{S} = \{S_1, \ldots, S_m\}$ of subsets of X, an optimal solution of the set cover problem is a subfamily $\mathcal{S}' \subseteq \mathcal{S}$ of sets with minimum cardinality such that their union is X.

Theorem 3.35 *Let $k > 1$ and $0 < \ell < 2k$ be two integers. The CTA problem is NP-hard on (k, ℓ)-tight graphs, moreover, there exists no polynomial time constant factor approximation algorithm for it if P≠NP.*

Proof Given an instance of the set cover problem, a family $\mathcal{S}$ on ground set X such that $|X| \geq 2k + 5$ and no two members of $\mathcal{S}$ cover X, let us construct a graph. (We note here that these assumptions do not change the complexity of the set cover problem.) First, take a (k, ℓ)-tight graph G_0 on a copy X' of X such that there are at least two disjoint MCT sets in G_0.

(Such a graph exists when $|X| \geq 2k + 5$. To show this, let us take a complete graph on $|X| - 2$ vertices. This induces $\frac{(|X|-2)(|X|-3)}{2}$ edges, which is always more than $k|X| - \ell$, since $\frac{(2k+3)(2k+2)}{2} = (2k + 3)(k + 1) > k(2k + 3) - \ell$, provided $\ell > 0$. Hence the complete graph on $|X| - 2$ vertices is (k, ℓ)-rigid. We can take a (k, ℓ)-tight spanning subgraph of it. Appending now two vertices with degree k each shows by Lemma 3.34 that there exists a tight graph on $|X|$ vertices that contains at least two disjoint MCT sets.)

Let us take another copy X'' of X and connect the copies x' and x'' of each $x \in X$ by an edge e_x. These $|X|$ edges will be the only black edges in our final graph (that are, edges that need to become redundant); every other edge will be red. Add new edges (not parallel to the previous ones) between X' and X'' until $d(x'') = k$ holds for every $x'' \in X''$. Let us denote the graph we got with these steps by G_1. Now, Lemma 3.34 and Lemma 3.14 imply the following.

Claim 3.36 *G_1 is (k,ℓ)-tight and the family of MCT sets in G_1 equals the family formed by all one element subsets of X''. Furthermore, $\mathcal{T}_{G_1}(x''y'')$ induces exactly two black edges e_x and e_y for each pair $x, y \in X$.* □

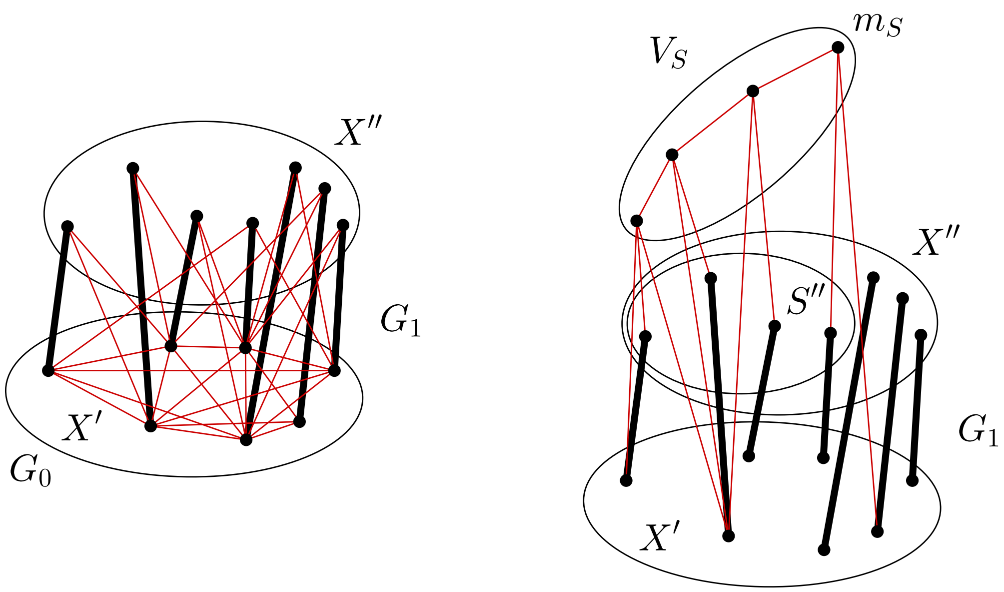


Figure 3.4: Construction of G^* for $(k,\ell) = (3,5)$. On the right side, we do not show the red (thin) edges of G_1.

Now for every $S \in \mathcal{S}$ we make the following extension on G_1. Let S'' denote the copy of S in X''. Start with $V_S = \emptyset$. First, choose one vertex from S'' and $k-1$ vertices from X' and add a new vertex v with (red) edges to these k vertices. Add v to V_S. Later, when V_S is not empty, take the last vertex that is added to V_S, say v, one new vertex from S'', and $k-2$ vertices from X' and add a new vertex w of degree k connecting to these k vertices. Add also w to V_S. Repeat the above addition until there are no more vertices in S'' that were not used in such a step. (See Figure 3.4)

Let us denote the graph that we get from G_1 after running the procedure above for each $S \in \mathcal{S}$ by $G^* = (V^*, E^*)$. For an arbitrary $S \in \mathcal{S}$, $d(v) = k$ holds for every $v \in V_S$ at the moment when it is added to V_S, however, in the subsequent step this is increased to $k+1$, except for the last vertex added to V_S that we call m_S. By Lemma 3.34(b), this shows that the only MCT sets in G^* are the singletons $M_S := \{m_S\}$ for all $S \in \mathcal{S}$. By the construction, it is also easy to see that V_S is co-tight in G^* for every $S \in \mathcal{S}$. By Lemma 3.19, we may assume that the optimal solution of the CTA problem consists of edges between MCT sets, that is, between the vertices m_S ($S \in \mathcal{S}$).

Claim 3.37 *Let $S_i, S_j \in \mathcal{S}$ and let $e = m_{S_i} m_{S_j}$ be an edge connecting the two (k,ℓ)-MCT sets M_{S_i} and M_{S_j} of G^*. Then a black edge e_x is contained in $\mathcal{T}_{G^*}(e)$ if and only if $x \in S_i \cup S_j$.*

Proof Since V_S is co-tight for each $S \in \mathcal{S}$, $\mathcal{T}_{G^*}(e) \subseteq G^*[X' \cup X'' \cup V_{S_i} \cup V_{S_j}] =: G_{ij}$ by Lemma 2.11, and hence $\mathcal{T}_{G^*}(e) = \mathcal{T}_{G_{ij}}(e)$. Now, by using Lemma 3.34(c) several times following the construction of V_{S_i} and V_{S_j}, we get that $\mathcal{T}_{G_0}(x''y'') \subset \mathcal{T}_{G_{ij}}(e)$ for each pair $x, y \in S_i \cup S_j$. Hence $\mathcal{T}_{G^*}(e) = \mathcal{T}_{G_{ij}}(e)$ induces the black edge e_x for each $x \in S_i \cup S_j$ by Claim 3.36. On the other hand, $z'' \notin V_{G_{ij}}(e)$ for $z \in X - (S_i \cup S_j)$ since $\{z''\}$ is co-tight in G_{ij} by Lemma 3.34(b). Hence the black edge e_z is not induced by $\mathcal{T}_{G^*}(e) = \mathcal{T}_{G_{ij}}(e)$. □

Remember, that we only need to add edges so that all the black edges become redundant. Thus from Claim 3.37 it is easy to see that any (not necessarily optimal) solution of the CTA problem on G^* of cardinality q gives a (not necessarily optimal) solution of the set cover problem that uses at most $2q$ sets and every (not necessarily optimal) solution of the set cover problem with cardinality q gives a (not necessarily optimal) solution of the CTA problem with cardinality $\lceil \frac{q}{2} \rceil$. However, there is no constant factor approximation of the set cover problem unless P=NP by [58]. Therefore, there is no constant factor approximation of the CTA problem unless P=NP. This finishes the proof of Theorem 3.35. □

To show the NP-hardness of the general problem for $k < \ell$, we shall modify the above NP-hardness construction in such a way that we add $2k - \ell$ parallel copies of each edge during the construction. Also, we shall attach MCT sets – other than the singletons used in the above proof – using Lemma 3.34.

Theorem 3.38 *Let k and ℓ be two integers such that $0 < k < \ell < 2k$. Then the general augmentation problem is NP-hard on (k,ℓ)-rigid graphs, moreover, there exists no polynomial time constant factor approximation algorithm for it if P≠NP.*

Proof Let a denote the greatest common divisor of $2k-\ell$ and $\ell-k$, and let $b = \frac{2k-\ell}{a}$ and $c = \frac{\ell-k}{a}$. Let us define a tree $H = (V \cup F, E)$ on $b+c+1$ vertices, called a **(k,ℓ)-caterpillar**, as follows. Let $V = \{v_1, \ldots, v_b\}$, and let F be a set of vertices with cardinality of $c+1$. Let $E = P \cup L$, where $P = \{v_i v_{i+1} : i = 1, \ldots, b-1\}$ is a path on V, and L consists of edges between V and F in such a way that $d_L(v_1) = \lceil \frac{\ell-k}{2k-\ell} \rceil$, $d_L(v_i) = \lceil \frac{\ell-k}{2k-\ell} i \rceil - \lceil \frac{\ell-k}{2k-\ell}(i-1) \rceil$ for each $i \in \{2, \ldots, b-1\}$ and $d_L(v_b) = \lceil \frac{\ell-k}{2k-\ell} b \rceil - \lceil \frac{\ell-k}{2k-\ell}(b-1) \rceil + 1$. Hence $|L| = \lceil \frac{\ell-k}{2k-\ell} b \rceil + 1 = c+1$ thus we can distribute the edges so that $d_L(f) = 1$ for each $f \in F$. See Figure 3.5 for an illustration in the case of $(k, \ell) = (14, 22)$.

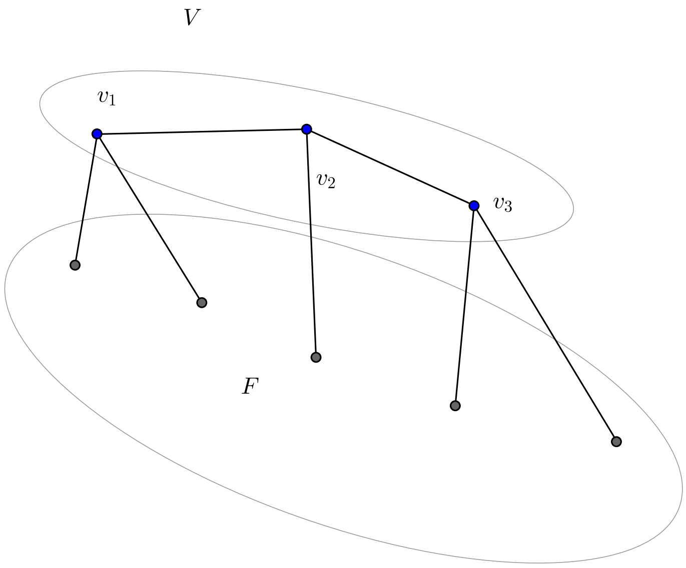


Figure 3.5: A $(14, 22)$-caterpillar H. In this case, $a = 2$, $b = 3$ and $c = 4$. One can observe that $6H$ is indeed $(14, 22)$-sparse and satisfies the conditions of Lemma 3.34.

For a set E and $c \in \mathbb{Z}_+$ let $\boldsymbol{cE}$ denote the multiset that arises by taking c copies of each element of E. For a graph $G = (V, E)$, $\boldsymbol{cG}$ denotes the graph (V, cE). Let us consider the graph $H' = (2k - \ell)H$, that is, a (k, ℓ)-caterpillar $2k - \ell$ times. As $|P| + |L| = b + c$, H' has $\frac{2k-\ell+\ell-k}{a}(2k - \ell) = kb = k|V|$ edges.

$(2k - \ell)$ parallel copies of a tree is (k, ℓ)-sparse, since every $X \subset V$ for which $|X| \geq 2$ induces at most $i(X) \leq (|X|-1)(2k-\ell) = 2k|X|-2k-\ell(|X|-1) \leq k|X|+k(|X|-2)-\ell(|X|-1) \leq k|X|-\ell$, where the last inequality comes from $k < \ell$. Hence H' is (k, ℓ)-sparse.

Moreover, we claim that $e_{H'}(X) > k|X|$ for each $X \subsetneq V$. First let us prove it to sub-paths of P. Let us start with a subpath P' starting from v_1. Now $e_{H'}(P') = (2k - \ell)(j + \lceil \frac{\ell-k}{2k-\ell} j \rceil) \geq (2k - \ell)(j + \frac{\ell-k}{2k-\ell} j) = kj = k|P'|$, and equation holds only if $P' = P$. Consider now the sub-path from $v_1 \neq v_i$ to v_j and denote it by P'. Let us denote the "upper fractional" of a real X by $/X/$. That is, $/X/ = \lceil X \rceil - X$. Now $e_{H'}(P') = (2k - \ell)(j - i + 2 + \lceil \frac{\ell-k}{2k-\ell} j \rceil - \lceil \frac{\ell-k}{2k-\ell}(i-1) \rceil) = (2k - \ell)(j - i + 2 + \frac{\ell-k}{2k-\ell} j + \Big/ \frac{\ell-k}{2k-\ell} j \Big/ - \frac{\ell-k}{2k-\ell}(i-1) - \Big/ \frac{\ell-k}{2k-\ell}(i-1) \Big/) = kj - ki + 3k - \ell + (2k - \ell)(\Big/ \frac{\ell-k}{2k-\ell} j \Big/ - \Big/ \frac{\ell-k}{2k-\ell}(i-1) \Big/) > kj - ki + 3k - \ell - 2k + \ell = k(j - i + 1) = k|P'|$. (The unequality holds since $0 \leq /A/ < 1$.) By this it is clear that $e_{H'}(X) > k|X|$ for each $X \subsetneq V$. Therefore, H' fulfills the conditions of Lemma 3.34.

We can now use H' in the proof of Theorem 3.35 instead of vertices of degree k since $|S| = c+1 \geq 2$. The only difference is that we do not naturally have the second copy X'' in the construction of G_1. Instead, we perform $|X|$ extensions on G_0 with H' in such a way that in the step where we added V^x for $x \in X$ in the proof of Theorem 3.35, now we connect V^x only to X' and one of the edges, say, e_x connecting V^x and X' will connect an element $x'' \in V^i$ to the copy x' of x. The second copy X'' of X (in which we take the copy S'' of each set $S \in \mathcal{S}$ in the final phase of the construction) will be the set of these vertices x''. The $(2k-\ell)$ parallel copies of the edges e_x for all $x \in X$ will be the only black edges in our final graph (that is, edges that need to be redundant in the end); every other edge will be red.

This way, for each input $\mathcal{S}$ on ground set X of the set cover problem, we obtain an instance G^* of the CTA problem on which a solution of any (not necessarily optimal) solution of the CTA problem of cardinality q gives a (not necessarily optimal) solution of the set cover problem that uses at most $2q$ sets and every (not necessarily optimal) solution of the set cover problem with cardinality q gives a (not necessarily optimal) solution of the CTA problem on G^* with cardinality $\lceil \frac{q}{2} \rceil$. Observe that each edge of G^* is present $2k-\ell$ times except the edges of G_0.

Now, given the instance G^*, defined above, we construct an instance $\bar{G}^*$ of the general problem by adding a new parallel copy to each red edge of G^*. It may mean more parallel new edges added between two vertices. It is clear that every red edge of G^* is (k,ℓ)-redundant in $\bar{G}$. On the other hand, for an edge e of G_0, $\mathcal{T}_{G^*}(e)$ is a subgraph of the (k,ℓ)-tight subgraph G_0 of G^* by Lemma 2.11, while for an edge $e = uv$ that we added later in the construction of G^*, $\mathcal{T}_{G^*}(e) = G^*[\{u,v\}]$ since the set $\{u,v\}$ induces $2k-\ell$ parallel edges in G^*, and hence it is (k,ℓ)-tight in G^*. Therefore, no black edge of G^* is (k,ℓ)-redundant in $\bar{G}$, and hence a (not necessarily optimal, but constant approximation) solution of the general problem on $\bar{G}^*$ is a (not necessarily optimal, but constant approximation) solution of the CTA problem on G^* and vice versa. This finishes the proof of Theorem 3.38. $\square$

Chapter 4

(k, ℓ)-M-connected hypergraphs

In this chapter, we present our results concerning the connections between (k, ℓ)-redundancy and (k, ℓ)-M-connectivity. Most of these results were already known for $(k, \ell) = (2, 3)$ [32]. Here, we generalize them to every (k, ℓ) pair that satisfies $0 < \ell < 2k$, except for the result of Lemma 4.5 that requires $0 < \ell \leq \frac{3}{2}k$. Besides these generalizations, we present some new structural results on (k, ℓ)-M-components. This chapter lays the groundwork for Chapter 5. The results achieved in this chapter will also play a crucial role in efficiently determining the (k, ℓ)-M-components of a graph and computing the (k, ℓ)-M-component hypergraph of a (k, ℓ)-rigid graph (presented in Section 8.2). This chapter is based on [50, Section 3].

Here, we remind the reader to some well-known facts from matroid theory. An equivalence relation can be defined on the ground set S of an arbitrary matroid $\mathcal{M}$ (by using the circuit axioms of a matroid), as follows. Two elements $x, y \in S$ are equivalent if there exists a circuit C of $\mathcal{M}$ such that $x, y \in C$. The equivalence classes of this matroid are called *components* of $\mathcal{M}$.

We use this concept to the so-called (k, ℓ)-sparsity matroid, defined in Section 2.2. Remember, the edge sets of spanning (k, ℓ)-tight subgraphs of a (k, ℓ)-rigid graph G correspond to the bases of the (k, ℓ)-sparsity matroid on G. If G is not (k, ℓ)-rigid, this matroid is not of full rank. Nonetheless, regardless of the rigidity of G one can define the components of its (k, ℓ)-sparsity matroid. The components of the $(2, 3)$-sparsity matroid (that is, the 2-dimensional rigidity matroid) are sometimes called the *M-components* of G (see, e.g., in [32]). By extending this notion to other sparsity matroids, we will call a component of the (k, ℓ)-sparsity matroid of G a $\boldsymbol{(k, \ell)}$**-M-component**. Note that if an edge e of G is not redundant, then $\{e\}$ is a (k, ℓ)-M-component of G and it is called a **trivial** (k, ℓ)-M-component of G. Here, we refer to the more general definition of redundant edges that can be understood in non-(k, ℓ)-rigid graphs, as well, see in

Section 2.2. (See Figure 5.1 for an illustration of nontrivial $(2,3)$-M-components in a $(2,3)$-rigid graph.) We note that here our definition of (k,ℓ)-M-component stands for arbitrary graphs, while in [50] we defined this concept exclusively for (k,ℓ)-rigid graphs.

Let us also show the following properties of the (k,ℓ)-M-components.

Observation 4.1 *For a graph G let C be a nontrivial (k,ℓ)-M-component of G. Then C is an induced subgraph of G.*

Proof Suppose that $i,j \in V(C)$. Then there exist edges e_i and e_j, incident to i and j, respectively, for which $e_i, e_j \in C$. Now, there is a circuit $C' \subseteq C$ containing e_i and e_j, hence $i,j \in V(C')$. However, this means that there exists a (k,ℓ)-tight subgraph $T \subset C'$ for which $i,j \in V(T)$ and hence $\mathcal{T}_T(ij) \subset C'$ by Lemma 2.9. If ij is an edge of G, then $\mathcal{T}_T(ij) + ij$ is a circuit that intersects C', thus the equivalence relation on the matroid circuits shows that $ij \in C$. $\square$

As we defined the (k,ℓ)-M-components to non-rigid graphs, we need to claim that the (k,ℓ)-M-components still form (k,ℓ)-rigid subgraphs. We note that this statement is far from trivial, the proof of it can be found in [21, Lemma 2.4].

Lemma 4.2 *[21] Let G' be a (k,ℓ)-M-component subgraph of the graph G. Then G' is (k,ℓ)-rigid.* $\square$

Lemma 4.3 *Let $G=(V,E)$ be a graph and let $G^*=(V,E^*)$ be an arbitrary maximal spanning (k,ℓ)-sparse subgraph of G. Then every trivial (k,ℓ)-M-component of G is contained in E^*, and, for any nontrivial (k,ℓ)-M-component C of G, $i_{G^*}(V(C)) = k|V(C)| - \ell$.*

Proof If C is a trivial (k,ℓ)-M-component of G, then C consists of a single non-redundant edge e of G. Remember, in this case non-redundant means that the number of edges in the maximal (k,ℓ)-sparse subgraph of G is *not* equal to the number of edges in the maximal (k,ℓ)-sparse subgraph of $G-e$. Thus e must be an edge of G^*.

Suppose now that C is nontrivial. Let $B = E^* \cap C$, that is, $i_{G^*}(V(C)) = |B|$. Now B must be a maximal (k,ℓ)-sparse subgraph of C, since otherwise we may add edges from C to G^* while still maintaining its sparsity (as the edges in C are only contained in (k,ℓ)-circuits of G consisting of the edges of C by the definition of a (k,ℓ)-M-component). This, together with Lemma 4.2 shows that $|B| = k|V(C)| - \ell$. $\square$

If G has only one (k,ℓ)-M-component, then it is called **(k,ℓ)-M-connected**. With a straightforward extension of [32, Lemma 3.1] (that claims the same for $(2,3)$-M-connectivity and $(2,3)$-redundancy) it is easy to see that the (k,ℓ)-M-connectivity of a graph implies its (k,ℓ)-redundancy.

Lemma 4.4 *If G is (k,ℓ)-M-connected, then G is (k,ℓ)-redundant.*

Proof By Lemma 4.2 G itself is (k,ℓ)-rigid. Since G is (k,ℓ)-M-connected, every edge is contained in a (k,ℓ)-M-circuit of G, hence they are all (k,ℓ)-redundant resulting in G being (k,ℓ)-redundant. □

The converse implication is not always true. However, for our purpose, extending the result from Jackson and Jordán [32] is enough. They proved the following to $(k,\ell)=(2,3)$.

Lemma 4.5 *Let k and ℓ be positive integers such that $0<\ell\leq\frac{3}{2}k$ and let G be a $(c_{k,\ell}+1)$-connected and (k,ℓ)-redundant graph where $c_{k,\ell}:=\left\lceil\frac{\ell}{k}\right\rceil$. If $k<\ell$, then also suppose that G has no two vertices which are connected by more than $2k-\ell$ edges. Then G is (k,ℓ)-M-connected.*

Proof Suppose that G is not (k,ℓ)-M-connected and let $H_1,\dots,H_q$ be its (k,ℓ)-M-components. Notice that $|H_i|\neq 1$ for $i=1,\dots,q$, because G is (k,ℓ)-redundant. Let $X_i=V(H_i)-\bigcup_{j\neq i}V(H_j)$ denote the set of vertices that do not belong to any (k,ℓ)-M-component other than H_i. Let $Y_i=V(H_i)-X_i$. Clearly $|V|=\sum\limits_{i=1}^{q}|X_i|+|\bigcup\limits_{i=1}^{q}Y_i|$ and $\sum\limits_{i=1}^{q}|Y_i|\geq 2|\bigcup\limits_{i=1}^{q}Y_i|$ hence $|V|\leq\sum\limits_{i=1}^{q}|X_i|+\frac{1}{2}\sum\limits_{i=1}^{q}|Y_i|$. Moreover, notice that by the $(c_{k,\ell}+1)$-connectivity of G $|Y_i|\geq c_{k,\ell}+1$. (More precisely we can only claim that $|Y_i|\geq c_{k,\ell}+1$ when $|V(H_i)|\geq c_{k,\ell}+1$, however, this is obvious if $c_{k,\ell}=1$ and follows from our assumption on the number of parallel edges in G if $k<\ell$ and thus $c_{k,\ell}=2$.)

Let us now choose a (k,ℓ)-tight subgraph $G^*=(V,E^*)$ of G. Let $B_i=H_i\cap E^*$ for $i=1,\dots,q$. Note that $\bigcup\limits_{i=1}^{q}B_i=E^*$. Hence, by using the above inequalities and Lemma 4.3, we get $k|V|-\ell=|\bigcup\limits_{i=1}^{q}B_i|=\sum\limits_{i=1}^{q}|B_i|=\sum\limits_{i=1}^{q}(k|V(H_i)|-\ell)=k\sum\limits_{i=1}^{q}|X_i|+k\sum\limits_{i=1}^{q}|Y_i|-q\ell=k(\sum\limits_{i=1}^{q}|X_i|+\frac{1}{2}\sum\limits_{i=1}^{q}|Y_i|)+\frac{k}{2}\sum\limits_{i=1}^{q}|Y_i|-q\ell\geq k|V|+\frac{k}{2}\sum\limits_{i=1}^{q}|Y_i|-q\ell\geq k|V|+\frac{k(c_{k,\ell}+1)q}{2}-q\ell$. If $0<\ell\leq k$, then the previous inequality gives $k|V|-\ell\geq k|V|+q\frac{2}{2}k-q\ell>k|V|-\ell$, a contradiction. If $k<\ell\leq\frac{3}{2}k$, then it gives $k|V|-\ell\geq k|V|+q\frac{3}{2}k-q\ell>k|V|-\ell$, also a contradiction. □

Notice that, for example, if G is simple, then G surely has no two vertices which are connected by more than $2k-\ell$ edges.

For a graph $G = (V, E)$, let $\boldsymbol{\mathcal{H}_G = (V, \mathcal{E})}$ be a hypergraph, called the **(k,ℓ)-M-component hypergraph** of G, such that $\mathcal{E}$ consists of the not (k,ℓ)-redundant edges of E and $k|V(C)| - \ell$ parallel copies of the hyperedge formed on $V(C)$ for each nontrivial (k,ℓ)-M-component C of G. This hypergraph was defined previously by Fekete and Jordán [16] for $(k,\ell) = (2,3)$ and rigid graphs.

Lemma 4.6 *Let $G = (V, E)$ be a graph, let G^* be a maximal spanning (k,ℓ)-sparse subgraph of G and let $\mathcal{H}_G$ be the (k,ℓ)-M-component hypergraph of G. Then $i_{\mathcal{H}_G}(X) \leq i_{G^*}(X)$ holds for each $X \subseteq V$. Furthermore, equality holds exactly when X induces either all or none of the edges of each (k,ℓ)-M-component of G.*

Proof Let E' denote the set of not (k,ℓ)-redundant edges of G and $H_1, \ldots, H_t$ denote the non-trivial (k,ℓ)-M-components of G.

Note that $|G^* \cap H_i| = k|V(H_i)| - \ell = i_{\mathcal{H}_G}(V(H_i))$ holds for every $i = 1, \ldots, t$ by Lemma 4.3. Notice that, for each $e \in E'$, the relations $e \in E^*$ and $e \in \mathcal{H}_G$ also hold. Recall that the (k,ℓ)-M-components partition the edge set of G and the nontrivial (k,ℓ)-M-components are induced subgraphs by Observation 4.1. Observe also that, for $X \subseteq V$ and $i \in \{1, \ldots, t\}$, either $X \cap V(H_i)$ induces no hyperedge in $\mathcal{H}_G$ or $V(H_i) \subseteq X$. Hence, we have $i_{G^*}(X) = i_{E'}(X) + \sum_{i=1}^{t} i_{G^*}(X \cap V(H_i)) \geq i_{E'}(X) + \sum_{i=1}^{t} i_{\mathcal{H}_G}(X \cap V(H_i)) = i_{\mathcal{H}_G}(X)$ for each $X \subseteq V$ where equality holds exactly when for all $i = 1, \ldots, t$ either $X \cap V(H_i)$ induces no edge in G^* or $V(H_i) \subseteq X$. $\square$

Lemma 4.6 has the following corollary.

Observation 4.7 *If G is a (k,ℓ)-rigid graph, then the (k,ℓ)-M-component hypergraph $\mathcal{H}_G$ of G is a (k,ℓ)-tight hypergraph. Furthermore, if X induces a (k,ℓ)-tight subhypergraph of $\mathcal{H}_G$, then $G[X]$ is a (k,ℓ)-rigid subgraph of G.*

The following lemma may be understood as the converse of Lemma 4.3.

Lemma 4.8 *Let $\mathcal{H} = (V, \mathcal{E})$ be a (k,ℓ)-tight hypergraph. Suppose, for a hyperedge $e \in \mathcal{E}$ that e has exactly $k|V(e)| - \ell$ parallel copies in $\mathcal{E}$. Let $\mathcal{H}'$ be the hypergraph we get by deleting all the $k|V(e)| - \ell$ parallel copies of e from $\mathcal{E}$ and inserting an arbitrary (k,ℓ)-tight spanning subgraph on $V(e)$. Then $\mathcal{H}'$ is also (k,ℓ)-tight.*

Proof As the number of (hyper)edges does not change we only need to show the (k,ℓ)-sparsity of $\mathcal{H}'$. For the sake of contradiction suppose that $\mathcal{H}'$ is not (k,ℓ)-sparse. If it is not sparse, then also by the number of hyperedges it contains at least one circuit. Let Y denote the vertex set of a circuit in $\mathcal{H}'$. By the (k,ℓ)-sparsity of $\mathcal{H}$, $|V(e) \cap Y| \geq 2$. Hence Lemma 2.9 may be used on the (k,ℓ)-tight subgraph of $\mathcal{H}'$ induced by $V(e)$ and on Y minus one edge which is not induced by $V(e)$. This shows that $V(e) \cup Y$ induces a (k,ℓ)-rigid subgraph in $\mathcal{H}'$ that is not (k,ℓ)-tight which contradicts $i_{\mathcal{H}}(V(e) \cup Y) = i_{\mathcal{H}'}(V(e) \cup Y)$. □

We shall use the following lemma to show that the global rigidity augmentation problem – formalized as Problem 4 in Chapter 5 – is polynomially solvable for all rigid inputs (in contrast to the rigidity augmentation problem, see Section 3.4).

Lemma 4.9 *Let $G = (V, E)$ be a (k,ℓ)-rigid graph, let $\mathcal{H}_G = (V, \mathcal{E})$ be the (k,ℓ)-M-component hypergraph of G and let F be an edge set on V.*

(i) If $G + F$ is (k,ℓ)-M-connected, then $\mathcal{H}_G + F$ is (k,ℓ)-redundant.

(ii) If $\mathcal{H}_G + F$ is (k,ℓ)-redundant, then $G + F$ is (k,ℓ)-redundant.

Proof (i) As $\mathcal{H}_G$ is a (k,ℓ)-tight hypergraph by Observation 4.7, each $f \in F$ is redundant in $\mathcal{H}_G + F$. Let us take now a hyperedge $e' \in \mathcal{E}$. Let $e \in E$ be any edge from the (k,ℓ)-M-component corresponding to e'. As $G + F$ is (k,ℓ)-M-connected, there exists an M-circuit C of $G + F$ such that $e, f \in C$ for any $f \in F$. Let us choose a (k,ℓ)-tight spanning subgraph $G^* = (V, E^*)$ of G such that $C - f \subset E^*$. This can be done by the matroid base axioms. Clearly, $e \in \mathcal{T}_{G^*}(f)$. Now $i_{\mathcal{H}_G}(X) \leq i_{G^*}(X)$ holds by Lemma 4.6 for all $X \subseteq V(\mathcal{T}_{G^*}(f))$, which results that $V(\mathcal{T}_{G^*}(f)) \subseteq V(\mathcal{T}_{\mathcal{H}_G}(f))$ by Lemma 2.11. This shows that $e' \in \mathcal{T}_{\mathcal{H}_G}(f)$ implying that e' is redundant in $\mathcal{H}_G + F$.

