<table>
<thead>
<tr>
<th>Title</th>
<th>Combinatorial Rigidity and Generation of Discrete Structures (Dissertation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Tanigawa, Shinichi</td>
</tr>
<tr>
<td>Citation</td>
<td>Kyoto University (京都大学)</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2010-03-23</td>
</tr>
<tr>
<td>URL</td>
<td><a href="https://doi.org/10.14989/doctor.k15317">https://doi.org/10.14989/doctor.k15317</a></td>
</tr>
<tr>
<td>Type</td>
<td>Thesis or Dissertation</td>
</tr>
<tr>
<td>Textversion</td>
<td>author</td>
</tr>
<tr>
<td>Source</td>
<td>Kyoto University</td>
</tr>
</tbody>
</table>
Combinatorial Rigidity and Generation of Discrete Structures

Shin-ichi Tanigawa

Graduate School of Engineering
Kyoto University
2010
Doctoral Dissertation

submitted to

Graduate School of Engineering, Kyoto University

in partial fulfillment of the requirement for the degree of

DOCTOR OF ENGINEERING
Abstract

The well-known Maxwell rule asserts that, if a 2-dimensional bar-and-joint framework (i.e., a structure consisting of rigid bars connected by universal joints) is statically rigid, then the number of bars is at least twice the number of joints minus three. It is notable that this rule does not concern with how a structure is realized in the space; it suggests a deep dependence between the rigidity and the topology of structures.

Deciding whether a structure is rigid or flexible is indeed the most basic problem in the structural engineering, and combinatorial rigidity can answer it only by looking at the topologies of structures. The most famous result in this context is the theorem of Laman proposed in 1970 which asserts the converse direction of the Maxwell rule for almost all situations, implying that the general behavior of 2-dimensional bar-and-joint frameworks can be captured by a combinatorial condition without looking at the geometry of frameworks.

In this dissertation, with the aid of rigidity theory since Laman’s result, we prove new combinatorial characterizations of the rigidity of several types of structures and develop efficient algorithms for generating discrete structures based on combinatorial characterizations.

First, we consider a problem of partitioning a graph into rooted-forests, and as a generalization of Tutte-Nash-Williams tree-packing theorem, we present a necessary and sufficient condition for a graph to be decomposed into edge-disjoint rooted-forests. This result leads to a new combinatorial characterization of the generic rigidity of 2-dimensional bar-joint-slider frameworks. In particular, we prove that, even though the directions of sliders are pre-determined and degenerate (i.e., some sliders have the same direction), it is combinatorially decidable whether the framework is rigid or not. We consequently extend Laman’s counting theorem, Crapo’s 3tree2-partition theorem, and the Henneberg construction of bar-and-joint frameworks to bar-joint-slider frameworks.

Next, we deal with the so-called Molecular conjecture posed by Tay and Whiteley in 1984. We solve this long-standing open problem affirmatively. In particular, we obtain a combinatorial characterization of the generic rigidity of panel-and-hinge frameworks in terms of the number of edge-disjoint spanning trees that can be packed into the underlying graphs. As a corollary, we obtain a combinatorial characterization of the generic rigidity of 3-dimensional bar-and-joint frameworks of the square of graphs.

We then deal with the problem of enumerating 2-dimensional minimally rigid bar-and-joint frameworks connecting a given set of $n$ joints. Based on the well-known reverse search paradigm, we present an algorithm for enumerating non-crossing minimally rigid frameworks in $O(n^3)$ time per output. Subsequently, we generalize the idea to develop a general enumeration technique that can apply to arbitrary non-crossing geometric graph classes. We show that our new technique provides not only faster algorithms for some enumeration problems.
compared with existing ones but also first algorithms for various problems that had not been considered to the best of our knowledge. In particular, using a generalization of Laman’s theorem to bar-joint-slider frameworks, we propose an efficient algorithm for enumerating all non-crossing $k$-degree-of-freedom mechanisms consisting of given joints some of which are connected with external environment by a set of sliders.
Acknowledgments

I am deeply grateful to my supervisor Naoki Katoh for giving me the opportunity of working on this thesis. He introduced me research area in discrete mathematics and theoretical computer science, and gave continuous supports for studying these areas. I could get a number of helpful suggestions from his deep insight and inspiration.

I would like to express my gratitude to David Avis, Takuya Kinoshita, Makoto Ohsaki, and Illeana Streinu for their patience, kindness, and guidance during the project of the enumeration described in Chapter 7. David Avis also gave me an excellent opportunity to visit McGill University from September 2007 to January 2008. I would like to thank him and members of his research group for many interesting discussions during my stay. I would like to thank Audrey Lee-St. John, Meera Sitharam, Ileana Streinu, and Louis Theran for several useful comments on the results of Chapter 6 at Barbados Workshop 2009. Also I learned a lot about matroids and submodular functions in the seminar of Fujishige-Iwata group, which are essential ingredients for completing this thesis.

I wish to express my gratitude to Professors Izuru Takewaki and Teruyuki Monnai for taking the time and effort to read my dissertation as my thesis committee.

Special thanks go to Atsushi Takizawa, Wonyong Koo, Naoyuki Kamiyama, and all members in Katoh laboratory of Kyoto University for their warm friendship.

Finally, my deepest gratitude goes to my family for ever being a source of support and encouragement.
## Contents

1 Introduction  
1.1 Rigidity Theory  
1.1.1 Rigidity  
1.1.2 Infinitesimal rigidity  
1.1.3 Static rigidity  
1.1.4 Combinatorial rigidity  
1.2 Applications  
1.2.1 Applications to structural engineering  
1.2.2 Other applications  
1.3 Contributions of this dissertation  

2 Preliminaries  
2.1 Basic Definitions  
2.2 Matroid Theory  
2.2.1 Matroids  
2.2.2 Submodular functions  
2.2.3 Matroids induced by submodular functions  
2.2.4 Matroid union  

3 Rigidity Theory  
3.1 Bar-and-joint Frameworks  
3.1.1 Rigidity  
3.1.2 Infinitesimal rigidity  
3.1.3 Generic rigidity  
3.1.4 Combinatorial rigidity  
3.2 Body-and-hinge Frameworks  
3.2.1 Infinitesimal motions of a rigid body  
3.2.2 Infinitesimal rigidity  
3.2.3 Combinatorial rigidity  

4 A Rooted-forest Partition with Uniform Vertex Demand  
4.1 Introduction  

vii
8 Enumerating Non-crossing Geometric Graphs 157
8.1 Introduction ......................................................... 157
8.2 Lexicographically Ordered Edge-constrained Triangulations ........ 160
  8.2.1 Notation ......................................................... 160
  8.2.2 Edge-constrained lexicographically largest triangulation ....... 162
  8.2.3 Improving flips .................................................... 164
  8.2.4 Deleting and inserting constrained edges ....................... 166
  8.2.5 Maintaining the F-constrained lexicographically largest triangulation 166
8.3 Enumerating Non-crossing Geometric Graphs ........................... 168
  8.3.1 General idea ...................................................... 168
  8.3.2 Time complexity of Algorithm 1 ................................ 170
  8.3.3 Applications of Algorithm 1 ...................................... 171
8.4 \( I \)-independent Minimal Representative Sets ............................ 175
  8.4.1 Enumerating triangulations based on edge insertions ........... 176
  8.4.2 Enumerating \( I \)-independent minimal representative sets ........ 179
  8.4.3 Application of Algorithm 2 ....................................... 180
8.5 Other Applications ....................................................... 181
8.6 Conclusion .............................................................. 183

9 Conclusion 185
Chapter 1

Introduction

A planar pin-jointed structure is one of the most popular structural models used in engineering (see Figure 1.1). The inequality
\[ b \geq 2j - 3 \]
is a necessary condition for the structure (with \( b \) bars and \( j \) joints) to be rigid. This condition has been already discovered in the time of James Clark Maxwell [88] and is currently widely known as Maxwell’s condition or Maxwell’s rule not only in structural engineering but also in physics. It is particularly notable that Maxwell’s rule does not concern with how a structure is “realized” in the space, and it suggests a deep dependence between the rigidity and the topology of structures. Indeed, in 1970, Laman proved that the converse direction is also true for “almost all”\(^1\) cases. Namely, the general behavior of 2-dimensional pin-jointed structures is captured by a combinatorial condition without looking at the geometry of structures.

Figure 1.1: A planar pin-jointed structure.

Taking over progress of combinatorial rigidity theory since Laman’s result, this dissertation addresses the following two themes:

- Develop new combinatorial characterizations of the rigidity of several types of structures.
- Develop combinatorial algorithms for generating structures based on combinatorial characterizations.

In order to clarify historical background as well as our contributions and applications, let us briefly (and informally) explain fundamental notions of rigidity theory.

\(^1\)The meaning of “almost all” will be clarified later.
1.1 Rigidity Theory

1.1.1 Rigidity

Up to the mid 20th century, there are two lines of mathematical researches on the rigidity: one is on the rigidity of structures consisting of bars and joints by Maxwell, Cremona, etc.; the other focuses on that of polyhedra by Cauchy, Alexandrov, etc., triggered by Euler’s conjecture. (See, e.g., [45, Chapter 1] for more detailed history.) Those researches have been currently extended to a mathematical field, called rigidity theory. Rigidity theory only concerns with rigidity/flexibility of structures, and it basically ignores elasticity, stability, and realizability caused by materials. For example, a joint is considered as a point in the target space; while a bar implies a segment connecting two points specifying the distance between two joints. Hence, by regarding each joint as a vertex and each bar as an edge, we can consider a planar pin-jointed structure as a graph drawn on the plane. In rigidity theory, such a geometric graph is called a bar-and-joint framework, and it is usually written by a pair \((G, p)\) of a graph \(G = (V, E)\) and a mapping \(p : V \to \mathbb{R}^2\), where \(V\) and \(E\) denote the sets of vertices and edges, respectively, and \(p\) specifies the position of each vertex on the plane. A mapping \(p\) is called a joint configuration.

What is an appropriate definition of rigidity/flexibility of a bar-and-joint framework? One usually says that an object is “flexible” if it can be deformed with a small fraction of force. This concept can be naturally formulated as follow: A bar-and-joint framework \((G, p)\) is flexible if it can be converted to a non-congruent framework by continuously moving each joint without changing the length of each bar; \((G, p)\) is rigid if it is not flexible.

For example, a triangle framework (Figure 1.2(a)) is rigid while a parallelogram (Figure 1.2(b)) is flexible in 2-dimensional space.

![Figure 1.2](image1.png)

(a) A triangle framework. (b) A parallelogram framework.

This definition is naturally extended to higher dimensional bar-and-joint frameworks \((G, p)\), where \(G = (V, E)\) is a graph and \(p\) maps each vertex to a point in \(\mathbb{R}^d\); a tetrahedron (Figure 1.3(a)) is rigid while a square pyramid (Figure 1.3(b)) is flexible in 3-dimensional space.

We will also discuss distinct types of structures, e.g., body-and-bar frameworks (consisting of rigid bodies connected by rigid bars), body-and-hinge frameworks (consisting of rigid bodies connected by hinges), panel-and-hinge frameworks (consisting of rigid panels connected by hinges). See Figure 1.4.
1.1. Rigidity Theory

1.1.2 Infinitesimal rigidity

Let us have a closer look at the rigidity/flexibility of a bar-and-joint framework \((G, p)\). We shall denote a continuous motion of joints at a time \(t\) by \(p_t\) (starting from \(p\)). Since each edge \(uv \in E\) is drawn as a “rigid” bar in \((G, p)\), the distance between \(p_t(u)\) and \(p_t(v)\) must be fixed, i.e.,

\[
||p_t(u) - p_t(v)|| = \text{const.} 
\]

for all \(uv \in E\), where \(|| \cdot ||\) denotes the Euclidean distance. A common strategy to deal with this system of equations is to take a first order approximation; this amounts to conditions of the form

\[
(p_0(u) - p_0(v)) \cdot (p_0'(u) - p_0'(v)) = 0 
\]

for all \(uv \in E\),

where \(\cdot\) denotes the Euclidean inner product and \(p_0'\) denotes the derivative of \(p_t\) at \(t = 0\) (see Figure 1.5). In other words, fixing the initial state of the framework, we are interested in only the assignment of a velocity vector \(p'(v)\) with each joint \(p(v)\) in such a way that

\[
(p(u) - p(v)) \cdot (p'(u) - p'(v)) = 0 
\]

for all \(uv \in E\). (1.1)

Such an assignment \(p' : V \rightarrow \mathbb{R}^2\) of velocities is called an *infinitesimal motion* of \((G, p)\). An infinitesimal motion is said to be *trivial* if it is the derivative of some isometric motion in \(\mathbb{R}^2\): translations to \(x\)-axis and \(y\)-axis of the whole framework (Figure 1.6(a)(b)), the derivative of a rotation around the origin (Figure 1.6(c)), and linear combinations of these three motions.

A framework is called *infinitesimally rigid* if any infinitesimal motion of \((G, p)\) is trivial. For example, an infinitesimal motion of a parallelogram illustrated in Figure 1.7 is nontrivial, and hence this framework is not infinitesimally rigid, being *infinitesimally flexible*. On the
other hand, it can be easily checked that any infinitesimal motion of a triangle framework is trivial (if the three joints do not lie on a line), and hence a triangle is infinitesimally rigid.

Note that an infinitesimally rigid framework may not be rigid. For example, frameworks shown in Figure 1.8 are both rigid in 2-dimensional space but not infinitesimally rigid (the arrows indicate nontrivial infinitesimal motions). It is however known that infinitesimal rigidity implies rigidity.

1.1.3 Static rigidity

It should be noted that the infinitesimal rigidity is an equivalent concept to the static rigidity that is familiar to structural engineers. Roughly speaking, the static rigidity asks whether one can assign an internal stress to each bar in such a way that stresses satisfy conditions for static equilibrium. The equivalence of these rigidity models has been discovered not only in the mathematical context (see e.g., [29, 45, 129]) but also in the engineering context (e.g., [22, 96, 113]). Although we do not refer to the static rigidity in the subsequent chapters, we would like to have a closer look at this equivalence to show the correspondence between mathematical terms and engineering terms.

We now consider a 2-dimensional bar-and-joint framework (called a pin-jointed truss, or a pin-jointed assembly in the context of engineering) consisting of $j$ joints and $b$ bars,
which is “free” on the plane, i.e., rigid motions (isometric motions) are always allowed. Suppose that an external force is assigned to each joint. We denote these forces by a \((2j)\)-dimensional vector \(f = (f_{1,x}, f_{1,y}, \ldots, f_{j,x}, f_{j,y})\). The external forces induce internal stresses (tension/compression) of bars, which can be denoted by a \(b\)-dimensional vector \(t\). The equilibrium equations are then written as

\[ At = f, \]  

where \(A\) is the so-called equilibrium matrix with the size \(2j \times b\). If \(t\) satisfies \(At = 0\), it is called a state of self-stress, and the number of independent states of self-stress is denoted by \(s\). Namely, \(s\) is the dimension of the null space of \(A\). By a fundamental result of linear algebra, we have

\[ r + s = b, \]  

where \(r\) is rank of \(A\). An external force \(f\) is in equilibrium if

\[ \sum_{1 \leq i \leq j} f_i = 0 \quad \sum_{1 \leq i \leq j} f_i \times p_i = 0, \]

where \(f_i = (f_{i,x}, f_{i,y})\) is the force applied at a joint \(i\), \(p_i\) is the coordinate of \(i\), and \(\times\) denotes the cross product (i.e., \(f_i \times p_i\) is the moment around the origin). A framework is called statically rigid if for any \(f\) in equilibrium there is an internal stress \(t\) that solves (1.2). It is not difficult to check that the set of equilibrium forces forms a \((2j - 3)\)-dimensional vector space (see, e.g., [31]). Therefore, a framework is statically rigid if and only if

\[ r = 2j - 3. \]  

On the other hand, in the kinematic analysis, we consider the relation between the elongations of the bars and the displacements of the joints (ignoring the higher-order deformations). Let us denote by a \(b\)-dimensional vector \(d\) and a \((2j)\)-dimensional vector \(e\) the elongations of the bars and the displacements of the joints, respectively. Then, the compatibility relation is written as

\[ Bd = e \]

where \(B\) is the kinematic matrix (compatibility matrix) whose size is \(b \times 2j\). Indeed, \(Bd = 0\) is identical to the system of equations given in (1.1) by putting \(d = p'\). Hence, the null space of \(B\) (that is called the rigidity matrix in the context of rigidity theory) is identical to the space of infinitesimal motions. Furthermore, it is observed that \(B\) is exactly the transpose of \(A\) (see, e.g., [96]). In the context of structural engineering, a solution of \(Bd = 0\) is called
an (inextensional) mechanism. There are two types of mechanisms: rigid motions (trivial motions) and internal mechanisms (nontrivial motions). Since the set of rigid motions forms a 3-dimensional vector space, we have

\[ r + m + 3 = 2j, \]  

where \( m \) is the number of independent internal mechanisms. A framework is infinitesimally rigid if and only if \( m = 0 \), and equivalently

\[ r = 2j - 3 \]  

by (1.5). By (1.4) and (1.6), we conclude that a framework is statically rigid if and only if it is infinitesimally rigid.

In particular, combining (1.3) with (1.5), we obtain

\[ m - s = 2j - 3 - b, \]  

which is a precise form of Maxwell’s rule; because \( m = 0 \) implies \( b \geq 2j - 3 \). A framework is \textit{statically determinate} if \( s = 0 \); otherwise \textit{statically indeterminate}. Similarly, it is \textit{kinematically determinate} if \( m = 0 \); otherwise \textit{kinematically indeterminate}. If \( s = m = 0 \), a framework is called \textit{isostatic}.

1.1.4 Combinatorial rigidity

It is obvious that the infinitesimal rigidity does depend on the position of joints: The frameworks shown in Figure 1.9 are infinitesimally rigid while the frameworks shown in Figure 1.8 with the same underlying graphs are infinitesimally flexible. Indeed, \( m \) or \( s \) of the equation (1.7) is determined by rank of the associated matrix \( B \) (or \( A \)).

Figure 1.9: Bar-and-joint frameworks, which have the same underlying graphs as those of Figure 1.8, are infinitesimally rigid on generic joint configurations.

Nevertheless, investigating \textit{combinatorial properties} of frameworks has gained prominence since the 1970’s in rigidity theory. Notice that in Figure 1.8 each joint configuration satisfies some “special” geometric relation: in the framework (a) three points lie on a line while in (b) three lines concurrent. Gluck [44] noticed that, if one assumes a \textit{generic} joint configuration, that is, a joint configuration without “special” geometric relations, then the infinitesimal rigidity of frameworks is uniquely determined only by graphs. It could be seen from the definition that almost all possible joint configurations are generic (see Chapter 3 for more detail). Asimow and Roth [8] further proved that, if one assumes a generic joint configuration,
1.1. Rigidity Theory

then the rigidity and the infinitesimal rigidity coincide. Therefore, the infinitesimal rigidity on generic joint configurations, which can be combinatorially characterized, captures the general behavior of the rigidity, so-called the generic rigidity.

One of the most famous results in rigidity theory is Laman’s theorem that classifies rigid/flexible bar-and-joint frameworks in terms of the underlying graphs.

**Theorem 1.1** (Laman [79]). Let \( G = (V, E) \) be a graph. Then, for any generic joint configuration \( p \), the bar-and-joint framework \((G, \mathbf{p})\) is rigid in 2-dimensional space if and only if \( G \) contains a subgraph \( H = (V, E_H) \) such that

- \(|E_H| = 2|V| - 3\), and
- \(|F| \leq 2|V(F)| - 3\) for any nonempty \( F \subseteq E_H \),

where \( V(F) \) denotes the set of vertices that are incident to some edge of \( F \).

For example, consider the framework illustrated in Figure 1.10(a). The underlying graph \( G = (V, E) \) has 6 vertices and 9 edges, and hence \(|E| = 2|V| - 3\) holds. If this framework is infinitesimally rigid, then \( G \) must satisfy the second condition of Theorem 1.1 (since \( G \) itself satisfies the first condition). However, setting \( F \) as the set of bold edges in the figure, we have \(|F| = 6 > 5 = 2 \cdot 4 - 3 = 2|V(F)| - 3\). This certifies that the framework (a) is infinitesimally flexible. On the other hand, consider the framework illustrated in Figure 1.10(b). Taking any subgraph consisting of 2\(|V| - 3\) edges as \( H \), it can be checked that \( H \) satisfies the conditions of Theorem 1.1 by comparing the numbers of vertices and edges; the framework (b) is infinitesimally rigid.

![Figure 1.10: (a) An infinitesimally flexible framework. (b) An infinitesimally rigid framework.](image)

For example, consider the framework illustrated in Figure 1.10(a). The underlying graph \( G = (V, E) \) has 6 vertices and 9 edges, and hence |E| = 2|V| - 3 holds. If this framework is infinitesimally rigid, then \( G \) must satisfy the second condition of Theorem 1.1 (since \( G \) itself satisfies the first condition). However, setting \( F \) as the set of bold edges in the figure, we have \(|F| = 6 > 5 = 2 \cdot 4 - 3 = 2|V(F)| - 3\). This certifies that the framework (a) is infinitesimally flexible. On the other hand, consider the framework illustrated in Figure 1.10(b). Taking any subgraph consisting of 2\(|V| - 3\) edges as \( H \), it can be checked that \( H \) satisfies the conditions of Theorem 1.1 by comparing the numbers of vertices and edges; the framework (b) is infinitesimally rigid.

Namely, Laman’s theorem characterizes the generic rigidity of bar-and-joint frameworks in the plane only by a relation of the cardinality of vertices and edges as Maxwell’s condition does. Laman’s theorem also asserts that a bar-and-joint framework satisfying the two conditions of Theorem 1.1 is isostatic. Indeed, it can be observed that a planar pin-jointed framework given in Figure 1.1, which is isostatic, satisfies these two conditions.

In the above example, we have checked that the framework (b) satisfies the conditions of Laman’s theorem by brute force manner. However, the number of subsets \( F \) of \( E_H \) exponentially increases as the number of joints increases. Indeed, Laman’s theorem does not directly provide an efficient algorithm for checking the infinitesimal rigidity of bar-and-joint frameworks. Nevertheless, due to the beautiful combinatorial structure behind Laman’s condition, efficient algorithms [43, 55, 65, 80] have been developed, notably these algorithms computing the degree of freedom of a framework theoretically and practically faster than the conventional
numerical analysis of a matrix.

It is surprising that a combinatorial characterization of the infinitesimal rigidity of 3-
dimensional bar-and-joint frameworks, which is definitely a main issue of rigidity theory, has
not been yet found. Since Maxwell’s condition in 3-dimensional space is

\[ |E| \geq 3|V| - 6, \]

one might guess that for a generic joint configuration a 3-dimensional bar-and-joint framework
\((G, p)\) is rigid if and only if \(G\) has a subgraph \(H = (V, E_H)\) such that

- \(|E_H| = 3|V| - 6\), and
- \(|F| \leq 3|V(F)| - 6\) for any nonempty \(F \subseteq E_H\) (with \(|V(F)| \geq 3\)).

However, there is a famous counterexample called the “double banana” (Figure 1.11) that
satisfies the above counting conditions but is not infinitesimally rigid in 3-dimensional space.
Despite elaborate works after Laman’s result, this problem still remains open. The existence
of such a difficult problem might be a reason why rigidity theory is so attractive.

![Figure 1.11: The double banana.](image)

## 1.2 Applications

Let us discuss applications of rigidity theory. Since rigidity theory itself deals with abstracted
models of structures, it of course serves the underlying theory of structural engineering. Such
a fundamental knowledge should be rigorous as possible as we can, and our study has been
also done in a mathematically precise formulation with concrete mathematical proofs. In
Section 1.2.1, we shall show more specific applications to structural engineering, which must
enhance the value of our works.

Thanks to wide range of applications such as bioinformatics, computer aided design, lo-
calization of sensor networks, etc., rigidity theory attains prominence in theoretical computer
science these days. We will briefly introduce these applications in Section 1.2.2.

### 1.2.1 Applications to structural engineering

*Structure optimization*, which concerns with how to generate an optimum structure or find
optimum values of design variables subject to prespecified constraints, is one of major topics in
structural engineering (see e.g. [15, 90, 94]). A basic approach for tackling problems of finding
1.2. Applications

good structures is summarized as follows: The problem is first modeled and formulated as a mathematical programming problem; then it is solved by using some optimization techniques. In this process, one usually focuses on the validity or novelty of modeling more than solvability of formulated problems. However, for meeting involved practical requirements, a resulting mathematical programming problem is getting hard to solve. In particular, in a problem of finding special types of structures such as compliant mechanisms, mobility equipments, tensegrity structures, and etc., a usual mathematical programming formulation has high nonlinearity, where even finding a convergent solution becomes nontrivial (see e.g., [92, 93]).

As an (artificial) example of applications of rigidity theory, let us consider the problem of finding a pin-jointed mechanism defined as follows: given a set of joints in the plane and a positive integer \( k \), find a \( k \) degree-of-freedom (or simply \( k \)-dof) link mechanism connecting given joints with the minimum total bar length. When \( k = 1 \), this problem can be formulated as that of finding a structure whose first eigenvalue of the stiffness matrix is equal to zero the others being positive real (see, e.g., [93] for more detail). However, rigidity theory tells us that a minimal \( k \)-dof mechanism (i.e., a \( k \)-dof mechanism such that removing any bar results in a mechanism that is not \( k \)-dof) has a beautiful combinatorial property: A family of the underlying graphs of minimal \( k \)-dof mechanisms forms a set of bases of a matroid (that will be defined in Chapter 2). Hence, from a result of combinatorial optimization it follows that finding an optimal \( k \)-dof mechanism can be solved by the following greedy algorithm:

Step 0: Initialize a mechanism \( M \) as that consisting of a given joints without any bar.
Step 1: Consider a set of bars connecting all pairs of joints, and label each of them in the order of nondecreasing lengths.
Step 2: From the smallest labeled bar \( b \), continue the following process until finding a \( k \)-dof mechanism:
   Step 2-1: If the insertion of \( b \) decreases the degree of freedom of \( M \), then insert \( b \) to \( M \).
   Step 2-2: Proceed to the next labeled bar \( b \).

Clearly, this algorithm can be implemented so that it works in polynomial time in the number of joints.

A much difficult problem in the approach of structural optimization is to enumerate all topologies of \( k \)-dof mechanisms. For \( k = 1 \), this problem is discussed in the literature of structural optimization by Kawamoto, Bendsøe, and Sigmund [77] (but taking symmetry into consideration). Unfortunately, their approach is almost a brute-force method just using only Maxwell’s rule. However, combining results of rigidity theory and a recent development of a graph enumeration technique, we can develop an algorithm for enumerating all \( k \)-dof mechanisms connecting a given set of joints in \( O(n) \) time per each, where \( n \) denotes the number of joints.

The problem of finding a minimum length \( k \)-dof mechanism seems rather artificial. The next example shows an application that is more practical. A compliant mechanism is a new type of flexible mechanism that transfers an input force to another point through elastic-
ity of materials. Although a compliant mechanism is usually modeled as a continuum with elastic joints, it is possible to generate a similar mechanism by using a bar-joint system. Ohsaki and Nishiwaki [92] presented a method for generating multi-stable bar-and-joint compliant mechanisms that have multiple self-equilibrium states, using a nonlinear programming approach. Such a mechanism can be used as a switching device, robot hand, gripper, deployable structure, etc. In their method, the optimal locations of bars and joints are found from a highly connected initial structure (i.e. the complete graph). However, due to high nonlinearity of the analysis and optimization problems, the nonlinear programming problem should be solved many times starting from different initial solutions to obtain a few types of mechanisms. Since the compliant bar-and-joint mechanism is usually statically determinate, the optimization problem can be solved easily if the design space is limited to statically determinate structures. Motivated by this observation, we developed an efficient algorithm for enumerating isostatic bar-and-joint frameworks consisting of non-crossing bars based on rigidity theory (that will be greatly discussed in Chapter 7). Combining an implementation of our algorithm with a nonlinear programming approach, we obtained many new compliant mechanisms with up to 13 joints [91].

1.2.2 Other applications

Protein conformation. To identify flexible/rigid region in a protein is one of the central issues in the field of molecular biology as this could provide insight into its function and a means to predict possible changes of structural flexibility by environmental factors such as temperature and pH [117].

Let us consider a molecule consisting of atoms connected by covalent bonds. Each bond constraints the distance between two atoms. In addition, it is known that the angle between two bonds intersecting at an atom is fixed. Namely, if there are a bond connecting atoms A and B and a bond connecting B and C, then the distance between A and C is also determined even though no bond exists between A and C. Such a constraints system can be modeled as a bar-and-joint framework of the square of a graph. The square of a graph $G = (V, E)$ is defined as $G^2 = (V, E^2)$, where $E^2 = E \cup \{uv \in V \times V : u \neq v \text{ and } uw, wv \in E \text{ for some } w \in V \setminus \{u, v\}\}$. However, as we have briefly mentioned in Section 1.1.4, the rigidity of 3-dimensional bar-and-joint frameworks is not well understand yet. See e.g. [57, 58, 66, 131, 133] for more details on this application.

It is known that a molecule can be also modeled as a body-and-hinge framework by regarding each atom (vertex) as a rigid body and each bond (edge) as a hinge since in the square of a graph a vertex and its neighbor vertices always form a complete graph. Tay and Whiteley posed a conjecture, known as the Molecular Conjecture, which asks the combinatorial properties of the rigidity of such body-and-hinge frameworks.

\[\text{Notice however that as will be remarked in Section 6.1, this body-and-hinge framework has a special property such that all the hinges (lines) incident to a body are intersecting each other at the center of the body. Hence, Tay-Whiteley's theorem that characterizes the rigidity of body-and-hinge frameworks cannot be applied to such a special framework. See Chapter 6 for more detail.}\]
A proof of the Molecular Conjecture provides us with a combinatorial algorithm for computing not only the degree of freedom but also rigid clusters significantly faster than the conventional numerical analysis. In fact, several combinatorial algorithms such as 3D pebble game [64] were developed, based on which a computer software such as FIRST [39, 66] (Floppy Inclusions and Rigid Substructure Topography) specialized to the static analysis of molecules were developed. Later, the pebble game algorithm was further embedded on several new methods for analyzing the dynamic motions of proteins, such as ROCK [82], FRODA [122], and the Protein Folding Server by Amato [6]. Meanwhile, in spite of the lack of the rigorous proof of the Molecular Conjecture, empirical evidences that support the conjecture have been accumulated [117, 133] over years. We will prove this conjecture affirmatively in Chapter 6.

**Computer Aided Design.** CAD (computer aided design) becomes essential tools for engineers to design complicated systems. In CAD, a design is built on a collection of primitive geometric objects such as points, line segments, and circular arcs with a set of constraints such as distance, incidence, symmetry, and angular constraints. 3D-CAD further deals with solid objects such as polyhedrons and spheres. A fundamental task of CAD is to maintain the system of constraints under additions and deletions of objects or constraints, to compute the degree of freedom of the design, to detect the consistency of the system, and to decompose the objects into dependent groups in reasonable computational time.

In principle, solving such a geometric constraints system is an algebraic problem, but if we consider the generic situation the solution could be described in some combinatorial terms as we have seen in Section 1.1.4 (bar-and-joint frameworks can be considered as distance-constraints systems). For example, Servatius and Whiteley [107] considered the systems of direction constraints and proved a combinatorial characterization for the (local) uniqueness of the solutions, which is described as in the form of Laman’s theorem (see also Whiteley [129] and Jackson and Jordán [63]). Angle constraints, however, have proved to be more challenging [102]. Nevertheless, there are rich amounts of researches on geometric constraints system taking rigidity theory into accounts. See a survey paper [52] for more details.

**Network localization.** Let us introduce the *global rigidity* that is one of the most important concepts in rigidity theory. A bar-and-joint frameworks \((G, p)\) is called global rigid if all of bar-and-joint frameworks equivalent to \((G, p)\) are congruent. Under the generic assumption, it is combinatorially decidable whether \((G, p)\) is global rigid in \(\mathbb{R}^2\) [18, 27]. This result answers several questions arising from the sensor network localization.

In the theory of wireless sensor networks, there is a scheme for maintaining a sensor network with a small cost in which each node can measure the distances to its neighbors and in addition a few special nodes (called beacons or anchors) know their locations. (It is preferable to design such a network with anchors as small as possible for low-cost, low energy, and other technical reasons, see, e.g., [86].) In this scheme, the process of computing the locations of the nodes is called the *network localization*. Namely, in the network localization
problem, given the distances between some pairs of nodes as well as the locations of some anchors, we are asked to determine the locations of all nodes. A fundamental question for the robust maintenance of a sensor network is whether a solution to the network localization problem is unique or not; if it is not unique, how many new anchors are needed for the unique solution and where should we place them?

Since the locations of anchors are known, we also have distance information between pairs of anchors. Therefore, a solution of the localization problem is unique if and only if the graph consisting of the edges between pairs of nodes whose distances are known is globally rigid. Based on this observation, several combinatorial algorithms are proposed for the localization problem and the related ones (see e.g., [9, 36]).

1.3 Contributions of this dissertation

This dissertation consists of eight chapters. In Chapter 2, we review basic facts about combinatorial optimization, in particular on matroids and those induced by submodular functions. In Chapter 3, we review the basic facts on rigidity theory. We will provide formal definitions of the infinitesimal rigidity of bar-and-joint frameworks and body-and-hinge frameworks, and shows combinatorial properties of the infinitesimal rigidity with emphasis on the relation to matroid theory discussed in Chapter 2. The main results of this dissertation will appear in Chapter 4 through Chapter 8. Here we give summaries of these chapters and illustrate the connection among the problems we will deal with.

Chapter 4: Rooted-forest Partition with Uniform Vertex Demand. As we have seen in Theorem 1.1, celebrated Laman’s theorem characterizes the generic rigidity of 2-dimensional bar-and-joint frameworks in terms of a relation of the numbers of vertices and edges. Instead of Laman’s condition, several equivalent characterizations are known. In particular, we will see in Chapter 3 that characterizing the rigidity in terms of a tree-partition (i.e., the property of a graph to be decomposed into several tree-shaped pieces) is commonly appeared in rigidity theory. These results are based on the Tutte-Nash-Williams tree-packing theorem, which is one of the most famous results in combinatorial optimization.

In this chapter we will propose an extension of the Tutte-Nash-Williams tree-packing theorem to rooted-forest partitions (that will be formally defined in Chapter 4) such that each vertex is spanned by exactly $d$ rooted-forests among them, where $d$ is a given integer. Although this chapter concerns only with the topic of graph theory, our newly proposed rooted-forest-partition will be directly used for characterizing the rigidity of bar-and-joint frameworks having line-sliders in Chapter 5.

Chapter 5: Infinitesimal Rigidity of Bar-and-slider Frameworks. Laman’s characterization is quite useful for developing efficient algorithms for problems related to bar-and-joint frameworks. However, Laman’s theorem does not actually concerns with how the framework is connected to the external environment, which restricts the range of applica-
1.3. Contributions of this dissertation

In a practical situation a framework is connected to walls and the floor by sliders and pins, and even if a framework does not satisfy Laman’s condition it could be rigid as a whole structure as shown in Figure 1.12. Since pinning a joint can be regarded as attaching two distinct sliders to this joint, we may assume that a bar-and-joint framework is connected to external environment via sliders. Such a framework is called a bar-and-slider framework.

![Figure 1.12: A framework that does not satisfy Laman’s condition but is rigid.](image)

In this chapter, we shall propose a generalization of Laman’s theorem to the infinitesimal rigidity of bar-and-slider frameworks. The other combinatorial results on bar-and-joint frameworks, e.g., those in terms of graph-decomposition and inductive construction, will be also generalized to the case of bar-and-slider frameworks. The proof is done by using the rooted-forest partition theorem proposed in Chapter 4.

Chapter 6: A Proof of the Molecular Conjecture. A $d$-dimensional body-and-hinge framework is a structure consisting of rigid bodies connected by hinges in $d$-dimensional space (see Figure 1.4(b)). As a counterpart of Laman’s theorem on bar-and-joint frameworks, the generic rigidity of a body-and-hinge framework has been characterized in terms of the underlying graph independently by Tay [114] and Whiteley [127] as follows: a graph $G$ can be realized as an infinitesimally rigid body-and-hinge framework by mapping each vertex to a body and each edge to a hinge if and only if $\binom{d+1}{2} G$ contains $\binom{d+1}{2}$ edge-disjoint spanning trees, where $\binom{d+1}{2} G$ is the graph obtained from $G$ by replacing each edge by $\binom{d+1}{2}$ parallel edges. In 1984 they jointly posed a question about whether their combinatorial characterization can be further applied to a singular case [115]. Specifically, they conjectured that $G$ can be realized as an infinitesimally rigid body-and-hinge framework if and only if $G$ can be realized as those with the additional “hinge-coplanar” property, i.e., panel-and-hinge frameworks (Figure 1.4(c)).

Notice that the theorem of Tay and Whiteley is only applied to generic body-and-hinge frameworks; a body-and-hinge framework is said to be generic if there is no “special” geometric relation among the positions of hinges. As we have seen in bar-and-joint frameworks, even if a framework is rigid on a generic joint configuration, it could be flexible on some special joint configuration. This phenomenon also happens in the infinitesimal rigidity of body-and-hinge frameworks. Since a panel-and-hinge framework may be a nongeneric body-and-hinge framework (because all the hinges incident to each body must be contained in a common hyperplane), it is not obvious whether the combinatorial characterization of Tay and Whiteley can be applied to panel-and-hinge frameworks.
The conjecture of Tay and Whiteley is now called the Molecular Conjecture due to the equivalence between the infinitesimal rigidity of 3-dimensional panel-and-hinge frameworks and that of bar-and-joint frameworks derived from molecules in 3-dimensional space. Indeed, algorithms used by some software [6, 39, 82, 122] for analyzing flexible/rigid region in proteins rely their correctness upon the theory of structural rigidity (see Section 1.2.2). From a mathematical point of view, however, the correctness proof is incomplete because it relies on the Molecular Conjecture, which has been a long-standing open problem over twenty-five years. In the 2-dimensional case this conjecture has been proved by Jackson and Jordán in 2006. In this chapter we will prove this conjecture affirmatively for general dimension.

Chapter 7: Enumerating Non-crossing Minimally Rigid Bar-and-joint Frameworks. Recall that a rigid bar-and-joint framework is called isostatic or minimally rigid if removing any bar results in a flexible framework. Apart from the theoretical part of rigidity theory, in this chapter we shall discuss how to efficiently generate all generically minimally rigid bar-and-joint frameworks (simply called minimally rigid frameworks). By Laman’s theorem, we know that a graph $G = (V, E)$ can be realized as a minimally rigid framework in the plane if and only if $|E| = 2|V| - 3$ and $|F| \leq 2|V(F)| - 3$ holds for any nonempty $F \subseteq E$. Therefore, the problem of enumerating minimally rigid frameworks is equivalent to the enumeration of graphs satisfying Laman’s condition. We will first show that, combining a known combinatorial property of minimally rigid frameworks and an enumeration technique by Uno, a collection of minimally rigid frameworks of $n$ joints can be enumerated in $O(n^3)$ time per each.

However, the total number of minimally rigid frameworks is very huge: for a generic joint configuration with $n$ joints, it amounts to $\Omega(n^{cn})$ for some constant $c$, and hence it is impractical to enumerate all minimally rigid frameworks; it is necessary to impose reasonable constraints to decrease the size of output. In this chapter, we shall consider the following two constraints:

(non-crossing constraint): Any pair of bars may not intersect each other except possibly for their endpoints.

(bar-inclusion constraint): Several bars are prespecified to be included.

We will propose an algorithm for enumerating all minimally rigid frameworks satisfying these two constraints in $O(n^3)$ time per each.

Chapter 8: Enumerating Non-crossing Geometric Graphs. A non-crossing geometric graph is an embedded graph on a point set on the plane consisting of non-crossing straight-line edges. Enumerating non-crossing geometric graphs on a given point set is a fundamental problem in computational geometry, and several algorithms have been proposed for e.g., triangulations [11, 20], non-crossing spanning trees [2, 11], pointed pseudo-triangulations [16, 21], non-crossing connected graph [2], and so on (see Chapter 7 for the definitions of each graph class). All the previous known algorithms rely on a particular property of each graph class; they cannot be applied to other graph classes.
Table 1.1: Time complexities of new algorithms and previous ones for non-crossing geometric graphs.

<table>
<thead>
<tr>
<th>Class</th>
<th>New results</th>
<th>Previous best</th>
</tr>
</thead>
<tbody>
<tr>
<td>plane straight-line graphs</td>
<td>$O(pg(P))$</td>
<td>$O(n \log n \cdot pg(P))$ [2]</td>
</tr>
<tr>
<td>connected graphs</td>
<td>$O(cg(P))$</td>
<td>$O(n \log n \cdot cg(P))$ [2]</td>
</tr>
<tr>
<td>spanning trees</td>
<td>$O(n \cdot tri(P) + st(P))$</td>
<td>$O(n \log n \cdot st(P))$ [2]</td>
</tr>
<tr>
<td>minimally rigid frameworks</td>
<td>$O(n^2 \cdot mrf(P))$</td>
<td>$O(n^3 \cdot mrf(P))$ (Chap. 7)</td>
</tr>
<tr>
<td>perfect matchings</td>
<td>$O(n^{3/2} \cdot tri(P) + n^{5/2} pm(P))$</td>
<td>—</td>
</tr>
</tbody>
</table>

In this section, we shall generalize the idea proposed in Chapter 7 to arbitrary non-crossing geometric graph classes. Our new enumeration technique provides faster algorithms for various enumeration problems compared with existing ones, such as those for plane straight-line graphs, non-crossing connected graphs, non-crossing spanning trees, and non-crossing minimally rigid frameworks. Moreover, since the idea is quite simple, it can be applied possibly to many other enumeration problems, for which enumeration algorithms have not been known to the best of our knowledge, such as non-crossing matchings, non-crossing red-and-blue matchings, non-crossing $k$-vertex or $k$-edge connected graphs or non-crossing directed geometric graphs. In Table 1.1 we list the time complexities of (a part of) new algorithms obtained in this paper, where we use the following notations to denote the numbers of graphs on a point set $P$: $pg(P)$ for plane straight-line graphs, $cg(P)$ for non-crossing spanning connected graphs, $st(P)$ for non-crossing spanning trees, $mrf(P)$ for non-crossing minimally rigid frameworks, $tri(P)$ for triangulations, and $pm(P)$ for non-crossing perfect matchings.

Moreover, combining our general enumeration technique and a generalization of Laman’s theorem to bar-and-slider frameworks, we obtain an efficient algorithm for enumerating all non-crossing $k$-degree-of-freedom mechanisms connecting given joints, some of which are connected with external environment by a set of sliders. In addition, our general technique can be applied to the problem of enumerating non-crossing $k$-degree-of-freedom mechanisms on a non-generic joint configuration.

Connection among the chapters. Figure 1.13 explains the connection among the chapters. We can split the thesis into two parts: the first part deals with the theoretical part of rigidity theory, and the second one does with the design of efficient algorithms for generating discrete structures based on results of rigidity theory. In order to derive a combinatorial characterization of the infinitesimal rigidity of bar-and-slider frameworks in Chapter 5, we need a necessary and sufficient condition of a graph to be decomposed into edge-disjoint rooted-forests given in Chapter 4. An idea for enumerating non-crossing minimally rigid graph proposed in Chapter 7 will be abstracted to a general technique for non-crossing geometric graphs in Chapter 8. A combination of this general enumeration technique and a generalization of Laman’s theorem to bar-and-slider frameworks given in Chapter 5 leads to an efficient algorithm for enumerating non-crossing minimally rigid bar-and-slider frameworks.
Publications. Chapter 4 comes from the work that will appear in Proceedings of the 4th Workshop on Algorithms and Computation [76]. Chapter 5 comes from the work appeared in Proceedings of the 20th International Symposium on Algorithms and Computation [75]. Chapter 6 comes from a work appeared in Proceedings of 25th ACM Symposium on Computational Geometry [72]. Chapter 7 is based on joint works with David Avis, Makoto Ohsaki, and Illeana Streinu that have appeared in Graph and Combinatorics [12] and Discrete & Computational Geometry [13]. Chapter 8 is based on the author’s two papers [73, 74]: Chapter 8.2 is a part of work appeared in Discrete Applied Mathematics [73], and the rest of Chapter 8 has appeared in Discrete & Computational Geometry [74].
Chapter 2

Preliminaries

In this chapter, we discuss basic definitions in Section 2.1 and fundamental facts of matroid theory in Section 2.2. A matroid, which is an abstraction of linear dependence and cycles of graphs, is a key notion for investigating the underlying combinatorial properties of the rigidity. As we will see in Chapter 3, every combinatorial characterization of the infinitesimal rigidity such as Laman’s theorem given in Theorem 1.1, Crapo’s theorem (Theorem 3.19), or Recski’s theorem (Theorem 3.18) is derived from the matroidal structure of bar-and-joint frameworks. Indeed, we will see several kinds of matroids in every chapter.

2.1 Basic Definitions

Let $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ denote the set of integral, rational, and real numbers, respectively. The $n$-dimensional vector space of real $n$-tuples is denoted by $\mathbb{R}^n$. For a finite set $X$, we denote its cardinality by $|X|$. When a set $X$ is a subset of a set $Y$, we write $X \subseteq Y$, and when $X$ is a proper subset of $Y$ (i.e., $X \subseteq Y$ and $X \neq Y$), we write $X \subset Y$ or simply $X \subset Y$. Throughout this dissertation, we do not distinguish a singleton $\{x\}$ and its element $x$. The union of a set $X$ and a singleton $\{x\}$ is sometimes abbreviated by $X + x$ while $X \setminus \{x\}$ by $X - x$, if it is clear from the context.

Let $V$ be a nonempty finite set. We denote the collection of all subsets of $V$ by $2^V$. A real-valued function defined on $2^V$ is called a set function on the ground set $V$.

An undirected graph (or simply, a graph) is a pair $G = (V, E)$ of a finite set $V$ of elements called vertices and a finite set $E$ of unordered pairs of vertices called edges. An edge consisting of $u$ and $v$ of $V$ is simply denoted by $uv$ or $vu$ rather than $\{u, v\}$, and $u$ and $v$ are called the endpoints. For $e \in E$ with $e = uv$, we say that $e$ connects $u$ and $v$, $e$ is incident to $u$ and $v$, and $u$ and $v$ are adjacent in $G$. If $u = v$, then an edge $e = uv$ is called a self-loop or (simply, a loop). A graph $G' = (V', E')$ is called a subgraph of $G$ if $V' \subseteq V$ and $E' \subseteq E$. If $E' = \{uv \in E : u, v \in V'\}$, then $G' = (V', E')$ is called the induced subgraph of $G$ by $V'$ and we write $G' = G[V']$.

For a finite set $V$, the complete graph on $V$ is the graph $(V, \{uv : u, v \in V\})$, and we denote it by $K(V)$. If $|V| = n$, it is also denoted by $K_n$. Also, we simply use the notation
To indicate the edge set of $K(V)$ if it is clear from the context.

In a multigraph $G = (V,E)$, $E$ is defined as a multiset of unordered pairs of vertices, i.e., there are possibly two or more edges $e_1, e_2, \ldots, e_k$ connecting the same two vertices, and these edges are called multiple edges or parallel. If there are no multiple edges, a graph is particularly called simple. However, we do not usually distinguish simple graphs and multigraphs, and simply call them just graphs if it is clear from the context.

We say that $F \subseteq E$ spans $v \in V$ if there is some edge $e \in F$ that is incident to $v$. The set of vertices spanned by $F \subseteq E$ is denoted by $V(F)$. If $V(F) = V$, $F$ is called a spanning set. The edge-induced subgraph by $F$ is written by $G[F] = (V(F), F)$.

A path between $u \in V$ and $\tilde{u} \in V$ is a sequence of edges in $E$ of the form $(uv_1, v_1v_2, \ldots, v_{k-1}v_k, v_kv)$. A path is called a cycle if $u = \tilde{u}$ and no vertex is repeated among $u, v_1, \ldots, v_k$. $G$ is called connected if there is a path between $u$ and $\tilde{u}$ for each $u, \tilde{u} \in V$. The maximal connected subgraphs of $G$ are called connected components of $G$ and let $\kappa(G)$ denote the number of them. Similarly, for an edge subset $F \subseteq E$, let us denote by $\kappa(F)$ the number of connected components of $G[F] = (V(F), F)$.

A forest is a subset of edges without a cycle. For a connected graph $G = (V,E)$, a subset $F \subseteq E$ is called a spanning tree if $F$ spans $V$ but has no cycle. It is well known that for any forest $F \subseteq E$ we have

$$|F| = |V| - \kappa((V,F)) = |V(F)| - \kappa(F).$$

In particular, if $F$ is a spanning tree, we have $|F| = |V| - 1$. See Figure 2.1 for examples.

![Figure 2.1:](image)

A collection of edge subsets $\{F_1, \ldots, F_k\}$ is (edge-)disjoint if $F_i \cap F_j = \emptyset$ for any $1 \leq i, j \leq k$ with $i \neq j$. For example, a collection of spanning trees $\{T_1, \ldots, T_k\}$ is said to be $k$ edge-disjoint spanning trees if $\{T_1, \ldots, T_k\}$ is disjoint.

For any vertex subset $X \subset V$, the set of edges $uv \in E$ with $u \in X$ and $v \in V \setminus X$ is denoted by $\delta_E(X)$ or simply by $\delta(X)$ if the underlying edge set is clear from the context. For a connected graph $G = (V,E)$, $F \subseteq E$ is called a (edge-)cut if removing $F$ from $G$ results in a disconnected graph. Obviously, $\delta(X)$ is a cut for any $X \subseteq V$ with $X \neq \emptyset, V$. $G$ is called $k$-edge-connected if the size of any cut is at least $k$. Analogously, $G$ is called $k$-vertex-connected if removing any $k - 1$ vertices still keep a graph connected.
2.2 Matroid Theory

2.2.1 Matroids

A matroid, first introduced by Whitney in 1935, is an abstract theory of linear dependence, and the importance of matroids for combinatorial optimization was revealed by J. Edmonds in the 1960s [104]. A notion of matroids was becoming important in rigidity theory in the 1980s, and it is indeed an essential tool in our study. For more detailed description on matroids, refer to, e.g., [95, 104].

A matroid \( M = (E, I) \) is a pair of a finite set \( E \) and a collection \( I \) of subsets of \( E \) (i.e., \( I \subseteq \mathcal{P}(E) \)) satisfying the following conditions:

(I1) \( \emptyset \in I \);
(I2) If \( X \subseteq Y \in I \), then \( X \in I \);
(I3) If \( X, Y \in I \) and \( |X| < |Y| \), then there is an element \( v \in Y \setminus X \) such that \( X + v \in I \).

The elements of \( I \) are called independent, while a subset of \( E \) not contained in \( I \) is called dependent. A maximal independent set is called a base of \( M \).

A matroid has an equivalent definition in terms of bases: a collection \( B \) of subsets of \( E \) forms a set of bases of a matroid on \( E \) if and only if it satisfies the following conditions:

(B1) \( B \) is non-empty;
(B2) If \( B_1, B_2 \in B \) and \( x \in B_1 \setminus B_2 \), then there is an element \( y \in B_2 \setminus B_1 \) such that \( B_1 - x + y \in B \).

The proof can be seen in e.g. [95, Corollary 1.2.5]. Property (B2) is referred to as the base exchange property. A spanning set of \( M \) is a subset of \( E \) that contains a base. An inclusionwise minimal dependent set is called a circuit.

The rank function of \( M = (E, I) \) is a set function \( r_M : 2^E \to \mathbb{Z} \) defined by

\[
r_M(F) = \max\{|Y| : Y \subseteq F, Y \in I\}; \tag{2.1}
\]

and \( r_M(F) \) is called the rank of \( F \). This implies that \( F \) is independent if and only if \( r_M(F) = |F| \). The rank of a matroid \( M \) is defined as \( r_M(E) \).

By (I3), it is easy to see that any base of \( M \) has the same cardinality equal to \( r_M(E) \). Also, \( F \) is a spanning set if and only if \( r_M(F) \) is equal to the rank of \( M \).

Example 2.1 (Linear matroid). Let \( A \) be a \( m \times n \)-matrix over some field. Let \( E_1 \) be the set of row vectors of \( A \), and \( I_1 \) be the collection of linearly independent subsets of \( E_1 \). Then, \( M_1 = (E_1, I_1) \) forms a matroid called a linear matroid. Checking (I1) and (I2) is trivial. Consider \( X, Y \in I_1 \) with \( |X| < |Y| \). The dimensions of the linear spaces spanned by \( X \) and \( Y \) are equal to \( |X| \) and \( |Y| \), respectively, and hence there is \( v \in Y \) that cannot be spanned by \( X \). We thus have \( X + v \in I_1 \), implying (I3). A base of \( M_1 \) is indeed a base (in the sense of linear algebra) of the row space of \( A \), and the rank of \( F \subseteq E_1 \) is equal to the dimension of the linear space spanned by \( F \subseteq E_1 \). \( \Box \)
Example 2.2 (Uniform matroid). For a finite set $E_2$ and an integer $k \geq 0$, let $\mathcal{I}_2 = \{X \subseteq E_2 : |X| \leq k\}$. Clearly, a pair $\mathcal{M}_2 = (E_2, \mathcal{I}_2)$ is a matroid and it is called a uniform matroid. The set function
\[
r_{\mathcal{M}_2}(F) = \min\{|F|, k\}
\]
is the rank function of $\mathcal{M}_2$. □

Example 2.3 (Graphic matroid). Suppose that we are given an undirected graph $G = (V, E_3)$. Let $\mathcal{I}_3 = \{F \subseteq E_3 : F$ is a forest$\}$. It is easy to see that a pair $\mathcal{M}_3 = (E_3, \mathcal{I}_3)$ satisfies (I1) and (I2). Consider forests $X, Y \in \mathcal{I}_3$ with $|X| < |Y|$. Since $(V, X)$ has a larger number of connected components than $(V, Y)$, there is an edge $e \in Y \setminus X$ such that $e$ connects two distinct connected components of $(V, X)$. As $X + e$ does not contain a cycle, (I3) is satisfied. Thus $\mathcal{M}_3$ is a matroid and it is called a graphic matroid. The rank function is defined by
\[
r_3(F) = \max\{|T| : \text{a subset } T \subseteq F \text{ does not contain a cycle}\}
= |V| - \kappa((V, F))
= |V(F)| - \kappa(F).
\]
□

Basic properties on circuits. The followings are well-known properties of matroids.

Lemma 2.1. For a base $B$ of a matroid $\mathcal{M} = (E, \mathcal{I})$ and an element $e$ with $e \in E \setminus B$, $B + e$ contains the unique circuit, denoted by $C(B, e)$. Moreover, $e \in C(B, e)$.

The proof can be seen in e.g. [95, Corollary 1.2.6]. $C(B, e)$ of Lemma 2.1 is called the fundamental circuit of $e$ with respect to $B$.

Lemma 2.2. Let $\mathcal{M}$ be a matroid. For two circuits $C_1$ and $C_2$ of $\mathcal{M}$ with $e \in C_1 \cap C_2$ and $f \in C_1 \setminus C_2$, there exists a circuit $C_3 \subseteq (C_1 \cup C_2) - e$ satisfying $f \in C_3$

This is called a circuit elimination property. For the proof, see, e.g., [95, Proposition 1.4.11].

Restriction, truncation, and connectivity. The restriction of a matroid $\mathcal{M} = (E, \mathcal{I})$ to $F \subseteq E$, denoted by $\mathcal{M}|F$, is defined as
\[
\mathcal{M}|F = (F, \{I \in \mathcal{I} : I \subseteq F\}).
\]
It is easy to see that $\mathcal{M}|F$ is a matroid on $F$. A base of $\mathcal{M}|F$ is a maximal independent set of $\mathcal{M}$ contained in $F \subseteq E$, which is simply called a base of $F$.