(ii) As G is a (k,ℓ)-rigid graph, each $f \in F$ is redundant in $G + F$. It is also obvious that every edge that is contained in a nontrivial (k,ℓ)-M-component is redundant by Lemma 4.4. Now let us consider an edge e that is not redundant in G. That is, $e \in E \cap \mathcal{E}$. Now, as $\mathcal{H}_G$ is (k,ℓ)-tight and $\mathcal{H}_G + F$ is (k,ℓ)-redundant, there is an $f \in F$, such that $e \in \mathcal{T}_{\mathcal{H}_G}(f)$ thus $\mathcal{H}_G - e + f$ is (k,ℓ)-tight. Now by using Lemma 4.8 sequentially on the nontrivial hyperedges starting with $\mathcal{H}_G - e + f$ we can get a (k,ℓ)-tight graph G^*, as the conditions of Lemma 4.8 are met after every step we made. In every step an arbitrary (k,ℓ)-tight subgraph can be inserted, hence we may insert the one from G provided by Lemma 4.3. Thus $G^* \subset G$, G^* is (k,ℓ)-tight and $e \notin G^*$. This shows that e is (k,ℓ)-redundant in G. □

Now by Lemma 4.9 if G meets the conditions of Lemma 4.5, then we can make G (k, ℓ)-redundant by adding edges that make $\mathcal{H}_G$ (k, ℓ)-redundant. This latter can be done polynomially by Theorem 3.1.

Chapter 5

Globally rigid augmentation of rigid graphs

Let us return to our motivating examples from the introduction. Suppose that we have some sensors in the plane with known distances between some pairs of these sensors. At least how many sensor-locations do we need to measure exactly to be able to reconstruct the exact location of each sensor? This is the so-called *global rigidity pinning* (or anchoring) problem, investigated in detail in Section 7.1. Sometimes measuring the exact sensor-locations is too expensive or even impossible. Instead, one may ask at least how many new distances need to be measured so that the distances uniquely determine the positions of the sensors (up to isometry). This problem is called the *global rigidity augmentation* problem. The global rigidity augmentation problem was already investigated in $\mathbb{R}^2$ and Jordán showed a constant factor approximation algorithm for it [39]. In this chapter, we consider an exact solution of a more general problem with rigid graph inputs. (Recall that reconstructing the position of the sensors is a challenging task, even if they are uniquely determined by the framework (see [2, 43, 68]), and we do not address this problem in this dissertation.)

This chapter is based on the results published in [50], which is an extended version of the paper presented in the CIAC21 conference [47]. A preliminary version of [50] can be found -in the technical report [48]. We note that the results of this chapter are more general then the results presented in [50], as there we only expanded the proof of the $(k, \ell) = (2, 3)$ special case, while sketching the proofs for other (k, ℓ) pairs. In this chapter, we provide the proof for every (k, ℓ) pair with $k < \ell \leq \frac{3}{2}k$. We sketch the proof in the case of $\ell \leq k$, because it does not need any additional ideas, just the right application of the previous results.

Besides rigidity and global rigidity in $\mathbb{R}^2$ (which are characterized in Theorems 2.1 and 2.3, respectively), there are some other types of frameworks for which both rigidity and global rigidity are characterized by sparsity and connectivity properties of their underlying graphs (with some genericity assumptions), for example for body-bar frameworks [11, 70, 72], for body-hinge and body-bar-hinge frameworks [34, 41, 71, 73, 77], and for bar-joint frameworks which are restricted to lying (and moving) on some given surface in $\mathbb{R}^3$ such as a sphere [13, 76] or a cylinder [35, 63].

In this chapter, we consider the following meta-problem related to the above-mentioned versions of rigidity and global rigidity.

Problem 3 *Given a graph $G = (V, E)$, find an edge set F of minimum cardinality on the same vertex set, such that $G + F = (V, E \cup F)$ is globally rigid.*

We give an optimal solution and a min-max theorem that can show a solution to this problem in various rigidity settings.

5.1 Preliminaries

In this section we collect the basic definitions and results that we shall use, including the formal definition of the combinatorial problem family solved.

To simplify the presentation of our results, let $\boldsymbol{c_{k,\ell}} := \max\left\{\left\lceil \frac{\ell}{k} \right\rceil, 0\right\}$, that is, $c_{k,\ell}$ is zero if $\ell \leq 0$, one if $0 < \ell \leq k$, and two if $k < \ell < 2k$. Observe, that this notation of $c_{k,\ell}$ is consistent with that of Lemma 4.5. With its help Lemma 2.9 can be rephrased as follows.

Lemma 5.1 *Let $\mathcal{H} = (V, \mathcal{E})$ be a (k, ℓ)-sparse hypergraph on at least three vertices, and let $\mathcal{H}_1 = (V_1, \mathcal{E}_1)$ and $\mathcal{H}_2 = (V_2, \mathcal{E}_2)$ be (k, ℓ)-tight subhypergraphs of $\mathcal{H}$. If $|V_1 \cap V_2| \geq c_{k,\ell}$, then $\mathcal{H}_1 \cup \mathcal{H}_2$ is a (k, ℓ)-tight subhypergraph of $\mathcal{H}$.* □

Also, Proposition 2.5 can be generalized to (k, ℓ)-rigid graphs. This result is folklore and parts of its proof can be found in [40, 54], but let us provide here a complete proof.

Proposition 5.2 *If $G = (V, E)$ is a (k, ℓ)-rigid graph for which $|V| \geq 3$, then G is $c_{k,\ell}$-connected.*

Proof If $\ell \leq 0$ and hence $c_{k,\ell} = 0$, the statement follows by the definition of 0-connectivity.

Suppose now that $k, \ell > 0$, hence $c_{k,\ell} \geq 1$. Let $G^* = (V, E^*)$ be a (k, ℓ)-tight spanning subgraph of G. Suppose that G is not $c_{k,\ell}$-connected, thus G^* is not $c_{k,\ell}$-connected, either.

Therefore, there exists a partition of V, so that $C_1 \cup X \cup C_2 = V$, C_1, X and C_2 are pairwise disjoint, $d_{G^*}(C_1, C_2) = 0$ and $|X| \leq c_{k,\ell} - 1$. Notice that X might be empty (for example, if $c_{k,\ell} = 1$).

By the (k, ℓ)-tightness of G^*, $i_{G^*}(C_1 \cup X) \leq k|C_1 \cup X| - \ell$ and $i_{G^*}(X \cup C_2) \leq k|X \cup C_2| - \ell$. On the other hand, $k(|C_1| + |X| + |C_2|) - \ell = |E^*| \leq i_{G^*}(C_1 \cup X) + i_{G^*}(X \cup C_2) \leq k(|C_1| + |X| + |C_2|) - \ell + |X| - \ell$, meaning $\ell \leq |X|$. This contradicts $|X| \leq c_{k,\ell} - 1$, as here $c_{k,\ell} = \left\lceil \frac{\ell}{k} \right\rceil$. $\square$

Based on Proposition 5.2 and inspired by Theorem 2.3, one may be interested in the following problem as an extension of Problem 1.

Problem 4 *Given a (k, ℓ)-rigid graph $G = (V, E)$ with $|V| > 3$, find a graph $H = (V, F)$ with a minimum cardinality edge set F, such that $G \cup H = (V, E \cup F)$ is (k, ℓ)-redundant and $(c_{k,\ell} + 1)$-connected.*

In this section, we give a min-max theorem and a polynomial algorithm for Problem 4 for all integer pairs of (k, ℓ) where $\max(0, \ell) \leq k$ and also for $0 < k < \ell \leq \frac{3}{2}k$ with the extra assumption that the input is a **simple** graph (that is, it contains no parallel edges and no loops). In all cases, the output edge set F can be provided in such a way that $F \cap E = \emptyset$ if such augmentation is possible (that is, if the complete graph on V is (k, ℓ)-redundant).

This latter condition is important, as we aim to use the results proved in Lemma 4.5. This is because any graph H, with minimum cardinality edge set for which $G + H$ is $(c_{k,\ell} + 1)$-connected and $\mathcal{H}_G + H$ is (k, ℓ)-redundant is also an optimal solution of Problem 4 by Lemmas 4.5 and 4.9. However, here the results of Chapter 3 help, as $\mathcal{H}_G$ is a (k, ℓ)-tight hypergraph by Observation 4.7.

Also, Problem 4 covers numerous versions of global rigidity. The most prominent is the global rigidity augmentation problem in $\mathbb{R}^2$ for rigid graphs. With the extra assumption that we use edges for the augmentation only from the complement of the input graph, a solution to Problem 4 also solves the global rigidity augmentation problem on the cylinder $C \subset \mathbb{R}^3$ for rigid graphs by Theorems 2.7 and 2.8. We note that the global rigidity augmentation problem in $\mathbb{R}^1$ can also be considered as a special case of Problem 4, however, its solution is much easier [14].

5.2 Connectivity augmentation

As we already investigated how to add an optimal edge set to a (k,ℓ)-tight hypergraph to get a (k,ℓ)-redundant hypergraph in Chapter 3, we shall now also consider how to augment its connectivity. Vertex-connectivity augmentation problems have quite extensive literature (see [14, 30, 38] for related results and [18] for a survey) of which we only need some basic ones due to the special conditions of our problem.

By Proposition 5.2, every (k,ℓ)-tight graph G is $c_{k,\ell}$-connected and thus we augment a $c_{k,\ell}$-connected graph to a $(c_{k,\ell}+1)$-connected graph where $c_{k,\ell}$ is 0, 1 or 2. There exist several methods to deal with these particular problems, even linear time algorithms [14, 30]. However, we also need to augment G to a (k,ℓ)-redundant graph hence we follow simpler ideas mainly from [38].

Let $G=(V,E)$ be a c-connected graph. Let us call a set $X \subset V$ of cardinality c a **min-cut** of G, if $G-X$ is not connected. For a min-cut X of G, let $\boldsymbol{b^c_X(G)}$ denote the number of components of $G-X$. Let $\boldsymbol{b^c(G)}$ denote the maximum value of $b^c_X(G)$ over all min-cuts X of G if there exists any, and let $b^c(G) := 1$ otherwise. Clearly, any edge set F that augments G to a $(c+1)$-connected graph needs to induce a connected graph on the components of $G-X$ for every min-cut X. Thus $|F| \geq b^c(G)-1$. A set $P \subsetneq V$ is called a **$(c+1)$-fragment** of a c-connected graph G which is not $(c+1)$-connected if $N_G(P)$ is a min-cut of G and P induces a connected subgraph of G. Let us denote the maximum number of pairwise disjoint $(c+1)$-fragments by $\boldsymbol{t^c(G)}$. Increasing the connectivity of a c-connected graph G which is not $(c+1)$-connected is equivalent to increasing the number of neighbors of each $(c+1)$-fragment of G. Hence, for any edge set F that augments G to a $(c+1)$-connected graph, $|V(F)| \geq t^c(G)$ must hold. These with Proposition 5.2 imply the following statement.

Lemma 5.3 *Given a (k,ℓ)-rigid graph G. The minimum number of edges that augments G to a $(c_{k,\ell}+1)$-connected graph is at least* $\max\left\{b^{c_{k,\ell}}(G)-1, \left\lceil \frac{t^{c_{k,\ell}}(G)}{2} \right\rceil\right\}$. □

Let us call an inclusion-wise minimal $(c+1)$-fragment a **$(c+1)$-end**. As every $(c+1)$-fragment contains at least one $(c+1)$-end, $t^c(G)$ is equal to the number of pairwise disjoint $(c+1)$-ends. It is easy to see that, for $c=1$, the $(c+1)$-ends are pairwise disjoint. As we will see in the following lemma, this statement is also true for $c=2$, even though in this case the structure is slightly more difficult as there are two types of min-cuts. A min-cut $\{u,v\}$ of a 2- but not 3-connected graph G is called a **weak min-cut** if it separates another min-cut $\{u',v'\}$

of G, that is, u' and v' are in different connected components of $G - \{u, v\}$. Note that in this case the min-cut $\{u', v'\}$ is also weak and $b^2_G(\{u, v\}) = b^2_G(\{u', v'\}) = 2$. If a min-cut is not weak then it is called a **strong min-cut**. When $(k, \ell) = (2, 3)$, the structure of G is much simpler by the following result of Jackson and Jordán [32].

Lemma 5.4 *[32] Let G be a $(2, 3)$-rigid graph. Then G contains no weak min-cuts.* □

Lemma 5.4 immediately implies the following statement when $(k, \ell) = (2, 3)$. However, it holds for general pairs of k and ℓ, as well.

Lemma 5.5 *Let G be a $c_{k,\ell}$-connected graph. Then the $(c_{k,\ell}+1)$-ends of G are pairwise disjoint.*

Proof If G is $(c_{k,\ell} + 1)$-connected, the statement holds obviously. Also, if $k \geq \ell$ thus $c_{k,\ell} \leq 1$, then the $(c_{k,\ell} + 1)$-ends of G are clearly pairwise disjoint.

Now suppose that $k < \ell$, hence $c_{k,\ell} = 2$. Let C_1 and C_2 be two intersecting 3-ends and let $N(C_1) = \{u_1, v_1\}$ and $N(C_2) = \{u_2, v_2\}$ be the two (weak) min-cuts defining C_1 and C_2. We may suppose that $u_1 \in C_2$ and $u_2 \in C_1$. Now we can conclude that $N(C_1 \cap C_2) = \{u_1, u_2\}$ that contradicts the minimality of the 3-ends C_1 and C_2. □

We do not need to understand how we can get then $(c_{k,\ell}+1)$-ends efficiently – yet. However, as they will play an important role in the algorithmic solution, we present the data structures behind them in Subsection 8.4.1.

5.3 The min-max theorem

In this section we shall merge the results on the problem of augmenting a (k, ℓ)-tight hypergraph to a (k, ℓ)-redundant hypergraph and on the $(c_{k,\ell} + 1)$-connectivity augmentation problem to a new min-max theorem for Problem 4 by mixing the statements of Theorem 3.5 and Lemma 5.3, as follows.

Theorem 5.6 *Let $k > 0$ and ℓ be two integers such that $\ell \leq \frac{3}{2}k$. Let $G = (V, E)$ be a (k, ℓ)-rigid graph on at least k^2+2 vertices. Suppose moreover, that G is simple, if $k < \ell$. Let $\mathcal{H}_G = (V, \mathcal{E})$ be the (k, ℓ)-M-component hypergraph of G. If G is $(c_{k,\ell} + 1)$-connected, (k, ℓ)-tight and there is no (k, ℓ)-co-tight set in $\mathcal{H}_G$, then any new edge makes G (k, ℓ)-redundant (and $(c_{k,\ell}+1)$-connected). Otherwise,*

$\min\{|F| : G + F = (V, E \cup F)$ *is* (k, ℓ)*-redundant and* $(c_{k,\ell} + 1)$*-connected*$\} =$
$= \max\Big\{b^{c_{k,\ell}}(G) - 1,\ \max\Big\{\Big\lceil\frac{|\mathcal{A}|}{2}\Big\rceil : \mathcal{A}$ *is a family of disjoint* (k, ℓ)*-co-tight sets of* $\mathcal{H}_G$ *and* $(c_{k,\ell} +$ $1)$*-fragments of* $G\Big\}\Big\}$.

Note that, for a non-tight (k, ℓ)-rigid graph G which is not (k, ℓ)-M-connected, $\mathcal{H}_G$ always has a (k, ℓ)-co-tight set since the vertex set of a hyperedge corresponding to a nontrivial M-component is (k, ℓ)-tight and hence the complement of its vertex set is (k, ℓ)-co-tight. Similarly, $(2, 3)$-tight graphs contain $(2, 3)$-co-tight sets as any edge of G forms a $(2, 3)$-tight subgraph of G. Also, if G is already (k, ℓ)-M-connected and $(c_{k,\ell} + 1)$-connected, then both sides in Theorem 5.6 are 0. Nonetheless, if G is (k, ℓ)-tight for $(k, \ell) \neq (2, 3)$, it can happen that G has no (k, ℓ)-co-tight sets (see in Chapter 3).

We note that $k^2 + 2 > 2k$ always holds.

As mentioned, our main tool to prove Theorem 5.6 for (k, ℓ)-rigid (and not for only (k, ℓ)-tight) inputs is the usage of the M-component hypergraph defined in Chapter 4. If $G + F$ is (k, ℓ)-redundant and $(c_{k,\ell} + 1)$-connected, then Lemma 4.5 can be used to prove that it is (k, ℓ)-M-connected and hence $\mathcal{H}_G + F$ is (k, ℓ)-redundant (by Lemma 4.9) except when $\ell > k$ and $G + F$ has more than $2k - \ell$ parallel edges between two vertices. The following statement implies that this exceptional case can be avoided.

Lemma 5.7 *Let* $k > 0$ *and* ℓ *be two integers such that* $\ell \leq \frac{3}{2}k$*, and let* $G = (V, E)$ *be a* (k, ℓ)*-rigid graph on at least* $k^2 + 2$ *vertices. Then there exists an edge set* F *with* $\min\{|F'| :$ $G + F' = (V, E \cup F')$ *is* (k, ℓ)*-redundant and* $(c_{k,\ell} + 1)$*-connected*$\}$ *edges for which* $G + F$ *is* (k, ℓ)*-redundant,* $(c_{k,\ell} + 1)$*-connected and no edge in* F *is parallel to any edge in* G*.*

Proof Let F be a minimum cardinality edge set for which $G + F$ is (k, ℓ)-redundant, $(c_{k,\ell} + 1)$-connected and amongst such edge sets, F has the minimum number of parallel edges with G. Assume that an edge $e \in F$ is parallel to some edge e' of G. As the omission of e from F does not affect the $(c_{k,\ell} + 1)$-connectivity of $G + F$, we only need to deal with the (k, ℓ)-redundancy of $G + F$.

Let $G' = (V, E')$ be a (k, ℓ)-tight spanning subgraph of G inducing e'. As we saw in Section 3.2 a simple complete graph K_V on V is always (k, ℓ)-redundant if $|V| \geq k^2 + 2$. Hence, by Lemma 2.12, $E' = \bigcup_{f \in K_V - E'} \mathcal{T}_{G'}(f)$, that is, for each edge e_i in E' (in particular, for e') there exists an edge $f \in K_V - E'$ such that $e_i \in \mathcal{T}_{G'}(f)$. Thus $\mathcal{T}_{G'}(e) \subseteq \mathcal{T}_{G'}(f')$ for an edge $f' \in K_V - E'$ by Lemma 2.11. This combined with the fact that $E' = \bigcup_{f \in F \cup (E - E')} \mathcal{T}_{G'}(f)$ by Lemma 2.12

results that $E' = \bigcup_{f \in (F-e)\cup(E-E'-e)\cup f'} \mathcal{T}_{G'}(f)$ also holds, that is, $F' = F - e \cup f'$ is also a minimal edge set for which $G + F'$ is (k,ℓ)-redundant, $(c_{k,\ell}+1)$-connected and has fewer edges parallel to the edges of G than F (since, if f' would be parallel to an edge $e^* \in E - E' - e$, $\mathcal{T}_{G'}(e) \subseteq \mathcal{T}_{G'}(e^*)$ would contradict the minimality of F), a contradiction. Thus F contains no parallel edge to G. $\square$

We start this section by proving Theorem 5.6 for $k < \ell \leq \frac{2}{3}$, because of the importance of $(k,\ell) = (2,3)$ in rigidity theory. Later in this section we sketch how the presented method can be generalized to solve the cases for $\ell \leq k$.

5.3.1 Proof of Theorem 5.6 for $k < \ell \leq \frac{3}{2}k$

For the sake of simplicity, we shall omit the prefix (k,ℓ) from several notions in this subsection such as (k,ℓ)-tight graph or set, (k,ℓ)-co-tight set, (k,ℓ)-MCT set, (k,ℓ)-M-component or (k,ℓ)-redundant and (k,ℓ)-rigid graphs. Note, that if we are talking about hypergraphs, we keep the notions (k,ℓ)-rigid and (k,ℓ)-redundant to make it easier to distinguish. As in this case $c_{k,\ell} = 2$ we may omit it from the superscript of $b^2_X(G)$ and $b^2(G)$. We call the min-cuts, which contain exactly two vertices, cut-pairs.

If G is 3-connected, then Theorem 5.6 follows directly by Theorem 3.5 and Lemmas 4.5, 4.9 and 5.7. For a non-3-connected graph G, the min $\geq$ max implication in Theorem 5.6 is obvious by Lemmas 2.11, 4.5, 4.9, 5.3 and 5.7. To prove the min $\leq$ max part, let us consider the family of all MCT sets of $\mathcal{H}_G$ and 3-ends of G. Let us call the inclusion-wise minimal elements of this family the **atoms** of G. (In Figure 5.1, these are the three sets formed by the highlighted vertices: the big (blue) circles form an MCT set of $\mathcal{H}_G$, the (grey) square vertex forms an MCT set of $\mathcal{H}_G$ which is also a 3-end of G, and the (red) triangle vertices form a 3-end of G.) Let us denote the family of atoms by $\boldsymbol{\mathcal{A}}^*$. We shall show that the atoms are pairwise disjoint and there exists a set of $\max\left\{b(G)-1, \left\lceil \frac{|\mathcal{A}^*|}{2} \right\rceil\right\}$ edges that augments G to a redundant and 3-connected graph. Hence we first need to prove the following.

Lemma 5.8 *Let $G = (V,E)$ be a (k,ℓ)-rigid graph which is not 3-connected where $k < \ell \leq \frac{3}{2}k$. Then the atoms of G are pairwise disjoint.*

To prove Lemma 5.8, we need the following three statements.

Observation 5.9 *Suppose that C is a (k,ℓ)-co-tight set in the (k,ℓ)-tight hypergraph $\mathcal{H}_G = (V,\mathcal{E})$, and $C' \subsetneq C$ such that $d_{\mathcal{H}_G}(C', C - C') = 0$. Then C' is also (k,ℓ)-co-tight.*

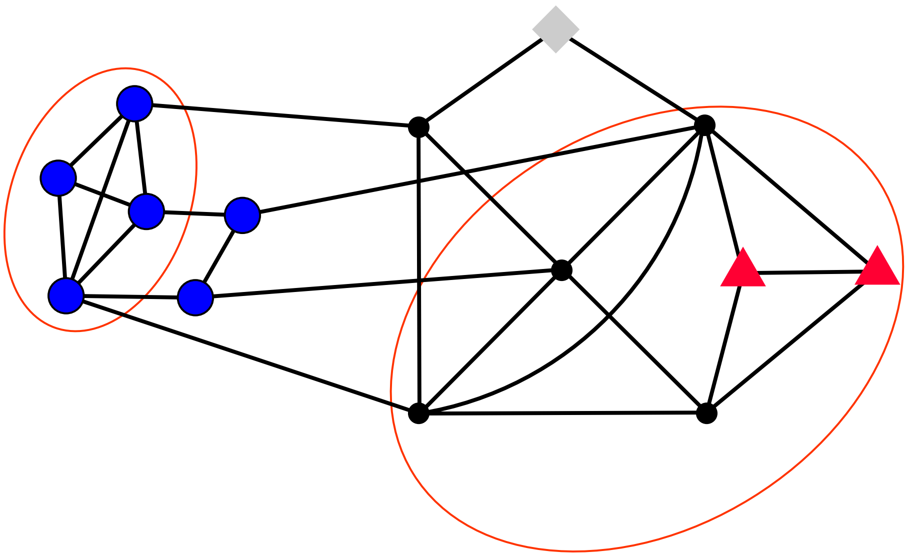


Figure 5.1: A $(2, 3)$-rigid graph with its M-components (encircled). The M-component hypergraph has two MCT sets, one formed by the big (blue) circles and the other formed by the (grey) square. The 3-ends are formed by the (red) triangles and the (grey) square. Adding an edge between the (grey) square and one (red) triangle augments the graph to 3-connected. Adding then one edge between the (grey) square and one (blue) circle augments the graph to redundantly hence globally rigid.

Proof Recall that $d_{\mathcal{H}_G}(C', C - C') = 0$ means that no hyperedge of $\mathcal{H}_G$ has a vertex in both C' and $C - C'$. This implies that $|\widehat{E}(C)| = |\widehat{E}(C')| + |\widehat{E}(C - C')|$. Recall, that in a (k, ℓ)-tight hypergraph $\mathcal{H}$, for a $C \subsetneq V$ $e_{\mathcal{H}}(C) \geq k|C|$ and the two sides equal, if and only if C is a (k, ℓ)-co-tight set. Thus if $|\widehat{E}(C')| \geq k|C'| + 1$, then $|\widehat{E}(C - C')| \leq k|C - C'| - 1$, a contradiction. $\square$

Lemma 5.10 *Let $G = (V, E)$ be a (k, ℓ)-rigid graph which is not 3-connected where $k < \ell \leq \frac{3}{2}k$. If $a \in A \in \mathcal{A}^*$ is a vertex from an atom of G, then there is no $v \in V$ such that a and v form a cut-pair.*

Proof Suppose first that A is a 3-end, and its neighbor set is $N(A) = \{u', v'\}$. If $(k, \ell) = (2, 3)$, the statement follows immediately by Lemma 5.4. Similarly, if $(k, \ell) \neq (2, 3)$ and $\{u', v'\}$ is a strong cut-pair, then the statement follows immediately by Lemma 5.5. However, if $(k, \ell) \neq (2, 3)$, $\{u', v'\}$ may be a weak cut-pair which separates the two vertices of $\{a, v\}$. In this case, a is a cut vertex of $G[A \cup N(A)]$ that separates u' and v'. Moreover, $G[A \cup N(A)] - a$ has exactly two components since otherwise a would be a cut vertex of G (see Figure 5.2 for an illustration). Note that $|A| \geq 2$ must hold since G is a simple (k, ℓ)-rigid graph in which each vertex has a degree of at least k that is at least 3, as $(k, \ell) \neq (2, 3)$ and $\ell > k$. Thus one of the two connected components in $G[A \cup N(A)] - a$, say the component U' containing u' has a cardinality of at least two. Now $N_G(U' - u') = \{u', a\}$, and hence $U' - u' \subsetneq A$ is a 3-fragment of G, contradicting the fact that A is a 3-end. Hence we proved the statement if A is a 3-end.

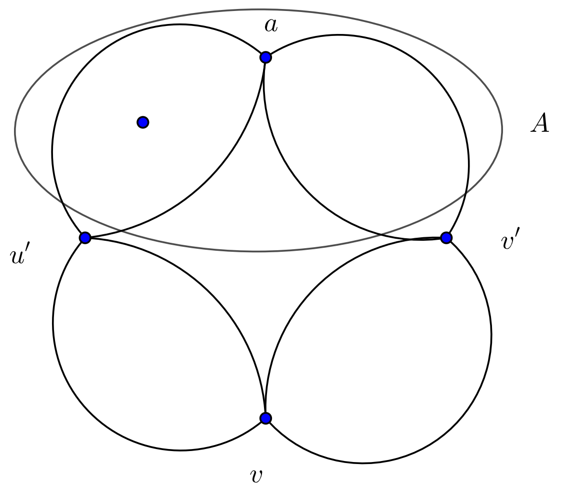


Figure 5.2: If the 3-end A would contain an element a of a cut-pair, then we would obtain a smaller 3-end.

Now let A be an MCT set of $\mathcal{H}_G$. Then $\mathcal{H}_G[V - A]$ is tight and hence Observation 4.7 implies that $G[V - A]$ is rigid. Suppose that a, v forms a cut-pair for $a \in A$ and $v \in V$.

Suppose first that $|V - A| > 2$. Then $G[V - A]$ is 2-connected by Proposition 5.2. Thus $V - A$ intersects only one component of $G - \{a, v\}$, otherwise v would be a cut-vertex in $G[V - A]$. Now $A - a$ contains at least one component of $G - \{a, v\}$ (which contains a 3-end of G), contradicting the minimality of A.

Now assume that $|V - A| \leq 2$. By the minimality of A, it cannot contain any components of $G - \{a, v\}$. Thus $V - A$ consists of two vertices from the two components of $G - \{a, v\}$. However, this contradicts the fact that $\mathcal{H}_G[V - A]$ is tight, because every trivial component of $\mathcal{H}_G$ is also an edge of G. □

Lemma 5.11 *Let $G = (V, E)$ be a (k, ℓ)-rigid graph which is not 3-connected where $k < \ell \leq \frac{3}{2}k$. Let $\mathcal{H}_G = (V, \mathcal{E})$ be its M-component hypergraph. Let C and L be two distinct atoms of G such that C is a (k, ℓ)-MCT set of $\mathcal{H}_G$ and L is a 3-end of G. Then there is no M-component of G which has a vertex set intersecting both $C - L$ and L.*

Proof For the sake of a contradiction, suppose that there exists an M-component of G with vertex set M such that $M \cap L \neq \emptyset$ and $M \cap (C - L) \neq \emptyset$. By Lemma 5.10, $|C \cap N_G(L)| = 0$ thus this M-component cannot be trivial. Consequently, $G[M]$ is M-connected and hence redundant and thus 2-connected. Therefore, $N_G(L) \subset M$. Using Lemma 5.10 again, we conclude that $|\widehat{E}(C-M)| \leq |\widehat{E}(C)| - (k|M| - \ell) = k|C| - (k|M| - \ell) \leq k|C| - (k|C \cap M| + k|N_G(L)| - \ell) < k|C-M|$, where the second inequality comes from $|C \cap N_G(L)| = 0$. As $|C-M| < |C| \leq |V|-2$, $|\widehat{E}(C-M)| < k|C-M|$ is a contradiction by the fact that $|\widehat{E}(X)| \geq k|X|$ holds for each $X \subset V$ with $|X| \leq |V| - 2$. □

Proof of Lemma 5.8 Let $\mathcal{C}^*$ denote the family of MCT sets of $\mathcal{H}_G$ and let $\mathcal{L}^*$ denote the family of 3-ends of G. By Lemma 5.5, the members of $\mathcal{L}^*$ are pairwise disjoint.

Suppose that $C \in \mathcal{C}^* \cap \mathcal{A}^*$ and $L \in \mathcal{L}^* \cap \mathcal{A}^*$. By Lemma 5.11, $d_{\mathcal{H}_G}(C \cap L, C - L) = 0$. Then, by Observation 5.9, either $C \cap L = \emptyset$ or $C \cap L$ is co-tight in $\mathcal{H}_G$ contradicting the minimality of C.

Suppose now that there exist two distinct intersecting sets $C_1, C_2 \in \mathcal{C}^* \cap \mathcal{A}^*$. By Lemma 3.20, $|C_1 \cup C_2| \geq |V| - 1$ contradicting Lemma 5.10 as G is not 3-connected. □

Note that when G is 3-connected, Lemma 5.8 does not always hold (see Theorem 3.10).

Now, we turn to prove that there exists a set of $\max\left\{b(G)-1, \left\lceil \frac{|\mathcal{A}^*|}{2} \right\rceil\right\}$ edges that augments $\mathcal{H}_G$ to a (k,ℓ)-redundant hypergraph (for $k < \ell \leq \frac{3}{2}k$) and G to a 3-connected graph. Recall, that a set X is called a *transversal* of a family $\mathcal{S}$ if $|X \cap S| = 1$ for each $S \in \mathcal{S}$. Let P be a transversal of $\mathcal{A}^*$. As the members of $\mathcal{A}^*$ are pairwise disjoint if G is not 3-connected by Lemma 5.8, choosing one arbitrary vertex from every $A \in \mathcal{A}^*$ obtains a transversal of $\mathcal{A}^*$. Observe that P is a minimum cardinality vertex set that intersects all MCT sets and 3-ends, and consequently all co-tight sets and 3-fragments. Hence, $|\mathcal{A}| \leq |P|$ holds for an arbitrary family $\mathcal{A}$ of disjoint co-tight sets and 3-fragments. We shall show now that a connected graph on P augments G to a 3-connected graph and $\mathcal{H}_G$ to a (k,ℓ)-redundant hypergraph. Later, we will reduce the number of edges needed for this augmentation to the optimum value.