For a matroid $\mathcal{M} = (E, \mathcal{I})$ and an integer $k$, we define a truncated matroid as
\[
\mathcal{M}^k = (E, \{I \in \mathcal{I} : r(I) \leq k\}).
\]
Again, it is obvious that \( M^k \) satisfies (I1)(I2)(I3), and hence \( M^k \) is a matroid. The rank function of \( M^k \) is written by

\[
r_{M^k}(F) = \min\{r_M(F), k\}.
\]

Let us consider two disjoint finite sets \( E_1 \) and \( E_2 \), and let \( M_1 = (E_1, I_1) \) and \( M_2 = (E_2, I_2) \) be matroids on \( E_1 \) and \( E_2 \). The direct sum of \( M_1 \) and \( M_2 \) is defined as

\[
M_1 \oplus M_2 = (E_1 \cup E_2, \{I_1 \cup I_2 : I_1 \in I_1, I_2 \in I_2\}),
\]

which is obviously a matroid on \( E_1 \cup E_2 \). It follows from the definition that, for any \( F_1 \in E_1 \) and \( F_2 \in E_2 \),

\[
r_{M_1 \oplus M_2}(F_1 \cup F_2) = r_{M_1}(F_1) + r_{M_2}(F_2).
\]

Define the relation \( \sim \) on \( E \) as follows. For \( e, f \in E \), \( e \sim f \) holds if and only if \( e = f \) or there exists a circuit that contains both \( e \) and \( f \) in \( M \). Then, it is known that \( \sim \) is an equivalence relation on \( E \), i.e., \( a \sim b \) and \( b \sim c \) imply \( a \sim c \) for any \( a, b, c \in E \) (see, e.g., [95, Proposition 4.1.3] for the proof). The equivalence classes induced by \( \sim \) are called the connected components of \( M \). In order to avoid confusion, we call a connected component of a matroid an M-connected component.

It is known that, if \( X_1, X_2, \ldots, X_k \) are the M-connected components of \( M \), then

\[
M = M|X_1 \oplus M|X_2 \oplus \cdots \oplus M|X_k,
\]

\[
r_M = r_{M|X_1} + r_{M|X_2} + \cdots + r_{M|X_k},
\]

see e.g. [95, Corollary 4.2.13]. If \( E \) is itself an M-connected component of \( M \), then \( M \) is said to be M-connected. Namely, \( M \) is M-connected if and only if for any \( e, f \in E \) there is a circuit containing \( e \) and \( f \).

### 2.2.2 Submodular functions

Submodular system is a natural generalization of matroids, which appears in the systems of graphs and networks. For more information on submodular functions, refer to, e.g., [41, 104].

For a finite set \( E \), a set function \( f : 2^E \to \mathbb{R} \) is called submodular if

\[
f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)
\]

for each pair of subsets \( X, Y \subseteq E \). A function \( f' : 2^E \to \mathbb{R} \) is called nondecreasing if \( f'(X) \leq f'(Y) \) for each pair of subsets \( X, Y \subseteq E \) with \( X \subseteq Y \).

It is easy to see that the submodularity is closed under the addition and multiplication of non-negative scalars.

**Lemma 2.3.** Let \( f_1, f_2 : 2^E \to \mathbb{R} \) be submodular functions and \( \lambda \geq 0 \) be a nonnegative real number. Then, the functions \( \lambda f_1 \) and \( f_1 + f_2 \) are also submodular.
Example 2.4 (Rank function of a matroid). An example of a submodular function we have already seen is the rank function of a matroid $M = (E, I)$. To see this, consider $X, Y \subseteq E$. Let $B_{X \cap Y}$ be a base of $X \cap Y$. Let $B'_X$ be a base of $X$. Then, since $|B_{X \cap Y}| \leq |B'_X|$, there is an element $w \in B'_X \setminus B_{X \cap Y}$ such that $B_{X \cap Y} + w \in I$ by (I3). Continuing this process, we obtain a set of elements $W \subseteq B'_X \setminus B_{X \cap Y}$ such that $B_{X \cap Y} \cup W \in I$ and $|B_{X \cap Y} \cup W| = |B'_X|$. Thus, setting $B_X = B_{X \cap Y} \cup W$, $B_X$ is a base of $X$ with $B_{X \cap Y} \subseteq B_X$. Similarly, by (I3), we can take a base $B_{X \cup Y} \setminus Y$ of $X \cup Y$ with $B_X \subseteq B_{X \cup Y}$. By (I2), $B_{X \cup Y} \setminus (B_X \setminus B_{X \cap Y})$ is independent. Since $B_{X \cup Y} \setminus (B_X \setminus B_{X \cap Y}) \subseteq Y$, we have $|B_{X \cup Y} \setminus (B_X \setminus B_{X \cap Y})| \leq r_M(Y)$. Consequently,

$$r_M(X \cup Y) + r_M(X \cap Y) = |B_{X \cup Y}| + |B_{X \cap Y}|$$

$$= |B_{X \cup Y} \cap B_X| + |B_{X \cup Y} \setminus B_X| + |B_{X \cap Y}|$$

$$= |B_X| + |B_{X \cup Y} \setminus (B_X \setminus B_{X \cap Y})| \quad \text{(by } B_{X \cap Y} \subseteq B_X \subseteq B_{X \cup Y})$$

$$\leq r_M(X) + r_M(Y).$$

Thus $r_M$ is submodular. Indeed, matroids can be characterized in terms of set functions as follows: $r : 2^E \to \mathbb{Z}$ is a rank function of a matroid if and only if

(R1) $0 \leq r(F) \leq |F|$ for any $F \subseteq E$;

(R2) $r$ is nondecreasing;

(R3) $r$ is submodular.

The proof can be seen in e.g., [95, Section 1.3].

Example 2.5 (Cut functions). A more specific example of submodular functions is a cut function of a graph $G = (V, E)$. Recall that, for any $X \subseteq V$, $\delta(X)$ denotes the set of edges $e \in E$ such that one endpoint of $e$ is in $X$ and the other is in $V \setminus X$. Then, we have, for any $X, Y \subseteq V$,

$$|\delta(X)| + |\delta(Y)| \geq |\delta(X \cup Y)| + |\delta(X \cap Y)|.$$  \hfill (2.5)

This can be easily seen by counting the contribution of each edge to each side of (2.5). Hence, the set function $f_1(F) := |\delta(F)| \ (F \subseteq E)$ is submodular.

Example 2.6. Here is another example arising from a graph $G = (V, E)$. Recall that, for any $F \subseteq E$, $V(F)$ denotes the set of vertices spanned by $F$. Notice that, for any $X, Y \subseteq E$,

$$V(X \cup Y) = V(X) \cup V(Y) \quad \text{and} \quad V(X \cap Y) \subseteq V(X) \cap V(Y).$$

Hence we have

$$|V(X)| + |V(Y)| = |V(X) \cup V(Y)| + |V(X) \cap V(Y)| \geq |V(X \cup Y)| + |V(X \cap Y)|$$ \hfill (2.6)

implying that the set function $f_2(F) := |V(F)| \ (F \subseteq E)$ is submodular.
2.2.3 Matroids induced by submodular functions

We have seen that matroids are characterized by integer-valued nondecreasing submodular functions with one additional condition (R1). It is known that any integer-valued nondecreasing submodular functions (possibly loosing (R1)) can generate a matroid;

**Proposition 2.4.** Let \( f : 2^E \rightarrow \mathbb{Z} \) be an integer-valued nondecreasing submodular function. Then, \((E, \mathcal{I}_f)\) is a matroid, where

\[
\mathcal{I}_f = \{ I \subseteq E : |I'| \leq f(I') \text{ for all nonempty } I' \subseteq I \}. \tag{2.7}
\]

See e.g., [95, Proposition 12.1.1] for the proof. \((E, \mathcal{I}_f)\) of Proposition 2.4 is referred to as the matroid induced by \( f \), denoted by \( M_f \).

**Example 2.7** ((\(k, l\))-sparse graphs). Let \( G = (V, E) \) be a (multi)graph. For positive integers \( k \) and \( l \), define a set function \( \varrho_{k,l} : 2^E \rightarrow \mathbb{Z} \) by

\[
\varrho_{k,l}(F) = k|V(F)| - l \tag{2.8}
\]

for \( F \subseteq E \). As shown in Example 2.6, the function \(|V(\cdot)|\) is integer-valued, nondecreasing, and submodular. By Lemma 2.3, we see that \( \varrho_{k,l} \) is an integer-valued nondecreasing submodular function as well, and it thus induces the matroid \( M_{\varrho_{k,l}} \). An independent set and a base of \( M_{\varrho_{k,l}} \) are called \((k, l)\)-sparse and \((k, l)\)-tight, respectively, in e.g. [37, 80, 112].

For example, let us consider \( \varrho_{1,1}(F) = |V(F)| - 1 \). \tag{2.9}

It can be seen that \( \varrho_{1,1} \) induces the graphic matroid of the graph \( G \) (introduced in Example 2.3) as follows. If \( F \subseteq E \) is independent in the graphic matroid, then \(|F| = |V(F)| - \kappa(F) \leq |V(F)| - 1 = \varrho_{1,1}(F)\), and hence \( F \) is independent in \( M_{\varrho_{1,1}} \). On the other hand, if \( F \) is dependent in the graphic matroid, then there is a cycle \( C \) contained in \( F \). This implies that there is an edge subset \( C \subseteq F \) with \(|C| = |V(C)| > \varrho_{1,1}(C)\). Thus, \( F \) is dependent in \( M_{\varrho_{1,1}} \) as well.

Another well-known example is

\[
\varrho_{1,0}(F) = |V(F)|. \tag{2.10}
\]

The matroid \( M_{\varrho_{1,0}} \) is known as the bicircular matroid. It can be seen that \( F \subseteq E \) is independent in \( M_{\varrho_{1,0}} \) if and only if each connected component of \((V, F)\) contains at most one cycle. Such a graph is called a pseudoforeset [42, 43].

An important example in our context is the case of \( k = 2 \) and \( l = 3 \). Let us consider the set function \( \varrho_{2,3} \) on the edge set of the complete graph \( K(V) \), that is,

\[
\varrho_{2,3}(F) = 2|V(F)| - 3, \tag{2.11}
\]

for any \( F \subseteq K(V) \). Then, Laman’s theorem (Theorem 1.1) asserts that, on a generic joint configuration \( p \), a bar-and-joint framework \((G = (V, F), p)\) is isostatic in 2-dimensional space if and only if \( E \) is a base of \( M_{\varrho_{2,3}} \). Equivalently, \((G, p)\) is rigid if and only if \( E \) is a spanning set of \( M_{\varrho_{2,3}} \). This matroid \( M_{\varrho_{2,3}} \) will be referred to as the generic rigidity matroid in Chapter 3.
Rank functions of induced matroids. For a finite set $F$, we shall denote by $\mathcal{P}(F)$ the collection of all possible partitions $\{F_0, F_1, \ldots, F_m\}$ of $F$ for some integer $m$ with $0 \leq m \leq |F|$ such that $F_i \neq \emptyset$ for each $i = 1, \ldots, m$ (and $F_0$ may be empty). The following proposition is the so-called Dilworth truncation specialized to matroids (i.e., the rank function is restricted to the unit hypercube), see, e.g., [104, Section 48]. We shall provide a direct proof for the completeness.

**Proposition 2.5.** Let $f$ be an integer-valued nondecreasing submodular function on $E$ satisfying $f(F) \geq 0$ for any nonempty $F \subseteq E$. Then, for any nonempty $F \subseteq E$, the rank $r_f(F)$ of $F$ in $\mathcal{M}_f$ is given by

$$r_f(F) = \min\{|F_0| + \sum_{i=1}^m f(F_i) : \{F_0, F_1, \ldots, F_m\} \in \mathcal{P}(F)\}. \quad (2.12)$$

**Proof.** For the proof, we need rather general two observations, Claim 2.6 and Claim 2.7.

For a matroid $\mathcal{M}$ and a base $B$ of $\mathcal{M}$, define an undirected bipartite graph $\Delta_{\mathcal{M},B}$ as follows: one vertex class of $\Delta_{\mathcal{M},B}$ is $B$ and the other is $E \setminus B$. Each $e \in E \setminus B$ is joined to $e' \in B$ in $\Delta_{\mathcal{M},B}$ if and only if $e' \in C(B,e)$.

**Claim 2.6.** Let $\mathcal{M}$ be an $M$-connected matroid on at least two elements and $B$ be a base of $\mathcal{M}$. Then, $\Delta_{\mathcal{M},B}$ is connected.

**Proof.** Let $E$ be the ground set of $B$. We remark that, since $\mathcal{M}$ is $M$-connected with at least two elements, no loop (i.e., a dependent singleton) exists in $\mathcal{M}$ and both $B$ and $E \setminus B$ are nonempty.

Suppose, for a contradiction, that $\Delta_{\mathcal{M},B}$ is disconnected. Since $\mathcal{M}$ is $M$-connected, there exists some circuit that intersects more than one connected component in the graph $\Delta_{\mathcal{M},B}$. We shall take such a circuit $C_1$ so that

$$|C_1 \cap (E \setminus B)|$$

is minimized. Note that, since $B$ is independent, $C_1$ contains at least one element of $E \setminus B$. Let $e$ be such an element, and let $F \subseteq E$ be (the vertex set of) the connected component of $\Delta_{\mathcal{M},B}$ that contains $e$. From the definition of $C_1$, $C_1 \notin F$ holds. Hence we can take an element $f \in C_1 \setminus F$. There are two cases depending on $f \in B$ or $f \in E \setminus B$.

Suppose $f \in B$. Let us apply Lemma 2.2 to $C_1$ and $C(B,e)$, where $e \in C_1 \cap C(B,e)$ and $f \in C_1 \setminus C(B,e)$ hold from $C(B,e) \subseteq F$. There exists a circuit $C_2 \subseteq (C_1 \cap C(B,e)) - e$ with $f \in C_2$. Notice that $C_2 \cap F \neq \emptyset$ holds by $C_2 \cap C(B,e) \neq \emptyset$ and $C_2 \cap (E \setminus F) \neq \emptyset$ holds by $f \in C_2$. Thus $C_2$ is a circuit that intersects more than one connected component in $\Delta_{\mathcal{M},B}$. Notice also that $|C_2 \cap (E \setminus B)| < |C_1 \cap (E \setminus B)|$ holds since $C(B,e) - e \subseteq B$ and $e \notin C_2$. Therefore, the existence of $C_2$ contradicts the choice of $C_1$.

Suppose $f \in E \setminus B$. Recall $f \notin F$. The definition of $\Delta_{\mathcal{M},B}$ and $F$ hence implies $C(B,f) \cap F = \emptyset$. Applying Lemma 2.2 to $C_1$ and $C(B,f)$, (where $f \in C_1 \cap C(B,f)$ and $e \in C_1 \setminus C(B,f)$), we obtain a circuit $C_3 \subseteq (C_1 \cap C(B,f)) - f$ with $e \in C_3$. Notice that $C_3 \cap F \neq \emptyset$ and $C_3 \cap (E \setminus F) \neq \emptyset$ by $e \in C_3$ and $C_3 \cap C(B,f) \neq \emptyset$. Notice, also, that $|C_3 \cap (E \setminus B)| <
|C_1 \cap (E \setminus B)| holds since C(B, f) − f ⊂ B and f is eliminated in C_3. The existence of C_3 contradicts the choice of C_1.

As a corollary of Claim 2.6, we obtain the following lemma.

**Claim 2.7.** Let M be an M-connected matroid on E consisting of at least two elements, and let B be a basis of M. Let \( I_e = C(B, e) - e \) for each \( e \in E \setminus B \). Then, there exists a sequence \( e_1, e_2, \ldots, e_m \) of the elements of \( E \setminus B \) satisfying \( \bigcup_{i=1}^{k} I_{e_i} \cap I_{e_{k+1}} \neq \emptyset \) for each \( k = 1, \ldots, m - 1 \), where \( m = |E \setminus B| \).

**Proof.** Note that, for each \( e \in E \setminus B \), the neighbor \( N(e) \) of \( e \) in \( \Delta_{M,B} \) is equal to \( I_e \) according to the definition of \( \Delta_{M,B} \). Let \( e_1 \) be an arbitrary element in \( E \setminus B \), and we choose the elements of \( E \setminus B \) one by one in an arbitrary order so that a sequence of elements \( e_1, e_2, \ldots, e_m \) satisfies \( \bigcup_{i=1}^{k} I_{e_i} \cap I_{e_{k+1}} \neq \emptyset \) from \( k = 1 \) through \( m - 1 \). Suppose, for the contradiction, that this process gets stuck after choosing \( k \)-th element of \( E \setminus B \) with \( k < m - 1 \). Then, there exists no edge between \( E \setminus (B \cup \{e_1, \ldots, e_k\}) \) and the neighbor \( N(\{e_1, \ldots, e_k\}) \) of \( \{e_1, \ldots, e_k\} \) in \( \Delta_{M,B} \) by \( N(\{e_1, \ldots, e_k\}) = \bigcup_{i=1}^{k} I_{e_i} \). This implies that \( \Delta_{M,B} \) is disconnected, contradicting Claim 2.6.

Now we are ready to show Proposition 2.5. (The proof strategy is based on that given in [95, Proposition 12.1.7].) For a nonempty subset \( F \subseteq E \), let \( g(F) = \min\{|F_0| + \sum_{i=1}^{m} f(F_i)|\} \), where the minimum is taken over all \( \{F_0, \ldots, F_m\} \in \mathcal{P}(F) \). Our goal is to prove \( r_f(F) = g(F) \).

We first show that \( g \) is nondecreasing. Consider two nonempty subsets \( F \) and \( F' \) of \( E \) with \( F' \subseteq F \). Let \( \{F_0, \ldots, F_m\} \) be a partition of \( F \) that takes the minimum value of \( |F_0| + \sum_{i=1}^{m} f(F_i)| \). Let \( F'_i = F_i \cap F' \) for \( i = 0, \ldots, m \). Then, \( \{F'_0, \ldots, F'_m\} \) is a partition of \( F' \) satisfying \( g(F') \leq |F'_0| + \sum_{i=1}^{m} f(F'_i) \leq |F_0| + \sum_{i=1}^{m} f(F_i) = g(F) \) since \( f \) is nondecreasing.

Let us consider the case when \( F \) is independent in \( M_f \). Since we have \( |F'| \leq f(F') \) for every nonempty \( F' \subseteq F \), \( |F| = \sum_{i=0}^{m} |F_i| \leq |F_0| + \sum_{i=1}^{m} f(F_i) \) holds for any \( \{F_0, \ldots, F_m\} \in \mathcal{P}(F) \), implying \( |F| \leq g(F) \). We also have \( g(F) \leq |F| \) since \( \{F(= F_0)\} \) is also a partition of \( F \) with \( m = 0 \). As a result, we obtain \( g(F) = |F| = r_f(F) \).

Thus, let us consider the case when \( F \) is not independent in \( M_f \). Let \( B \) be a base of \( M_f|F \). Since \( g \) is nondecreasing, we have

\[ r_f(F) = |B| = r_f(B) = g(B) \leq g(F), \quad (2.13) \]

where \( r_f(B) = g(B) \) follows from the independence of \( B \). Let us show the converse direction. Let \( X_1, \ldots, X_l \) be the M-connected components of \( M_f|F \). We shall verify

\[ \min\{|X_j|, f(X_j)|\} \leq r_f(X_j) \quad (2.14) \]

by splitting the argument into three cases.

Case 1: If \(|X_j| = 1\) and \( X_j \) is independent in \( M_f \), then it is obvious that \(|X_j| = 1 = r_f(X_j)|\).

Case 2: If \(|X_j| = 1\) and \( X_j \) is a loop in \( M_f \), then \( f(X_j) < |X_j| = 1 \) holds since \( X_j \) is a circuit. Thus, since \( f \) is integer-valued, we obtain \( f(X_j) = r_f(X_j) = 0 \).
Case 3: \(|X_j| \geq 2\). Note that \(\mathcal{M}_f|X_j\) is M-connected. Let \(B\) be a base of \(\mathcal{M}_f|X_j\). By Claim 2.7, there exists a sequence \(e_1, e_2, \ldots, e_m\) of \(X_j \setminus B\) satisfying \((\bigcup_{i=1}^k I_{e_i}) \cap I_{e_{k+1}} \neq \emptyset\) for each \(k = 1, \ldots, m - 1\), where \(m = |X_j \setminus B|\) and \(I_{e_i} = C(B, e_i) - e_i\). We shall show, by induction, that \(f((\bigcup_{i=1}^k I_{e_i}) \cup \{e_1, \ldots, e_k\}) \leq |\bigcup_{i=1}^k I_{e_i}|\) holds.

Let us consider the base case. Since \(I_{e_1} + e_1 = C(B, e_1)\) is a circuit, we have \(f(I_{e_1} + e_1) < |I_{e_1} + e_1| = |I_{e_1}| + 1\). Since \(I_{e_1}\) is independent, we also have \(|I_{e_1}| \leq f(I_{e_1})\). Combining them with \(f(I_{e_1}) \leq f(I_{e_1} + e_1)\), we obtain \(f(I_{e_1} + e_1) = |I_{e_1}|\). Applying the same argument, we also have \(f(I_{e_k} \cup e_k) = |I_{e_k}|\) for each \(k = 1, \ldots, m\).

Let us consider the case when \(k > 1\). By the submodularity of \(f\), we have

\[
\begin{align*}
f((\bigcup_{i=1}^{k+1} I_{e_i}) \cup \{e_1, \ldots, e_{k+1}\}) \\
\leq f((\bigcup_{i=1}^k I_{e_i}) \cup \{e_1, \ldots, e_k\}) + f(I_{e_{k+1}} + e_{k+1}) - f((\bigcup_{i=1}^k I_{e_i}) \cap I_{e_{k+1}}) \\
\leq |\bigcup_{i=1}^k I_{e_i}| + |I_{e_{k+1}}| - |(\bigcup_{i=1}^k I_{e_i}) \cap I_{e_{k+1}}| \\
= |\bigcup_{i=1}^{k+1} I_{e_i}|,
\end{align*}
\]

where we apply the induction hypothesis, \(f(I_{e_{k+1}} + e_{k+1}) = |I_{e_k}|\), and also \(|(\bigcup_{i=1}^k I_{e_i}) \cap I_{e_{k+1}}| \leq f((\bigcup_{i=1}^k I_{e_i}) \cap I_{e_{k+1}})\), which is obtained from the fact that \((\bigcup_{i=1}^k I_{e_i}) \cap I_{e_{k+1}}\) is a nonempty independent set of \(\mathcal{M}_f\). As a result, we obtain \(f(X_j) = f((\bigcup_{i=1}^m I_{e_i}) \cup \{e_1, \ldots, e_m\}) \leq |\bigcup_{i=1}^m I_{e_i}| \leq |B| = r_f(X_j)\), and we have verified (2.14) for each \(j = 1, \ldots, l\). Consequently, we obtain

\[
\begin{align*}
\rho_f(F) &= \sum_{j=1}^l r_f(X_j) \quad \text{(by (2.3))} \\
&\geq \sum_{j=1}^l \min\{|X_j|, f(X_j)| \quad \text{(by (2.14))} \\
&\geq \min\{\sum_{j'=1}^{l'} \min\{|F_{j'}|, f(F_{j'})| \quad \text{a partition } \{F_1, \ldots, F_{l'}\} \text{ of } F\} \\
&= g(F).
\end{align*}
\]

By (2.13) and (2.15), the proof is completed. \(\square\)

The following lemmas characterize a partition of an edge (sub)set for which the minimum of (2.12) is achieved in a graphic matroid or a bicircular matroid (introduced in Example 2.7). This implies that the rank of a graphic matroid or a bicircular matroid can be computed without checking all the partitions of (2.12).

**Lemma 2.8.** Let \(G = (V, E)\) be a graph, and \(\mathcal{M}_{g_{1,1}}\) be the graphic matroid on \(E\). Let \(m\) be the total number of connected components of \(G[F] = (V(F), F)\) and \(\{F_1, \ldots, F_m\}\) be a partition of \(F\) such that, for each \(j = 1, \ldots, m, F_j\) is the edge set of a connected component of \(G[F]\). Also, let \(F_0 = \emptyset\). Then, \(\{F_0, F_1, \ldots, F_m\} \in \mathcal{P}(F)\) takes the minimum value of (2.12). Namely, the rank of \(F\) can be written as \(\rho_{g_{1,1}}(F) = \sum_{i=1}^m g_{1,1}(F_i)\).

**Proof.** Let \(B_F\) be a base of \(F\), and let \(B_{F_i}\) be a base of \(F_i\) for each \(i = 1, \ldots, m\). Since \(F_i\) is connected, \(B_{F_i}\) is a spanning tree on \(V(F_i)\). Therefore, we have

\[
\rho_{g_{1,1}}(F) = |B_F| = \sum_{i=1}^m |B_{F_i}| = \sum_{i=1}^m (|V(F_i)| - 1) = \sum_{i=1}^m g_{1,1}(F_i).
\]

\(\square\)
Lemma 2.9. Let $G = (V, E)$ be a graph, and $\mathcal{M}_{1,0}$ be the bicircular matroid on $E$. Let \{${F_0, F_1, \ldots, F_m}$\} $\in \mathcal{P}(F)$ be a partition of $F$ as in the statement of Lemma 2.8. Then it takes the minimum value of (2.12). Namely, the rank of $F$ can be written as $r_{1,0}(F) = \sum_{i=1}^{m} \varrho_{1,0}(F_i)$.

Proof. The proof is basically the same as Lemma 2.9, and hence is omitted. □

2.2.4 Matroid union

For two matroids $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$ on the same ground set $E$, the matroid union $\mathcal{M}_1 \lor \mathcal{M}_2$ is defined by

$$\mathcal{M}_1 \lor \mathcal{M}_2 = (E, \{I_1 \cup I_2 : I_1 \in \mathcal{I}_1, I_2 \in \mathcal{I}_2\}).$$ (2.16)

Note that \{${I_1 \cup I_2 : I_1 \in \mathcal{I}_1, I_2 \in \mathcal{I}_2}$\} is the collection of the subsets of $E$ that can be partitioned into independent sets of $\mathcal{I}_1$ and $\mathcal{I}_2$. It is easy to see that $\mathcal{M}_1 \lor \mathcal{M}_2$ is again a matroid. It is known that the rank function of $\mathcal{M}_1 \lor \mathcal{M}_2$ is expressed by

$$r_{\mathcal{M}_1 \lor \mathcal{M}_2}(F) = \min\{r_{\mathcal{M}_1}(X) + r_{\mathcal{M}_2}(X) + |F \setminus X| : X \subseteq F\}. \tag{2.17}$$

For the proof, see e.g., [95, Theorem 12.3.1].

Consider the matroid union of $\mathcal{M}_f$ and $\mathcal{M}_g$ induced by integer-valued nondecreasing submodular functions $f$ and $g$ on $E$. Pym and Perfect [97] proved that $\mathcal{M}_f \lor \mathcal{M}_g$ is the matroid induced by the submodular function $f + g$, i.e., $\mathcal{M}_f \lor \mathcal{M}_g = \mathcal{M}_{f+g}$, if $f(F) \geq 0$ and $g(F) \geq 0$ hold for every $F \subseteq E$ including $\emptyset$. Whiteley further claimed $\mathcal{M}_f \lor \mathcal{M}_g = \mathcal{M}_{f+g}$ in [129] even in the case of $f(\emptyset) < 0$ or $g(\emptyset) < 0$. Although this statement is true for the union of the same matroids, say the union of graphic matroids, Jordán pointed out that this is not always true in general. Although $\mathcal{M}_f \lor \mathcal{M}_g = \mathcal{M}_{f+g}$ may not hold in general in the case of $f(\emptyset) < 0$ or $g(\emptyset) < 0$, we show a sufficient condition for that equality to be true.

Lemma 2.10. Let $f$ and $g$ be integer-valued nondecreasing submodular functions on $E$ satisfying $f(F) \geq 0$ and $g(F) \geq 0$ for every nonempty $F \subseteq E$. Then, $\mathcal{M}_f \lor \mathcal{M}_g = \mathcal{M}_{f+g}$ holds if, for any $F \subseteq E$, there exists a partition \{${F_0, F_1, \ldots, F_m}$\} $\in \mathcal{P}(F)$ that takes the minimum values of (2.12) for $r_f(F)$ and $r_g(F)$ simultaneously.

Proof. Let us consider $F \subseteq E$. Notice that, for any collection \{${F_1, \ldots, F_k}$\} of disjoint nonempty subsets of $F$, $\sum_{i=1}^{k} f(F_i) \geq r_f(\bigcup_{i=1}^{k} F_i)$ holds by Proposition 2.5. Hence, by (2.12) and (2.17), it is not difficult to see $r_{f+g}(F) \geq r_{\mathcal{M}_f \lor \mathcal{M}_g}(F)$ as follows:

$$r_{f+g}(F) = \min\{\sum_{i=1}^{m} (f(F_i) + g(F_i)) + |F_0| : \{F_0, \ldots, F_m\} \in \mathcal{P}(F)\}$$

$$\geq \min\{r_f(\bigcup_{i=1}^{m} F_i) + r_g(\bigcup_{i=1}^{m} F_i) + |F_0| : \{F_0, \ldots, F_m\} \in \mathcal{P}(F)\}$$

$$\geq r_{\mathcal{M}_f \lor \mathcal{M}_g}(F).$$

Let us show the converse direction. For $F \subseteq E$, let $X$ be the subset of $F$ that takes the minimum value of (2.17) for $r_{\mathcal{M}_f \lor \mathcal{M}_g}(F)$. By the assumption of the statement, there exists a
partition \( \{X_0, X_1, \ldots, X_m\} \in \mathcal{P}(X) \) that takes the minimum values of (2.12) for both \( r_f(X) \) and \( r_g(X) \) simultaneously (where \( X_i \neq \emptyset \) for each \( i = 1, \ldots, m \)). We claim \( X_0 = \emptyset \) since otherwise

\[
\begin{align*}
r_f(X) + r_g(X) + |F \setminus X| &= \sum_{i=1}^{m} (f(X_i) + g(X_i)) + 2|X_0| + |F \setminus X| \\
&= \sum_{i=1}^{m} (f(X_i) + g(X_i)) + |X_0| + |F \setminus (X \setminus X_0)| \\
&> \sum_{i=1}^{m} (f(X_i) + g(X_i)) + |F \setminus (X \setminus X_0)| \\
&\geq r_f(\bigcup_{i=1}^{m} X_i) + r_g(\bigcup_{i=1}^{m} X_i) + |F \setminus (X \setminus X_0)| \\
&= r_f(X \setminus X_0) + r_g(X \setminus X_0) + |F \setminus (X \setminus X_0)| \\
&\geq r_{\mathcal{M}_f \vee \mathcal{M}_g}(F),
\end{align*}
\]

which contradicts that \( X \) takes the minimum value of (2.17) for \( r_{\mathcal{M}_f \vee \mathcal{M}_g}(F) \). Since \( \{X_0, X_1, \ldots, X_m\} \) takes the minimum value of (2.12) with \( X_0 = \emptyset \), we obtain \( r_f(X) + r_g(X) = \sum_{i=1}^{m} (f(X_i) + g(X_i)) = \sum_{i=1}^{m} (f + g)(X_i) \geq r_{f+g}(X) \). Thus, by (2.12) and (2.17), we have

\[
\begin{align*}
r_{\mathcal{M}_f \vee \mathcal{M}_g}(F) &= |F \setminus X| + r_f(X) + r_g(X) \\
&\geq |F \setminus X| + r_{f+g}(X) \\
&= |F \setminus X| + \min\{|X_0| + \sum_{i=1}^{m'} (f + g)(X'_i) : \{X_0', \ldots, X_{m'}\} \in \mathcal{P}(X)\} \\
&\geq r_{f+g}(F).
\end{align*}
\]

\[\square\]

In particular, Lemma 2.10 implies \( \mathcal{M}_{2f} = \mathcal{M}_f \vee \mathcal{M}_f \) for any integer-valued nondecreasing submodular function \( f \).

**Example 2.8.** Using Lemma 2.10, we now easily derive a Tutte-Nash-Williams tree-packing theorem [89, 118], which is one of fundamental results in graph theory and will be utilized even in the context of rigidity theory. The following is of a slightly generalized form proposed by Whiteley [129].

**Theorem 2.11.** Let \( G = (V, E) \) be a (multi)graph, and let \( k \) and \( l \) be positive integers with \( 0 \leq l \leq k \). Then, \( E \) can be partitioned into edge-disjoint \( l \) spanning trees and \( k - l \) spanning pseudoforests if and only if \( E \) is (\( k, l \))-tight, i.e.,

- \( |E| = k|V| - l \),
- \( |F| \leq k|V(F)| - l \) for any nonempty \( F \subseteq E \).

**Proof.** \( E \) can be partitioned into edge-disjoint \( l \) spanning trees and \( k - l \) spanning pseudoforests if and only if \( E \) is a base of \( (\bigvee_{i=1}^{l} \mathcal{M}_{g_{i,1}}) \vee (\bigvee_{i=1}^{k-l} \mathcal{M}_{g_{i,0}}) \). Lemmas 2.8 and 2.9 imply that for any \( F \subseteq E \) there exists a partition of \( \mathcal{P}(F) \) that takes the minimum values of (2.12) for \( r_{g_{i,1}}(F) \) and \( r_{g_{i,0}}(F) \) simultaneously, and hence Lemma 2.10 can apply to the union of the graphic matroid and the bicircular matroid. Namely, we have

\[
(\bigvee_{i=1}^{l} \mathcal{M}_{g_{i,1}}) \vee (\bigvee_{i=1}^{k-l} \mathcal{M}_{g_{i,0}}) = \mathcal{M}_{l_{g_{1,1}} + (k-l)_{g_{1,0}}}.
\]

From \( l_{g_{1,1}}(F) + (k - l)_{g_{1,0}}(F) = k|V(F)| - l \), the statement follows. \( \square \)
For polynomial time algorithms that check the conditions of Theorem 2.11 or compute a decomposition explicitly, see, e.g., Imai [55], Gabow and Westermann [43], Lee and Streinu [80], or Streinu and Theran [112].
Chapter 3

Rigidity Theory

In this chapter, we discuss definitions and basic results of rigidity theory: bar-and-joint frameworks (Section 3.1) and body-and-hinge frameworks (Section 3.2). All contents of this chapter are known results, but it seems worth for providing the proofs not only for a self-contained introduction but also for the completeness since there are few expositions that covers all of these contents.

Throughout this chapter, $d$ denotes a positive integer.

3.1 Bar-and-joint Frameworks

This section is devoted to bar-and-joint frameworks. For more detail, see e.g., [26, 45, 129].

3.1.1 Rigidity

A $d$-dimensional bar-and-joint framework is defined as a pair $(G, p)$, where $G = (V, E)$ is an undirected graph having neither loops nor multiple edges and $p$ is a mapping from $V$ to $\mathbb{R}^d$, called a joint configuration. Namely, each vertex and each edge are regarded as a universal joint and a rigid bar connecting two joints, respectively, and each joint is allowed to move continuously keeping the lengths of the bars. See figures given in the introduction for examples.

The space of joint configurations, called the configuration space, is the collection of mappings from $V$ to $\mathbb{R}^d$, and it forms a real vector space of dimension $d|V|$. Hence, we may think of $p$ as a point in $\mathbb{R}^{d|V|}$. In other words, we identify $p$ with the $d|V|$-tuple of real numbers.

Two bar-and-joint frameworks $(G, p)$ and $(G, q)$ are called equivalent if

$$||p(u) - p(v)|| = ||q(u) - q(v)||$$

for any $uv \in E$, (3.1)

and they are called congruent if

$$||p(u) - p(v)|| = ||q(u) - q(v)||$$

for any $u, v \in V$. (3.2)

Clearly, the congruence implies the equivalence. A bar-and-joint framework $(G, p)$ is called rigid if the equivalence implies the congruence in a neighborhood of $p$. Formally, $(G, p)$
is defined to be \textit{rigid} if there is an \( \varepsilon > 0 \) such that every bar-and-joint framework \((G, q)\) equivalent to \((G, p)\) with \(|p - q| < \varepsilon\) is congruent to \((G, p)\). A framework is called \textit{flexible} if it is not \textit{rigid}. A rigid framework is called \textit{minimally rigid} or \textit{isostatic} if removing any bar results in a flexible framework.

The above definition of the rigidity does not concern with “motion” of frameworks. We have however (informally) introduced the concept of the rigidity in terms of the continuous motion in the introduction. This gives rise to another definition of the rigidity; a \(d\)-dimensional bar-and-joint framework \((G, p)\) is \textit{rigid} if, for every continuous path \(p_t \in \mathbb{R}^d|V|\) such that \((G, p_t)\) is equivalent to \((G, p)\) for all \(0 \leq t < \varepsilon\) with \(\varepsilon > 0\) and \(p_0 = p\), \((G, p_t)\) is congruent to \((G, p)\) for all \(0 \leq t < \varepsilon\). It is known that these two definitions of the rigidity are equivalent (see, e.g., [132, Theorem 60.1.30] or [26]).

### 3.1.2 Infinitesimal rigidity

**Infinitesimal congruence.** We say that \(\tilde{v} : \mathbb{R}^d \to \mathbb{R}^d\) is an \textit{infinitesimal congruence} of \(\mathbb{R}^d\) if

\[
(p - q) \cdot (\tilde{v}(p) - \tilde{v}(q)) = 0
\]

for any \(p, q \in \mathbb{R}^d\). (3.3)

In 2-dimensional space, typical examples of infinitesimal congruence are
\[
\begin{align*}
\tilde{v}_x : q \in \mathbb{R}^2 & \mapsto (1, 0) \\
\tilde{v}_y : q \in \mathbb{R}^2 & \mapsto (0, 1) \\
\tilde{v}_r : q = (q_x, q_y) \in \mathbb{R}^2 & \mapsto (q_y, -q_x),
\end{align*}
\]

which represent the translations along \(x\)-axis and \(y\)-axis and the rotation around the origin. In general dimension \(d\), typical examples are \(d\) translations along each axis and \(\binom{d}{2}\) rotations around \((d - 2)\)-affine subspaces.

Clearly the set of infinitesimal congruences of \(\mathbb{R}^d\) forms a vector space. We denote this space by \(\mathcal{V}^d\). It is well known that \(\tilde{v} : \mathbb{R}^d \to \mathbb{R}^d\) is an infinitesimal congruence if and only if it can be expressed as

\[
\tilde{v}(p) = Sp + p_0 \quad \text{for } p \in \mathbb{R}^d,
\]

where \(S\) is a skew symmetric matrix (i.e. \(S^\top = -S\)) and \(p_0 \in \mathbb{R}^d\). Indeed, one direction is easily observed; if \(\tilde{v}\) is written as (3.4), then for any \(p, q \in \mathbb{R}^d\)

\[
(p - q) \cdot (\tilde{v}(p) - \tilde{v}(q)) = (p - q) \cdot (Sp - Sq) = (p - q)S(p - q)^\top = 0,
\]

and hence \(\tilde{v}\) is an infinitesimal congruence. The proof of the converse direction can be seen in e.g., [26, Proposition 2.10].

Since the dimension of \(\{(S, p_0) : S \text{ is a } d \times d \text{ skew symmetric matrix, } p_0 \in \mathbb{R}^d\}\) is equal to \(\binom{d}{2} + d\), (3.4) implies the following well known fact.

**Proposition 3.1.**

\[
\dim \mathcal{V}^d = \binom{d+1}{2}. \tag{3.5}
\]
3.1. Bar-and-joint Frameworks

**Infinitesimal motions and rigidity matrices.** As we have mentioned in Chapter 1, taking the first order approximation is a common strategy of dealing with the distance constraints (3.1). An infinitesimal motion of a bar-and-joint framework \((G, p)\) is defined as an assignment \(v : V \to \mathbb{R}^d\) of a \(d\)-dimensional vector for each joint \(p(v)\) such that

\[
(p(v) - p(u)) \cdot (v(v) - v(u)) = 0 \quad \text{for each } uv \in E.
\]

(3.6)

Collecting the length constraints (3.6) for all \(e \in E\), we have a system of the \(|E|\) equations on the unknowns \(v(u), u \in V\). This system of linear equations is customarily written as \(R(G, p)v^T = 0\), where \(R(G, p)\) is a \(|E| \times d|V|\)-matrix, called the rigidity matrix of \((G, p)\). More precisely, \(R(G, p)\) is written as

\[
\cdots \quad u \quad \cdots \quad v \quad \cdots \\
\begin{pmatrix}
\vdots \\
0 \cdots p(u) - p(v) \cdots 0 \cdots p(v) - p(u) \cdots 0 \\
\vdots 
\end{pmatrix}
\]

(3.7)

where \(p(u) - p(v)\) (and \(p(v) - p(u)\), resp.) is regarded as a \(1 \times d\)-submatrix. Hence, \(v \in \mathbb{R}^{|d||V|}\) is an infinitesimal motion if and only if it is in the null space of \(R(G, p)\).

**Example 3.1.** Let us take a look at the 2-dimensional bar-and-joint framework \((G, p)\) shown in Figure 3.1(a). The rigidity matrix \(R(G, p)\) is written as

\[
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
p_x^2 - p_y^2 & p_x^1 - p_y^1 & p_x^2 - p_y^2 & p_x^1 - p_y^1 & 0 \\
p_y^2 - p_x^2 & p_x^3 - p_y^3 & p_y^2 - p_x^2 & p_x^3 - p_y^3 & 0 \\
p_x^3 - p_y^3 & p_y^4 - p_x^4 & p_x^3 - p_y^3 & p_y^4 - p_x^4 & 0 \\
p_y^4 - p_x^4 & p_x^3 - p_y^3 & p_y^4 - p_x^4 & p_x^3 - p_y^3 & 0 \\
p_x^3 - p_y^3 & p_y^4 - p_x^4 & p_x^3 - p_y^3 & p_y^4 - p_x^4 & 0
\end{pmatrix}
\]

Hence, if the joint configuration has the coordinates as Figure 3.1(b), the rigidity matrix becomes

\[
\begin{pmatrix}
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\
-1 & -1 & 0 & 0 & 1 & 1 & 0 & 0
\end{pmatrix}
\]

**Infinitesimal rigidity.** Let \(G = (V, E)\) be a graph and \((G, p)\) be a \(d\)-dimensional bar-and-joint framework. For an infinitesimal congruence \(\tilde{v} : \mathbb{R}^d \to \mathbb{R}^d\), we consider the restriction of \(\tilde{v}\) to \(\{p(u) : u \in V\}\), which is defined as \(v_0 : V \to \mathbb{R}^d\) with \(v_0(u) = \tilde{v}(p(u))\) for each \(u \in V\). It is obvious that \(v_0\) is an infinitesimal motion of \((G, p)\). An infinitesimal motion \(v\) is called trivial if it is obtained by the restriction of some infinitesimal congruence of \(\mathbb{R}^d\). We say that \((G, p)\) is infinitesimally rigid if any infinitesimal motion of \((G, p)\) is trivial.
Figure 3.1: 2-dimensional bar-and-joint frameworks.

Let us denote the space of infinitesimal motions of \((G, p)\) by \(V(G, p)\), that is, the null space of \(R(G, p)\), and let \(V_0(G, p)\) denote the space of the trivial infinitesimal motions. From the definition, we have \(V_0(G, p) \subseteq V(G, p)\), and \((G, p)\) is infinitesimally rigid if and only if they coincide.

**Proposition 3.2** (See e.g., [26, 45]). Let \((G, p)\) be a \(d\)-dimensional bar-and-joint framework, and let \(d_0 (0 \leq d_0 \leq d)\) be the dimension of the affine span of \(\{p(u) : u \in V\}\). Then, we have

\[
\dim V_0(G, p) = \begin{cases} 
\frac{(d+1)}{2} - \frac{(d-d_0)}{2} & \text{if } d_0 < d - 1, \\
\frac{(d+1)}{2} & \text{if } d_0 = d - 1 \text{ or } d.
\end{cases}
\] (3.8)

**Proof.** Let us consider \(\varphi : V^d \rightarrow V_0(G, p)\) that maps an infinitesimal congruence \(\tilde{v}\) to the restriction \(v_0\) of \(\tilde{v}\) onto \(\{p(u) : u \in V\}\). It is easy to check that \(\varphi\) is a linear map. Moreover, we have the following fact;

**Claim 3.3.** If \(d_0 = d - 1\) or \(d\), then \(\varphi\) is an isomorphism.

**Proof.** It is clear that \(\varphi\) is onto since any trivial infinitesimal motion of \(V(G, p)\) is (defined to be) a restriction of some infinitesimal congruence. To see that \(\varphi\) is one-to-one, suppose for a contradiction, that there exists a trivial infinitesimal motion \(v\) of \((G, p)\) which can be extended to two distinct infinitesimal congruences \(\tilde{v}_1\) and \(\tilde{v}_2\) of \(\mathbb{R}^d\). Let us define an infinitesimal congruence \(\tilde{v}\) by \(\tilde{v} = \tilde{v}_1 - \tilde{v}_2\). Since \(\tilde{v}_i\) is an extension of \(v\) for each \(i = 1, 2\), we have

\[
\tilde{v}(p(u)) = \tilde{v}_1(p(u)) - \tilde{v}_2(p(u)) = v(p(u)) - v(p(u)) = 0
\] (3.9)

for any \(u \in V\). Let us take \(d\) points among \(\{p(u) : u \in V\}\) which are affinely independent, and denote them by \(p_1, \ldots, p_d\). Let \(H\) be the hyperplane spanned by \(p_1, \ldots, p_d\).

We shall first show that \(\tilde{v}(q) = 0\) for any \(q \in \mathbb{R}^d \setminus H\). The definition of an infinitesimal congruence says that

\[
(q - p_i) \cdot (\tilde{v}(q) - \tilde{v}(p_i)) = 0 \quad \text{for each } i = 1, \ldots, d.
\]

Substituting \(\tilde{v}(p_i) = 0\) (shown in (3.9)), we obtain

\[
(q - p_i) \cdot \tilde{v}(q) = 0 \quad \text{for each } i = 1, \ldots, d.
\] (3.10)
When looking at $\tilde{v}(q)$ as variables, this is a system of $d$ linear equations with $d$ unknowns. Note that $q, p_1, \ldots, p_d$ are affinely independent since $q \in \mathbb{R}^d \setminus H$, and hence $q - p_1, q - p_2, \ldots, q - p_d$ are linearly independent. Thus, (3.10) implies $\tilde{v}(q) = 0$.

Therefore, since $\tilde{v}_1 \neq \tilde{v}_2$, there is a $q \in H$ such that $v(q) \neq 0$. Since $p_1, \ldots, p_d$ are affinely independent, we may assume that $q, p_1, \ldots, p_{d-1}$ are affinely independent. Again, by the definition of infinitesimal congruences, we have

$$(q - p_i) \cdot \tilde{v}(q) = 0 \quad \text{for each } i = 1, \ldots, d - 1.$$  

Since $q - p_1, \ldots, q - p_{d-1}$ are linearly independent, $\tilde{v}(q)$ must be orthogonal to $H$. However, taking any point $q' \in \mathbb{R}^d \setminus H$ for which $q - q'$ is not orthogonal to $\tilde{v}(q)$, we have $(q - q') \cdot (\tilde{v}(q) - \tilde{v}(q')) = (q - q') \cdot \tilde{v}(q) \neq 0$, which contradicts that $\tilde{v}$ is an infinitesimal congruence.

By Claim 3.3, we found that $\mathcal{V}_0(G, p)$ is isomorphic to $\mathcal{V}^d$. This implies $\dim \mathcal{V}_0(G, p) = \dim \mathcal{V}^d = \binom{d+1}{2}$ by Proposition 3.1.

Let us consider the case of $d_0 < d - 1$. In this case, it can be seen that the dimension of the kernel space of $\varphi$ is equal to $\binom{d-d_0}{d-d_0-2}$ as follows. Let $A$ be the affine space spanned by $\{p(u) : u \in V\}$. Observe that an infinitesimal congruence $\tilde{v}$ is in the kernel space of $\varphi$ (i.e., the infinitesimal motion at each point $p(u)$ caused by $\tilde{v}$ is zero) if and only if $\tilde{v}$ is a linear combination of infinitesimal rotations of $\mathbb{R}^d$ about $(d - 2)$-affine subspaces containing $A$. The set of such infinitesimal rotations forms a $\binom{d-d_0}{d-d_0-2}$-dimensional vector space. From a fundamental result of linear algebra and Proposition 3.1, we obtain

$$
\dim \mathcal{V}_0(G, p) = \dim \mathcal{V}^d - \dim \ker \varphi
= \binom{d+1}{2} - \binom{d-d_0}{d-d_0-2}.
$$

The following statement gives us an equivalent but more useful formulation of the infinitesimal rigidity.

**Proposition 3.4** (See e.g. [26, 45]). A $d$-dimensional bar-and-joint framework $(G, p)$ is infinitesimally rigid if and only if

$$
\text{rank } R(G, p) = \begin{cases} 
\binom{|V|}{2} & \text{if } |V| \leq d + 1, \\
|V| - \binom{d+1}{2} & \text{if } |V| \geq d + 1.
\end{cases}
\quad (3.11)
$$

**Proof.** Let $d_0$ be the dimension of the affine span of $\{p(u) : u \in V\}$.

(i) Let us consider the case of $|V| \geq d + 1$. Suppose $d_0 < d$, i.e., all of joints of $p$ lie on a hyperplane $H$. Let $v_H \in \mathbb{R}^d$ be a nonzero vector perpendicular to $H$. Take any vertex $a \in V$ and define an infinitesimal motion $v : V \to \mathbb{R}^d$ such that $v(a) = v_H$ and $v(u) = 0$ for all $u \in V - a$.

**Claim 3.5.** $v$ is a nontrivial infinitesimal motion of $(G, p)$. 

By Claim 3.3, we obtained that $\mathcal{V}_0(G, p)$ is isomorphic to $\mathcal{V}^d$. This implies $\dim \mathcal{V}_0(G, p) = \dim \mathcal{V}^d = \binom{d+1}{2}$ by Proposition 3.1.
Proof. An example is illustrated in Figure 3.2. It is easy to see that \( v \) is an infinitesimal motion since \( v_H \) is orthogonal to \( H \) which contains all joints. To see that \( v \) is nontrivial, suppose for a contradiction that \( v \) can be extended to an infinitesimal congruence \( \tilde{v} \) of \( \mathbb{R}^d \). In the same way as in the proof of Proposition 3.2, it can be seen that \( \tilde{v}(p) = 0 \) must hold for any point \( p \) contained in the affine subspace \( A \) spanned by \( \{ p(u) : u \in V \} \). Namely, \( v \) is a linear combination of infinitesimal rotations around \((d - 2)\)-dimensional affine spaces containing \( A \) if \( d_0 \leq d - 2 \) while \( \tilde{v} \) is the identity (i.e., \( \tilde{v}(p) = 0 \) for any \( p \in \mathbb{R}^d \)) if \( d_0 = d - 1 \). In particular, \( \tilde{v}(a) = 0 \) contradicts the definition of \( v(a) \).

Thus, \( d_0 = d \) is necessary for the infinitesimal rigidity.

According to the definition, \((G, p)\) is infinitesimally rigid if and only if

\[
\mathcal{V}(G, p) = \mathcal{V}_0(G, p). \tag{3.12}
\]

Also, by the fundamental result of linear algebra, we have

\[
\text{rank } R(G, p) + \text{dim } \mathcal{V}(G, p) = d|V|. \tag{3.13}
\]

Hence, combining Proposition 3.2 and these equalities, we see that \((G, p)\) is infinitesimally rigid if and only if \( \text{rank } R(G, p) = d|V| - \left( \frac{d+1}{2} \right) \).

(ii) Let us consider the case of \( |V| \leq d + 1 \). If \( d_0 < |V| - 1 \) (i.e., \( \{ p(u) : u \in V \} \) is affinely dependent), it can be shown that \((G, p)\) has a nontrivial infinitesimal motion in the same way as (i), and hence \( d_0 = |V| - 1 \) is necessary. Combining (3.12), (3.13), and Proposition 3.2, we see that \((G, p)\) is infinitesimally rigid if and only if

\[
\text{rank } R(G, p) = d|V| - \left( \frac{d+1}{2} - \frac{|V| - 1}{2} \right) = \left( \frac{|V|}{2} \right).
\]

Proposition 3.4 implies the well-known Maxwell rule, which we have seen in the introduction.

**Corollary 3.6** (Maxwell’s rule). Let \((G, p)\) be a \( d \)-dimensional bar-and-joint framework with \( |V| \geq d + 1 \). If \((G, p)\) is infinitesimally rigid, then \( |E| \geq d|V| - \left( \frac{d+1}{2} \right) \).

We define the **degree of freedom** of \((G, p)\) as

\[
\text{dim } \mathcal{V}(G, p) - \text{dim } \mathcal{V}_0(G, p),
\]
and we say that \((G, p)\) has \(k\)-degree of freedom if the degree of freedom is equal to \(k\).

With a slight modification of the proof of Proposition 3.4, it is straightforward to obtain the following characterization.

**Proposition 3.7.** Let \(G = (V, E)\) be a graph and \(p\) be a joint configuration such that the affine span of \(\{p(u) : u \in V\}\) is equal to \(d\) or \(|V| - 1\). Then, the \(d\)-dimensional bar-and-joint framework \((G, p)\) has \(k\)-degree of freedom if and only if

\[
\text{rank } R(G, p) = \begin{cases} 
\binom{|V|}{2} - k & \text{if } |V| \leq d + 1, \\
|V| - \binom{d+1}{2} - k & \text{if } |V| \geq d + 1.
\end{cases}
\]  

(3.14)

**Rigidity matroids.** In Example 2.1, we have defined a linear matroid, that is, a matroid on the set of row vectors of a matrix. We define a rigidity matroid as a linear matroid specialized to the rigidity matrix \(R(G, p)\). It is however convenient to define a matroid on \(E\) rather than the set of row vectors. Recall that there is a one-to-one correspondence between an edge \(e \in E\) and a row vector of \(R(G, p)\) (since each row vector indicates the bar length constraint (3.7) by \(e\)). Hence, for a bar-and-joint framework \((G, p)\), the rigidity matroid \(R(G, p)\) is defined as a pair \((E, \mathcal{I})\), where \(F \subseteq E\) is in the independent set \(\mathcal{I}\) if and only if the set of row vectors of \(R(G, p)\) corresponding to \(F\) is linearly independent.

Let \((G, p)\) be a \(d\)-dimensional bar-and-joint framework with \(|V| \geq d\). Then, by Proposition 3.4, \((G, p)\) is isostatic if and only if \(E\) is independent in \(R(G, p)\) with \(|E| = d|V| - \binom{d+1}{2}\). More precisely, \((G, p)\) has \(k\)-degree of freedom if and only if a base of \(R(G, p)\) has the cardinality equal to \(|E| = d|V| - \binom{d+1}{2} - k\). It is hence important to investigate when an edge subset \(F \subseteq E\) becomes independent in \(R(G, p)\) to compute the degree of freedom of a bar-and-joint framework. The following proposition is a fundamental property of the independence of the rigidity matroid.

**Theorem 3.8.** Let \((G, p)\) be a \(d\)-dimensional bar-and-joint framework. If an edge subset \(F' \subseteq E\) is independent in the rigidity matroid \(R(G, p)\), then \(|F'| \leq d|V(F')| - \binom{d+1}{2}\) holds for any \(F' \subseteq F\) such that \(\{p(u) : u \in V(F')\}\) affinely spans \(\mathbb{R}^d\).

**Proof.** Let \(F' \subseteq F\) be an edge subset such that \(\{p(u) : u \in V(F')\}\) affinely spans \(\mathbb{R}^d\). Let us consider the bar-and-joint framework induced by \(F'\), that is, a pair \((G[F'], p)\), where \(G[F'] = (V(F'), F')\) and we denote the restriction of \(p\) to \(V(F')\) simply by \(p\). The independence of \(F\) in \(R(G, p)\) implies that the row vectors of \(R(G, p)\) corresponding to \(F'\) are linearly independent. Hence, the row vectors of \(R(G[F'], p)\) are linearly independent. Since the space of the trivial infinitesimal motions of \(R(G[F'], p)\) has the dimension \(\binom{d+1}{2}\) by Proposition 3.2, we obtain

\[|F'| = \text{rank}(G[F'], p) = d|V(F')| - \dim V(G[F'], p) \leq d|V(F')| - \binom{d+1}{2}.\]
3.1.3 Generic rigidity

Generic joint configurations. Recall that $K(V)$ denotes the complete graph on a finite set $V$. A joint configuration $p$ is called generic if the rank of the rigidity matrix $R(K(V), p)$ as well as every minor of $R(K(V), p)$ have the maximum values taken over all joint configurations $p \in \mathbb{R}^{|V|}$. Note that each minor of the rigidity matrix is written as a polynomial of coordinates of $p$. If such a polynomial is not identically zero, then the set of its solutions forms an algebraic curve in the configuration space $\mathbb{R}^{|V|}$. Let $\mathcal{X}$ be the union of such algebraic curves for all minors of $R(K(V), p)$. Then, a generic configuration is equivalently defined as $p \in \mathbb{R}^{|V|} \setminus \mathcal{X}$. Since $\mathcal{X}$ is a closed semi-algebraic subset of measure zero in $\mathbb{R}^{|V|}$, we have the following fact.

Lemma 3.9. The set of all generic joint configurations is an open dense subset of $\mathbb{R}^{|V|}$ (where a subset $A \subset \mathbb{R}^d$ is said to be an open dense subset of $\mathbb{R}^d$ if the closure of $A$ is $\mathbb{R}^d$).

Generic rigidity. Lemma 3.9 implies that, if we take a joint configuration $p \in \mathbb{R}^{|V|}$ uniformly at random, then $p$ is generic with the probability 1 (see e.g. [45, 129]). This implies that, in almost all cases, the rigidity of frameworks is completely determined by the underlying graphs $G$. More precisely, we have

$$R(G, p) = R(G, q)$$

for any generic $p, q \in \mathbb{R}^{|V|}$. (3.15)

This gives rise to the concept of the generic rigidity. For a graph $G = (V, E)$, the generic rigidity matroid $R(G)$ on $E$ is defined as $R(G) = R(G, p)$ for a generic $p$. By (3.15), this is equivalently defined in terms of the independent sets in such a way that $F \subseteq E$ is independent in $R(G)$ if and only if $F$ is independent in $R(G, p)$ for all $p \in \mathbb{R}^{|V|} \setminus \mathcal{X}$.

Theorem 3.8 provides a necessary condition of the independence of an edge set in the generic rigidity matroid. We now show a sufficient condition, which is given in terms of inductive construction proposed in the statement.

Lemma 3.10. Suppose $G' = (V', E')$ is a graph such that $E'$ is independent in $R(K(V'))$. Let $v$ be a new vertex and $vu_1, vu_2, \ldots, vu_m$ be new edges connecting $v$ and distinct vertices $u_1, u_2, \ldots, u_m \in V'$ as shown in Figure 3.3. If $0 \leq m \leq d$, then $E' \cup \{vu_i : i = 1, \ldots, m\}$ is independent in $R(K(V' + v))$.

Proof. Define $G = (V, E)$ by $V = V' + v$ and $E = E' \cup \{vu_i : i = 1, \ldots, m\}$. Let $p' : V' \to \mathbb{R}^d$ be a generic joint configuration. Since $p'$ is generic and $m \leq d$, we may assume that
Let \( R_p \) and \( \text{Proposition 3.11.} \)

### 3.1. Bar-and-joint Frameworks

Let us discuss the relation of rigidity and infinitesimal rigidity.

By the assumption of the statement the row vectors of \( R \) consisting of two vertices \( V \) configuration such that \( R \) is independent in the rigidity matroid \( \mathcal{R}(K(V), \mathbf{p}) \), and also independent in the generic rigidity matroid \( \mathcal{R}(K(V)) \) by the definition of generic joint configurations.

Lemma 3.10 tells us that, if a bar-and-joint framework \((G, \mathbf{p})\) can be constructed from a smaller framework (e.g., from a segment framework) by applying a sequence of operations of the statement, then this sequence certifies the independence of the original edge set \( E \).

Let us consider the 3-dimensional case for an example. Let \( G_0 = (V_0, E_0) \) be a graph consisting of two vertices \( V_0 = \{a, b\} \) with \( E_0 = \{ab\} \), and let \( \mathbf{p}_0 : V \to \mathbb{R}^3 \) be a joint configuration such that \( \mathbf{p}_0(a) \neq \mathbf{p}_0(b) \). It is clear that \( \text{rank}(G_0, \mathbf{p}_0) = 1 \). We shall first apply an operation of Lemma 3.10 by inserting a new vertex \( v \) and two edges \( ca, cb \), Then we obtain a triangle framework \((G_1, \mathbf{p}_1)\) with \( \text{rank}(G_1, \mathbf{p}_1) = 3 \). Next, we shall apply an operation of Lemma 3.10 that inserts a new vertex \( d \) and three new edges \( da, db, dc \), which results in a tetrahedral framework \((G_2, \mathbf{p}_2)\) with \( \text{rank}(G_2, \mathbf{p}_2) = 6 \) and hence \((G_2, \mathbf{p}_2)\) is isostatic. We then continuously apply operations of Lemma 3.10; for an isostatic framework \((G_{i-1}, \mathbf{p}_{i-1})\), we insert a new vertex \( v_i \) and three new edges \( v_{ii}, v_{i1}, v_{i2}, v_{i3} \) for some distinct \( v_{ii}, v_{i2}, v_{i3} \in V_{i-1} \) as shown in Figure 3.4. By Lemma 3.10, it follows that \( E_i \) is independent and the resulting framework \((G_i, \mathbf{p}_i)\) satisfies

\[
\text{rank}(G_i, \mathbf{p}_i) = \text{rank}(G_{i-1}, \mathbf{p}_{i-1}) + 3 = 3|V_{i-1}| - 6 + 3 = 3|V_i| - 6.
\]

Namely, \((G, \mathbf{p}_i)\) is again isostatic. Generalizing this arguments to \(d\)-dimensional frameworks, we obtain the following fact.

**Proposition 3.11.** Let \( V \) be a finite set. In any dimension there exists an isostatic framework consisting of \(|V|\) joints. Hence, the rank of \( \mathcal{R}(K(V)) \) is equal to \( d|V| - \binom{d+1}{2} \) if \(|V| \geq d + 1 \), and equal to \( \binom{|V|}{2} \) if \(|V| \leq d + 1 \).

We remark that not all isostatic frameworks can be constructed in this way.

**Rigidity v.s. infinitesimal rigidity.** Let us discuss the relation of rigidity and infinitesimal rigidity.
Figure 3.4: A sequence of operations that generate isostatic frameworks.

**Proposition 3.12** (e.g., [26]). Let \((G, p)\) be a \(d\)-dimensional bar-and-joint framework. If \((G, p)\) is infinitesimally rigid, then \((G, p)\) is rigid.

This theorem is equivalently stated that if \((G, p)\) is flexible then \((G, p)\) is infinitesimally flexible. When \((G, p)\) is flexible, there must be a continuous path \(p_t \in \mathbb{R}^d|V|\) with \(p_0 = p\) such that \((G, p_t)\) is equivalent but non-congruent to \((G, p)\) for any \(0 < t < 1\). Hence, the proof of Proposition 3.12 is completed by proving that the derivative of \(p_t\) at \(t = 0\) is nontrivial. See, e.g., [26] for the detailed proof.

However, the converse direction of Theorem 3.12 is not true; a rigid framework may not be infinitesimally rigid as shown in Figure 1.8. Although the proof is not straightforward, it is important to note that the genericity of joint configurations fulfills the gap between infinitesimal rigidity and rigidity:

**Theorem 3.13** (Asimow and Roth [8]). Let \(G = (V, E)\) be a graph and \(p\) be a generic joint configuration. Then, \((G, p)\) is rigid if and only if \((G, p)\) is infinitesimally rigid.

### 3.1.4 Combinatorial rigidity

**Laman’s theorem.** In the previous subsection we have seen that (i) on generic joint configurations the rigidity and the infinitesimal rigidity coincide (Theorem 3.13) and (ii) almost all joint configurations are generic (Lemma 3.9). This implies that, if we understand the independence/dependence of the generic rigidity matroid, then we can reply to the question on the rigidity/flexibility (not necessarily infinitesimal rigidity) of bar-and-joint frameworks. The most important thing is, once again, that the independence/dependence of the generic rigidity matroid is a property of graphs that is purely combinatorial.

The following theorem is a restatement of Laman’s theorem given in the introduction (Theorem 1.1), which completely responds to the request for 2-dimensional case.

**Theorem 3.14** (Laman’s theorem [79]). The followings are equivalent for \(G = (V, E)\).

1. \((G, p)\) is isostatic (minimally rigid) in \(\mathbb{R}^2\) for any generic joint configuration \(p\).
2. \(\text{rank } R(G, p) = 2|V| - 3\) with \(|E| = 2|V| - 3\) for any generic joint configuration \(p\).
3. \(E\) is a base of the generic rigidity matroid \(R(K(V))\).
4. \(G\) satisfies the following counting condition: \(|E| = 2|V| - 3\) and \(|F| \leq 2|V(F)| - 3\) for any nonempty \(F \subseteq E\).

(i) \(\Leftrightarrow\) (ii) follows from Proposition 3.4. (ii) \(\Leftrightarrow\) (iii) follows from Proposition 3.11. (iii) \(\Rightarrow\) (iv)
follows from Theorem 3.8. The nontrivial part is (iv)⇒(ii). See, e.g., [45, 79, 85, 127, 129] for the proof.

The condition of (iv) is now referred to as Laman’s (counting) condition. Recall that, in Example 2.7, we have defined the set function \( \varphi_{2,3} : 2^E \to \mathbb{Z} \) as \( \varphi_{2,3}(F) = 2|V(F)|-3 \) (\( F \subseteq E \)).

**Corollary 3.15.** Let \( G = (V, E) \) be a graph. The generic rigidity matroid \( R(G) \) is equal to the matroid \( \mathcal{M}_{\varphi_{2,3}} \) induced by \( \varphi_{2,3} \) on \( E \).

**Proof.** Let us show that two independent sets of \( R(G) \) and \( \mathcal{M}_{\varphi_{2,3}} \) coincide. If \( F \subseteq E \) is independent in \( R(G) \), then it is independent in \( R(K(V)) \). Hence there exists a base \( B \) of \( R(K(V)) \) such that \( F \subseteq B \), and Laman’s theorem says that \( |F'| \leq 2|V(F')|-3 = \varphi_{2,3}(F') \) for any \( F' \subseteq B \). This implies that \( |F'| \leq \varphi_{2,3}(F') \) for any \( F' \subseteq F \), and \( F \) is independent in \( \mathcal{M}_{\varphi_{2,3}} \). The converse direction can be proved in the same manner, and hence is omitted. \( \square \)

**Minimal k-dof mechanisms.** We say that \( (G, p) \) is a minimal \( k \)-dof mechanism if \( (G, p) \) has \( k \) degree of freedom and removing any bar increases the degree of freedom. We now show a counterpart of Laman’s theorem for minimal \( k \)-dof mechanisms. Recall the truncated matroid introduced in Section 2.2.1. For a graph \( G = (V, E) \) and an integer \( k \), we shall define a \( k \)-truncated (2-dimensional) generic rigidity matroid as

\[
\mathcal{R}^k(G) = (E, \{ I \in I : r_{R(G)}(I) \leq 2|V| - 3 - k \})
\]  

(3.17)

where \( r_{R(G)} \) denotes the rank function of the generic rigidity matroid \( R(G) \). Then we have

\[
r_{R(G)}(F) = \min\{ r_{R(G)}(F), 2|V| - 3 - k \} \quad (F \subseteq E).
\]  

(3.18)

Notice that the rank of \( \mathcal{R}^k(K(V)) \) is equal to \( 2|V| - 3 - k \) since the rank of \( \mathcal{R}^k(K(V)) \) is equal to \( 2|V| - 3 \) by Proposition 3.11.

**Theorem 3.16.** Let \( G = (V, E) \) be a graph and \( k \) be a positive integer. Then the followings are equivalent.

(i) \( (G, p) \) is a minimal \( k \)-dof mechanism in \( \mathbb{R}^2 \) on any generic \( p \).

(ii) \( E \) is a base of the \( k \)-truncated generic rigidity matroid \( \mathcal{R}^k(K(V)) \).

(iii) \( G \) satisfies the following counting conditions \( |E| = 2|V| - 3 - k \) and \( |F| \leq 2|V(F)| - 3 \) for any nonempty \( F \subseteq E \).

**Proof.** ((iii)⇒(ii);) If \( G \) satisfies the counting condition, \( E \) is independent in \( R(K(V)) \) with \( |E| = 2|V| - 3 - k \) by Corollary 3.15, and is also independent in \( \mathcal{R}^k(K(V)) \) by (3.17). Since any independent set of \( \mathcal{R}^k(K(V)) \) has the cardinality at most \( 2|V| - 3 - k \), \( E \) is a base of \( \mathcal{R}^k(K(V)) \).

((ii)⇒(i);) If \( E \) is a base of \( \mathcal{R}^k(K(V)) \), then \( E \) is independent in \( R(K(V)) \) with \( |E| = 2|V| - 3 - k \). Therefore, for a generic joint configuration, the row vectors of \( R(G, p) \) are independent by Corollary 3.15. Hence rank \( R(G, p) = 2|V| - 3 - k \), and \( (G, p) \) is a minimal \( k \)-dof mechanism by Proposition 3.7.
Figure 3.5: (a)0-extension. (b)1-extension.

Figure 3.6: A Henneberg construction.

((i)⇒(iii):) If \((G, p)\) is a minimal \(k\)-dof mechanism, then the row vectors of \(R(G, p)\) are independent with \(|E| = 2|V| - 3 - k\) by Proposition 3.7. This implies (iii) by Corollary 3.15.

\(\square\)

Minimally rigid graphs. We now concentrate on 2-dimensional case. A graph \(G = (V, E)\) is called generically minimally 2-rigid graph if \((G, p)\) is isostatic on a generic \(p\). We simply refer to it as a minimally rigid graph. Equivalently, by Laman’s theorem, \(G\) is said to be minimally rigid if \(G\) satisfies Laman’s counting condition. A minimally rigid graph has several equivalent characterizations. Let us take a look at them.

The first equivalent characterization is written in terms of the so-called Henneberg construction [49]. Let us consider the following two operations for a graph \(G = (V, E)\):

0-extension: Add a new vertex \(v\) and connect it to two existing vertices \(a\) and \(b\) via two new edges (Figure 3.5(a)).

1-extension: Remove an existing edge \(ab\), add a new vertex \(v\), and connect \(v\) to the two endpoints \(a\) and \(b\) of the removed edge and to some other vertex \(c\) (Figure 3.5(b)).

We now consider generating graphs by sequentially applying these two operations.

**Theorem 3.17** (see e.g., [116, 129]). A graph \(G = (V, E)\) is minimally rigid if and only if it can be constructed from \(K_2\) by a sequence of 0- and 1-extensions (where \(K_2\) is the complete graph on two vertices).

We illustrate in Figure 3.6 an example of the Henneberg construction.

The other well known combinatorial characterizations of minimally rigid graphs have been done in terms of tree-partitions. The matroid theory explained in Chapter 2 provides an easy proof.
Theorem 3.18 (Recski [100]). A graph \( G = (V, E) \) is minimally rigid if and only if duplicating any edge \( e \in E \) results in a graph that can be partitioned into two edge-disjoint spanning trees.

Proof. Let us denote by \( G_e = (V, E_e) \) the graph obtained from \( G \) by duplicating an edge \( e \in E \) in parallel. An example is illustrated in Figure 3.7.

If \( G \) is minimally rigid, then by Laman’s counting condition we have \( |E_e| = 2|V| - 2 \) and \( |F| \leq 2|V(F)| - 2 \) for any \( F \subseteq E_e \). By the Tutte-Nash-Williams tree-packing theorem (Theorem 2.11), \( E_e \) can be partitioned into two edge-disjoint spanning trees for any \( e \in E \) as required.

Conversely, suppose that \( G \) is not minimally rigid. If \( |E| \neq 2|V| - 3 \), then \( E_e \) cannot be clearly partitioned into two edge-disjoint spanning trees for any \( e \in E \). Hence there must be a nonempty edge subset \( F \subseteq E \) such that \( |F| > 2|V(F)| - 3 \). Take \( e \in F \) and let \( e' \) be the edge of \( E_e \) parallel to \( e \). Then, \( F + e' \) satisfies not only \( F + e' \subseteq E_e \) but also \( |F + e'| > 2|V(F + e')| - 2 \). By the Tutte-Nash-Williams tree-packing theorem (Theorem 2.11) again, we found that \( E_e \) cannot be partitioned into two edge-disjoint spanning trees.

Theorem 3.19 (Crapo [30]). \( G = (V, E) \) is minimally rigid if and only if \( E \) can be partitioned into three trees \( \{T_1, T_2, T_3\} \) such that (i) each vertex is spanned by exactly two of them and (ii) any subtrees \( T'_i \subseteq T_i \) and \( T'_j \subseteq T_j \) with \( i \neq j \) does not span the same vertex subset, i.e., \( V(T_i) \neq V(T_j) \).

A partition satisfying the first condition is called “3tree2” and the second condition is called “proper”. Hence, Crapo’s partition is referred to as a proper 3tree2 partition. An example is illustrated in Figure 3.8.

Proof. Suppose that \( G = (V, E) \) is minimally rigid. By Theorem 3.18, \( E_e \) can be partitioned into two spanning trees \( S_1 \) and \( S_2 \) on \( V \) for an edge \( e \in E \). Let \( e' \) be the edge of \( E_e \) parallel to \( e \). Without loss of generality we assume \( e' \in S_1 \). Then, \( S_1 - e' \) is a spanning forest that consists of two connected components. Let \( T_1 \) and \( T_2 \) are the edge sets of these two connected components. Also, let \( T_3 = S_2 \). Then, it is easy to see that \( \{T_1, T_2, T_3\} \) is a 3tree2 partition of \( E \). If this is not proper, then there are two subtrees, say \( T'_1 \subseteq T_1 \) and \( T'_2 \subseteq T_2 \), which span the same vertex subset \( V(T_1 \cup T_2) \). We then have

\[
|T'_1 \cup T'_2| = |T'_1| + |T'_2| = |V(T'_1 \cup T'_2)| - 1 + |V(T'_1 \cup T'_2)| - 1 = 2|V(T'_1 \cup T'_2)| - 2,
\]
contradicting that \( G \) is minimally rigid. Thus \( \{T_1, T_2, T_3\} \) is a proper 3tree2 partition.

Suppose \( E \) admits a proper 3tree2 partition \( \{T_1, T_2, T_3\} \). For any \( F \subseteq E \), let \( F_i = F \cap T_i \) for \( i = 1, 2, 3 \). Then, since \( F_i \) is a forest, we have

\[
|F_i| = |V(F_i)| - \kappa(F_i) \quad (3.19)
\]

for \( i = 1, 2, 3 \) (where \( \kappa(F_i) \) denotes the number of connected components in \( G[F_i] \)). Also, since each vertex of \( V(F) \) is spanned by at most two forests among \( \{F_1, F_2, F_3\} \), we have

\[
\sum_{i=1,2,3}|V(F_i)| \leq 2|V(F)|. \quad (3.20)
\]

If \( F_i \neq \emptyset \) for every \( i = 1, 2, 3 \), then (3.19) and (3.20) imply

\[
|F| = \sum_{i=1,2,3}|F_i| = \sum_{i=1,2,3}(|V(F_i)| - \kappa(F_i))
= \sum_{i=1,2,3}|V(F_i)| - \sum_{i=1,2,3}\kappa(F_i)
\leq 2|V(F)| - 3.
\]

On the other hand, if (say) \( F_3 = \emptyset \), then the properness of \( \{T_1, T_2, T_3\} \) implies \( |F| = |F_1| + |F_2| \leq 2|V(F)| - 3 \); otherwise \( F_1 \) and \( F_2 \) become spanning trees that span \( V(F) \) simultaneously. \( \square \)

### 3.2 Body-and-hinge Frameworks

A \( d \)-dimensional body-and-hinge framework is a collection of \( d \)-dimensional bodies connected by hinges, where a hinge is a \((d-2)\)-dimensional affine subspace, i.e. pin-joints in 2-space, line-hinges in 3-space, plane-hinges in 4-space, etc. The bodies are allowed to move continuously in \( \mathbb{R}^d \) so that the motion of any two bodies connected by a hinge is a rotation around it (see Figure 3.9). The framework is called rigid if every motion provides a framework congruent to the original one as in the case of bar-and-joint frameworks.

In this section, we shall provide a formal definition of body-and-hinge frameworks following the description given in [62, 125]. Refer to [31, 62, 124, 125] for more detailed descriptions. Throughout this section, we denote \( \binom{d+1}{2} \) simply by \( D \).
3.2. Body-and-hinge Frameworks

3.2.1 Infinitesimal motions of a rigid body

A body is a set of points that affinely spans $\mathbb{R}^d$. An infinitesimal motion of a body $B$ is an infinitesimal congruence $v : B \rightarrow \mathbb{R}^d$ of the body, i.e.,

$$(p - q) \cdot (v(p) - p(q)) = 0 \quad \text{for any } p, q \in B.$$ 

By Proposition 3.2, it is easy to see that the set of infinitesimal motions of a body forms a $D$-dimensional vector space, i.e., an infinitesimal motion is a linear combination of $d$ translations and $(d - 2)$ rotations around $(d - 2)$-affine subspaces. These infinitesimal motions are elegantly coordinatized by using Plücker coordinates introduced below.

**Extensors.** For any point $p_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,d}) \in \mathbb{R}^d$, we now assign the homogeneous coordinate $(p_{i,1}) = (p_{i,1}, p_{i,2}, \ldots, p_{i,d}, 1)$, denoted by $p_i$ throughout this section. Let $U$ be a $(k-1)$-affine subspace of $\mathbb{R}^d$ determined by $k$ points $p_1, \ldots, p_k \in \mathbb{R}^d$. We denote by $M(p_1, \ldots, p_k)$ the $k \times (d+1)$-matrix whose $i$-th row is $p_i$, i.e.,

$$M(p_1, \ldots, p_k) = \begin{pmatrix}
    p_{1,1} & p_{1,2} & \cdots & p_{1,d} & 1 \\
    p_{2,1} & p_{2,2} & \cdots & p_{2,d} & 1 \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    p_{k,1} & p_{k,2} & \cdots & p_{k,d} & 1
  \end{pmatrix}. \quad (3.21)$$

For $1 \leq i_1 < i_2 < \cdots < i_{d-k+1} \leq d + 1$, the Plücker coordinate $P_{i_1, i_2, \ldots, i_{d-k+1}}$ of $U$ is defined as the $(-1)^{1 + i_1 + i_2 + \cdots + i_{d-k+1}}$ times the determinant of the $k \times k$-submatrix obtained from $M(p_1, \ldots, p_k)$ by deleting $i_j$-th columns for all $j$ with $1 \leq j \leq d - k + 1$. The Plücker coordinate vector of $U$ is defined as the $(d+1\choose k)$-dimensional vector obtained by writing down all of possible Plücker coordinates of $U$ in some predetermined order, say, the lexicographic order of the indices.

Grassmann-Cayley algebra (see, e.g., [124]) treats a Plücker coordinate vector at a symbolic level, that is, no coordinate basis is specified, and the symbolic version of a Plücker coordinate vector is referred to as a $k$-extensor, which is denoted by $p_1 \vee p_2 \vee \cdots \vee p_k$. Although we will work on the coordinatized version, we would like to exploit this terminology to follow the conventional notation.

Let $P = p_1 \vee \cdots \vee p_k$ and $Q = q_1 \vee \cdots \vee q_l$. The join of $P$ and $Q$ is defined as $P \vee Q = p_1 \vee \cdots \vee p_k \vee q_1 \vee \cdots \vee q_l$, that is, a $(d+1\choose k+1)$-dimensional vector consisting of $(k + l) \times (k + l)$-minors of $M(p_1, \ldots, p_k, q_1, \ldots, q_l)$ if $k + l \leq d + 1$ and otherwise 0. The following lemmas are fundamental properties of extensors.
Lemma 3.20. Let \( p_1, p_2, \ldots, p_k \) be \( k \) points in \( \mathbb{R}^d \). Then, \( p_1 \lor \cdots \lor p_k \neq 0 \) if and only if \( \{p_1, \ldots, p_k\} \) is linearly independent and equivalently \( \{p_1, \ldots, p_k\} \) is affinely independent.

Proof. \( p_1 \lor \cdots \lor p_k \neq 0 \) if and only if there is a nonzero \( k \times k \)-minor in \( M(p_1, \ldots, p_k) \). This is equivalent that rank \( M(p_1, \ldots, p_k) = k \), and that \( \{p_1, \ldots, p_k\} \) is linearly independent. \( \square \)

Lemma 3.21. Let \( p_1, p_2, \ldots, p_{d+1} \) be \( d+1 \) points in \( \mathbb{R}^d \) that are affinely independent. Then, the set of \( (d-1) \)-extensors \( \{p_{j_1} \lor p_{j_2} \lor \cdots \lor p_{j_{d-1}} : 1 \leq j_1 < j_2 < \cdots < j_{d-1} \leq d+1\} \) is linearly independent.

Proof. Suppose that it is dependent. Then, there exist scalars \( \lambda_{j_1,j_2,\ldots,j_{d-1}} \) for all \( D = \binom{d+1}{d-1} \) indices, indicating the dependence;

\[
\sum_{1 \leq j_1 < \cdots < j_{d-1} \leq d+1} \lambda_{j_1,j_2,\ldots,j_{d-1}} p_{j_1} \lor p_{j_2} \lor \cdots \lor p_{j_{d-1}} = 0, \tag{3.22}
\]

and at least one scalar must be nonzero. Without loss of generality, we assume \( \lambda_{1,2,\ldots,d-1} \neq 0 \). Then, taking the join of (3.22) with \( p_d \lor p_{d+1} \), we obtain \( \lambda_{1,2,\ldots,d-1} p_1 \lor \cdots \lor p_{d+1} = 0 \) by Lemma 3.20. Since \( P \) is affinely independent, we also have \( p_1 \lor \cdots \lor p_{d+1} \neq 0 \) by Lemma 3.20. This in turn implies \( \lambda_{1,2,\ldots,d-1} = 0 \), which is a contradiction. \( \square \)

The following lemma is a dual version of Lemma 3.21.

Lemma 3.22. Let \( p_1, p_2, \ldots, p_{d+1} \) be \( d+1 \) points in \( \mathbb{R}^d \) that are affinely independent. Then, the set of \( 2 \)-extensors \( \{p_{j_1} \lor p_{j_2} : 1 \leq j_1 < j_2 \leq d + 1\} \) is linearly independent.

Infinitesimal rotations. Let us review how to describe an infinitesimal rotation of a body around a \( (d-2) \)-affine subspace \( A \) in \( \mathbb{R}^d \). Let \( p_1, \ldots, p_{d-1} \) be \( d-1 \) points in \( \mathbb{R}^d \) that affinely span \( A \). Then,

\[
C(A) = p_1 \lor \cdots \lor p_{d-1} \tag{3.23}
\]

is a \( (d-1) \)-extensor associated with \( A \). Note that the 1-dimensional vector subspace of \( \mathbb{R}^D \) spanned by \( C(A) \) is determined independently of the choice of \( p_1, \ldots, p_{d-1} \).

Let \( q \in \mathbb{R}^d \) be a point in the body, and let us consider \( C(A) \lor q \), which is a \( d \)-extensor associated with the \( (d-1) \)-affine subspace \( H_q \) spanned by \( p_1, \ldots, p_{d-1}, q \). Then, \( C(A) \lor q \in \mathbb{R}^{d+1} \), and it is not difficult to see that the first \( d \) coordinates of \( C(A) \lor q \) represent a vector \( v_q \in \mathbb{R}^d \) orthogonal to \( H_q \) as shown in Figure 3.10.

Figure 3.10: The vector \( v_q \) consisting of the first \( d \) coordinates of \( C(A) \lor q \).
In fact, letting \( x = (x_1, \ldots, x_d) \) be an arbitrary point in \( H_q \), Lemma 3.20 implies

\[
C(A) \lor q \lor x = \begin{bmatrix}
  p_{1,1} & p_{1,2} & \cdots & p_{1,d} & 1 \\
  p_{2,1} & p_{2,2} & \cdots & p_{2,d} & 1 \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  p_{d-1,1} & p_{d-1,2} & \cdots & p_{d-1,d} & 1 \\
  q_1 & q_2 & \cdots & q_d & 1 \\
  x_1 & x_2 & \cdots & x_d & 1
\end{bmatrix} = 0,
\]

and this indeed represents the equation of the hyperplane \( H_q \) in \( \mathbb{R}^d \). The coefficient of \( x_i \) in this equation is equal to the \( i \)-th coordinate of \( C(A) \lor q \).

So let us denote the first \( d \) components of \( C(A) \lor q \) by \( v_q \) (that is orthogonal to \( H_q \)). Then, \( C(A) \lor q \) can be expressed by \( (v_q, -v_q \cdot q) \) because

\[
0 = C(A) \lor q \lor q = \begin{bmatrix}
  p_{1,1} & p_{1,2} & \cdots & p_{1,d} & 1 \\
  p_{2,1} & p_{2,2} & \cdots & p_{2,d} & 1 \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  p_{d-1,1} & p_{d-1,2} & \cdots & p_{d-1,d} & 1 \\
  q_1 & q_2 & \cdots & q_d & 1 \\
  q_1 & q_2 & \cdots & q_d & 1
\end{bmatrix}
\]

holds for some constant \( \lambda \). On the other hand, we also have

\[
\text{volume } K_q = \lambda \cdot \text{dist}(q)
\]

We now claim that \( \|v_q\| \) is proportional to the distance between \( A \) and \( q \). To see this, we denote by \( \text{dist}(q) \) the Euclidean distance between \( A \) and \( q \), and let \( v_q^{\text{unit}} \) be the normalized vector of \( v_q \) and let \( v_q^{\text{unit}} = (v_q^{\text{unit}}, 0) \in \mathbb{R}^{d+1} \). Let us consider the simplex \( K_q \) determined by \( \{p_1, \ldots, p_{d-1}, q, q + v_q^{\text{unit}}\} \). Then (since \( p_1, \ldots, p_{d-1} \) are fixed)

\[
\text{volume } K_q = \lambda \cdot \text{dist}(q)
\]

holds for some constant \( \lambda \). On the other hand, we also have

\[
\text{volume } K_q = C(A) \lor q \lor (q + v_q^{\text{unit}}) = C(A) \lor q \lor v_q^{\text{unit}} = (v_q, 1) \cdot (v_q^{\text{unit}}, 0) = \|v_q\|.
\]

Thus, we obtain \( \|v_q\| = \lambda \cdot \text{dist}(q) \) for some constant \( \lambda \).

In summary, we found that the vector consisting of the first \( d \) entries of \( C(A) \lor q \) is the velocity vector of an infinitesimal rotation around \( A \) at \( q \). We will henceforth refer to the \( D \)-dimensional vector \( C(A) \) as the center of the rotation.

**Infinitesimal translations.** We describe an infinitesimal translation of a rigid body in the direction of a (free) vector \( x \in \mathbb{R}^d \). We will consider it as a rotation around an axis at infinity.
Let \( \mathbf{x}_1, \ldots, \mathbf{x}_{d-1} \) be a basis of the orthogonal complement of the vector space spanned by \( \mathbf{x} \), and let \( \overline{\mathbf{x}}_i \) be the projective point at infinity in the direction \( \mathbf{x}_i \), that is, \( \overline{\mathbf{x}}_i = (\mathbf{x}_i, 0) \) for each \( i \). The definition of \( \vee \) is naturally extended to arbitrary projective points and hence let \( C(x) = \mathbf{x}_1 \vee \cdots \vee \mathbf{x}_{d-1} \).

In this setting, we will obtain the same observation as rotations. Let us consider an arbitrary point \( q \) in the body and the \( d \)-extensor \( C(x) \vee q \). It can be shown that the vector consisting of the first \( d \) coordinates is independent of \( q \) and is proportional to \( \mathbf{x} \). Indeed, \( C(x) \vee q \) consists of \( d \times d \)-minors of

\[
\begin{pmatrix}
\mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \cdots & \mathbf{x}_{1,d} & 0 \\
\mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \cdots & \mathbf{x}_{2,d} & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{x}_{d-1,1} & \mathbf{x}_{d-1,2} & \cdots & \mathbf{x}_{d-1,d} & 0 \\
q_1 & q_2 & \cdots & q_d & 1
\end{pmatrix},
\]

and the \( i \)-th coordinate of \( C(x) \vee q \) is written as

\[
(-1)^{1+i} \begin{vmatrix}
\mathbf{x}_{1,1} & \cdots & \mathbf{x}_{1,i-1} & \mathbf{x}_{1,i+1} & \cdots & \mathbf{x}_{1,d} \\
\mathbf{x}_{2,1} & \cdots & \mathbf{x}_{2,i-1} & \mathbf{x}_{2,i+1} & \cdots & \mathbf{x}_{2,d} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{x}_{d-1,1} & \cdots & \mathbf{x}_{d-1,i-1} & \mathbf{x}_{d-1,i+1} & \cdots & \mathbf{x}_{d-1,d}
\end{vmatrix}.
\]

Denote the first \( d \) components of \( C(x) \vee q \) by \( v \). Then, as in the case of infinitesimal rotation, \( C(x) \vee q = (v, -v \cdot q) \) follows from \( C(x) \vee q \vee q = 0 \). Therefore, for any vector \( \overline{x} \in \mathbb{R}^d \) orthogonal to \( x \), we have

\[
0 = C(x) \vee q \vee \overline{x} = (v, -v \cdot q) \cdot (\overline{x}, 0) = v \cdot \overline{x}.
\]

This implies that, for some constant scalar \( \lambda \), we have \( v = \lambda x \) and \( C(x) \vee q = \lambda(x, -x \cdot q) \).

We may henceforth refer to the \( D \)-dimensional vector \( C(x) \) as the center of the infinitesimal translation to in the direction \( x \).

**Arbitrary infinitesimal motions** Recall that an infinitesimal motion of a body is written as a linear combination of infinitesimal rotations and translations as we have mentioned at the beginning of this section. Let us denote by \( C_1, C_2, \ldots, C_D \) the centers of these infinitesimal translations and rotations. Then, the infinitesimal motion at a point \( p \) in the body is the first \( d \) coordinates of \( \sum_{i=1}^{D} (C_i \vee p) \). Hence, we call the \( D \)-dimensional vector \( S = \sum_{i=1}^{D} C_i \) the **screw center** of the infinitesimal motion. We note that a screw center cannot be represented as a \( (d-1) \)-extensor in general (see, e.g., [31]), but the set of all \( (d-1) \)-extensors spans the \( D \)-dimensional vector space (by Lemma 3.21). Therefore, the assignment of an infinitesimal motion to a body may be regarded as the assignment of a screw center, that is, the assignment of a \( D \)-dimensional vector. We define \( S \vee p \) by \( \sum_{i=1}^{D} (C_i \vee p) \).
Indeed, it is easy to check that the infinitesimal motion around a screw center $S$ actually preserves the distances between two points, say $p$ and $q$ in $\mathbb{R}^d$. Suppose that $u$ and $v$ represent the infinitesimal motions at $p$ and $q$ around $S$. Then, as we have seen in the previous two paragraphs, we have $S \vee p = (u, -u \cdot p)$ and $S \vee q = (v, -v \cdot q)$. Therefore,

$$
(u - v) \cdot (p - q) = u \cdot p - u \cdot q - v \cdot p + v \cdot q = -(u, -u \cdot p) \cdot (q, 1) - (v, -v \cdot q) \cdot (p, 1) = -S \vee p \vee q - S \vee q \vee p = 0,
$$

where the last equality follows from the definition of determinants.

### 3.2.2 Infinitesimal rigidity

**Hinge constraints.** In order to describe the infinitesimal rigidity of frameworks consisting of bodies and hinges, it is necessary to understand how each hinge constrains the relative motion of two bodies.

Suppose that two bodies $B$ and $B'$ are joined to a hinge, which is defined as a $(d - 2)$-affine subspace $A$ of $\mathbb{R}^d$, as shown in Figure 3.9. Recall that $C(A)$ denotes a $(d - 1)$-extensor associated with $A$ (see (3.23)). Also, we denote by $C(A)$ the 1-dimensional vector space (as a subspace of $\mathbb{R}^D$) spanned by $C(A)$.

Let us consider the situation in which screw centers $S$ and $S'$ are assigned to the bodies $B$ and $B'$, respectively. Then, the hinge $A$ constrains a relative motion of $B$ and $B'$ to be a rotation about $A$. Note that a motion at a point $p \in B$ relative to $B'$ is written by $(S - S') \vee p$ while an infinitesimal rotation around $A$ can be written by $C(A) \vee p$. Hence, there is a constant scalar $\lambda$ satisfying $(S - S') \vee p = \lambda(C(A) \vee p)$ for all $p \in B$. This condition can be simplified as follows.

**Proposition 3.23.** Let $B$ and $B'$ be two bodies that affinely span $\mathbb{R}^d$, respectively, and let $A$ be a $(d - 2)$-affine subspace of $\mathbb{R}^d$ such that $A \cap B$ affinely spans $A$. (For example $A \cap B$ should contain a segment when $d = 3$.) Suppose that two screw centers $S$ and $S'$ are assigned to $B$ and $B'$, respectively. Then, there exists a constant $\lambda \in \mathbb{R}$ such that

$$
(S - S') \vee q = \lambda(C(A) \vee q) \quad \text{for all } q \in B
$$

if and only if

$$
S - S' \in C(A). \tag{3.26}
$$

Namely, $S - S'$ is a screw center of an infinitesimal rotation of $B$ about $A$ if and only if (3.26) is satisfied.

**Proof.** (3.26)$\Rightarrow$(3.25) is trivial. So let us consider the case when there is a $\lambda$ satisfying (3.25). We take a set of points $p_1, \ldots, p_{d-1} \in A \cap B$ that affinely span $A$ and two more points $p_d, p_{d+1} \in B \setminus A$ such that $p_1, \ldots, p_{d-1}, p_d, p_{d+1}$ are affinely independent. By (3.25) we have

$$
(S - S') \vee p_{j_1} \vee p_{j_2} = \lambda(C(A) \vee p_{j_1} \vee p_{j_2}) \quad \text{for each } 1 \leq j_1 < j_2 \leq d + 1. \tag{3.27}
$$
Notice that the right hand side of (3.27) is nonzero if and only if \( j_1 = d \) and \( j_2 = d + 1 \) by (3.23) and Lemma 3.20. Let us now consider the left hand side. Since \( S - S' \) is a screw center, it can be described as a combination of \((d - 1)\)-extensors, say, \( S - S' = \sum_{j} a^i_j \lor a^j_k \lor \cdots \lor a^i_{d-1} \). Recall that the 2-extensor \( p_{j_1} \lor p_{j_2} \) is a \( D \)-dimensional vector consisting of Plücker coordinates arranged in the lexicographic order of the indices of Plücker coordinates (see Section 3.2.1). Let us denote by \( p_{j_1} \lor p_{j_2} \) the \( D \)-dimensional vector arranged in the reverse of the lexicographical order of the indices. Then, by the definition of the extensor, we have

\[
(a^i_1 \lor \cdots \lor a^i_{d-1}) \lor (p_i \lor p_j) = (a^i_1 \lor \cdots \lor a^i_{d-1}) \cdot (\overline{p_i} \lor \overline{p_j})
\]

Therefore, the left term of (3.27) can be written by using the dot product as

\[
(S - S') \lor p_{j_1} \lor p_{j_2} = \sum_{i} ((a^i_1 \lor \cdots \lor a^i_{d-1}) \lor (p_{j_1} \lor p_{j_2})) = \sum_{i} ((a^i_1 \lor \cdots \lor a^i_{d-1}) \cdot (\overline{p_{j_1}} \lor \overline{p_{j_2}})) = (S - S') \cdot (\overline{p_{j_1}} \lor \overline{p_{j_2}}).
\]

In total, (3.27) indicates the system of \( D \) linear equations on unknowns \( S - S' \) described as

\[
\begin{pmatrix}
\overline{p_1} \lor \overline{p_2} \\
\overline{p_1} \lor \overline{p_3} \\
\vdots \\
\overline{p_{d-1}} \lor \overline{p_{d+1}} \\
\overline{p_d} \lor \overline{p_{d+1}}
\end{pmatrix}
(S - S')^\top =
\begin{pmatrix}
0 \\
0 \\
\vdots \\
\lambda C(A) \lor p_d \lor p_{d+1}
\end{pmatrix}
\]

(3.28)

Since \( \{ p_{j_1} \lor p_{j_2} : 1 \leq j_1 < j_2 \leq d + 1 \} \) is linearly independent by Lemma 3.22, (3.28) implies that \( S - S' \) is determined uniquely. It is clear that \( S - S' = \lambda C(A) \) is a solution of this system, and we thus obtain \( S - S' \in \overline{C(A)} \).

**Body-and-hinge frameworks and infinitesimal motions.** A \( d \)-dimensional body-and-hinge framework is a pair \((G, p)\) of a multigraph \( G = (V, E) \) and a mapping \( p \) that associates a \((d - 2)\)-affine subspace \( p(e) \) of \( \mathbb{R}^d \) with each \( e \in E \). Namely, each vertex corresponds to a body and each edge corresponds to a hinge connecting two bodies (see Figure 3.11). The framework \((G, p)\) is called a body-and-hinge realization of \( G \) in \( \mathbb{R}^d \), and \( p \) is called a hinge configuration.

Recall a discussion given in Section 3.2.1: an infinitesimal motion of a body can be regarded as a \( D \)-dimensional vector, called a screw center, and Proposition 3.23 further tells us that a hinge constraint is written as a single equation (3.26). From this fact, we define an infinitesimal motion of \((G, p)\) as a mapping \( S : V \to \mathbb{R}^D \) such that

\[
S(u) - S(v) \in \overline{C(p(e))}
\]

(3.29)

for every \( e = uv \in E \). Namely, \( S \) is an assignment of a screw center \( S(u) \) to the body of \( u \in V \) (rather than actual “motion”). An infinitesimal motion \( S \) is called trivial if \( S(u) = S(v) \) for
all \( u, v \in V \), and \((G, p)\) is said to be \textit{infinitesimally rigid} if all infinitesimal motions of \((G, p)\) are trivial.

As before, we identify an infinitesimal motion \( S \) as a point in \( \mathbb{R}^{D|V|} \), which is a composition of \(|V|\) vectors \( S(v) \in \mathbb{R}^D \) for \( v \in V \). It is easy to see that the dimension of trivial motions is equal to \( D \) (see the proof of Proposition 3.24), and \( S : V \rightarrow \mathbb{R}^D \) is trivial if and only if it is a screw center of some infinitesimal congruence of \( \mathbb{R}^d \).

**Rigidity matrix.** We now define the rigidity matrix so that the null space is the set of infinitesimal motions of \((G, p)\). Since \( S \) is an infinitesimal motion of \((G, p)\) if and only if it satisfies (3.29), taking any basis \( \{r_1(p(e)), r_2(p(e)), \ldots, r_{D-1}(p(e))\} \) of the orthogonal complement of \( C(p(e)) \), we can say that \( S \) is an infinitesimal motion of \((G, p)\) if and only if

\[
(S(u) - S(v)) \cdot r_i(p(e)) = 0
\]

for all \( i \) with \( 1 \leq i \leq D - 1 \) and for all \( e = uv \in E \). Hence, the constraints to be an infinitesimal motion are described by a system of \((D - 1)|E|\) linear equations with unknown \( S \in \mathbb{R}^{D|V|} \). Consequently, we obtain a \((D - 1)|E| \times D|V|\)-matrix \( R(G, p) \) associated with this homogeneous system of linear equations, \( R(G, p)S^T = 0 \), where sequences of consecutive \((D - 1)\) rows of \( R(G, p) \) are indexed by elements of \( E \) and sequences of consecutive \( D \) columns are indexed by elements of \( V \). \( R(G, p) \) is described as

\[
\begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & u & \cdot & \cdot & v & \cdot & \cdot \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot0\cdots & r(p(e)) & \cdot0\cdots & -r(p(e)) & \cdot0\cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\end{pmatrix}
\]

where \( r(p(e)) \) denotes the \((D - 1) \times D\)-submatrix whose \( i \)-th row vector is \( r_i(p(e)) \), i.e.,

\[
r(p(e)) = \begin{pmatrix} r_1(p(e)) \\ \vdots \\ r_{D-1}(p(e)) \end{pmatrix}.
\]

We call \( R(G, p) \) the **rigidity matrix** of \((G, p)\).

The null space of \( R(G, p) \), which is the space of all infinitesimal motions, is denoted by \( \mathcal{V}(G, p) \). We remark that the dimension of \( \mathcal{V}(G, p) \), and equivalently the dimension of the
and the 5 × 5 matrix $A$ can be taken as follows:

$$A = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$ 

Hence a basis $\{S_1^*, \ldots, S_D^*\}$ is linearly independent implies that the rank of $R(G, p)$ is at most $D|V| - D = D(|V| - 1)$. Notice also that $\{S_1^*, \ldots, S_D^*\}$ spans the space of all trivial infinitesimal motions and thus $(G, p)$ is infinitesimally rigid if and only if the rank of $R(G, p)$ is exactly $D(|V| - 1)$.

More generally, the dimension of the space of nontrivial infinitesimal motions is called the degree of freedom of $(G, p)$, which is equal to $D(|V| - 1) - \text{rank } R(G, p)$.

**Example 3.2.** Let us show how $C(A), r_1(A)$ and $r(A)$ can be described for a $(d-2)$-affine subspace $A$ in low dimensional cases. In $d = 2$ (and $D = 3$), $A$ is a point in $\mathbb{R}^2$ and hence let us denote $A = p = (px, py) \in \mathbb{R}^2$. Then, according to the definition of $M(p)$ given in (3.21), we have

$$M(p) = (px, py, 1).$$

Following the definition (3.23), a screw center of $A$ can be taken as

$$C(A) = (1, -py, px).$$

Hence a basis $\{r_1(A), r_2(A)\}$ of the orthogonal complement of $\overline{C(A)}$ can be taken as follows:

$$r_1(A) = (py, 1, 0)$$

$$r_2(A) = (-px, 0, 1).$$

The $2 \times 3$-matrix $r(A)$ becomes

$$r(A) = \begin{pmatrix}
py & 1 & 0 \\
-px & 0 & 1
\end{pmatrix}. \tag{3.30}$$

Let $d = 3$ and $D = 6$. Let us consider a $(d-2)$-affine space $A$ (i.e., a line) passing through $p^1 = (p_x^1, p_y^1, p_z^1)$ and $p^2 = (p_x^2, p_y^2, p_z^2)$. Then, a screw center can be expressed by

$$C(A) = \left(\begin{vmatrix}
p_x^1 & 1 \\
p_y^1 & 0 \\
p_z^1 & 0
\end{vmatrix}, \begin{vmatrix}
p_x^2 & p_z^2 - p_y^2 \\
p_y^2 & p_z^2 - p_x^2 \\
p_z^2 & p_y^2 - p_x^2
\end{vmatrix}\right)$$

and the $5 \times 6$-matrix $r(A)$ becomes as follows:

$$r(A) = \begin{pmatrix}
r_1(A) & p_y^1 & p_z^1 & 1 & 0 & 0 \\
r_2(A) & p_x^2 & 0 & 0 & -p_y^1 & 1 \\
r_3(A) & 0 & -p_z^1 & 0 & -p_y^1 & 0 \\
r_4(A) & p_y^2 - p_y^1 & p_z^2 - p_z^1 & 0 & 0 & 0 \\
r_5(A) & 0 & p_z^2 - p_z^1 & 0 & p_y^2 - p_y^1 & 0
\end{pmatrix}. \tag{3.31}$$

with $p_y^2 - p_y^1 \neq 0$. \(\square\)
3.2.3 Combinatorial rigidity

Generic hinge configuration. Let \( A \) be a collection of \( (d-2) \)-affine subspaces of \( \mathbb{R}^d \). A hinge configuration \( \mathbf{p} : K(V) \to A \) is called generic if the ranks of \( R(K(V), \mathbf{p}) \) and its edge-induced submatrices take the maximum values over all hinge configurations. Similarly, for any \( E \subseteq K(V) \), a hinge configuration \( \mathbf{p}' : E \to A \) is called generic if it is the restriction of some generic hinge configuration on \( K(V) \) to \( E \). It is known that almost all hinge configurations are generic (see, e.g., [62]).

Tay-Whiteley’s theorem. For a multigraph \( G = (V, E) \) and a positive integer \( k \), the graph obtained by replacing each edge by \( k \) parallel edges is denoted by \( kG \). Tay [114] and Whiteley [127] independently proved that the generic infinitesimal rigidity of a body-and-hinge framework is determined by the underlying (multi)graph as follows.

**Theorem 3.25** (Tay [114], Whiteley [127]). Let \( G \) be a multigraph and \( \mathbf{p} \) be a generic hinge configuration. Then, the body-and-hinge framework \( (G, \mathbf{p}) \) is infinitesimally rigid in \( \mathbb{R}^d \) if and only if \( (\binom{d+1}{2} - 1)G \) contains \( \binom{d+1}{2} \) edge-disjoint spanning trees.

Let us consider the graph illustrated in Figure 3.11; (b) shows the underlying graph \( G \) of the body-and-hinge framework illustrated in (a). Figure 3.12(a) shows \( 5G \), and it contains six edge-disjoint spanning trees as illustrated in (b). Hence, the Tay-Whiteley theorem (Theorem 3.25) ensures that \( (G, \mathbf{p}) \) is infinitesimally rigid in \( \mathbb{R}^3 \) on every generic hinge configuration \( \mathbf{p} \).

Combining the Tay-Whiteley theorem with the Tutte-Nash-Williams tree packing theorem (Theorem 2.11), a Laman-type characterization of generic body-and-hinge frameworks is obtained as follows: \( (G, \mathbf{p}) \) is infinitesimally rigid on a generic hinge configuration if and only if there exists an edge subset \( F \) in \( (\binom{d+1}{2} - 1)G \) such that
- \( |F| = \binom{d+1}{2}(|V| - 1) \)
- \( |F'| \leq \binom{d+1}{2}(|V(F')| - 1) \) for any nonempty \( F' \subseteq F \).
Chapter 4

A Rooted-forest Partition with Uniform Vertex Demand

In Chapter 3 we have presented several combinatorial characterizations of the bar-and-joint rigidity and body-and-hinge rigidity. In particular, we have seen that characterizing the rigidity in terms of tree-partitions commonly appears in, e.g., Theorem 3.18, Theorem 3.19, and Theorem 3.25, wherein the Tutte-Nash-Williams tree-packing theorem (Theorem 2.11) is a fundamental tool for proofs.

In this chapter we propose an extension of the Tutte-Nash-Williams tree-packing theorem to rooted-forest partitions (which will be formally defined soon). Although this chapter concerns only with the topic of graph theory, our newly proposed rooted-forest-partition will be directly used for characterizing the rigidity of bar-and-joint frameworks having line-sliders in the next chapter.

4.1 Introduction

Theorem 2.11 with \( l = k \) asserts that \( G = (V, E) \) can be partitioned into \( k \) edge-disjoint spanning trees if and only if \( |E| = k|V| - k \) and \( |F| \leq k|V(F)| - k \) for any nonempty \( F \subseteq E \). This is the so-called Tutte-Nash-Williams tree-packing theorem [89, 118].

As a variant of the commonly studied trees or forests, we have also introduced a pseudo-forest in Section 2.7: a graph is a pseudoforest if each connected component contains at most one cycle. Whiteley [127] has proved a generalization of the tree-packing theorem by mixing spanning trees and spanning pseudoforests (as we have already seen in Theorem 2.11): for two integers \( k \) and \( l \) with \( k \geq l \), \( G \) can be partitioned into edge-disjoint \( l \) spanning trees and \( k - l \) spanning pseudoforests if and only if \( |E| = k|V| - l \) and \( |F| \leq k|V(F)| - l \) for any nonempty \( F \subseteq E \). Haas [48] has broadened the range of \( l \): for two integers \( k \) and \( l \) with \( k \leq l \leq 2k - 1 \), a graph \( G = (V, E) \) satisfies \( |E| = k|V| - l \) and \( |F| \leq k|V(F)| - l \) for any nonempty \( F \subseteq E \) if and only if it can be partitioned into \( l \) edge-disjoint trees such that each vertex is spanned by exactly \( k \) of them and any distinct \( l \) subtrees (with at least one edge) among them do not span a same vertex subset. Also, Frank and Szegő [40] and Fekete and
Szegö [37] have provided constructive characterizations of these sparse graphs.

In this chapter we focus on a **rooted-forest**, that is, a forest such that each connected component has a unique root. By regarding each self-loop in a graph as a root, a graph is said to be a **rooted-forest** if and only if each connected component contains exactly one loop but there exists no cycle consisting of non-loop edges. In this setting, a collection of subgraphs of rooted-forests forms a graph class that lies between forests and pseudoforests.

**Contribution.** We newly prove a necessary and sufficient condition of a graph (having self-loops) to be decomposed into edge-disjoint rooted-forests with two additional (nontrivial) conditions, a **precolored condition of roots** and a **vertex demand condition**. To impose these conditions, we shall consider a **loop-colored graph** $G = (V, E)$, i.e., each loop has some prespecified color (as shown in Figure 4.1(a)), and let $\{c_1, c_2, \ldots, c_k\}$ be the set of colors appearing in the loop set of $G$. Roughly speaking, the color set of the roots represents the set of distinct types of supplies, and among $k$ types of roots we require that every vertex receives $d$ distinct types of supplies through $d$ rooted-forests (where $d \leq k$).

To define our partition problem more formally, let us introduce the following terminology. For $F \subseteq E$, let $L(F)$ denote the set of loops contained in $F$. A subgraph $G' = (V', E')$ of $G$ is said to be a **rooted-forest colored in** $c_i$ if it satisfies the following three conditions:

(F1) $(V', E' \setminus L(E'))$ is a forest,

(F2) $E'$ does not contain any loop colored in $c_j$ with $j \neq i$, and

(F3) each connected component of $G'$ contains exactly one loop colored in $c_i$.

$G'$ is further called a **spanning rooted-forest** (colored in $c_i$) if $E'$ spans $V$ in addition to (F1)–(F3) (see Figure 4.1(b)). We shall refer to each connected component of a rooted-forest as a **rooted-tree**. Also, if $G[F] = (V(F), E)$ forms a rooted-forest for $F \subseteq E$, then $F$ is simply called a rooted-forest.

Given a loop-colored graph $G = (V, E)$ and a positive integer $d$, we generalize a concept of the forest partition to a partition $E = \{E_1, E_2, \ldots, E_k\}$ of $E$ into $k$ components such that

(P1) each $E_i$ is a rooted-forest colored in $c_i$,

(P2) each vertex is spanned by exactly $d$ components, i.e., $|\{i : \delta_E(v) \cap E_i \neq \emptyset\}| = d$ holds for each $v \in V$, where $\delta_E(v)$ denotes the set of edges of $E$ incident to $v$.

If a partition of $E$ satisfies the two conditions (P1) and (P2), we say that $E$ (or $G$) admits a **$(k, d)$-rooted-forest partition**. Figure 4.1(c) shows an example of a $(4, 3)$-rooted-forest partition. Finding such a partition may be considered as a coloring problem of non-loop edges into $k$ colors such that each color induces a rooted-forest and the number of distinct colors appearing around a vertex is equal to $d$.

For an edge set $F \subseteq E$, let $\chi(F)$ be the total number of distinct colors appearing in $L(F)$. The following forest partition theorem is our main result.

**Theorem 4.1.** Let $G$ be a loop-colored graph, $k$ be the number of colors used in $G$, and $d$ be a positive integer with $d \leq k$. Then, $G$ admits a proper $(k, d)$-rooted-forest partition if and only if it satisfies the following counting conditions:
4.2. Proof of Theorem 4.1

4.2.1 Case of $k = d$

We shall first consider a special case of Theorem 4.1: $k$, the number of colors appearing in $G$, is equal to the vertex demand $d$. Let us denote by $c_1, c_2, \ldots, c_d$ the colors that we shall
consider in this subsection. We prove that, if \( k = d \), then \((d,d)\)-rooted-forest partitions can be characterized in terms of the union of \( d \) matroids.

Recall that \( K(V) \) denotes the complete graph on a finite set \( V \). Let us denote by \( K^+(V) \) the loop-colored graph obtained from \( K(V) \) by attaching a loop colored in \( c_i \) to each vertex for every \( 1 \leq i \leq d \). (Namely \( d|V| \) loops are inserted in total.) We simply denote by \( K^+(V) \) the edge set of \( K^+(V) \) if it is clear from the context.