Lemma 5.12 *Suppose that G is a (k,ℓ)-rigid graph which is not 3-connected where $k < \ell \leq \frac{3}{2}k$. Let P be a transversal of $\mathcal{A}^*$, where $\mathcal{A}^*$ is the family of atoms of G. Then, for any connected graph $H = (P, F)$ on P, $G + F$ is 3-connected.*

Proof G is 2-connected by Proposition 5.2. Also, P contains no member of any cut-pair by Lemma 5.10. If there exists a cut-pair in $G+F$, then in one of its components there is no vertex from P, but P intersects all 3-ends and this component is the union of some 3-fragments of G which must contain a 3-end and hence an atom, a contradiction to the choice of P. $\square$

To show that $\mathcal{H}_G$ with a connected graph on P results a (k,ℓ)-redundant hypergraph, we extend the ideas of the proof of Theorem 3.5.

Lemma 5.13 *Let $G = (V, E)$ be a (k,ℓ)-rigid graph which is not 3-connected where $k < \ell \leq \frac{3}{2}k$. Let $\mathcal{H}_G = (V, \mathcal{E})$ be its M-component hypergraph. Let A, B be two atoms such that A is a (k,ℓ)-MCT set of $\mathcal{H}_G$. Then $A \cap N_{\mathcal{H}_G}(B) = \emptyset$.*

Proof Recall that A and B are disjoint by Lemma 5.8. Since G is not 3-connected, $|V-(A\cup B)| \geq 2$ by Lemma 5.10. Thus if both of A and B are MCT sets, then the statement follows by Lemma 3.12.

Suppose that B is a 3-end. By Lemma 5.8 $A - B = A$ hence Lemma 5.11 implies $A \cap N_{\mathcal{H}_G}(B) = \emptyset$. $\square$

Lemma 5.13 and the fact that 3-ends are not connected in G immediately imply the following.

Observation 5.14 *The vertex set P induces no edge in G.* $\square$

Lemma 5.15 *Suppose that G is a (k,ℓ)-rigid graph which is not 3-connected where $k < \ell \leq \frac{3}{2}k$ and $\mathcal{H}_G$ is its M-component hypergraph. Let $\mathcal{A}^*$ be the set of atoms of G and let P be a transversal of $\mathcal{A}^*$. Let F be an edge set of a connected graph on $P' \subseteq P$. Then $R_{\mathcal{H}_G}(F)$ is the minimal tight subhypergraph inducing all elements of P'. In particular, if F is the edge set of a star $K_{1,|P|-1}$ on the vertex set P, then $\mathcal{H}_G + F$ is (k,ℓ)-redundant.*

Proof Recall that $R_{\mathcal{H}_G}(F)$ denotes the set of redundant hyperedges of $\mathcal{H}_G$ in $\mathcal{H}_G + F$ and $R_{\mathcal{H}_G}(F) = \bigcup_{f\in F} \mathcal{T}_{\mathcal{H}_G}(f)$ by Lemma 2.12. We aim to use Lemma 3.15 here for the vertex set P.

As P does intersect every MCT set, we have to prove the second condition that is, for each $p \in P$ there exists a set D_p such that $D_p \subset V(\mathcal{T}_{\mathcal{H}_G}(pq))$ with $|D_p| \geq 2$ for all $q \in P - p$.

Let $A, B \in \mathcal{A}^*$ such that $p \in A$ and $q \in B$. We claim that $D_p := N_G(A)$ is a suitable set. By Proposition 5.2, $|D_p| \geq 2$. If A is an MCT set of $\mathcal{H}_G$, then Lemmas 5.8 and 5.13 imply that $(A \cup N_{\mathcal{H}_G}(A)) \cap B = \emptyset$. Hence, by the definition of $\mathcal{H}_G$ and Lemma 3.14, $A \cup N_G(A) \subseteq A \cup N_{\mathcal{H}_G}(A) \subset V(\mathcal{T}_{\mathcal{H}_G}(pq))$, and thus $D_p \subset V(\mathcal{T}_{\mathcal{H}_G}(pq))$. If A is a 3-end, then each $q \in P - p$ is an element of $V - (A \cup N_G(A))$ by Lemmas 5.8 and 5.10. Now the tightness of $\mathcal{T}_{\mathcal{H}_G}(pq)$ and the definition of $\mathcal{H}_G$ imply that $G[V(\mathcal{T}_{\mathcal{H}_G}(pq))]$ is rigid and hence 2-connected by Proposition 5.2.

Since p and q are from different connected components of $G-N_G(A)$, $D_p = N_G(A) \subset V(\mathcal{T}_{\mathcal{H}_G}(pq))$ follows.

Hence the conditions of Lemma 3.15 hold, which finishes our proof. $\square$

A direct generalization of Lemma 3.17 gives the following. The proof is easy to see by Observation 3.18 hence we omit it.

Lemma 5.16 *Let $G = (V, E)$ be a not 3-connected (k, ℓ)-rigid graph where $k < \ell \leq \frac{3}{2}k$ with M-component hypergraph $\mathcal{H}_G$. Let $\mathcal{A}^*$ be the set of atoms of G and let P be a transversal of $\mathcal{A}^*$. Suppose that $x_1, x_2, x_3, y \in P$ are distinct vertices. Let $\mathcal{T}^* = \mathcal{T}_{\mathcal{H}_G}(x_1y) \cup \mathcal{T}_{\mathcal{H}_G}(x_2y) \cup \mathcal{T}_{\mathcal{H}_G}(x_3y)$. Then $\mathcal{T}^* = \mathcal{T}_{\mathcal{H}_G}(x_1y) \cup \mathcal{T}_{\mathcal{H}_G}(x_2x_3)$ or $\mathcal{T}^* = \mathcal{T}_{\mathcal{H}_G}(x_2y) \cup \mathcal{T}_{\mathcal{H}_G}(x_1x_3)$ holds.* $\square$

Observe that the operation in Lemma 5.16 allows us to reduce the cardinality of the edge set used for the augmentation by maintaining the property that it augments $\mathcal{H}_G$ to a (k, ℓ)-redundant hypergraph (and hence G to a redundant graph by Lemma 4.9). However, we also need to maintain the 3-connectivity of $G + F$ to complete the proof of Theorem 5.6.

Proof of Theorem 5.6 for $k < \ell \leq \frac{3}{2}k$ As we have seen at the beginning of this subsection, we only need to prove the min $\leq$ max part of Theorem 5.6 and only for the case where G is not 3-connected. In this case, the atoms of G (denoted by $\mathcal{A}^*$) are pairwise disjoint by Lemma 5.8 and a tree on a transversal P of $\mathcal{A}^*$ augments G to a redundant and 3-connected graph with $|\mathcal{A}^*|-1$ edges by Lemmas 4.9, 5.12 and 5.15. Note that, as $\mathcal{A}^*$ consists of pairwise disjoint MCT sets of the M-component hypergraph $\mathcal{H}_G$ of G and 3-ends of G, the maximum in Theorem 5.6 is at least $\max\left\{b(G) - 1, \left\lceil \frac{|\mathcal{A}^*|}{2} \right\rceil\right\}$, furthermore, this latter value equals to $|\mathcal{A}^*| - 1$ when $|\mathcal{A}^*| \leq 3$ completing our proof for this case.

To reduce the number of edges needed for the augmentation, we do the following procedure. Let us define a vertex set $N \subseteq P$. The set N stands for "not fixed" vertices while vertices in $P - N$ are the "fixed" vertices. We can **fix** an edge xy by removing x and y from N and adding xy to F.

We shall keep some properties during the whole procedure:

1. For an arbitrary star S_N on the vertex set N, $\mathcal{H}_G + F + S_N$ is a (k, ℓ)-redundant hypergraph.
2. In every 3-end of $G + F$, there is at least one vertex from N.
3. $\max\left\{b(G + F) - 1, \left\lceil \frac{|N|}{2} \right\rceil\right\} + |F| = \max\left\{b(G) - 1, \left\lceil \frac{|P|}{2} \right\rceil\right\}$.

Notice that Properties 1–3 hold for $N = P$ and $F = \emptyset$ by Lemmas 5.12 and 5.15.

Remark 5.17 *Properties 2 and 1 ensure that $G + F + S_N$ is 3-connected and $\mathcal{H}_G + F + S_N$ is (k, ℓ)-redundant and thus $G + F + S_N$ is redundant by Lemma 4.9.*

Remark 5.18 *If $|N| \geq 4$, then from any two edges chosen on $x_1, x_2, x_3 \in N$ one may fix at least one of them (by Lemma 5.16) in such a way that this fixing maintains Property 1.*

Our plan is to find at least two possibilities to fix such that Property 2 is maintained and also, that $\max\left\{b(G + F) - 1, \left\lceil \frac{|N|}{2} \right\rceil\right\}$ decreases by one. Then by Remark 5.18 we can maintain Properties 1–3. This can be realized in the majority of the cases.

Lemma 5.19 *Let G be a not 3-connected (k, ℓ)-rigid graph where $k < \ell \leq \frac{3}{2}k$ with M-component hypergraph $\mathcal{H}_G$. Let $\mathcal{A}^*$ denote the atoms of G. Assume that $|\mathcal{A}^*| \geq 4$. Let P be a transversal on $\mathcal{A}^*$. Let $N \subseteq P$ be a vertex set and F be an edge set on P such that G, N and P satisfy Properties 1–3. If $|N| \geq \max\{4, b(G + F) + 1\}$, then we can choose $x, y \in N$, such that for $N - \{x, y\}$ and $F + \{xy\}$ (that is, for fixing xy) Properties 1–3 also hold.*

Proof We use the following method for the proof. Notice, that this can be turned into a polynomial time algorithm.

If $b(G + F) - 1 \geq \left\lceil \frac{|N|}{2} \right\rceil$, **then**

   If there is only one cut-pair (u, v) such that $b_{(u,v)}(G + F) = b(G + F)$, **then**

      Choose x_1, x_2 from a component of $G + F - \{u, v\}$ that contains at least two vertices from N. Let $x_3 \in N$ be a vertex from a component of $G + F - \{u, v\}$ that does not contain x_1 and x_2.

   else

      Let (u_1, v_1) and (u_2, v_2) be two cut-pairs for which $b_{(u_1,v_1)}(G + F) = b(G + F) = b_{(u_2,v_2)}(G + F)$. Choose $x_1, x_2 \in N$ from two different components of $G + F - \{u_1, v_1\}$ that do not contain $\{u_2, v_2\}$. Choose $x_3 \in N$ from a component of $G + F - \{u_2, v_2\}$ that does not contain $\{u_1, v_1\}$.

else

   If there is a cut-pair $\{u, v\}$ such that for one component of $G - \{u, v\}$, say K, $|N \cap K| \geq 2$ and $|N - K| \geq 2$, **then**

      Choose x'_1, x'_2 from $N \cap K$ and choose x_3 from $N - K$.

If every 3-end of $G + F + x_1'x_3$ contains a vertex from $N - \{x_1', x_3\}$, **then**

Let $x_1 = x_2 := x_1'$

else

Let $x_1 = x_2 := x_2'$.

6 **else** (Notice that if $b(G + F) = 1$, then this is the only possible case.)

Choose $x_1, x_2, x_3 \in N$ arbitrarily.

7 **If** $\mathcal{H}_G + F + S(N - \{x_1, x_3\}) + x_1x_3$ is (k, ℓ)-redundant, **then**

$x := x_1, y := x_3$.

else

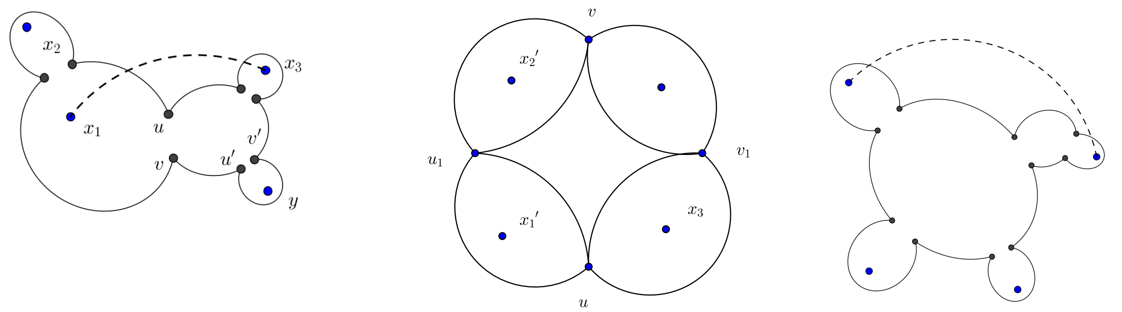
$x := x_2, y := x_3$.

First, we prove that the above method is consistent, that is, we can execute each of its steps. As $|N| \geq b(G + F) + 1$ and P contains no vertex from a cut-pair of G by Lemma 5.10, $|N| > b_{(u,v)}(G + F)$ for an arbitrary cut-pair $\{u, v\}$ hence there exists a component of $G + F - \{u, v\}$ that contains at least two vertices from N. This shows that we can choose vertices in Step **2** consistently. In Step **3** there are at least two components of $G + F - \{u_1, v_1\}$ that do not contain $\{u_2, v_2\}$ since $|N| \geq 4$ and thus $b_{(u_1,v_1)}(G + F) \geq 3$. The consistency of Steps **5** and **6** is obvious.

Now we show that the choice of x and y maintains Properties 1-3. We start with Property 2.

Claim 5.20 *Suppose that there is a strong cut-pair $\{u, v\}$ such that for one component of $G - \{u, v\}$, say K, $x_1, x_2 \in N \cap K$ and $x_3, y \in (V - K) \cap N$. Then fixing either x_1x_3 or x_2x_3 maintains Property 2.*

Proof Notice that the role of x_1 and x_2 is symmetric thus we might suppose that we fixed the edge x_1x_3. Suppose that we form a new 3-end L with it in $G + F$. Then necessarily $x_1, x_3 \in L$. If $x_2 \in L$ or $y \in L$, then Property 2 holds automatically. On the other hand, if none of them is in L, then, as the cut-pair $\{u, v\}$ is strong, there is a cut-pair of G in $K \cup \{u\}$ or in $K \cup \{v\}$ which separates x_1 from x_2 (see Figure 5.3a). There is another cut-pair in $V - K$ (other than $\{u, v\}$) which separates x_3 from y. Both remain cut-pairs after fixing the edge x_1x_3. However, this contradicts the assumption that L is 3-end, as $|N_G(L)| = 2$ must hold for a 3-end. $\square$



(a) Illustration of Claim 5.20. Notice, that we need the existence of the vertex y.

(b) All paths from x_2' to u which avoid v must induce u_1 hence $u \notin A_2'$ in Step **5**.

(c) In the case of Step **6**, we cannot form a new 3-end.

Figure 5.3: Illustration of the proofs of why the algorithm of Lemma 5.19 maintains Property 2.

Notice, that the conditions of this claim hold in Steps **2** and **3** thus with our choice of x_1, x_2, and x_3 Property 2 is maintained. Step **5** is more complicated, as the cut-pair chosen in Step **5** might be weak. If neither the fixing of $x_1'x_3$ nor the fixing of $x_2'x_3$ maintains Property 2, then it means that there is a 3-end with vertex set A_i in $G+F+x_i'x_3$ such that A_i contains no vertex from $N-\{x_i', x_3\}$ for $i=1,2$. Let $N_{G+F}(A_i)=\{u_i,v_i\}$ for $i=1,2$. Now, $N\cap A_i=\{x_i',x_3\}$ and $\{u,v\}$ (chosen in Step **5**) separates $\{u_i,v_i\}$ in $G+F$, as it separates x_i' and x_3 for $i=1,2$. This also means that x_i' is separated from any other vertex of N by, say, $\{u,u_i\}$ or $\{v,u_i\}$ since $K\cup\{u,v\}$ contains either u_i or v_i and this vertex (say, u_i) is a cut vertex in $(G+F)[K\cup\{u,v\}]$. Let us denote the vertex set of the corresponding component of $G-\{u,u_i\}$ or $G-\{v,u_i\}$ that contains only x_i' from N by A_i' for $i=1,2$. Without loss of generality, we may assume that x_1' is separated from any other vertex of N by $\{u,u_1\}$. Now, a similar argument and the existence of the 3-end A_1 in $G+F+x_1'x_3$ implies that x_3 is separated from any other vertex of N by $\{u,v_1\}$. Furthermore, all paths in $G[K\cup\{u,v\}]$ from x_2 to u contain u_1 and hence A_2' cannot contain u since otherwise, it should also contain u_1 and hence, by the connectivity of $G[K]$, all vertices from A_1' (in particular, x_1') contradicting that it contains only x_2' from N (see Figure 5.3b for an illustration). Hence, the existence of the 3-end A_2 in $G+F+x_2'x_3$ implies that x_3 is separated from any other vertex of N by $\{v,v_2\}$. However, in this case, v_1 and all the components of $G[V-K]-v_1$ other than A_2 must be in the component of $G[V-K]-v_2$ containing x_3 and v, and hence it must contain all the vertices in $N-K$, a contradiction. Hence we can conclude that Step **5** also maintains Property 2.

If $G+F$ is already 3-connected, then Property 2 is obvious. Otherwise, in Step **6**, every cut-pair cuts $G+F$ into two components one of which contains exactly one vertex from N

by the condition of Step **5** (see Figure 5.3c). For the sake of a contradiction, assume that $G+F+xy$ contains a 3-end L which contains no element of $N-\{x,y\}$. Let $N_G(L)=\{u,v\}$. Then $N\cap L=\{x,y\}$, $V-L-\{u,v\}\neq\emptyset$, and u,v is a cut pair of $G+F$. By the condition of Step **5**, (u,v) cuts $G+F$ into two components, one of which contains exactly one vertex from N. Hence, exactly L and $V-L-\{u,v\}$ are these two components. Moreover, as $|L\cap N|=2$, this implies $|N\cap(V-L-\{u,v\})|=1$, contradicting $|N|\geq 4$.

Now we show that our method maintains Property 3. Fixing any edge decreases $\left\lceil\frac{|N|}{2}\right\rceil$ by one while increasing F by one. When we chose x_1, x_2 and x_3 in Steps **5** or **6**, this fact is enough to keep Property 3 true as in these cases $\max\left\{b(G+F)-1,\left\lceil\frac{|N|}{2}\right\rceil\right\}>b(G+F)-1$. We need to show that if the condition in Step **1** is true, then we also decrease $b(G+F)$. If $b(G+F)-1\geq\left\lceil\frac{|N|}{2}\right\rceil$, then there can be at most two cut-pairs of $G+F$ satisfying $b_{(u,v)}(G+F)=b(G+F)$ by a simple calculation on the number of 3-ends (see for example [38, Lemma 2.3]). If there is only one, the pair (u,v) chosen in Step **2**, then we only need to decrease $b_{(u,v)}(G+F)$. Since x_1x_3 and x_2x_3 both connect two different components of $G+F-\{u,v\}$, $b_{(u,v)}(G+F)$ decreases by one after fixing any of them. If there are two such cut-pairs, (u_1,v_1) and (u_2,v_2) chosen in Step **3**, then we need to decrease $b_{(u_1,v_1)}(G+F)$ and $b_{(u_2,v_2)}(G+F)$ simultaneously. Again our choice of x_1x_3 and x_2x_3 guarantees this.

Clearly, if both x_1x_3 and x_2x_3 maintain Properties 2 and 3, then by Remark 5.18 one of them also maintains Property 1. We showed that this is the case in every step, except for Step **5**, where x_1 and x_2 are equal.

Claim 5.21 *If $x_1=x_2$ and x_3 is chosen according to Step **5**, then Property 1 is maintained after fixing the edge x_1x_3.*

Proof To see that Property 1 holds, observe that $\{u,v\}$ separates $x_1=x_2$ and x_3 and it also separates the vertices of $N-\{x_1,x_3\}$ by the condition in Step **5**. This implies that the star $S_{N-\{x_1,x_3\}}$ has an edge wz connecting two distinct components of $G-\{u,v\}$. Now $\mathcal{T}_{\mathcal{H}_G}(wz)$ and $\mathcal{T}_{\mathcal{H}_G}(x_1x_3)$ are (k,ℓ)-tight subhypergraphs of $\mathcal{H}_G$ (on at least three vertices) and hence their vertex sets induce (k,ℓ)-rigid subgraphs of G by Observation 4.7, which are 2-connected by Proposition 5.2. This implies that they both contain u and v. Hence Lemma 5.1 implies that $\mathcal{T}_{\mathcal{H}_G}(wz)\cup\mathcal{T}_{\mathcal{H}_G}(x_1x_3)$ is $(k,\ell)-$tight and hence $\mathcal{T}_{\mathcal{H}_G}(wx_1)\subseteq\mathcal{T}_{\mathcal{H}_G}(wz)\cup\mathcal{T}_{\mathcal{H}_G}(x_1x_3)$ by Lemma 2.11. This, with Lemmas 2.12 and 5.15, implies that $R_{\mathcal{H}_G}(S_{N-\{x_1,x_3\}}\cup x_1x_3)=R_{\mathcal{H}_G}(S_{N-\{x_1,x_3\}}\cup\{x_1x_3,wx_1\})=R_{\mathcal{H}_G}(S_N)$ and hence Property 1 remains true. □

Therefore, by Remark 5.18 and Claim 5.21, xy maintains Properties 1–3. This completes the proof of Lemma 5.19. $\square$

We apply Lemma 5.19 recursively until $|N| < \max\{4, b(G+F)+1\}$. To complete the proof of Theorem 5.6 for $k < \ell \leq \frac{3}{2}k$, we need to show the following.

Claim 5.22 *Let F, N be sets, such that they satisfy Properties 1–3 with G. If $2 \leq |N| \leq \max\{3, b(G+F)\}$, then, for an arbitrary star S_N on N, $G+F+S_N$ forms a 3-connected redundant graph for which $|F| + |S_N| = \max\left\{b(G)-1, \left\lceil \frac{|P|}{2} \right\rceil\right\}$.*

Proof $G+F+S_N$ is 3-connected and redundant by Remark 5.17. By Property 3 it is enough to show that $\max\left\{b(G+F)-1, \left\lceil \frac{|N|}{2} \right\rceil\right\} = |S_N| = |N|-1$. If $|N| = b(G+F)$, then $\max\left\{b(G+F)-1, \left\lceil \frac{|N|}{2} \right\rceil\right\} = |N|-1$ as $\left\lceil \frac{|N|}{2} \right\rceil \leq |N|-1$. On the other hand, if $|N| < b(G+F)$, then $2 \leq |N| \leq 3$ thus $\left\lceil \frac{|N|}{2} \right\rceil = |N|-1$. $\square$

Recall that $\mathcal{A}^*$ consists of pairwise disjoint MCT sets and 3-ends of G and hence the maximum in Theorem 5.6 is at least $\max\left\{b(G)-1, \left\lceil \frac{|\mathcal{A}^*|}{2} \right\rceil\right\}$. On the other hand, the above claim implies that G can be augmented to a redundant and 3-connected graph by the addition of an edge set of cardinality $\max\left\{b(G)-1, \left\lceil \frac{|P|}{2} \right\rceil\right\} = \max\left\{b(G)-1, \left\lceil \frac{|\mathcal{A}^*|}{2} \right\rceil\right\}$. This completes the proof of Theorem 5.6 for $k < \ell \leq \frac{3}{2}k$. $\square$

Observation 5.23 *The method in Lemma 5.19 adds edges only between vertices from P. This means that $G+F$ is a simple graph by our assumption on G and Observation 5.14. Thus if $(k,\ell) = (2,3)$, then $G+F$ is globally rigid in $\mathbb{R}^2$ by Theorem 2.3.*

5.3.2 Proof sketch of Theorem 5.6 for $\ell \leq k$

It is easy to see how the results presented in Section 5.1 with some elementary observations can be used to prove Theorem 5.6 in the case where $\ell \leq 0$. (Notice that in this case $c_{k,\ell} = 0$, thus we aim to augment G to a (k,ℓ)-redundant and connected graph.) We leave the details of this rather simple special case to the reader, which enables us to assume in what follows that k and ℓ are positive integers. This simplifies the presentation of the results. Let us now briefly sketch how the proof presented in Subsection 5.3.1 may be transferred to the values of $0 < \ell \leq k$. (We note that similar methods may be used also for the case where $\ell \leq 0$.) In this case, $c_{k,\ell} = 1$ thus we aim to augment G to a 2-connected and (k,ℓ)-redundant graph. This means that each 2-end is separated from G by a cut-vertex and thus cut-pairs in the statements and proofs should be changed to cut-vertices. In fact, all our proofs can be extended (almost) literally hence we only reprove the counterpart of Lemma 5.10 as its statement is slightly modified in this case.

Lemma 5.24 *Let k and ℓ be positive integers with $k \geq \ell$, let $G = (V, E)$ be a (k, ℓ)-rigid graph which is not 2-connected and let $a \in A \in \mathcal{A}^*$ be a vertex from an atom of G. Then a is not a cut-vertex in G.*

Proof If A is a 2-end, then the statement follows immediately by Lemma 5.5.

Now let A be a (k, ℓ)-MCT set of $\mathcal{H}_G$. Then $\mathcal{H}_G[V - A]$ is (k, ℓ)-tight and hence Observation 4.7 implies that $G[V - A]$ is (k, ℓ)-rigid and hence connected. For the sake of contradiction, suppose that $a \in A$ is a cut-vertex of G. This immediately implies that $|A| \geq 2$ and $A - a$ contains at least one component of $G - a$ by Observation 5.9. Thus $G - a$ contains a 2-end of G, contradicting the minimality of A. □

While this proves the correctness of the algorithm, we present a detailed description of the Algorithm in case of $0 < \ell \leq k$ in Section 8.4.3, where we also give an analysis of its running time.

We can compute the (k, ℓ)-M-connected hypergraph of any (k, ℓ)-rigid graph in polynomial time [17, 54, 69] (detailed in Section 8.2), as well, as all the (k, ℓ)-MCT sets of a (k, ℓ)-tight hypergraph (see Chapter 3, detailed analysis in Subsection 8.3.1) and all the 2-ends of a connected graph and 3-ends of a 2-connected graph [14, 30, 38]. Hence it is easy to see that the method presented in the proof of Theorem 5.6 yields a polynomial algorithm for finding the optimal edge set. Its detailed running time analysis is presented in Section 8.4 in Chapter 8.

Chapter 6

Minimal cost globally rigid subgraph

Consider the following motivating questions. Given a globally rigid graph, can we determine a minimum size witness that shows its global rigidity? Or assuming that the edges have some cost, can we determine a minimum cost globally rigid spanning subgraph? These types of questions are common in graph connectivity and they have a broad literature (see for example [31, 64] for some recent results). The motivating questions also fit in this line, as, for example, in $\mathbb{R}^1$ they ask exactly to find a minimum size or cost 2-connected spanning subgraph of a 2-connected graph, which are well-studied problems. It is known that both of these problems are NP-hard [20] and the best known approximation ratio for the minimum size 2-connected subgraph problem is $\frac{4}{3}$ [75], while there exists a 2-approximation algorithm for the minimum cost spanning 2-connected subgraph problem [52]. In case of metric cost functions, this latter approximation ratio can be improved to $\frac{3}{2}$ [20].

This chapter is motivated by the $\mathbb{R}^2$ versions of these globally rigid spanning subgraph problems. One of the applications that inspired this research is the already mentioned localization problem of two-dimensional wireless sensor networks. As presented in the Introduction, in this problem the goal is to compute the locations of all sensors, when only a subset of the pairwise distances and locations is available. Thus Problem 5 may emerge in applications when one wants to achieve, say, global rigidity in $\mathbb{R}^2$ by measuring (or recomputing, fixing, etc.) some pairwise distances in an optimal way. For example, it may happen that (i) certain distances are not computable, or more generally, the cost or time of computing pairwise distances may be different for different pairs, or preferences may be given to some pairs, or (ii) the level of noise in the distance data may be different, or (iii) the total length of the edges is a relevant factor, etc. Applying cost functions on the edges is a common way to encode this data.

Hence we pose the following problem.

Problem 5 *Given a (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected graph $G=(V,E)$ with $|V|>3$ and a cost function on its edges $c:E\to\mathbb{R}$. Suppose that $\ell>0$. Find a spanning subgraph $H=(V,E')$ of G such that H is (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected and the cost of the edge set of H (that is $c(E'):=\sum_{e\in E'} c(e)$) is as small as possible.*

The results in this chapter are loosely based on the results presented in [42]. The main differences are that in that paper we considered the minimum cost globally rigid subgraph problem, which in $\mathbb{R}^2$ coincides with a special case of Problem 5. While this chapter generalizes this result to (k,ℓ)-rigidity, the paper [42] presented results for the minimum cost globally rigid subgraph problem in other dimensions, with completely different techniques. Also, the approach for the proof of Theorem 6.4 completely differs here from the one shown in [42] due to the discoveries we made since submitting that paper.

Problem 5 includes even that version of Problem 4 when we are restricted to choosing the edges of F from a prescribed set. (We note that this version was not considered in Chapter 5, as little is known about it.) This can be seen, as we may choose E to be the union of the original graph edges and the edges that we can use for the augmentation. We can then set the cost of the original graph edges to 0, so that solving Problem 5 gives a solution to Problem 4 on the restricted edge set.

In contrast to Problem 4 though, we know that Problem 5 is NP-hard. It is quite easy to see that it contains the Hamiltonian cycle problem in case of $(k,\ell)=(1,1)$ and $c\equiv 1$. Also, if $(k,\ell)=(2,3)$, it is NP-hard to solve it optimally, as we shall see in Section 6.3. Hence, we restrict our attention to two special cases of Problem 5.

6.1 Minimal size globally rigid spanning subgraph

In the first case we suppose that $(k,\ell)=(2,3)$, that is, H is $(2,3)$-redundant, and 3-connected (or equivalently, globally rigid in $\mathbb{R}^2$ [32]). Moreover, we assume that $c\equiv 1$. In this case, there exists a $\frac{3}{2}$-approximation algorithm for the solution of Problem 5. Its core is the following observation.

Lemma 6.1 *[42] Suppose that $G=(V,E)$ is minimally globally rigid in $\mathbb{R}^2$ with $|V|\geq 4$. Then $|E|\leq 3|V|-6$.* $\square$

As any globally rigid graph in $\mathbb{R}^2$ is redundantly rigid [32], it is clear that any globally rigid graph H contains at least $2|V|-2$ edges. Hence, deleting edges from G until we get a minimally globally rigid graph indeed gives a $\frac{3}{2}$-approximation result. We shall analyse its running time in detail in Chapter 8 in Section 8.6.

6.2 Minimal cost globally rigid spanning subgraph

In the second special case let us assume that the graph G is the complete graph K_V on the vertex set V, moreover, c is a metric cost function, that is, c satisfies the triangle inequality, $c(uv) + c(vw) \geq c(uw)$ for any three $u, v, w \in V$. We note here that from the metric condition it follows obviously that c is non-negative. As we rely heavily on the results of Chapter 5 in this section we suppose that $\ell \leq \frac{3}{2}k$ and $k > 0$.

We shall use the following algorithm to give a feasible solution to Problem 5 with these constraints.

Algorithm 6.2 *Input: A complete graph on at least 4 vertices, $K_V = (V, E)$ along with a metric cost function $c : E \to \mathbb{R}$ and $\ell \leq \frac{3}{2}k$.*

Output: A spanning subgraph of K_V that is (k, ℓ)-redundant and $(c_{k,\ell} + 1)$-connected.