For each color \( c_i \), let us first consider the following function \( \tau_i : 2^{K^+(V)} \to \mathbb{Z} \): for \( F \subseteq K^+(V) \),

\[
\tau_i(F) = \varrho_{1,1}(F) + \chi_i(F) = |V(F)| - 1 + \chi_i(F),
\]

where \( \chi_i(F) \) is defined by

\[
\chi_i(F) = \begin{cases} 
1 & \text{ if } L(F) \text{ contains a loop colored in } c_i \\
0 & \text{ otherwise.}
\end{cases}
\]

Since \( \varrho_{1,1} \) and \( \chi_i \) are submodular, \( \tau_i \) is submodular by Lemma 2.3. Note

\[
\sum_i \chi_i(F) = \chi(F).
\]

Also \( \tau_i \) is nondecreasing, and hence it induces a matroid, denoted by \( \mathcal{M}_{\tau_i} \), on \( K^+(V) \).

**Lemma 4.3.** An edge set \( F \subseteq K^+(V) \) is a base of \( \mathcal{M}_{\tau_i} \) if and only if it is a spanning rooted-forest colored in \( c_i \).

**Proof.** Suppose that \( F \) is a spanning rooted-forest colored in \( c_i \). Then (F2) implies that \( F \) does not contain any loop colored in \( c_j \) with \( j \neq i \). Hence, for any \( F' \subseteq F \), \( \chi_i(F') = 1 \) if and only if \( F' \) contains a loop. From (F1) and (F3), it is not difficult to see that \( F \) is independent in \( \mathcal{M}_{\tau_i} \); for any \( F' \subseteq F \) with \( L(F') = \emptyset \), we have \( |F'| = |V(F')| - 1 = \tau_i(F') \) because \( F' \) is a forest, while for any \( F' \subseteq F \) with \( L(F') \neq \emptyset \) we have \( |F'| = |V(F')| = \tau_i(F') \) because \( F' \) is a forest with each connected component containing at most one. Let us show that \( F \) is a base. Since \( F \) spans \( V \), each connected component of \( (V,F) \) contains exactly one loop by (F3). This implies that the number of connected components of \( (V,F) \) is equal to \( |L(F)| \). We thus obtain \( |F \setminus L(F)| = |V| - |L(F)| \), implying \( |F| = |V| \). Since any independent set has the cardinality at most \( |V| \) by (4.2), \( F \) is a base of \( \mathcal{M}_{\tau_i} \).

Conversely, suppose that \( F \) is a base. Notice that, for any loop \( e \) colored in \( c_j \) with \( j \neq i \), \( \{e\} \) is dependent in \( \mathcal{M}_{\tau_i} \) since \( |\{e\}| > \tau_i(\{e\}) = 0 \). Any independent set in \( \mathcal{M}_{\tau_i} \) thus contains no loop colored in \( c_j \) with \( j \neq i \), implying (F2). Since \( F \) is independent, we have \( |F'| \leq |V(F')| - 1 \) for any \( F' \subseteq F \setminus L(F) \) by (4.2), and hence \( F \setminus L(F) \) is an independent set of the graphic matroid on \( K(V) \). Namely, \( F \setminus L(F) \) forms a forest on \( V \), implying (F1). To show (F3), suppose for a contradiction that \( F \) contains a subset \( F' \) such that \( G[F'] \) is connected with \( |L(F')| \geq 2 \). We then have \( |F'| = |V(F')| - 1 + |L(F')| > |V(F')| = \tau_i(F') \), contradicting the independence of \( F \). Hence, each connected component of the graph \( (V,F) \) contains at most one loop. Since the number of connected components of \( (V,F) \) is equal to \( |V| - |F \setminus L(F)| \), which is equal to \( |L(F)| \) by \( |V| = |F| \), each connected component of \( (V,F) \) contains exactly one loop. This implies that \( F \) satisfies (F3) and also \( F \) spans \( V \). \[\square\]
We now consider characterizing independent sets of $\bigvee_{i=1}^{d} M_{\tau_{i}}$ in terms of the counting condition. To apply Lemma 2.10, we just need to show the following property.

**Lemma 4.4.** Let $c_{i}$ be a color and let $F \subseteq K^{+}(V)$. Let $m$ be the total number of connected components of $G[F]$ and $\{F_{1}, \ldots, F_{m}\}$ be a partition of $F$ such that, for each $j = 1, \ldots, m$, $F_{j}$ is the edge set of a connected component of $G[F]$. Also, let $F_0 = \emptyset$. Then, the partition $\{F_0, F_1, \ldots, F_m\}$ takes the minimum value for the rank formula (2.12) of $r_{\tau_{i}}$. Namely, the rank $r_{\tau_{i}}(F)$ of $F$ in $M_{\tau_{i}}$ can be written as $r_{\tau_{i}}(F) = \sum_{i=1}^{m} \tau_{i}(F_{i})$.

**Proof.** For $1 \leq l \leq m$, let $B_{l}$ be a base of $M_{\tau_{i}} | F_{l}$, where $M_{\tau_{i}} | F_{l}$ denotes the restriction of $M_{\tau_{i}}$ to $F_{l}$ (see Section 2.2.1). We no prove

$$\tau_{i}(F_{l}) = |B_{l}|. \quad (4.4)$$

(i) Let us consider the case where $F_{l}$ does not contain a loop colored in $c_{i}$. Since $G[F_{l}]$ is connected, there exists an edge subset of $F_{l}$ that forms a spanning tree on $V(F_{l})$. This edge set is independent by Lemma 4.3. Moreover, since any independent set of $M_{\tau_{i}}$ contains no loop colored in $c_{j}$ with $j \neq i$ and also $F_{l}$ contains no loop colored in $c_{i}$, a spanning tree is a maximal independent set of $F_{l}$ in $M_{\tau_{i}}$. We thus obtain $\tau_{i}(F_{l}) = |V(F_{l})| - 1 = |B_{l}|$ by $\chi_{c}(F) = 0$.

(ii) Let us consider the case where $F_{l}$ contains a loop colored in $c_{i}$. Since $G[F_{l}]$ is connected, there exists an edge subset of $F_{l}$ consisting of a spanning tree on $V(F_{l})$ with one loop colored in $c_{i}$. By Lemma 4.3, this edge subset is a base $B_{l}$ of $F_{l}$ with $|B_{l}| = |V(F_{l})|$. Thus $\tau_{i}(F_{l}) = |V(F_{l})| = |B_{l}|$ holds by $\chi_{c}(F_{l}) = 1$.

Thus, for both cases, (4.4) is verified. Let $B = \bigcup_{l=1}^{m} B_{l}$. It is obvious $M_{\tau_{i}} | F = M_{\tau_{i}} | F_{1} \oplus \cdots \oplus M_{\tau_{i}} | F_{m}$ since $G[F_a]$ and $G[F_b]$ are vertex-disjoint for any $1 \leq a, b \leq m$ with $a \neq b$. This implies that $B$ is a base of $M_{\tau_{i}} | F$. As a result, we obtain $r_{\tau_{i}}(F) = |B| = \sum_{l=1}^{m} |B_{l}| = \sum_{l=1}^{m} \tau_{i}(F_{l})$, and hence $\{F_0, F_1, \ldots, F_m\}$ with $F_0 = \emptyset$ takes the minimum value of (2.12) for $r_{\tau_{i}}(F)$.

Combining Lemma 2.10 and Lemma 4.4, we obtain $\bigvee_{i=1}^{d} M_{\tau_{i}} = M_{\sum \tau_{i}}$, and the following lemma follows.

**Lemma 4.5.** Let $F \subseteq K^{+}(V)$. Then, $F$ is a base of $\bigvee_{i=1}^{d} M_{\tau_{i}}$ if and only if it satisfies the counting condition (C1)/(C2).

**Proof.** Let us consider $F' \subseteq F$. By Lemma 4.4, the minimum value of (2.12) for $r_{\tau_{i}}(F')$ is taken by the partition $\{F'_0, F'_1, \ldots, F'_m\}$ such that $F'_0 = \emptyset$ and each $F'_l$ is the edge set of a connected component of $G[F']$. (Note that the definition of $\{F'_0, F'_1, \ldots, F'_m\}$ is not related to the coloring of loops.) We can hence apply Lemma 2.10 to obtain $\bigvee_{i=1}^{d} M_{\tau_{i}} = M_{\sum \tau_{i}}$. Namely $\bigvee_{i=1}^{d} M_{\tau_{i}}$ is the matroid induced by the submodular function $\sum_{i=1}^{d} \tau_{i}$. Observe that $\sum_{i=1}^{d} \tau_{i}(F') = d|V(F')| - d + \sum_{i=1}^{d} \chi_{i}(F') = d|V(F')| - d + \chi(F') = d|V(F')| - d + \min\{d, \chi(F'), d\}$ for any $F' \subseteq F$ by (4.3) and $\chi(F') \leq d$, and hence $\sum_{i=1}^{d} \tau_{i}$ is equal to $d$ in (4.1). This implies that $F$ is a base of $\bigvee_{i=1}^{d} M_{\tau_{i}}$ if and only if it satisfies the counting condition (C1)/(C2).
We are now ready to prove Theorem 4.1 for \(k = d\).

**Proof of Theorem 4.1 for \(k = d\).** Lemma 4.3 implies that a base of \(\mathcal{M}_{\tau}\) is exactly a spanning rooted-forest colored in \(c_i\). Hence, an edge set is a base of \(\bigvee_{i=1}^{d} \mathcal{M}_{\tau}\) if and only if it can be partitioned into \(d\) edge subsets \(E_i\) each of which is a spanning rooted-forest colored in \(c_i\), and equivalently it admits a \((d,d)\)-rooted-forest partition. Combining Lemma 4.5 with this fact, we conclude that an edge set admits a \((d,d)\)-rooted-forest partition if and only if it satisfies the counting condition. \(\square\)

The argument based on the matroid union (combination of Lemmas 2.8, 2.10, and 4.4) also enables us to obtain a decomposition theorem for \(k < d\):

**Theorem 4.6.** Let \(G = (V, E)\) be a loop-colored graph and \(k\) be the number of colors appearing in \(G\). Also, let \(d\) be an integer with \(k \leq d\). Then, \(E\) satisfies

- \(|E| = d|V| - d + \chi(E)\), and
- \(|F| \leq d|V(F)| - d + \chi(F)\) for every \(F \subseteq E\),

if and only if \(E\) admits a partition \(\mathcal{E}\) into \(d\) components \(\{E_1, \ldots, E_k, E_{k+1}, \ldots, E_d\}\) such that

- \(E_i\) is a spanning rooted-forest colored in \(c_i\) for each \(1 \leq i \leq k\), and
- \(E_i\) is a spanning tree on \(V\) for each \(k + 1 \leq i \leq d\).

### 4.2.2 Case of \(k > d\)

Let us first show the necessity of Theorem 4.1.

**Proof of the necessity of Theorem 4.1.** Let \(\{E_1, \ldots, E_k\}\) be a \((k,d)\)-rooted-forest partition of \(E\). Note that (P1) implies \(|E_i| = |V(E_i)|\) for each \(i\). Notice also that (P2) implies \(\sum_{i=1}^{k} |V(E_i)| = d|V|\) since each vertex appears in exactly \(d\) sets among \(V(E_i), i = 1, \ldots, k\). Therefore, we have \(|E| = \sum_{i=1}^{k} |E_i| = \sum_{i=1}^{k} |V(E_i)| = d|V|\), implying (C1).

(C2) can be shown in a similar manner. For any \(F \subseteq E\), let \(F_i = F \cap E_i, i = 1, \ldots, k\).

Let \(s = \chi(F)\) and \(t = |\{i : F_i \neq \emptyset\}|\). Note that \(s \leq t\). Without loss of generality, we assume that \(F_i \neq \emptyset\) for \(1 \leq i \leq t\) (and \(F_i = \emptyset\) for \(t + 1 \leq i \leq k\)) and \(L(F_i) \neq \emptyset\) for \(1 \leq i \leq s\) (and \(L(F_i) = \emptyset\) for \(s + 1 \leq i \leq k\)). Then, by (P1), we have the following fact: \(|F_i| \leq |V(F_i)|\) for each \(1 \leq i \leq s\) while \(|F_i| \leq |V(F_i)| - 1\) for each \(s + 1 \leq i \leq t\).

Also, since each vertex of \(V(F)\) is spanned by at most \(d\) sets among \(F_1, \ldots, F_t\) by (P2), we have

\[
\sum_{i=1}^{t} |V(F_i)| \leq d|V(F)|. \quad (4.5)
\]

To see (C2), suppose for a contradiction that \(F \subseteq E\) violates the counting condition (C2), i.e., \(|F| \geq \mu_d(F) + 1 = d|V(F)| - d + \min\{d, s\} + 1\) (recall \(s = \chi(F)\)). Then,

\[
|F| = \sum_{i=1}^{t} |F_i| \leq \sum_{i=1}^{s} |V(F_i)| + \sum_{i=s+1}^{t} |V(F_i)| - 1 \leq t|V(F)| - (t - s),
\]

and hence

\[
d|V(F)| - d + \min\{d, s\} + 1 \leq |F| \leq t|V(F)| - (t - s). \quad (4.6)
\]
On the other hand, by using (4.5), we also have

\[ |F| = \sum_{i=1}^t |F_i| \leq (\sum_{i=1}^t |V(F_i)|) - (t-s) \leq d|V(F)| - (t-s), \]

and hence

\[ d|V(F)| - d + \min\{d,s\} + 1 \leq |F| \leq d|V(F)| - (t-s). \quad (4.7) \]

If \( t < d \), then (4.6) implies

\[ 0 \geq (d-t)|V(F)| - d + \min\{d,s\} + 1 + (t-s) \geq \min\{d,s\} - s + 1 = 1 \]

by \( |V(F)| \geq 1 \) and \( s \leq t < d \). This is a contradiction.

If \( t \geq d \), then \( t + \min\{d,s\} \geq d + s \) holds for any \( t \) and \( s \) (with \( t \geq d \) and \( t \geq s \)). Hence, (4.7) implies \( 1 \leq 0 \), which is a contradiction. \( \square \)

To show the sufficiency, we need some terminologies. Suppose that a loop-colored graph \( G = (V,E) \) satisfies the counting condition (C1)(C2). An edge set \( F \subseteq E \) is called tight if \( |F| = \mu_d(F) \). Note that \( G[F] = (V(F), F) \) satisfies the counting condition for any tight set \( F \) with \( \chi(F) \geq d \). A tight set \( F \) is said to be small if \( |V(F)| = 1 \) (and hence \( F \) consists of loops attached to a vertex), and otherwise large. Also, a tight set is connected if it induces a connected subgraph, and otherwise disconnected. We need the following two easy observations.

**Lemma 4.7.** Let \( G \) be a loop-colored graph satisfying the counting condition (C1)(C2). Suppose that \( G \) has a disconnected tight set \( F \). Then, the edge set of any connected component of \( G[F] \) is also tight.

**Proof.** Since \( G[F] \) is disconnected, \( F \) can be partitioned into \( F_1 \) and \( F_2 \) such that \( V(F_1) \cap V(F_2) = \emptyset \) and \( V(F_1) \cup V(F_2) = V(F) \). We then have

\[ \mu_d(F) = |F| = \sum_{i=1,2} |F_i| \leq \sum_{i=1,2} \mu_d(F_i) = \sum_{i=1,2} (d|V(F_i)| - d + \min\{d,\chi(F_i)\}) = d|V(F)| - d + (\min\{d,\chi(F_1)\} + \min\{d,\chi(F_2)\}) - d \leq d|V(F)| - d + \min\{d,\chi(F)\} = \mu_d(F) \]

where we used \( \min\{d,\chi(F_1)\} + \min\{d,\chi(F_2)\} \leq d + \min\{d,\chi(F)\} \). Therefore, the equality must hold everywhere, and in particular we have \( |F_i| = \mu_d(F_i) \) for \( i = 1,2 \). Namely, both \( F_1 \) and \( F_2 \) are tight. Continuing this argument recursively until \( F_1 \) becomes connected, we see that the edge set of any connected component is tight. \( \square \)

**Lemma 4.8.** Let \( G \) be a loop-colored graph satisfying the counting condition (C1)(C2). Suppose that \( G \) is disconnected. Then, each connected component of \( G \) satisfies the counting condition (C1)(C2).
Proof. Let $G_1 = (V_1, E_1), \ldots, G_m = (V_m, E_m)$ with $m > 1$ be the connected components of $G$. Since $G$ satisfies (C2), $G_i$ also satisfies (C2) for any $i$. We will show that $G_i$ satisfies (C1), i.e., $|E_i| = d|V_i|$. For any $E_i$, we have

$$d|V| = |E| = |E_i| + |E \setminus E_i|$$

$$\leq \mu_d(E_i) + \mu_d(E \setminus E_i)$$

$$\leq d|V(E_i)| - d + \min\{d, \chi(E_i)\} + d|E \setminus E_i|$$

implying $d \leq \min\{d, \chi(E_i)\}$, and hence we obtain $\chi(E_i) \geq d$. Also, since $E$ is a tight set, Lemma 4.7 implies that the edge set of every connected component of $G$ is also tight. Namely, $|E_i| = \mu_d(E_i) = d|V(E_i)| = d|V_i|$ since $\chi(E_i) \geq d$. \hfill $\Box$

Let us start the proof of the sufficiency of our main theorem.

Proof of the sufficiency of Theorem 4.1. By Lemma 4.8, we can assume that $G$ is connected. Also, we can assume $|V| \geq 2$ since Theorem 4.1 trivially holds if $|V| = 1$. (If $V = \{v\}$, then the counting condition says that $E$ consists of $d$ loops attached to $v$ each of which has the different color.)

The proof is done by induction on $|E \setminus L(E)|$. The following two lemmas cope with the case when $G$ has a large tight set.

**Lemma 4.9.** If $G$ contains a large tight set $F$ with $\chi(F) < d$, then $G$ admits a $(k, d)$-rooted-forest partition.

**Proof.** Let us consider a tight set $F$ with $\chi(F) < d$ as illustrated in Figure 4.2(a), i.e.,

$$|F| = \mu_d(F) = d|V(F)| - d + \chi(F). \quad (4.8)$$

Let $s = \chi(F)$. Without loss of generality, we assume that the colors appearing in $L(F)$ are $c_1, c_2, \ldots, c_s$.

By Theorem 4.6, $F$ can be partitioned into $s$ spanning rooted-forests and $d - s$ spanning trees on $V(F)$. Hence, $F$ contains a spanning tree on $V(F)$, implying that $F$ is connected and $F \setminus L(F) \neq \emptyset$.

We shall consider the contraction of $F$. Since $F$ is connected, $V(F)$ is contracted to a vertex, denoted by $v^*$. Let $H'$ be the resulting graph, i.e., $H' = (V \setminus V(F) \cup \{v^*\}, E \setminus F))$\textsuperscript{1} (see Figure 4.2(c)). Furthermore, we shall insert a new loop colored in $c_i$ attached to the new vertex $v^*$ for each $1 \leq i \leq s$. Let us denote the set of these newly inserted loops by $S^*$, and let $H$ be the resulting graph (obtained by inserting $S^*$ to $H'$) as shown in Figure 4.2(d). To avoid ambiguities, we shall denote by $V_G(F')$ and $V_H(F')$ the sets of vertices in $G$ and $H$ spanned by an edge set $F'$, respectively, throughout the proof of this lemma.

We claim the following.

\textsuperscript{1}For the simplicity of notations, we shall identify the edge set $E \setminus F$ of $G$ with the edge set of the new graph $H'$ by implicitly assuming a mapping of the corresponding endpoints. For example, an edge $e = (a, b)$ of $G$ with $a \in V(F)$ and $b \in V \setminus V(F)$ represents an edge connecting $v^*$ and $b$ in $H'$.
4.2. Proof of Theorem 4.1

Figure 4.2: Illustration of graphs used in the proof of Lemma 4.9 for $k = 5$ and $d = 3$. (a) A graph satisfying the counting condition (C1)(C2), where a shaded disk covers a large tight set $F$ with $s = \chi(F) = 2$. (b) A partition $\{F_1, F_2, F_3\}$ of $F$. (c) $H'$. (d) $H$. (e) A partition $\{E_1', E_2', E_3', E_4', E_5'\}$ of $(E \setminus F) \cup S^*$. (f) $\{E_1, \ldots, E_5\}$ is a partition of $E$ satisfying (P1) and (P2), see Figures 4.3~4.6 for more detail.

Claim 4.10. $H$ satisfies the counting condition (C1)(C2).

Proof. It is easy to check that $H$ satisfies (C1). Suppose for a contradiction that $H$ violates (C2) and there is an edge set $C$ in $H$ with

$$|C| \geq \mu_d(C) + 1 = d|V_H(C)| - d + \min\{d, \chi(C)\} + 1. \quad (4.9)$$

Then, trivially, $v^* \in V_H(C)$ holds (since otherwise the original graph $G$ has an edge set, $C$, violating the counting condition). Also, we may assume $S^* \subset C$ because the insertion of $S^* \setminus C$ into $C$ increases the left term of (4.9) by $|S^* \setminus C|$ while it increases the right term by at most $|S^* \setminus C|$.

Notice that $V_H(C) = (V_G(C \setminus S^*) \setminus V_G(F)) \cup \{v^*\}$ by $v^* \in V_H(C)$. Hence, we have

$$|V_H(C)| = |V_G(C \setminus S^*) \setminus V_G(F)| + 1. \quad (4.10)$$

Also, we need the following two equalities (4.11) and (4.12): By the definition of $S^*$,

$$|S^*| = \chi(S^*) = s = \chi(F). \quad (4.11)$$

By $\chi(S^*) = \chi(F)$ and $S^* \subset C$,

$$\chi(C) = \chi((C \setminus S^*) \cup F). \quad (4.12)$$
Therefore, we obtain
\[
\begin{align*}
|(C \setminus S^*) \cup F| &= |C| - |C \cap S^*| + |F| \quad \text{(by } C \cap F = \emptyset) \\
&= |C| + |F| - |S^*| \quad \text{(by } S^* \subset C) \\
&\geq d(|V_H(C)| + |V_G(F)|) - 2d + \min\{d, \chi(C)\} + \chi(F) - |S^*| + 1 \quad \text{(by (4.8) (4.9))} \\
&= d(|V_H(C)| + |V_G(F)|) - 2d + \min\{d, \chi(C)\} + 1 \quad \text{(by (4.11))} \\
&= d(|V_G((C \setminus S^*) \cup F)| - d + \min\{d, \chi((C \setminus S^*) \cup F)\} + 1 \quad \text{(by (4.10) (4.12))} \\
&= \mu_d((C \setminus S^*) \cup F) + 1.
\end{align*}
\]
This implies that \( G \) has an edge subset, \((C \setminus S^*) \cup F\), violating the counting condition \((C2)\), which is a contradiction.

Therefore, \( H \) satisfies the counting condition \((C1)(C2)\), and hence \((E \setminus F) \cup S^*\), the edge set of \( H \), admits a \((k, d)\)-rooted-forest partition \( \mathcal{E}' = \{E'_1, E'_2, \ldots, E'_k\} \) by induction (see Figure 4.2(e)).

The condition \((P2)\) of a \((k, d)\)-rooted-forest partition implies that each vertex is spanned by exactly \( d \) rooted-forests. Note that \( v^* \) is spanned by \( E'_i \) for every \( 1 \leq i \leq s \) because, in \( H \), \( v^* \) is incident to a loop colored in \( c_i \) (recall the definition of \( S^* \)). Hence, there are exactly \( d - s \) other rooted-forests that span \( v^* \), and without loss of generality we may assume that \( E'_{s+1}, \ldots, E'_d \) span \( v^* \).

By Theorem 4.6, \( F \) can be partitioned into \( \{F_1, F_2, \ldots, F_d\} \) such that \( F_i \) is a spanning rooted-forest colored in \( c_i \) for \( 1 \leq i \leq s \) and is a spanning tree on \( V(F) \) for \( s + 1 \leq i \leq d \) (see Figure 4.2(b)). Let
\[
E_i = \begin{cases} 
(E'_i \setminus S^*) \cup F_i & \text{for } 1 \leq i \leq s \\
E'_i \cup F_i & \text{for } s + 1 \leq i \leq d \\
E'_i & \text{for } d + 1 \leq i \leq k.
\end{cases}
\]
See Figure 4.2(f) (and Figures 4.3~4.6 below) for an example. Then, observe that \( \mathcal{E} = \{E_1, \ldots, E_k\} \) is a partition of \( E \) since \( \{F_1, \ldots, F_d\} \) is a partition of \( F \) and \( \{E'_1 \setminus S^*, \ldots, E'_s \setminus S^*, E'_{s+1}, \ldots, E'_k\} \) is a partition of \( E \setminus F \). We now prove that \( \mathcal{E} \) is a \((k, d)\)-rooted-forest partition.

**Claim 4.11.** \((V, E_i)\) is a rooted-forest colored in \( c_i \) for each \( 1 \leq i \leq k \).

**Proof.** Let us first show that \( E_i = (E'_i \setminus S^*) \cup F_i \) forms a rooted-forest colored in \( c_i \) for \( 1 \leq i \leq s \). The following two figures (Figures 4.3 and 4.4) show how \( E_i \) is constructed in the example of Figure 4.2. We should remark that, in the middle graphs of these figures, \( E'_i \setminus S^* \) is depicted on \( V \) (not on \((V \setminus V(F)) \cup \{v^*\}\)), and observe that contracting \( V(F) \) to a vertex \( v^* \) and then inserting a loop colored in \( c_i \) to \( v^* \) we obtain a rooted-forest \( E'_i \) on \( V \setminus V(F) \cup \{v^*\} \) (as depicted in Figure 4.2(e)).

Let us start to show that \( E_i \) is a rooted-forest. Recall that \( F_i \) is a rooted-forest spanning \( V(F) \). Recall also that \( E'_i \) is a rooted-forest in \( H \). Hence, considering \( E'_i \setminus S^* \) on \( V \) (by expanding \( v^* \) to \( V(F) \)), it forms a forest such that each connected component contains at
4.2. Proof of Theorem 4.1

most one loop colored in $c_i$ (as shown in the middle graphs of Figures 4.3 and 4.4). More precisely, a connected component of $E'_i \setminus S^*$ on $V$ misses a loop if and only if it spans some vertex of $V(F)$. In $E_i$, such a connected component (missing a loop) is connected to a rooted-subtree of $F_i$ since $F_i$ spans $V(F)$, and their union forms a rooted-tree on $V$. Therefore, every connected component of $E_i$ forms a rooted-tree.

Let us next show that $E_i = E'_i \cup F_i$ is a rooted-forest colored in $c_i$ for $s + 1 \leq i \leq d$. Figure 4.5 shows how $E_i$ is constructed in this case. The proof proceeds as in the previous case. Recall that $E'_i$ is a rooted-forest spanning $v^*$ in $H$. Let $T$ be the rooted-subtree of $E'_i$ which spans $v^*$ on $V \setminus V(F) \cup \{v^*\}$. Then, since $F_i$ is defined to be a spanning tree on $V(F)$, $T \cup F_i$ forms a rooted-tree on $V$. Also, any rooted-tree of $E'_i$ other than $T$ clearly remains a rooted-tree on $V$. Therefore, every connected component of $E_i$ forms a rooted-tree.

The remaining is to show that $E_i = E'_i$ is a rooted-forest colored in $c_i$ for $d + 1 \leq i \leq k$. Figure 4.6 are examples of this case, and it can be easily seen that $E_i$ is a rooted-forest colored in $c_i$ on $V$ because $E'_i$ is not related to $v^*$ in $H$ (and $V(F)$ in $G$).

By the above claim, we found that $\mathcal{E} = \{E_1, \ldots, E_k\}$ satisfies (P1). (P2) can be easily checked: each vertex of $V \setminus V(F)$ is spanned by $d$ rooted-forests as done by components of $\mathcal{E}'$ and each vertex of $V(F)$ is spanned by $E_1, \ldots, E_k$ as done by $F_1, \ldots, F_k$. As a result, $\mathcal{E}$ is a $(k, d)$-rooted-forest partition of $E$.

Lemma 4.12. If $G$ contains a large and connected tight set $F$ with $\chi(F) \geq d$ and $V(F) \subseteq V$, then $G$ admits a $(k, d)$-rooted-forest partition.

Proof. Let us consider a large and connected tight set $F$ with $\chi(F) \geq d$ and $V(F) \subseteq V$ as illustrated in Figure 4.7(a). Let $s = \chi(F)$. Without loss of generality, we assume that
the colors appearing in $L(F)$ are $c_1, c_2, \ldots, c_s$. By induction, $F$ admits a $(s, d)$-rooted-forest partition $\{F_1, F_2, \ldots, F_s\}$.

We shall consider the operation that removes all the edges of $F$ and then inserts a new loop colored in $c_i$ to each vertex of $V(F_i)$ for each $i = 1, \ldots, s$. Namely, for $i = 1, \ldots, s$, we prepare a set of new loops $S_i^*$ composed of one loop colored in $c_i$ for each vertex of $V(F_i)$, (where “new” means $S_i^* \cap E = \emptyset$), and we consider $(E \setminus F) \cup S^*$, where $S^* = \bigcup_{i=1}^s S_i^*$ (see Figure 4.7(c) and (d)). Since each vertex of $V(F)$ is spanned by exactly $d$ rooted-forests among $F_1, \ldots, F_s$, we inserted $d$ new loops to each vertex of $V(F)$ after all.

We claim the following.

**Claim 4.13.** $(E \setminus F) \cup S^*$ satisfies the counting condition (C1)(C2).

If this is true, then $(E \setminus F) \cup S^*$ admits a $(k, d)$-rooted-forest partition $E' = \{E'_1, E'_2, \ldots, E'_k\}$ by induction (Figure 4.7(e)). Let

$$E_i = \begin{cases} (E'_i \setminus S_i^*) \cup F_i & \text{for } 1 \leq i \leq s \\ E'_i & \text{for } s + 1 \leq i \leq k \end{cases}$$

It can be shown as in the proof of Claim 4.11 that $E = \{E_1, E_2, \ldots, E_k\}$ satisfies (P1) and (P2). (See Figure 4.7(f) for an example.) As a result, $E$ is a $(k, d)$-rooted-forest partition of $E$.

Hence, the proof of Lemma 4.12 is completed by proving Claim 4.13. Checking (C1) is trivial. Let us verify (C2). Suppose, for a contradiction, that $(E \setminus F) \cup S^*$ contains an edge subset $C$ satisfying $|C| \geq \mu_d(C) + 1$. Note that

$$C \cap F = \emptyset \quad (4.13)$$

because $C \subseteq E \setminus F \cup S^*$ and $S^*$ is a set of “new” loops (i.e., $S^* \cap F = \emptyset$). Also, we have the
4.2. Proof of Theorem 4.1

followings:

\[ C \cap S^* \neq \varnothing \]  \hspace{1cm} (4.14)

\[ V(S^*_i) \subseteq V(F) \text{ and } V(S^*) = V(F), \]  \hspace{1cm} (4.15)

where (4.14) follows from the fact that \( C \) violates the counting condition \( (C2) \) but \( C \setminus S^* \subseteq E \) does not, and (4.15) directly follows from the definitions of \( S^*_i \) and \( S^* \). (4.14) and (4.15) implies

\[ V(C) \cap V(F) = V(C) \cap V(S^*) \neq \varnothing. \]  \hspace{1cm} (4.16)

Also, since \( S^*_i \) consists of loops, one loop for each vertex of \( V(F_i) \), we have

\[ |C \cap S^*_i| \leq |V(C) \cap V(S^*_i)|. \]  \hspace{1cm} (4.17)

Therefore, we obtain

\[ |C \cap S^*| = \sum_{i=1}^{s} |C \cap S^*_i| \leq \sum_{i=1}^{s} |V(C) \cap V(S^*_i)| = \sum_{i=1}^{s} |(V(C) \cap V(F)) \cap V(S^*_i)|, \]  \hspace{1cm} (4.18)

where the rightmost equation follows from \( V(S^*_i) \subseteq V(F) \). In order to evaluate the rightmost term of (4.18), we use the property \( (P2) \): in the \((s, d)\)-rooted-forest partition \( \{F_1, \ldots, F_s\} \) of \( F \), each vertex \( v \in V(F_i) \) is spanned by exactly \( d \) rooted-forests. Since \( S^*_i \) spans the same set of vertices as \( F_i \), each vertex of \( v \in V(C) \cap V(F) \) is spanned by exactly \( d \) edge sets among \( S^*_1, \ldots, S^*_s \). This implies

\[ \sum_{i=1}^{s} |(V(C) \cap V(F)) \cap V(S^*_i)| = d |V(C) \cap V(F)|. \]  \hspace{1cm} (4.19)
On the other hand, focusing on the number $\chi(C)$ of colors appearing in $C$, we clearly have $|\{i : V(C) \cap V(S_i^*) \neq \emptyset\}| \leq \chi(C)$. This implies
\[
\sum_{i=1}^{n} |V(C) \cap V(S_i^*)| = \sum_{i: V(C) \cap V(S_i) \neq \emptyset} |V(C) \cap V(S_i^*)| \leq \chi(C) \cdot |V(C) \cap V(S^*)|
\]
where the last equation follows from (4.16). In total, combining this relation with (4.18) and (4.19), we obtain
\[
|C \cap S^*| \leq \min\{d, \chi(C)\} \cdot |V(C) \cap V(F)|.
\]
Since $|C| \geq \mu_d(C) + 1$ and $|F| = \mu_d(F) = d|V(F)|$, we also have
\[
|C| + |F| \geq d|V(C)| - d + \min\{d, \chi(C)\} + 1 + d|V(F)|
\]
\[
= d|V(C) \cup V(F)| + d|V(C) \cap V(F)| - d + \min\{d, \chi(C)\} + 1
\]
\[
= d|V(C \cup F)| + d|V(C) \cap V(F)| - d + \min\{d, \chi(C)\} + 1.
\]

However, since $(C \setminus S^*) \cup F$ is an edge set contained in $G$, this contradicts that $(C \setminus S^*) \cup F$ satisfies the counting condition (C2). This completes the proof of Claim 4.13 as well as that of Lemma 4.12.

By the previous two lemmas, we now concentrate on $G$ which contains no large and connected tight set $F$ with $V(F) \subseteq V$. Let us assume that $G$ is a counterexample of the statement of Theorem 4.1, and prove a property of this counterexample in the next lemma. We then prove that the existence of a counterexample $G$ leads to a contradiction.

**Lemma 4.14.** Suppose that $G$ admits no $(k, d)$-rooted-forest partition (i.e., $G$ is a counterexample). Let $l$ be a loop colored in $c_l$ attached to a vertex $v$ in $G$. Then, any vertex $u$ adjacent to $v$ possesses (at least) one of the following properties in $G$:

(i) $u$ is incident to a loop colored in $c_l$, or

(ii) $u$ is incident to $d$ loops.

**Proof.** Let $e$ be an edge connecting $u$ and $v$ in $G$. Consider the following operation that generates a new graph: remove $e$ and then insert a new loop $l'$ colored in $c_l$ to $u$. If the resulting graph, denoted by $G'$, satisfies the counting condition, then $G'$ admits a $(k, d)$-rooted-forest partition $\mathcal{E}' = \{E'_1, \ldots, E'_{k-1}, E'_i \setminus \{l'\} \cup$
\{e\}, E_{i+1}', \ldots, E_k\}. Then, it is not difficult to see that \(\mathcal{E}\) is a \((k, d)\)-rooted-forest partition of \(E\), contradicting that \(G\) is a counterexample.

Therefore, \(G'\) cannot satisfy the counting condition (C2) (obviously \(G'\) satisfies (C1)), and there exists an edge subset \(C\) in \(G'\) satisfying \(|C| \geq \mu_d(C) + 1\). Let us take \(C\) as an inclusionwise minimal edge subset satisfying this inequality. Clearly \(l' \in C\) (since otherwise an edge subset of \(G\) violates the counting condition). We claim

\[
v \notin V(C).
\]

To see this, suppose \(v \in V(C)\). Let \(C' = C \setminus \{l'\} \cup \{e, l\}\). We then have \(V(C') = V(C)\) by \(v \in V(C)\), and we also have \(|C'| \geq |C|\) by \(l' \in C\) and \(e \notin C\). Also, \(\chi(C') = \chi(C)\) holds since \(l\) has the same color as \(l'\). Thus, \(|C'| \geq |C| \geq d|V(C)| - d + \min\{d, \chi(C)\} + 1 = d|V(C')| - d + \min\{d, \chi(C')\} + 1 = \mu_d(C') + 1\), contradicting that \(G\) satisfies the counting condition since \(C' \subseteq E\). Hence (4.25) holds.

(4.25) implies \(V(C) \not\subseteq V\). Consider \(C \setminus \{l'\}\). Then, by \(C \setminus \{l'\} \subseteq E\), \(|C \setminus \{l'\}| \leq \mu_d(C \setminus \{l'\})\) holds. Combining this inequality, \(|C| \geq \mu_d(C) + 1\), and \(\mu_d(C \setminus \{l'\}) \leq \mu_d(C)\), we obtain \(|C \setminus \{l'\}| = \mu_d(C \setminus \{l'\})\). Therefore, \(C \setminus \{l'\}\) is a tight set with \(V(C \setminus \{l'\}) \not\subseteq V\). Recall that \(G\) cannot have any large connected tight set \(F\) with \(V(F) \not\subseteq V\) by Lemmas 4.9 and 4.12. Moreover, Lemma 4.7 implies that every connected component of a disconnected tight set is again a tight set. Hence, the minimality of \(C\) implies that \(C \setminus \{l'\}\) is connected. As a result, \(C \setminus \{l'\}\) must be a small tight set that spans \(u\), i.e., \(C \setminus \{l'\}\) consists of a set of loops attached to \(u\). Recall that \(C\) violates the counting condition. Namely, adding a loop \(l'\) colored in \(c_i\) into \(C \setminus \{l'\}\) violates the counting condition, implying that either \(C \setminus \{l'\}\) contains a loop colored in \(c_i\) or \(C \setminus \{l'\}\) consists of \(d\) loops according to the definition of \(\mu_d\) (see (4.1)).

By using the last lemma, we now show that the existence of a counterexample derives a contradiction. Suppose, for a contradiction, that \(G\) admits no \((k, d)\)-rooted-forest partition. If \(G\) has a vertex incident to \(d\) loops, then properties (i) and (ii) of Lemma 4.14 imply that any vertex adjacent to \(v\) is also incident to \(d\) loops. Since \(G\) is connected (due to Lemma 4.8), we see that every vertex of \(G\) is incident to \(d\) loops. In total we have \(d|V|\) loops in \(G\). However, since \(G\) is connected with \(|V| \geq 2\), \(E \setminus L(E) \neq \emptyset\) holds. Therefore, we obtain \(|E| > d|V|\), contradicting that \(G\) satisfies the counting condition (C1).

Hence \(G\) has no vertex incident to \(d\) loops. The property (i) of Lemma 4.14 then implies that, if there is a vertex incident to a loop colored in \(c_i\), then any adjacent vertex also has a loop colored in \(c_i\). Since \(G\) is connected, continuing this process, we eventually see that every vertex is incident to a loop colored in \(c_i\). As a result, every vertex is incident to \(k\) loops (\(k\) denotes the number of colors appearing in the set of loops of \(G\)). By \(k \geq d\), this contradicts that \(G\) has no vertex incident to \(d\) loops.

This completes the proof of Theorem 4.1.

Note that the above proof is constructive, i.e., it provides an explicit way to construct a \((k, d)\)-rooted-forest partition. Let us restate the last step of the above proof formally (since it will be used in the next section).
Let $G = (V, E)$ be a loop colored graph satisfying the counting condition. Suppose that $G$ has no large and connected tight set $F$ with $V(F) \subseteq V$. Then, $G$ has two adjacent vertices $u$ and $v$ and a loop $l$ attached to $v$ such that removing an edge connecting $u$ and $v$ and then inserting a new loop to $u$ with the same color as $l$ results in a graph satisfying the counting condition (C1)/(C2).

4.3 Algorithms

In this section we shall briefly discuss how to check whether a loop-colored graph satisfies the counting condition (C1)/(C2) or not. Imai [55] has shown that, for a given graph $H = (V, E)$ and two integers $k$ and $l$, it can be decided whether $H$ satisfies $|F| \leq k|V(F)| - l$ for every nonempty $F \subseteq E$ in polynomial time. Our algorithm is an extension of Imai’s technique. Based on this algorithm, we shall show how to compute a large and connected tight set in Section 4.3.2, and provide an algorithm for constructing a $(k, d)$-rooted-forest partition in Section 4.3.3.

We use the following conventional notations. In a digraph $D = (U, A)$ with a node set $U$ and an arc set $A$, $a = (u, v) \in A$ indicates an arc from $u$ to $v$ and called a leaving arc from $u$ (and an entering arc to $v$). Similarly, for any $S \subseteq U$, $a = (u, v)$ is a leaving arc from $S$ if $u \in S$ and $v \notin S$. A network $N = (D, c)$ is a pair of a digraph $D$ and a capacity function $c$ on $A$. In the subsequent discussions we will focus on a network having two designate nodes $s$ and $t$, called a source and a sink, respectively. An $s-t$ flow $f$ is a function on $A$ with $0 \leq f(a) \leq c(a)$ for every $a \in A$ satisfying the flow conservation law at each node (except for $s$ and $t$). The value of $f$ is the sum of $f(a)$ over all leaving arcs $a$ from $s$. For any $S \subseteq U$ with $s \in S$ and $t \notin S$, the set $\delta_N(S)$ of arcs leaving from $S$ is called an $s-t$ cut, and its weight is defined as the sum of $c(a)$ over all $a \in \delta_N(S)$.

4.3.1 Checking the counting condition

Let us first claim our result.

Theorem 4.16. Let $G = (V, E)$ be a loop-colored graph, $k$ be the number of colors appearing in $G$, $d$ be an integer with $d \leq k$. Then, one can check whether $G$ satisfies the counting condition (C1)/(C2) or not in $O(d^5|V|^2)$ time.

Proof. Since (C1) can be trivially checked from the input, we assume that $|E| = d|V|$ holds. Now let us consider how to check (C2).

To explain the basic idea, let us first consider how to decide whether $G = (V, E)$ satisfies $|F| \leq d|V(F)|$ for every $F \subseteq E$. We shall define an auxiliary network $N_1 = (D_1, c_1)$ as follows. The node set of $D_1$ is defined as $E \cup V \cup \{s, t\}$, where $s$ and $t$ are a sink and a source, respectively, and the arc set $A_1$ of $D_1$ is defined as $A_1 = \{(e, v) \in E \times V : e$ is incident to $v$ in $G\} \cup \{(s, e) : e \in E\} \cup \{(v, t) : v \in V\}$. Namely, each non-loop edge has two leaving arcs while a loop has one leaving arc in $N_1$ (see Figures 4.8(a) and (b) for an example). The capacity function $c_1$ on $A_1$ is defined as $c_1(a) = \infty$ if $a \in E \times V$, $c_1(a) = 1$ if $a \in \{(s, e) : e \in E\}$, and $c_1(a) = 1$ if $a \in \{(v, t) : v \in V\}$.
if \( a \in \{s\} \times E \), and \( c_1(a) = d \) if \( a \in V \times \{t\} \). Then, the network \( N_1 \) has the following property on \( s-t \) cuts. For any \( F \subseteq E \) and any node subset \( S \subseteq E \cup V \cup \{s\} \) of \( N_1 \) satisfying \( s \in S \) and \( S \cap E = F \), the weight of the \( s-t \) cut \( \delta_{N_1}(S) \) is at least \(|E \setminus F| + d|V(F)|\), and moreover this value can be achieved when \( S \cap V = V(F) \). Therefore, the max-flow min-cut theorem (see e.g., [104]) implies that \(|F| \leq d|V(F)|\) holds for any \( F \subseteq E \) if and only if the value of a maximum \( s-t \) flow in \( N_1 \) is equal to \(|E|\). Thus, one can check whether \( G \) satisfies \(|F| \leq d|V(F)|\) for every \( F \subseteq E \) by a maximum flow algorithm.

Checking \(|F| \leq d|V(F)| - d\) can be performed by extending the above idea. For an edge \( e \in E \), define a capacity function \( c_{1,e} \) on \( A_1 \) as \( c_{1,e}(a) = d + 1 \) if \( a = (s,e) \) and otherwise \( c_{1,e}(a) = c_1(a) \). Let \( N_{1,e} = (D_1, c_{1,e}) \). Namely, we increase the capacity of the arc \((s,e)\) from 1 to \( d + 1 \). Suppose that there exists a maximum \( s-t \) flow \( f \) in \( N_1 \) whose value is equal to \(|E|\). Then, as in the case of \( N_1 \), it can be observed from the max-flow min-cut theorem that \(|F| \leq d|V(F)| - d\) holds for any \( e \in F \subseteq E \) if and only if the value of a maximum flow in \( N_{1,e} \) is equal to \(|E| + d\). Therefore, one can check whether \( G = (V,E) \) satisfies \(|F| \leq d|V(F)| - d\) for any \( F \subseteq E \) by computing the values of maximum \( s-t \) flows in \( N_{1,e} \) for all \( e \in E \) [55, Theorem 2.1]. Observe that one can compute a maximum \( s-t \) flow of \( N_{1,e} \) from that of \( N_1 \) by at most \( d \) applications of a flow augmentation. Since finding an augmenting path can be performed in \( O(|A_1|) \) time, the total computational time is bounded by \( T_{\text{max-flow}}(N_1) + O(d|E||A_1|) \), where \( T_{\text{max-flow}}(N_1) \) is the time for computing a maximum \( s-t \) flow in \( N_1 \). Thus, we have a way to check the condition \(|F| \leq d|V(F)| - d\) by the technique of [55].

In order to take into account the term, \( \min\{d, \chi(F)\} \), appearing in our counting condition \((C2)\), we shall insert a gadget into \( N_1 \). Let us define a network \( N_2 = (D_2, c_2) \) as follows. The node set of \( D_2 \) is \( L(E) \cup C \cup \{s,t,n\} \), where \( C \) denotes the set \( \{c_1, c_2, \ldots, c_k\} \) of colors appearing in \( G \), \( s \) and \( t \) are a source and a sink, respectively, and \( n \) is a special node. The arc set \( A_2 \) of \( D_2 \) is defined as \( A_2 = \{(e,c) \in L(E) \times C : e \text{ is colored in } C\} \cup \{(s,e) | e \in L(E)\} \cup \{(c,n) | c \in C\} \cup \{(n,t)\} \) (see Figure 4.8(c)). Also, the capacity function \( c_2 \) on \( A_2 \) is defined as \( c_2(a) = \infty \) if \( a \in L(E) \times C \), \( c_2(a) = 1 \) if \( a \in \{s\} \times L(E) \), \( c_2(a) = 1 \) if \( a \in C \times \{n\} \), and \( c_2(a) = d \) if \( a = (n,t) \). Then, the network \( N_2 \) has the following property: for any nonempty \( F \subseteq L(E) \) and any node subset \( S \subseteq L(E) \cup C \cup \{s,n\} \) of \( N_2 \) satisfying \( s \in S \) and \( S \cap L(E) = F \), the weight of the \( s-t \) cut \( \delta_{N_2}(S) \) is at least \(|L(E) \setminus F| + \min\{d, \chi(F)\}\) (and this value can be achieved).

We now put \( N_1 \) and \( N_2 \) together by overlapping the node set \( L(E) \cup \{s,t\} \), and denote the resulting network by \( N = (D,c) \). Namely, \( D \) is a digraph on \( E \cup V \cup C \cup \{s,t,n\} \) whose arc set is \( A = A_1 \cup A_2 \) (see Figure 4.8(d)), and \( c \) is a capacity function defined as \( c(a) = \infty \) if \( a \in E \times (V \cup C) \), \( c(a) = 1 \) if \( a \in \{s\} \times E \), \( c(a) = d \) if \( a \in V \times \{t\} \), \( c(a) = 1 \) if \( a \in C \times \{n\} \), and \( c(a) = d \) if \( a = (n,t) \). Then, for any \( F \subseteq E \) and any node subset \( S \) of \( N \) satisfying \( s \in S \) and \( S \cap E = F \), the weight of the cut \( \delta_N(S) \) is at least \(|E \setminus F| + d|V(F)| + \min\{d, \chi(F)\}\) (and this value can be achieved). Therefore, by the max-flow min-cut theorem, \(|F| \leq d|V(F)| + \min\{d, \chi(F)\}\) holds for any \( F \subseteq E \) if and only if the value of a maximum \( s-t \) flow in \( N \) is equal to \(|E|\).
The extension of this algorithm for coping with the term $-d$ (i.e. checking $|F| \leq d |V(F)| - d + \min \{d, \chi(F)\}$) can be done in the same technique as above; the algorithm checks whether the value of a maximum flow of $N$ can be augmented from $|E|$ to $|E| + d$ when increasing the capacity of an arc $(s, e)$ from 1 to $d + 1$ for each $e \in E$. Therefore, one can check whether $|F| \leq d |V(F)| - d + \min \{d, \chi(F)\}$ is satisfied for any nonempty $F \subseteq E$ in $T_{\text{max-flow}}(N) + O(d |E||A|)$ time. Note that the cardinality of the arc set $A$ (and the node set) of $N$ is bounded by $O(|E| + |V|)$.

It can be shown that a maximum $s-t$ flow of $N$ can be computed in $O((d(|E| + |V|))^{3/2})$ time by replacing each arc $a$ with the capacity $c(a)$ by $c(a)$ parallel arcs with unit capacity. Namely, we shall construct an unweighted directed network $N'$ in which the value of a maximum $s-t$ flow is equal to that of $N$. Although the capacity of an arc $a$ with $a \in E \times V$ is unbounded in $N$, we may replace it by $d$ parallel arcs (because in any $s-t$ flow $f$ of $N$ the value $f(a)$ of $a = (e, v) \in E \times V$ is bounded by $d$). Similarly, we may replace an arc $a \in E \times \mathcal{C}$ by an arc with unit capacity. Therefore, we obtain an unweighted directed network $N'$ with the size $O(d(|E| + |V|))$. A maximum $s-t$ flow in an unweighted directed network with $m$ arcs can be computed in $O(m^{3/2})$ time by computing maximum arc-disjoint $s-t$ paths (see, e.g., [104, Corollary 9.6.a]). Thus, in our case, $T_{\text{max-flow}}(N) = O((d(|E| + |V|))^{3/2})$ holds.

Putting $|E| = d |V|$, the algorithms works in $O(d^3 |V|^2)$ time in total. This completes the proof of Theorem 4.16.
4.3. Algorithms

4.3.2 Detecting large tight sets

Let \( G = (V, E) \) be a loop-colored graph satisfying the counting condition (C2). In order to develop an algorithm for constructing a \((k, d)\)-rooted-forest partition of \( G \), we need to show how to compute an edge-inclusionwise minimal tight set and detect a large tight set if exists in \( G \). Recall that \( F \subseteq E \) is called tight if \(|F| = \mu_d(F)\) holds. By the submodularity of \( \mu_d \), for any two tight sets \( F_1 \) and \( F_2 \) with \( F_1 \cap F_2 \neq \emptyset \), we have \(|F_1 \cup F_2| + |F_1 \cap F_2| \leq \mu_d(F_1 \cup F_2) + \mu_d(F_1 \cap F_2) \leq \mu_d(F_1) + \mu_d(F_2) = |F_1| + |F_2| = |F_1 \cup F_2| + |F_1 \cap F_2| \), implying that both \( F_1 \cup F_2 \) and \( F_1 \cap F_2 \) are tight. Hence, for any edge \( e \), there exists a unique minimal/maximal tight set containing \( e \) (if there is a tight set containing \( e \)).

**Lemma 4.17.** Let \( G = (V, E) \) be a loop-colored graph satisfying the counting condition (C2), i.e., \( E \) is independent in \( \mathcal{M}_{\mu_0} \), and let \( e \in E \). Suppose that we know a maximum \( s-t \) flow \( f \) in the network \( N = (D, c) \) defined in Section 4.3.1. Then, the minimal/maximal tight set containing \( e \) can be computed in \( O(d^2|V|) \) time.

**Proof.** Let \( N_e = (D, c_e) \) be the network obtained from \( N \) by increasing the capacity of the arc \((s, e)\) from 1 to \( d + 1 \). Since \( E \) is independent, the value of a maximum \( s-t \) flow \( f_e \) in \( N_e \) is equal to \(|E| + d\), and \( f_e \) can be computed in \( O(d^2|V|) \) time from \( f \) by augmenting the flow \( d \) times (recall that the size of the network is \( O(d|V|) \)). Also, since all the capacities are integers, we can assume that \( f_e \) is integer-valued.

Let \( R_e \) be the residual digraph after computing a maximum \( s-t \) flow \( f_e \) by the Ford-Fulkerson max-flow algorithm. Namely, \( R_e \) is a digraph obtained from \( D \) by deleting an arc \( a_1 \) if \( f_e(a_1) = c_e(a_1) \) and by inserting the reverse arc of \( a_2 \) if \( f_e(a_2) > 0 \).

It is obvious that if \( e \) is reachable to \( t \) in \( R_e \), then \( G \) has no tight set containing \( e \). We hence assume that \( e \) is not reachable to \( t \) in \( R_e \). Let \( S \) be the set of nodes which are reachable from \( e \) in \( R_e \). Similarly, let \( T \) be the set of nodes which are not reachable to \( t \) in \( R_e \). Note \( e \in S \subseteq T \) since \( e \) is not reachable to \( t \) in \( R_e \). We claim that \( S \cap E \) and \( T \cap E \) are the minimal and maximal tight sets containing \( e \), respectively, (and hence the minimal and maximal tight sets can be computed in \( O(d|V|) \) time if we have \( f_e \)). We now provide a proof of this claim for the completeness.

Let us first consider \( S \). The followings are its properties.

**Claim 4.18.**

(i) \( s \in S \) and \( t \notin S \).

(ii) \( V \cap S = V(E \cap S) \).

(iii) \( |C \cap S| = \chi(E \cap S) \).

(iv) The special node \( v \) is contained in \( S \) if \(|C \cap S| > d \) (and not contained in \( S \) if \(|C \cap S| < d \)).

**Proof.** To see (i), notice that \( f_e((s, e)) = c_e((s, e)) \) holds, and hence the arc \((e, s)\) exists in \( R_e \). \( t \notin S \) follows from the assumption that \( e \) is not reachable to \( t \).

(ii) is because that \( R_e \) has an arc from \( e' \in E \) to \( v \in V \) if and only if \( v \) is an endpoint of \( e' \) in \( G \).
(iii) follows from the same reason as (ii).

To see (iv), recall that the leaving arcs from \( n \) in \( N_e \) has the capacity \( d \). Hence, if \(|C \cap S| > d\), there exists a node \( c_i \in S \cap C \) with \( f_e((c_i, n)) = 0 < c((c_i, n)) \), implying that \( n \) is reachable from a node \( c_i \in C \cap S \) and reachable from \( e \) in \( R_e \).

By (i), \( \delta_{N_e}(S) \) is an \( s-t \) cut in \( N_e \). Moreover, since there is no edge leaving from \( S \) in the residual graph, \( \delta_{N_e}(S) \) is actually a minimum \( s-t \) cut in \( N_e \). Thus, \( \delta_{N_e}(S) = |E| + d \).

Combining this equation and the properties of Claim 4.18, we obtain \(|E| + d = \delta_{N_e}(S) = |E \setminus (E \cap S)| + d|V \cap S| + \min\{d, |C \cap S|\} = |E| - |E \cap S| + d|V(E \cap S)| + \min\{d, \chi(E \cap S)\} \), implying \(|E \cap S| = \mu_d(E \cap S)\).

The minimality of \( E \cap S \) can be seen easily. Suppose, for a contradiction, that there is a tight set \( F \) with \( e \in F \subseteq E \cap S \). Let \( S' = F \cup V(F) \cup (C \cap F) \cup \{s\} \) if \( \chi(F) \leq d \) and otherwise \( S' = F \cup V(F) \cup (C \cap F) \cup \{s, n\} \). Then, by Claim 4.18, \( S' \subseteq S \) holds. Also, since \( F \) is tight, we have \( \delta_{N_e}(S') = |E \setminus F| + d|V(F)| + \min\{d, \chi(F)\} = |E| + d \). Hence, \( \delta_{N_e}(S') \) is a minimum \( s-t \) cut of \( N_e \). This implies that there is no leaving arc from \( S' \) in \( R_e \), implying that an element of \( S \setminus S'(\neq \emptyset) \) is not reachable from \( e \) in \( R_e \). This is a contradiction, and thus \( E \cap S \) is a minimal tight set.

Next we show that \( E \cap T \) is a maximal tight set. It is easy to see that the properties of Claim 4.18 also hold for \( T \):

**Claim 4.19.**

(i) \( s \in T \) and \( t \notin T \).

(ii) \( V \cap T = V(E \cap T) \).

(iii) \( |C \cap T| = \chi(E \cap T) \).

(iv) The special node \( n \) is contained in \( T \) if \( |C \cap T| \geq d \) and not contained in \( T \) if \(|C \cap T| < d\).

Furthermore, \( \delta_{N_e}(T) \) is a minimum \( s-t \) cut in \( N_e \). This is because that there is no arc entering to the complement of \( T \) in \( N_e \) by the definition of \( T \). Hence, in the same reason as \( S \cap E \), \( T \cap E \) must be tight.

To show the maximality let us take any set \( F \subseteq E \) with \( T \cap E \subseteq F \). Since \( v \in F \setminus T \) is reachable to \( t \), if we increase the capacity of the arc \((s, v)\) from 1 to 2 we can argument the flow \( f_e \) by one. Since the value of a flow is at most the weight of a cut, we have \(|E| + d + 1 \leq |E \setminus F| + d|V(F)| + \min\{d, \chi(F)\} \), implying \(|F| < \mu_d(F)\).

Recall that a tight set \( F \) is said to be large if \(|V(F)| \geq 2\).

**Lemma 4.20.** Let \( G = (V, E) \) be a loop-colored graph satisfying the counting condition (C2). Then, one can find a large and connected tight set \( F \) with \( V(F) \subseteq V \) (or report that no such edge set exists) in \( O(d^2|V||E \setminus L(E)|) \) time.

**Proof.** From the definition, any large and connected tight set contains a non-loop edge. Also, by Lemma 4.7, a minimal tight set containing an edge is always connected. These two observations imply that there exists a large and connected tight set \( F' \) with \( V(F') \subseteq V \) if and only if there exists a non-loop edge \( e \) such that a minimal tight set \( F' \) containing \( e \) satisfies
Compute a large and connected tight set \( F' \subseteq V \). Hence, by computing a minimal tight set \( F' \) containing \( e \) for all \( e \in E \setminus L(E) \), we can find a desired tight set or report that no such edge set exists. The time complexity follows from Lemma 4.17.

**Remark.** Let us consider a general loop-colored graph \( G = (V, E) \); \( E \) may be dependent in \( \mathcal{M}_{\mu d} \). Maximal tight sets help to speed up the detection of a maximal independent set contained in \( E \). Observe that, for an independent set \( I \subseteq E \) and a non-loop edge \( e \in E \setminus I \), \( I \cup \{e\} \) is independent if and only if there is no tight set \( F \subseteq I \) that spans the endpoints of \( e \). Similarly, if \( e \in E \setminus I \) is a loop, \( I \cup \{e\} \) is independent if and only if there is no tight set \( F \subseteq I \) that spans the endpoint of \( e \) with \( \chi(F \cup \{e\}) = \chi(F) \) (if \( \chi(F) < d \)). Therefore if we maintain all maximal tight sets contained in \( I \) and prepare a flog for each pair of vertices (and similarly for each pair of a vertex and a color) that answers whether there is a tight set that spans these two vertices, then we can answer whether \( I \cup \{e\} \) is independent or not in \( O(1) \) time. Such a technique of maintaining maximal tight sets was considered in e.g., \([17, 80]\), and in our case the total time complexity of detecting a base can be bounded by \( O(d^5|V|^2) \).

### 4.3.3 Constructing a \((k, d)\)-rooted-forest partition

Let us consider how to construct a \((k, d)\)-rooted-forest partition. Recall that the proof of the sufficiency of Theorem 4.1 is constructive; for a given loop-colored graph \( G = (V, E) \), it provides a way to construct a \((k, d)\)-rooted-forest partition recursively. We shall use a parameter \( t = |E \setminus L(E)| \) which measures the depth of the recursion, and \( T(t) \) denotes the time for computing a \((k, d)\)-rooted-forest partition for \( G \) with \( t = |E \setminus L(E)| \). Our algorithm is described as follows:

1. **Compute a large and connected tight set \( F \) with \( V(F) \subseteq V \) in \( O(d^2|V|t) \) time (Lemma 4.20). If one exists, then the algorithm splits the problem into two subproblems for \( F \) and \( E \setminus F \), and then combine partitions of \( F \) and \( E \setminus F \). The detailed procedure is split into two cases depending on whether \( \chi(F) < d \) or not:
   - **(i-i)** If \( F \) satisfies \( \chi(F) < d \), then compute a partition of \( F \) into \( \chi(F) \) rooted-forests and \( d - \chi(F) \) spanning trees (as in Theorem 4.6). Such a partition can be computed by the matroid union algorithm by Cunningham [33]. The Cunningham algorithm decomposes an independent set of the matroid union into independent sets of each matroid in \( O(m^{5/2}Q) \) time, where \( m \) is the size of a ground set and \( Q \) is the time needed for an independence test of each matroid. In our case, \( m = O(d|V|) \) and \( Q = O(|V|) \). A partition of \( E \setminus F \) can be computed by the recursion (see the proof of Lemmas 4.9 for more details). In total, it takes \( T(t - 1) + O(d^{5/2}|V|^{7/2}) \) time.
   - **(i-ii)** If \( \chi(F) \geq d \), then partitions of \( F \) and \( E \setminus F \) can be computed by the recursion as shown in the proof of Lemma 4.12. In this case, the total computation time becomes \( T(t - s) + T(s) + O(d^2|V|t) \) for some integer \( s \) with \( 1 \leq s \leq t - 1 \), where \( O(d^2|V|t) \) came from the time for computing \( F \).
(ii) If there exists no large connected tight set $F$ with $V(F) \subseteq V$ in $G$, then, by Lemma 4.15, $G$ contains two adjacent vertices $u$ and $v$ and a loop $l$ attached to $v$ such that removing an edge $e$ connecting $u$ and $v$ and then inserting a new loop to $u$ with the same color as $l$ results in a graph $G'$ satisfying the counting condition (C1)(C2). It is not difficult to compute such $u, v$ and $l$ in $O(d|V|kt)$ time by using the auxiliary network $N$ and its maximum flow $f$. Since the size of $|E \setminus L(E)|$ is decreased by one, a $(k, d)$-rooted-forest partition $E' = \{E'_1, \ldots, E'_k\}$ of $G'$ can be computed in $T(t - 1)$ time by the recursion. Let $E = \{E'_1, \ldots, E'_i-1, E'_i \setminus \{l'\} \cup \{e\}, E'_i+1, \ldots, E'_k\}$. Then $E$ is a $(k, d)$-rooted-forest partition of $G$. Therefore, it takes $T(t-1) + O(d^2|V|kt)$ time in total.

In the worst case (case (i-i)), we have $T(t) \leq T(t-1) + O(d^5/2|V|^{7/2})$, implying $T(t) \leq O(d^5/2|V|^{7/2}t)$. Therefore, the time complexity of our algorithm is bounded by $T(|E \setminus L(E)|) \leq T(d|V|) = O(d^{7/2}|V|^{9/2})$.

**Theorem 4.21.** Let $G = (V, E)$ be a loop-colored graph satisfying the counting condition, $k$ be the number of colors used in $G$, and $d$ be an integer with $d \geq k$. Then, one can find a $(k, d)$-rooted-forest partition in polynomial time.

### 4.4 Conclusion

We have proved a necessary and sufficient condition for a loop-colored graph to admit a $(k, d)$-rooted-forest partition, that is, a partition into $k$ edge-disjoint rooted-forests such that each vertex is spanned by exactly $d$ rooted-forests among them. Also, we have provided an efficient algorithm for checking whether a given graph satisfies the necessary and sufficient condition and that for constructing a $(k, d)$-rooted-forest partition explicitly. These results will be used in the next chapter for deriving a combinatorial characterization of the infinitesimal rigidity of bar-and-slider frameworks and for checking whether a frameworks is generically rigid or not.

We should remark that, to the best of our knowledge, few things are known about a general graph decomposition problem into edge-disjoint rooted-forests (i.e., no uniform vertex demand). The following problem is fundamental: given a loop-colored graph $G = (V, E)$ in which the total number of loop-colors are $k$, can we decide whether $E$ is partitioned into $k$ edge-disjoint rooted-forests in polynomial time? Of course, we know that deciding whether a graph can be decomposed into $k$ edge-disjoint rooted-forests can be solved in polynomial time by the matroid partition algorithm (see, e.g., [43]). The main difference between a forest and a rooted-forest is that a subset of a forest is also a forest while a subset of a rooted-forest is not necessarily a rooted-forest. Therefore, even though $E$ is independent in the matroid $M_{\sum \tau_i}$ (introduced in Section 4.2.1), we do not know whether $E$ is actually partitioned into $k$ edge-disjoint rooted-forests.
Chapter 5

On the Infinitesimal Rigidity of Bar-and-slider Frameworks

A bar-and-slider framework is a bar-and-joint framework a part of whose joints are constrained by using line-sliders. Such joints are allowed to move only along the sliders. Streinu and Theran [111] proved a combinatorial characterization of the infinitesimal rigidity of generic bar-and-slider frameworks in two dimensional space. In this chapter we propose a generalization of their result. In particular, we prove that, even though the directions of sliders are predetermined and degenerate, i.e., some sliders have the same direction, it is combinatorially decidable whether the framework is infinitesimally rigid or not. The proof is based on a rooted-forest partition proposed in Chapter 4.

5.1 Introduction

The celebrated Maxwell-Laman theorem (Theorem 3.14) states that, if \( p \) is generic, \( (G, p) \) is minimally rigid if and only if \( G \) satisfies \(|E| = 2|V| - 3\) and \(|F| \leq 2|V(F)| - 3\) for all nonempty \( F \subseteq E \). Instead of Laman’s counting condition, several equivalent characterizations are known as discussed in Section 3.1.4. Crapo’s theorem (Theorem 3.19) asserts that \( G \) is minimally rigid if and only if \(|E| = 2|V| - 3\) holds and \( E \) admits a proper 3tree2-partition, that is, a partition into three trees \( T_1, T_2, T_3 \) such that (i) each vertex is spanned by exactly two of them and (ii) any subtrees \( T'_i \subseteq T_i \) and \( T'_j \subseteq T_j \) with \( i \neq j \) does not span the same vertex subset. A partition of \( E \) into trees (or forests) is called proper if it satisfies condition (ii). We will use the term “proper” in a broad sense throughout this chapter.

Streinu and Theran [111] have extended Crapo’s characterization to bar-and-slider frameworks in a natural way. A bar-and-slider framework is a bar-and-joint framework a part of whose joints are constrained by using sliders. A slider at a vertex \( u \in V \) is a line, which constrains a point \( p(u) \) to be on this line. As in [81, 111], we shall handle each slider as a loop of a graph to extract the combinatorial aspect of frameworks. Let \( G = (V, E) \) be an undirected graph that may have some loops, and let us denote the set of loops in \( F \subseteq E \) by \( L(F) \) and the set of loops incident to a vertex \( u \in V \) by \( \delta_{L(E)}(u) \). Then, a bar-and-slider
A framework is defined as a triple \((G, p, d)\), where \(p : V \to \mathbb{R}^2\) is a joint configuration, and \(d : L(E) \to \mathbb{R}^2\) represents a direction of each slider. Namely, for \(u \in V\) and \(e \in \delta_{L(E)}(u)\), \(\{p(u) + td(e) : t \in \mathbb{R}\}\) is a line representing a slider incident to \(p(u)\), and \(p(u)\) is allowed to move along this line (see Figures 5.1).

A framework is \textit{minimally rigid} if removing any bar or slider results in a framework that is not rigid. The following is a result of [111].

\begin{align*}
\text{Theorem 5.1.} \quad \operatorname{[111]} & \quad \text{Let } G = (V, E) \text{ be an undirected graph. If } E \text{ can be partitioned into two colored classes } \{R, B\} \text{ such that} \\
\quad & \quad \text{(i) } \{R \setminus L(R), B \setminus L(B)\} \text{ is a proper 2forest partition}\footnote{In [111], this is called an \textit{induced-cut 2-forest} in order to emphasize the existence of a monochromatic cut in any induced subgraph.} \text{ of } E \setminus L(E), \text{ and} \\
\quad & \quad \text{(ii) each connected component of the graphs } (V, R) \text{ and } (V, B) \text{ contains exactly one loop of its color,} \\
\end{align*}

then there exist a joint configuration \(p\) and a direction mapping \(d\) such that \((G, p, d)\) is infinitesimally rigid. In particular, each of \(L(R)\) and \(L(B)\) is realized as a slider parallel to the \(x\)-axis and the \(y\)-axis, respectively.

Figure 5.1(c) shows an example of a partition of \(E\) satisfying (i) and (ii). As a corollary of the algorithm by Streinu and Theran [112] for checking the sparsity of a graph, it is known that \(E\) admits a proper bipartition satisfying (i)(ii) if and only if

\begin{align*}
\text{(L1)} & \quad |E| = 2|V|, \\
\text{(L2)} & \quad |F| \leq 2|V(F)| - 3 \text{ for every nonempty } F \subseteq E \setminus L(E), \text{ and} \\
\text{(L3)} & \quad |F| \leq 2|V(F)| \text{ for every } F \subseteq E.
\end{align*}

**Contribution.** We will provide an extension of this result. Theorem 5.1 says that, if \(E\) admits a proper bipartition \(\{R, B\}\) satisfying (i)(ii), then each of \(L(R)\) and \(L(B)\) is realized as an \(x\)-slider and a \(y\)-slider, respectively. It is not however obvious whether a specified loop is realized as either an \(x\)-or a \(y\)-slider until we actually construct a partition. In most practical situations, the directions of sliders are predetermined, and then we are not allowed to realize the predetermined \(x\)-slider as a \(y\)-slider or vice versa. This raises the following question. Given a set of joints connected by some bars as well as some sliders whose directions are specified
and moreover some of which may have the same direction (e.g. $x$-direction or $y$-direction), we would like to decide whether it is rigid or flexible.

Notice that, even though a joint configuration is generic, $(G, p, d)$ could be either rigid or flexible depending on $d$. (For example, consider a bar-and-slider framework all of whose sliders have the same direction. Then, the framework is definitely flexible no matter how bars and sliders are connected.) We will however prove that the generic rigidity does not actually depend on the specific value of $d$, and it is combinatorially decidable whether a framework is rigid or not even though the directions of sliders are “predetermined” and “degenerate”.

To extract a combinatorial aspect of this problem, we shall consider a loop-colored graph introduced in Chapter 4, in which each loop has some predetermined color. In this chapter, to specify the coloring of loops, we shall denote a loop-colored graph by $G = (V, E, c)$, where $c$ is a mapping from $L(F)$ to a finite set $C$ of colors. We then redefine a bar-and-slider framework as a triple $(G, p, d)$, where $G$ is a loop-colored graph, $p$ is a joint configuration, and $d$ is a direction mapping from $C$ to $\mathbb{R}^2$ (not from $L(E)$) such that $d(c)$ and $d(c')$ are linearly independent for any pair of distinct colors $c, c' \in C$. A loop colored in $c \in C$ is supposed to be realized as a slider with the direction $d(c)$ (see Figure 5.2). A main result of this chapter is stated as follows.

**Theorem 5.2.** Let $G = (V, E, c)$ be a loop-colored graph. Then, for any direction mapping $d$ and for any generic joint-configuration $p$, the bar-and-slider framework $(G, d, p)$ is infinitesimally minimally rigid if and only if $G$ satisfies (L1)\sim(L3) as well as

$$\text{(L4)} \quad |F| \leq 2|V(F)| - 1 \text{ for any } F \subseteq E \text{ such that all loops of } L(F) \text{ are monochromatically colored.}$$

This theorem claims a stronger statement than Theorem 5.1; our result reveals that in any degenerate situation of slider sets the generic rigidity can be characterized by a simple system of linear inequalities, while Theorem 5.1 only proves the existence of a direction mapping $d$ that attains a rigid realization; the result of Theorem 5.1 cannot be applied to situations where directions of sliders are predetermined and degenerate.

**Related works.** As for the related works of bar-and-slider frameworks, we should mention the pinning problem of a bar-and-joint framework in the plane. In this problem, given a bar-and-joint framework (having certain degree of freedom), we would like to stabilize it by fixing the positions of the smallest number of joints. Lovász [83] showed, as an application of

Figure 5.2: A loop-colored graph $G$ and a bar-and-slider framework $(G, p, d)$. 

and Figure 5.2: A loop-colored graph $G$ and a bar-and-slider framework $(G, p, d)$.
his matroid matching algorithm, that the pinning problem can be reduced to a 2-polymatroid matching problem, and is solvable in polynomial time. Fekete [35] provided a simpler min-max characterization of the optimal value in generic case. Fekete and Jordán further discussed in [36] the pinning problem from the viewpoint of generic global rigidity. Notice that pinning down a joint reduces the degree of freedom of a framework by at most two, while attaching a slider at some joint reduces the degree of freedom by at most one. In fact, attaching a slider seems easier to handle than pinning a joint and we thereby obtain a much clearer combinatorial characterization of the rigidity of bar-and-slider frameworks.

Servatius, Shai and Whiteley [106] have recently presented a counting condition for the pinned bar-and-joint framework. Since such a pinning can be handled by attaching two distinct sliders to a vertex simultaneously, the result of [106] is a special case of bar-and-slider frameworks. We should remark that the notable result of [106] is a characterization of assur graphs and its geometric realization, where a pinned rigid framework is called an assur graph if there exists no proper pinned rigid subgraph in it.

Concerning direction-constraints, Servatius and Whiteley [107] considered an extension of Laman’s theorem in which some of bars constrains the direction of two endpoints rather than the length. However, the result can apply only to a generic joint configuration, which means that the directions given by the constraints are assumed to be distinct. Further results extended to the higher dimensional case or the global rigidity can be seen in [129] and [63].

Organization of this chapter. In Section 5.2 we define the infinitesimal rigidity of bar-and-slider frameworks. The necessity of Theorem 5.2 follows straightforwardly from the definition of the rigidity (in the same way as Theorem 3.8). In Section 5.3.1 we shall propose a generalization of Crapo’s theorem in Theorem 5.4, and a Henneberg-type constructive characterization will be proposed in Section 5.3.2. In Section 5.4, we will provide a proof of the sufficiency of Theorem 5.2 based on the \((k, d)\)-rooted-forest-partition theorem given in Chapter 4.

5.2 Infinitesimal Rigidity of Bar-and-slider Frameworks

Recall that a bar-and-joint framework is defined as a pair \((G, p)\) of a graph \(G\) and a joint configuration \(p : V \to \mathbb{R}^2\). An infinitesimal motion of \((G, p)\) is defined as an assignment \(v : V \to \mathbb{R}^2\) of a 2-dimensional vector for each joint \(p(v)\) such that

\[
(p(v) - p(u)) \cdot (v(v) - v(u)) = 0 \quad \text{for each } uv \in E. \tag{5.1}
\]

We refer to (5.1) as the length constraint by the bar \(p(u)p(v)\).

On the other hand, a bar-and-slider framework is defined as a triple \((G, p, d)\), where \(G\) is a loop-colored graph, and \(d\) is a direction mapping from a set of colors to \(\mathbb{R}^2\).

Rigidity of a bar-and-slider framework is defined in a way similar to the case of a bar-and-joint framework, but it counts even trivial motions as its degree of freedom. \((G, p, d)\) is rigid if there exists no continuous motion of \(p\) that converts to a distinct framework under
bar length constraints as well as slider constraints. Again, we shall consider the first-order rigidity of this concept: An infinitesimal motion \( v : V \to \mathbb{R}^2 \) of \((G, p, d)\) satisfies (5.1) as well as direction constraints written by

\[
v(u) \cdot d(c(e)) = 0 \quad \text{for each } u \in V \text{ and } e \in \delta_L(E)(u),
\]

(5.2)

where \( d(c(e)) \) denotes a vector orthogonal to \( d(c(e)) \). Taking new rows corresponding to the direction constraints (5.2) into account, we obtain the rigidity matrix \( R(G, p, d) \), whose size becomes \( |E| \times 2|V| \). It is called infinitesimally rigid if no infinitesimal motion exists (except for 0), equivalently the rank of \( R(G, p, d) \) is equal to \( 2|V| \).

A joint configuration \( p \) is said to be generic (with respect to a given \( G \) and a given \( d \)) if the rank of \( R(G, p, d) \) as well as those of row-induced submatrices take the maximum values over all joint configurations.

The following result supplies the necessity of Theorem 5.2.

**Theorem 5.3.** Let \( G \) be a loop-colored graph, \( d \) be any direction mapping, and \( p \) be a generic joint configuration with respect to \( G \) and \( d \). If \((G, p, d)\) is infinitesimally minimally rigid in 2-dimensional space, then \( G \) satisfies (L1) to (L4).

**Proof.** Since \((G, p, d)\) is infinitesimally rigid, \( \text{rank} R(G, p, d) = 2|V| \) holds. Also, due to the minimality, the row vectors of \( R(G, p, d) \) are linearly independent. These imply \( |E| = 2|V| \) and (L1).

For \( F \subseteq E \setminus L(E) \), we can apply the result of bar-and-joint frameworks: Theorem 3.13 implies \( |F| \leq 2|V(F)| - 3 \) for any \( F \subseteq E \setminus L(E) \), and hence (L2) holds.

To see (L3), take any \( F \subseteq E \) and consider the subframework \((G[F], p, d)\) induced by \( F \), (where we denote the restrictions of \( p \) and \( d \) to \( V(F) \) and to \( L(F) \) simply by \( p \) and \( d \) again). Since \( F \) is independent in \( R(G, p, d) \), it is also independent in \( R(G[F], p, d) \). Since \( \text{rank} R(G[F], p, d) \leq 2|V(F)| \) holds from the definition of the rigidity matrix, we obtain \( |F| \leq 2|V(F)| \).

To see (L4), suppose that there is \( F \subseteq E \) such that \( |F| = 2|V(F)| \) and all loops of \( F \) are monochromatically colored. Consider again the subframework \((G[F], p, d)\) induced by \( F \). Since \( F \) is independent in \( R(G, p, d) \), we have \( \text{rank} R(G[F], p, d) = |F| = 2|V(F)| \). Hence, \((G[F], p, d)\) is infinitesimally rigid, i.e., it is fixed on the plane. However, since all loops of \( F \) are monochromatically colored, say colored in \( c \), there exists a nontrivial infinitesimal motion \( v : V \to \mathbb{R}^2 \) of \((G[F], p, d)\) such that \( v(u) = d(c) \) for \( u \in V \). This contradicts that \((G[F], p, d)\) is infinitesimally rigid. Thus \( |F| \leq 2|V(F)| - 1 \) must hold. \( \square \)

5.3 Combinatorial Results

Let \( G = (V, E, c) \) be a loop-colored graph, and \( \{c_1, c_2, \ldots, c_k\} \) be the set of colors appearing in \( G \). We say that \( G \) satisfies the strong counting condition if it satisfies (L1) to (L4). In this section we shall reveal properties of graphs satisfying the strong counting condition based on results of Chapter 4.
On the Infinitesimal Rigidity of Bar-and-slider Frameworks

5.3.1 Crapo-type characterization

Recall Theorem 4.1 with \(d = 2\); a loop-colored graph \(G = (V, E, c)\) admits a \((k, 2)\)-rooted-forest partition \(\{E_1, \ldots, E_k\}\), i.e.,

\[(P1) \text{ each } E_i \text{ is a rooted-forest colored in } c_i, \text{ and} \]
\[(P2) \text{ each vertex is spanned by exactly 2 components} \]

if and only if \(|E| = 2|V| \text{ and } |F| \leq 2|V(F)| - 2 + \min\{2, \chi(F)\}\) for any nonempty \(F \subseteq E\) (where \(\chi(F)\) denotes the total number of colors appearing in \(L(F)\)). The latter condition is equivalent to the following four counting conditions; (L1), (L3), (L4), and

\[(L2') \text{ } |F| \leq 2|V(F)| - 2 \text{ for any nonempty } F \subseteq E \setminus L(E). \]

Let us refer to the set of the counting conditions (L1), (L2'), (L3), and (L4) as the weak counting condition (that was called just the counting condition in Chapter 4). Note that the weak counting condition is properly weaker than the strong counting condition, i.e., the strong condition implies the weak one.

As before, a \((k, 2)\)-rooted-forest partition is said to be proper if no two subtrees from \(E_i \setminus L(E_i)\) and \(E_j \setminus L(E_j)\) span the same set of vertices for any \(1 \leq i, j \leq k\) with \(i \neq j\).

Figure 5.3 shows examples of \((k, 2)\)-rooted-forest partitions.

The following forest partition theorem generalizes a result of Crapo (Theorem 3.19):

**Theorem 5.4.** Let \(G = (V, E, c)\) be a loop-colored graph, and let \(k\) be the number of colors used in \(G\). Then, \(G\) satisfies the strong counting condition if and only if \(E\) admits a proper \((k, 2)\)-rooted-forest partition.

**Proof.** Suppose that \(G\) satisfies the strong counting condition, then it also satisfies the weak counting condition. Hence, by Theorem 4.1, \(E\) admits a \((k, 2)\)-rooted-forest partition \(\mathcal{E} = \{E_1, \ldots, E_k\}\). Suppose, for a contradiction, that \(\mathcal{E}\) is not proper. Then, there exist two subtrees, say \(T_1 \subseteq E_1 \setminus L(E_1)\) and \(T_2 \subseteq E_2 \setminus L(E_2)\), which span the same set of vertices. Hence, we have \(|T_1 \cup T_2| = |V(T_1)| - 1 + |V(T_2)| - 1 = 2|V(T_1 \cup T_2)| - 2\) with \(L(T_1 \cup T_2) = \emptyset\), contradicting condition (L2).

Suppose that \(G\) admits a proper \((k, 2)\)-rooted-forest partition \(\mathcal{E} = \{E_1, \ldots, E_k\}\). Then, by Theorem 4.1, \(G\) satisfies (L1), (L3), and (L4). Let us show (L2). For any \(F \subseteq E\), let \(F_i = F \cap E_i, i = 1, \ldots, k\), and let \(l = |\{i : F_i \neq \emptyset\}|\). Without loss of generality we assume that

Figure 5.3: (a) A proper \((k, 2)\)-rooted-forest partition for \(k = 3\). (b) A non-proper \((k, 2)\)-rooted-forest partition for \(k = 3\).
Suppose, for a contradiction, that there exists $F \subseteq E$ satisfying $L(F) = \emptyset$ and $|F| \geq 2|V(F)| - 2$ simultaneously. Since $L(F) = \emptyset$, $F_i$ is a forest, and hence we have $|F_i| \leq |V(F_i)| - 1$ for each $1 \leq i \leq l$. Thus, $l \geq 2$. Also, we have $|F| = \sum_{i=1}^{l} |F_i| \leq (\sum_{i=1}^{l} |V(F_i)|) - l \leq 2|V(F)| - l$, which implies $l = 2$ and $|F| = 2|V(F)| - 2$. However, this implies that $F_1$ and $F_2$ are both spanning trees on $V(F)$, contradicting that $E$ is proper. Thus, (L2) holds. \hfill \square

Recall that an edge set satisfying the weak counting condition is a base of the matroid induced by the integer-valued nondecreasing submodular function $\mu_2$ written as

$$\mu_2(F) = 2|V(F)| - 2 + \min\{\chi(F), 2\} \quad (F \subseteq E). \quad (5.3)$$

Similarly, it can be seen that an edge set satisfying the strong counting condition is a base of the matroid induced by $\mu'_2 : 2^E \rightarrow \mathbb{Z}$ defined by

$$\mu'_2(F) = \begin{cases} 
q_{2,3}(F) = 2|V(F)| - 3 & \text{if } L(F) = \emptyset \\
\mu_2(F) = 2|V(F)| - 2 + \min\{\chi(F), 2\} & \text{otherwise}. 
\end{cases} \quad (5.4)$$

### 5.3.2 Henneberg-type characterization

In order to prove Theorem 5.2, we introduce one more characterization of graphs satisfying the strong counting condition.

We refer to a vertex $v$ as a $(a, b)$-vertex in $G$ if $\delta_{E \setminus L(E)}(v) = a$ and $\delta_{L(E)}(v) = b$. We now show that any $(k, 2)$-rooted-forest partition can be constructed by a sequence of six operations (including two Henneberg operations) from the empty graph. Suppose that a loop-colored graph $G = (V, E, c)$ satisfies the strong counting condition, and each edge is colored according to a proper $(k, 2)$-rooted-forest partition. Let us consider the following six operations, each of which extends not only the graph but also the coloring of the edge set by newly inserting a $(a, b)$-vertex (see Figure 5.4):

**$(0,2)$-extension**: Add a new vertex $v$ with two new loops $l_1$ and $l_2$ attached to $v$. Color $l_1$ and $l_2$ in different ways.

**$(1,1)$-extension**: Choose a vertex $x \in V$. Let $c_1$ and $c_2$ be colors incident to $x$ in $G$. Add a new vertex $v$ with a loop $l$ incident to $v$, and connect $vx$. Color $vx$ in $c_1$ and $l$ in $c_2$.

**$(1,2)$-extension**: Choose a loop $e \in L(E)$, and let $x \in V$ be a vertex incident to $e$ in $G$. Let $c_1$ be the color of $e$, and $c_2$ be another color incident to $x$ in $G$. Remove $e$ from $G$, add a new vertex $v$ with two loops $l_1$ and $l_2$ incident to $v$ and connect $vx$. Color $vx$ and $l_1$ in $c_1$ and $l_2$ in $c_2$.

**$(2,1)$-extension**: Choose a loop $e \in L(E)$, and let $x \in V$ be a vertex incident to $e$ in $G$. Choose a vertex $y \in V$ other than $x$. Let $c_1$ be the color of $e$, and $c_2$ be a color incident to $y$ different from $c_1$. Remove $e$ from $G$, add a new vertex $v$ with a loop $l$ incident to $v$, and connect $vx$ and $vy$. Color $vx$ and $l$ in $c_1$ and $vy$ in $c_2$. 


On the Infinitesimal Rigidity of Bar-and-slider Frameworks

(2,0)-extension (0-extension): Choose two distinct vertices $x, y \in V$. Let $c_1$ be a color incident to $x$ and $c_2$ be a color incident to $y$ different from $c_1$. Add a new vertex $v$, and connect $vx$ and $vy$. Color $vx$ in $c_1$ and $vy$ in $c_2$.

(3,0)-extension (1-extension): Choose a non-loop edge $e = xy$ and a vertex $z \in V$ other than $x, y$. Let $c_1$ be a color of $e$ and $c_2$ be a color incident to $z$ different from $c_1$. Remove $e$, add a new vertex $v$, and connect $vx, vy, vz$. Color $vx, vy$ in $c_1$ and $vz$ in $c_2$.

When referring to some (or any) operation among the six, we shall call it an $(a, b)$-extension. A loop-colored graph $G'$ obtained by an $(a, b)$-extension means the resulting graph of each operation (ignoring the colors of non-loop edges). Let us first claim our result.

**Theorem 5.5.** A loop-colored graph $G$ satisfies the strong counting condition if and only if $G$ can be obtained from the empty graph by a sequence of $(a, b)$-extensions.

**Corollary 5.6.** Any proper $(k, 2)$-rooted-forest partition can be constructed by a sequence of $(a, b)$-extensions.

It is easy to check that each operation extends a proper $(k, 2)$-rooted-forest partition of the edge set, and thus the resulting graph satisfies the strong counting condition by Theorem 5.4. This implies the sufficiency of Theorem 5.5:

**Lemma 5.7.** Let $G$ be a loop-colored graph satisfying the strong counting condition, and $G'$ be a loop-colored graph that is obtained from $G$ by an $(a, b)$-extension. Then, $G'$ satisfies the strong counting condition.

In order to prove the necessity of Theorem 5.5, we now consider the inverse operation of an $(a, b)$-extension. We shall first consider the inverse operation of an $(3, 0)$-extension.
Lemma 5.8. Let $G$ be a loop-colored graph satisfying the strong counting condition, and $v$ be a $(3,0)$-vertex in $G$ with the neighbors $N_G(v) = \{x,y,z\} \subset V$. Then, there is an edge $e \in \{xy, yz, zx\}$ such that $G - v + e$ satisfies the strong counting condition (where $G - v + e$ is the graph obtained by the inverse operation of the $(3,0)$-extension at $v$).

Proof. For the proof, we define $\alpha : 2^E \to \{0,1\}$ such that $\alpha(F) = 1$ if $F$ contains a loop, and otherwise $\alpha(F) = 0$. Notice

$$\mu_2'(F) = 2|V(F)| - 3 + \alpha(F) + \min\{2, \chi(F)\} \quad (F \subseteq E).$$

An edge subset $F$ is said to be $\mu_2'$-tight if $\mu_2'(F) = |F|$.

Suppose, for a contradiction, that there exists no edge $e \in \{xy, yz, zx\}$ such that $G - v + e$ satisfies the strong counting condition. Since $G - v + xy$ does not satisfy the strong counting condition, there exists $F_1 \subseteq E$ that violates the strong counting condition (i.e., $|F_1 + xy| > \mu_2'(F_1 + xy)$) with $x, y \in V(F_1)$ and $v \notin V(F_1)$. Note $|F_1| > \mu_2'(F_1 + xy) - 1 \geq \mu_2'(F_1) - 1$. Note also $|F_1| \leq \mu_2'(F_1)$ because $G$ satisfies the strong counting condition. Therefore, $|F_1| = \mu_2'(F)$ holds, and hence $F_1$ is $\mu_2'$-tight. Symmetrically, there exist $\mu_2'$-tight sets $F_2, F_3 \subseteq E$ such that $y, z \in V(F_2)$ and $z, x \in V(F_3)$. We take inclusion-wise-maximal sets of these properties as $F_1, F_2, F_3$, respectively.

Let us prove the following two claims.

Claim 5.9. $z \notin V(F_1), x \notin V(F_2), \text{ and } y \notin V(F_3)$.

Proof. Suppose $z \in V(F_1)$. Let $F = F_1 + vx + vy + vz$. It is obvious that $\alpha(F) = \alpha(F_1)$ and $\chi(F) = \chi(F_1)$. Hence, we have

$$|F| = |F_1| + 3 = \mu_2'(F_1) + 3$$
$$= 2|V(F_1)| + \alpha(F_1) + \min\{2, \chi(F_1)\}$$
$$= 2|V(F) + v| - 2 + \alpha(F) + \min\{2, \chi(F)\}$$
$$= \mu_2'(F) + 1.$$ 

This contradicts that $G$ satisfies the strong counting condition. The other case is symmetric. 

Claim 5.10. $V(F_1) \cap V(F_2) = \{y\}, V(F_2) \cap V(F_3) = \{z\}, \text{ and } V(F_3) \cap V(F_1) = \{x\}$.

Proof. Let us show $V(F_1) \cap V(F_2) = \{y\}$. If $F_1 \cap F_2 \neq \emptyset$, then by the submodularity of $\mu_2'$ we have $|F_1| + |F_2| = \mu_2'(F_1) + \mu_2'(F_2) \geq \mu_2'(F_1 \cup F_2) + \mu_2'(F_1 \cap F_2) \geq |F_1 \cup F_2| + |F_1 \cap F_2| = |F_1| + |F_2|$. Hence, the equality holds everywhere, in particular $|F_1 \cup F_2| = \mu_2'(F_1 \cup F_2)$. Namely, $F_1 \cup F_2$ is $\mu_2'$-tight, which contradicts the maximality of $F_1$. 


In addition, by using $\alpha(F_1) + \alpha(F_2) \geq \alpha(F_1 \cup F_2)$ and $\min\{2, \chi(F_1)\} + \min\{2, \chi(F_2)\} \geq \min\{2, \chi(F_1 \cup F_2)\}$, we have

$$\mu'_2(F_1 \cup F_2) \geq |F_1 \cup F_2| = |F_1| + |F_2| \quad \text{(by } F_1 \cap F_2 = \emptyset),$$

$$= \mu'_2(F_1) + \mu'_2(F_2)$$

$$= \sum_{i=1,2,3} (2|V(F_i)| - 3 + \alpha(F_i) + \min\{2, \chi(F_i)\})$$

$$\geq 2(|V(F_1 \cup F_2)| + |V(F_1) \cap V(F_2)|) - 6 + \alpha(F_1 \cup F_2)$$

$$+ \min\{2, \chi(F_1 \cup F_2)\}$$

$$= \mu'_2(F_1 \cup F_2) + 2|V(F_1) \cap V(F_2)| - 3).$$

This implies $|V(F_1) \cap V(F_2)| \leq 1$. Since $\{x, y\} \subseteq V(F_1)$ and $\{y, z\} \subseteq V(F_2)$, we obtain $\{y\} = V(F_1) \cap V(F_2)$. \hfill \Box

From these two claims, we can show that $F_1 \cup F_2 \cup F_3$ is also $\mu'_2$-tight:

$$|F_1 \cup F_2 \cup F_3| = |F_1| + |F_2| + |F_3| \quad \text{(by Claim 5.10)}$$

$$= \sum_{i=1,2,3} (2|V(F_i)| - 3 + \alpha(F_i) + \min\{2, \chi(F_i)\})$$

$$\geq \sum_{i=1,2,3} (2|V(F_i)| - 3 + \alpha(F_1 \cup F_2 \cup F_3) + \min\{2, \chi(F_1 \cup F_2 \cup F_3)\}$$

$$\geq 2|V(F_1 \cup F_2 \cup F_3)| - 6 + \alpha(F_1 \cup F_2 \cup F_3) + \min\{2, \chi(F_1 \cup F_2 \cup F_3)\}$$

$$+ \sum_{i,j \in \{1,2,3\}, i \neq j} 2|V(F_i) \cap V(F_j)| - 6$$

$$= \mu'_2(F_1 \cup F_2 \cup F_3) + \mu'_2(F_1 \cup F_2 \cup F_3)$$

Since $F_1 \subseteq F_1 \cup F_2 \cup F_3$, this contradicts the maximality of $F_1$. Thus, there exists an edge $e \in \{xy, yz, zx\}$ such that $G - v + e$ satisfies the strong counting condition. \hfill \Box

The following lemma claims the existence of small degree vertices.

**Lemma 5.11.** Let $G = (V, E, c)$ be a connected loop-colored graph with $|V| > 1$ satisfying the weak counting condition. Then, $G$ has at least one $(1,1)$, $(1,2)$, $(2,1)$, $(2,0)$ or $(3,0)$-vertex.

**Proof.** Let $l = |L(E)|$ and $E' = E \setminus L(E)$. Then, by (L1) and (L2'), we have $|E'| + l = 2|V|$ and $l \geq 2$. Let $n_i = |\{v \in V : |\delta_{E'}(v)| = i\}|$ for $i \geq 0$. Since $G$ is connected, $n_0 = 0$. Counting the total degrees of the vertices in $(V, E')$, we have $\sum_{i=1}^{\infty} n_i = 2|E'|$. Also, $\sum_{i=1}^{\infty} n_i = |V|$ is obvious. Hence, substituting them into $|E'| + l = 2|V|$, we obtain $(\sum_{i=1}^{\infty} n_i) + 2l = 4|V|$, implying $3n_1 + 2n_2 + n_3 \geq 2l$. If there exists a vertex $v$ such that $\delta_{E'}(v) = 1$, then we can easily check $v$ is either a $(1,1)$- or $(1,2)$-vertex by (L3). Hence, let us assume $n_1 = 0$. Then, $2n_2 + n_3 \geq 2l$ holds and at least one of $n_2 > 0$ or $n_3 > 0$ holds since $l \geq 2$.

Let $l_2$ and $l_3$ be the total number of loops incident to the vertices of $\{v : |\delta_{E'}(v)| = 2\}$ and $\{v : |\delta_{E'}(v)| = 3\}$, respectively. Suppose for a contradiction that $G$ has no $(2,1)$-, $(2,0)$- and $(3,0)$-vertices. Then, we have $l_2 \geq 2n_2$ and $l_3 \geq n_3$. Hence, $2n_2 + n_3 \geq 2l \geq 2(l_2 + l_3) \geq 4n_2 + 2n_3$, implying that $n_2 = n_3 = 0$. However, this contradicts that either $n_2 > 0$ or $n_3 > 0$ holds. \hfill \Box
The following lemma ensures that, if no \((1, 1), (1, 2), (2, 0), (3, 0)\)-vertices exist, then there exists a \((2, 1)\)-vertex at which the inverse operation of a \((2, 1)\)-extension can be performed.

**Lemma 5.12.** Let \(G = (V, E, c)\) be a connected loop-colored graph with \(|V| > 1\) satisfying the weak counting condition. Suppose that there exists no \((1, 1), (1, 2), (2, 0), (3, 0)\)-vertex. Then, for any \((k, 2)\)-rooted-forest partition \(E\) of \(G\), there exists a \((2, 1)\)-vertex \(v\) such that the two non-loop edges incident to \(v\) belong to distinct components of \(E\).

**Proof.** Let \(E = \{E_1, E_2, \ldots, E_k\}\), and let \(V'\) be the set of \((2, 1)\)-vertices contained in \(G\). By Lemma 5.11, \(V' \neq \emptyset\).

The proof proceeds by induction on \(|V|\). When \(|V| = 2\), due to the lemma assumption, \(G\) consists of two \((2, 1)\)-vertices \(u\) and \(v\), one loop for each vertex (with the distinct colors) and two parallel edges between \(u\) and \(v\). In this case, clearly the two parallel edges belong to the different components of \(E\) by (P1), and hence both \(u\) and \(v\) satisfy the property of the statement.

Let us consider the general case. When \(|V \setminus V'| = 0\), all the vertices are \((2, 1)\)-vertices, and hence \(E \setminus L(E)\) forms a cycle (since \(G\) is connected). In this case, since each component induces a forest, there exists a vertex \(v\) that is incident to two non-loop edges of the distinct colors. Therefore, the statement follows.

Let us consider the case of \(|V \setminus V'| > 0\). In this case, since \(V' \neq \emptyset\) and \(V \setminus V' \neq \emptyset\), there exists a sequence \(v_0v_1 \ldots v_{j+1}\) of vertices with \(j \geq 1\) satisfying the following three properties:

(i) \(v_i\) is a \((2, 1)\)-vertices for each \(1 \leq i \leq j\),
(ii) \(v_i\) is not a \((2, 1)\)-vertices for \(i = 0, j + 1\), and
(iii) \(v_iv_{i+1} \in E\) for each \(0 \leq i \leq j\). Namely, \(v_0v_1 \ldots v_{j+1}\) is a maximal chain of \((2, 1)\)-vertices.

We can assume that all the edges \(v_iv_{i+1}\) for \(0 \leq i \leq j\) belong to the same component of \(E\), say \(E_1\) (colored in \(c_1\)), since otherwise some \(v_i\) becomes a \((2, 1)\)-vertex satisfying the property of the statement. Let \(G'\) be the colored graph obtained from \(G\) by removing \(v_1, v_2, \ldots, v_j\) and then inserting a new edge between \(v_0\) and \(v_{j+1}\). Notice that \(G'\) has no \((1, 1), (1, 2), (2, 0), (3, 0)\)-vertex, and neither \(v_0\) nor \(v_{j+1}\) is a \((2, 1)\)-vertex because the number of edges incident to each vertex does not differ between \(G\) and \(G'_{c_1}\). Moreover, when constructing \(G'\) from \(G\), if we color the new edge \(v_0v_{j+1}\) in \(c_1\), then we get a \((k, 2)\)-rooted-forest partition \(E'\) of \(G'_{c_1}\). Hence \(G'\) satisfies the weak counting condition by Theorem 4.1, and by induction \(G'\) has a \((2, 1)\)-vertex \(v\) such that the two non-loop edges incident to \(v\) belong to the distinct components of \(E'\). Since neither \(v_0\) nor \(v_{j+1}\) is a \((2, 1)\)-vertex in \(G'\), \(v\) differs from \(v_0\) and \(v_{j+1}\). This implies that \(v\) is incident to the same set of edges in \(G\) as in \(G'\) (with the same coloring), and hence \(v\) is a \((2, 1)\)-vertex in \(G\) satisfying the property of the statement. \(\square\)

Now we are ready to show Theorem 5.5.

**Proof of Theorem 5.5.** The sufficiency has been done by Lemma 5.7.

Let \(G = (V, E, c)\) be a loop-colored graph satisfying the strong counting condition. The proof proceeds by induction on \(|V|\). The base case (i.e. \(|V| = 0\)) is trivial. Let us consider the case of \(|V| > 0\). Also, we may assume that \(G\) is connected since otherwise we are done by applying the induction to each connected subgraphs.
By Theorem 5.4, $G$ admits a proper $(k, 2)$-rooted-forest partition $\mathcal{E}$, and hence we assume that every edge is colored according to the partition $\mathcal{E}$.

Let us consider the case when $G$ has a $(1, 2)$-vertex $v$. Consider the graph $G' = (V - v, E', c')$ and a partition $\mathcal{E}'$ of $G'$ obtained from $G$ by the inverse operation of the $(1, 2)$-extension at $v$ as shown in Figure 5.4(c) (from right to left). Then, it is easy to check that $\mathcal{E}'$ satisfies (P1) and (P2), and hence it is a $(k, 2)$-rooted-forest partition. Moreover, since $\mathcal{E}$ is proper, $\mathcal{E}'$ is also proper. (Notice that, when converting from $G$ to $G'$, only one loop is inserted, which does not affect the properness condition of the partition.) Thus, $\mathcal{E}'$ is a proper $(k, 2)$-rooted-forest partition, and hence $G'$ satisfies the strong counting condition by Theorem 5.4. By induction, we have a sequence of $(a, b)$-extensions that constructs $G'$ from the empty graph. Consequently, $G$ can be constructed by a sequence of $(a, b)$-extensions.

Similarly, if $G$ has a $(0, 2)$-, $(1, 1)$-, $(1, 2)$-, or $(2, 0)$-vertex $v$, then we apply the inverse operation of the corresponding $(a, b)$-extension at $v$. It can be seen that the resulting graph $G'$ satisfies the strong counting condition by using Theorem 5.4. Applying the induction, we obtain a sequence of $(a, b)$-extensions that constructs $G$.

If $G$ has a $(3, 0)$-vertex $v$, then $G$ can be converted to a smaller graph $G'$ by the inverse operation of the $(3, 0)$-extension at $v$ by Lemma 5.8. Applying the induction to $G'$, we obtain a sequence of $(a, b)$-extensions that constructs $G$.

If $G$ has no $(1, 1)$-, $(1, 2)$-, $(2, 0)$-, $(3, 0)$-vertex, then by Lemma 5.12 there exists a $(2, 1)$-vertex $v$ such that the two non-loop edges incident to $v$ belong to distinct components of $\mathcal{E}$. Hence, at $v$, we can perform the inverse operation of a $(2, 1)$-extension (including the edge-coloring). The resulting partition $\mathcal{E}'$ is a proper $(k, 2)$-rooted-forest partition, and hence the resulting graph $G'$ satisfies the strong counting condition by Theorem 5.4. Applying the induction to $G'$, we obtain a sequence of $(a, b)$-extension that constructs $G$. This completes the proof.

\[\square\]

5.4 Infinitesimally Rigid Bar-and-slider Frameworks

We now provide a proof of the following theorem.

**Theorem 5.13.** Let $G$ be a loop-colored graph satisfying the strong counting condition, and let $d$ be a direction mapping on the set of colors appearing in $G$. Then, there exists a joint configuration $p$ such that the bar-and-slider framework $(G, p, d)$ is infinitesimally rigid in the plane.

Note that, if one particular realization $(G, p, d)$ is rigid, then $(G, q, d)$ becomes rigid for all generic joint configurations $q$ as explained in Section 3.1.3, and hence Theorem 5.13 implies the sufficiency of Theorem 5.2. Hence, our task for completing the proof of Theorem 5.2 is to show Theorem 5.13. Our proof is basically the same as the proof of Laman’s theorem by Tay and Whiteley [116] or Whiteley [129], which is done by induction on $|V|$ based on the Henneberg construction (see Section 3.1.4).
Proof of Theorem 5.13. The proof is done by induction on $|V|$ as follows. The base case is trivial. Let us consider the case of $|V| > 0$. By Theorem 5.5, there is a sequence of $(a, b)$-extensions that constructs $G$. Let $G'$ be the last loop-colored graph in this sequence before $G$. By induction, there is a rigid realization $(G', p', d)$. Therefore, if we prove that each $(a, b)$-extension preserves the infinitesimal rigidity, then we obtain a rigid realization of $G$, completing the proof.

Recall a $0$-extension mentioned in Section 3.1.4. It is known that, when performing a $0$-extension at the geometric level, it preserves the infinitesimal rigidity of a 2-dimensional bar-and-joint framework if we put the new joint $p(a)$ in such a way that it does not lie on the line passing through $p(a)$ and $p(b)$, see [116, Proposition 3.1] or [129, Lemma 2.1.3]. This result can be extended to bar-and-slider case without any modification to create a vertex (joint) of degree two, i.e., a $(2, 0)$-vertex. Also, it is known that a 1-extension (mentioned in Section 3.1.4) creates a degree-three-vertex preserving the infinitesimal rigidity, and the proof of [116, Proposition 3.2] or [129, Theorem 2.2.2] can apply to a $(3, 0)$-vertex for our purpose. We thus omit the detailed description on the case where $G$ is obtained from $G'$ by a $(2, 0)$-extension or a $(3, 0)$-extension. Also, we omit a trivial case where $G$ is obtained by a $(0, 2)$-extension and prove the remaining three cases.

(i, $(1, 1)$-extension) Suppose that $G$ is obtained from $G'$ by a $(1, 1)$-extension, where a new $(1, 1)$-vertex $v$ is inserted with a loop $e_1$ and a non-loop edge $vx$ for some $x \in V \setminus \{v\}$ (see Figure 5.5). By induction, we have a rigid realization $(G', p', d)$ of $G'$. We define a joint configuration $p : V \to \mathbb{R}^2$ as follows: $p(u) = p'(u)$ for $u \in V \setminus \{v\}$ and $p(v)$ is any vector in $\mathbb{R}^2$ such that $p(v) - p(x)$ is not orthogonal to the direction $d(c(e_1))$, that is, the direction of the slider associated with $e_1$.

We claim that $(G, p, d)$ is rigid. Note that the subframework of $(G, p, d)$ induced by $V \setminus \{v\}$ is infinitesimally rigid, because this subframework is the same framework as $(G', p', d')$ that is rigid. Hence, if there exists an infinitesimal motion $v : V \to \mathbb{R}^2$ for $(G, p, d)$, then $v(u) = 0$ must hold for any $u \in V \setminus \{v\}$. Moreover, due to the length constraint (5.1) by the bar $p(v)p(x)$, $v(x) = 0$ implies that $v(v)$ is orthogonal to $p(v) - p(x)$. Hence, the direction constraint by the slider associated with $e_1$ forces $v(v)$ to be 0. Therefore, $(G, p, d)$ is rigid.

(ii, $(1, 2)$-extension) Suppose that $G$ is obtained from $G'$ by a $(1, 2)$-extension, where (as shown in Figure 5.6) a loop $e_3$ with the color $c_1$ attached to $x \in V \setminus \{v\}$ is removed, and then a $(1, 2)$-vertex $v$ is inserted with the non-loop edge $vx$ and two loops $e_1$ and $e_2$ attached to $v$ colored in $c_1$ and $c_2$, respectively.

Define a joint configuration $p : V \to \mathbb{R}^2$ as follows: $p(u) = p'(u)$ for $u \in V \setminus \{v\}$ and $p(v)$ is a vector in $\mathbb{R}^2$ such that $p(v) - p(x)$ is orthogonal to the direction $d(c_1)$ of the slider associated with $e_3$ (that is the same direction as that of the slider associated with $e_1$).

We claim that $(G, p, d)$ is rigid. Suppose not, and there exists a nonzero infinitesimal motion $v : V \to \mathbb{R}^2$. Let $v|_{V \setminus \{v\}}$ be the restriction of $v$ to $V \setminus \{v\}$. Since the joint $p(v)$ is incident to two distinct sliders in $(G, p, d)$, $v(v) = 0$ must hold. If $v(x)$ is also zero, then $v|_{V \setminus \{v\}}$ must be a nonzero infinitesimal motion for $(G', p', d)$, which is a contradiction. Therefore, $v(x)$ is nonzero, and is orthogonal to $p(v) - p(x)$ because $v$ must satisfy the
length constraint (5.1) by the bar $p(v)p(x)$. This implies that $v(x)$ can also satisfy the direction constraint (5.2) by the slider associated with $e_3$ whose direction $d(c_1)$ is orthogonal to $p(v) - p(x)$. Since all the edges of $G'$ except for $e_3$ are contained in $G$, $v$ satisfies all the length and direction constraints appearing in $(G', p', d)$. Therefore, $v|_{V \setminus \{v\}}$ is a nonzero infinitesimal motion for $(G', p', d)$, contradicting that $(G', p', d)$ is rigid.

(iii, $(2,1)$-extension) Let us consider the final case. Suppose that $G$ is obtained from $G'$ by a $(2,1)$-extension, where (as shown in Figure 5.7) a loop $f_1$ with the color $c_1$ attached to $x \in V \setminus \{v\}$ is removed, and then a $(2,1)$-vertex $v$ is inserted with a loop $e_1$ attached to $v$ colored in $c_1$ and two non-loop edges $vx$ and $vy$. Define $p$ as follows; $p(u) = p(u')$ for $u \in V \setminus \{v\}$ and $p(v)$ is a vector in $\mathbb{R}^2$ such that $p(v) - p(x)$ is orthogonal to the direction $d(c_1)$ (as shown in Figure 5.7).

We claim that $(G, p, d)$ is infinitesimally rigid. To see this, suppose that there exists a nonzero infinitesimal motion $v$ for $(G, p, d)$. Due to the direction constraint by the slider associated with $e_1$, $v(v)$ is a scalar multiple of $d(c_1)$, and consequently $v(x)$ is also a scalar
5.5 Conclusion

We have presented combinatorial characterizations of the infinitesimal rigidity of bar-and-slider frameworks in terms of a counting condition, a graph-decomposition, and an inductive construction that generalize results of bar-and-joint frameworks given in Section 3.1.4.

We should remark that in Theorem 4.16 we have presented an algorithm that can decide whether a given loop-colored graph satisfies the weak counting condition or not in $O(|V|^2)$ time. More generally, in Section 4.3.2 we showed that a maximal independent set of the matroid $\mathcal{M}_{\mu_2}$ can be detected in $O(|V|^2)$ time. Of course, using the same auxiliary network, we can detect a maximal independent set of the generic rigidity matroid in $O(|V|^2)$ time [55]. Therefore, for a given loop-colored graph $G = (V, E)$, a maximal independent set of $E$ in $\mathcal{M}_{\mu_2}$ can be detected in $O(|V|^2)$ time: first compute a maximal independent set $F_1 \subseteq E \setminus L(E)$ in the rigidity matroid and then compute a maximal loop set $F_2 \subseteq L(E)$ such that $F_1 \cup F_2$ is independent in $\mathcal{M}_{\mu_2}$ by the method of Section 4.3.2. This implies that, assuming a generic joint configuration, we can compute the degree of freedom of a bar-and-slider framework in $O(|V|^2)$ time.