1. *Compute a minimum cost (k, ℓ)-rigid spanning subgraph $T = (V, E')$ of K_V.*

2. If *T is $(c_{k,\ell} + 1)$-connected and its (k, ℓ)-M-connected hypergraph has at most 2 disjoint (k, ℓ)-MCT sets,* ***then***

Find two vertices $\{u, v\}$, so that $\{u, v\}$ intersects every (k, ℓ)-MCT sets.

Output *$G' = T + uv$.*

3. else

Let the family of atoms be $\mathcal{A}^*$

Find a transversal of $\mathcal{A}^*$, let it be P.

4. *Compute a minimum cost spanning tree (P, F) on P.*

Output *$G' = (V, E' + F)$.*

Notice that in Step 1 we first choose every edge with cost 0 (even if they are not independent), and then we add edges that are independent of the already chosen ones in the (k, ℓ)-sparsity matroid. It can be done in polynomial time in a greedy way. This method is similar to the one presented in Section 8.2, the difference being the cost of each edge. We will describe the method used here in detail in Section 8.6. Observe, that if each cost is positive, then T is (k,ℓ)-tight.

Lemma 6.3 *The output of Algorithm 6.2 is (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected.*

Proof If Algorithm 6.2 terminates at Step 2, then $G+uv$ is $(c_{k,\ell}+1)$-connected, and by Theorem 3.10 $G+uv$ is also (k,ℓ)-redundant.

Suppose now that Algorithm 6.2 terminates at Step 4. If $k \le \ell$, it is easy to see that G' is 2-connected, as we eliminated every cut-vertex. If $k \le \ell \le \frac{3}{2}k$, then we can apply Lemma 5.12 to show that G' is 3-connected. Now in either case Lemma 3.15 together with Observation 5.14 prove that G' is indeed (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected. □

The output that Algorithm 6.2 gives is not necessarily optimal. In fact, giving an optimal solution to Problem 5 is NP-hard even with these restrictions (as shown in Section 6.3). Nonetheless, we can conclude that Algorithm 6.2 gives a near-optimal solution.

Theorem 6.4 *Consider an instance of Problem 5, where G is the complete graph K_V on at least 4 vertices, and for the cost function c we know that $c(uv)+c(vw) \ge c(uw)$ for any three $u,v,w \in V$. Suppose that $0 < \ell \le \frac{3}{2}k$ are integers. Then the output of Algorithm 6.2 gives a feasible solution for Problem 5 with the cost at most 3-times the optimum. Moreover, if $\ell = k$ or $\ell = k+1$, then the cost of this solution is at most $1+\frac{2}{k}$ times the optimum.*

Proof Lemma 6.3 shows that G' (that is, the output of Algorithm 6.2) is indeed (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected, hence feasible.

To verify the approximation ratio consider an optimal solution G^*. Let OPT denote the total cost of the edges of G^*. Since G^* is (k,ℓ)-rigid, while $T=(V,E')$ is minimum cost (k,ℓ)-rigid, we have $c(E') \le OPT$.

Claim 6.5 *If $k=\ell$ or $k+1=\ell$, then G^* contains k edge-disjoint spanning trees.*

Proof First let us prove that G^* is (k,k)-rigid.

If $k=\ell$, then G^* is obviously (k,k)-rigid. If $k+1=\ell$, then since G^* is $(k,k+1)$-redundant, there exists a $(k,k+1)$-tight spanning subgraph H^* of G^*-e for any fixed edge e of G^*. However, it is easy to observe that H^*+e is (k,k)-sparse (since H^* is $(k,k+1)$-sparse). By the number of edges H^*+e contains, it must be (k,k)-tight. Hence again, G^* is (k,k)-rigid.

If G^* is (k,k)-rigid, then by the famous result of Nash-Williams [62], G^* contains k edge-disjoint spanning trees. $\square$

Let t denote the number of disjoint spanning trees G^* contains. As G^* is connected by Proposition 5.2 (since $\ell > 0$ thus $c_{k,\ell} \geq 1$), $t \geq 1$.

Suppose first that the output is obtained in Step 2. G^* contains at least t edge-disjoint uv-paths hence there is a uv-path S with $c(S) \leq \frac{OPT}{t}$. By using that c is metric, we get $c(uv) \leq c(S) \leq \frac{OPT}{t}$ and hence $c(E'+uv) \leq OPT + \frac{OPT}{t}$. If $t = k$, it results $(1+\frac{1}{k})OPT$, while in any case, it is at most $2OPT$, which satisfies the claim of the Theorem.

Next, suppose that the output is obtained in Step 4. Since G^* contains t edge-disjoint spanning trees, a minimum cost spanning tree F^* of K_V satisfies $c(F^*) \leq \frac{OPT}{t}$. Furthermore, it is well-known that if c is metric and $P \subseteq V$ then the cost of a minimum cost spanning tree in $K[P]$ has cost at most $2c(F^*)$. This follows by doubling the edges of F^* to get an Eulerian graph J and then short-cutting an Eulerian walk of J to obtain a spanning cycle C on P. Since c is metric and C contains a spanning tree of $K[P]$, it follows that the minimum cost spanning tree on P has cost at most $2c(F^*)$. Hence, in this case, $c(G') \leq (1+\frac{2}{t})OPT$, which is either $3OPT$ or $(1+\frac{2}{k})OPT$ depending on the value of t. $\square$

It is easy to observe that the running time of Algorithm 6.2 is polynomial. Its exact analysis is detailed in Chapter 8.

Let us focus on the special case $(k,\ell) = (2,3)$ now. By Theorem 2.3 we know that this is the minimum cost globally rigid spanning subgraph problem in $\mathbb{R}^2$.

Observation 6.6 *Given a metric cost function on a complete graph we can find a spanning globally rigid subgraph in $\mathbb{R}^2$, with its cost being at most 2-times the optimal solution.*

We have a family of instances showing that the approximation ratio of Algorithm 6.2 is not better than $\frac{3}{2}$ in case of $(2,3)$. Consider a complete graph K on $2s+1$ vertices, for some integer $s \geq 2$. Fix a subset W of vertices of size s and define the costs of the edges of K so that the edges between vertices in W are of cost 2 while the cost of every other edge is equal to 1. The algorithm may find, as the minimum cost spanning tight subgraph, a graph in which each vertex in W is an MCT set on its own. See Figure 6.1 for the case $s = 5$. The minimum cost tree on these vertices has a total cost of $2s-2$. Thus the output has cost $4s-1+2s-2 = 6s-3$. On the other hand, it is not hard to see that a feasible solution of cost $4s$ exists.

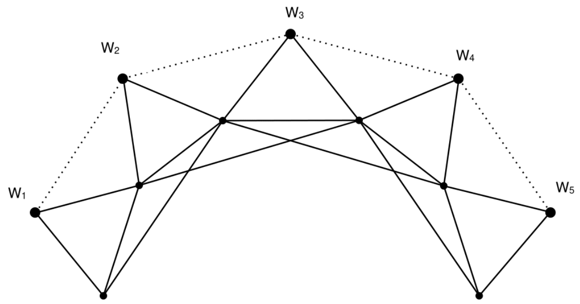


Figure 6.1: The solid edges correspond to the spanning tight subgraph. The dotted edges form a tree on the transversal of the atoms.

6.2.1 Conjecture of approximation ratio

We suspect that Algorithm 6.2 gives a 1.5-approximation in case $(k, \ell) = (2, 3)$ instead of a 2-approximation. We have computational evidence to support this claim. We have conducted an experiment in which we tested Algorithm 6.2 on 40000 randomly generated instances of size 10 to 50 vertices. We determined the optimal cost of the globally rigid spanning subgraphs by using integer programming tools, and the worst ratio that Algorithm 6.2 gave was ~1.44. However, we didn't manage to prove this ratio. This work was done in collaboration with Péter Madarasi, who developed the integer programming tools for the experiment.

6.2.2 Euclidean cost functions

In the Euclidean version of our problems the vertices correspond to points in $\mathbb{R}^2$ and the cost of an edge is the Euclidean distance of its endpoints. In this version, which may occur for example in the network localization problem for $(k, \ell) = (2, 3)$, Algorithm 6.2 has a better approximation ratio.

In order to show this, recall that in the Euclidean Steiner Tree Problem we are given a set P of points in the plane and the goal is to find a tree of minimum total length, which contains P. The tree may use points not in P. The ratio of the total length of a shortest spanning tree on P and the total length of a shortest Steiner tree with respect to P is the so-called *Steiner ratio*. It was proved in [9] that the Steiner ratio is at most 1.22.

We can use this fact in the analysis of our algorithm and deduce that $c(F) \leq 1.22c(F^*) \leq \frac{1.22}{t}OPT$, following the notation of Theorem 6.4. Thus the approximation ratio of the Euclidean version is at most $(1 + \frac{1.22}{t})OPT$, that is, 1.61 in the $(k, \ell) = (2, 3)$ case.

6.3 Complexity results

In this section we show that the problems considered in this chapter are NP-hard. Since global rigidity is equivalent to 2-connectivity in $\mathbb{R}^1$, finding a smallest globally rigid spanning subgraph of a graph G on the line is more general than the Hamiltonian cycle problem. Hence, Problem 5 is NP-hard for $(k, \ell) = (1, 1)$.

We shall introduce here another really common operation in rigidity theory. The **cone** of a graph G is obtained from G by adding a new vertex v and new edges from v to all vertices of G. See Figure 6.2 for an illustration. Connelly and Whiteley [12] proved that a graph G is globally rigid in $\mathbb{R}^d$ if and only if the cone of G is globally rigid in $\mathbb{R}^{d+1}$. We can apply it to globally rigid graphs in $\mathbb{R}^2$.

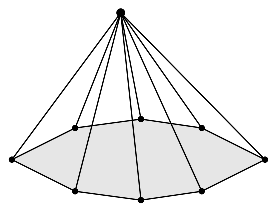


Figure 6.2: The cone graph of a graph.

Theorem 6.7 *It is NP-hard to find a minimum cost globally rigid spanning subgraph in $\mathbb{R}^2$ of a given complete graph $G = (V, E)$ with respect to a metric cost function $c : E \to \mathbb{R}$. This shows that it is NP-hard to solve Problem 5.*

Proof We shall reduce the Hamiltonian cycle problem to our problem. Consider an instance $H = (V, E)$ of the Hamiltonian cycle problem. Let G be the cone of H, where the new vertex is denoted by v, and let K be the complete graph on vertex set $V \cup \{v\}$. We assign costs to the edges of K as follows.

For every edge $e = uv$ with $u, v \in V$ we let $c(e) = 1.1$ (resp. $c(e) = 1.9$) if $uv \in E$ (resp. if $uv \notin E$). For the remaining edges e of K, which are incident with v, we define $c(e) = 1$. We claim that H has a Hamiltonian cycle if and only if the minimum cost globally rigid spanning subgraph of K, with respect to c, has a total cost of $2.1|V|$.

To see this, first, suppose that H has a Hamiltonian cycle C. It is easy to see that the cone graph of C is globally rigid in $\mathbb{R}^2$. The total cost of the cone is $1.1|V| + |V| = 2.1|V|$. Next, suppose that there is a globally rigid spanning subgraph F of K with a cost at most $2.1|V|$. Since every globally rigid subgraph of K has at least $2|V|$ edges, the definition of c implies that F has exactly $2|V|$ edges and that it contains every edge incident with v.

Thus F is the cone graph of a 2-connected spanning subgraph C of H with $|V|$ edges. This shows that C is a Hamiltonian cycle in H. This completes the proof. $\square$

Chapter 7

Additional results

In this chapter, we present some extra topics that are closely related to the results of the previous chapters, particularly that of Chapters 3 and 5. These are mostly applications and consequences of the results already presented in this dissertation. Some of the results of this chapter are mentioned in [50] and [51], while others are unpublished. We finish this chapter with some open questions that hopefully motivate further research.

7.1 Pinning problems

Let us first consider a problem already mentioned in the introduction, which is also closely related to the augmentation problems. This is the so-called **pinning problem** from rigidity theory. In the pinning problem, the goal is to anchor a minimal set of joints in a bar-joint framework such that the resulting framework becomes rigid/redundantly rigid/globally rigid. It can be particularly useful in case of localization problems, as anchoring/localizing any point also provides valuable information on the location of the whole framework.

In the generic case, pinning can be modelled by adding a complete graph on the anchored vertices to our input (see [39]). This input may be a graph or a hypergraph. We can talk about pinning down a graph to a rigid/redundantly rigid/globally rigid graph and a hypergraph to a (k, ℓ)-rigid/(k, ℓ)-redundant hypergraph. Some of these problems have been investigated beforehand, especially pinning problems in $\mathbb{R}^2$ (that are, the $(k, \ell) = (2, 3)$ versions on graph inputs). See such examples in [15, 16, 39, 79]. In this section, we also focus most of our attention on the $(k, \ell) = (2, 3)$ special cases.

Determining the minimum number of vertices to pin down in a graph, so that it becomes $(2,3)$-rigid, is possible in polynomial time. Fekete showed a polynomial algorithm and a min-max theorem to this question [15]. It can be reduced to a maximum matching problem in an auxiliary graph, as we shall see it in Subsection 7.1.1.

Jordán [40] obtained the optimal $(2,3)$-redundant pinning of a $(2,3)$-tight graph by using [40, Theorem 3.9.13.] (which is a special case of Theorem 3.11 for $(2,3)$-tight graphs). The main idea behind Jordán's work is similar to Observation 3.7: it can be seen that each co-tight set must contain a pinned vertex.

Fekete and Jordán investigated the global rigidity version of the pinning problem, where they looked for a minimum size vertex set that pins down G to a globally rigid graph. They gave a $\frac{5}{2}$-approximation algorithm to this problem [16]. Their approach was somewhat similar to ours from Chapter 5, they pinned down G to an M-connected and later to a 3-connected graph. They used the matroid matching problem on hypergraphic matroids to approach the M-connectivity pinning problem. We also mention that they gave a randomized algorithm that results a 2-approximation solution to this problem. Both of these methods are based on approximating a solution to the matroid parity problem for a hypergraphic matroid for which they used a $\frac{3}{2}$-approximation as a polynomial time deterministic algorithm, and an optimal randomized algorithm, respectively. With an optimal polynomial time solution to this particular problem, their method would yield a 2-approximation for the global rigidity pinning problem. Nonetheless, no exact polynomial algorithm is known yet to solve the matroid parity problem for hypergraphic matroids.

In this chapter, with the help of the results of Chapters 3 and 5, we match their approximation ratio of 2 with an exact polynomial time algorithm on general inputs. Moreover, in this section we give an optimal solution, if the input graph is rigid. It is also worth noting that our algorithm is significantly simpler than the aforementioned ones. Also, our ideas can be applied to more general versions of the problem, for general (k,ℓ) pairs with $0<\ell\leq\frac{3}{2}k$, not just $(k,\ell)=(2,3)$.

We collect the algorithmic results of the pinning problems in Section 8.5.

7.1.1 Rigidity pinning problem

We shall start with understanding the algorithm of Fekete for pinning down G to a rigid graph, as presented in [15]. In this particular – so-called *rigidity pinning* – problem we are given a non-rigid graph $G=(V,E)$ and we look for a minimum cardinality vertex set P, so that $G+K_P$

is rigid. Notice that $|P| > 1$ always holds. We note that the rigidity pinning problem will be one of our tools used in other pinning problems (mainly in Subsection 7.2.3).

First, suppose that the input graph G is sparse. In this case, Fekete proved that the vertex set P with at least 2 vertices pins down G to a rigid graph if and only if $2|X| \leq e_G(X)$ for all $X \in V - P$ [15]. Hence we aim to find a set X for which for any $Y \subseteq X$ we have $2|Y| \leq e_G(Y)$. This may be formulated as a matching problem in an auxiliary graph, as follows [15].

Consider the bipartite graph $B(G) = (E, V^*, E^*)$ that we get by assigning one vertex for each edge from E and two vertices v_1, v_2 in V^* for every vertex $v \in V$, and connecting a pair e, v_i by an edge $ev_i \in E^*$, if and only if e is incident with v in G. Observe, that the condition $2|X| \leq e_G(X)$ on the set $X \subseteq V$ is equivalent to Hall's condition for perfect matching in the bipartite graph $B(G)$. Hence, for any P that pins G down to a rigid graph, the set $\{v_i | v \in V - P\}$ can be matched in $B(G)$. Now let us extend $B(G)$ by adding an edge v_1v_2 for each $v \in V$. Let $B^*(G)$ denote this extended graph (which is not bipartite anymore). Fekete showed that the minimal pinning set can be formed by the set of vertices P, such that $v \in P$ if and only if the maximum size matching in $B^*(G)$ does not match both v_1 and v_2 to members of E. It can be shown that these vertices indeed form a pinning set, and Fekete proved that no smaller size pinning set is possible [15].

Now if $G = (V, E)$ is not sparse, we can follow the approach of [39, Lemma 3.2.]. Let us take a maximal sparse spanning subgraph $G^* = (V, E^*)$ of G. We can find a minimal pinning set P of G^*, as shown before. We claim that P is indeed a minimal set that pins G down to a rigid graph. For the sake of contradiction, assume that we have a smaller pinning set P' that pins down G to a rigid graph, but obviously does not pin G^* to a rigid graph, as $|P|$ was the optimum for that. Hence there is an edge $e \in E + E(K(P')) - (E^* + E(K(P'))) = E - E^*$ for which $E^* + e$ is independent (because $E^* + E(K(P'))$ can be extended to a rigid graph only using edges from $E + E(K(P')) - (E^* + E(K(P')))$ by the matroid property). This contradicts the maximality of G^*. Hence P is the optimal pinning set for G, as well.

7.1.2 Redundant rigidity pinning

In this version of the pinning problem we are given a hypergraph $\mathcal{H}$. We need to choose a minimum cardinality vertex set P, so that $\mathcal{H} + K_P$ is (k, ℓ)-redundant, where K_P denotes the complete graph on the vertices of P.

If $\mathcal{H}$ is (k, ℓ)-tight, then P must intersect every co-tight set by Observation 3.7. However, by

Theorem 3.9 either there are two vertices covering all co-tight sets or the inclusion-wise minimal co-tight (MCT) sets are disjoint. By this and Lemma 3.15 choosing one vertex from each MCT set gives an optimal solution to the redundant pinning problem (if $\mathcal{H}$ is (k, ℓ)-tight).

If $\mathcal{H}$ is (k, ℓ)-rigid, it is NP-hard to give an optimal solution to the redundant rigidity pinning problem, as it can be deduced from the construction introduced in Section 3.4. For the same reason, it is NP-hard even to give any constant factor approximation to this problem.

7.1.3 Global rigidity pinning

More can be said about the *global rigidity pinning problem*. In this version of the pinning problem we are given a graph G and the goal is to find a minimum cardinality vertex set P, so that $G + K_P$ is $(2, 3)$-redundant and 3-connected (that is, globally rigid in $\mathbb{R}^2$).

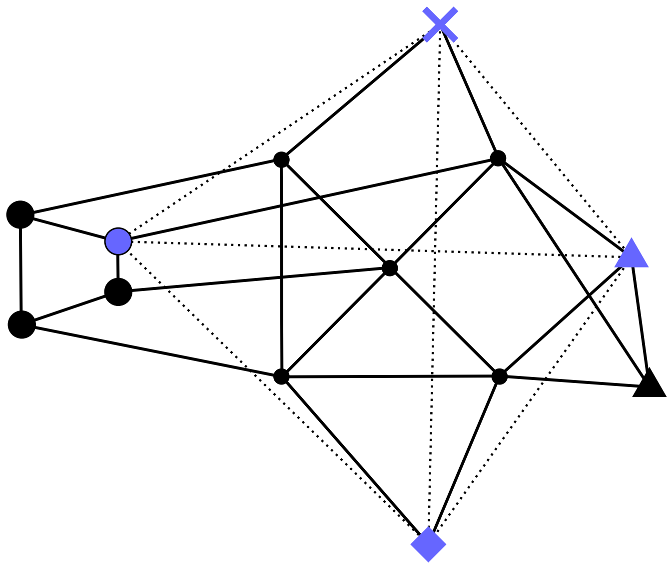


Figure 7.1: Pinning down a rigid graph in $\mathbb{R}^2$ (solid edges) to a globally rigid graph. This graph cannot be pinned down to a globally rigid graph with less than four vertices, as it has 4 atoms (represented by different shaped vertices). On the other hand, adding a complete graph (dashed edges) on the chosen vertices (light blue colored) makes the graph globally rigid in $\mathbb{R}^2$.

Let $G = (V, E)$ be a rigid (but not globally rigid) graph that we want to pin down to a globally rigid graph. If G is 3-connected, then the problem becomes pinning down $\mathcal{H}_G$ to $(2, 3)$-redundant, which can be solved optimally as presented in Subsection 7.1.2. Hence we may assume that G is not 3-connected. Now, it is clear that each 3-end of G needs to be pinned down to eliminate its cut-pairs. On the other hand, each MCT set of $\mathcal{H}_G$ needs to be pinned down by Lemmas 2.11, 4.5 and 4.9. However, by Lemma 5.8 all the atoms of G are pairwise disjoint.

Hence we must pin down at least one vertex from each atom of G. By Lemmas 5.12 and 5.15 pinning down exactly one vertex from each atom results a globally rigid graph and thus this is an optimal pinning.

Thus we can conclude this in the following new theorem:

Theorem 7.1 *Let $G = (V, E)$ be a $(2,3)$-rigid simple graph on at least 6 vertices. Let $\mathcal{H}_G$ denote the M-component hypergraph of G. Suppose that G is either not 3-connected, or G is 3-connected and there are at least three pairwise disjoint $(2,3)$-MCT sets in $\mathcal{H}_G$. Then*

$$\min\{|P| : G + K_P \text{ is globally rigid in } \mathbb{R}^2\} =$$
$$= \max\{|\mathcal{A}| : \mathcal{A} \text{ is a family of disjoint } (2,3)\text{-co-tight sets of } \mathcal{H}_G \text{ and } 3\text{-fragments of } G\}.$$

If G is 3-connected, and there are at most two pairwise disjoint $(2,3)$-MCT sets in $\mathcal{H}_G$, then G can be pinned down optimally to globally rigid with 2 vertices.

Notice here that – as we discussed previously – we always need to pin down at least two vertices to pin down a G to a globally rigid graph. If we consider the physical reality of localization and pinning, we could even assume three vertices to be necessary. This assumption of at least three vertices was used in [15, 39].

Nonetheless, there is a scenario, when a graph might be pinned down to globally rigid using only one vertex. Let us consider here the globally rigid pinning of *partially pinned graphs*. The origin of this scenario is that in some applications, some vertices may already be pinned down. This can be modelled by a graph $G = (V, E)$ and a set $\emptyset \neq V' \subseteq V$ of the already pinned vertices. We look for a set $P \subseteq V - V'$ of minimum cardinality for which $G \cup K_{P \cup V'}$ is globally rigid. Note that $|P| = 1$ might be a feasible solution this case. If $G \cup K_{V'}$ is rigid and not 3-connected, then this problem can be solved optimally, since we only need to pin down those atoms of $G \cup K_{V'}$ that do not contain any vertex from V'. If $G \cup K_{V'}$ is rigid and 3-connected, we only need to pin down vertices from those MCT sets of $\mathcal{H}_{G \cup K_{V'}}$ that do not contain any vertex from V'. We note that this observation is true, but less useful when $G \cup K_{V'}$ is 3-connected and $\mathcal{H}_{G \cup K_{V'}}$ can be made redundant with the help of only one edge. We need to handle this case separately by checking every vertex, if pinning of that results in a globally rigid graph. See the algorithm in detail in Section 8.5.

7.1.4 Cost of pinning

Sometimes pinning down some vertices might cost more than others. In such a case we aim to pin down the vertex set with minimum cost. We assume a non-negative cost function $c : V \to \mathbb{R}_+$ on the vertices of G.

Let us first consider the minimum cost rigidity pinning problem in $\mathbb{R}^2$. As we concluded in Subsection 7.1.1, the rigidity pinning problem can be reduced to a matching problem on an auxiliary graph, where the vertex-pairs of which not both v_1 and v_2 are matched to vertices from E form the pinning set. This observation can be also used, if we look for the minimum cost vertex set to pin down. In this case, we give the cost $c(v)$ to every edge incident to v_1 and v_2, except for the edge between v_1 and v_2 that has cost 0. Now we look for a maximum cost matching in $B^*(G)$. The same vertices will still form a pinning set, that is, a minimum cost pinning set, as the cost function is non-negative. Hence we can find a minimum cost set that pins down G to a rigid graph in $\mathbb{R}^2$.

Now, let us investigate the problem of pinning down a tight hypergraph $\mathcal{H}$ to a redundant hypergraph with minimum cost. To approach this problem we must assume that there is no single edge ij, so that $\mathcal{H}+ij$ is redundant. With this assumption – as we saw earlier – every MCT set of $\mathcal{H}$ must be pinned down to pin down $\mathcal{H}$ to a redundant hypergraph (see Subsection 7.1.2). As in this case the MCT sets are pairwise disjoint (by Theorem 3.9), pinning down the cheapest vertex from each MCT set gives us a minimum cost redundant pinning of $\mathcal{H}$

Similarly, if we consider the global rigidity pinning problem on a rigid graph G, we have to assume that either G is not 3-connected, or if G is 3-connected, then there is no single edge that augments $\mathcal{H}_G$ to a redundant hypergraph. Then all of the atoms of G needs to be pinned down (see in Subsection 7.1.3). Thus we can conclude that the optimal solutions in this case can be reached even with cost involved by simply choosing the cheapest vertex from the atoms, or MCT sets of $\mathcal{H}_G$, respectively. We present an efficient algorithm solving it in Section 8.5. (We note here, that in case of $(k, \ell) = (2, 3)$, Jordán showed a polynomial time $\frac{5}{2}$-approximation algorithm for the minimum cost global rigidity pinning problem, regardless of the input [39]. His solution combined with our ideas presented in this subsection can give a $\frac{3}{2}$-approximation algorithm when the input is rigid in $\mathbb{R}^2$, 3-connected and there exists an edge that makes $\mathcal{H}_G$ redundant.)

7.2 Non-rigid inputs

Sometimes the conditions of rigidity or tightness do not hold for the input. Meanwhile, solutions of the augmentation problems that occurred in Chapters 3 and 5 and the problems of Section 7.1 all require these types of conditions. In this section we investigate what can we state about these problems with inputs that do not satisfy rigidity or tightness conditions.

For the sake of simplicity, we work with graphs in this section. We note, nonetheless, that our approaches generalize well to hypergraphs without much difficulty.

The best-understood problem of this kind is to add a minimum cardinality edge set to a graph G to make G (k, ℓ)-rigid. By the matroid property of (k, ℓ)-rigidity, presented in Section 2.2, we know that this problem can be solved by first detecting a maximal (k, ℓ)-sparse subgraph of G and then greedily adding new edges to this subgraph until it becomes (k, ℓ)-tight hence making G (k, ℓ)-rigid. Such a solution is described in several papers, including [7, 36, 54, 69]. Also, the algorithm we present in Section 8.1 can be modified to solve this problem in $O(|V|^2)$ running time.

7.2.1 The redundant augmentation problem on non-rigid graphs

Let us consider Problem 1 on graphs that are not (k, ℓ)-rigid. As we saw in Chapter 3, there are differences in how to approach this problem depending on $k < \ell$ or not.

If $k < \ell < 2k$, then even giving a constant factor approximation is NP-hard by Theorem 3.38. Hence we shall restrict our investigation to the following version of the problem: Given a (k, ℓ)-*sparse* graph G, find an edge set F, such that $G + F$ is (k, ℓ)-redundant and $|F|$ is of minimum cardinality. First notice that $G + F$ is (k, ℓ)-rigid. Hence for a minimum cardinality edge set F_1 for which $G + F_1$ is (k, ℓ)-rigid, $|F_1| \leq |F|$. As mentioned before and presented in Section 8.1, we can find F_1, so that $G + F_1$ is (k, ℓ)-tight (remember, G is (k, ℓ)-sparse). Given $G + F_1$, we can use Theorem 3.1 and the results of Chapter 3 to find an optimal edge set F_2, so that $G + F_1 + F_2$ is (k, ℓ)-redundant. On the other hand, $G + F + F_1$ is also (k, ℓ)-redundant resulting $|F_2| \leq |F|$. Therefore $|F_1| + |F_2| \leq 2|F|$, which proves that we can give a 2-approximation algorithm to Problem 1 for (k, ℓ)-sparse graphs.

If $\ell < k$ though, we do not need the restriction on the input. In this case, we can prove a similar result to the previous one for any non-(k, ℓ)-rigid graph, with a fundamentally identical approach. Given a graph G, we can add a minimum cardinality edge set F_1, so that $G + F_1$ is

(k,ℓ)-rigid, as seen before. Now we can execute the reduction procedure presented in Section 3.1 to get an (m,ℓ)-tight graph G' from $G+F_1$. We can now add an optimal edge set F_2 to G', so that $G'+F_2$ is (m,ℓ)-redundant hence $G+F_1+F_2$ is (k,ℓ)-redundant by Proposition 3.4. Similarly to the case of $k<\ell$, we can conclude that $|F_1|\leq|F|$ and $|F_2|\leq|F|$ for an optimal edge set F. Hence this method gives a 2-approximation for Problem 1 for any input, if $\ell\leq k$.

Finding an optimal solution to this problem could be really interesting. Hence solving Problem 1 for (k,ℓ)-sparse inputs will appear in the Open Problems section, in Subsection 7.6.1.

7.2.2 Globally rigid augmentation problem on non-rigid graphs

Let us consider Problem 4 for non-rigid inputs, as well. We note that an approximation solution with an approximation factor of 10 was already known for this problem if $(k,\ell)=(2,3)$ [39]. We improve this factor significantly to approximation ratio 2 with a method similar to that of presented in Subsection 7.2.1. First, we find a minimum cardinality edge set F_1, such that $G'=G+(V,F_1)$ is a (k,ℓ)-rigid graph (which can be simple if $k<\ell$, see [17, 54]). Then, using the method presented in Chapter 5 (detailed in the algorithms of Section 8.4) we augment G' to a (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected graph with a new edge set F_2. Any edge set F that augments G to a (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected graph must also augment G to a (k,ℓ)-rigid graph. Thus $|F_1|\leq|F|$ holds. On the other hand, if $G+F$ is (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected, then $G+F_1+F$ is obviously also (k,ℓ)-redundant and $(c_{k,\ell}+1)$-connected, therefore $|F_2|\leq|F|$. Thus $|F_1|+|F_2|\leq 2|F|$ proves the approximation ratio of this solution to be 2.

7.2.3 Pinning problems on non-rigid graphs

As we saw in Subsection 7.1.1, we can pin down non-rigid graphs to a rigid graph optimally. Let us now focus on other pinning problems with non-rigid inputs.

Globally rigid pinning problem If G is not rigid, then the method we used in Subsection 7.1.3 for the global rigidity pinning problem cannot be applied directly. However, we can follow the ideas of the approximation algorithms presented in Subsections 7.2.1 and 7.2.2 for the augmentation problems of non-rigid graphs. We can pin down G optimally to a rigid graph by the method presented in Subsection 7.1.1. Let P_1 denote such an optimal vertex set. Now, $G+K_P$ is already rigid, hence we may use our method to pin it down to globally rigid optimally with another vertex set P_2. Now $G + K_{P_1} + K_{P_2}$ is globally rigid resulting $G + K_{P_1 \cup P_2}$ being globally rigid, as well. Hence the set $P_1 \cup P_2$ pins down G to a globally rigid graph. This results a 2-approximation solution for the minimum cardinality pinning set. This is because an optimal pinning set P pins down G to a rigid graph, hence $|P_1| \leq |P|$. On the other hand, as $G + K_{P_1} + K_P$ is globally rigid, $|P_2| \leq |P|$ by similar reasoning to the augmentation version. This proves that $|P_1 \cup P_2| \leq 2|P|$.

Partially pinned graphs Let us consider the settings of partially pinned graphs. We aim to pin down the partially pinned graph G with the already pinned set V' to a globally rigid graph. Suppose that $G \cup K_{V'}$ is not rigid. Similarly to the previous result, we can give a 2-approximation algorithm to pin down $G \cup K_{V'}$ to a rigid graph. This is because we can modify the algorithm of Subsection 7.1.1 in such a way that we delete all edges of v'_i for $v' \in V'$ (that is, the already pinned vertices), except for the one between v'_1 and v'_2. Now looking for the maximum size matching in this graph, the vertices that do not match perfectly to E form a pining set, which is minimum size amongst the ones that include V'. Hence in such a way we get a minimum cardinality set $P_1 \subseteq V - V'$ for which $G \cup K_{P_1 \cup V'}$ is rigid. Now we can pin down $G \cup K_{P_1 \cup V'}$ to a globally rigid graph optimally, as presented in Subsection 7.1.3. This will result a 2-approximation with a similar analysis to the previous cases.

7.3 Characterization of tight graphs

As we saw in Section 2.1 the structures of tight and rigid graphs (in $\mathbb{R}^2$) are often studied and well-understood. Moreover, as we present it in Section 8.1, there exist effective algorithms to find these, or add edges so that a sparse graph becomes tight [7, 36, 54, 55]. Nevertheless, in this section, we add a new characterization of $(2,3)$-tight graphs. This characterization may later lead to a method that could improve the approximation ratios of 2, presented in Section 7.2. In this whole section, we restrict ourselves to $(2,3)$-rigidity, and we use the notions accordingly.

As we saw in Chapter 3, co-tight sets play a key role in the rigidity augmentation problem. However, some properties of the co-tight sets are useful to characterize tight graphs, as well.

Lemma 7.2 *Let $G = (V, E)$ be a $(2,3)$-sparse graph on at least 3 vertices. Then G is $(2,3)$-tight if and only if for any $C \subset V$ for which $|C| \leq |V| - 2$, $e_G(C) \geq 2|C|$ holds.*

We need to notice that the conditions in Lemma 7.2 are difficult to check, therefore they cannot be used directly to decide the tightness of a graph. Similar (but not identical) observations play key roles in the works of Fekete [15] and Fekete and Jordán [16], as we saw in Section 7.1.

Proof If G is tight, then we know that for any C for which $|C| \leq |V| - 2$, $e_G(C) \geq 2|C|$ with $e_G(C) = 2|C|$ holding for co-tight sets by the sparsity of $V - C$.

For the other direction suppose that G is not tight, so that there exists an edge e for which $G + e$ is still sparse. Now let F denote a (possibly empty) edge set, so that $G + e + F$ is tight, which exists by the matroid nature of rigidity. Let us consider any co-tight set in $G + F + e$, say C, for which $e \in \widehat{E}_{G+F+e}(C)$. There exists such a co-tight set, as any edge in G forms a tight set on its own, and there are edges that are not parallel with e by $|V| \geq 3$. Now $e_{G+F+e}(C) = 2|C|$ as C is co-tight, while by our condition $e_G(C) \geq 2|C|$, a contradiction. $\square$

One may notice, that Lemma 7.2 can be generalized easily to every (k, ℓ) pair, where we can ensure that there exists at least one (k, ℓ)-co-tight set which e intersects, as that is the only part where we used that we are restricted to $(2, 3)$-rigidity. This is true for example for every (k, ℓ) pair where $\ell = 2k - 1$.

7.4 Simple graphs

In several applications in rigidity theory (see [35, 63]), we deal with graphs and not hypergraphs, moreover, it must be assumed that all considered graphs are simple. Hence, only those augmentations can be accepted that maintain this property, that is, the input, as well as the output graph is simple.

Let us now consider the redundant augmentation problem (Problem 1) on simple graphs, with the extra requirement that $G + F$ is simple. We even consider here a slight generalization of this problem that we call the $\boldsymbol{E'}$**-free general problem**: *Given an (m, ℓ)-rigid graph $G = (V, E)$ and $E' \subseteq E$, find a graph $H = (V, F)$ on the same vertex set with minimum number of edges, such that $G \cup H$ is (m, ℓ)-redundant and the edges in F are not parallel to the edges in E'.* We call the version where the input is (m, ℓ)-tight the $\boldsymbol{E'}$**-free reduced problem**.

First, let us notice that an optimal augmenting edge set never contains two parallel edges by Lemma 2.12. It is easy to see that our method presented in Chapter 3 solves this problem for $(k, 2k - 1)$-tight inputs (when a solution exists), since in this case two parallel edges form a circuit of the sparsity matroid. For general (m, ℓ), the problem is still solvable, as follows. By using the same reduction as in Section 3.1, we can reduce the solution of the E'-free general problem to the solution of the E'-free reduced problem when $m \geq \ell$. Hence we only need to solve the E'-free reduced problem.

First, assume that the optimal solution is a single edge uv, that is, $\mathcal{T}(uv) = G$ for a pair $u, v \in V$. If $uv \notin E'$, then uv is an optimal solution of the E'-free reduced problem. Hence we may assume that $uv \in E' \subseteq E$. Let E^* denote the edges not in E' that we may use. If $G + E^*$ is not (m, ℓ)-redundant, then the E'-free reduced problem is not solvable. Thus suppose that $G + E^*$ is (m, ℓ)-redundant, that is, $\mathcal{T}(uv) = G = R(E^*) = \bigcup_{e \in E^*} \mathcal{T}(e)$. Then, as $uv \in E$, $uv \in \mathcal{T}(e)$ for an edge $e \in E^*$. Hence, $G = \mathcal{T}(uv) \subseteq \mathcal{T}(e) = G$ by Lemma 2.11.

Next, assume that the output consists of a set F of at least two edges and the conditions of (A) (introduced in Section 3.2) hold for G. Then no edge in F is parallel to any edge $e \in E$ by Lemma 3.13 since the algorithm outputs edges connecting members of $\mathcal{C}^*$ in this case.

Finally, assume that the output consists of a set F of at least two edges and (A) does not hold for G. If (A2) or (A3) does not hold for G, then G has at most c^2+2 vertices by Section 3.2, and hence the solution of the E'-free problem is straightforward by checking all possibilities. When G violates (A1), we may delete its isolated vertices with $m(v) = 0$ as done in Section 3.2. If (A) holds for the resulting graph then the output of our algorithm does not use any edge parallel to any edge $e \in E$ by Lemma 3.13 and we are done. Otherwise, the resulting graph again has at most c^2+2 vertices. Here we need to be careful since it is possible that the E'-free augmentation is not solvable for the arising graph, while it is solvable for G (for example by using some edges uv from a deleted vertex with $m(u) = 0$). However, observe that the sets of edges in $\mathcal{T}(u_1v)$ and $\mathcal{T}(u_2v)$ are the same for $u_1, u_2, v \in V$ with $m(u_1) = m(u_2) = d(u_1) = d(u_2) = 0$, moreover, the edge set of $\mathcal{T}(u_1u_2)$ is a subset of both. Hence, in this case, we may add back a single vertex u with $m(u) = d(u) = 0$ and solve the E'-free reduced problem on the arising graph on at most $c^2 + 3$ vertices by checking all possibilities.

7.5 Adding multiple edges

In some applications, we cannot add just one edge to augment some properties, instead, we need to add several parallel copies of each edge. One such problem arises when one wants to augment a 'rigid generic body-hinge framework' to 'globally rigid' by adding some hinges (see details in [41]). This can be modelled by a graph G, where we want to minimize $|F|$, such that

$G + cF$ is (k, k)-redundant.

Note that, if G is a (k, ℓ)-tight graph and F is a set of edges on the same vertex set, then $R(cF) = \bigcup_{f \in cF} \mathcal{T}(f) = \bigcup_{f \in F} \mathcal{T}(f) = R(F)$ by Lemma 2.12. Hence, for any (k, ℓ)-rigid graph G, $G + cF$ is (k, ℓ)-redundant if and only if $G + F$ is (k, ℓ)-redundant. Therefore, for Problem 1 it makes no difference if we use one edge or c parallel copies for augmentation hence Theorem 3.5 also provides a solution to this parallel copied version.

7.6 Open problems

Combinatorial rigidity is a dynamically developing field, there are countless open problems one can investigate. In this section we collect some of the most interesting ones in order to inspire further research on these topics. Some of them seem to be quite approachable and promising to solve while others resisted our attempts for a long time. All of these problems are open to the best of our knowledge and we encourage the reader to try to solve them.

7.6.1 Redundant augmentation of (k, ℓ)-sparse graphs

As we saw in Chapter 3, the (k, ℓ)-redundant augmentation problem (Problem 1) is well-understood, if the input is (k, ℓ)-tight. In [51] we showed an $O(|V|^2)$ solution to it, while in Section 8.3 we give an $O(|V|^2)$ time solution for a special case of it. Meanwhile, the same problem is NP-hard without any constant factor approximation, if the input is (k, ℓ)-rigid and $k < \ell$. Hence the natural question arises: what can we say if the input is (k, ℓ)-sparse?

Let k and ℓ be integers with $k > 0$ and $\ell < 2k$ and let $G = (V, E)$ be a (k, ℓ)-sparse graph. Find a graph $H = (V, F)$ on the same vertex set with minimum number of edges, such that $G \cup H = (V, E \cup F)$ is (k, ℓ)-redundant.

None of the papers talking about (k, ℓ)-redundant augmentations ([22, 44, 51]) considers this case specifically. We don't know if this problem is NP-hard or it can be solved optimally in polynomial time. The closest we get is the polynomial time 2-approximation solution presented in Section 7.2.1. As the method of proof for the NP-hardness in case of (k, ℓ)-rigid inputs – shown in Section 3.4 – proved that finding a constant factor approximation for (k, ℓ)-rigid inputs is NP-hard (Theorem 3.38), the existence of constant factor approximation hints that there might be a polynomial time algorithm solving this problem. Maybe the characterizations of tight graphs presented in Section 7.3 could help with it.

7.6.2 Globally rigid augmentation of general graphs

A similar question arises with the global rigidity augmentation problem. While Theorem 5.6 showed that it is possible to solve Problem 4 for (k, ℓ)-rigid graphs, if $\ell \leq \frac{3}{2}k$, for general graphs we could not give any better than a 2-approximation solution in Subsection 7.2.2. Hence, the following question remains open.

Let k and ℓ be integers with $k > 0$ and $0 < \ell \leq \frac{3}{2}k$ and let $G = (V, E)$ be a graph. Find a graph $H = (V, F)$ on the same vertex set with minimum number of edges, such that $G \cup H = (V, E \cup F)$ is (k, ℓ)-redundant and $(c_{k,\ell} + 1)$-connected.

For $(k, \ell) = (2, 3)$ this question is especially interesting, as this replicates Problem 3 in two dimensions.

We strongly believe that the key to solving this problem is to solve it for (k, ℓ)-sparse graphs first. This is because with the (k, ℓ)-M-component hypergraph any graph can be translated to a (k, ℓ)-sparse hypergraph, for which techniques developed for graphs probably can be transferred. To this end solving the problem from Subsection 7.6.1 might provide useful insights.

We also note that for this problem, in contrast to the redundant augmentation problem, there exists no complexity result proving its NP-hardness for certain classes of graphs.

7.6.3 Minimum cost globally rigid subgraph problem

Let us repeat Problem 5 here.

Given a graph $G = (V, E)$ with $|V| > 3$ and a cost function on its edges $c : E \to \mathbb{R}$. Suppose that $\ell > 0$. Find a spanning subgraph $H = (V, E')$, such that $H \subseteq G$ spans V, H is (k, ℓ)-redundant and $(c_{k,\ell} + 1)$-connected and the cost of the edge set of H (which is $c(E')$) is as small as possible.

As we discussed in the introduction of Chapter 6, Problem 5 is NP-hard. Also, it incorporates the problem presented in Subsection 7.6.2. Hence for this open problem, we do not try to solve Problem 5 optimally, instead, we are interested in a polynomial time constant factor approximation algorithm for it.

We note that Problem 5 is a generalization of several well-known problems for which constant factor approximations are known. Such problems are the travelling salesman problem (TSP) or weighted 2-vertex connectivity augmentation from a restricted edge set. As the approximation algorithms for these problems require different approaches, we aim to determine special cases for which Problem 5 can be solved within constant factor.

Also, we are extremely interested in determining the exact approximation ratio of our algorithm from Section 6.2 for metric cost function and complete graph, especially if $(k, \ell) = (2, 3)$. Our conjecture is stated in Subsection 6.2.1.

7.6.4 Augmentation with complete graphs of constant size

In all of the problems presented in this dissertation we aimed to add a minimum size edge set. The only exception is the pinning problems, which we might consider as adding an arbitrary sized complete graph. As a common generalization of these ideas, one might ask the same problems with

adding the edge set of minimum number of complete graphs to G, so that we assume that all of these complete graphs are on at most h vertices.

These problems seem solvable with the tools presented in this dissertation. They might have applications with sensor localization problems, when, say, we could switch on or off the pairwise distance measurements for a given type of sensors – albeit, only together.

7.6.5 Co-tight sets in higher dimensional rigidity

The co-tight sets have their roots in 2-dimensional rigidity, in fact, their original name was co-rigid sets. While they generalize to (k, ℓ) from $(2, 3)$ really well and they lead us to the results presented in this dissertation, it might be interesting to check if they can play an analogous role for augmentation problems regarding higher dimensional rigidity. While, for example, the rigidity in $\mathbb{R}^3$ cannot be described with count-matroids, the edges form a matroid in that case, nonetheless.

Given a minimally rigid graph $G = (V, E)$ in d dimensions ($d \geq 3$), find a minimum cardinality edge set F so that $G + F$ is redundantly rigid in $\mathbb{R}^d$. Do co-rigid sets play any role in the solution of this problem?

We are not aware of any similar result to Theorem 3.9 for minimal co-rigid sets in a minimally rigid graph in $\mathbb{R}^3$. Considering the central role of Theorem 3.9 in the proof of Theorem 3.1, it would be natural to start the investigation with its generalization.

7.6.6 Common framework for rigidity and connectivity

We saw that connectivity has several connections to rigidity/count-matroids (for example in Theorems 2.3 and 2.6 or in Section 5.2, not to mention the 1-dimensional rigidity and global rigidity). Moreover, these connections seem to have really deep roots.

We mention some reasons that support this idea. First, we could unify the structure of the (k, ℓ)-MCT-s with the $(c_{k,\ell}+1)$-ends resulting in the atoms in Chapter 5. Moreover, the similarities between Lemma 5.5 and Theorem 3.9 seem to strengthen this bond even further. Once we have a transversal of the (k, ℓ)-MCT sets or the $(c_{k,\ell}+1)$-ends, one can observe the similarities between the methods of augmenting rigidity and augmenting connectivity with these vertices, respectively. Moreover, deeper investigations reveal further similarities between Lemma 2.9 and [37, Lemma 1.2] – they are basically the same, just the meaning of “tight” sets changes.

Nonetheless, we could not come up with a unified structure that would handle both of these theories thus we needed to prove everything for both. It would be really fascinating if one could show such a bond between these two fields of combinatorics. Also, this could promote the use of ideas from connectivity theory in the field of combinatorial rigidity.

7.6.7 $O(|V|^2)$ algorithm for minimum cost (k, ℓ)-tight spanning subgraph

Finding the minimum cost (k, ℓ)-tight subgraph of G is quite simple in $O(|V|^3)$ time, as we shall present it in Chapter 8. However, doing so in $O(|V|^2)$ time seems to be open for $k < \ell$.

Let $G = (V, E)$ be a (k, ℓ)-rigid graph and $c : E \to \mathbb{R}$ a cost function on the edge set of G. Give an $O(|V|^2)$ running time algorithm that finds a minimum cost (k, ℓ)-tight spanning subgraph of G.

We note that it is claimed in [55] that the data structure presented there can handle this problem in $O(|V|^2)$ time. However, the proof of this statement seems to be incomplete.

Let us go here into details of the problem with the proof presented in [55]. The specific point of interest is the proof of the running time for $union(E_i, E_j)$, updating the matrix *VM* and the *bounded property*. The problem arises when $k < \ell$ and the algorithm needs the *union* of more than two components. Such a case is certainly possible, for example if $(k, \ell) = (2, 3)$, G is a circle of length four, and we add a chord to G thus making G rigid and merging all the previous components. As there is no *union* operation for more than two sets, we need to use *union* sequentially in order to merge the components into one large component. However, during these steps the *bounded property* does not necessarily hold for the intermediate edge sets if $k < \ell$. As a generalization of our previous example, let us say that $(k, \ell) = (2, 3)$ and there are four components $C_1, \ldots, C_4$ so that $|C_1 \cap C_2| = |C_2 \cap C_3| = |C_3 \cap C_4| = |C_4 \cap C_1| = 1$. Adding one edge between $C_1 \cap C_2$ and $C_3 \cap C_4$ may merge all these components into one large component. However, applying $C_1' =$*union*(C_1, C_2) and then $C_3' =$ *union*(C_3, C_4) can cause an intermediate state, when the next step in the algorithm is *union*(C_1', C_3'), and the *bounded property* does not hold for C_1' and C_3'. Hence, when we refresh the instances in *VM* we might need to refresh instances that are already 1 thus disrupting the running time analysis.

In other words, it is not sufficient to show the running time only for *union*(E_i, E_j), it is also necessary to introduce a new *union*$(E_1, E_2, \ldots, E_k)$ operation with the required running time analysis.

Hence this question still seems to be open, the solution of which would be interesting for theory and application alike.

Chapter 8

Algorithmic aspects

In this chapter, we show efficient algorithms for the problems presented in Chapters 3, 4, 5 and 6 together with a well-known algorithm to find a maximal (k,ℓ)-sparse subgraph of a graph. As we already mentioned, several algorithmic results from this dissertation already achieve polynomial running time. Most of these algorithms can be implemented quite easily, however, we shall improve upon the näive running times of multiple problems with in-depth analysis.

This chapter is based on the algorithms and analysis from [49]. Here we shall mention that in [51, Section 6] we presented an $O(|V|^2)$ running time algorithm to make an (m,ℓ)-tight graph (m,ℓ)-redundant that we do not incorporate in this chapter. The reason is that while its core ideas are also present in Section 8.3, its implementation is more complicated. Moreover, the algorithm from [51] does not contain any extra information useful for the results of Sections 8.4 and 8.6. Any reader, who is interested specifically in the (m,ℓ)-redundant augmentation algorithms for graphs or hypergraphs is referred to [51].

8.1 Testing (k,ℓ)-sparsity for graphs

Let us first briefly summarize the algorithm for testing the (k,ℓ)-rigidity of a graph (see [7, 36, 54] for more details). When we speak about the running time of an algorithm containing (k,ℓ)-sparsity, throughout this chapter we assume that k and ℓ are fixed. Hence they can be considered constants in the running time analysis.

Several results could be generalized to (m,ℓ)-sparsity, as well, for which we need to assume the (*) property, introduced in Subsection 3.2. As a reminder, the property is as follows:

(*) There exists a (universal) constant $c > 0$ such that $m(v) \leq c$ for every $v \in V$ for every hypergraph $\mathcal{H} = (V, \mathcal{E})$. We may also suppose that $|\ell| \leq c$.

We shall also suppose that the input graph has at most $O(|V|^2)$ edges. The latter assumption is natural since if two vertices of the input are connected by more than $2k - \ell$ (or $2c - \ell$) edges, then every edge above this limit is redundant. The algorithm of this section is based on the Orientation Lemma of Hakimi [26] and uses in-degree constrained orientations.

Lemma 8.1 (Orientation lemma [26]) *Let $G = (V, E)$ be a graph and let $m : V \to \mathbb{Z}_+$. Then G has an orientation such that the in-degree of each vertex v is at most $m(v)$ if and only if $i(X) \leq \sum_{v \in X} m(v)$ holds for each $X \subseteq V$.*

To check the (k,ℓ)-rigidity of G, the algorithm constructs a maximal sparse subgraph of G by considering its edges one by one. Throughout the construction, we maintain a sparse subgraph of G, denoted by G' and we also keep an orientation $\vec{G}'$ of G' in which the in-degree of each vertex v is at most k. Before adding an edge ij to G', we try to find a reorientation of G' in which the sum of in-degrees of i and j is at most $2k - \ell - 1$, while the in-degree of any other vertex $v \in V - \{i, j\}$ is at most k. Such reorientation (if exists) can be found in $O(|V|k) = O(|V|)$ time by running backwards DFSs on $\vec{G}'$ from i and j and switching the orientation of paths ending at i or j (remember, we consider k as a constant). If we find an eligible reorientation, then, by Lemma 8.1, each set X containing both i and j induces at most $k|X| - \ell - 1$ edges in G' and hence adding the edge ij to G' maintains its sparsity. Otherwise, the edge ij can be omitted, moreover, (during the backwards DFS) we find a minimal set containing both i and j with in-degree zero in $\vec{G}'$. This is the minimal set X which contains i and j and induces $k|X| - \ell$ edges in G'. Hence X is the vertex set of the minimal (k,ℓ)-tight subgraph of G' containing both i and j.

Lee and Streinu [54] reduced the running time of the algorithm above from $O(|V||E|)$ to $O(|V|^2)$. The main idea is to maintain the family of *rigid components* of G'. These are the rigid subgraphs of G. With the help of the rigid components, one can omit those edges in constant time, which are not needed to construct a maximal (k,ℓ)-sparse subgraph of G. One can update the structure of rigid components in $O(|V|)$ time, when an edge is added. Besides this update, we need a data structure, which helps to decide in constant time whether two vertices are in the same component. To get an easily updatable data structure that fulfills this goal, we follow the ideas of [55] to take the edges in a *breadth-first* manner, that is, in such a way that we take all the edges incident with the same vertex $v \in V$ in a row. As long as we take these edges, we maintain a $0-1$ vector of length $|V|-1$, which is one at a coordinate corresponding to a vertex $w \in V - v$ if and only if v and w are induced by a (k, ℓ)-tight subgraph of G'. (Hence v and w belong to the same rigid component.) When we add a new edge to G' or start to investigate another vertex in the breadth-first search, this vector can be updated in $O(|V|)$ time. This along with the constant time omission of edges already in the same rigid component implies a total running time of $O(|V|^2)$ and space complexity of $O(|V|)$ for finding the maximal (k, ℓ)-sparse subgraph of a graph G. This algorithm obviously can be used for testing the (k, ℓ)-rigidity of a graph, as well.

This presented algorithm also works for testing the (k, ℓ)-sparsity and (k, ℓ)-tightness of a hypergraph [69]. This is due to the fact that Lemma 8.1 can be generalized to directed hyperedges of directed hypergraphs. However, its running time is slightly worse, since the size of the hyperedges affects the running time of the backwards DFS subroutines. Fortunately, the (k, ℓ)-M-component hypergraph has $k|e| - \ell$ parallel copies of each hyperedge e which implies that the running time in this special case does not increase.

This algorithm can be generalized to find an (m, ℓ)-tight subgraph of an (m, ℓ)-rigid graph G with little extra effort. The running time, in this case is, a little more complicated, as a general (m, ℓ)-tight graph may not have $O(|V|)$ edges, hence finding a reorientation also can be slower. However, by assuming (*) the running time of finding a reorientation reduces to $O(2c|V|) = O(|V|)$. We note that in most of the applications of (m, ℓ)-tight graphs (that originated from (k, ℓ)-tight graphs) (*) holds automatically.

8.2 Algorithmic construction of the M-component hypergraph

By the definition and the properties of the (k, ℓ)-M-component hypergraph given in Chapter 4 and by the algorithm presented in Section 8.1, it can be concluded that the (k, ℓ)-M-component hypergraph of G can be constructed in polynomial time. Berg showed in his PhD dissertation that for $(k, \ell) = (2, 3)$ this running time can be as good as $O(|V|^2)$ [6]. Here we approach the problem in a slightly different manner and generalize it to every (k, ℓ) pair, where $0 < \ell < 2k$. Our algorithm constructs the (k, ℓ)-M-component hypergraph of a graph G in $O(|V|^2)$ running time. As an additional value, this solution expands the data structures used and it is consistent with the later use of the (k, ℓ)-M-component hypergraphs. We also implemented this algorithm in C++. The details of the implementation are presented in the Appendix.

For building the (k,ℓ)-M-component hypergraph, we shall use similar techniques to the ones sketched in Section 8.1. During the procedure, we collect the already used edges in graph G'. We maintain the (k,ℓ)-M-component hypergraph $\mathcal{H}_{G'} = (V, \mathcal{E}')$ of G', and its orientation $\vec{\mathcal{H}}_{G'}$ in which the in-degree of each vertex is at most k. We note that we consider directed hypergraphs with hyperedges having one head and multiple tails. For technical reasons and to obtain some extra data in our output, we will also maintain a maximal (k,ℓ)-sparse subgraph $G^* = (V, E^*)$ of G', however, this data is not needed for the algorithm. At the beginning of our procedure G', G^*, $\mathcal{H}_{G'}$ and $\vec{\mathcal{H}}_{G'}$ are all empty.

To achieve the $O(|V|^2)$ running time, we need to deploy a rather complex data structure, illustrated by Figure 8.1. Besides storing each edge of $\mathcal{H}_{G'}$ corresponding to a trivial M-component of G', we store just one instance of every (undirected) hyperedge of $\mathcal{H}_{G'}$ that corresponds to a nontrivial M-component of G'. We store the directed hyperedges on the same vertex set as pointers to the underlying undirected hyperedge and a pointer to their head. For every vertex v, we keep a doubly linked list of pointers to the directed hyperedges of which v is the head of. We call it the **head-list** of v.

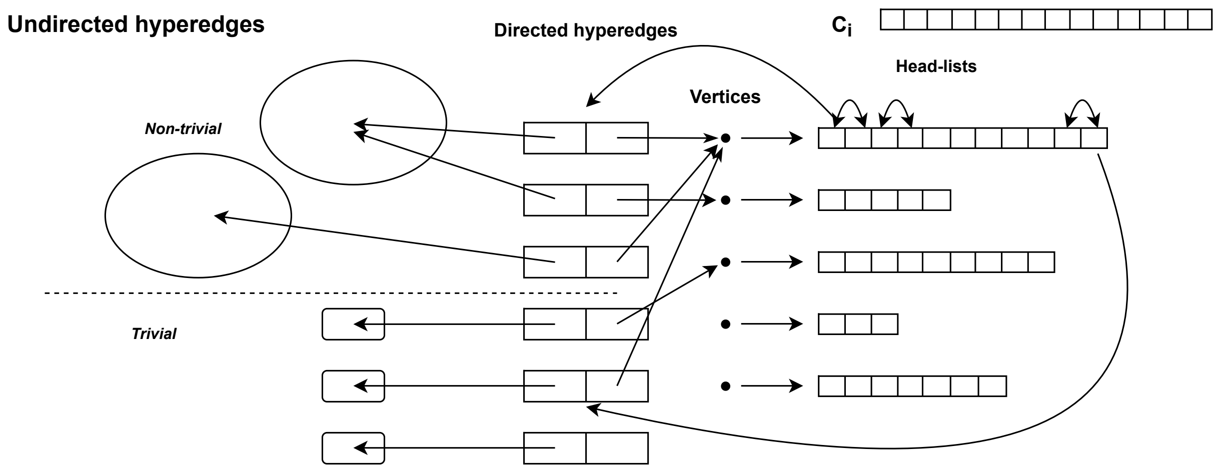


Figure 8.1: Illustration of the data structure used for Algorithm 8.3.

The following lemma implies that $\mathcal{H}_{G'}$ can always be stored in $O(|V|)$ space with our data structure, since G' has at most $k|V| - \ell$ trivial M-components.

Lemma 8.2 *Given a graph $G = (V, E)$, let $\mathcal{C}$ denote the family of the vertex sets of its nontrivial (k,ℓ)-M-components. Then* $\sum_{C \in \mathcal{C}} |C| \le 2(k|V| - \ell)$.

Proof By Lemma 4.6 $\mathcal{H}_G$ is a (k,ℓ)-sparse hypergraph. Let m_i denote the number of non-trivial (k,ℓ)-M-components of G with cardinality i, where $i \in \{1, \ldots, |V|\}$. (Note that $m_1 = 0$ when

$k \leq \ell$.) Then counting the hyperedges of $\mathcal{H}_G = (V, \mathcal{E})$ corresponding to the non-trivial M-components of G we get that its number equals $\sum_{i=1}^{|V|}(ki-\ell)m_i$. Hence $\sum_{i=1}^{|V|}(ki-\ell)m_i \leq |\mathcal{E}| \leq k|V|-\ell$. Observe that $i \leq 2(ki-\ell)$ holds for each $i \in \{2,\ldots,|V|\}$ as from $2k > \ell$ we get $i \leq 2ki - \ell i$. $1 \leq 2(k-\ell)$ also holds when $k > \ell$ (and hence m_1 can be nonzero). Thus we get $\sum_{C\in\mathcal{C}}|C| = \sum_{i=1}^{|V|} i m_i \leq \sum_{i=1}^{|V|} 2(ki-\ell)m_i \leq 2(k|V|-\ell)$, as claimed. $\square$

Let us take the edges of G one by one in a breadth-first manner. When we reach a new vertex i, we create a 0-1 vector c_i of length $|V|$. The instances of 1 in c_i indicate the vertices j for which there exists a (k,ℓ)-M-component containing i and j together (that is, c_i is 1 exactly at the $\mathcal{H}_{G'}$-neighbors of i). This vector can be created for any particular vertex in $O(|V|)$ time as we can traverse all the undirected hyperedges of $\mathcal{H}_{G'}$ in $O(|V|)$ time by Lemma 8.2.

Now with a new edge ij, we repeat the following subroutine.

Algorithm 8.3 *Input: A maximal (k,ℓ)-sparse subgraph G^* of a graph G', the M-component hypergraph $\mathcal{H}_{G'}$ of G', its orientation $\vec{\mathcal{H}}_{G'}$ where the maximum in-degree is k (all of these stored in the data structure mentioned above along with the vector c_i), and an edge ij.*

Output: A maximal (k,ℓ)-sparse subgraph G^* of $G'+ij$, the M-component hypergraph $\mathcal{H}_{G'+ij}$ of $G'+ij$, and its orientation $\vec{\mathcal{H}}_{G'+ij}$ where the maximum in-degree is k.

1. *Check in c_i whether i and j are contained in the same (k,ℓ)-M-component.*

2. ***If*** *there exists a (k,ℓ)-M-component that contains i and j,* ***then***

Return *the input unchanged*

3. ***else***

Find a reorientation of $\vec{\mathcal{H}}_{G'}$ in which the sum of in-degrees of i and j is at most $2k-\ell-1$ and the in-degree of $v \in V - \{i,j\}$ is at most k by performing constant number of backward BFSs from i and j and switching the orientation of paths ending at i or j and starting at a vertex of degree less than k.

4. ***If*** *there exists such a reorientation,* ***then***

Add ij to E^* and to $\mathcal{E}'$, and orient it so that $d_{\vec{\mathcal{H}}_{G'}}(i) \leq k$ and also $d_{\vec{\mathcal{H}}_{G'}}(j) \leq k$.

5. ***else***

(The last backward BFS finds a minimal set X which has in-degree zero in $\vec{\mathcal{H}}_{G'}$ and contains both i and j.)

Update $\mathcal{H}_{G'}$ by removing all hyperedges induced by X and adding $k|X| - \ell$ copies of X as a hyperedge to $\mathcal{E}'$. The heads of these hyperedges in $\vec{\mathcal{H}}_{G'}$ will be exactly the heads of the omitted hyperedges. G^* remains the same in this case.