Recall that in Chapter 4 we have proved a necessary and sufficient condition for the existence of a $(k, d)$-rooted-forest partition for general $d$ as a generalization of the Tutte-
Nash-Williams tree-packing theorem. Although detailed description is omitted here, applying Whiteley’s proof technique given in [127], it can be easily observed that \((k, d)\)-rooted-forest partitions characterize infinitesimal rigidity of \(d\)-dimensional body-and-bar frameworks with the external constraints.
Chapter 6

A Proof of the Molecular Conjecture

A \(d\)-dimensional body-and-hinge framework introduced in Section 3.2 is the collection of \(d\)-dimensional rigid bodies connected by hinges. The generic infinitesimal rigidity of a body-and-hinge framework has been characterized in terms of the underlying multigraph independently by Tay [114] and Whiteley [127] as follows: A multigraph \(G\) can be realized as an infinitesimally rigid body-and-hinge framework by mapping each vertex to a body and each edge to a hinge if and only if \(\left(\binom{d+1}{2} - 1\right)\) \(G\) contains \(\binom{d+1}{2}\) edge-disjoint spanning trees, where \(\left(\binom{d+1}{2} - 1\right)\) \(G\) is the graph obtained from \(G\) by replacing each edge by \(\left(\binom{d+1}{2} - 1\right)\) parallel edges. In 1984 they jointly posed a question about whether their combinatorial characterization can be further applied to a nongeneric case [115]. Specifically, they conjectured that \(G\) can be realized as an infinitesimally rigid body-and-hinge framework if and only if \(G\) can be realized as that with the additional “hinge-coplanar” property, i.e., all the hinges incident to each body are contained in a common hyperplane. This conjecture is known as the Molecular Conjecture due to the equivalence between the infinitesimal rigidity of “hinge-coplanar” body-and-hinge frameworks and that of bar-and-joint frameworks derived from molecules in 3-dimension. In this chapter we prove this long-standing conjecture affirmatively for general dimension.

6.1 Introduction

Let \(G = (V, E)\) be a multigraph which may contain multiple edges. A body-and-hinge framework is defined as a pair \((G, \mathbf{q})\) where \(\mathbf{q}\) is a mapping from \(e \in E\) to a \((d - 2)\)-dimensional affine subspace \(\mathbf{q}(e)\) in \(\mathbb{R}^d\). The framework \((G, \mathbf{q})\) is called a body-and-hinge realization of \(G\) in \(\mathbb{R}^d\).

As we have already seen in Section 3.2.2, the infinitesimal rigidity of body-and-hinge frameworks can be formulated in terms of a linear homogeneous system by using the fact that any continuous rotation of a point around a \((d - 2)\)-dimensional affine subspace or any translation to a fixed direction can be described by a \(\binom{d+1}{2}\)-dimensional vector, so-called a screw center,
Figure 6.1: A panel-and-hinge framework.

and the degree of freedom is determined by the rank of the associated rigidity matrix given in (3.7).

We assume that the dimension $d$ is a fixed integer with $d \geq 2$, and we shall use the notation $D$ to denote $\binom{d+1}{2}$ (as in Section 3.2). For a multigraph $G = (V, E)$ and a positive integer $k$, the graph obtained by replacing each edge by $k$ parallel edges is denoted by $kG$. In this chapter, for our special interest in $(D-1)G$, we shall use the simple notation $\tilde{G}$ to denote $(D - 1)G$ and let $\tilde{E}$ be the edge set of $\tilde{G}$. Tay-Whiteley’s theorem (Theorem 3.25) asserts that a multigraph $G$ can be realized as an infinitesimally rigid body-and-hinge framework in $\mathbb{R}^d$ if and only if $\tilde{G}$ contains $D$ edge-disjoint spanning trees.

A body-and-hinge framework $(G, p)$ is called hinge-coplanar if, for each $v \in V$, all of the $(d - 2)$-dimensional affine subspaces $p(e)$ for the edges $e$ incident to $v$ are contained in a common $(d - 1)$-dimensional affine subspace (i.e. a hyperplane). In this case replacing each body by a rigid panel does not change the rigidity of the framework. Thus, a hinge-coplanar body-and-hinge framework is said to be a panel-and-hinge framework (see Figure 6.1(c)).

Note that “hinge-coplanarity” is a restriction on hinge configurations; the resulting configuration spaces are necessarily nongeneric. Therefore, there may be no rigid panel-and-hinge framework $(G, p)$ even if $G$ satisfies the tree packing condition of Tay-Whiteley’s theorem (Theorem 3.25). In 1984, Tay and Whiteley [115] jointly posed the following question:

**Conjecture 6.1.** ([115]) Let $G = (V, E)$ be a multigraph. Then, $G$ can be realized as an infinitesimally rigid body-and-hinge framework in $\mathbb{R}^d$ if and only if $G$ can be realized as an infinitesimally rigid panel-and-hinge framework in $\mathbb{R}^d$.

Conjecture 6.1 is known as the Molecular Conjecture which has appeared in several different forms [129, 132] and has been a long standing open problem in rigidity theory. For the case when $d = 2$, Whiteley [128] proved the conjecture affirmatively for a special class of multigraphs in 1989 and recently the conjecture has been completely proved by Jackson and Jordán [61]. In [61] they replaced each body of a panel-and-hinge framework by a rigid bar-and-joint framework and reduced the problem to that for bar-and-joint frameworks. By using well-investigated properties of 2-dimensional bar-and-joint frameworks, they successfully proved the conjecture. Also, Jackson and Jordán [62] showed a sufficient condition of the graph to have a panel-and-hinge realization in higher dimension; $G$ has a panel-and-hinge realization in $\mathbb{R}^d$ if $(d - 1)G$ has $d$ edge-disjoint spanning trees.

**Contribution.** We settle the Molecular Conjecture affirmatively in general dimension. Although the overall strategy of our proof resembles that of [61], their proof relies on a combina-
torial characterization of 2-dimensional bar-and-joint frameworks whose general dimensional counterpart is not known.

In $\mathbb{R}^3$ the rigidity of panel-and-hinge frameworks has a special connection with the flexibility of molecules, which we have briefly explained in Section 1.2.2; A molecule can be modeled as a bar-and-joint framework of the square of a graph, where the square of a graph $G = (V, E)$ is defined as $G^2 = (V, E^2)$ with $E^2 = E \cup \{uv \in V \times V | u \neq v \text{ and } uw, wv \in E \text{ for some } w \in V \setminus \{u, v\}\}$. A molecule can be also modeled as a body-and-hinge framework by regarding each atom (vertex) as a rigid body and each bond (edge) as a hinge since in the square of a graph a vertex and its neighbor always form a complete graph. Notice however that this body-and-hinge framework has a “special” hinge configuration, i.e., all the hinges (lines) incident to a body are intersecting each other at the center of the body. Such a hinge configuration is called \textit{hinge-concurrent}, and thus a molecule can be modeled as a hinge-concurrent body-and-hinge framework [115, 131–133].

Projective duality reveals the reason why Conjecture 6.1 is called the “Molecular Conjecture”. Recall that taking projective dual in $\mathbb{R}^3$ transforms points to planes, lines to lines, and planes to points preserving their incidences. This means that the dual of a hinge-concurrent body-and-hinge framework is exactly a panel-and-hinge framework. Crapo and Whiteley [31] showed that infinitesimal rigidity is invariant under projective duality, which implies that $G$ has an infinitesimally rigid panel-and-hinge realization if and only if it has an infinitesimally rigid hinge-concurrent body-and-hinge realization. Therefore, the correctness of the Molecular Conjecture suggests that the flexibility of proteins can be combinatorially investigated by using the well-developed tree-packing algorithm on the underlying graphs. In fact, the so-called “pebble game” algorithm [80, 112] for packing spanning trees is implemented in several softwares, e.g., FIRST [39, 66], ROCK [82] and others [6, 122]. Our result provides the theoretical validity of the algorithms behind such softwares.

\textbf{Organization.} The formal definition and detailed explanation on the infinitesimal rigidity of body-and-hinge frameworks has been given in Section 3.2. In Section 6.2, we will briefly discuss the relation between the deficiency of graphs and the degree of freedom of body-and-hinge frameworks. In Section 6.3 and Section 6.4, we will investigate the combinatorial property of multigraphs $G$ such that $\tilde{G}$ contains $D$ edge-disjoint spanning trees. Such graphs are called \textit{body-and-hinge rigid graphs}. In particular, edge-inclusionwise minimal body-and-hinge rigid graphs are called \textit{minimally body-and-hinge rigid graphs}. In Section 6.4, we will show that any minimally body-and-hinge rigid graph can be reduced to a smaller minimally body-and-hinge rigid graph by the contraction of a rigid subgraph or a splitting off operation (defined in Section 6.4) at a vertex of degree two. This implies that any minimally body-and-hinge rigid graph can be constructed from a smaller body-and-hinge rigid graph by the inversions of these two operations. Finally, in Section 6.5 and Section 6.6, we will provide a proof of the Molecular Conjecture by showing that any minimally body-and-hinge rigid graph $G$ has a rigid panel-and-hinge realization. The proof is done by induction on the graph size. More precisely, following the construction of a graph proposed in Section 6.4, we convert $G$ to
a smaller minimally body-and-hinge rigid graph $G'$. By the induction hypothesis there exists a rigid panel-and-hinge realization of $G'$. We will show that we can extend this realization to that of $G$ with a slight modification so that the resulting framework becomes rigid.

### 6.2 Deficiency and Degree of Freedom

Recall that, for $X \subseteq V$, $\delta_G(X)$ denotes $\{uv \in E \mid u \in X, v \notin X\}$, and let $d_G(X) = |\delta_G(X)|$. Throughout this chapter, a partition $\mathcal{P}$ of $V$ is a collection $\{V_1, V_2, \ldots, V_m\}$ of vertex subsets for some positive integer $m$ such that $V_i \neq \emptyset$ for $1 \leq i \leq m$, $V_i \cap V_j = \emptyset$ for any $1 \leq i, j \leq m$ with $i \neq j$ and $\bigcup_{i=1}^{m} V_i = V$. Note that $\{V\}$ is a partition of $V$ for $m = 1$. Let $\delta_G(\mathcal{P})$ and $d_G(\mathcal{P})$ denote the set, and the number, of edges of $G$ connecting distinct subsets of $\mathcal{P}$, respectively. Similarly, let $\delta_F(\mathcal{P})$ and $d_F(\mathcal{P})$ denote the set, and the number, of edges among $F$ connecting distinct components of $\mathcal{P}$.

Theorem 3.25 states a strong relation between the infinitesimal rigidity of body-and-hinge frameworks and edge-disjoint spanning trees. To further investigate the combinatorial aspect of body-and-hinge frameworks, we need to introduce an equivalent variation of the Tutte-Nash-Williams tree-packing theorem (Theorem 2.11).

**Theorem 6.2.** ([89, 118]) A multigraph $H = (V, E)$ contains $c$ edge-disjoint spanning trees if and only if $d_H(\mathcal{P}) \geq c(|\mathcal{P}| - 1)$ holds for each partition $\mathcal{P}$ of $V$.

**Proof.** ("only if"-part:) Suppose that there exist $c$ edge-disjoint spanning trees $T_1, \ldots, T_c$. Let $\mathcal{P}$ be a partition of $V$. Then, it is easy to see that $d_{T_i}(\mathcal{P}) \geq |\mathcal{P}| - 1$ for any $1 \leq i \leq c$. Hence we have $d_H(\mathcal{P}) \geq \sum_{i=1}^{c} d_{T_i}(\mathcal{P}) \geq c(|\mathcal{P}| - 1)$.

("if"-part:) Suppose that $d_H(\mathcal{P}) \geq c(|\mathcal{P}| - 1)$ holds for each partition $\mathcal{P}$ of $V$. We remark that, by setting $\mathcal{P} = \{\{u\} : u \in V\}$, we have $|E| = d_H(\mathcal{P}) \geq c(|V| - 1)$. We first claim the following:

**Claim 6.3.** If $|E| > c(|V| - 1)$, then there is an edge $e' \in E$ such that $d_{H-e'}(\mathcal{P}) \geq c(|\mathcal{P}| - 1)$ for any partition $\mathcal{P}$ of $V$.

**Proof.** Let us take an edge $e \in E$. If $d_{H-e}(\mathcal{P}) \geq c(|\mathcal{P}| - 1)$ for any $\mathcal{P}$, then we are done. Otherwise, there exists a partition $\mathcal{P}_e$ of $V$ such that $e \in \delta_H(\mathcal{P}_e)$ and

$$d_H(\mathcal{P}_e) = c(|\mathcal{P}_e| - 1). \quad (6.1)$$

We shall take such a partition $\mathcal{P}_e$ so that $|\mathcal{P}_e|$ is largest. Denote $\mathcal{P}_e = \{V_1, V_2, \ldots, V_m\}$. From $|E| > c(|V| - 1)$, it follows that $\mathcal{P}_e$ has some component $V_i$ for which $H[V_i]$ contains an edge, say $e'$. (since otherwise $|E| = d_H(\mathcal{P}_e) = c(|V| - 1)$ holds by (6.1)). Without loss of generality, we assume that $V_1$ is such a component. We shall show that $e'$ is a desired edge of the statement.

Suppose, for a contradiction, that there is a partition $\mathcal{P}_{e'} = \{V_1', V_2', \ldots, V_{m'}\}$ of $V$ such that $e' \in \delta_H(\mathcal{P}_{e'})$ and

$$d_H(\mathcal{P}_{e'}) = c(|\mathcal{P}_{e'}| - 1). \quad (6.2)$$

Suppose, for a contradiction, that there is a partition $\mathcal{P}_{e'} = \{V_1', V_2', \ldots, V_{m'}\}$ of $V$ such that $e' \in \delta_H(\mathcal{P}_{e'})$ and

$$d_H(\mathcal{P}_{e'}) = c(|\mathcal{P}_{e'}| - 1). \quad (6.2)$$
Without loss of generality, we assume that there is an integer $k$ such that $V'_i \cap V_1 \neq \emptyset$ for $1 \leq i \leq k$ and $V'_i \cap V_1 = \emptyset$ for $k + 1 \leq i \leq m'$. Note that $k \geq 2$ since $e' \in \delta_H(P_{e'})$. Let $\mathcal{P}_{V_1} = \{V'_1 \cap V_1, \ldots, V'_k \cap V_1\}$. Then, $\mathcal{P}_{V_1}$ is a partition of $V_1$. Also, define a partition $Q_e$ of $V$ by $\{V'_1 \cap V_1, V'_2 \cap V_1, \ldots, V'_k \cap V_1, V_2, V_3, \ldots, V_m\}$. Notice that $k \geq 2$ implies $|Q_e| > |P_e|$, and hence the maximality of $|P_e|$ implies

$$d_H(Q_e) > c(|Q_e| - 1). \quad (6.3)$$

Notice also

$$d_H(Q_e) = d_H(P_e) + d_H(V_1)(P_{V_1}). \quad (6.4)$$

Hence, by (6.1), (6.3), (6.4), and $|Q_e| = |P_e| + |P_{V_1}| - 1$, we obtain

$$d_H(V_1)(P_{V_1}) = d_H(Q_e) - d_H(P_e) > c(|Q_e| - |P_e| + 1) - 1) = c(|P_{V_1}| - 1). \quad (6.5)$$

On the other hand, denoting a partition $\{(\bigcup_{i=1}^k V'_i), V'_{k+1}, V'_{k+2}, \ldots, V'_m\}$ of $V$ by $Q_{e'}$, we have

$$d_H(Q_{e'}) \leq d_H(P_{e'}) - d_H(V_1)(P_{V_1})$$

$$< c(|P_{e'}| - 1) - c(|P_{V_1}| - 1) \quad (\text{by (6.2) and (6.5)})$$

$$= c(|Q_{e'}| - 1) \quad (\text{by } |Q_{e'}| + |P_{V_1}| - 1 = |P_{e'}|).$$

This contradicts that $d_H(P) \geq c(|P| - 1)$ for any partition $P$ of $V$.

Thus, for any partition $P$ such that $e' \in \delta_H(P)$, we have $d_H(P) > c(|P| - 1)$, implying that $d_{H-e'}(P) \geq c(|P| - 1)$ for any partition $P$ of $V$. \hfill \square

By applying Claim 6.3 repeatedly, we obtain an edge subset $E' \subseteq E$ such that $|E'| = c(|V| - 1)$ and $d_{E'}(P) \geq c(|P| - 1)$ for any partition $P$. Let us show that $E'$ can be partitioned into $c$ edge-disjoint spanning trees on $V$.

For any nonempty $F \subseteq E'$, let $P_F$ be the partition of $V$ defined as $\{\{u\} : u \in V \setminus V(F)\} \cup \{V(F)\}$. Then, we have

$$|P_F| = |\{\{u\} : u \in V \setminus V(F)\} \cup \{V(F)\}| = |V \setminus V(F)| + 1,$$

implying

$$|V| = |P_F| + |V(F)| - 1.$$ 

Similarly, it is not difficult to see

$$|E'| \geq d_{E'}(P_F) + |F|.$$ 

Thus, we obtain

$$|F| \leq |E'| - d_{E'}(P_F) \leq c(|V| - 1) - c(|P_F| - 1) = c(|V(F)| - 1).$$

Namely, for any nonempty $F \subseteq E'$, $|F| \leq c(|V(F)| - 1)$ holds, implying that $E'$ can be partitioned into $c$ edge-disjoint spanning trees by Theorem 2.11. This completes the proof. \hfill \square
We use the following conventional notation. For a partition $\mathcal{P}$ of $V$ and a multigraph $G$, the $D$-deficiency of $\mathcal{P}$ in $G$ is defined by

$$\text{def}_G(\mathcal{P}) = D(|\mathcal{P}| - 1) - d_G(\mathcal{P})$$

and the $D$-deficiency of $G$ is defined by

$$\text{def}(G) = \max\{\text{def}_G(\mathcal{P}) : \mathcal{P} \text{ is a partition of } V\}.$$ 

Note that $\text{def}(G) \geq 0$ since $\text{def}_G(\{V\}) = 0$. Theorem 6.2 implies that $e_G$ has $D$ edge-disjoint spanning trees if and only if $\text{def}(e_G) = 0$.

Another well-known characterization of an edge set containing $D$ edge-disjoint spanning trees is written in terms of matroids as given in Chapter 2. Let us consider the matroid $M_{\varrho_D,D}(e_G)$ on $e_E$ induced by the nondecreasing submodular function $\varrho_{D,D} : 2^E \to \mathbb{Z}$ written by $\varrho_{D,D}(F) = D|V(F)| - D$ for $F \subseteq E$ (c.f. Section 2.7). Let us denote $M_{\varrho_D,D}(e_G)$ simply by $M(e_G)$ throughout this chapter. In the proof of Section 2.11, we have seen that $M(e_G)$ is the union of $D$ graphic matroids on $E$, which means that an edge set is independent if and only if it can be partitioned into $D$ edge-disjoint forests.

Theorem 3.25 now implies that a multigraph $G$ can be realized as an infinitesimally rigid body-and-hinge framework if and only if the rank of $M(e_G)$ is equal to $D(|V| - 1)$. A more detailed relation between the deficiency of a graph and the degree of freedom of a body-and-hinge framework can be found in [62]. Let us summarize these preliminary results.

**Proposition 6.4.** ([62, 114, 127]) The followings are equivalent for a multigraph $G = (V, E)$:

(i) A generic body-and-hinge framework $(G, p)$ has $k$ degree of freedom.

(ii) A generic body-and-hinge framework $(G, p)$ satisfies $R(G, p) = D(|V| - 1) - k$.

(iii) $\text{def}(G) = k$.

(iv) The rank of $M(G)$ is equal to $D(|V| - 1) - k$, i.e., a base of $M(G)$ can be partitioned into $D$ edge-disjoint forests whose total cardinality is equal to $D(|V| - 1) - k$.

Let us note the following relation, observed from (iii) and (iv), between the deficiency and the cardinality of a base of $M(G)$: for a multigraph $G = (V, E)$ and a base $B$ of $M(G)$,

$$|B| + \text{def}(G) = D(|V| - 1). \quad (6.6)$$

This is equivalent to the forest packing theorem described in [104, Theorem 51.1].

### 6.3 Body-and-hinge Rigid Graphs

In this section we shall further investigate combinatorial properties of body-and-hinge frameworks. Let $G = (V, E)$ be a multigraph. Proposition 6.4 says that a graph $G$ satisfying $\text{def}(G) = k$ for some integer $k$ can be realized as a generic body-and-hinge framework having
6.3. Body-and-hinge Rigid Graphs

$k$ degree of freedom. Inspired by this fact, we simply say that $G$ is a \textit{k-dof-graph} if $\text{def}(\tilde{G}) = k$ for some nonnegative integer $k$. In particular, to emphasize the relation between 0-dof-graphs and infinitesimal rigidity given in Proposition 3.25, we sometimes refer to a 0-dof-graph as a body-and-hinge rigid graph.

Recall that $E$ denotes the edge set of $eG$. For $e \in E$, let $\tilde{e}$ denote the set of corresponding $D-1$ parallel copies of $e$ in $\tilde{E}$. For $F \subseteq E$, let $\tilde{F} = \bigcup_{e \in F} \tilde{e}$. We index the edges of $\tilde{e}$ by $1 \leq i \leq D-1$, and $e_i$, or $(e)_i$, denotes the $i$-th element in $\tilde{e}$. It is not difficult to see the following fact.

**Lemma 6.5.** Let $G$ be a body-and-hinge rigid graph. Then, $G$ is 2-edge-connected.

**Proof.** Suppose that $G$ is not 2-edge-connected. Then, there exists a nonempty subset $V'$ of $V$ satisfying $d_G(V') \leq 1$ and $V \setminus V' \neq \emptyset$. Consider a partition $\mathcal{P} = \{V', V \setminus V'\}$ of $V$. Then, we have $\text{def}(\tilde{G}) \geq D(|\mathcal{P}| - 1) - (D - 1)d_G(\mathcal{P}) \geq 1$, contradicting $\text{def}(\tilde{G}) = 0$.

**Remark.** Let $b$ and $c$ be positive integers and let $q = b/c$. A multigraph $G = (V, E)$ is called \textit{q-strong} if $cG$ contains $b$ edge-disjoint spanning trees. Such graphs were first introduced by Gusfield [47], where he considered the maximum value of $q$ for $G$ to be q-strong. This value was later called the \textit{strength} of $G$ by Cunningham [32]. Checking whether $G$ is q-strong or not can be solved in polynomial time, if $q$ is regarded as a constant, by explicitly constructing $cG$ and checking the existence of $b$ edge-disjoint spanning trees in it, which can be efficiently done by a forest packing algorithm (and hence it can be checked in polynomial time whether $G$ is a body-and-hinge rigid graph). Cunningham [32] provided a strongly polynomial time algorithm for checking whether $G$ is q-strong and also computing the strength of $G$. The concept of strength has been extended to a general matroid by Catlin et al. [23].

In this paper, for our particular interest in $q = \frac{D}{D-1}$, we named a $\frac{D}{D-1}$-strong graph as a body-and-hinge rigid graph. For the rigidity of body-and-hinge frameworks, Jackson and Jordán [57, 59] recently provided several results on the $q$-strength of multigraphs, which are basically concerned with partitions of $V$ maximizing the deficiency. They also defined a \textit{minimally q-strong graph} (with respect to edge-inclusion) and showed that a q-strong subgraph contained in a minimally q-strong graph is also minimal, which is a special case of Lemma 6.7 given in the next subsection.

### 6.3.1 Minimally body-and-hinge rigid graphs

A \textit{minimal k-dof-graph} is a $k$-dof-graph in which removing any edge results in a graph that is not a $k$-dof-graph. In particular, a minimal 0-dof-graph is called a \textit{minimally body-and-hinge rigid graph}. In this section we prove several new combinatorial properties of a minimal $k$-dof-graph, which will be utilized in the proof of the Molecular Conjecture.

**Lemma 6.6.** Let $G$ be a minimal $k$-dof-graph for some nonnegative integer $k$. Then, $G$ is not 3-edge-connected.
Proof. Suppose, for a contradiction, that $G$ is 3-edge-connected. We shall show that the graph $G_e$ obtained by removing an edge $e = uv \in E$ is still a $k$-dof-graph, which contradicts the minimality of $G$.

Consider any partition $\mathcal{P} = \{V_1, V_2, \ldots, V_{|\mathcal{P}|}\}$ of $V$. If $u$ and $v$ are both in the same vertex subset of $\mathcal{P}$, then $d_G(\mathcal{P}) = d_{G_e}(\mathcal{P})$, and consequently $\text{def}_{\tilde{G}}(\mathcal{P}) = \text{def}_{\tilde{G_e}}(\mathcal{P}) \leq k$ holds.

Suppose $u$ and $v$ are contained in distinct subsets of $\mathcal{P}$. Without loss of generality, we assume that $u \in V_1$ and $v \in V_2$. Since $G$ is 3-edge-connected, we have $d_{G_e}(V_i) \geq 2$ for $i = 1, 2$ and $d_{G_e}(V_i) \geq 3$ for $i = 3, \ldots, |\mathcal{P}|$. Hence, we have $d_{G_e}(\mathcal{P}) \geq \left\lceil \frac{3(|\mathcal{P}|-2)+2}{2} \right\rceil \geq \frac{3}{2}|\mathcal{P}|-1$, which implies,

$$\text{def}_{\tilde{G_e}}(\mathcal{P}) \leq D(|\mathcal{P}|-1) - (D-1)\left(\frac{3}{2}|\mathcal{P}|-1\right) = -\frac{|\mathcal{P}|}{2}(D-3) - 1.$$  

Since $d \geq 2$ and $D \geq 3$, we obtain $\text{def}_{\tilde{G_e}}(\mathcal{P}) < 0 \leq k$. Consequently, $\text{def}_{\tilde{G_e}}(\mathcal{P}) \leq k$ holds for any partition $\mathcal{P}$ of $V$, implying that $G_e$ is a $k$-dof-graph and contradicting the minimality of $G$.

For a multigraph $G = (V, E)$, (6.6) implies that an edge $e \in E$ can be removed from $G$ without changing the deficiency of $\tilde{G}$ if and only if there exists a base $B$ of the matroid $\mathcal{M}(\tilde{G})$ such that $B \cap \tilde{e} = \emptyset$. Equivalently, a graph $G = (V, E)$ is a minimal $k$-dof-graph for some nonnegative integer $k$ if and only if $B \cap \tilde{e} \neq \emptyset$ for any edge $e \in E$ and any base $B$ of $\mathcal{M}(\tilde{G})$.

From this observation, it is not difficult to see the following fact.

**Lemma 6.7.** Let $G = (V, E)$ be a minimal $k$-dof-graph for some nonnegative integer $k$ and let $G' = (V', E')$ be a subgraph of $G$. Suppose $G'$ is a $k'$-dof-graph for some nonnegative integer $k'$. Then $G'$ is a minimal $k'$-dof-graph.

**Proof.** Consider $\mathcal{M}(\tilde{G}')$, which is the matroid $\mathcal{M}(\tilde{G})$ restricted to $\tilde{E}'$. Recall that the set of bases of $\mathcal{M}(\tilde{G}')$ is the set of maximal members of $\{B \cap \tilde{E}' \mid B$ is a base of $\mathcal{M}(\tilde{G})\}$ (see, e.g., [95, Chapter 3, 3.1.15]). Since $B \cap \tilde{e} \neq \emptyset$ holds for any base $B$ of $\mathcal{M}(\tilde{G})$ and for any $e \in E$ from the minimality of $G$, $(B \cap \tilde{E}') \cap \tilde{e} \neq \emptyset$ holds for any base $B$ of $\mathcal{M}(\tilde{G})$ and for any $e \in E'$. Since any base $B'$ of $\mathcal{M}(\tilde{G}')$ can be written as $B \cap \tilde{E}'$ with some base $B$ of $\mathcal{M}(\tilde{G})$, $B' \cap \tilde{e} \neq \emptyset$ holds for any $e \in E'$. This implies that $G'$ is a minimal $k'$-dof-graph for some nonnegative integer $k'$.

In a multigraph, an edge pair is called a cut pair if the removal of these two edges disconnects the graph. By Lemmas 6.5 and 6.6, we see that any minimally body-and-hinge rigid graph $G$ has a cut pair. Using this property, we can actually show a nice combinatorial property: any 2-edge-connected minimal $k$-dof-graph contains a vertex of degree two or three. Since this is not directly used in our proof of the Molecular Conjecture, we omit the proof. Later, we shall also present a similar property in Lemma 6.16.

### 6.3.2 Rigid subgraphs

Let $G$ be a multigraph. We say that a subgraph $G'$ of $G$ is a rigid subgraph if $G'$ is a 0-dof-graph, i.e., $\tilde{G}'$ contains $D$ edge-disjoint spanning trees on the vertex set of $G'$. In this
subsubsection we prove the following three lemmas related to rigid subgraphs.

**Lemma 6.8.** Let $G = (V, E)$ be a multigraph and let $X$ be a circuit of the matroid $\mathcal{M}(\tilde{G})$. Then, $G[V(X)]$ is a rigid subgraph of $G$. More precisely, $X - e$ can be partitioned into $D$ edge-disjoint spanning trees on $V(X)$ for any $e \in X$.

**Proof.** A circuit $X$ is a minimal dependent set of $\mathcal{M}(\tilde{G})$ satisfying $|X| > D(|V(X)| - 1)$, and $X - e$ is independent in $\mathcal{M}(\tilde{G})$ for any $e \in X$, see, e.g., [95]. From $|X| > D(|V(X)| - 1)$, we have $|X - e| \geq D(|V(X)| - 1)$. On the other hand, since $X - e$ is independent, we also have $|X - e| \leq D(|V(X - e)| - 1) \leq D(|V(X)| - 1)$. As a result, $|X - e| = D(|V(X)| - 1)$ holds and hence $X - e$ can be partitioned into $D$ edge-disjoint spanning trees on $V(X)$. This implies that the graph $G[V(X)]$ induced by $V(X)$ is a 0-dof-graph and equivalently a rigid subgraph. □

**Lemma 6.9.** Let $G = (V, E)$ be a minimal $k$-dof-graph for a nonnegative integer $k$ and let $G' = (V', E')$ be a rigid subgraph of $G$. Then, the graph obtained from $G$ by contracting $E'$ is a minimal $k$-dof-graph.

**Proof.** Let $H$ be the graph obtained by contracting $E'$. By Lemma 6.5, $G'$ is connected and hence $V'$ becomes a single vertex after the contraction of $E'$. Let $v^*$ be this new vertex in $H$, that is, $H = ((V \setminus V') \cup \{v^*\}, E \setminus E')$.

Let $B_{\tilde{G}'}$ be a base of $\mathcal{M}(\tilde{G}')$. Then, we have $|B_{\tilde{G}'}| = D(|V'| - 1)$ since $G'$ is a 0-dof-graph. Also, there exists a base $B$ of $\mathcal{M}(\tilde{G})$ which contains $B_{\tilde{G}'}$ as its subset. Let $\{F_1, F_2, \ldots, F_D\}$ be a partition of $B$ into $D$ edge-disjoint forests on $V$. We claim the following:

$$F_i \cap \tilde{E}' \text{ forms a spanning tree on } V' \text{ for each } 1 \leq i \leq D. \quad (6.7)$$

To see this, notice that $B_{\tilde{G}'} \subseteq B \cap \tilde{E}'$ implies $|B \cap \tilde{E}'| \geq |B_{\tilde{G}'}| = D(|V'| - 1)$. On the other hand, from the fact that $F_i \cap \tilde{E}'$ is independent in a graphic matroid, we also have $|F_i \cap \tilde{E}'| \leq |V(F_i) \cap \tilde{E}'| \leq |V(B \cap \tilde{E}')| - 1 \leq |V'| - 1$ for each $1 \leq i \leq D$. These imply $|B \cap \tilde{E}'| = \sum_{i=1}^{D} |F_i \cap \tilde{E}'| \leq D(|V'| - 1) \leq |B \cap \tilde{E}'|$ and the equalities hold everywhere, implying $|F_i \cap \tilde{E}'| = |V'| - 1$. Thus, (6.7) holds.

Due to (6.7), after the contraction of $\tilde{E}'$, $F_i \setminus \tilde{E}'$ does not contain a cycle in $\tilde{H}$ and again forms a forest on $(V \setminus V') \cup \{v^*\}$. This implies that $\{F_1 \setminus \tilde{E}', \ldots, F_D \setminus \tilde{E}'\}$ is a partition of $B \setminus \tilde{E}'$ into $D$ edge-disjoint forests on $(V \setminus V') \cup \{v^*\}$ and hence $B \setminus \tilde{E}'$ is independent in $\mathcal{M}(\tilde{H})$. Since $|B \setminus \tilde{E}'| = |B| - |B_{\tilde{G}'}| = D(|V| - 1) - k - D(|V'| - 1) = D(|V \setminus V' \cup \{v^*\}| - 1) - k$, def($\tilde{H}$) $\leq k$ follows from (6.6).

To see def($\tilde{H}$) $\geq k$, let us consider a base $B_{\tilde{H}} \subseteq \tilde{E} \setminus \tilde{E}'$ of $\mathcal{M}(\tilde{H})$. Let $\{S_1, \ldots, S_D\}$ be a partition of $B_{\tilde{H}}$ into $D$ edge-disjoint forests on $(V \setminus V') \cup \{v^*\}$. Also, since $G'$ is a 0-dof-graph, a base $B_{\tilde{G}'}$ of $\mathcal{M}(\tilde{G}')$ can be partitioned into $D$ edge-disjoint spanning trees $\{T_1, \ldots, T_D\}$ on $V'$. Then, it is not difficult to see that $S_i \cup T_i$ forms a forest on $V$ for each $i$, and thus $B_{\tilde{H}} \cup B_{\tilde{G}'}$ is an independent set of $\mathcal{M}(\tilde{G})$. This implies $|B_{\tilde{H}} \cup B_{\tilde{G}'}| \leq D(|V| - 1) - k$. Substituting $|B_{\tilde{H}}| = D(|V \setminus V' \cup \{v^*\}| - 1) - \text{def}(\tilde{H})$ and $|B_{\tilde{G}'}| = D(|V'| - 1)$, we obtain def($\tilde{H}$) $\geq k$. 

The minimality of $H$ can be checked by the same argument. Suppose, for a contradiction, that there exists a base $B'_H$ of $\mathcal{M}(\tilde{H})$ which contains no edge of $\tilde{e}$ for some $e \in E \setminus E'$. Then, $B'_H \cup B_G$ is again a base of $\mathcal{M}(\tilde{G})$ which contains no edge of $\tilde{e}$, contradicting the minimality of the original graph $G$.

Notice that, for every circuit $X$ of $\mathcal{M}(\tilde{G})$, $V(X)$ induces a 2-edge-connected subgraph by Lemma 6.5 and Lemma 6.8. This fact leads to the following property of a multigraph that is not 2-edge-connected.

**Lemma 6.10.** Let $G = (V, E)$ be a minimal $k$-dof-graph. Let $P = \{V_1, V_2\}$ be a partition of $V$ and let $G_i = G[V_i]$ for $i = 1, 2$. Then, we have the following:

- If $d_G(P) = 1$, then $k = \text{def}(\tilde{G}_1) + \text{def}(\tilde{G}_2) + 1$.
- If $d_G(P) = 0$, then $k = \text{def}(\tilde{G}_1) + \text{def}(\tilde{G}_2) + D$.

**Proof.** Let us consider the case of $d_G(P) = 1$. We denote by $uv$ the edge connecting between $V_1$ and $V_2$ with $u \in V_1$ and $v \in V_2$. Let $E_1$ and $E_2$ be the edge sets of $G_1$ and $G_2$ and let $E_3 = \{uv\}$ and $G_3 = \{(u, v), E_3\}$. Then, $\{E_1, E_2, E_3\}$ is a partition of $E$ and moreover there exists no circuit in $\mathcal{M}(\tilde{G})$ which intersects more than one set among $\{\tilde{E}_1, \tilde{E}_2, \tilde{E}_3\}$ since any circuit induces a 2-edge-connected subgraph in $G$ by Lemma 6.5 and Lemma 6.8. Thus, we can decompose the matroid as $\mathcal{M}(\tilde{G}) = \mathcal{M}(\tilde{G}_1) \oplus \mathcal{M}(\tilde{G}_2) \oplus \mathcal{M}(\tilde{G}_3)$, where $\oplus$ denotes the direct sum of the matroids (see Section 2.2.1). Since the rank of the direct sum of matroids is the sum of the ranks of these matroids, we obtain $D(|V| - 1) - k = D(|V_1| - 1) - \text{def}(\tilde{G}_1) + D(|V_2| - 1) - \text{def}(\tilde{G}_2) + D - 1$, where we used the obvious fact that the rank of $\mathcal{M}(\tilde{G}_3)$ is equal to $D - 1$. By $|V| = |V_1| + |V_2|$, we eventually obtain the claimed relation.

The proof for the case $d_G(P) = 0$ is basically the same, and hence it is omitted.

### 6.4 Operations for Minimal $k$-dof-graphs

In this section we shall discuss two simple operations on a minimal $k$-dof-graph. One operation is the contraction of a proper rigid subgraph; $G' = (V', E')$ is called a proper rigid subgraph if it is a rigid subgraph of $G$ satisfying $1 < |V'| < |V|$. We have already seen in Lemma 6.9 that the contraction of a rigid subgraph produces a smaller minimal $k$-dof-graph. Another operation is a so-called splitting off operation, whose definition will be given in the next subsection. Our goal of this section is to show Lemma 6.18, which states that any minimal $k$-dof-graph can be always converted to a smaller minimal $k$-dof-graph or minimal $(k-1)$-dof-graph by a contraction of a proper rigid subgraph or a splitting off at a vertex of degree two. We will use this result in the proof of the Molecular Conjecture for applying the induction. Also, as a corollary, we will obtain Theorem 6.19; any minimally body-and-hinge rigid graph can be constructed by a sequence of these two simple operations, which must be an interesting result in its own right.
6.4. Operations for Minimal \( k \)-dof-graphs

6.4.1 Splitting off operation at a vertex of degree two

In this subsection, we shall examine a splitting off operation that converts a minimal \( k \)-dof-graph into a smaller graph, which is analogous to that introduced for \( 2k \)-edge-connected graphs [84]. For a vertex \( v \) of a graph \( G \), we denote by \( N_G(v) \) the set of neighbors of \( v \) (i.e., the set of vertices adjacent to \( v \) in \( G \)). A splitting off at \( v \) is an operation which removes \( v \) and then inserts new edge(s) between vertices of \( N_G(v) \). We shall consider such an operation only at a vertex \( v \) of degree two. Let \( N_G(v) = \{a, b\} \). We denote by \( G^ab \) the graph obtained from \( G \) by removing \( v \) (and the edges incident to \( v \)) and then inserting a new edge \( ab \). The operation that produces \( G^ab \) from \( G \) is called a splitting off at \( v \) (along \( ab \)). The main result of this subsection is Lemma 6.13, which claims that the splitting off does not increase the deficiency but may not preserve the minimality of the resulting graph. Before showing Lemma 6.13, let us first investigate the relation between independent sets of \( \mathcal{M}(\tilde{G}) \) and those of \( \mathcal{M}(\tilde{G}^ab) \) in the following lemmas.

Lemma 6.11. Let \( G = (V, E) \) be a \( k \)-dof-graph which has a vertex \( v \) of degree 2 with \( N_G(v) = \{a, b\} \). For any independent set \( I \) of \( \mathcal{M}(\tilde{G}) \), there exists an independent set \( I' \) of \( \mathcal{M}(\tilde{G}^ab) \) satisfying \( |I'| = |I| - D \) and \( |\tilde{a}b \cap I'| < D - 1 \).

Proof. Let \( h = |(\tilde{a}a \cup \tilde{b}b) \cap I| \). Let \( \{F_1, \ldots, F_D\} \) be a partition of \( I \) into \( D \) edge-disjoint forests on \( V \). Since \( d_G(v) = 2 \), clearly \( d_{F_i}(v) \leq 2 \) holds for each \( i = 1, \ldots, D \), (where \( d_{F_i}(v) \) denotes the number of edges of \( F_i \) incident to \( v \)). Let \( h' \) be the number of forests \( F_i \) satisfying \( d_{F_i}(v) = 2 \). Note that \( 2h' + (D - h') = h \) and \( h \leq 2(D - 1) \), which imply \( h' \leq D - 2 \). For a forest \( F_i \) satisfying \( d_{F_i}(v) = 1 \), removing the edge incident to \( v \) results in a forest on \( V - v \). For a forest \( F_i \) satisfying \( d_{F_i}(v) = 2 \), removing the edges incident to \( v \) and inserting an edge of \( ab \), we can also obtain a forest on \( V - v \). We hence convert each \( F_i \) to a forest \( F'_i \) on \( V - v \) by the above operations such that \( |F'_i| = |F_i| - 1 \) for each \( i \). Moreover, since the total number of edges of \( ab \) needed to convert \( F_i \) to \( F'_i \) is equal to \( h' \), which is less than \( |ab| = D - 1 \), \( F'_1, \ldots, F'_D \) can be taken to be edge-disjoint in \( \tilde{G}^ab \). Let \( I' = \bigcup_{i=1}^{D} F'_i \). Then, clearly, \( I' \) is an independent set of \( \mathcal{M}(\tilde{G}^ab) \) with cardinality \( |I'| = |I| - D \) as required.

The inverse operation of the splitting off at a vertex of degree two is called edge-splitting. More formally, the edge-splitting (along an edge \( ab \)) is the operation that removes an edge \( ab \) and then inserts a new vertex \( v \) with the two new edges \( va \) and \( vb \). The following lemma supplies the converse direction of Lemma 6.11.

Lemma 6.12. Let \( H = (V, E) \) be a \( k \)-dof-graph, \( ab \) be an edge of \( H \), and \( H^ab \) be the graph obtained by the edge-splitting along \( ab \). Let \( I' \) be an independent set of \( \mathcal{M}(\tilde{H}) \) with \( h' = |\tilde{a}b \cap I'| \). Then, (i) if \( h' < D - 1 \), there exists an independent set \( I \) of \( \mathcal{M}(\tilde{H}^ab) \) satisfying \( |I| = |I'| + D \) and \( |I \cap \tilde{v}b| = h' + 1 \) and (ii) otherwise there exists an independent set \( I \) of \( \mathcal{M}(\tilde{H}^ab) \) satisfying \( |I| = |I'| + D - 1 \).

Proof. Let \( \{F'_1, \ldots, F'_D\} \) be a partition of \( I' \) into \( D \) edge-disjoint forests on \( V \). Without loss of generality, we assume \((ab)i \in F'_i \) for each \( 1 \leq i \leq h' \).
Let us first consider the case of $h' < D - 1$. Consider the extension of each forest as follows: $F_i = F'_i - (ab)_i + (va)_i + (vb)_i$, for each $1 \leq i \leq h'$, $F_i = F'_i + (va)_i$, for $h' + 1 \leq i \leq D - 1$ and $F_D = F'_D + (vb)_{h'+1}$ (for $i = D$). Then, $F_1, \ldots, F_D$ are $D$ edge-disjoint forests contained in $H^v_{ab}$ and $\bigcup_{i=1}^D F_i$, denoted by $I$, is an independent set of $M(H^v_{ab})$. Since $|F_i| = |F'_i| + 1$, we have $|I| = |I'| + D$ and also $|I \cap \tilde{v}| = h' + 1$ as required.

When $h' = D - 1$, let $F_i = F'_i - (ab)_i + (va)_i + (vb)_i$, for each $1 \leq i \leq D - 1$ and $F_D = F'_D$ (for $i = D$). Then, $\bigcup_{i=1}^D F_i$ is an independent set of $M(H^v_{ab})$ with cardinality $|I'| + D - 1$. □

We are now ready to discuss the deficiency of the graph obtained by a splitting off operation.

**Lemma 6.13.** Let $G = (V, E)$ be a minimal $k$-dof-graph which has a vertex $v$ of degree 2 with $N_G(v) = \{a, b\}$. Then, (i) $G^v_{ab}$ is either a $k$-dof-graph or a minimal $(k - 1)$-dof-graph. Moreover, (ii) $G^v_{ab}$ is a $k$-dof-graph if and only if there is a base $B'$ of $M(G^v_{ab})$ satisfying $|ab \cap B'| < D - 1$.

**Proof.** Let $B$ be a base of $M(\tilde{G})$. Then, by Lemma 6.11, there exists an independent set $I'$ of $M(G^v_{ab})$ satisfying $|I'| = |B| - D$ and $|ab \cap I'| < D - 1$. Since $|I'| = |B| - D = D(|V| - 1) - k - D = D(|V \setminus \{v\}| - 1) - k$, the rank of $M(G^v_{ab})$ is at least $D(|V \setminus \{v\}| - 1) - k$. This implies

$$\text{def}(G^v_{ab}) \leq k \tag{6.8}$$

by (6.6), where the equality holds if and only if $I'$ is a base of $M(G^v_{ab})$. Thus, $\text{def}(G^v_{ab}) = k$ holds if and only if there is a base $B'$ of $M(G^v_{ab})$ with $|ab \cap B'| < D - 1$. In this case, $G^v_{ab}$ is a $k$-dof-graph.

Let us consider the case where every base $B'$ of $M(G^v_{ab})$ satisfies $|ab \cap B'| = D - 1$. In this case $G^v_{ab}$ is not a $k$-dof-graph and hence (6.8) implies

$$\text{def}(G^v_{ab}) \leq k - 1. \tag{6.9}$$

By Lemma 6.12 (ii), there exists an independent set $J$ of $M(\tilde{G})$ satisfying $|J| = |B'| + D - 1 = D(|V| - 1) - (\text{def}(G^v_{ab}) + 1)$, where $|B'| = D(|V \setminus \{v\}| - 1) - \text{def}(G^v_{ab})$. We thus obtain $k = \text{def}(\tilde{G}) \leq \text{def}(G^v_{ab}) + 1$ by (6.6). Combining it with (6.9), we eventually obtain $\text{def}(G^v_{ab}) = k - 1$. Thus, $G^v_{ab}$ is a $(k - 1)$-dof-graph.

The minimality of $G^v_{ab}$ can be checked by using Lemma 6.12(ii) again. Suppose there exists an edge $e$ such that $G^v_{ab} - e$ is still a $(k - 1)$-dof-graph. Note $e \neq ab$ since every base $B'$ of $M(G^v_{ab})$ satisfies $|ab \cap B'| = D - 1$ now. Hence, taking a base which contains no edge of $\tilde{e}$ in $M(G^v_{ab})$ and then extending it by applying Lemma 6.12(ii) we will have a base of $M(\tilde{G})$ which also contains no edge of $\tilde{e}$, contradicting the minimality of $G$. Therefore, $G^v_{ab}$ is a minimal $(k - 1)$-dof-graph. □

Applying Lemma 6.13 to the case of $k = 0$, we see that, for a minimally body-and-hinge rigid graph $G$, $G^v_{ab}$ is always body-and-hinge rigid. However, as we mentioned, a splitting off may not preserve the minimality of $G^v_{ab}$. For example, for a minimally body-and-hinge
6.4. Operations for Minimal $k$-dof-graphs

Figure 6.2: (a) An example of a minimal 0-dof-graph $G$ such that $G_{v}^{ab}$ is not a minimal 0-dof-graph for $d = 2$ and $D = 3$. Notice that $G' = G_v$ is a 0-dof-graph and hence $G_{v}^{ab}$ is not minimal. (b) An example of a minimal 0-dof-graph $G$ such that $G_{v}^{ab}$ is not minimal and also $G_v$ is not a 0-dof-graph for $d = 2$ and $D = 3$. Notice that there exist three edge-disjoint spanning trees in $G_{v}^{ab}$ that contain no edge of $e$.

rigid graph $G'$ shown in Figure 6.2(a), consider the graph $G$ obtained from $G'$ by attaching a new vertex $v$ via the two new edges $va$ and $vb$. Then, $G$ (the left one of Figure 6.2(a)) is a minimally body-and-hinge rigid graph. On the other hand, the graph obtained from $G$ by splitting off at $v$ is not minimally body-and-hinge rigid while just removing $v$ (without inserting the new edge $ab$) from $G$ produces a minimally body-and-hinge rigid graph. We call this operation as the removal of $v$ to distinguish it from the splitting off at $v$. More formally, we denote by $G_{v}$ the graph obtained by the removal of a vertex $v$ of degree two if one exists.

Lemma 6.14. Let $G=(V,E)$ be a $k$-dof-graph in which there exists a vertex $v$ of degree 2 with $N_G(v) = \{a,b\}$. Then, $\text{def}(G_{v}) \geq k$ holds. Moreover, if $\text{def}(G_{v}) = k$, then there exists a base $B$ of $M(G_{v})$ satisfying $|\{v\} \cap B| = 1$.

Proof. Consider a base $B'$ of $M(G_{v})$. Since $\tilde{G}_{v}$ is a subgraph of $G_{v}^{ab}$, $B'$ is an independent set of $M(\tilde{G}_{v})$. Also, since $G$ can be obtained from $G_{v}^{ab}$ by the edge-splitting along $ab$, Lemma 6.12(i) can be applied to $\tilde{G}_{v}$ with $h' = 0$ to derive that there exists an independent set $I$ of $M(\tilde{G})$ satisfying $|I| = |B'| + D$ and $|\{v\} \cap I| = 1$. This implies that $|I| = |B'| + D = D(|V| - 1) - \text{def}(\tilde{G}_{v})$ by $|B'| = D(|V| - 2) - \text{def}(\tilde{G}_{v})$. Thus, $k = \text{def}(\tilde{G}) \leq \text{def}(\tilde{G}_{v})$ holds by (6.6).

If $k = \text{def}(\tilde{G}) = \text{def}(\tilde{G}_{v})$ holds, then $I$ is a base of $M(\tilde{G})$ and thus a desired base exists.

We remark that there is a situation in which $G_{v}$ is not a $k$-dof-graph and also $G_{v}^{ab}$ is not a minimal $k$-dof-graph. Figure 6.2(b) shows such an example.

Remarks. The concept of the splitting off was originated by Lovász [84], where he proved that the splitting off at a vertex $v$ does not decrease the connectivity between two vertices except for $v$. Also, he proved that a graph is $2k$-edge-connected if and only if it can
be constructed from a single vertex by a sequence of two operations keeping the $2k$-edge-connectivity: one is an edge addition and the other is the inverse of a splitting off operation. Characterizing graphs having some specific property in terms of an inductive construction is an important topic. In particular it is known that a graph has $k$ edge-disjoint spanning trees if and only if it can be constructed from a single vertex by a sequence of three simple operations keeping the property [89]. We should remark that this result cannot be directly applied to $q$-strong graphs for a rational $q$ which is the case for our problem and there is no result concerning a construction of $q$-strong graphs to the best of our knowledge.

6.4.2 Minimal $k$-dof-graphs having no proper rigid subgraph

As shown in Figure 6.2, a splitting off does not preserve the minimality in general. However, if we concentrate on a graph which has no proper rigid subgraph, it can be shown that a splitting off preserves the minimality. We hence concentrate on graphs having no proper rigid subgraph throughout this subsection. Let us first show properties of such graphs in Lemmas 6.15 and 6.16.

**Lemma 6.15.** Let $G = (V,E)$ be a minimal $k$-dof-graph which contains no proper rigid subgraph. Then, the followings hold.

(i) If $k = 0$, then $(D - 1)|E| < D(|V| - 1) + D - 1$.

(ii) If $k > 0$, then $\tilde{E}$ is the base of $\mathcal{M}(\tilde{G})$ and hence $(D - 1)|E| = D(|V| - 1) - k$.

**Proof.** (i) Let us consider the case of $k = 0$. Let $e$ be an arbitrary edge of $E$, and let $h^*$ be the minimum value of $|\tilde{e} \cap B|$ taken over all bases $B$ of $\mathcal{M}(\tilde{G})$. Also, let $B^*$ be a base of $\mathcal{M}(\tilde{G})$ satisfying $|\tilde{e} \cap B^*| = h^*$. Notice $h^* \geq 1$ due to the minimality of $G$. We shall show the following fact:

$$\tilde{E} \setminus \tilde{e} \subset B^*. \quad (6.10)$$

Suppose, for a contradiction, that an edge $f_i \in \tilde{E} \setminus \tilde{e}$ is not contained in $B^*$. We consider the fundamental circuit $X$ within $B^* + f_i$. Then, $G[V(X)]$ is a rigid subgraph by Lemma 6.8. Since there exists no proper rigid subgraph in $G$, $V(X) = V$ must hold. Moreover, $X \cap \tilde{e} \neq \emptyset$ also holds since otherwise there exist $D$ edge-disjoint spanning trees on $V$ which contain no edge of $\tilde{e}$ by Lemma 6.8, contradicting the minimality of $G$. Therefore, there exists the base $B = B^* + f_i - e_j \in \mathcal{M}(\tilde{G})$ satisfying $|B \cap \tilde{e}| < |B^* \cap \tilde{e}| = h^*$, where $e_j \in X \cap \tilde{e}$. This contradicts the choice of $B^*$, and hence (6.10) follows.

By (6.10) and $|B^*| = D(|V| - 1)$, we obtain that the total number of edges in $\tilde{G}$ is equal to $|B^*| + (|\tilde{e}| - h^*) = D(|V| - 1) + (D - 1 - h^*)$, which is less than $D(|V| - 1) + D - 1$ by $h^* \geq 1$.

(ii) Let us consider the case of $k > 0$. We shall show the following fact, which is analogous to (6.10):

$$\tilde{E} \text{ is independent in } \mathcal{M}(\tilde{G}). \quad (6.11)$$

Suppose for a contradiction that $\tilde{E}$ is dependent. Then, there exists a base $B$ of $\mathcal{M}(\tilde{G})$ that does not contain an edge $f_i \in \tilde{E}$. Consider again the fundamental circuit $X$ within $B + f_i$. 
6.4. Operations for Minimal $k$-dof-graphs

By Lemma 6.8, $G[V(X)]$ is a rigid subgraph. Since $G$ contains no proper rigid subgraph from the lemma assumption, $V(X) = V$ must hold. Moreover, $V(X) = V$ implies that $G$ contains $D$ edge-disjoint spanning trees on $V$, which consist of edges of $X$. This in turn implies $k = 0$, contracting $k > 0$. Thus, (6.11) follows.

The following lemma shows the existence of small degree vertices.

**Lemma 6.16.** Let $G = (V, E)$ be a 2-edge-connected minimal $k$-dof-graph which contains no proper rigid subgraph. Then, either $G$ is a cycle of at most $d$ vertices or it contains a chain $v_0v_1 \ldots v_d$ of length $d$ such that $v_iv_{i+1} \in E$ for $0 \leq i \leq d - 1$ and $d_G(v_i) = 2$ for $1 \leq i \leq d - 1$.

**Proof.** Let us denote the average degree of the vertices of $G$ by $d_{\text{avg}}$. Lemma 6.15 implies $(D - 1)|E| < D(|V| - 1) + D - 1 < D|V|$. Hence, we have

$$d_{\text{avg}} = \frac{2|E|}{|V|} < \frac{2D}{D - 1} = 2 + \frac{2}{D - 1} \leq 3,$$

(6.12)

where the last inequality follows from $D \geq 3$. This implies that $G$ has a vertex of degree two.

If $G$ is a cycle, then the statement clearly holds. (If it consists of more than $d$ vertices, then the latter property holds.) Hence let us consider the case where $G$ contains a vertex of degree more than two. For a nonnegative integer $i$, let $X_i = \{v \in V \mid d_G(v) = i\}$. Note that, since $G$ is 2-edge-connected, $X_0 = \emptyset$ and $X_1 = \emptyset$ hold. We say that a chain $u_0u_1 \ldots u_j$ is maximal if $d_G(u_0) > 2$, $d_G(u_j) > 2$ and $d_G(u_i) = 2$ for all $1 \leq i \leq j - 1$. Let $C$ be the collection of all maximal chains in $G$. Note that $C$ is nonempty (because $G$ is not a cycle), and each vertex of degree two belongs to exactly one maximal chain. Suppose, for a contradiction, that the length of each maximal chain is at most $d - 1$. Then, each maximal chain contains at most $d - 2$ vertices of degree two and hence we have

$$|X_2| \leq (d - 2)|C|. \quad (6.13)$$

For a maximal chain $u_1u_2 \ldots u_j$, we call the edges $u_1u_2$ and $u_{j-1}u_j$ the end edges of the chain. Then the set of all end edges of the maximal chains in $C$ is a subset of the edges incident to the vertices of $\bigcup_{i \geq 3} X_i$. Hence we have

$$2|C| \leq \sum_{i \geq 3} i|X_i|. \quad (6.14)$$

Combining (6.13) and (6.14), we obtain

$$2|X_2| \leq \sum_{i \geq 3} i(d - 2)|X_i|. \quad (6.15)$$

Summing up this inequality and (the twice of) $|V| = \sum_{i \geq 2} |X_i|$, we further obtain

$$\sum_{i \geq 3}(i(d - 2) + 2)|X_i| \geq 2|V|. \quad (6.15)$$

It is not difficult to see that the following inequality holds.

$$(D - 1)(i - 2) \geq i(d - 2) + 2 \quad \text{for all } i \geq 3.$$
Hence, by (6.15), we have
\[ \sum_{i \geq 3} (D - 1)(i - 2)|X_i| \geq 2|V|. \] (6.16)

As a result,
\[ d_{\text{avg}} = \frac{\sum_{i \geq 2} i|X_i|}{|V|} = 2 + \frac{\sum_{i \geq 3} (i - 2)|X_i|}{|V|} \]
\[ = 2 + \frac{\sum_{i \geq 3} (D - 1)(i - 2)|X_i|}{(D - 1)|V|} \]
\[ \geq 2 + \frac{2}{D - 1} \]
\[ > d_{\text{avg}}, \]
which is a contradiction.

Let us start to investigate the deficiencies of graphs obtained by the operations defined in Section 6.4.1, assuming that \( G \) contains no proper rigid subgraph.

Lemma 6.17. Let \( G = (V, E) \) be a minimal \( k \)-dof-graph with \( |V| \geq 3 \) which contains no proper rigid subgraph. Let \( v \) be a vertex of degree two. Then, \( \text{def}(G_v) > k. \)

Proof. Note that \( G_v \) is a proper subgraph of \( G \). Since there exists no proper rigid subgraph in \( G \), \( G_v \) is not a 0-dof-graph. This proves the statement for \( k = 0. \)

When \( k > 0, \) \( \bar{E} \) is the base of \( \mathcal{M}(\bar{G}) \) from Lemma 6.15(ii). Since \( \bar{E} \) is the unique base of \( \mathcal{M}(\bar{G}) \) and \( |v\bar{b} \cap \bar{E}| \neq 1 \) holds, Lemma 6.14 implies \( \text{def}(G_v) > k. \)

Lemma 6.18. Let \( G = (V, E) \) be a minimal \( k \)-dof-graph which contains no proper rigid subgraph. Then, for any vertex \( v \) of degree two with \( N_G(v) = \{a, b\} \), the followings hold:

(i) If \( k = 0 \), then \( G_v^{ab} \) is a minimal 0-dof-graph.

(ii) If \( k > 0 \), then \( G_v^{ab} \) is a minimal \( (k-1) \)-dof-graph.

Proof. We remark that \( G_v \) is not a \( k \)-dof-graph by Lemma 6.17. Also, by Lemma 6.13, \( G_v^{ab} \) is either a \( k \)-dof-graph or a minimal \( (k-1) \)-dof-graph.

Let us show (i). In the case of \( k = 0 \), \( G_v^{ab} \) is clearly a 0-dof-graph and thus we only need to show the minimality of \( G_v^{ab} \). To see this, we claim the following:

For any circuit \( X \) of the matroid \( \mathcal{M}(\tilde{G}_v) \), \( X \cap \tilde{a} \tilde{b} \neq \emptyset \) holds. (6.17)

Suppose \( X \cap \tilde{a} \tilde{b} = \emptyset \). Then, \( X \) is a subset of \( \bar{E} \). Hence, \( X \) is a circuit of \( \mathcal{M}(\bar{G}) \) with \( v \notin V(X) \). Lemma 6.8 says that \( G[V(X)] \) is a proper rigid subgraph of \( G \), which contradicts the lemma assumption. Thus (6.17) follows.

In order to show (i), suppose that \( G_v^{ab} \) is not a minimal 0-dof-graph. Then, there exists an edge \( e \) such that \( G_v^{ab} - e \) is still a 0-dof-graph. (Note \( e \neq ab \) since \( G_v \) is not a 0-dof-graph by Lemma 6.17.) Also, there exists a base \( B_1 \) of \( \mathcal{M}(\tilde{G}_v) \) with \( B_1 \cap \tilde{c} = \emptyset \). Let \( h = |B_1 \cap \tilde{a} \tilde{b}| \),
and let us denote the edges of \( B_1 \cap \tilde{ab} \) by \((ab)_1, \ldots, (ab)_h\). Also we denote the edges of \( \tilde{e} \) by \( e_1, \ldots, e_{D-1} \) as usual. We repeatedly perform the following process for \( i = 1, \ldots, h \): Insert an edge \( e_i \in \tilde{e} \) into \( B_i \) and let \( X_i \) be the fundamental circuit of \( B_i + e_i \) (in \( \mathcal{M}(\tilde{G}_{v}^{ab}) \)). By (6.17), \( X_i \cap \tilde{ab} \neq \emptyset \) holds and hence we can obtain a new base \( B_{i+1} \) by removing \((ab)_i \) in \( X_i \cap \tilde{ab} \). (Namely, we obtain the base \( B_{i+1} = B_i + e_i - (ab)_i \) of \( \mathcal{M}(G_{v}^{ab}) \).) Repeating this process, we eventually obtain the base \( B_{h+1} \) of \( \mathcal{M}(G_{v}^{ab}) \), which contains no edge of \( \tilde{ab} \). Note that \( B_{h+1} \) is a base of \( \mathcal{M}(\tilde{G}_{v}) \) as well as a base of \( \mathcal{M}(G_{v}^{ab}) \) with cardinality \(|B_{h+1}| = |B_1| = D(|V| \setminus \{v\} | - 1)\). Therefore, \( \text{def}(\tilde{G}_{v}) = 0 \) holds by (6.6). This contradicts that \( G_{v} \) is not a 0-dof-graph (by Lemma 6.17) and thus (i) follows.

Next let us show (ii). If \( G_{v}^{ab} \) is not a \( k \)-dof-graph, then the statement follows because \( G_{v}^{ab} \) is either a \( k \)-dof-graph or a minimal \((k - 1)\)-dof-graph by Lemma 6.13. Suppose, for a contradiction, that \( G_{v}^{ab} \) is a \( k \)-dof-graph. By Lemma 6.13 there exists a base \( B' \) of \( \mathcal{M}(G_{v}^{ab}) \) satisfying \(|\tilde{ab} \cap B'| < D - 1\). Without loss of generality, we assume \((ab)_1 \notin B'\). Consider the fundamental circuit \( Y \) of \( B' + (ab)_1 \) and let \( G' = G[V(Y)] \). Then, by Lemma 6.8, \( G' \) is a 0-dof-graph on \( V(Y) \). Since \( G_{v}^{ab} \) is a \( k \)-dof-graph with \( k > 0 \), \( G' \) must be a proper subgraph of \( G_{v}^{ab} \), i.e., \( V(Y) \) is a proper subset of \( V \setminus \{v\} \). Let \( I = Y - (ab)_1 \). Then, Lemma 6.8 also says that \( I \) can be partitioned into \( D \) edge-disjoint spanning trees on \( V(Y) \). Hence, \( I \) is an independent set of \( \mathcal{M}(\tilde{G}_{v}) \) with \(|I \cap \tilde{ab}| < D - 1\) due to \((ab)_1 \notin I \). We apply the edge-splitting operation in \( G' \) along \( ab \). Lemma 6.12(i) implies that the resulting graph contains an independent set with cardinality \(|I| + D\), which is equal to \( D(|V(Y)| - 1) + D = D(|V(Y) \cup \{v\}| - 1) \). Hence the resulting graph is a 0-dof-graph. Moreover, this 0-dof-graph is a proper subgraph of \( G \) since \( V(Y) \) is a proper subset of \( V \setminus \{v\} \). Therefore, \( G \) contains a proper rigid subgraph, which contradicts the lemma assumption.

\(\square\)

6.4.3 Inductive constructions

Combining the results obtained so far, it is not difficult to prove the following construction of minimally body-and-hinge rigid graphs.

**Theorem 6.19.** Let \( G \) be a minimally body-and-hinge rigid graph with \(|V| \geq 2\). Then, there exists a sequence \( G = G_1, G_2, \ldots, G_m \) of minimally body-and-hinge rigid graphs such that

- \( G_m \) is a graph consisting of two vertices \( \{u, v\} \) and two parallel edges connecting \( u \) and \( v \), and
- \( G_{i+1} \) is obtained from \( G_i \) by either the splitting off at a vertex of degree 2 or the contraction of a proper rigid subgraph for each \( i = 1, \ldots, m - 1 \).

**Proof.** By Lemma 6.5, any minimally body-and-hinge rigid graph is a 2-edge-connected 0-dof-graph. Hence, if \( G \) contains no proper rigid subgraph, then \( G \) has a vertex of degree two by Lemma 6.16. Combining this fact with Lemma 6.18, either (i) \( G \) contains a proper rigid subgraph \( G' = (V', E') \) or (ii) there exists a vertex \( v \) of degree 2 with \( N_G(v) = \{a, b\} \) such that \( G_{v}^{ab} \) is a minimal 0-dof-graph, that is, a minimally body-and-hinge rigid graph. Recall that, if (i) holds, then the graph obtained by the contraction of \( E' \) is again a minimally body-and-hinge rigid graph by Lemma 6.9. Hence, in either case, we can obtain a new minimally
body-and-hinge rigid graph \( G_2 = (V_2, E_2) \) with \( 2 \leq |V_2| < |V| \). By inductively repeating this process, we eventually obtain a minimally body-and-hinge rigid graph \( G_m \) which consists of two vertices and two parallel edges between them. This inductive process shows a desired sequence of minimally body-and-hinge rigid graphs.

\[ \square \]

### 6.5 Infinitesimally Rigid Panel-and-hinge Realizations

Recall the notation given in Section 3.2: \( C(p(e)) \) denotes a \((d - 1)\)-extensor associated with the \((d - 2)\)-dimensional affine space \( p(e) \), \( \{r_1(p(e)), \ldots, r_{D-1}(p(e))\} \) denotes a basis of the orthogonal complement of the 1-dimensional vector space spanned by \( C(p(e)) \), and \( r(p(e)) \) denotes the \((D - 1) \times D\)-matrix whose \( i \)-th row vector is \( r_i(p(e)) \). Note that \( \text{rank } r(p(e)) = D - 1 \) holds.

Recall that the rigidity matrix of \((G, p)\) is written as

\[
R(G, p) = e = uv \begin{pmatrix}
\cdots & u & \cdots & v & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & 0 \cdots & r(p(e)) & \cdots & 0 \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & 0 \cdots & -r(p(e)) & \cdots & 0 \cdots \\
\end{pmatrix}
\quad (6.18)
\]

where consecutive \( D - 1 \) rows are associated with an edge \( e \in E \) and consecutive \( D \) columns are associated with a vertex \( v \in V \). Let us denote the \((D - 1) \times D|V|\)-submatrix associated with \( e \in E \) by \( R(G, p; e) \) for each \( e \in E \), i.e.,

\[
R(G, p; e) = (\cdots \cdots r_u(p(e)) \cdots 0 \cdots -r_v(p(e)) \cdots 0 \cdots). \quad (6.19)
\]

We remark \( \text{rank } R(G, p; e) = D - 1 \) since \( \text{rank } r(p(e)) = D - 1 \). Also, we consider the one-to-one correspondence between \( e_i \in \bar{e} \) and the \( i \)-th row of \( R(G, p; e) \), which is denoted by \( R(G, p; e_i) \). Namely, for \( e = uv \in E \) and \( 1 \leq i \leq D - 1 \), it is a \( D|V| \)-dimensional vector described as

\[
R(G, p; e_i) = (\cdots 0 \cdots r_i(p(e)) \cdots 0 \cdots -r_i(p(e)) \cdots 0 \cdots).
\]

Any vector in \( \mathbb{R}^{D|V|} \) can be regarded as a composition of \( |V| \) vectors in \( \mathbb{R}^D \) each of which is associated with a vertex \( v \in V \) in a natural way.

Similarly, let us denote by \( R(G, p; v) \) the \((D - 1)|E| \times D\)-submatrix of \( R(G, p) \) induced by the consecutive \( D \) columns associated with \( v \). For \( F \subseteq E \) and \( X \subseteq V \), \( R(G, p; F, X) \) denotes the submatrix of \( R(G, p) \) induced by the rows of \( R(G, p; e) \) for \( e \in F \) and the columns of \( R(G, p; v) \) for \( v \in X \).

We need the following technical lemma.

\begin{lemma}
Let \( (G, p) \) be a body-and-hinge framework in \( \mathbb{R}^d \) for a multigraph \( G = (V, E) \). Then, for any vertex \( v \in V \), \( \text{rank } R(G, p; E, V \setminus \{v\}) = \text{rank } R(G, p) \) holds, i.e., the rank of the rigidity matrix is invariant under the removal of the consecutive \( D \) columns associated with \( v \).
\end{lemma}
6.5. Infinitesimally Rigid Panel-and-hinge Realizations

Proof. For $1 \leq i \leq D$, let $b_i$ be the vector in $\mathbb{R}^{D|V|}$ such that the $i$-th coordinate of the consecutive $D$ coordinates associated with $v$ is equal to 1 and the other entries are all 0. Let $R'$ be the matrix obtained from $R(G, p)$ by adding $b_i$ as new rows for all $i$. Then, appropriate fundamental row operations changes $R'$ to the following form:

$$
R' = \begin{pmatrix}
  v & V \setminus \{v\} \\
  I & 0 \\
  R(G, p) & 0
\end{pmatrix}
\xrightarrow{\text{Row operations}}
\begin{pmatrix}
  v & V \setminus \{v\} \\
  I & 0 \\
  R(G, p; E, V \setminus \{v\}) & 0
\end{pmatrix}
$$

where $I$ denotes the $D \times D$ identity matrix. This implies $\text{rank } R' = \text{rank } R(G, p; E, V \setminus \{v\}) + D$. Hence, the statement is true if and only if $\text{rank } R' = \text{rank } R(G, p) + D$ holds or, equivalently, no vector spanned by $\{b_1, b_2, \ldots, b_D\}$ is contained in the row space of $R(G, p)$. Suppose, for a contradiction, that a nonzero vector $b'$ spanned by $\{b_1, b_2, \ldots, b_D\}$ is contained in the row space of $R(G, p)$. Let us denote it by $b' = \lambda_1 b_1 + \cdots + \lambda_D b_D$ with $\lambda_i \in \mathbb{R}, 1 \leq i \leq D$ and, without loss of generality, we assume $\lambda_1 \neq 0$. Let $S_1^*$ be the vector in $\mathbb{R}^{D|V|}$ whose first coordinate of the consecutive $D$ coordinates associated with each vertex is equal to 1 and the other entries are all 0. Then it is not difficult to see that $S_1^*$ is in the orthogonal complement of the row space of $R(G, p)$. However, we have $b' \cdot S_1^* = \lambda_1 \neq 0$, which contradicts that $b'$ is contained in the row space of $R(G, p)$. \qed

6.5.1 Generic nonparallel panel-and-hinge realizations

Before providing a proof of the Molecular Conjecture, we need to mention the generic property of panel-and-hinge realizations for a simple graph introduced by Jackson and Jordán [62]. For a panel-and-hinge realization $(G, p)$, let $\Pi_{G,p}(v)$ denote the panel associated with $v \in V$, that is, a $(d-1)$-affine subspace containing all of the hinges $p(e)$ of the edges $e$ incident to $v$. For a simple graph $G$ (i.e., no parallel edges exist in $G$), $(G, p)$ is called a nonparallel panel-and-hinge realization if $\Pi_{G,p}(u)$ and $\Pi_{G,p}(v)$ are not parallel for any distinct $u, v \in V$; $\Pi_{G,p}(u)$ and $\Pi_{G,p}(v)$ are said to be nonparallel if $\Pi_{G,p}(u) \cap \Pi_{G,p}(v)$ is a $(d-2)$-affine subspace. As Jackson and Jordán mentioned in [62, Section 7], each entry of the rigidity matrix $R(G, p)$ of a nonparallel panel-and-hinge realization $(G, p)$ can be described in terms of the coefficients appearing in the equations expressing $\Pi_{G,p}(v)$ for $v \in V$, and hence each minor of the rigidity matrix is a polynomial of these coefficients. The rigidity of nonparallel panel-and-hinge realizations thus has a generic property.

Let us look into the details. For a simple graph $G = (V, E)$, consider a mapping $c : V \to \mathbb{R}^d$ such that $c(u)$ and $c(v)$ are linearly independent for each $u, v \in V$ with $u \neq v$. Then, the $(d-1)$-affine subspace (i.e. panel) associated with $v \in V$ with respect to $c$ is defined as $\Pi(v) = \{x \in \mathbb{R}^d \mid x \cdot c(v) = 1\}$. Since $c(u)$ and $c(v)$ are linearly independent, $\Pi(u) \cap \Pi(v)$ is a $(d-2)$-dimensional affine space. Hence, the mapping $c$ induces a mapping $p$ on $E$, that is, $p(wv) = \Pi(u) \cap \Pi(v)$ for each $wv \in E$, and $(G, p)$ is a nonparallel panel-and-hinge framework of $G$. Conversely, given a nonparallel $(G, p)$, $p$ induces the unique mapping $c : V \to \mathbb{R}^d$ such that $\Pi_{G,p}(v) = \{x \in \mathbb{R}^d \mid x \cdot c(v) = 1\}$ for each $v \in V$ (provided that no panel passes through the origin and no vertex of degree one exists).
We say that \((G, \mathbf{p})\) is generic if the set of coordinates of \(c(v)\) for all \(v \in V\) is algebraically independent over the rational field.\(^1\) We note that almost all nonparallel panel-and-hinge realizations are generic. Since each entry of \(R(G, \mathbf{p})\) is a polynomial of the coordinates of \(c(v)\) for \(v \in V\), the rank of \(R(G, \mathbf{p})\) takes the maximum value over all nonparallel panel-and-hinge realizations of \(G\) if \((G, \mathbf{p})\) is generic.

It is known that, even though \((G, \mathbf{p})\) has some parallel panels, we can perturb them so that the resulting realization becomes nonparallel without decreasing the rank of the rigidity matrix (see [61] or [62, Lemma 7.1]). The following lemma states a special case of this result, but let us provide a proof for completeness.

**Lemma 6.21.** Let \(G\) be a simple graph and \((G, \mathbf{p})\) be a panel-and-hinge realization of \(G\). Suppose there exists a pair \((a, b) \in V \times V\) with \(a \neq b\) satisfying \(\Pi_{G, \mathbf{p}}(a) = \Pi_{G, \mathbf{p}}(b)\) and that \(\Pi_{G, \mathbf{p}}(u)\) and \(\Pi_{G, \mathbf{p}}(v)\) are nonparallel for every pair \((u, v) \in V \times V\) with \(u \neq v\) except for \((a, b)\). Then, there is a nonparallel panel-and-hinge realization \((G, \mathbf{p}')\) satisfying \(\text{rank } R(G, \mathbf{p}') \geq \text{rank } R(G, \mathbf{p})\).

**Proof.** We shall only consider the case of \(ab \in E\). The case of \(ab \notin E\) can be handled similarly. Note that \(ab\) is unique because \(G\) is simple.

Since the rank of the rigidity matrix is invariant under an isometric transformation of the whole framework, we can assume \(\Pi_{G, \mathbf{p}}(a) = \Pi_{G, \mathbf{p}}(b) = \{x = (x_1, \ldots, x_d) \in \mathbb{R}^d \mid x_d = 0\}\) and \(\mathbf{p}(ab) = \{x \in \mathbb{R}^d \mid x_{d-1} = 0, x_d = 0\}\). We shall rotate \(\Pi_{G, \mathbf{p}}(a)\) continuously around \(\mathbf{p}(ab)\).

To indicate the rotation, let us introduce a parameter \(t \in \mathbb{R}\) and define \(\Pi'(a) = \{x \in \mathbb{R}^d \mid tx_{d-1} + x_d = 0\}\). Note that \(\Pi'(a) = \Pi_{G, \mathbf{p}}(a)\) and \(\mathbf{p}(ab) \subseteq \Pi'(a) \cap \Pi_{G, \mathbf{p}}(b)\) for any \(t \in \mathbb{R}\).

Since \(\Pi'(a)\) and \(\Pi_{G, \mathbf{p}}(v)\) are nonparallel for any \(v \in V \setminus \{a, b\}\), from the lemma assumption, there exists a small \(\varepsilon > 0\) such that \(\Pi'(a)\) and \(\Pi_{G, \mathbf{p}}(v)\) are nonparallel within \(-\varepsilon < t < \varepsilon\).

Hence, the following mapping \(\mathbf{p}'\) on \(E\) is thus well defined within \(-\varepsilon < t < \varepsilon\):

\[
\mathbf{p}'(e) = \begin{cases} 
\mathbf{p}(e) & \text{if } e \in E \setminus \delta_G(a) \cup \{ab\}, \\
\Pi'(a) \cap \Pi_{G, \mathbf{p}}(v) & \text{if } e = av \in \delta_G(a) \setminus \{ab\}.
\end{cases}
\]

Notice that \(\mathbf{p}'(e) = \mathbf{p}(e)\) for all \(e \in E\) and \((G, \mathbf{p}') = (G, \mathbf{p})\) holds. Also, \((G, \mathbf{p}')\) is a nonparallel panel-and-hinge realization for any \(0 < t < \varepsilon\). Since each \(\mathbf{p}'(e), e \in E\) moves continuously with respect to \(t\), each minor of \(R(G, \mathbf{p}')\) can be described as a continuous function of \(t\) within \(-\varepsilon < t < \varepsilon\). This implies that there exists a small \(\varepsilon'\) with \(0 < \varepsilon' \leq \varepsilon\) such that \((G, \mathbf{p}')\) is a nonparallel panel-and-hinge realization satisfying \(\text{rank } R(G, \mathbf{p}') \geq \text{rank } R(G, \mathbf{p})\) for any \(0 < t < \varepsilon'\). \(\square\)

Let us introduce the concept of nondegenerate frameworks for multigraphs. Let \(G\) be a multigraph and let \((G, \mathbf{p})\) be a panel-and-hinge realization of \(G\). We say that \((G, \mathbf{p})\) is nondegenerate if \(\Pi_{G, \mathbf{p}}(u) \cap \Pi_{G, \mathbf{p}}(v) \neq \emptyset\) for any \(u, v \in V\), i.e., either \(\Pi_{G, \mathbf{p}}(u)\) and \(\Pi_{G, \mathbf{p}}(v)\)

\(^1\)The definition in terms of algebraic independence produces a smaller class of frameworks than that of conventional generic frameworks (in terms of the maximality of rigidity matrices). We just use this definition to make our proof simpler.
are nonparallel or $\Pi_{G,p}(u) = \Pi_{G,p}(v)$. Extending the discussions so far, it is not difficult to see the following fact.

**Lemma 6.22.** Let $G$ be a multigraph and $(G,p)$ be a panel-and-hinge realization of $G$. Then, there exists a nondegenerate panel-and-hinge realization $(G,p')$ satisfying $\text{rank } R(G,p') \geq \text{rank } R(G,p)$.

**Proof.** Consider the partition $\mathcal{V}$ of $V$ defined in such a way that $u$ and $v$ belong to the same subset of vertices if and only if $\Pi_{G,p}(u) = \Pi_{G,p}(v)$ holds. For $V_i \in \mathcal{V}$, we define the panel $\Pi_{G,p}(V_i)$ as $\Pi_{G,p}(V_i) = \Pi_{G,p}(v)$ for a vertex $v \in V_i$. Also, for $V_i \in \mathcal{V}$, let $E(V_i) = \{uv \in E \mid u,v \in V_i\}$. Then, it is not difficult to see that each entry of the rigidity matrix $R(G,p)$ can be described in terms of the coefficients of the equations representing $\Pi_{G,p}(V_i)$ for $V_i \in \mathcal{V}$ and the coefficients of the equations representing $p(uv)$ of $uv \in E(V_i)$ for $V_i \in \mathcal{V}$.

Suppose $(G,p)$ is degenerate, i.e., $\Pi_{G,p}(u)$ and $\Pi_{G,p}(v)$ are parallel with $\Pi_{G,p}(v_i) \neq \Pi_{G,p}(v_j)$ for some $u \in V_i \in \mathcal{V}$ and $v \in V_j \in \mathcal{V}$. Then, from the definition of $\mathcal{V}$, $V_i \neq V_j$ holds. We shall continuously rotate the panel $\Pi_{G,p}(V_i)$ and the hinges $p(e)$ of $e \in E(V_i)$ preserving their incidences. Since any minor of $R(G,p)$ can be written as a polynomial of the coefficients representing $\Pi_{G,p}(V_i)$ and the hinges $p(e)$ of $e \in E(V_i)$ (when fixing the other panels), the rank of the rigidity matrix does not decrease if the continuous rotation is small enough. Repeating this process for any pair of parallel panels, we eventually obtain a desired nondegenerate framework. \qed

### 6.5.2 Molecular Conjecture

We now start to show our main result. We first claim the rigidity of graphs consisting of a small number of vertices since it will be used several times (including the base case of the induction).

**Lemma 6.23.** Let $G = (V,E)$ be the graph consisting of two vertices $\{u,v\}$ and two parallel edges $\{e,f\}$ between $u$ and $v$. Then, $G$ can be realized as an infinitesimally rigid panel-and-hinge framework $(G,p)$ such that $\Pi_{G,p}(u) = \Pi_{G,p}(v)$. In particular, $\text{rank } R(G,p;\{e,f\},v) = D$ holds if $p(e) \neq p(f)$.

**Proof.** Let $\text{span}(C(p(e)))$ and $\text{span}(C(p(f)))$ be the vector spaces spanned by $(d-1)$-extensors $C(p(e))$ and $C(p(f))$ associated with $p(e)$ and $p(f)$, respectively. A $(d-1)$-extensor of a $(d-2)$-dimensional affine space is uniquely determined up to a scalar multiplication. This implies that $\text{span}(\text{span}(C(p(e)))) = \text{span}(C(p(f)))$ if and only if $p(e) = p(f)$. Therefore, if $p(e) \neq p(f)$, the orthogonal complements of $\text{span}(C(p(e)))$ and $\text{span}(C(p(f)))$, that is, the row spaces of $R(p(e))$ and $R(p(f))$ are distinct. We hence have $\text{rank } R(G,p;\{e,f\},v) = D$ by $\text{rank } R(G,p;\{e,f\},v) = \text{rank } R(p(e)) = D - 1$. Since we can realize $G$ as a framework $(G,p)$ such that $\Pi_{G,p}(u) = \Pi_{G,p}(v)$ and $p(e) \neq p(f)$, the statement follows. \qed

If $G$ is a cycle graph, its realization can be easily analyzed directly from the definition of infinitesimal motions. The detailed calculation can be seen in [31, Proposition 3.4] or
Lemma 6.24 ([31, 130]). Let $G = (V, E)$ be a cycle graph with $3 \leq |V| \leq D$. Then, $G$ can be realized as an infinitesimally rigid nonparallel panel-and-hinge framework $(G, p)$.

Let us state the main theorem of this paper.

Theorem 6.25. Let $G = (V, E)$ be a minimal $k$-dof-graph with $|V| \geq 2$ for some nonnegative integer $k$. Then, there exists a (nonparallel, if $G$ is simple) panel-and-hinge realization $(G, p)$ in $\mathbb{R}^d$ satisfying $\text{rank}\, R(G, p) = D(|V| - 1) - k$.

Since the proof is quite long, let us first write up a corollary which follows from Theorem 6.25. The following theorem proves the Molecular Conjecture in a strong sense combined with Proposition 6.4.

Theorem 6.26. Let $G = (V, E)$ be a multigraph. Then, $G$ can be realized as a panel-and-hinge framework $(G, p)$ in $\mathbb{R}^d$ which satisfies $\text{rank}\, R(G, p) = D(|V| - 1) - \text{def}(\tilde{G})$.

Proof. Let $k = \text{def}(\tilde{G})$. By Proposition 6.4, we have $\text{rank}\, R(G, p) \leq D(|V| - 1) - k$ for any realization $(G, p)$ of $G$. When $G$ is not a minimal $k$-dof-graph, we can remove some edges from $G$ keeping the deficiency of $\tilde{G}$ so that the resulting graph becomes a minimal $k$-dof-graph. Let $G' = (V, E')$ be the obtained minimal $k$-dof-graph. (If $G$ is a minimal $k$-dof-graph, then let $G' = G$.) By Theorem 6.25, there is a panel-and-hinge realization $(G', q)$ satisfying $R(G', q) = D(|V| - 1) - k$. Moreover, by Lemma 6.22, we may assume that $(G', q)$ is nondegenerate. This means that, for any $u, v \in V$, $\Pi_{G', q}(u) \cap \Pi_{G', q}(v)$ is either a $(d - 2)$-affine subspace or a $(d - 1)$-affine subspace in $\mathbb{R}^d$. Define a mapping $p$ on $E$ such that $p(uv) = q(uv)$ for $uv \in E'$ and otherwise $p(uv)$ is a $(d - 2)$-affine subspace contained in $\Pi_{G', q}(u) \cap \Pi_{G', q}(v)$. It is obvious that $(G, p)$ is a panel-and-hinge realization of $G$ and moreover $\text{rank}\, R(G, p) \geq \text{rank}\, R(G', q) = D(|V| - 1) - k$. We thus obtain a panel-and-hinge realization satisfying $\text{rank}\, R(G, p) = D(|V| - 1) - k$. \qed

As we mentioned in Section 6.1, in 3-dimensional space the projective dual of nonparallel panel-and-hinge frameworks are “hinge-concurrent” body-and-hinge frameworks, which are also called molecular frameworks [60, 130] because they are used to study the flexibility of molecules. Since the infinitesimal rigidity is invariant under the duality [31, Section 3.6] (see also [126] for more detailed descriptions on the related topic), it follows from Theorem 6.26 that a simple graph $G = (V, E)$ can be realized as a molecular framework $(G, p)$ which satisfies $\text{rank}\, R(G, p) = D(|V| - 1) - \text{def}(\tilde{G})$.

Let us consider bar-and-joint rigidity in 3-dimensional space. For a graph $G$ of the minimum degree at least two, Whiteley showed in [131] that $G$ can be realized as a rigid molecular framework if and only if $G^2$ can be realized as an infinitesimally rigid bar-and-joint framework. Jackson and Jordán proved in [57, 60] that the Molecular Conjecture (Conjecture 6.1) is equivalent to the following statement:
Corollary 6.27. Let $G = (V, E)$ be a graph with minimum degree at least two. Then $r(G^2) = 3|V| - 6 - \text{def}(G)$, where $r$ denotes the rank function of the 3-dimensional generic rigidity matroid.

Further combinatorial results on 3-dimensional bar-and-joint frameworks of square graphs can be found in [57, 60].

### 6.6 Proof of Theorem 6.25

The proof is done by induction on $|V|$. Let us consider the base case $|V| = 2$. Let $V = \{u, v\}$. By Lemma 6.6, any minimal $k$-dof-graph is not 3-edge-connected and hence we have three possible cases: (i) $E$ is empty, (ii) $E$ consists of a single edge $e$ connecting $u$ and $v$, and (iii) $E$ consists of two parallel edges $\{e, f\}$ between $u$ and $v$. The cases (i) and (ii) are trivial since any realization satisfies the statement. The case (iii) has been treated in Lemma 6.23.

Let us consider $G$ with $|V| \geq 3$. We shall split the proof into three cases:

- Section 6.6.1 deals with the case where $G$ is not 2-edge-connected.
- Section 6.6.2 deals with the case where $G$ contains a proper rigid subgraph.
- Section 6.6.3 deals with the case where $G$ is 2-edge-connected and does not contain any proper rigid subgraph.

In each case, we will assume the following induction hypothesis on $|V|:$

\[
\text{For any minimal } k_H\text{-dof-graph } H = (V_H, E_H) \text{ for some nonnegative integer } k_H \text{ with } |V_H| < |V|, \text{ there is a (nonparallel, if } G_H \text{ is simple) panel-and-hinge realization } (G_H, p_H) \text{ in } d\text{-dimensional space with rank } R(G_H, p_H) = D(|V_H| - 1) - k_H. \tag{6.20}
\]

#### 6.6.1 The case where $G$ is not 2-edge-connected

This case is handled rather easily but illustrates a basic strategy of the subsequent arguments.

**Lemma 6.28.** Let $G = (V, E)$ be a minimal $k$-dof-graph which is not 2-edge-connected. Suppose that (6.20) holds. Then, there is a (nonparallel, if $G$ is simple) panel-and-hinge realization $(G, p)$ in $\mathbb{R}^d$ satisfying $\text{rank } R(G, p) = D(|V| - 1) - k.$

**Proof.** Let us consider the case where $G$ is connected. (The case where $G$ is disconnected graphs is omitted.) Since $G$ has a cut edge $uv$, $G$ can be partitioned into two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $u \in V_1$, $v \in V_2$, $V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$ and $\delta_G(\{V_1, V_2\}) = \{uv\}$. Let $k_1$ and $k_2$ be the deficiencies of $\widetilde{G}_1$ and $\widetilde{G}_2$, respectively. Then, $k = k_1 + k_2 + 1$ holds by Lemma 6.10 and also $G_i$ is a minimal $k_i$-dof-graph for each $i = 1, 2$ by Lemma 6.7. By (6.20), we have a (nonparallel, if $G_i$ is simple) panel-and-hinge realization $(G_i, p_i)$ satisfying $\text{rank } R(G_i, p_i) = D(|V_i| - 1) - k_i$ for each $i = 1, 2$. Since the choices of $p_1$ and $p_2$ are independent of each other and also since the rank of the rigidity matrix is invariant under an isometric transformation of the whole framework, we can take $p_1$ and $p_2$
such that \( \Pi_{G_1,p_1}(v_1) \) and \( \Pi_{G_2,p_2}(v_2) \) are nonparallel for any pair of \( v_1 \in V_1 \) and \( v_2 \in V_2 \). In particular, \( \Pi_{G_1,p_1}(u) \cap \Pi_{G_2,p_2}(v) \) is a \((d-2)\)-affine subspace in \( \mathbb{R}^d \). Define a mapping \( p \) as

\[
p(e) = \begin{cases} p_1(e) & \text{if } e \in E_1 \\ p_2(e) & \text{if } e \in E_2 \\ \Pi_{G_1,p_1}(u) \cap \Pi_{G_2,p_2}(v) & \text{if } e = uv. \end{cases}
\]

Then, \((G, p)\) is a (nonparallel, if \( G \) is simple) panel-and-hinge realization of \( G \). By \( \delta_G(\{V_1, V_2\}) = \{uv\} \), the rigidity matrix \( R(G, p) \) can be described as

\[
R(G, p) = \begin{pmatrix}
E_1 & 0 \\
0 & E_2
\end{pmatrix}
\begin{pmatrix}
V_1 & V_2 \\
0 & R(G, p; uv)
\end{pmatrix}
\begin{pmatrix}
V_1 \setminus \{u\} & V_2 \setminus \{v\} \\
0 & R(G_2, p_2)
\end{pmatrix}
\begin{pmatrix}
R(G_1, p_1; E_1, V_1 \setminus \{u\}) & R(G_1, p_1; E_1, u) \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
r(p(uv)) & -r(p(uv)) \\
0 & 0
\end{pmatrix}.
\]

Note that \( \text{rank } R(G_1, p_1; E_1, V_1 \setminus \{u\}) = \text{rank } R(G_1, p) = D(|V_1| - 1) - k_1 \) holds by Lemma 6.20. Note also \( \text{rank } r(p(uv)) = D - 1 \) from the definition. Hence, by \( k = k_1 + k_2 + 1 \) and \( |V| = |V_1| + |V_2| \), we obtain \( \text{rank } R(G, p) \geq \text{rank } R(G_1, p_1; E_1, V_1 \setminus \{u\}) + \text{rank } r(p(uv)) + \text{rank } R(G_2, p_2) = D(|V_1| - 1) - k_1 + (D - 1) + D(|V_2| - 1) - k_2 = D(|V| - 1) - (k_1 + k_2 + 1) = D(|V| - 1) - k. \)

### 6.6.2 Case where \( G \) contains a proper rigid subgraph

Let us first describe a proof sketch. Let \( G' = (V', E') \) be a proper rigid subgraph in a minimal \( k\)-dof-graph \( G = (V, E) \). Note that \( G' \) is a minimal 0-dof-graph by Lemma 6.7 with \( 1 < |V'| < |V| \). Let \( G/E' = ((V \setminus V') \cup \{v^s\}, E \setminus E') \) be the graph obtained from \( G \) by contracting the edges of \( E' \), where \( v^s \) is the new vertex obtained by the contraction. Then, by Lemma 6.9, \( G/E' \) is a minimal \( k\)-dof-graph with \( |(V \setminus V') \cup \{v^s\}| < |V| \). Therefore, by the inductive hypothesis (i.e. (6.20)), there exist panel-and-hinge realizations \((G', p_1)\) and \((G/E', p_2)\) satisfying \( \text{rank } R(G', p_1) = D(|V'| - 1) \) and \( \text{rank } R(G/E', p_2) = D(|V\setminus V'| \cup \{v^s\}) - 1) - k \). Based on these realizations, we shall construct a realization of \( G \). Intuitively, we shall replace the body associated with \( v^s \) in \((G/E', p_2)\) by \((G', p_1)\), by regarding \((G', p_1)\) as a rigid body in \( \mathbb{R}^d \) and show that the rank of the resulting framework becomes \( \text{rank } R(G', p_1) + \text{rank } R(G/E', p_2) \), which is equal to \( D(|V| - 1) - k \). Because of the lack of the genericity for multigraphs, we will need three subcases. Lemma 6.29 deals with the case where \( G \) is not a simple graph. Lemma 6.30 deals with the case where \( G \) is simple and there exists a proper rigid subgraph \( G' = (V', E') \) such that \( G/E' \) is also simple. Lemma 6.32 deals with the rest of the cases.

**Lemma 6.29.** Let \( G = (V, E) \) be a minimal \( k\)-dof-graph with \( |V| \geq 3 \). Suppose that \( G \) is not simple and also that (6.20) holds. Then, there is a panel-and-hinge realization \((G, p)\) in \( \mathbb{R}^d \) satisfying \( \text{rank } R(G, p) = D(|V| - 1) - k \).
Let $e$ and $f$ be multiple edges connecting $a$ and $b$ for some $a, b \in V$. Then, notice that the graph $G[\{e, f\}]$ edge-induced by $\{e, f\}$ is a proper rigid subgraph in $G$ since $\tilde{e} \cup \tilde{f}$ contains $D$ edge-disjoint spanning trees on $\{a, b\}$. Hence we can assume $G' = G[\{e, f\}] = (V' = \{a, b\}, E' = \{e, f\})$. By Lemma 6.23, there exists a panel-and-hinge realization $(G', p_1)$ of $G'$ such that $\text{rank}(R(G', p_1)) = D$ and $\Pi_{G', p_1}(a) = \Pi_{G', p_1}(b)$. Also, by (6.20), there exists a panel-and-hinge realization $(G/E', p_2)$ satisfying $\text{rank}(R(G/E', p_2)) = D(|V| - 2) - k$. Since the choices of $p_1$ and $p_2$ are independent of each other, we can take $p_1$ and $p_2$ in such a way that $\Pi_{G/E', p_2}(v^*) = \Pi_{G', p_1}(a) = \Pi_{G', p_1}(b)$. Define a mapping on $E$ as

$$p(e) = \begin{cases} p_1(e) & \text{if } e \in E' \ (\{e, f\}) \\ p_2(e) & \text{if } e \in E \setminus E'. \end{cases} \quad (6.21)$$

Intuitively, $(G, p)$ is a panel-and-hinge realization of $G$ obtained from $(G/E', p_2)$ by identifying the panels of $a$ and $b$ with that of $v^*$, and in fact it can be easily verified from the definition that $(G, p)$ is a panel-and-hinge realization satisfying $\Pi_{G, p}(u) = \Pi_{G/E', p_2}(u)$ for each $u \in V \setminus \{a, b\}$ and $\Pi_{G, p}(u) = \Pi_{G/E', p_2}(v^*)$ for each $u \in \{a, b\}$.

Let us take a look at the rigidity matrix $R(G, p)$:

$$R(G, p) = \begin{pmatrix} V' \\ E \setminus E' \end{pmatrix} \begin{pmatrix} \text{rank}(R(G', p_1)) & 0 \\ 0 & \text{rank}(R(G, p; E \setminus E', V \setminus V')) \end{pmatrix} \quad (6.22)$$

Since $p(e) = p_2(e)$ for every $e \in E \setminus E'$ by (6.21), we can take\(^2\) the entries of $R(G, p; E \setminus E', V \setminus V')$ to be

$$R(G, p; E \setminus E', V \setminus V') = R(G/E', p_2; E \setminus E', V \setminus V'). \quad (6.23)$$

We also remark that $R(G/E', p_2; E \setminus E', V \setminus V')$ is the matrix obtained from $R(G/E', p_2)$ by deleting the $D$ consecutive columns associated with $v^*$. Hence, by (6.23) and Lemma 6.20, we obtain

$$\text{rank}(R(G, p; E \setminus E', V \setminus V')) = \text{rank}(R(G/E', p_2; E \setminus E', V \setminus V')) = \text{rank}(R(G/E', p_2)). \quad (6.24)$$

As a result, by (6.22) and (6.24), we obtain

$$\text{rank}(R(G, p)) \geq \text{rank}(R(G', p_1)) + \text{rank}(R(G, p; E \setminus E', V \setminus V')) = \text{rank}(R(G', p_1)) + \text{rank}(R(G/E', p_2)) = D + D(|V| - 2) - k = D(|V| - 1) - k.$$
Lemma 6.30. Let $G = (V, E)$ be a minimal $k$-dof-graph with $|V| \geq 3$. Suppose that $G$ is simple and contains a proper rigid subgraph $G' = (V', E')$ such that $G/E'$ is simple. Also suppose that (6.20) holds. Then, there exists a nonparallel panel-and-hinge realization $(G, p)$ satisfying $\text{rank } R(G, p) = D(|V| - 1) - k$.

Proof. The story of the proof is basically the same as that of Lemma 6.29 although we need a slightly different mapping $p$. By Lemma 6.9 and (6.20), there exist nonparallel panel-and-hinge realizations $(G', p_1)$ and $(G/E', p_2)$ satisfying $\text{rank } R(G', p_1) = D(|V| - 1)$ and $\text{rank } R(G/E', p_2) = D(|V' \cup \{v\}| - 1) - k$. From the definition of generic nonparallel panel-and-hinge realizations of simple graphs discussed in Section 6.5.1, $p_1$ and $p_2$ can be taken in such a way that the set of all coefficients appearing in the equations expressing the hyperplanes $\Pi_{G', p_1}(v_1)$ for all $v_1 \in V'$ and $\Pi_{G/E', p_2}(v_2)$ for all $v_2 \in V \setminus V'$ is algebraically independent over $\mathbb{Q}$. We define a mapping $p$ on $E$ as follows:

$$p(e) = \begin{cases} p_1(e) & \text{if } e \in E' \\ p_2(e) & \text{if } e \in E \setminus (E' \cup \delta_G(V')) \\ \Pi_{G/E', p_2}(u) \cap \Pi_{G', p_1}(v) & \text{if } e = uv \in \delta_G(V') \text{ with } u \in V \setminus V', v \in V'. \end{cases}$$                      (6.25)

Then, $(G, p)$ is a nonparallel panel-and-hinge realization of $G$ since all $p(e)$ for $e \in \delta_G(v_1)$ are contained in $\Pi_{G', p_1}(v_1)$ for each $v_1 \in V'$ and all $p(e)$ for $e \in \delta_G(v_2)$ are contained in $\Pi_{G/E', p_2}(v_2)$ for each $v_2 \in V \setminus V'$. Consider the rigidity matrix $R(G, p)$, which can be described in the same way as (6.22). We shall derive $\text{rank } R(G, p; E \setminus E', V \setminus V') = \text{rank } R(G/E', p_2)$ as was done in the proof of the previous lemma. This will prove $\text{rank } R(G, p) = D(|V| - 1) - k$ as before. To obtain $\text{rank } R(G, p; E \setminus E', V \setminus V') = \text{rank } R(G/E', p_2)$, we shall compare $p_2$ with the restriction $p_{E \setminus E'}$ of $p$ on $E \setminus E'$, that is,

$$p_{E \setminus E'}(e) = \begin{cases} p_2(e) & \text{if } e \in E \setminus (E' \cup \delta_G(V')) \\ \Pi_{G/E', p_2}(u) \cap \Pi_{G', p_1}(v) & \text{if } e = uv \in \delta_G(V') \text{ with } u \in V \setminus V', v \in V'. \end{cases}$$                      (6.26)

The body-and-hinge framework $(G/E', p_{E \setminus E'})$ is not a panel-and-hinge realization in general; all the hinges $p_{E \setminus E'}(e)$ of $e \in \delta_{G/E'}(u)$ are contained in the panel $\Pi_{G/E', p_2}(u)$ for each $u \in V \setminus V'$ while the hinges $p_{E \setminus E'}(e)$ of $e \in \delta_{G/E'}(v^*)$ may not be on a panel. Intuitively, $(G/E', p_{E \setminus E'})$ is a body-and-hinge realization of $G/E'$ such that every $v \in V \setminus V'$ is realized as a panel $\Pi_{G/E', p_2}(v)$ while $v^*$ may be realized as a $d$-dimensional body.

Claim 6.31. $\text{rank } R(G/E', p_{E \setminus E'}) \geq \text{rank } R(G/E', p_2)$ holds.

Proof. As mentioned in [62] and also in Section 6.5.1, each entry of the rigidity matrix of a nonparallel realization can be written in terms of the coefficients appearing in the equations representing the panels associated with $v \in V$. Although $(G/E', p_{E \setminus E'})$ is not a panel-and-hinge realization, due to the definition (6.26), each entry of $R(G/E', p_{E \setminus E'})$ is also described in terms of the coefficients appearing in the equations expressing $\Pi_{G', p_1}(v_1)$ for $v_1 \in V'$ and $\Pi_{G/E', p_2}(v_2)$ for $v_2 \in V \setminus V'$. Notice that the collection of all the nonparallel panel-and-hinge realizations of $G/E'$ is a subset of the collection of all possible realizations of $G/E'$.
Let us take a vertex-inclusionwise maximal proper rigid subgraph. We shall consider a different approach from those of the previous lemmas, based on the removal of a minimal dof-graph and equivalently a rigid subgraph of \( V \). Since \( G \) is simple. Then, \((6.20)\) holds. Then, there exists a nonparallel panel-and-hinge realization \( G \) satisfying \( R(G, p) = D(|V| - 1) - k \).

**Proof.** We shall consider a different approach from those of the previous lemmas, based on the following fact.

**Claim 6.33.** Suppose that there exists no proper rigid subgraph \( G' = (V', E') \) such that \( G/E' \) is simple. Then, \( G \) is a 0-dof-graph and there exists a vertex \( v \) of degree two such that \( G_v \) is a minimal 0-dof-graph, where \( G_v \) is the graph obtained from \( G \) by the removal of \( v \).

**Proof.** Let us take a vertex-inclusionwise maximal proper rigid subgraph \( G' = (V', E') \) of \( G \). Note that \( G' \) is a minimal 0-dof-graph by Lemma 6.7. Since \( G/E' \) is not simple, there exist a vertex \( v \in V \setminus V' \) and two edges, denoted by \( e \) and \( f \), which are incident to \( v \) and some other vertices of \( V' \). Let \( G'' = (V' \cup \{v\}, E' \cup \{e, f\}) \). Then, \( G' \) is the graph obtained from \( G'' \) by the removal of \( v \). By Lemma 6.14, we obtain \( \def(G'') \leq \def(G') = 0 \). Hence \( G'' \) is also a 0-dof-graph and equivalently a rigid subgraph of \( G \). Since \( G' \) is taken as a vertex-inclusionwise maximal proper rigid subgraph of \( G \), \( G'' \) cannot be a proper rigid subgraph and consequently \( V = V' \cup \{v\} \) holds. This implies that \( G'' \) contains \( D \) edge-disjoint spanning trees on \( V \) and \( G \) is a 0-dof-graph. Also, the minimality of \( G \) implies \( G = G'' \). Since \( v \) is a vertex of degree two in \( G'' = G \), the removal of \( v \) from \( G \) results in the minimal 0-dof-graph \( G' \).
Let \( v \) be a vertex of degree two whose removal results in a minimal 0-dof-graph \( G_v \) as shown in Claim 6.33. Let \( N_G(v) = \{ a, b \} \). Note that \( G_v \) is simple since \( G \) is simple. Hence, by (6.20), there exists a nonparallel panel-and-hinge realization \( (G_v, q) \) satisfying rank \( R(G_v, q) = D(|V \setminus \{v\}| - 1) \). We define a mapping \( p \) on \( E \) as follows:

\[
p(e) = \begin{cases} 
q(e) & \text{if } e \in E \setminus \{va, vb\} \\
\Pi^a \cap \Pi_{G_v,q}(a) & \text{if } e = va \\
\Pi^b \cap \Pi_{G_v,q}(b) & \text{if } e = vb,
\end{cases}
\]

where \( \Pi^a \) is a \((d - 1)\)-affine subspace that is not parallel to \( \Pi_{G_v,q}(u) \) for \( u \in V \setminus \{v\} \) and does not contain \( \Pi_{G_v,q}(a) \cap \Pi_{G_v,q}(b) \) (so that \( \Pi^a \cap \Pi_{G_v,q}(a) \neq \Pi^a \cap \Pi_{G_v,q}(b) \)). Clearly, \((G, p)\) is a nonparallel panel-and-hinge realization. Let us take a look at the rigidity matrix \( R(G, p) \):

\[
R(G, p) = \begin{pmatrix}
\begin{array}{c}
\begin{pmatrix}
R(G, p; va, v) & * \\
R(G, p; vb, v) & *
\end{pmatrix}
\end{array}

\end{pmatrix}
\begin{array}{c}
\begin{pmatrix}
va \\
vb
\end{pmatrix}

\end{array}
\begin{array}{c}
\begin{pmatrix}
V \\setminus \{v\}
\end{pmatrix}

\end{array}
\begin{array}{c}
\begin{pmatrix}
0 \\
R(G, p; E \setminus \{va, vb\}, V \setminus \{v\})
\end{pmatrix}

\end{array}
\end{pmatrix}
\begin{array}{c}
\begin{pmatrix}
v
\end{pmatrix}

\end{array}
\begin{array}{c}
\begin{pmatrix}
va, vb
\end{pmatrix}

\end{array}
\begin{array}{c}
\begin{pmatrix}
E \setminus \{va, vb\}
\end{pmatrix}

\end{array}
\begin{array}{c}
\begin{pmatrix}
V \setminus \{v\}
\end{pmatrix}

\end{array}
\end{pmatrix}
\tag{6.29}
\]

Since \( G_v = (V \setminus \{v\}, E \setminus \{va, vb\}) \) and also \( p(e) = q(e) \) holds for any \( e \in E \setminus \{va, vb\} \), we can take the entries of the rigidity matrix in such a way that \( R(G, p; E \setminus \{va, vb\}, V \setminus \{v\}) = R(G_v, q) \). Also, notice that the top-left \( 2(D - 1) \times D \)-submatrix, i.e. \( R(G, p; \{va, vb\}, v) \), has rank equal to \( D \) by Lemma 6.23. Thus, by (6.29), we obtain rank \( R(G, p) \geq \text{rank } R(G, p; \{va, vb\}, v) + \text{rank } R(G, p; E \setminus \{va, vb\}, V \setminus \{v\}) = D + \text{rank } R(G_v, q) = D + D(|V \setminus \{v\}| - 1) = D(|V| - 1) \).

This completes the case where \( G \) contains a proper rigid subgraph.

6.6.3 The case where \( G \) is 2-edge-connected and contains no proper rigid subgraph

Let us consider the remaining case for proving Theorem 6.25; \( G \) is 2-edge-connected and has no proper rigid subgraph. We shall further split this case into two subcases depending on whether \( k > 0 \) or \( k = 0 \). The following Lemma 6.34 deals with the case of \( k > 0 \).

Let us first show the following two easy observations, which contribute to the claim that \( G_v^{ab} \) is always simple.

Lemma 6.34. Let \( G = (V, E) \) be a 2-edge-connected minimal \( k \)-dof-graph with \(|V| \geq 3 \) for some nonnegative integer \( k \). Suppose also that \( G \) contains no proper rigid subgraph. Then, the following holds:

(i) If \(|V| = 3\), then \( k = 0 \) and there is a nonparallel panel-and-hinge realization \( (G, p) \) satisfying rank \( R(G, p) = D(|V| - 1) \).

(ii) If \(|V| \geq 4\), then \( G_v^{ab} \) is a simple graph for any vertex \( v \) of degree two, where \( N_G(v) = \{a, b\} \).
Proof. We remark that $G$ is simple since multiple edges induce a proper rigid subgraph. When $|V| = 3$, $G$ is a triangle because $G$ is simple and 2-edge-connected. Hence, by Lemma 6.24, there is a nonparallel panel-and-hinge framework $(G, p)$ satisfying $\text{rank } R(G, p) = D(|V| - 1)$.

Let us consider (ii). If $G_{va}^{ab}$ is not simple, then $G_{va}^{ab}$ contains two parallel edges between $a$ and $b$ because the original graph $G$ is simple. This implies $ab \in E$. Therefore $G$ contains a triangle $G[\{va, vb, ab\}]$ as its subgraph. Since a triangle is a 0-dof-graph, $G$ contains a proper rigid subgraph, contradicting the lemma assumption.

**Lemma 6.35.** Let $G = (V, E)$ be a 2-edge-connected minimal $k$-dof-graph with $|V| \geq 3$ for some integer $k$ with $k > 0$. Suppose that there exists no proper rigid subgraph in $G$ and that (6.20) holds. Then, there is a nonparallel panel-and-hinge realization $(G, p)$ in $\mathbb{R}^d$ satisfying $\text{rank } R(G, p) = D(|V| - 1) - k$.

**Proof.** By Lemma 6.16, $G$ has a vertex $v$ of degree two with $N_G(v) = \{a, b\}$. Since $k > 0$ and there is no proper rigid subgraph in $G$, Lemma 6.18 implies that $G_{va}^{ab}$ is a minimal $(k - 1)$-dof-graph. Also, by Lemma 6.34, we can assume that $G_{va}^{ab}$ is simple. Hence, by (6.20), there exists a generic nonparallel panel-and-hinge realization $(G_{va}^{ab}, q)$ in $\mathbb{R}^d$, which satisfies

$$\text{rank } R(G_{va}^{ab}, q) = D(|V| \setminus \{v\} - 1) - (k - 1).$$

(6.30)

Let $E_v = E \setminus \{va, vb\}$. Define a mapping $p_1$ on $E$ as follows:

$$p_1(e) = \begin{cases} q(e) & \text{if } e \in E_v = E \setminus \{va, vb\} \\ L & \text{if } e = va \\ q(ab) & \text{if } e = vb, \end{cases}$$

(6.31)

where $L$ is a $(d - 2)$-affine subspace contained in $\Pi_{G_{va}^{ab}, q}(a)$. In Figure 6.3, we illustrate this realization. We need to prove the following fact.

**Claim 6.36.** $(G, p_1)$ is a panel-and-hinge realization of $G$ for any choice of $(d - 2)$-affine subspace $L$ contained in $\Pi_{G_{va}^{ab}, q}(a)$.

**Proof.** From the definition of $p_1$ and from the fact that $(G_{va}^{ab}, q)$ is a panel-and-hinge realization, every hinge $p_1(e)$ for $e = uv \in E \setminus \{va, vb\}$ is appropriately contained in the panels $\Pi_{G_{va}^{ab}, q}(u)$ and $\Pi_{G_{va}^{ab}, q}(v)$. This implies that, for every $u \in V \setminus \{v, a, b\}$, all the hinges $p_1(e)$ for $e \in \delta_G(u)$ are contained in $\Pi_{G_{va}^{ab}, q}(u)$.
Notice that $p_1(vb) = q(ab) \subset \Pi_{G_0^v, q}(b)$ and hence all the hinges $p_1(e)$ for $e \in \delta_G(b)$ are contained in $\Pi_{G_0^v, q}(b)$. Similarly, $p_1(va) = L \subset \Pi_{G_0^v, q}(a)$ implies that all the hinges $p_1(e)$ for $e \in \delta_G(a)$ are contained in $\Pi_{G_0^v, q}(a)$.