Next, we show that the above procedure maintains our data structure correctly.

Lemma 8.4 *Algorithm 8.3 outputs a maximal (k,ℓ)-sparse subgraph G^* of $G' + ij$, the (k,ℓ)-M-component hypergraph $\mathcal{H}_{G'+ij}$ of $G' + ij$, and its orientation $\vec{\mathcal{H}}_{G'+ij}$ where the maximum in-degree is k.*

Proof If the algorithm stops at STEP 2, then i and j are contained in the same (k,ℓ)-M-component and the edge ij cannot be used to construct larger (k,ℓ)-M-components. Hence $G' + ij$ has the same (k,ℓ)-M-components as before and G^* is still its maximal (k,ℓ)-sparse subgraph.

If we find an eligible reorientation of $\vec{\mathcal{H}}_{G'}$ in STEP 3, then $\mathcal{H}_{G'} + ij$ is (k,ℓ)-sparse by the hypergraphic version of Lemma 8.1, since it implies that each set X containing i and j can induce at most $2k - \ell + k(|X| - 2) = k|X| - \ell$ edges in $\mathcal{H}_{G'} + ij$. Now let us prove that $G^* + ij$ is also (k,ℓ)-sparse.

Claim 8.5 *$G^* + ij$ is (k,ℓ)-sparse.*

Proof Let $X \subseteq V$ be an arbitrary set. When X does not contain both i and j, $i_{G^*+ij}(X) = i_{G^*}(X) \le k|X| - \ell$ is obvious. Therefore, we may assume that X contains both i and j.

Then $i_{G^*+ij}(X) = i_{G^*}(X) + 1 \le k|X| - \ell + 1$ and hence the (k,ℓ)-sparsity condition follows whenever $i_{G^*}(X) < k|X| - \ell$. Hence we may assume that $i_{G^*}(X) = k|X| - \ell$. By the second statement in Lemma 4.6 and the (k,ℓ)-sparsity of $\mathcal{H}_{G'} + ij$ there is a (k,ℓ)-M-component of G' with vertex set C, so that X induces some, but not all the edges of G' induced by C, $C \nsubseteq X$. Lemma 4.6 implies in contrast that $k|C| - \ell \le i_{\mathcal{H}_{G'}}(C) \le i_{G^*}(C) \le k|C| - \ell$, that is, equality holds throughout. As X and C share an edge, we can use Lemma 2.9 to show that $i_{G^*}(X \cup C) = k|X \cup C| - \ell$. However, again by the second statement of Lemma 4.6 and the (k,ℓ)-sparsity of $\mathcal{H}_{G'} + ij$, there exists a (k,ℓ)-M-component of G', so that $X \cup C$ induces some, but not all of its edges. Subsequently adding these vertex sets of (k,ℓ)-M-components of G' to our set with which it shares at least one edge, we get the set V, contradicting the (k,ℓ)-sparsity of $\mathcal{H}_{G'} + ij$. Therefore, $G^* + ij$ is (k,ℓ)-sparse. $\square$

Now Claim 8.5 also implies that ij is non-redundant in $G' + ij$ since it increases the size of the maximal (k,ℓ)-sparse subgraph G^* of G' and hence $\mathcal{H}_{G'} + ij$ is the M-component hypergraph of $G' + ij$.

The orientation of $\mathcal{H}_{G'} + ij$ provided by the algorithm in STEP 4 obviously satisfies our conditions.

When no eligible reorientation exists (in STEP 5), the algorithm takes the set X with in-degree 0. As no reorientation can be made, every vertex has in-degree k in $X - \{i,j\}$ and $d_{\vec{H}_{G'}}(i) + d_{\vec{H}_{G'}}(j) = 2k - \ell$. Hence X induces $k(|X| - 2) + 2k - \ell = k|X| - \ell$ hyperedges in $\mathcal{H}_{G'}$. By its construction, one can see that X is the unique minimal set with the property of containing i and j and inducing $k|X| - \ell$ hyperedges simultaneously.

By Lemma 4.6 and the (k,ℓ)-sparsity of G^*, X induces exactly $k|X| - \ell$ edges in G^*. Moreover, this is true for any other maximal (k,ℓ)-sparse subgraph of G' other than G^*, as well. Hence $G' + ij$ cannot contain any (k,ℓ)-M-circuit which is not a subgraph of $G'[X] + ij$. That is, the vertex set Y of the (k,ℓ)-M-component of $G' + ij$ containing both i and j is a subset of X.

As Y forms a (k,ℓ)-M-component in $G' + ij$ and ij is a (k,ℓ)-redundant edge in $G' + ij$, G^* is still a maximal (k,ℓ)-sparse subgraph of $G' + ij$ and the (k,ℓ)-M-component hypergraph $\mathcal{H}_{G'+ij}$ of $G' + ij$ induces at least $k|Y| - \ell$ parallel copies of the hyperedges on Y. Hence, by Lemma 4.6, Y induces exactly $k|Y| - \ell$ hyperedges in $\mathcal{H}_{G'+ij}$ and $G^*[Y]$ is (k,ℓ)-tight. By the definition of (k,ℓ)-M-components, we can conclude that Y induces no edge of any other (k,ℓ)-M-component of $G' + ij$. Note that the definition of (k,ℓ)-M-components implies that the (k,ℓ)-M-components of $G' + ij$ other than Y are also (k,ℓ)-M-components of G'. Thus Y induces either all or none of the edges of each (k,ℓ)-M-component of G'. By applying the second statement of Lemma 4.6 to $\mathcal{H}_{G'}$, G^* and Y, we can see that $\mathcal{H}_{G'+ij}[Y]$ is a (k,ℓ)-tight hypergraph containing i and j. This together with the definition of X proves that $X \subseteq Y$ and hence $X = Y$.

Finally, it is obvious that the in-degree of each vertex is at most k in the orientation provided by the algorithm. $\square$

Next, we show that Algorithm 8.3 needs $O(|V|)$ running time and the total running time of our algorithm which computes the M-component hypergraph is $O(|V|^2)$.

Theorem 8.6 *Let k and ℓ be positive integers such that $\ell < 2k$. Then the (k,ℓ)-M-component hypergraph $\mathcal{H}_G$ of a graph $G = (V,E)$ (with at most $O(|V^2|)$ edges) can be calculated in $O(|V|^2)$ time. Also, a maximal (k,ℓ)-sparse subgraph G^* of G can be found in the same running time.*

Moreover, the algorithm provides an orientation $\vec{\mathcal{H}}_G$ of $\mathcal{H}_G$ that can be used to decide whether $\mathcal{H}_G + ij$ is (k,ℓ)-sparse and, if not, we can determine $\mathcal{T}_{\mathcal{H}_G}(ij)$ for arbitrary $i,j \in V$ in $O(|V|)$ running time.

Proof We shall use the algorithm presented above. We have seen that the vector c_i used in our data structure can be maintained in $O(|V|^2)$ total time if we consider the edges of G in a breadth-first manner. During the algorithm, we run Algorithm 8.3 for each edge.

Let us show that each step of Algorithm 8.3 requires at most $O(|V|)$ running time. Step 1 can be run in $O(1)$ time using c_i, obviously. Next, we try to find a reorientation by running backwards BFSs from i and j on $\vec{\mathcal{H}}_{G'}$. With the head-lists, each BFS needs $O(|V|)$ running time, since we only need to traverse each undirected hyperedge of $\mathcal{H}_{G'}$ once, which is fast by Lemma 8.2. However, we also need to refresh the head-lists in every reorientation. Adding a directed hyperedge to the head-list of v in $O(1)$ is trivial, and as we store the head list in doubly linked lists, we can also remove directed hyperedges from the head-list of each vertex on a reoriented path in $O(1)$ time. As we need to run at most $2k - \ell = O(1)$ reorientations, this requires only $O(|V|)$ total time. It is obvious that Step 4 runs in $O(1)$ time. In Step 5, to meet our running time constraints, we first create one instance of the undirected hyperedge induced by X. Then we loop through all the undirected hyperedges of $\mathcal{T}_{\mathcal{H}_{G'}}(ij)$, and remove them. Now we refresh the appropriate directed edges: we redirect the pointers of the underlying undirected hyperedge, while we do not change the head or the head-lists. After this step we update c_i. This whole procedure requires only $O(|V|)$ time.

Claim 8.7 *There are only $O(|V|)$ edges of G for which we run Steps 3-5 from Algorithm 8.3.*

Proof Note that we add at most $k|V| - \ell = O(|V|)$ edges to G^* during the whole algorithm, since it is (k,ℓ)-sparse. Observe also that, when a new (k,ℓ)-M-component arises, we merge at least two (k,ℓ)-M-components. Only the edges of G^* arise as a new (trivial) M-component during the algorithm, hence there are at most $O(|V|)$ merges. □

By Claim 8.7 the steps of Algorithm 8.3 that require $O(|V|)$ running time run at most $O(|V|)$ times while all other edges are omitted in Step 2 in $O(1)$ time. Thus the total running time is indeed $O(|V|^2)$.

When we take an extra edge ij, we can use the orientation provided and run Steps 3-5 of Algorithm 8.3 in $O(|V|)$ time again to decide whether it maintains the sparsity of $\mathcal{H}_G$. If it does not, we calculate $\mathcal{T}_{\mathcal{H}_G}(ij)$, which is the subhypergraph induced by the new hyperedge provided by STEP 5. □

We shall analyse briefly the spatial complexity of the algorithm we presented. Obviously, the input might consist of $O(|V|^2)$ size of data, however, we do not need to store all of it. $O(|V|)$ storage is needed for storing V and E^*. By Lemma 8.2 the undirected hyperedges use only $O(|V|)$ space and hence the directed hyperedges can be stored in only $O(|V|)$ space, as well. The order of the vertices assures us that at any given point we need only $O(|V|)$ storage for the c_i vectors. Hence, the additional space complexity of our algorithm is $O(|V|)$.

8.3 Making (k,ℓ)-M-component hypergraphs (k,ℓ)-redundant

In this section we present an $O(|V|^2)$ algorithm to solve the reduced (k,ℓ)-redundant augmentation problem for (k,ℓ)-M-component hypergraphs. This solution will match the optimum value presented in Theorem 3.1. In this section we make some clear restrictions on the input, as we present our solution only to M-component hypergraphs, instead of any hypergraphs and also we present it to (k,ℓ)-rigidity instead of (m,ℓ)-rigidity. We have two main reasons to do so.

The first is that the algorithm is much simpler this way. We can use some extra insights from Chapter 4 by using only the (k,ℓ)-M-component hypergraphs and giving the solution to (k,ℓ)-redundancy instead of (m,ℓ)-redundancy simplifies the methods. For example, we do not need to assume the conditions (A) presented in Section 3.2, as for (k,ℓ)-tightness they hold automatically. This also made the implementation of this algorithm much simpler, as it is presented in the Appendix.

The other reason is that for the algorithms of Sections 8.4 and 8.6 solving the problem for (k,ℓ)-M-component hypergraphs is sufficient. Anyone interested in a detailed description of an $O(|V|^2)$ algorithm that works for (m,ℓ)-tight graphs satisfying the (*) condition or a similarly efficient solution to general (m,ℓ)-tight hypergraphs is referred to [51]. In that paper we analyse these problems with an algorithm based on the details of the proof of Theorem 3.11. Also, the case when G is an (m,ℓ)-rigid graphs with $m \leq \ell$ was solved with $O(|V|^2)$ running time in an earlier paper of Király [44]. We shall also note that most of the ideas of this section translate to (m,ℓ)-redundant augmentation problems on hypergraphs and can be generalized with some extra work to several special cases of general inputs, as well.

With that said, throughout this section, let G be a (k, ℓ)-rigid graph on at least $k^2 + 2$ vertices, where k and ℓ are positive integers such that $\ell < 2k$. (We note that this restriction on the number of vertices has no implication on the running time, as k is a constant hence we exclude constant number of graphs. The problem can always be solved for constant number of graphs via brute force in $O(1)$ running time. We also note that $k^2 + 2 > 2k$ always holds.) $\mathcal{H}_G$ will denote the (k, ℓ)-M-component hypergraph of G, G^* will be a (k, ℓ)-tight spanning subgraph of G and, if not stated otherwise, $\mathcal{T}(ij)$ and $V(ij)$ will denote $\mathcal{T}_{\mathcal{H}_G}(ij)$ and $V_{\mathcal{H}_G}(ij)$ for $i, j \in V$, respectively.

8.3.1 Finding a transversal of the MCT sets

In this first half of the algorithm we shall find a transversal of the (k, ℓ)-MCT sets of the (k, ℓ)-M-component hypergraph. This part will play a key role in Sections 8.4 and 8.6, as well. We start with running the algorithm of Section 8.2 on G. As mentioned there, this method generates the auxiliary directed hypergraph $\vec{\mathcal{H}}_G$ that can be used to determine $\mathcal{T}(ij)$ for arbitrary $i, j \in V$ in $O(|V|)$ running time.

Our algorithm will use the following greedy subroutine several times.

Algorithm 8.8 *Input: The (k, ℓ)-M-component hypergraph $\mathcal{H}_G = (V, \mathcal{E})$ of a (k, ℓ)-rigid graph G on at least $k^2 + 2$ vertices $(0 < \ell < 2k)$, a set $L \subseteq V$ and a vertex $i \in V$.*

Output: A (minimum cardinality) vertex set $V'(i, L) = \{j_2, \dots, j_r\}$ such that $V = \bigcup_{s=2}^{r} V(ij_s)$.

1. *Initialize $V'(i, L) := \emptyset$. All vertices are unmarked.*
2. *Mark i.*
3. *Explore all vertices $j \in L$ first, then all other vertices $j \in V - L$:*
4. ***If** j is unmarked **then**:*
5. *Calculate $\mathcal{T}(ij)$;*
6. *Mark all unmarked vertices in $V(ij)$;*
7. *$V'(i, L) := [V'(i, L) - V(ij)] + j$.*
8. ***Return** $V'(i, L)$*

Since by the end of Algorithm 8.8 every vertex is marked, it follows that every vertex is part of $V(ij_s)$ for some $j_s \in V'(i, L)$. Therefore this method indeed gives an output for which $V = \bigcup_{s=2}^{r} V(ij_s)$.

Observation 8.9 *If $V'(i,L) = \{j\}$ for a vertex $j \in V$, then $\mathcal{T}(ij) = \mathcal{H}_G$, and hence the edge ij makes $\mathcal{H}_G$ (k,ℓ)-redundant.*

The set L is added to the algorithm to make it compatible with the global rigidity augmentation problem from Chapter 5. In Section 8.4 we use it to find a transversal of the atoms of G. As it shall be described in detail, L will denote the vertices from the 3-ends then. L is not used in this section.

The following lemma will be used several times in this section.

Lemma 8.10 *Let G be a (k,ℓ)-rigid graph on at least k^2+2 vertices and $\mathcal{H}_G$ its (k,ℓ)-M-component hypergraph. Let G^* be a spanning (k,ℓ)-tight subgraph of G. Let $i \in V$ be such that $d_{G^*}(i) \leq 2k-1$.*

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two (k,ℓ)-tight subhypergraphs of $\mathcal{H}_G$ such that $i \in V(\mathcal{T}_1) \cap V(\mathcal{T}_2)$ and $|V(\mathcal{T}_1)| \geq 3$, $|V(\mathcal{T}_2)| \geq 3$. Then $\mathcal{T}_1 \cup \mathcal{T}_2$ is a (k,ℓ)-tight subhypergraph of $\mathcal{H}_G$ and $d_{\mathcal{H}_G}(V(\mathcal{T}_1) - V(\mathcal{T}_2), V(\mathcal{T}_2) - V(\mathcal{T}_1)) = 0$.

Proof By Observation 4.7 $G^*[V(\mathcal{T}_1)]$ and similarly $G^*[V(\mathcal{T}_2)]$ are (k,ℓ)-tight subgraphs of G^*. By Lemma 2.13, i has a degree at least k in both of them implying that $\mathcal{T}_1$ and $\mathcal{T}_2$ have a common edge incident with i and hence $|V(\mathcal{T}_1) \cap V(\mathcal{T}_2)| \geq 2$. Now our statement follows by Lemma 2.9. $\square$

An output $V'(i,L)$ is called **simple**, if $|V(i,j)| \geq 3$ for every $j \in V'(i,L)$. Let us show now how Algorithm 8.8 can be used with a simple output to determine if there is one edge uv, so that $\mathcal{H}_G + uv$ is (k,ℓ)-redundant.

Lemma 8.11 *Let G, $\mathcal{H}_G$, G^* and i be defined as in Lemma 8.10 and let $V'(i,L)$ be the output of Algorithm 8.8 with inputs G and i (and L as an arbitrary subset of V). Suppose that there is an edge uv so that $\mathcal{H}_G + uv$ is (k,ℓ)-redundant. Suppose moreover that $V'(i,L)$ is simple and $|V'(i,L)| \geq 2$. Then there exists $j_2, j_3 \in V - i$ such that $V'(i,L) = \{j_2, j_3\}$, and there exists a vertex $y \in V$ such that $\mathcal{T}(j_2y) = \mathcal{H}_G$ or $\mathcal{T}(j_3y) = \mathcal{H}_G$.*

Proof Let $j_2, j_3 \in V'(i,L)$ be two vertices for which $u \in V(ij_2)$ and $v \in V(ij_3)$. Note that $v \notin V(ij_2)$, since otherwise $\mathcal{H}_G = \mathcal{T}(uv) \subseteq \mathcal{T}(ij_2)$ would hold by Lemma 2.11, contradicting $|V'(i,L)| \geq 2$. By the run of Algorithm 8.8 $j_3 \notin V(ij_2)$. Similarly one can show that $u, j_2 \notin V(ij_3)$. As $V'(i,L)$ is simple, both $V(ij_2)$ and $V(ij_3)$ have a cardinality of at least three. Hence

Lemma 8.10 implies that $\mathcal{T}(ij_2)\cup\mathcal{T}(ij_3)$ is (k,ℓ)-tight and $d_{\mathcal{H}_G}(V(ij_2)-V(ij_3), V(ij_3)-V(ij_2)) = 0$. Since $\mathcal{T}(ij_2)\cup\mathcal{T}(ij_3)$ induces both u and v, $\mathcal{H}_G = \mathcal{T}(uv) \subseteq \mathcal{T}(ij_2)\cup\mathcal{T}(ij_3)$ by Lemma 2.11. Hence $V'(i,L) = \{j_2, j_3\}$, as we claimed.

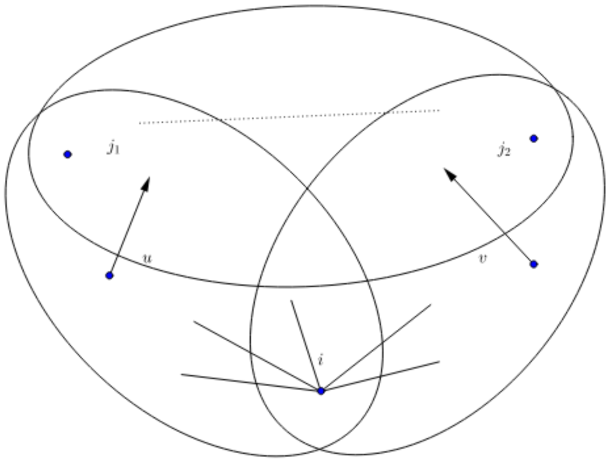


Figure 8.2: There exists a vertex $y \in V$ such that $\mathcal{T}(j_2y) = \mathcal{H}_G$ or $\mathcal{T}(j_3y) = \mathcal{H}_G$

In case of $|V(ij_2) - V(ij_3)| = 1$ (or $|V(ij_3) - V(ij_2)| = 1$, respectively), $u = j_2$ (or $v = j_3$, respectively) hence the statement is obviously true by having $y = v$ (or $y = u$, respectively). Thus we may suppose that $|V(ij_2) - V(ij_3)| \geq 2$ and $|V(ij_3) - V(ij_2)| \geq 2$. Let us consider $\mathcal{T}(j_2j_3)$. As $d_{\mathcal{H}_G}(V(ij_2) - V(ij_3), V(ij_3) - V(ij_2)) = 0$, no (graph) edge connects j_2 to j_3 in $\mathcal{H}_G$ and hence $\mathcal{H}_G[\{j_2, j_3\}]$ is not (k,ℓ)-tight. Thus $|V(j_1j_2)| \geq 3$. As $V(ij_2)\cup V(ij_3) = V$, $V(j_2j_3)$ intersects one of $V(ij_2)$ and $V(ij_3)$ (say, $V(ij_2)$) in at least 2 vertices. Hence by Lemma 2.9 $\mathcal{T}(ij_2)\cup\mathcal{T}(j_2j_3)$ is (k,ℓ)-tight, containing i and j_3. Thus $\mathcal{T}(ij_3)\subseteq\mathcal{T}(ij_2)\cup\mathcal{T}(j_2j_3)$ by Lemma 2.11. Consequently, $V(j_2j_3) \supseteq V(ij_3) - V(ij_2)$ and similarly $V(j_2j_3) \supseteq V(ij_2) - V(ij_3)$. This implies that $u, v \in V(j_2j_3)$, resulting $\mathcal{H}_G = \mathcal{T}(uv) \subseteq \mathcal{T}(j_2j_3)$ by Lemma 2.11, as claimed. $\square$

Lemma 8.12 *Let G, $\mathcal{H}_G$, G^* and i be defined as in Lemma 8.10 and let $V'(i,L)$ be the output of Algorithm 8.8 with inputs G and i (and L as an arbitrary subset of V). Suppose that there is no edge uv such that $\mathcal{H}_G + uv$ is (k,ℓ)-redundant and that $V'(i,L)$ is simple with $|V'(i,L)| \geq 2$. Then every vertex of $V'(i,L)$ is contained in a (k,ℓ)-MCT set.*

Proof For any $v \in V'(i,L)$, $\bigcup_{j\in V'(i,L)-v}\mathcal{T}(ij)$ is (k,ℓ)-tight by the sequential application of Lemma 8.10 (note that $\bigcup_{j\in V'(i,L)-v}\mathcal{T}(ij)$ cannot be empty by the condition $|V'(i,L)| \geq 2$). Hence $V - \bigcup_{j\in V'(i,L)-v} V(ij)$ is a (k,ℓ)-co-tight set which contains v and has a (k,ℓ)-MCT subset

C, which has no vertex from $V'(i,L) \cup \{i\}$. Now as $\bigcup_{j \in V'(i,L)} \mathcal{T}(ij) = \mathcal{H}_G$, Observation 3.7 and Lemma 2.12 imply that $v \in C$ is from a (k,ℓ)-MCT set. $\square$

Let us now consider the case, when $V'(i,L)$ is not simple.

Lemma 8.13 *Let G, $\mathcal{H}_G$, G^* and i be defined as in Lemma 8.10 and let $V'(i,L)$ be the output of Algorithm 8.8 with inputs G and i (and L as an arbitrary subset of V). Suppose moreover that $V'(i,L)$ is not simple and $|V'(i,L)| \geq 2$. Let $N := \{v | v \in V'(i,L)$ and $|V(iv)| = 2\}$. If there exists an edge uv so that $G + uv$ is (k,ℓ)-redundant, then there exists $n \in N$ such that there is a vertex $y \in V$ for which $\mathcal{T}(ny) = \mathcal{H}_G$. Otherwise, there exists a vertex $n \in N$, so that n is contained in a (k,ℓ)-MCT set of $\mathcal{H}_G$.*

Proof As $V'(i,L)$ is not simple, N is nonempty. Suppose first that there exists an edge uv so that $\mathcal{H}_G + uv$ is (k,ℓ)-redundant.

If $u \in N$ or $v \in N$, the statement obviously holds. However, if there are the vertices $j_2, j_3 \in V'(i,L)$ for which, say $u \in V(ij_2)$ and $v \in V(ij_3)$ with $|V(ij_2)| \geq 3$ and $|V(ij_3)| \geq 3$, then similarly to the proof of Lemma 8.11, $T(ij_2) \cup T(ij_3) = \mathcal{H}_G$ by Lemma 8.10. This contradicts $N \neq \emptyset$.

If there is no such edge uv, then let $\mathcal{T} = \bigcup\{\mathcal{T}(ij) : j \in V'(i,L) - N\}$. Now $\mathcal{T}$ is nonempty by our assumption on $|V| \geq k^2 + 2$ and the fact that $|N| \leq 2k - 1$. $\mathcal{T}$ is a (k,ℓ)-tight subhypergraph of $\mathcal{H}_G$ by the sequential application of Lemma 8.10, resulting N forming a (k,ℓ)-co-tight set. Hence N contains a (k,ℓ)-MCT set by Lemma 3.9, and thus a vertex is included in a (k,ℓ)-MCT set. $\square$

If we have a vertex from a (k,ℓ)-MCT set, we can use it with the help of the following lemma.

Lemma 8.14 *Let G, $\mathcal{H}_G$, and G^* be defined as in Lemma 8.10. Suppose that there is no edge uv such that $\mathcal{H}_G + uv$ is (k,ℓ)-redundant. Let i' be a vertex from a (k,ℓ)-MCT set of $\mathcal{H}_G$ and let L be an arbitrary set. Let $V'(i',L)$ be the result of Algorithm 8.8 with the input $\mathcal{H}_G$, i' and L. Then $V'(i',L) \cup \{i'\}$ is a transversal of the (k,ℓ)-MCT sets of $\mathcal{H}_G$, and hence every vertex from $V'(i',L)$ is a vertex from a (k,ℓ)-MCT set of $\mathcal{H}_G$. Moreover, if $L \cap C \neq \emptyset$ for a (k,ℓ)-MCT set C for which $i' \notin C$, then $V'(i',L) \cap C \cap L \neq \emptyset$.*

Proof If i' is from a (k,ℓ)-MCT set of $\mathcal{H}_G$, then by Lemma 3.19 we know that the inclusion-wise maximal generated (k,ℓ)-tight subhypergraphs of $\mathcal{H}_G$ are exactly the ones generated by two vertices from different (k,ℓ)-MCT sets. Hence it follows from Observation 3.7 and Lemma 3.19 that $V'(i',L)$ consists of exactly one vertex from every other (k,ℓ)-MCT set. As we check first the vertices in L, it is clear that we choose a vertex from $L\cap C$ for each (k,ℓ)-MCT set C for which $C\cap L\neq\emptyset$ and $i\notin C$ by Lemma 3.19. □

Lemma 8.15 *Algorithm 8.8 runs in $O(|V|^2)$ time.*

Proof Remember that we can compute $\mathcal{T}(ij)$ in $O(|V|)$ running time for any vertex j by Theorem 8.6. Hence, as we calculate $\mathcal{T}(ij)$ only $O(|V|)$ times, the total running time concludes in $O(|V|^2)$. The marking and the set operations can be executed without additional complexity in $O(|V|)$ running time each. □

Now by these results we present the first part of our algorithm.

Algorithm 8.16 *Input: (k,ℓ)-rigid graph G on at least k^2+2 vertices, where $0<\ell<2k$ and a vertex set $L\subseteq V$.*

Output: If there exists an edge e for which $\mathcal{H}_G+e$ is (k,ℓ)-redundant (where $\mathcal{H}_G$ denotes the (k,ℓ)-M-component hypergraph of G), then an edge ij for which $\mathcal{H}_G+ij$ is (k,ℓ)-redundant. Otherwise, a transversal vertex set P on the (k,ℓ)-MCT sets of $\mathcal{H}_G$, so that $|L\cap P|$ is maximal amongst all the transversals of the (k,ℓ)-MCT sets.

1. *Run the algorithm of Theorem 8.6 on G. Get the (k,ℓ)-M-component hypergraph, $\mathcal{H}_G$ and a (k,ℓ)-tight subgraph, G^*.*
2. *Choose a vertex i with minimum degree from G^*.*
3. *Run Algorithm 8.8 with i and L, resulting $V'(i,L)$.*
4. ***If** $V'(i,L)=\{j\}$, **then***

   ***Return** the edge ij*
5. ***If** $V'(i,L)=\{j_1,j_2\}$, **then***

   Check if $\mathcal{T}(j_1v)=\mathcal{H}_G$ or $\mathcal{T}(j_2v)=\mathcal{H}_G$ for every $v\in V$.

   If** there exists such an edge jv, **then

   ***Return** jv.*
6. ***If** $V'(i,L)$ is not simple, **then***

$$N:=\{v|v\in V'(i,L)\text{ and }|V(iv)|=2\}$$

else

$N := \{v\}$ for any $v \in V'(i,L)$.

7. ***For*** *every $i' \in N$,*

Run Algorithm 8.8 with i' and L resulting $V'(i',L)$.

8. *Choose $i_0 = argmin\{|V'(i',L)| : i' \in N\}$.*

9. *Run Algorithm 8.8 with i_0 and L resulting $V'(i_0,L)$. Take $i_1 \in V'(i_0,L)$*

10. ***If*** *$V'(i_0,L) = \{i_1\}$,* ***then***

Return *the edge $i_0 i_1$*

11. ***else***

Run Algorithm 8.8 with i_1 and L resulting $V'(i_1,L)$.

Return *$P := V'(i_1,L) \cup \{i_1\}$*

Lemma 8.17 *Let $G = (V,E)$ be a (k,ℓ)-rigid graph on at least k^2+2 vertices, where $0 < \ell < 2k$, and let $\mathcal{H}_G$ be its (k,ℓ)-M-component hypergraph. Algorithm 8.16 results either one edge ij, such that $\mathcal{H}_G + ij$ is (k,ℓ)-redundant, or a transversal vertex set P on the (k,ℓ)-MCT sets of $\mathcal{H}_G$, so that $|P \cap L|$ is maximal. The running time of Algorithm 8.16 is $O(|V|^2)$.*

Proof Let G^* be the (k,ℓ)-tight subgraph of G that we get in Step 1 in Algorithm 8.16. By summing up the degrees one can show that G^* has a vertex i, so that $d_{G^*}(i) \leq 2k-1$ (remember, now $\ell > 0$). Hence $d_{G^*}(i) \leq 2k-1$ holds for the vertex i chosen in Step 2.

By Observation 8.9 and Lemmas 8.11 and 8.13, the algorithm returns with one edge in Step 4, 5 or 10 if and only if such an edge exists for which $\mathcal{H}_G + e$ is (k,ℓ)-redundant. Hence we may assume that there is no edge the addition of which can make $\mathcal{H}_G$ (k,ℓ)-redundant. Therefore, Lemmas 8.13 and 8.12 imply that at least one vertex from N is from a (k,ℓ)-MCT set of $\mathcal{H}_G$. By Lemma 8.14, for this vertex i_0, $V'(i_0,L) \cup \{i_0\}$ is a transversal of the (k,ℓ)-MCT sets of $\mathcal{H}_G$. Note that $V'(i,L) \cup \{i\}$ must intersect each (k,ℓ)-MCT set C for arbitrary choice of $i \in V$ since otherwise $V(iv) \subseteq V - C$ holds for each $v \in V'(i,L)$ by Lemma 2.11, contradicting its construction. Hence the choice of i_0 in Step 8 indeed results a vertex from a (k,ℓ)-MCT set.

Now Lemma 8.14 implies that the vertex $i_1 \in V'(i_0,L)$ chosen in Step 9 is contained by a (k,ℓ)-MCT set C of $\mathcal{H}_G$, moreover, if $C \cap L \neq \emptyset$, then $i_1 \in L$. This implies, by using Lemma 8.14 again that P is a transversal of the (k,ℓ)-MCT sets of $\mathcal{H}_G$, so that $|L \cap P|$ is maximal amongst all the transversals of the (k,ℓ)-MCT sets.

Step 1 has $O(|V|^2)$ running time by Theorem 8.6. We can easily find i in $O(|V|^2)$ running time (in fact, $O(|V|)$ time is enough, as G^* has $O(|V|)$ edges). Step 3 runs in $O(|V|^2)$ time by Lemma 8.15. As we compute $\mathcal{T}(ij)$ only $O(|V|)$ times in Step 5, this can be executed in $O(|V|^2)$ total time by Theorem 8.6. As $d_{G^*}(i) \leq 2k-1$, $|N| \leq \frac{2k-1}{2k-\ell}$, hence N has $O(1)$ size and Steps 7 and 8 run in $O(|V|^2)$ time by Lemma 8.15. Steps 9 and 11 need $O(|V|^2)$ running time again by Lemma 8.15. Therefore, the total running time of Algorithm 8.16 is indeed $O(|V|^2)$. □

8.3.2 Optimal augmenting edge set

As we saw in Algorithm 8.16, if there exists an edge ij, so that $\mathcal{H}_G + ij$ is (k,ℓ)-redundant, then we can find such in $O(|V|^2)$ running time. Otherwise, we can find a transversal P on the (k,ℓ)-MCT sets of $\mathcal{H}_G$. By Lemma 3.16 we can use it to add $|P|-1$ edges that make $\mathcal{H}_G$ (k,ℓ)-redundant. By Theorem 3.5 this is at most two times the optimum.

Now we show, how we can find an optimal augmenting edge set in $O(|V|^2)$ time, if there is no single edge ij, so that $\mathcal{H}_G+ij$ is (k,ℓ)-redundant. The algorithm will heavily rely on Lemma 3.17. This algorithm is identical to the one mentioned after Algorithm 6.7 in [45]. However, here we present it in detail. Note that the main ideas of this algorithm already appeared in [22, 44].

From now on, we always suppose that L is empty in Algorithm 8.8. Let us refer to $V'(i,\emptyset)$ as $V'(i)$. Also, this part of the algorithm works for any (k,ℓ)-tight hypergraph hence we shall not assume that $\mathcal{H}$ is a (k,ℓ)-M-component hypergraph.

Algorithm 8.18 *Input: A (k,ℓ)-tight hypergraph $\mathcal{H}=(V,\mathcal{E})$, where $0<\ell<2k$.*

Output: A list of edges F for which $\mathcal{H}+F$ is (k,ℓ)-redundant, so that $|F|$ is minimal.

1. *Run Algorithm 8.16.*

__If__ it returns an edge ij, __then__

__Return__ $F=\{ij\}$.

(else it returned a transversal P)

3. *Suppose that $P=\{i_1,\dots,i_h\}$ and let $F=\emptyset$.*

4. ***While*** *$h\geq 4$* ***do***

Compute $\mathcal{T}(i_1i_{h-2})$ and $\mathcal{T}(i_{h-1}i_h)$.

__If__ $\mathcal{T}(i_1i_{h-2})\cup\mathcal{T}(i_{h-1}i_h)=\mathcal{T}(i_1i_{h-2})\cup\mathcal{T}(i_1i_{h-1})\cup\mathcal{T}(i_1i_h)$, __then__

$F:=F+i_{h-1}i_h$.

else

$F := F + i_{h-2}i_h$,

$i_{h-2} := i_{h-1}$.

$h := h - 2$.

5. *(Final step)*

If $h = 2$, ***then***

$F := F + i_1 i_2$.

If $h = 3$, ***then***

$F := F \cup \{i_1 i_2, i_1 i_3\}$.

Return F.

Theorem 8.19 *Given a (k,ℓ)-tight hypergraph $\mathcal{H} = (V, \mathcal{E})$ with $0 < \ell < 2k$, Algorithm 8.18 finds a minimum cardinality edge set F, such that $\mathcal{H} + F$ is (k,ℓ)-redundant in $O(|V|^2)$ running time.*

Proof The running time of Step 1 comes from Lemma 8.17. In the rest of the algorithm we use the fact that $h \leq |V|$ and that we decrease h in every execution of Step 4, which requires $O(|V|)$ time to run by Theorem 8.6.

As we already mentioned, $\mathcal{H} + S_P$ is (k,ℓ)-redundant by Lemma 3.15, where S_P is a star on the vertices of P. Each step that we make in Step 4 maintains this property by Lemma 3.17, while also decreasing the number of edges in $S_P + F$ by one. Finally, we can conclude that after executing Algorithm 8.18 $\mathcal{H} + F$ is indeed (k,ℓ)-redundant and satisfies the optimality conditions of $\left\lceil \frac{|P|}{2} \right\rceil$. □

One can notice that Algorithms 8.8 and 8.16 use only $O(|V|)$ additional space. Hence Algorithm 8.18 may run using only $O(|V|)$ additional space, similarly to the (k,ℓ)-M-component algorithm from Section 8.2.

8.4 Algorithm for global rigidity augmentation

In this section we present an efficient implementation for the global rigidity augmentation problem (Problem 4) that realizes the optimum given by Theorem 5.6 in $O(|V|^2)$ running time. As we make some minor changes to the algorithm of Lemma 5.19 we shall present it in detail for the $0 < \ell \leq k$ case. We note that the case of $k < \ell \leq \frac{3}{2}k$ can be solved also in $O(|V|^2)$ time as shown in detail in [49]. Its main ideas and tools are identical to the ones presented here, however, significant effort is needed to understand and integrate the 3-connectivity structures. As those structures fell out of the scope of this dissertation we do not present them in detail here.

8.4.1 Efficient connectivity structures

The main difference, that arises in the $0 < \ell \leq k$ and $0 < k < \ell \leq \frac{3}{2}k$ cases is based on their different connectivity properties. Remember, we defined $c_{k,\ell} = \left\lceil \frac{\ell}{k} \right\rceil$ for positive k and ℓ. As presented in Chapter 5 we may always assume that G is not $(c_{k,\ell}+1)$-connected, as in that case Algorithms 8.3 and 8.16 with Algorithm 8.18 could provide an optimal solution in $O(|V|^2)$ time. We can check the $(c_{k,\ell}+1)$-connectivity of G in $O(|V|^2)$ time [8].

This, together with the algorithm from Section 8.2 gives rise to an interesting result.

Theorem 8.20 *One can decide in $O(|V|^2)$ running time if a graph G is globally rigid in $\mathbb{R}^2$.* □

Proof A graph G on at least four vertices is globally rigid in $\mathbb{R}^2$ if and only if it is 3-connected and $(2,3)$-M-connected by Theorem 2.3 and Lemma 4.4. (We note that this idea is already presented in [32].) Now we can check the 3-connectivity of G in $O(|V|^2)$ running time by [8], and we can get its $(2,3)$-M-component hypergraph and hence decide its $(2,3)$-M-connectivity in $O(|V|^2)$ running time by Theorem 8.6. Therefore, we can decide in $O(|V|^2)$ running time if G is globally rigid in $\mathbb{R}^2$. □

To add edges making graphs that are not globally rigid to globally rigid efficiently requires more work, though. As hinted in Section 5.2, we need to introduce new structures on which we can track the 2- and 3-ends, the min-cuts and the edge additions.

The structure of the 2-connected components of a connected graph. Let us first consider the case of $0 < \ell \leq k$ and hence $c_{k,\ell} = 1$.

From a connected but not 2-connected graph G we can obtain the so-called block-cut tree. This is a data structure, where we consider the 2-connected components (that is, the inclusion-wise maximal 2-connected subgraphs) and the cut-vertices of G as the nodes of a tree. There is an edge between a node representing a 2-connected component and a node representing a cut-vertex in G, if the cut-vertex is in the 2-connected component. Let us call the tree that forms this way **block-cut tree** (or **BC-tree**) and let us denote it by $\boldsymbol{BC(G)}$. The 2-ends of G correspond to the leaves of $BC(G)$. For a node c_v representing a cut-vertex v of G in $BC(G)$, the connected components of $BC(G)-c_v$ correspond to the connected components of $G-v$. For a graph $G=(V,E)$ we can build $BC(G)$ in $O(|V|+|E|)$ running time by using the algorithm of Hopcroft and Tarjan [28]. If we add a new edge to G then its BC-tree can be calculated in $O(|V|)$ time by the following result of Rosenthal and Goldner [66].

Lemma 8.21 *[66] Let $G=(V,E)$ be a connected but not 2-connected graph. Let $u,v \in V$ be two vertices which are not cut-vertices and let u' and v' be the vertices of $BC(G)$ which correspond to the 2-connected components containing u and v, respectively. Then $BC(G+uv) = BC(BC(G)+u'v')$.* □

The structure of the 3-connected components of a 2-connected graph. In the case of $k < \ell \leq \frac{3}{2}k$ and hence $c_{k,\ell} = 2$, we can use a similar tree-structure to the BC-tree to follow the 3-connected components of G. This structure is called the *SPQR-tree* defined by Di Battista and Tamassia [4, 5] and it is based on the triconnected components and 3-blocks of a 2-connected graph [74]. As the detailed analysis of the SPQR-trees exceeds the scope of this dissertation, let us just claim their main properties that we shall use later on. We note that a brief, yet complete description of the SPQR-trees can be found in [49].

Lemma 8.22 *(following [4, 25, 29, 49]) For a 2-connected graph $G=(V,E)$ which is not 3-connected, there exists a tree – denoted by $SPQR(G)$ – with $O(|V|)$ vertices, called the SPQR-tree of G, with the following properties. $SPQR(G)$ can be built in $O(|V|+|E|)$ running time. If we add an edge e to G, we can compute the SPQR-tree of $G+e$ with the use of $SPQR(G)$ in $O(|V|)$ time. The SPQR-tree supports the execution of the steps presented in Lemma 5.19, each in $O(|V|)$ running time.* □

8.4.2 Finding a transversal of atoms

Remember, we may assume that G is not $(c_{k,\ell}+1)$-connected. In this case, the atoms of G are pairwise disjoint by Lemma 5.8. In this subsection, we present how to find the transversals of the atoms of G in $O(|V|^2)$ running time. This set is a key input of the algorithm presented in Lemma 5.19, and also its modified version, when $0 < \ell \leq k$, see in Algorithm 8.24.

Lemma 8.23 *Let k and ℓ be positive integers such that $0 < \ell \leq \frac{3}{2}k$. Let $G=(V,E)$ be a (k,ℓ)-rigid graph on at least k^2+2 vertices that is not $(c_{k,\ell}+1)$-connected and is simple if $k < \ell$. Then one can find a transversal P of the atoms of G in $O(|V|^2)$ running time.*

Proof First, we find all the $(c_{k,\ell}+1)$-ends of G. We can obtain this by first computing $BC(G)$ or $SPQR(G)$ in $O(|V|^2)$ running time by the results of Subsection 8.4.1. The leaves of $BC(G)$ and $SPQR(G)$ correspond to the 2-ends and 3-ends, respectively [28, 25]. As P must intersect every (k,ℓ)-MCT set, we can use Algorithm 8.16 (with $O(|V|^2)$ running time by Lemma 8.17) on $\mathcal{H}_G$ and the set L which is the union of all $(c_{k,\ell}+1)$-ends.

If there is no edge uv so that $\mathcal{H}_G+uv$ is (k,ℓ)-redundant, then by Lemma 8.17 Algorithm 8.16 provides a transversal X of the (k,ℓ)-MCT sets in such a way that it intersects the most $(c_{k,\ell}+1)$-ends possible. After we obtained X in $O(|V|^2)$ time we can choose one vertex from each $(c_{k,\ell}+1)$-end which is not covered yet. Hence we can compute a transversal P of the atoms of G in $O(|V|^2)$ running time.

On the other hand, if there is an edge $u'v'$ so that $\mathcal{H}_G+u'v'$ is (k,ℓ)-redundant, then Algorithm 8.16 finds such an edge uv by Lemma 8.17. If there are any (k,ℓ)-MCT sets that are also atoms, they must contain either u or v by Observation 3.7. In fact, if there are two of them, then one of them must contain u while the other contains v. Given a transversal of the $(c_{k,\ell}+1)$-ends P_0 such that $|P_0\cap\{u,v\}|$ is as large as possible (that is, we chose u or v from a $(c_{k,\ell}+1)$-end if possible), either P_0, $P_0+\{u\}$, $P_0+\{v\}$ or $P_0+\{u,v\}$ is a transversal of the atoms. We can test each of them to determine the actual transversal. The algorithm for this is as follows.

Let X denote the actual vertex set of interest. If X is a transversal on the atoms of G, then given the edge set F of any connected graph on X, $\mathcal{H}_G+F$ is (k,ℓ)-redundant by Lemma 3.15. In contrast, if X does not contain a transversal of the atoms, then $\mathcal{H}_G+F$ clearly cannot be (k,ℓ)-redundant, as there is a (k,ℓ)-MCT set which is not intersected by $V(F)$ contradicting Observation 3.7. By Lemma 2.12 and Theorem 8.6, It can be checked in $O(|V|^2)$ time whether $\mathcal{H}_G+F$ is (k,ℓ)-redundant and hence we can find out in $O(|V|^2)$ time whether P_0, $P_0+\{u\}$, $P_0+\{v\}$ or $P_0+\{u,v\}$ is a transversal of the atoms of G in this case. □

8.4.3 Globally rigid augmentation algorithm for $0<\ell\le k$

To get an optimal augmenting edge set for Problem 4 in $O(|V|^2)$ running time, we need to use a slightly modified version of the algorithm of Lemma 5.19. In this subsection, we consider the case of $0<\ell\le k$ instead of $k<\ell\le\frac{3}{2}k$ that was presented in Subsection 5.3.1.

The input of the algorithm is a (k,ℓ)-rigid graph G that is not 2-connected. We noted that it is possible to check the 2-connectivity of a graph in $O(|V|+|E|)$ running time [28].

The other input is a transversal P of the atoms of G that we can get in $O(|V|^2)$ running time by Lemma 8.23. We also need the (k,ℓ)-M-component hypergraph of G, which is available in $O(|V|^2)$ time by Theorem 8.6.

As now we want to make a connected graph 2-connected, for the sake of simplicity, let us denote $b^1_{\{v\}}(G)$ by $b_v(G)$ and $b^1(G)$ by $b(G)$. Remember, we denote a star on the vertex set X by S_X. The algorithm will use the same set notation for N and F as the algorithm of Lemma 5.19, and we will maintain similar Properties, as in Subsection 5.3.1:

1. For an arbitrary star S_N on the vertex set N, $\mathcal{H}_G+F+S_N$ is a (k,ℓ)-redundant hypergraph.
2. In every 2-end of $G+F$, there is at least one vertex from N.
3. $\max\left\{b(G+F)-1, \left\lceil \frac{|N|}{2} \right\rceil\right\} + |F| = \max\left\{b(G)-1, \left\lceil \frac{|P|}{2} \right\rceil\right\}$.

Algorithm 8.24 ***Input:*** *A (k,ℓ)-redundant graph $G=(V,E)$ (where $0<\ell\le k$) that is not 2-connected, the (k,ℓ)-M-component hypergraph $\mathcal{H}_G$ of G, along with its orientation $\vec{\mathcal{H}}_G$ where the in-degree of each vertex is at most k, the block-cut tree $BC(G)$ of G, and a transversal P of the atoms of G.*

Output: *An edge set F, so that $G+F$ is (k,ℓ)-redundant and 2-connected, and $|F|$ matches the value of Theorem 5.6.*

$N := P$, $F := \emptyset$

While $|N| \ge \max\{4, b(G+F)+1\}$ ***do***

If $b(G+F)-1 \ge \left\lceil \frac{|N|}{2} \right\rceil$, ***then***

If *there is only one cut-vertex v such that $b_v(G+F)=b(G+F)$,* ***then***

Choose x_1, x_2 from a component of $G+F-\{v\}$ that contains at least two vertices from N. Let $x_3 \in N$ be a vertex from a component of $G+F-\{v\}$ that does not contain x_1 and x_2.

else

Let v_1 and v_2 be two cut-vertices for which $b_{v_1}(G+F)=b(G+F)=b_{v_2}(G+F)$. Choose $x_1, x_2 \in N$ from two different components of $G+F-\{v_1\}$ that do not contain v_2. Choose $x_3 \in N$ from a component of $G+F-\{v_2\}$ that does not contain v_1.

else

7 ***If*** *there is a cut-vertex v such that for one component of $G - \{v\}$, say K, $|N \cap K| \geq 2$ and $|N - K| \geq 2$,* ***then***

Choose x_1, x_2 from $N \cap K$ and choose x_3 from $N - K$.

8 ***else*** *(Notice that if $b(G+F) = 1$, then this is the only possible case.)*

Choose $x_1, x_2, x_3 \in N$ arbitrarily.

9 ***If*** *$\mathcal{H}_G + F + S(N - \{x_1, x_3\}) + x_1x_3$ is (k, ℓ)-redundant,* ***then***

$x := x_1, y := x_3$.

else

$x := x_2, y := x_3$.

10 *$F := F + \{xy\}$, $N := N - \{x, y\}$. Refresh $BC(G+F)$.*

11 *$F := F \cup S(N)$.* ***Return:*** *F.*

By the modification of Lemma 5.19 presented in Subsection 5.3.2, Algorithm 8.24 indeed returns an optimal edge set given by Theorem 5.6. The input of Algorithm 8.24 can be computed in $O(|V|^2)$ time by the results of Section 8.2, [28] and Lemma 8.23. Hence the only part needed to be proven is the $O(|V|^2)$ running time of Algorithm 8.24.

Theorem 8.25 *Let G be a simple (k, ℓ)-rigid graph, where $0 < \ell \leq k$. Then Algorithm 8.24 can be executed in $O(|V|^2)$ time.*

Proof While computing $BC(G)$ in $O(|V| + |E|)$ running time [28], let us also store $N \cap C$ with the corresponding node of $BC(G)$ for every 2-connected component C. As F is initialized to $\emptyset$ in Algorithm 8.24, at the start $BC(G+F) = BC(G)$. It is easy to verify that $b_v(G+F) = d_{BC(G+F)}(v)$ for any cut-vertex v of G. Hence computing all the values of $b_v(G+F)$ for all the cut-vertices takes linear time in $|V|$. From this, we can get $b(G+F)$ in $O(|V|)$ running time, as well, as finding the cut-vertex v for which $b_v(G+F) = b(G+F)$, or the (cut-)vertices u and v for which $b_v(G+F) = b(G+F) = b_u(G+F)$. With this and the stored vertices from $N \cap C$ we can find the vertices x_1, x_2 and x_3 in Steps **4** and **5** in $O(|V|)$ time. $BC(G)$ with the stored vertices from $N \cap C$ can help us find the appropriate K in Step **7** also in $O(|V|)$ time. By Theorem 8.3, we can compute $\mathcal{T}_{\mathcal{H}_G}(ij)$ in $O(|V|)$ running time thus executing Step **9** in $O(|V|)$ time.

Now, for STEP **10**, we need to refresh the BC-tree of $G+F$, as well, as the content of $N \cap C$. However, as $BC(G+e) = BC(BC(G)+e)$ holds for any edge e by Lemma 8.21, and $BC(G)$ has $O(|V|)$ edges, we can refresh both in $O(|V|)$ running time. Hence each execution of the core of the loop takes only $O(|V|)$ time. By the condition of STEP **2**, the fact that $|N| \leq |V|$ and that $|N|$ decreases with each execution of the loop, we can conclude that the whole algorithm takes at most $O(|V|^2)$ time to run. □

8.4.4 Global rigidity augmentation algorithm for $k < \ell \leq \frac{3}{2}k$

Let us now sketch the global rigidity augmentation algorithm for $k < \ell \leq \frac{3}{2}k$. For this, we shall use the algorithm of Lemma 5.19 as a subroutine, while we maintain $SPQR(G)$ with the help of Lemma 8.22. We note that a complete proof of this case with details on the SPQR-tree can be found in [49].

Algorithm 8.26 ***Input:*** *A (k,ℓ)-redundant graph $G=(V,E)$ (where $k < \ell \leq \frac{3}{2}k$) that is not 3-connected, the (k,ℓ)-M-component hypergraph $\mathcal{H}_G$ of G, along with its orientation $\vec{\mathcal{H}}_G$ where the in-degree of each vertex is at most k, the SPQR-cut tree $SPQR(G)$ of G, and a transversal P of the atoms of G.*

Output: *An edge set F, so that $G+F$ is (k,ℓ)-redundant and 3-connected, and $|F|$ matches the value of Theorem 5.6.*

$N := P$, $F := \emptyset$
While $|N| \geq \max\{4, b^2(G+F)+1\}$ ***do***
    Run the algorithm of Lemma 5.19. Results the vertices x and y.
    $F := F + \{xy\}$, $N := N - \{x,y\}$. *Refresh $SPRQ(G+F)$.*
$F := F \cup S(N)$. ***Return:*** F.

We can get the inputs of Algorithm 8.26 in $O(|V|^2)$ running time by the results of Section 8.2, [25] and Lemma 8.23, respectively.

By Lemma 5.19 and Theorem 5.6, Algorithm 8.26 results in an optimal solution for Problem 4 if $k < \ell \leq \frac{3}{2}k$. Let us prove its $O(|V|^2)$ running time.

Theorem 8.27 *Let G be a simple (k,ℓ)-rigid graph for $k < \ell \leq \frac{3}{2}k$, which is not 3-connected. Algorithm 8.26 can be executed in $O(|V|^2)$ running time.*

Proof (sketch) As the SPQR tree of $G+F$ always contains $O(|V|)$ edges in a tree structure, similarly to the block-cut tree in Theorem 8.25, we can compute $b^2(G+F)$ in $O(|V|)$ time in Step **2**. Lemma 8.22 states that Step **3** can be executed in $O(|V|)$ time. The same Lemma claims that the maintenance can take $O(|V|)$ time in Step **4**. Hence one execution of the core of the loop in Algorithm 8.26 takes $O(|V|)$ time.

As the $|P| \leq |V|$, and $|N|$ decreases with every execution of the loop, the whole running time of Algorithm 8.26 is $O(|V|^2)$. □

We note that the $(k,\ell) = (2,3)$ case, that is, the two-dimensional globally rigid augmentation of rigid graphs, is also included in this subcase. Moreover, any (k,ℓ)-rigid graph for $k < \ell \leq \frac{3}{2}k$ with no weak cut-pair can be handled with a slightly simpler approach. (Remember, as a consequence of Lemma 5.4 any $(2,3)$-rigid graph is such.) This is because in that case, the SPQR-tree contains no *polygons* (see definition in [25, 49]), hence its structure is simpler and there exist easier methods to compute it in $O(|V|^2)$ running time. As we did not expand the algorithms regarding the SPQR-tree, we do not elaborate on this special case, either.

8.5 Algorithmic solutions of pinning problems

We saw in Section 7.1 that the pinning problems are often closely related to the corresponding augmentation problems. Therefore, with the help of the algorithms for the augmentation problems, developing efficient algorithms for various pinning problems will be rather simple. Let us note that finding a smallest set that pins down an arbitrary graph to a rigid graph (see in Subsection 7.1.1) was already known to be solvable in $O(|V|^2)$ time [15].

Let us consider first the redundant rigidity pinning problem from Subsection 7.1.2. Remember, here we aim to find a minimum cardinality vertex set P that pins down a (k,ℓ)-tight hypergraph $\mathcal{H}$ to a (k,ℓ)-redundant hypergraph. Solving this requires finding a transversal of the (k,ℓ)-MCT sets of $\mathcal{H}$, which can be done in $O(|V|^2)$ running time by Lemma 8.17. Hence finding the optimal pinning set can also be achieved in $O(|V|^2)$ time.

A similar solution can be given to the global rigidity pinning problem from Subsection 7.1.3. Remember, in this version of the problem, we are given a rigid graph G and we need to find a minimum cardinality vertex set P, so that $G + K_P$ is $(2,3)$-redundant and 3-connected (that is, globally rigid in $\mathbb{R}^2$). As noticed in Subsection 7.1.3, in this case we need to pin down one vertex from each atom of G. Hence we need to find a transversal of the atoms of G. This can be done in $O(|V|^2)$ running time by Lemma 8.23 if G is not 3-connected, and in the same running time by Lemma 8.17 if G is 3-connected. Hence we can conclude the following theorem.

Theorem 8.28 *Given a $(2,3)$-rigid graph G, one can find a smallest vertex set P in $O(|V|^2)$ running time, so that $G + K_P$ is globally rigid in $\mathbb{R}^2$.* □

Now let us consider the two problems of pinning down a partially pinned graph to a globally rigid graph and the minimum cost global rigidity pinning problem. In both cases we are given a rigid graph G (which includes the complete graph on the already pinned vertices in the first case). If G is not 3-connected, or G is 3-connected but $\mathcal{H}_G$ cannot be made redundant with the help of a single edge, then the key in both problems is to find all the vertices in all the MCT sets of $\mathcal{H}_G$. This can be done in $O(|V|^2)$ time by the following method.

First, let us run the algorithm of Section 8.2. By Theorem 8.6 it can be executed in $O(|V|^2)$ time, and, as a byproduct, it returns a directed hypergraph $\vec{\mathcal{H}}_G$ that can be used to find $\mathcal{T}_{\mathcal{H}_G}(ij)$ in $O(|V|)$ running time for an arbitrary edge ij. Now we can find a transversal of the MCT sets of $\mathcal{H}_G$ in $O(|V|^2)$ time by Lemma 8.17. Let us choose one vertex from this transversal, say i. Obviously, i is a vertex from an MCT set, while all the other vertices of the transversal are vertices from other MCT sets. We can use a modified version of Algorithm 8.8. Starting from i, we calculate $\mathcal{T}(ij)$ for each vertex, not just the unmarked ones. By Theorem 8.6 this can be executed in sum $O(|V|^2)$ running time. For each such calculated subgraph $\mathcal{T}(ij)$, we check if any other vertex (say v) from the previously calculated transversal of the MCT sets is present in $V(\mathcal{T}(ij))$. Note that by Lemma 3.19 there can be only one such vertex v. If there is so, j is in the same MCT set as v. Note that with this method we get every vertex from all the MCT sets that are not containing i by Lemma 3.19. Hence, repeating this method from another vertex of the transversal we can get all the vertices from every MCT sets in $O(|V|^2)$ running time. Consequently, we can find the minimum cost vertex set that pins down a rigid graph to a globally rigid graph, or a tight hypergraph to a redundant hypergraph in $O(|V|^2)$ running time (provided that it has at least three pairwise disjoint MCT sets). Similarly, a partially pinned rigid graph can be pinned down to a globally rigid graph optimally in $O(|V|^2)$ running time, if G is not 3-connected, or G is 3-connected but $\mathcal{H}_G$ cannot be made redundant with the help of a single edge.

Let us now consider the cases, where G is 3-connected and $\mathcal{H}_G$ can be made redundant with the help of a single edge. We note that we can recognize such a case in $O(|V|^2)$ running time with Algorithm 8.16. The minimum cardinality vertex set that pins down a partially pinned rigid graph to a globally rigid graph of this type consists of at most two vertices, hence the actual optimum is a set of either one or two vertices. Algorithm 8.16 can provide us with two such vertices in $O(|V|^2)$ running time. We need to decide if there is a single vertex, the pinning of which makes $\mathcal{H}$ redundant. Obviously, in $O(|V|^2)$ running time we can decide for each vertex, if the pinning of it satisfies this requirement. Checking this property for each vertex one by one results an $O(|V|^3)$ running time algorithm for this special case.

The complexity of finding a minimum cost vertex set to pin down, so that it makes a rigid graph G globally rigid or a tight hypergraph $\mathcal{H}$ redundant if G is 3-connected and there is a single edge that augments $\mathcal{H}_G$ to a redundant hypergraph, and there is a single edge that augments $\mathcal{H}$ to a redundant hypergraph, respectively, is unknown.

8.6 Approximation of minimum cost globally rigid subgraphs

Remember, there are two completely different approaches to Problem 5 from Chapter 6 depending on whether $c \equiv 1$ with $(k, \ell) = (2, 3)$, or c is a metric cost function on a complete graph.

Let us first consider the solution given to the minimum size globally rigid spanning subgraph problem presented in Section 6.1. By Theorem 8.20 one can determine in $O(|V|^2)$ time if a graph is globally rigid in two dimensions. This observation is used, when we implement the näive method of greedily deleting edges while possible, so that G remains globally rigid. As edges that are not possible to delete at any given state of the graph cannot be deleted later either, this method results an $O(|V|^4)$ algorithm.

However, its running time can be sped up to $O(|V|^3)$ with some easy considerations. Let us first find a graph $(V, \bar{E}) = \bar{G} \subseteq G$, so that $\bar{G}$ is (2,3)-M-connected with 'few' edges. We can do this by running Algorithm 8.3 and storing the edges that do not satisfy the condition of Step 2. As shown in Claim 8.7 from Theorem 8.6, we add at most $O(|V|)$ edges to G^* during the whole algorithm and we also have at most $O(|V|)$ merges resulting $|\bar{E}| = O(|V|)$. We can find $\bar{G}$ in $O(|V|^2)$ time by Theorem 8.6.

Now for the 3-connectivity, we need one more tool, as follows. For any two distinct vertices $u, v \in V$, the **local connectivity** denoted by $\boldsymbol{\kappa_G(u, v)}$ is the maximum number of internally disjoint paths connecting u and v. By the famous theorem of Menger, the value of $\min\{\kappa_G(u, v) : u, v \in V\}$ is equal to the (vertex)-connectivity of G [59].

Theorem 8.29 *[8, Theorem 2.16] For a graph $G = (V, E)$ and a positive integer k we can construct a subgraph $G' = (V, E')$ of G with $|E'| \leq k(|V|-1)$ in $O(k(|E|+|V|))$ running time, so that for any two distinct vertices $u, v \in V$ the local connectivity $\kappa_{G'}(u, v) = \min(k, \kappa_G(u, v))$.* $\square$

Theorem 8.29 achieves the $O(k(|V|+|E|))$ running time with the use of k *scan first searches*.

Now let us use the sparse 3-connectivity certificate $G' = (V, E')$, provided by Theorem 8.29 in $O(|V|^2)$ time. We construct the graph $\tilde{G} = (V, \bar{E} \cup E')$, which is a globally rigid subgraph of G with at most $O(|V|)$ edges that we can get in $O(|V|^2)$ time. If we run the näive algorithm on $\tilde{G}$ instead of G (that is, deleting edges from $\tilde{G}$, while $\tilde{G}$ is still globally rigid), the running time decreases to $O(|V|^3)$. The graph that remains after the conclusion of the näive algorithm on $\tilde{G}$ is a $\frac{3}{2}$-approximation of the minimum size globally rigid subgraph by Lemma 6.1, as it is globally rigid, and indeed minimal.

In the algorithmic approximation of the minimum cost globally rigid spanning subgraph problem – presented in Section 6.2 – the main difficulty lies in determining a minimum cost (k, ℓ)-tight subgraph of the weighted graph $G = (V, E)$. We discussed this problem in detail in Subsection 7.6.7. Let us sketch here a possible solution for it with $O(|V|^3)$ running time. First, we modify the order of the edges, so that we process them in a cost increasing order. Using Algorithm 8.3, one can check, if the next edge in the order is independent in the (k, ℓ)-sparsity matroid in $O(|V|)$ running time by Theorem 8.6. This cummulates to $O(|V|^3)$ running time over the possible $O(|V|^2)$ edges.

As we presented in Subsection 7.6.7, it is still open whether we can find a minimum cost (k, ℓ)-tight subgraph of a general graph with a general cost function in $O(|V|^2)$ running time. Nonetheless, I would not rule out that one could show an algorithm for finding a minimum cost (k, ℓ)-tight subgraph of G faster than $O(|V|^3)$. For example, if $\ell \leq k$, the data structure from [55] solves the problem in $O(|V|^2)$ running time. We also hope that the open problem of Subsection 7.6.7 gets a positive answer soon. Hence for the sake of flexibility, let us denote the running time of the minimum cost (k, ℓ)-tight subgraph problem by $O(\tilde{C})$. It is not impossible either that the metric cost function or the fact that G is a complete graph may help in developing an $O(|V|^2)$ running time algorithm for this specific case.

The last step depends on the minimum cost (k, ℓ)-tight subgraph G^* of G. In case G^* is 3-connected and $\mathcal{H}_G$ can be made (k, ℓ)-redundant with the help of one edge, this edge can be found in $O(|V|^2)$ running time by the results of Section 8.3. If, on the other hand, G^* is either not 3-connected, or there is no single edge that makes $\mathcal{H}_G$ (k,ℓ)-redundant, we can find a transversal P of the atoms of G^* in $O(|V|^2)$ time by Lemma 8.23. Now, we can determine a minimum cost spanning tree on P in $O(|V|^2)$ time. In both cases, the result is a constant factor approximation by Theorem 6.4. The running time can be bounded by $O(|V|^2 + \tilde{C})$, which is clearly at most $O(|V|^3)$.

Bibliography

[1] T. G. Abbot. Generalizations of Kempe's universality theorem. Master's thesis, MIT, 2008. http://web.mit.edu/tabbott/www/papers/mthesis.pdf.

[2] B. D. O. Anderson, I. Shames, G. Mao, and B. Fidan. Formal theory of noisy sensor network localization. *SIAM J. Discrete Math.*, 24:684–698, 01 2010. https://doi.org/10.1137/100792366.

[3] L. Asimow and B. Roth. The rigidity of graphs. *Transactions of the American Mathematical Society*, 245:279–289, 1978. https://doi.org/10.1090/S0002-9947-1978-0511410-9.

[4] G. Di Battista and R. Tamassia. On-line graph algorithms with SPQR-trees. In M.S. Paterson, editor, *Automata, Languages and Programming*, pages 598–611, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0032061.

[5] G. Di Battista and R. Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15(4):302–318, 1996. https://doi.org/10.1007/BF01961541.

[6] A. R. Berg. *Rigidity of Frameworks and Connectivity of Graphs*. PhD thesis, Aarhus University, Denmark, 2004.

[7] A. R. Berg and T. Jordán. Algorithms for graph rigidity and scene analysis. In G. Di Battista and U. Zwick, editors, *Algorithms - ESA, 11th Annual European Symposium, Proceedings*, volume 2832 of *Lecture Notes in Computer Science*, pages 78–89. Springer, 2003. https://doi.org/10.1007/978-3-540-39658-1_10.

[8] J. Cheriyan, M.-Y. Kao, and R. Thurimella. Scan-first search and sparse certificates: An improved parallel algorithm for k-vertex connectivity. *SIAM Journal on Computing*, 22:157–174, 02 1993. https://doi.org/10.1137/0222013.

[9] F. Chung and R. Graham. A new bound for Euclidean Steiner minimal trees. *Annals of the New York Academy of Sciences*, 440:328–346, 12 2006. https://doi.org/10.1111/j.1749-6632.1985.tb14564.x.

[10] R. Connelly. Generic global rigidity. *Discrete & Computational Geometry*, 33(4):549–563, 2005. https://doi.org/10.1007/s00454-004-1124-4.

[11] R. Connelly, T. Jordán, and W. Whiteley. Generic global rigidity of body-bar frameworks. *J. Comb. Theory, Ser. B*, 103(6):689–705, 2013. https://doi.org/10.1016/j.jctb.2013.09.002.

[12] R. Connelly and W. Whiteley. Second-order rigidity and prestress stability for tensegrity frameworks. *SIAM J. Discrete Math.*, 9(3):453–491, 1996. https://doi.org/10.1137/S0895480192229236.

[13] R. Connelly and W. Whiteley. Global rigidity: The effect of coning. *Discrete & Computational Geometry*, 43(4):717–735, 2010. https://doi.org/10.1007/s00454-009-9220-0.

[14] K. P. Eswaran and R. E. Tarjan. Augmentation problems. *SIAM Journal on Computing*, 5(4):653–665, 1976. https://doi.org/10.1137/0205044.

[15] Zs. Fekete. Source location with rigidity and tree packing requirements. *Operations Research Letters*, 34(6):607–612, 2006. https://doi.org/10.1016/j.orl.2005.10.005.

[16] Zs. Fekete and T. Jordán. Uniquely localizable networks with few anchors. In S.E. Nikoletseas and J.D.P. Rolim, editors, *Algorithmic Aspects of Wireless Sensor Networks*, pages 176–183, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. https://doi.org/10.1007/11963271_16.

[17] A. Frank. *Connections in Combinatorial Optimization.* Oxford University Press, 2011.

[18] A. Frank and T. Jordán. Graph connectivity augmentation. In A. Brandstädt K.T. Thulasiraman, S. Arumugam and T. Nishizeki, editors, *Handbook of Graph Theory, Combinatorial Optimization, and Algorithms*, CRC Computer and Information Science Series, pages 315–348. Chapman and Hall, 2016.

[19] A. Frank and T. Király. Combined connectivity augmentation and orientation problems. *Discrete Appl. Math.*, 131(2):401–419, 2003. https://doi.org/10.1016/S0166-218X(02)00460-2.

[20] G. N. Frederickson and J. Ja'ja'. On the relationship between the biconnectivity augmentation and travelling salesman problems. *Theoretical Computer Science*, 19(2):189–201, 1982. https://doi.org/10.1016/0304-3975(82)90059-7.

[21] D. Garamvölgyi, T. Jordán, and Cs. Király. Count and cofactor matroids of highly connected graphs. Technical Report TR-2022-12, Egerváry Research Group, Budapest, 2022. egres.elte.hu.

[22] A. García and J. Tejel. Augmenting the rigidity of a graph in $\mathbb{R}^2$. *Algorithmica*, 59(2):145–168, 2011. https://doi.org/10.1007/s00453-009-9300-9.

[23] S. J. Gortler, A. D. Healy, and D. P. Thurston. Characterizing generic global rigidity. *American Journal of Mathematics*, 132(4):897–939, 2010. https://doi.org/10.1353/ajm.0.0132.

[24] J. E. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity.* AMS Graduate studies in mathematics Vol. 2. American Mathematical Soc., 1993. https://doi.org/10.1090/gsm/002.

[25] Carsten Gutwenger and Petra Mutzel. A linear time implementation of spqr-trees. In Joe Marks, editor, *Graph Drawing*, pages 77–90, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44541-2_8.

[26] S. L. Hakimi. On the degrees of the vertices of a directed graph. *J. Franklin Inst.*, 279(4):290–308, 1969. https://doi.org/10.1016/0016-0032(65)90340-6.

[27] B. Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21(1):65–84, 1992. https://doi.org/10.1137/0221008.

[28] J. E. Hopcroft and R. E. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, jun 1973. https://doi.org/10.1145/362248.362272.

[29] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973. https://doi.org/10.1137/0202012.

[30] T. S. Hsu and V. Ramachandran. A linear time algorithm for triconnectivity augmentation. *Annual Symposium on Foundations of Computer Science (Proceedings)*, pages 548–559, 1991. https://doi.org/10.1109/SFCS.1991.185418.

[31] C. Hunkenschröder, S. Vempala, and A. Vetta. A 4/3-approximation algorithm for the minimum 2-edge connected subgraph problem. *ACM Trans. Algorithms*, 15(4):1–28, October 2019. https://doi.org/10.1145/3341599.

[32] B. Jackson and T. Jordán. Connected rigidity matroids and unique realizations of graphs. *J. Comb. Theory, Ser. B*, 94:1–29, 2005. https://doi.org/10.1016/j.jctb.2004.11.002.

[33] B. Jackson and T. Jordán. Brick partitions of graphs. *Discrete Mathematics*, 310(2):270–275, 2010. https://doi.org/10.1016/j.disc.2008.09.034.

[34] B. Jackson and T. Jordán. The generic rank of body-bar-and-hinge frameworks. *Eur. J. Comb.*, 31(2):574–588, 2010. https://doi.org/10.1016/j.ejc.2009.03.030.

[35] B. Jackson and A. Nixon. Global rigidity of generic frameworks on the cylinder. *J. Comb. Theory, Ser. B*, 139:193–229, 2019. https://doi.org/10.1016/j.jctb.2019.03.002.

[36] D. J. Jacobs and B. Hendrickson. An algorithm for two dimensional rigidity percolation: The pebble game. *Journal of Computational Physics*, 137:346–365, 1997. https://doi.org/10.1006/jcph.1997.5809.

[37] T. Jordán. Optimal and almost optimal algorithms for connectivity augmentation problems. In Giovanni Rinaldi and Laurence A. Wolsey, editors, *Proceedings of the 3rd Integer Programming and Combinatorial Optimization Conference, Erice, Italy, April 29 - May 1, 1993*, pages 75–88. CIACO, 1993.

[38] T. Jordán. On the optimal vertex-connectivity augmentation. *J. Comb. Theory, Ser. B*, 63:8–20, 1995. https://doi.org/10.1006/jctb.1995.1002.

[39] T. Jordán. Rigid and globally rigid graphs with pinned vertices. In A. Schrijver Gy. O. H. Katona and T. Szőnyi, editors, *Fete of Combinatorics and Computer Science*, volume 20 of *Bolyai Society Mathematical Studies*, pages 151–172. Springer, 2010. https://doi.org/10.1007/978-3-642-13580-4_7.

[40] T. Jordán. Combinatorial rigidity: Graphs and matroids in the theory of rigid frameworks. In *Discrete Geometric Analysis*, volume 34 of *MSJ Memoirs*, pages 33–112. Mathematical Society of Japan, Japan, 2016. https://doi.org/10.2969/msjmemoirs/03401C020.

[41] T. Jordán, Cs. Király, and S. Tanigawa. Generic global rigidity of body-hinge frameworks. *J. of Comb. Theory, Ser. B*, 117:59 – 76, 2016. https://doi.org/10.1016/j.jctb.2015.11.003.

[42] T. Jordán and A. Mihálykó. Minimum cost globally rigid subgraphs. In I. Bárány, Gy. O. H. Katona, and A. Sali, editors, *Building Bridges II*, volume 28 of *Bolyai Society Mathematical Studies*, pages 257–278. Springer Berlin Heidelberg, 2019. https://doi.org/10.1007/978-3-662-59204-5_8.

[43] P. Kaewprapha, J. Li, and N. Puttarak. Network localization on unit disk graphs. *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pages 1–5, 2011. https://doi.org/10.1109/GLOCOM.2011.6134494.

[44] Cs. Király. Rigid graphs and an augmentation problem. Technical Report TR-2015-03, Egerváry Research Group, Budapest, 2015. egres.elte.hu.

[45] Cs. Király and A. Mihálykó. Sparse graphs and an augmentation problem. Technical Report TR-2020-06, Egerváry Research Group, Budapest, 2020. egres.elte.hu, Preliminary version of [51].

[46] Cs. Király and A. Mihálykó. Sparse graphs and an augmentation problem. In Daniel Bienstock and Giacomo Zambelli, editors, *Integer Programming and Combinatorial Optimization*, pages 238–251, Cham, 2020. Springer International Publishing. Conference version of [51]. https://doi.org/10.1007/978-3-030-45771-6_19.

[47] Cs. Király and A. Mihálykó. Globally rigid augmentation of minimally rigid graphs in $\mathbb{R}^2$. In Tiziana Calamoneri and Federico Corò, editors, *Algorithms and Complexity*, pages 326–339, Cham, 2021. Springer International Publishing. Conference version of [50]. https://doi.org/10.1007/978-3-030-75242-2_23.

[48] Cs. Király and A. Mihálykó. Globally rigid augmentation of rigid graphs. Technical Report TR-2021-04, Egerváry Research Group, Budapest, 2021. egres.elte.hu, Preliminary version of [50].

[49] Cs. Király and A. Mihálykó. Fast algorithms for sparsity matroids and the global rigidity augmentation problem. Technical Report TR-2022-05, Egerváry Research Group, Budapest, 2022+. to appear, egres.elte.hu.

[50] Cs. Király and A. Mihálykó. Globally rigid augmentation of rigid graphs. *SIAM Journal on Discrete Mathematics*, 36(4):2473–2496, 2022. https://doi.org/10.1137/21M1432417.

[51] Cs. Király and A. Mihálykó. Sparse graphs and an augmentation problem. *Math. Program.*, 192(1):119–148, 2022. https://doi.org/10.1007/s10107-021-01689-0.

[52] G. Kortsarz and Z. Nutov. Approximating node connectivity problems via set covers. In Klaus Jansen and Samir Khuller, editors, *Approximation Algorithms for Combinatorial Optimization*, pages 194–205, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44436-X_20.

[53] G. Laman. On graphs and rigidity of plane skeletal structures. *J. Engineering Mathematics*, 4:331–340, 1970. https://doi.org/10.1007/BF01534980.

[54] A. Lee and I. Streinu. Pebble game algorithms and sparse graphs. *Discrete Mathematics*, 308(8):1425–1437, 2008. https://doi.org/10.1016/j.disc.2007.07.104.

[55] A. Lee, I. Streinu, and L. Theran. Finding and maintaining rigid components. In *Proceedings of the 17th Canadian Conference on Computational Geometry, CCCG'05, University of Windsor, Ontario, Canada, August 10-12, 2005*, pages 219–222, 2005.

[56] M. Lorea. On matroidal families. *Discrete Mathematics*, 28(1):103–106, 1979. https://doi.org/10.1016/0012-365X(79)90190-0.

[57] L. Lovász and Y. Yemini. On generic rigidity in the plane. *SIAM Journal on Algebraic and Discrete Methods*, 3:91–98, 1982. https://doi.org/10.1137/0603009.

[58] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, September 1994. https://doi.org/10.1145/185675.306789.

[59] K. Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927. https://doi.org/10.4064/FM-10-1-96-115.

[60] A. Mihálykó. On count matroids. *Alkalmazott Matematikai Lapok*, 38:141–153, 2021. (in Hungarian, with English abstract). https://doi.org/10.37070/AML.2021.38.1.10.

[61] A. Mihálykó. A new approach to redundantly rigid augmentations. In *Proceedings of the 11th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, HJ2019, University of Tokyo, Tokyo, Japan, May 27-30, 2019*, pages 277–285, 2019.

[62] C. St. J. A. Nash-Williams. Decomposition of finite graphs into forests. *J. London Math. Soc.*, 39:12, 1961. https://doi.org/10.1112/jlms/s1-39.1.12.

[63] A. Nixon, J. C. Owen, and S. C. Power. Rigidity of frameworks supported on surfaces. *SIAM Journal on Discrete Mathematics*, 26(4):1733–1757, 2012. https://doi.org/10.1137/110848852.

[64] Zeev Nutov. A $4 + \varepsilon$ approximation for k-connected subgraphs. *Journal of Computer and System Sciences*, 123:64–75, 2022. https://doi.org/10.1016/j.jcss.2021.07.006.

[65] H. Pollaczek-Geiringer. Über die Gliederung ebener Fachwerke. *ZAMM - Journal of Applied Mathematics and Mechanics*, 7(1):58–72, 1927. https://doi.org/10.1002/zamm.19270070107.

[66] A. Rosenthal and A. Goldner. Smallest augmentations to biconnect a graph. *SIAM Journal on Computing*, 6(1):55–66, 1977. https://doi.org/10.1137/0206003.

[67] J.B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. Technical report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA, 1979.

[68] A. So and Y. Ye. Theory of semidefinite programming for Sensor Network Localization. *Mathematical Programming*, 109:405–414, 01 2005. https://doi.org/10.1007/s10107-006-0040-1.

[69] I. Streinu and L. Theran. Sparse hypergraphs and pebble game algorithms. *European Journal of Combinatorics*, 30(8):1944–1964, 2009. https://doi.org/10.1016/j.ejc.2008.12.018.

[70] T.-S. Tay. Rigidity of multi-graphs I: Linking rigid bodies in n-space. *J. Comb. Theory, Ser. B*, 36(1):95–112, 1984. https://doi.org/10.1016/0095-8956(84)90016-9.

[71] T.-S. Tay. Linking $(n-2)$-dimensional panels in n-space II: $(n-2,2)$-frameworks and body and hinge structures. *Graphs and Combinatorics*, 5(1):245–273, 1989. https://doi.org/10.1007/BF01788678.

[72] T.-S. Tay. Henneberg's method for bar and body frameworks. *Structural Topology*, 17:53–58, 1991.

[73] T.-S. Tay. Linking $(n-2)$-dimensional panels in n-space I: $(k-1,k)$-graphs and $(k-1,k)$-frames. *Graphs and Combinatorics*, 7(3):289–304, 1991. https://doi.org/10.1007/BF01787636.

[74] W. T. Tutte. *Connectivity in graphs*, volume 15 of *Mathematical Expositions*. University of Toronto Press, 1966. https://doi.org/10.3138/9781487584863.

[75] S. Vempala and A. Vetta. Factor 4/3 approximations for minimum 2-connected subgraphs. In Klaus Jansen and Samir Khuller, editors, *Approximation Algorithms for Combinatorial Optimization*, pages 262–273, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44436-X_26.

[76] W. Whiteley. Cones, infinity and one-story buildings. *Structural Topology*, 8:53–70, 1983.

[77] W. Whiteley. The union of matroids and the rigidity of frameworks. *SIAM J. Discrete Math.*, 1(2):237–255, 1988. https://doi.org/10.1137/0401025.

[78] W. Whiteley. Some matroids from discrete applied geometry. In J. E. Bonin, J. G. Oxley, and B. Servatius, editors, *Matroid Theory*, volume 197 of *Contemporary Mathematics*, pages 171–311. AMS, 1996. https://doi.org/10.1090/conm/197/02540.

[79] W. Whiteley. Rigidity and scene analysis. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry (2nd edition)*, pages 1327–1354. CRC Press, Boca Raton, 2004. https://doi.org/10.1201/9781420035315.ch60.

Appendix

In this Appendix we present the details and the results of the implementation of some of the algorithms from Chapter 8. These algorithms are the following:

- The algorithm presented in Section 8.2. That is, computing the (k, ℓ)-M-component hypergraph of an input graph G for any $0 < \ell < 2k$.
- The algorithms presented in Section 8.3. That is, given a (k, ℓ)-tight (k, ℓ)-M-component hypergraph $\mathcal{H}$ with $0 < \ell < 2k$, the program returns a minimum size edge-set F so that $\mathcal{H} + F$ is (k, ℓ)-redundant. This includes Algorithms 8.8, 8.16 and 8.18.

The implementation faithfully follows the structure of the corresponding Sections and references them in comments frequently. The source code and the running instructions are available at `https://github.com/mihalykoandras/rigidityAugmentations`.

Running time and spatial complexity analysis

The code is an efficient solution of the aforementioned problems with reaching the $O(|V|^2)$ running time and using only $O(|V|)$ extra space. According to the results of Sections 8.2 and 8.3, the program only solves the problems with input graphs on at least k^2+2 vertices, however, it returns a set of edges even for smaller graphs – with no guarantee of correctness.

The correctness of the program was tested on several (k, ℓ) values and multiple input graphs. We would like to present one specific type of graph in which the correctness of the algorithm is quite easy to verify. We shall also present the detailed running time on this graph class later.

Consider the following procedure. We generate three random trees on the same vertex set and denote their union by T. Now for every vertex v in T we replace v with a complete graph of $d_T(v)+1$ vertices, called *bodies*, so that each vertex of these bodies has at most one edge that

Number of vertices (thousand)	Running time (in seconds)	Used memory (in Kb)
1	0.03	4912
5	0.64	10532
10	4.08	17480
20	29.12	32400
30	72.89	45920
40	151.51	60116
50	243.46	74296
60	348.26	88660
70	501.64	104064
80	664.48	119536
90	841.13	132056
100	1053.97	145108
110	1298.5	159124
120	1535.79	173920
130	1826.83	187312
140	2204.95	205188
150	2437.37	216496
160	2833.49	232900
170	3158.55	246812

Table 8.3: Running time and memory consumption on large sparse $(2,3)$-rigid graphs. Notice that the number of vertices is given in thousands.

is not spanned by the body itself. Let us denote this new graph by G. Clearly, the bodies of G are $(2,3)$-rigid, as each $v \in T$ has degree of at least three. Now, by the results of Tay [70], G is a $(2,3)$-rigid graph. Moreover, the bodies of G represent the nontrivial $(2,3)$-M-components of G and the edges connecting these parts represent the trivial M-components. This property allows us to check the results of the M-component hypergraph constructor algorithm (Algorithm 8.3). The fact that after running the algorithms of Section 8.3 on this M-component hypergraph we get a $(2,3)$-redundant hypergraph reaching the optimum stated in Theorem 3.1 shows the correctness of the results found by the implementation.

In Table 8.3, we provide some details of the running time and memory usage regarding the tests on graphs of the above introduced type. These results are visualized in Figure 8.4. As one can clearly see, the running time is in fact $O(|V|^2)$, with $O(|V|)$ extra memory used.

For even more insights, we repeat the experiments on similarly generated graphs. In this second case, however, we construct the original trees so that they have a vertex with degree $O(|V|)$. In this way, the input graphs have $O(|V|^2)$ edges. In fact, all of these graphs have a number of edges between $0.064|V|^2$ and $0.067|V|^2$. Even though this ratio seems small, notice that in the largest case, with more than 20000 vertices, this means nearly 30 million edges –

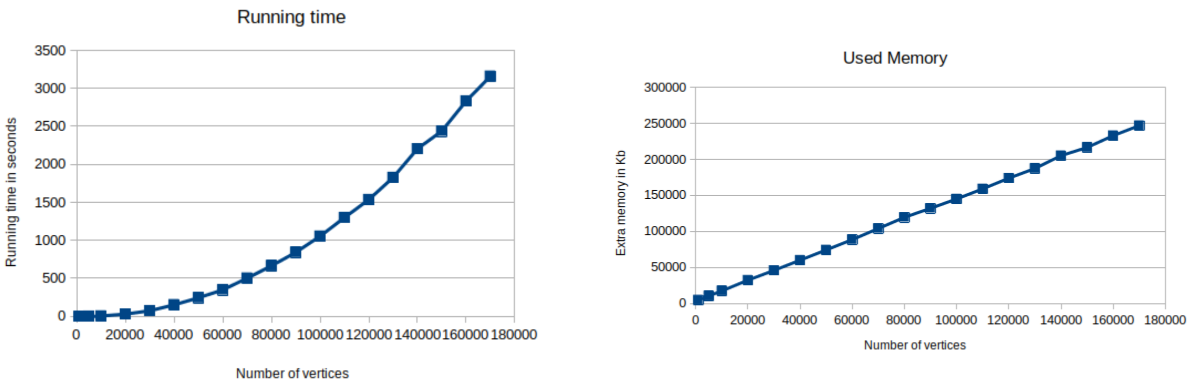


Figure 8.4: Running time and memory consumption of the instances presented in Table 8.3. The $O(|V|^2)$ running time and the $O(|V|)$ extra space both can be seen clearly.

which is comparable in size to the Western European road network graph. Even to store a graph of this size consumes a significant amount of space. The results of these runs are shown in Table 8.5. Again, it is possible to confirm that the running time is in fact $O(|V|^2)$.

Number of vertices	Running time (in seconds)
1674	0.01
2514	0.88
3354	1.49
4194	2.42
6294	7.48
8394	10.59
10494	19.07
12594	30.04
14694	47.73
16794	68.00
18894	88.49
20944	119.28

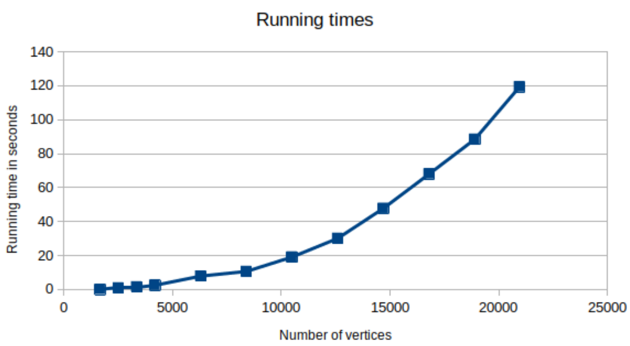


Table 8.5: Running time with graphs that have $O(|V|^2)$ edges. The peculiar number of vertices comes from the method of generation. In this case, there is no use of analysing the memory, as the storage used by the input graph dominates the total memory footprint. The slight decrease in performance compared to its sparse counterpart is not surprising, but we can observe the $O(|V|^2)$ running time here, as well. Larger tests were made impossible by memory constraints.

Technical details

The code is written in C++14 and it is tested with gcc 9.3.0 compiler. All the tests were run with a computer of the following specifications:

- **CPU:** Intel Core i3-9100F, 4 cores

- **Frequency:** 4100 MHz
- **Memory:** 16 Gb
- **Operations system:** Ubuntu 20.04.4 LTS

Summary

The dissertation investigates several topics regarding count matroids and (k,ℓ)-rigidity with special attention to the two-dimensional combinatorial rigidity. These topics are as follows:

- **Augmenting (k,ℓ)-rigidity** *Given a minimally rigid graph in two dimensions, what is the minimum cardinality edge set to add so that the graph becomes redundantly rigid?*

  This question was first considered by García and Tejel [22], who presented an $O(|V|^2)$ algorithm to give an optimal solution. Later Király generalized their method to every (k,ℓ)-tight graph, when $0 < \ell \leq \frac{3}{2}k$ [45]. We present **new structural results** on (k,ℓ)-tight graphs with the use of the (k,ℓ)-co-tight sets. With these we give a **min-max theorem** for the optimal edge set for every (k,ℓ)-tight graph when $\ell < 2k$. All these results can be generalized to (k,ℓ)-tight **hypergraphs**, as well.

  We also prove that the problem is **NP-hard** for general **(k,ℓ)-rigid graphs** but can be solved in polynomial time if $\ell \leq k$.

- **Globally rigid augmentation of rigid graphs.** *Given a rigid graph in $\mathbb{R}^2$, what is the minimum cardinality edge set to add so that the graph becomes globally rigid?*

  By Jackson and Jordán [32] we know that globally rigid graphs are the redundantly rigid and 3-connected graphs. With the use of (k,ℓ)-co-tight sets and 3-ends we conclude **new structural elements** of rigid graphs called atoms, with which we form a **min-max theorem** for global rigidity augmentation. This theory can be generalized for every (k,ℓ)-rigid graph where $0 < \ell \leq \frac{3}{2}k$. The structural results can also solve the **global rigidity pinning problem**. For both problems, 2-approximations are achievable if the input is not rigid.

- **Minimal cost globally rigid spanning subgraph.** *Given a complete graph with a cost function on its edges, give a minimum cost globally rigid spanning subgraph of it.*

  As the complete graph is globally rigid, there exists a minimum cost globally rigid spanning subgraph of it. We prove the **NP-hardness** of the problem, even if the cost function is metric. However, with metric cost function we give a **2-approximation** algorithm that

results a globally rigid graph. The approximation factor is improved to **1.61** in the case of **Euclidean cost function**. The algorithm can be generalized to other (k, ℓ)-values also resulting constant factor approximations.

- **Rigidity and global rigidity algorithms.** We gave **algorithms with** $\boldsymbol{O(|V|^2)}$ running time for versions of the **redundant and the global rigidity augmentation** problems presented above and we also provided an $O(|V|^2)$ algorithm to determine the $\boldsymbol{(k, \ell)}$**-M-component hypergraph** of any graph. For the minimum cost globally rigid subgraph problem our approximation algorithm achieves $O(|V|^3)$ running time. We accompanied some of our algorithms with implementation in C++.

The dissertation is based on the following papers and publications (in order): [42, 61, 46, 60, 51, 47, 48, 50, 49].

Összefoglalás

Jelen disszertáció a ritkasági matroidok és (k,ℓ)-ritkaság témaköréből vizsgál számos problémát, megkülönböztetett figyelemmel a kétdimenziós kombinatorikus merevségben előforduló kérdésekre. A főbb témakörök a következők:

- **$(\boldsymbol{k},\boldsymbol{\ell})$-redundánssá növelés.** *Adott egy minimálisan merev gráf kétdimenzióban, találjunk olyan minimális méretű élhalmazt, amely redundánsan merevvé növeli.*

  A kérdést először García és Tejel [22] vizsgálta, és az optimális élhalmaz megtalálására egy $O(|V|^2)$ futásidejű algoritmust adtak. Módszerüket később Király általánosította minden olyan (k,ℓ)-kritikus gráfra, ahol $0 < \ell \leq \frac{3}{2}k$ [45]. Ezen dolgozatban a (k,ℓ)-kritikus gráfokon bemutatott **új strukturális eredmények** segítségével, a (k,ℓ)-ko-kritikus halmazokat felhasználva egy **min-max tételt** mutatunk az optimális növelő halmaz élszámára (k,ℓ)-kritikus gráfok esetén, ha $\ell < 2k$. Az eredményeket (k,ℓ)-kritikus **hipergráfokra** is általánosítottuk.

  Bizonyítottuk továbbá, hogy a probléma **NP-nehéz** általános **$(\boldsymbol{k},\boldsymbol{\ell})$-merev gráfok** esetén, míg ha $\ell \leq k$, polinomiális időben megoldható merev bemenetre is.

- **Merev gráfok globálisan merevvé növelése.** *Adott egy merev gráf kétdimenzióban, találjunk olyan minimális méretű élhalmazt, amely globálisan merevvé növeli.*

  Jackson és Jordán [32] eredménye alapján tudjuk, hogy a globálisan merev gráfok kétdimenzióban a redundánsan merev és 3-összefüggő gráfok. Felhasználva ismereteinket a ko-kritikus halmazokról és 3-végekről, a merev gráfok **új strukturális tulajdonságait** mutatjuk be, amelyek segítségével egy **min-max tételt** tudunk adni a globálisan merev növelésre. Ezen eredményket általánosítottuk (k,ℓ)-merev gráfokra is, ha $0 < \ell \leq \frac{3}{2}k$. A bemutatott strukturális tulajdonságok egyben a **globálisan merev leszúrási** probléma optimális megoldásához is elvezetnek merev gráfok esetén. Általános (nem merev) gráfokra mindkét problémára egy 2-közelítő algoritmus érhető el.

- **Minimális költségű globálisan merev feszítő részgráf.** *Adott egy teljes gráf és egy költségfüggvény az élein, adjunk minimális költségű globálisan merev feszítő részgráfját.*

Mivel a teljes gráf globálisan merev, létezik minimális költségű globálisan merev feszítő részgráfja. Megmutatjuk a probléma **NP-nehézségét**, ami még olyan megszorítás mellett is fennáll, ha a költségfüggvény metrikus, vagyis teljesíti a háromszög-egyenlőtlenséget. Metrikus költségfüggvény mellett viszont adunk egy **2-közelítő algoritmust**, amely globálisan merev gráfot eredményez. Az approximációs faktor tovább javítható **1.61**-re euklideszi költségek mellett. Az algoritmust más (k, ℓ) párokra is általánosítjuk, így ismét konstans faktorú közelítő algoritmust kapva.

- **Merevségi és globális merevségi algoritmusok.** A tézisben $\boldsymbol{O(|V|^2)}$ **futásidejű algoritmust** adunk a **redundáns és a globálisan merev növelési** probléma előbbiekben bemutatott verzióira. Továbbá mutatunk egy $O(|V|^2)$ algoritmust, amely meghatározza egy tetszőleges gráf $\boldsymbol{(k, \ell)}$**-M-komponens hipergráfját**. A minimális költségű globálisan merev feszítő részgráf feladatra adott algoritmusunk $O(|V|^3)$ futásidejű. Néhány bemutatott algoritmust C++-ban implementáltunk.

A disszertáció a következő publikációk alapján készült (időrendben): [42, 61, 46, 60, 51, 47, 48, 50, 49].