Finally, as for the two hinges $p_1(va)$ and $p_1(vb)$ for $\delta_G(v) = \{va, vb\}$, $p_1(vb) = q(ab) \subset \Pi_{G_0^v, q}(a)$ and $p_1(va) = L \subset \Pi_{G_0^v, q}(a)$ hold. Hence, all the hinges $p_1(e)$ for $e \in \delta_G(v)$ are contained in $\Pi_{G_0^v, q}(a)$. Thus $(G, p_1)$ is a panel-and-hinge realization.

Although $\Pi_{G, p_1}(v)$ and $\Pi_{G, p_1}(a)$ are parallel in $(G, p_1)$, at the end of the proof we will convert $(G, p_1)$ to a nonparallel panel-and-hinge realization by slightly rotating $\Pi_{G, p_1}(v)$ without decreasing the rank of the rigidity matrix. The followings are an important observation provided by the configuration $p_1$ of (6.31): From $p_1(e) = q(e)$ for every $e \in E_v$,

$$R(G, p_1; E_v, V \setminus \{v\}) = R(G_{v}^{ab}, q; E_v, V \setminus \{v\}),$$

(i.e., the part of the framework $(G, p_1)$ which is not related to $\{va, vb, ab\}$ is exactly the same as that of $(G_{v}^{ab}, q)$). Let us take a look at $R(G, p_1)$:

$$R(G, p_1) = \begin{bmatrix}
  v & a & b & V \setminus \{v, a, b\} \\
v & -r(p_1(va)) & r(p_1(va)) & 0 & 0 \\
v & 0 & r(p_1(vb)) & -r(p_1(vb)) & 0 \\
E_v & R(G, p_1; E_v, V \setminus \{v\}) & 
\end{bmatrix}.$$  \hspace{1cm} (6.33)

We shall perform fundamental column operations on $R(G, p_1)$ which add the $j$-th column of $R(G, p_1; v)$ to that of $R(G, p_1; a)$ for each $1 \leq j \leq D$ one by one, i.e., $R(G, p_1)$ is changed to

$$R(G, p_1) = \begin{bmatrix}
v & a & b & V \setminus \{v, a, b\} \\
v & -r(p_1(va)) & r(p_1(va)) & 0 & 0 \\
v & 0 & r(p_1(vb)) & -r(p_1(vb)) & 0 \\
E_v & R(G, p_1; E_v, V \setminus \{v\}) & 
\end{bmatrix}.$$  \hspace{1cm} (6.34)

Substituting $p_1(vb) = q(ab)$ and (6.32) into (6.34), $R(G, p_1)$ can be expressed by

$$R(G, p_1) = \begin{bmatrix}
v & a & b & V \setminus \{v, a, b\} \\
v & -r(p_1(va)) & r(p_1(va)) & 0 & 0 \\
v & 0 & r(q(ab)) & -r(q(ab)) & 0 \\
E_v & R(G_{v}^{ab}, q; E_v, V \setminus \{v\}) & 
\end{bmatrix}.$$  \hspace{1cm} (6.35)

where we used the fact that $R(G_{v}^{ab}, q)$ is expressed by

$$R(G_{v}^{ab}, q) = \begin{bmatrix}
a & b & V \setminus \{v, a, b\} \\
E_v & -r(q(ab)) & r(q(ab)) & 0 \\
E_v & R(G_{v}^{ab}, q; E_v, V \setminus \{v\}) & 
\end{bmatrix}.$$  \hspace{1cm} (6.36)
Therefore, from (6.30) and (6.35), we can show that \( R(G, p_1) \) has the desired rank as follows:

\[
\text{rank } R(G, p_1) \geq \text{rank } r(p_1(v_a)) + \text{rank } R(G^a_{v_b}, q)
\]
\[
= D - 1 + D(|V \setminus \{v\}| - 1) - (k - 1)
\]
\[
= D(|V| - 1) - k.
\]

As we have remarked, \( (G, p_1) \) can be converted to a nonparallel realization without decreasing the rank of the rigidity matrix by Lemma 6.21. Thus, the statement follows.

The remaining case is the most difficult one, where \( G \) is a 2-edge-connected minimal 0-dof-graph and contains no proper rigid subgraph. Since the proof is quite complicated, we shall first show the 3-dimensional case, which must be most important. We then deal with the general dimensional case.

3-dimensional case \( (D = 6) \)

**Lemma 6.37.** Let \( G = (V, E) \) be a 2-edge-connected minimal 0-dof-graph with \( |V| \geq 3 \). Suppose that there exists no proper rigid subgraph in \( G \) and that (6.20) holds. Then, there is a nonparallel panel-and-hinge realization \( (G, p) \) in 3-dimensional space satisfying \( \text{rank } R(G, p) = D(|V| - 1) \).

**Proof.** Let us first describe the proof outline. Since \( d = 3 \) and no proper rigid subgraph exists in \( G \), Lemma 6.16 implies that there exist two vertices of degree two which are adjacent with each other. Let \( v \) and \( a \) be two such vertices and let \( N_G(v) = \{a, b\} \) for some \( b \in V \) and \( N_G(a) = \{v, c\} \) for some \( c \in V \) (see Figure 6.4(a)). Lemma 6.18 implies that both \( G^a_{v_b} \) and \( G^v_{a_c} \) are minimal 0-dof-graphs. Here \( G^v_{a_c} \) is the graph obtained by performing the splitting off operation at \( a \) along \( vc \). By (6.20), there exist generic nonparallel panel-and-hinge realizations \( (G^a_{v_b}, q) \) and \( (G^v_{a_c}, q_\varphi) \) where \( q_\varphi \) will be defined later in (6.50).

We shall first consider the realization \( (G, p_1) \) of \( G \) based on \( (G^a_{v_b}, q) \) which is exactly the same as the one we considered in Lemma 6.35 (see Figure 6.4(d)). Contrary to the case of Lemma 6.35, we need to construct two more panel-and-hinge realizations \( (G, p_2) \) and \( (G, p_3) \). These two realizations are constructed based on the graphs \( G^a_{v_b} \) and \( G^v_{a_c} \), respectively. The realization \( (G, p_2) \) can be constructed in the manner symmetric to \( (G, p_1) \) by changing the role of \( a \) and \( b \) as shown in Figure 6.4(e). Although \( (G, p_1) \) and \( (G, p_2) \) are not nonparallel, we will convert them to nonparallel panel-and-hinge realizations by slightly rotating the panel of \( v \) without decreasing the rank of the rigidity matrix by using Lemma 6.21.

It can be shown that, in the 2-dimensional case, at least one of \( R(G, p_1) \) and \( R(G, p_2) \) attains the desired rank \( D(|V| - 1) \) as we will see in the general dimensional case. However, for \( d = 3 \), there is a case where ranks of \( R(G, p_1) \) and \( R(G, p_2) \) may be less than the desired value. To complete the proof, we will introduce one more realization \( (G, p_3) \). It is not difficult to see that \( G^v_{a_c} \) is isomorphic to \( G^a_{v_b} \) (see Figure 6.4(b)(f)) and hence there is the realization \( (G^v_{a_c}, q_\varphi) \) that represents the same panel-and-hinge framework as \( (G^a_{v_b}, q) \) by regarding the
A Proof of the Molecular Conjecture

Figure 6.4: The graphs and the realizations given in the proof of Lemma 6.37 around $v$, where the panels associated with the vertices other than $v, a, b, c$ are omitted. (a)$G$, (b)$G_{ab}^v$, (c)$G_{ab}^v, q$, (d)$(G, p_1)$, (e)$(G, p_2)$, (f)$G_{ac}^v$, (g)$G_{ac}^v(q_p)$ and (h)$G, p_3$).

panel of $a$ in $(G_{ab}^v, q)$ as that of $v$ in $(G_{ac}^v, q_v)$ as shown in Figure 6.4(g). We shall then construct the realization $(G, p_3)$ based on $(G_{ac}^v, q_v)$ as shown in Figure 6.4(h) (see (6.50) and (6.51) for the definitions of $q_v$ and $p_3$). Again, we can convert $(G, p_3)$ to a nonparallel realization without decreasing rank of the rigidity matrix. Since $(G_{ac}^v, q_v)$ and $(G_{ab}^v, q)$ are isomorphic, they have the same rigidity matrix, and hence we may think that the new hinge $p_3(ac)$ is inserted into $R(G_{ab}^v, q)$. Since the position of $p_3(ac)$ is different from that of $p_1(va)$ or $p_2(vb)$, we expect that $p_3(ac)$ would eliminate a nontrivial infinitesimal motion appearing in $(G, p_1)$ and $(G, p_2)$. We actually show that at least one of $(G, p_1)$, $(G, p_2)$ and $(G, p_3)$ attains the desired rank by computing ranks of the rigidity matrices of these realizations with certain choices of $p_1(va), p_2(vb), p_3(ac)$.

Let us start the rigorous proof. As remarked in the proof of the previous lemma, $G$ is a simple graph (since otherwise multiple edges induce a proper rigid subgraph). Also, by Lemma 6.34, we can assume that $|V| \geq 4$ and $G_{ab}^v$ is simple in the subsequent proof. Hence, by (6.20), there exists a generic nonparallel panel-and-hinge realization $(G_{ab}^v, q)$ in $\mathbb{R}^d$, which satisfies

$$\text{rank } R(G_{ab}^v, q) = D(|V \setminus \{v\}| - 1). \tag{6.37}$$

Let $E_v = E \setminus \{va, vb\}$. Let us recall that the mapping $p_1$ on $E$ was defined by (6.31). We symmetrically define a mapping $p_2$ on $E$ as follows:

$$p_2(e) = \begin{cases} 
q(e) & \text{if } e \in E_v \\
q(ab) & \text{if } e = va \\
L' & \text{if } e = vb,
\end{cases} \tag{6.38}$$

where $L'$ is a $(d - 2)$-affine subspace contained in $\Pi_{G_{ab}^v(q)}(b)$. Then, the argument symmetric to Claim 6.36 implies that $(G, p_2)$ is a panel-and-hinge realization of $G$. The frameworks $(G, p_1)$ and $(G, p_2)$ are depicted in Figure 6.4(d) and (e), respectively.
6.6. Proof of Theorem 6.25

Putting aside for a while the discussion concerning how \( R(G, \mathbf{p}_3) \) is represented, we shall first investigate when \( R(G, \mathbf{p}_1) \) or \( R(G, \mathbf{p}_2) \) takes the desired rank (for some choice of \( L \subset \Pi_{G_v^*, \mathbf{q}(a)} \) or \( L' \subset \Pi_{G_v^*, \mathbf{q}(b)} \)). \( R(G, \mathbf{p}_1) \) was described by (6.35), and \( R(G, \mathbf{p}_2) \) is given as follows.

\[
R(G, \mathbf{p}_2) = \begin{pmatrix}
\begin{array}{ccc}
\ & \ & \\
\ & v & \ \\
& a & b & V \setminus \{v, a, b\}
\end{array}
\end{pmatrix}
\begin{pmatrix}
v \\
a \\
b \\
E_v
\end{pmatrix}
\begin{pmatrix}
\begin{array}{c}
\begin{pmatrix}
r(p_2(va)) \\
r(p_2(va)) \\
r(p_2(va)) \\
E_v
\end{pmatrix} \\
-vr(p_2(va)) \\
vr(p_2(va)) \\
0
\end{array}
\end{pmatrix}
\begin{pmatrix}
v \\
0 \\
0 \\
R(G, \mathbf{p}_2; E_v, V \setminus \{v\})
\end{pmatrix}
\]

(6.39)

In a manner similar to the case of \( R(G, \mathbf{p}_1) \), we perform the fundamental column operations on \( R(G, \mathbf{p}_2) \) which add the \( j \)-th column of \( R(G, \mathbf{p}_2; v) \) to that of \( R(G, \mathbf{p}_2; b) \) for each \( 1 \leq j \leq D \). Substituting \( p_2(va) = q(ab) \) and \( R(G, \mathbf{p}_2; E_v, V \setminus \{v\}) = R(G_{v}^{ab}, \mathbf{q}; E_v, V \setminus \{v\}) \) into the resulting matrix, we obtain

\[
R(G, \mathbf{p}_2) = \begin{pmatrix}
\begin{array}{c}
\begin{pmatrix}
v \\
a \\
b \\
E_v
\end{pmatrix}
\end{array}
\end{pmatrix}
\begin{pmatrix}
\begin{array}{c}
\begin{pmatrix}
vr(p_2(va)) \\
vr(p_2(va)) \\
vr(p_2(va)) \\
E_v
\end{pmatrix} \\
v \\
0 \\
0 \\
R(G_{v}^{ab}, \mathbf{q})
\end{array}
\end{pmatrix}
\]

(6.40)

(see the proof of Lemma 6.35 for more details). Let us remark the difference between \( k = 0 \) (the current situation) and \( k > 0 \) (Lemma 6.35). In the proof of Lemma 6.35, we have proved that \( (G, \mathbf{p}_1) \) attains the desired rank as rank \( R(G, \mathbf{p}_1) \geq D - 1 + \text{rank } R(G_{v}^{ab}, \mathbf{q}) = D - 1 + D(|V \setminus \{v\}| - 1) - (k - 1) = D(|V| - 1) - k \), where rank \( R(G_{v}^{ab}, \mathbf{q}) \) was equal to \( D(|V \setminus \{v\}| - 1) - (k - 1) \) since \( G_{v}^{ab} \) was a minimal \((k - 1)\)-dof-graph. In contrast, because of (6.37), we only have rank \( R(G, \mathbf{p}_1) \geq D - 1 + \text{rank } R(G_{v}^{ab}, \mathbf{q}) = D - 1 + D(|V \setminus \{v\}| - 1) = D(|V| - 1) - 1 \) in the current situation, which does not complete the proof. (Similarly, rank \( R(G, \mathbf{p}_2) \geq D(|V| - 1) - 1 \).

We shall further convert \( R(G, \mathbf{p}_1) \) of (6.35) to the matrix given in (6.48) by applying appropriate fundamental row operations based on the claim below (Claim 6.38). The matrix given by (6.48) has the following two properties: (i) the entries of the submatrix \( R(G, \mathbf{p}_1; \{v\}_1, \{v\}) \) (whose size is \( 1 \times D|V \setminus \{v\}| \)) are all zero, and (ii) the bottom-right submatrix (denoted by \( R(G_{v}^{ab} \setminus \{ab\}, \mathbf{q}) \) in (6.48)) has rank equal to \( D(|V \setminus \{v\}| - 1) \). This implies that, if the top-left \( D \times D \)-submatrix of (6.48) has full rank, then rank \( R(G, \mathbf{p}_1) = D + D(|V \setminus \{v\}| - 1) = D(|V| - 1) \) holds. Symmetrically, we shall convert \( R(G, \mathbf{p}_2) \) to that given in (6.49) which has the properties similar to (i) and (ii). If one of the top-left \( D \times D \) submatrices of (6.48) and (6.49) has full rank, we are done. However, this is not always the case. Hence, we introduce another realization \( (G, \mathbf{p}_3) \), whose rigidity matrix is given in (6.54). In a manner similar to \( R(G, \mathbf{p}_1) \) and \( R(G, \mathbf{p}_2) \), we convert \( R(G, \mathbf{p}_3) \) to that given in (6.60). We will show that the top-left \( D \times D \) submatrix has full rank in at least one of (6.48), (6.49) and (6.60), which completes the proof.

Let us first show how \( R(G, \mathbf{p}_1) \) given in (6.35) is converted to (6.48). For this, let us focus on \( R(G_{v}^{ab}, \mathbf{q}) \) given in (6.36) for a while. We say that a row vector of \( R(G_{v}^{ab}, \mathbf{q}) \) is redundant if removing it from \( R(G_{v}^{ab}, \mathbf{q}) \) does not decrease the rank of the matrix. The following claim is a
key observation, which states a relation between the combinatorial matroid $M(G_{v}^{ab})$ (i.e., the union of the $D$ graphic matroids) and the linear matroid derived from the row dependence of $R(G_{v}^{ab}, q)$.

**Claim 6.38.** In $R(G_{v}^{ab}, q)$, at least one of row vectors associated with $ab$ is redundant.

**Proof.** Recall a combinatorial property of $G_{v}^{ab}$ given in Lemma 6.13: Since $G_{v}^{ab}$ is a minimal $k$-dof-graph (with $k = 0$), Lemma 6.13(ii) says that there exists a base $B'$ of the matroid $M(G_{v}^{ab})$ satisfying $|B' \cap \tilde{a}b| < D - 1$. This implies that $\tilde{G}_{v}^{ab}$ has some redundant edge $(ab)_i$ among $\tilde{a}b$ with respect to $M(\tilde{G}_{v}^{ab})$ (i.e., removing $(ab)_i$ from $\tilde{G}_{v}^{ab}$ does not decrease the rank of $M(\tilde{G}_{v}^{ab})$). We show that this redundancy also appears in the linear matroid derived from $R(G_{v}^{ab}, q)$.

Let $B'$ be a base of the matroid $M(\tilde{G}_{v}^{ab})$ satisfying $|B' \cap \tilde{a}b| < D - 1$ mentioned above. Let $h = |B' \cap \tilde{a}b|$. Then, we have $h \leq D - 2$ and $|B'| = D(|V \setminus \{v\}| - 1)$ (since $G_{v}^{ab}$ is a 0-dof-graph). We shall consider the graph $G_{v}$ that is obtained from $G_{v}^{ab}$ by removing $ab$. Clearly, $B' \setminus \tilde{a}b$ is an independent set of $M(\tilde{G}_{v})$ with cardinality $D(|V \setminus \{v\}| - 1) - h$ and hence we have $\text{def}(\tilde{G}_{v}) \leq h$ by (6.6). Namely, $G_{v}$ is a $k'$-dof-graph for some nonnegative integer $k'$ with $k' \leq h \leq D - 2$. Also, $G_{v}$ is minimal by Lemma 6.7. Let $q_{E_{v}}$ denote the restriction of $q$ to the edge set $E_{v}$ of $G_{v}$. We claim the following.

$$\text{rank } R(G_{v}, q_{E_{v}}) = D(|V \setminus \{v\}| - 1) - k' \geq D(|V \setminus \{v\}| - 1) - (D - 2). \quad (6.41)$$

To see this, recall that, by the induction hypothesis (6.20), the rigidity matrix of any generic nonparallel panel-and-hinge realization of $G_{v}$ takes rank equal to $D(|V \setminus \{v\}| - 1) - k'$. Recall also that $(G_{v}^{ab}, q)$ was defined as a generic nonparallel realization of $G_{v}^{ab}$ in the inductive step. Hence, $q$ was taken in such a way that the set of all the coefficients appearing in the equations expressing the panels is algebraically independent over $Q$. This property clearly remains in the realization restricted to $E_{v}$ and hence $(G_{v}, q_{E_{v}})$ is also a generic nonparallel panel-and-hinge realization of $G_{v}$. Thus, (6.41) holds.

Note that $R(G_{v}, q_{E_{v}})$ is the matrix obtained from $R(G_{v}^{ab}, q)$ by removing the $D - 1$ rows associated with $ab$. Note also that $R(G_{v}^{ab}, q)$ has rank equal to $D(|V \setminus \{v\}| - 1)$ by (6.37).

Hence, from (6.41), adding at most $D - 2$ row vectors associated with $ab$ to the rows of $R(G_{v}, q_{E_{v}})$, we obtain a set of row vectors which spans the row space of $R(G_{v}^{ab}, q)$. This implies that at least one row vector associated with $ab$ is redundant.

Let $i^{*}$ be the index of a redundant row associated with $ab$ shown in Claim 6.38. Namely, denoting by $R(G_{v}^{ab} \setminus (ab)_i^{*}, q)$ the matrix obtained from $R(G_{v}^{ab}, q)$ by removing the row associated with $(ab)_i^{*}$, we have

$$\text{rank } R(G_{v}^{ab} \setminus (ab)_i^{*}, q) = \text{rank } R(G_{v}^{ab}, q) = D(|V \setminus \{v\}| - 1), \quad (6.42)$$

The genericity of a nonparallel panel-and-hinge realization for a simple graph says that, if one particular nonparallel realization achieves rank equal to $D(|V \setminus \{v\}| - 1) - k'$, then all generic nonparallel realizations attain the same rank.
where the last equation follows from (6.37). Also, since \( R(G^{ab}, p; (ab)_i) \) is redundant, it can be expressed as a linear combination of the other row vectors of \( R(G^{ab}, q) \). Thus, there exists a scalar \( \lambda_{e_j} \) for each \( e_j \in E_v \cup \{ab\} \) such that

\[
\sum_{1 \leq j \leq D-1} \lambda_{(ab)j} R(G^{ab}, q; (ab)_j) + \sum_{e \in E_v, 1 \leq j \leq D-1} \lambda_{e_j} R(G^{ab}, q; e_j) = 0, \tag{6.43}
\]

where

\[
\lambda_{(ab)\ast} = 1. \tag{6.44}
\]

In other words, the fundamental row operations change the matrix \( R(G^{ab}, q) \) so that the row associated with \( (ab)_\ast \) becomes a zero vector. Such row operations can be extended to \( R(G, p) \) of (6.35), by focusing on the fact that \( R(G, p; E \setminus \{vb\}, V \setminus \{v\}) \) is equal to \( R(G^{ab}, q) \), so that the part of a row vector of \( R(G, p) \) corresponding to \( V \setminus \{v\} \) becomes zero as in Figure 6.5.

Let us show that the resulting matrix can be described as (6.48) (up to some row exchanges). To show this, let us take a look at these row operations more precisely. Since the row associated with \( (ab)_\ast \) in \( R(G^{ab}, q) \) corresponds to the row associated with \( (vb)_\ast \) in \( R(G, p) \) of (6.35), performing such row operations is equivalent to the addition of

\[
\sum_{1 \leq j \leq D-1, j \neq \ast} \lambda_{(ab)j} R(G, p; (vb)_j) + \sum_{e \in E_v, 1 \leq j \leq D-1} \lambda_{e_j} R(G, p; e_j) \tag{6.45}
\]

to the row associated with \( (vb)_\ast \) in (6.35). Namely, the row operations convert \( R(G, p; (vb)_\ast) \) of (6.35) to

\[
\sum_{1 \leq j \leq D-1} \lambda_{(ab)j} R(G, p; (vb)_j) + \sum_{e \in E_v, 1 \leq j \leq D-1} \lambda_{e_j} R(G, p; e_j), \tag{6.46}
\]

where \( \lambda_{(ab)\ast} = 1. \) (Compare it with the left hand side of (6.43).) From \( R(G, p; E \setminus \{va\}, V \setminus \{v\}) = R(G^{ab}, q) \) and (6.43), the components of (6.46) associated with \( V \setminus \{v\} \) become all zero. As for the remaining part of the new row vector (6.46) (i.e., the components associated with \( v \)), since \( R(G, p; E_v, v) = 0 \) holds as shown in (6.35), we have

\[
\sum_{1 \leq j \leq D-1} \lambda_{(ab)j} R(G, p; (vb)_j, v) + \sum_{e \in E_v, 1 \leq j \leq D-1} \lambda_{e_j} R(G, p; e_j, v) = \sum_{1 \leq j \leq D-1} \lambda_{(ab)j} R(G, p; (vb)_j, v)
\]

Figure 6.5: Row operations on \( R(G, p) \).
Note that \( R(G, p_1; (vb)_j, v) = r_j(p_1(vb)) = r_j(q(ab)) \) by the definition of the rigidity matrix and by \( p_1(vb) = q(ab) \). Thus, we have seen that the new row vector (6.46) can be written as

\[
(\sum_j \lambda_{(ab)} r_j(q(ab)), \quad V \setminus \{v\})^T.
\]

(6.47)

As a result, the fundamental row operations change \( R(G, p_1) \) of (6.35) to the matrix described as (up to some row exchange of \((vb)_j, v\))

\[
R(G, p_1) = \begin{pmatrix}
va & R(G_v^{ab} \setminus (ab)_i, q) \\
(vb)_j, v & \sum_j \lambda_{(ab)} r_j(q(ab))
\end{pmatrix}
\]

(6.48)

Applying the symmetric argument to \( R(G, p_2) \) shown in (6.40), we also have

\[
R(G, p_2) = \begin{pmatrix}
vb & R(G_v^{ab} \setminus (ab)_i, q) \\
(va)_j, v & \sum_j \lambda_{(ab)} r_j(q(ab))
\end{pmatrix}
\]

(6.49)

Note that the same \( \lambda_{(ab)}, 1 \leq j \leq D - 1 \) and the index \( i^* \) are used in (6.48) and (6.49) since they are determined by \((G_v^{ab}, q)\), and are independent of \( p_1 \) and \( p_2 \). It is not difficult to see that, if the top-left \( D \times D \)-submatrix of (6.48) (that is, \( R(G, p_1; (vb)_j, v) \) has full rank, then we obtain rank \( R(G, p_1) \geq D + \text{rank} R(G_v^{ab} \setminus (ab)_i, q) = D(|V| - 1) \) by (6.48) and (6.42), and we are done. Symmetrically, if the top-left \( D \times D \)-submatrix of (6.49) has full rank, then \( R(G, p_2) \) attains the desired rank. In \( d = 2 \) we can show that this \( D \times D \)-submatrix has full rank in at least one of (6.48) and (6.49) but this is not always true for \( d \geq 3 \). Hence, we shall introduce another framework \((G, p_3)\) in the following discussion.

Let us consider the splitting off at \( a \) along \( vc \). (Recall that \( a \) is a vertex of degree two.) Then, since \( v \) and \( a \) are adjacent vertices of degree two, it is not difficult to see that the resulting graph \( G_v^{ab} \) is isomorphic to \( G_v^{ac} \) (see Figure 6.4(b) and (f)) by the isomorphism \( \varphi : V \setminus \{a\} \rightarrow V \setminus \{v\} \), i.e., \( \varphi(v) = a \) and \( \varphi(u) = u \) for \( u \in V \setminus \{v, a\} \). The isomorphism \( \varphi \) induces the mapping \( q_\varphi \) on \( E \setminus \{va, ac\} \cup \{vc\} \) in a natural way defined by, for \( e = uw \in E \setminus \{va, ac\} \cup \{vc\} \),

\[
q_\varphi(e) = q(\varphi(u)\varphi(w)) = \begin{cases}
q(ab) & \text{if } e = vb \\
q(ac) & \text{if } e = vc \\
q(e) & \text{otherwise}.
\end{cases}
\]

(6.50)

The isomorphism \( \varphi \) between \( G_v^{ab} \) and \( G_v^{ac} \) implies that \( (G_v^{ac}, q_\varphi) \) and \( (G_v^{ab}, q) \) represent the same panel-and-hinge framework in \( \mathbb{R}^3 \). In particular, we have \( \Pi_{G_v^{ac}, q_\varphi}(u) = \Pi_{G_v^{ab}, q}(u) \) for each \( u \in V \setminus \{v, a\} \) and \( \Pi_{G_v^{ac}, q_\varphi}(v) = \Pi_{G_v^{ab}, q}(a) \).

We shall construct a similar extension of the mapping \( q_\varphi \) as was done in \( p_1 \) or \( p_2 \). Define
a mapping $p_3$ on $E$ as

$$p_3(e) = \begin{cases} q_0(e) & \text{if } e \in E \setminus \{va, ac\} \\ L'' & \text{if } e = ac \\ q_0(vc) & \text{if } e = va \end{cases} \quad (6.51)$$

where $L''$ is a $(d-2)$-affine subspace contained in $\Pi_{G_{v}^{eb}, q}(c) = \Pi_{G_{v}^{q}, q}(c)$ and it can be shown as in Claim 6.36 that $(G, p_3)$ is a panel-and-hinge realization of $G$ in $\mathbb{R}^d$ for any choice of $L'' \subset \Pi_{G_{v}^{eb}, q}(c)$ (see Figure 6.4(h)). Combining (6.50) and (6.51), we actually have

$$p_3(e) = \begin{cases} q(ac) & \text{if } e = va \\ q(ab) & \text{if } e = vb \\ L'' & \text{if } e = ac \\ q(e) & \text{otherwise.} \end{cases} \quad (6.52)$$

Therefore, we have

$$r(p_3(va)) = r(q(ac))$$
$$r(p_3(vb)) = r(q(ab))$$

$$R(G, p_3; E \setminus \{va, vb, ac\}, V \setminus \{v, a\}) = R(G_{v}^{eb}, q; E \setminus \{va, vb, ac\}, V \setminus \{v, a\}). \quad (6.53)$$

We are now going to convert the rigidity matrix $R(G, p_3)$ to that given in (6.60), which is an analogous form to (6.48) or (6.49). Let us take a look at $R(G, p_3)$:

$$R(G, p_3) = \begin{pmatrix} a & v & c & b & V \setminus \{v,a,b,c\} \\ va & -r(p_3(va)) & 0 & r(p_3(ac)) & 0 \\ vb & 0 & r(p_3(vb)) & 0 & -r(p_3(vb)) \\ ac & 0 & 0 & 0 & 0 \\ va & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6.54)$$

Consider the fundamental column operations which add the $j$-th column of $R(G, p_3; a)$ to that of $R(G, p_3; c)$ for each $1 \leq j \leq D$. Then $R(G, p_3)$ results in the matrix as follows:

$$R(G, p_3) = \begin{pmatrix} a & v & c & b & V \setminus \{v,a,b,c\} \\ va & -r(p_3(va)) & 0 & r(p_3(ac)) & 0 \\ vb & 0 & r(p_3(vb)) & 0 & -r(p_3(vb)) \\ ac & 0 & 0 & 0 & 0 \\ va & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6.55)$$

Substituting (6.53) into (6.55), $R(G, p_3)$ becomes

$$R(G, p_3) = \begin{pmatrix} a & v & c & b & V \setminus \{v,a,b,c\} \\ va & -r(p_3(va)) & 0 & r(p_3(ac)) & 0 \\ vb & 0 & r(q(ab)) & 0 & -r(q(ab)) \\ ac & 0 & 0 & 0 & 0 \\ va & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6.56)$$
Note that \( R(G^{ab}, q) \) can be described by (up to some row exchanges)

\[
R(G^{ab}, q) = \begin{pmatrix}
  a & c & b & V \setminus \{v, a, b, c\} \\
  r(q(ab)) & 0 & -r(q(ab)) & 0 \\
  r(q(ac)) & -r(q(ac)) & 0 & 0 \\
  0 & R(G^{ab, q}; E \setminus \{va, vb, ac\}, V \setminus \{v, a\}) \\
\end{pmatrix}
\]

(6.57)

which is equal to \( R(G, p_3; E \setminus \{ac\}, V \setminus \{a\}) \) given in (6.56). Therefore, (6.56) becomes

\[
R(G, p_3) = \begin{pmatrix}
  a & V \setminus \{a\} \\
  r(p_3(ac)) & 0 \\
  0 & r(p_3(va)) \\
  0 & R(G^{ab, q}; E \setminus \{va, vb, ac\}, V \setminus \{v, a\}) \\
\end{pmatrix}.
\]

(6.58)

Recall that the row of \( R(G^{ab}, q) \) associated with \((ab)i^*\) is redundant, i.e., removing the row associated with \((ab)i^*\) from \( R(G^{ab}, q) \) preserves the rank of \( R(G^{ab}, q) \) as shown in (6.42). In order to indicate the dependence of the row vectors within \( R(G^{ab}, q) \), we have introduced \( \lambda_e \in \mathbb{R} \) for each \( e \in E_v \cup ab \) (where \( E_v = E \setminus \{va, vb\} \)) such that they satisfy (6.43).

We shall again consider the row operations which convert \( R(G^{ab}, q); (ab)i^* \) to a zero vector within \( R(G^{ab}, q) \) and apply these row operations to the matrix \( R(G, p_3) \) of (6.58) as shown in Figure 6.6.

![Figure 6.6: Row operations on \( R(G, p_3) \).](image)

Note that, within the equation \( R(G, p_3; E \setminus \{ac\}, V \setminus \{a\}) = R(G^{ab}, q) \) of (6.58), the rows associated with \((ab)j\) and \((ac)j\) in \( R(G^{ab}, q) \) correspond to those associated with \((vb)j\) and \((va)j\) in \( R(G, p_3) \), respectively, (compare (6.56) with (6.57)). In particular, the row of \((ab)i^*\) in \( R(G^{ab}, q) \) corresponds to that of \((vb)i^*\) in \( R(G, p_3) \). Therefore, the resulting matrix can be described as (6.60).

Let us check it more formally. We perform the fundamental row operations which add

\[
\sum_{1 \leq j \leq D-1, j \neq i^*} \lambda_{(ab)j} R(G, p_3; (vb)j) + \sum_{1 \leq j \leq D-1} \lambda_{(ac)j} R(G, p_3; (va)j) + \sum_{e \in E_v \setminus \{va, vb, ac\}} \lambda_e R(G, p_3; e_j)
\]
to $R(G, p_3; (vb)_i)$ of (6.58). Namely, by $\lambda_{(ab),*} = 1$, the resulting row becomes

$$
\sum_{1 \leq j \leq D-1} \lambda_{(ab),j} R(G, p_3; (vb)_j) + \sum_{1 \leq j \leq D-1} \lambda_{(ac),j} R(G, p_3; (va)_j) + \sum_{e \in E \setminus \{va, vb, ac\}} \lambda_{e, j} R(G, p_3; e_j). \tag{6.59}
$$

Since there is the row correspondence between (6.56) and (6.57) described as,

$$
\begin{align*}
R(G, p_3; vb, V \setminus \{a\}) &= R(G^b_v, q; ab), \\
R(G, p_3; va, V \setminus \{a\}) &= R(G^b_v, q; ac), \\
R(G, p_3; e, V \setminus \{a\}) &= R(G^b_v, q; e)
\end{align*}
$$

for each $e \in E \setminus \{va, vb, ac\}$,

(6.43) implies

$$
\sum_{1 \leq j \leq D-1} \lambda_{(ab),j} R(G, p_3; (vb)_j, V \setminus \{a\}) + \sum_{1 \leq j \leq D-1} \lambda_{(ac),j} R(G, p_3; (va)_j, V \setminus \{a\}) + \sum_{e \in E \setminus \{va, vb, ac\}} \lambda_{e, j} R(G, p_3; e_j, V \setminus \{a\}) = 0,
$$

and hence the part of the new row vector (6.59) associated with $V \setminus \{a\}$ actually becomes zero. The rest part of (6.59) is the $D$ consecutive components associated with $a$. Since the entries of $R(G, p_3; E \setminus \{ac, va\}, a)$ of (6.58) are all zero, we have

$$
\begin{align*}
&\sum_{1 \leq j \leq D-1} \lambda_{(ab),j} R(G, p_3; (vb)_j, V) + \sum_{1 \leq j \leq D-1} \lambda_{(ac),j} R(G, p_3; (va)_j, V) + \sum_{e \in E \setminus \{va, vb, ac\}} \lambda_{e, j} R(G, p_3; e_j, V) \\
&= \sum_{1 \leq j \leq D-1} \lambda_{(ac),j} R(G, p_3; (va)_j, V) \quad \text{(by } R(G, p_3; E \setminus \{ac, va\}, a) = 0 \text{ by (6.58))} \\
&= -\sum_{1 \leq j \leq D-1} \lambda_{(ac),j} r_j(p_3(va)) \quad \text{(by } R(G, p_3; va, a) = -r(p_3(va)) \text{ by the definition)} \\
&= -\sum_{1 \leq j \leq D-1} \lambda_{(ac),j} r_j(q(ac)). \quad \text{(by } p_3(va) = q(ac) \text{ by (6.53))}
\end{align*}
$$

Therefore, the new row vector (6.59) is actually described as

$$
( -\sum_j \lambda_{(ac),j} r_j(q(ac)), V\setminus\{a\}, \begin{bmatrix} V\setminus\{a\} & 0 \\ 0 & 0 \end{bmatrix} ),
$$

and as a result $R(G, p_3)$ of (6.58) is changed to the following matrix by the row operations:

$$
R(G, p_3) = (ab)_r \begin{pmatrix}
\begin{bmatrix}
-\lambda_{(ac),j} r_j(q(ac)) \\
\sum_{1 \leq j \leq D-1} -r(p_3#ac) \\
\end{bmatrix} & V\setminus\{a\} \\
0 & 0 \\
\end{pmatrix}
\end{pmatrix}, \quad \begin{pmatrix}
c \begin{bmatrix}
\begin{pmatrix}
-\lambda_{(ac),j} r_j(q(ac)) \\
\sum_{1 \leq j \leq D-1} -r(p_3#ac) \\
\end{bmatrix} & V\setminus\{a\} \\
0 & 0 \\
\end{pmatrix} \\
\end{pmatrix}, \quad (6.60)
where $R(G^a_v \backslash (ab)_i, q)$ was defined as the matrix obtained from $R(G^a_v, q)$ by removing the row associated with $(ab)_i$ (see (6.42)).

We finally show in the following Claim 6.39 that at least one of the top-left $D \times D$-submatrices of (6.48), (6.49), and (6.60) has full rank. Let us simply denote these three submatrices by $M_1$, $M_2$ and $M_3$, respectively, i.e.,

$$M_1 = \begin{pmatrix} r(p_1(va)) \\ \sum_j \lambda_{(ab)} r_j(q(ab)) \end{pmatrix}, \quad M_2 = \begin{pmatrix} r(p_2(vb)) \\ \sum_j \lambda_{(ab)} r_j(q(ab)) \end{pmatrix},$$

$$M_3 = \begin{pmatrix} r(p_3(ac)) \\ \sum_j \lambda_{(ac)} r_j(q(ac)) \end{pmatrix}.$$

From the definitions (6.31), (6.38) and (6.51), $p_1(va) = L$, $p_2(vb) = L'$, and $p_3(ac) = L''$ hold, where we may choose any $(d-2)$-affine subspaces $L \subset \Pi_{G^a_v,q(a)}$, $L' \subset \Pi_{G^a_v,q(b)}$, and $L'' \subset \Pi_{G^a_v,q(c)}$, respectively. Hence, $M_1$, $M_2$, and $M_3$ are described as

$$M_1 = \begin{pmatrix} r(L) \\ \sum_j \lambda_{(ab)} r_j(q(ab)) \end{pmatrix}, \quad M_2 = \begin{pmatrix} r(L') \\ \sum_j \lambda_{(ab)} r_j(q(ab)) \end{pmatrix}, \quad M_3 = \begin{pmatrix} r(L'') \\ \sum_j \lambda_{(ac)} r_j(q(ac)) \end{pmatrix}. \quad (6.61)$$

The following Claim 6.39 implies that at least one of $R(G, p_1)$, $R(G, p_2)$ and $R(G, p_3)$ attains the desired rank $D(|V| - 1)$. (For example, if $M_3$ has full rank, then we have rank $R(G, p_3) \geq \text{rank} M_3 = \text{rank } R(G^a_v \backslash (ab)_i, q) = D + D(|V| - 2) = D(|V| - 1)$ from (6.60) and (6.42), completing the proof of Lemma 6.37).

**Claim 6.39.** At least one of $M_1$, $M_2$ and $M_3$ has full rank for some choices of $L \subset \Pi_{G^a_v,q(a)}$, $L' \subset \Pi_{G^a_v,q(b)}$, and $L'' \subset \Pi_{G^a_v,q(c)}$.

**Proof.** Let $r \in \mathbb{R}^D$ be $\sum_j \lambda_{(ab)} r_j(q(ab))$. Note that $r$ is a nonzero vector because $r_1(q(ab)), \ldots, r_{D-1}(q(ab))$ are linearly independent and also $\lambda_{(ab)} = 1$. Suppose that $M_1$ given in (6.61) does not have full rank. Then, $r$ is contained in the row space of $r(L)$. This is equivalent to that $r$ is contained in the orthogonal complement of the vector space spanned by a $(d-1)$-extensor (2-extensor) $C(L)$ associated with $L$. Similarly, if $M_2$ given in (6.61) does not have full rank, then $r$ is contained in the orthogonal complement of the vector space spanned by $C(L')$.

We claim the following: If $M_3$ does not have full rank, then $r$ is also contained in the orthogonal complement of the space spanned by $C(L'')$. To see this, we need to remind ourselves of the equation (6.43) of $D|V|$-dimensional vectors (indicating the row dependence within $R(G^a_v, q)$). Focusing on the $D$ consecutive components associated with $a$ of (6.43), we have

$$0 = \sum_{e \in E_v \cup \{a\} \atop 1 \leq j \leq D-1} \lambda_{e_j} R(G^a_v, q; e_j, a). \quad (6.62)$$
Recall that $R(G^a_v, \mathbf{q}; e, a) = 0$ holds if $e$ is not incident to $a$ in $G^a_v$ according to the definition of a rigidity matrix. Since only $ab$ and $ac$ are incident to $a$ in $G^a_v$, we actually have

$$
\sum_{e \in E \cup \{ab\}} \sum_{1 \leq j \leq D-1} \lambda_{e,j} R(G^a_v, \mathbf{q}; e, j, a) = \sum_{1 \leq j \leq D-1} \lambda_{(ab),j} R(G^a_v, \mathbf{q}; (ab), j, a) + \sum_{1 \leq j \leq D-1} \lambda_{(ac),j} R(G^a_v, \mathbf{q}; (ac), j, a) = \sum_{1 \leq j \leq D-1} \lambda_{(ab),j} r_j(\mathbf{q}(ab)) + \sum_{1 \leq j \leq D-1} \lambda_{(ac),j} r_j(\mathbf{q}(ac)).$

Combining this and (6.62), we consequently obtain

$$
r = - \sum_{1 \leq j \leq D-1} \lambda_{(ac),j} r_j(\mathbf{q}(ac)).$$

(6.63)

If $M_3$ does not have full rank, then $\sum_j \lambda_{(ac),j} r_j(\mathbf{q}(ac))$ is contained in the orthogonal complement of the space spanned by $C(L'')$, which means that $r$ is contained in the orthogonal complement of the space spanned by $C(L'')$ as well, by (6.63).

As a result, we found that, if none of $M_1$, $M_2$ and $M_3$ has full rank, then there exists a nonzero vector $r \in \mathbb{R}^D$ contained in the orthogonal complement of the vector space spanned by the set of vectors (2-extensors)

$$
\left( \bigcup_{L \subset \Pi(a)} C(L) \bigcup \left( \bigcup_{L' \subset \Pi(b)} C(L') \bigcup \left( \bigcup_{L'' \subset \Pi(c)} C(L'') \right) \right),
$$

(6.64)

where $\Pi(v)$ denotes $\Pi_{G^a_v, \mathbf{q}}(v)$ for each $v \in \{a, b, c\}$. Therefore, in order to show that at least one of $M_1$, $M_2$ and $M_3$ has full rank, it is enough to show that the dimension of the vector space spanned by (6.64) is equal to $D$.

To show this, let us take four points in $\mathbb{R}^3$ as follows; $p_1 = \Pi(a) \cap \Pi(b) \cap \Pi(c), p_2 \in \Pi(a) \cap \Pi(b) \setminus \Pi(c), p_3 \in \Pi(b) \cap \Pi(c) \setminus \Pi(a)$, and $p_4 \in \Pi(c) \cap \Pi(a) \setminus \Pi(b)$. Since $(G^a_v, \mathbf{q})$ is a generic nonparallel framework, the set of the coefficients appearing in the equations expressing $\Pi(a), \Pi(b)$, and $\Pi(c)$ is algebraically independent over the rational field, and thus we can always take such four points in such a way that they are affinely independent. Also, notice that any line connecting two points among $\{p_1, \ldots, p_4\}$ is contained in $\Pi(a) \cup \Pi(b) \cup \Pi(c)$, and hence any 2-extensor obtained from two points among $\{p_1, \ldots, p_4\}$ belongs to (6.64). Recall that, by Lemma 3.21, the set of 2-extensors $\{p_i \vee p_j \mid 1 \leq i < j \leq 4\}$ is linearly independent, which means that the dimension of the vector space spanned by (6.64) is equal to $\binom{4}{2} = 6(= D)$.

As a result, at least one of $(G, p_1)$, $(G, p_2)$ and $(G, p_3)$ attains rank equal to $D(|V| - 1)$. As we already remarked, although $(G, p_1)$, $(G, p_2)$ and $(G, p_3)$ are not nonparallel, we can convert them to nonparallel panel-and-hinge realizations by slightly rotating the panel associated with $v$ (or that associated with $a$) without decreasing the rank of the rigidity matrix by Lemma 6.21. This completes the proof of Lemma 6.37.

\[\square\]
General dimension

Finally let us describe the general dimensional version of Lemma 6.37.

**Lemma 6.40.** Let $G = (V, E)$ be a 2-edge-connected minimal 0-dof-graph with $|V| \geq 3$. Suppose that there exists no proper rigid subgraph in $G$ and that (6.20) holds. Then, there is a nonparallel panel-and-hinge realization $(G, p)$ in $\mathbb{R}^d$ satisfying $R(G, p) = D(|V| - 1)$.

**Proof.** By Lemma 6.16, either $G$ is a cycle of length at most $d$ or $G$ has a chain of length $d$. If $G$ is a cycle of length at most $d$, then we are done by Lemma 6.24. Hence, let us consider the case where $G$ has a chain $v_0 v_1 v_2 \ldots v_d$ of length $d$ (where $d_G(v_i) = 2$ for $1 \leq i \leq d - 1$).

The proof strategy is exactly the same as $d = 3$. In general case, we shall consider $d$ distinct frameworks based on the chain $v_0 v_1 v_2 \ldots v_d$. For each $1 \leq i \leq d - 1$, the graph $G_{v_i}^{v_i-1v_{i+1}}$ obtained by the splitting off at $v_i$ is a minimal 0-dof-graph by Lemma 6.18. Let us simply denote $G_{v_i}^{v_i-1v_{i+1}}$ by $G_i$ for each $1 \leq i \leq d - 1$. By (6.20), there exist nonparallel panel-and-hinge realizations $(G_i, q_i)$ for $1 \leq i \leq d - 1$ such that they represent the same framework in $\mathbb{R}^d$. Based on $(G_i, q_i)$, we shall construct $d$ distinct frameworks for $G$ in the same way as in 3-dimensional case. We shall show that at least one of them attains the desired rank.

Let us start the proof. Consider $G_1 = (V \setminus \{v_1\}, E \setminus \{v_0v_1, v_1v_2\} \cup \{v_0v_2\})$. By (6.20), there exists a generic nonparallel panel-and-hinge framework $(G_1, q_1)$ which satisfies

$$R(G_1, q_1) = D(|V| - 2).$$ (6.65)

We first define two frameworks $(G, p_0)$ and $(G, p_1)$ based on $(G_1, q_1)$; for $e \in E$,

$$p_0(e) = \begin{cases} q_1(e) & \text{if } e \in E \setminus \{v_0v_1, v_1v_2\} \\ L_0 & \text{if } e = v_0v_1 \\ q_1(v_0v_2) & \text{if } e = v_1v_2 \end{cases}$$ (6.66)

$$p_1(e) = \begin{cases} q_1(e) & \text{if } e \in E \setminus \{v_0v_1, v_1v_2\} \\ q_1(v_0v_2) & \text{if } e = v_0v_1 \\ L_1 & \text{if } e = v_1v_2 \end{cases}$$ (6.67)

where $L_0$ is a $(d - 2)$-affine subspace contained in $\Pi_{G_1, q_1}(v_0)$ and $L_1$ is the one contained in $\Pi_{G_1, q_1}(v_2)$. Then, as was shown in Claim 6.36, it is not difficult to see that $(G, p_0)$ is a panel-and-hinge framework such that $\Pi_{G, p_0}(u) = \Pi_{G_1, q_1}(u)$ for $u \in V \setminus \{v_1\}$ and $\Pi_{G, p_0}(v_1) = \Pi_{G_1, q_1}(v_0)$. Similarly, $(G, p_1)$ is a panel-and-hinge framework such that $\Pi_{G, p_1}(u) = \Pi_{G_1, q_1}(u)$ for $u \in V \setminus \{v_1\}$ and $\Pi_{G, p_1}(v_1) = \Pi_{G_1, q_1}(v_2)$.

Let us take a look at $R(G, p_0)$ given by

$$R(G, p_0) = \begin{pmatrix} v_1 & v_0 & v_2 & V \setminus \{v_0, v_1, v_2\} \\ v_0v_1 & r(p_0(v_0v_1)) & -r(p_0(v_0v_1)) & 0 \\ v_0v_2 & 0 & -r(p_0(v_1v_2)) & 0 \\ v_1v_2 & 0 & 0 & R(G, p_0; E \setminus \{v_0v_1, v_1v_2\}, V \setminus \{v_1\}) \end{pmatrix}.$$ (6.68)
We shall apply the matrix manipulation as was given in the proof of Lemma 6.37, which converts the matrix of (6.33) to that of (6.48). Here, the vertices \( v_1, v_0 \) and \( v_2 \) play the roles of \( v, a \) and \( b \) of \( R(G, p_1) \) in the previous lemma, respectively. The matrix \( R(G, p_0) \) of (6.68) is changed to the following form by appropriate column and row operations by using the fact that the rows associated with \( v_0v_2 \) in \( R(G_1, q_1) \) correspond to those associated with \( v_1v_2 \) in \( R(G, p_0) \):

\[
R(G, p_0) = \begin{pmatrix}
& v_1 & \cdots & \cdots & \cdots \\
& r(L_0) & 0 & 0 & 0 \\
& \sum_j \lambda_{(v_0v_2)} r_j(q_1(v_0v_2)) & 0 & 0 & 0 \\
& * & R(G_1 \setminus (v_0v_2)^* \cdot q_1(1))
\end{pmatrix}, \tag{6.69}
\]

where several new notations appearing in (6.69) are defined as follows: the integer \( i^* \) is the index of a redundant row vector among those associated with \( v_0v_2 \) in \( R(G_1, q_1) \) (such a redundant edge always exists by Claim 6.38), \( R(G_1 \setminus \{(v_0v_2)^*\}, q_1) \) denotes the matrix obtained by removing the row of \((v_0v_2)^*\) from \( R(G_1, q_1) \) and satisfies

\[
\text{rank } R(G_1 \setminus \{(v_0v_2)^*\}, q_1) = \text{rank } R(G_1, q_1) = D(|V| - 2). \tag{6.70}
\]

Also, the scalar \( \lambda_{(v_0v_2)} \) comes from the reducibility of \( R(G_1, q_1; (v_0v_2)^*) \) within \( R(G_1, q_1) \), i.e., since \( R(G_1, q_1; (v_0v_2)^*) \) is redundant in \( R(G_1, q_1) \), it can be expressed by a linear combination of the other row vectors of \( R(G_1, q_1) \) and hence we have introduced \( \lambda_{e_j} \) for each \( e \in E \setminus \{v_0v_1, v_1v_2\} \cup \{v_0v_2\} \) and \( 1 \leq j \leq D - 1 \) such that \( \lambda_{(v_0v_2)^*} = 1 \) and

\[
\sum_{e \in E \setminus \{v_0v_1, v_1v_2\} \cup \{v_0v_2\}} \lambda_{e_j} R(G_1, q_1; e_j) = 0. \tag{6.71}
\]

This dependency will play a key role in the proof.

Symmetrically, we can convert \( R(G, p_1) \) to the following matrix by appropriate row and column fundamental operations:

\[
R(G, p_1) = \begin{pmatrix}
& v_1 & \cdots & \cdots & \cdots \\
& r(L_1) & 0 & 0 & 0 \\
& \sum_j \lambda_{(v_0v_2)} r_j(q_1(v_0v_2)) & 0 & 0 & 0 \\
& * & R(G_1 \setminus (v_0v_2)^* \cdot q_1(1))
\end{pmatrix}, \tag{6.72}
\]

Notice that the row vectors associated with \( v_0v_2 \) in \( R(G_1, q_1) \) correspond to those with \( v_0v_1 \) in \( R(G, p_1) \).

We are now going to construct the other \( d - 2 \) frameworks. Consider \( G_i = G_0^{v_i-1v_{i+1}} = (V \setminus \{v_i\}, E \setminus \{v_i-1v_i, v_i_{i+1}\} \cup \{v_i-1v_{i+1}\}) \) for \( 2 \leq i \leq d - 1 \). We shall focus on the following isomorphism \( \vartheta_i : V \setminus \{v_i\} \to V \setminus \{v_i\} \) between \( G_1 \) and \( G_i \) for each \( 2 \leq i \leq d - 1 \):

\[
\vartheta_i(u) = \begin{cases} 
    u & \text{if } u \in V \setminus \{v_1, v_2, \ldots, v_{i-1}, v_1\} \\
    v_{j+1} & \text{if } u = v_j \in \{v_1, v_2, \ldots, v_{i-1}\}.
\end{cases} \tag{6.73}
\]
Then, based on the isomorphism $\varrho_i$, we consider the nonparallel panel-and-hinge framework $(G_i, q_i)$ for $2 \leq i \leq d - 1$, which is exactly the same framework as $(G_1, q_1)$ such that

$$\Pi_{G_i, q_i}(u) = \Pi_{G_1, q_1}(\varrho_i(u)) \quad \text{for each } u \in V \setminus \{v_i\}. \quad (6.74)$$

More formally, $(G_i, q_i)$ is defined by the mapping $q_i$ on the edge set of $G_i$, which is $E \setminus \{v_{i-1}v_i, v_iv_{i+1}\} \cup \{v_{i-1}v_{i+1}\}$, defined as follows:

$$q_i(uw) = q_1(\varrho_i(u)\varrho_i(w)) = \begin{cases} q_1(uw) & \text{if } uw \in E \setminus \{v_0v_1, v_1v_2, \ldots, v_d-v_d\} \\ q_1(v_0v_2) & \text{if } uw = v_0v_1 \\ q_1(v_jv_{j+1}) & \text{if } uw = v_{j-1}v_j \text{ for } 2 \leq j \leq i - 1 \\ q_1(v_iv_{i+1}) & \text{if } uw = v_{i-1}v_{i+1} \\ q_1(v_jv_{j+1}) & \text{if } uw = v_{j-1}v_j \text{ for } i + 1 \leq j \leq d - 1. \end{cases} \quad (6.75)$$

Based on $(G_i, q_i)$, we shall construct the framework $(G, p_i)$ for each $2 \leq i \leq d - 1$ as follows (see Figure 6.7):

$$p_i(e) = \begin{cases} q_i(e) & \text{if } e \in E \setminus \{v_{i-1}v_i, v_iv_{i+1}\} \\ q_i(v_{i-1}v_{i+1}) & \text{if } e = v_{i-1}v_i \\ L_i & \text{if } e = v_iv_{i+1} \end{cases} \quad (6.76)$$

where $L_i$ is a $(d - 2)$-affine subspace contained in $\Pi_{G_i, q_i}(v_{i+1})$. Note that $\Pi_{G_i, q_i}(v_{i+1}) = \Pi_{G_1, q_1}(\varrho_i(v_{i+1})) = \Pi_{G_1, q_1}(v_{i+1})$ by (6.73) and (6.74). Hence, $L_i$ is a $(d - 2)$-affine subspace satisfying

$$L_i \subset \Pi_{G_1, q_1}(v_{i+1}). \quad (6.77)$$

As was shown in Claim 6.36, it is not difficult to see that $(G, p_i)$ is a panel-and-hinge framework satisfying $\Pi_{G, p_i}(u) = \Pi_{G_1, q_1}(\varrho_i(u)) = \Pi_{G_1, q_1}(q_1(u))$ for each $u \in V \setminus \{v_i\}$ and $\Pi_{G, p_i}(v_i) = \Pi_{G_i, q_i}(v_{i+1}) = \Pi_{G_1, q_1}(v_{i+1})$. Hence, $(G, p_i)$ is a panel-and-hinge framework such that only the panels of $v_i$ and $v_{i+1}$ coincide and all the other pairs of panels are nonparallel. We remark that $(G, p_i)$ can be converted to a nonparallel panel-and-hinge framework without decreasing the rank of the rigidity matrix by Lemma 6.21.

Combining (6.75) and (6.76), we have, for $2 \leq i \leq d - 1$,

$$p_i(e) = q_1(e) \quad \text{for } e \in E \setminus \{v_0v_1, v_1v_2, \ldots, v_d-v_d\}$$

$$p_i(v_0v_1) = q_1(v_0v_2)$$

$$p_i(v_jv_{j+1}) = q_1(v_jv_{j+1}) \quad \text{for } 2 \leq j \leq i \quad (6.78)$$

$$p_i(v_{i+1}) = L_i$$

$$p_i(v_j'v_{j'+1}) = q_1(v_j'v_{j'+1}) \quad \text{for } i + 1 \leq j' \leq d - 1 \text{ (if } i \neq d - 1).$$
Let us consider $R(G, p_i)$:

$$R(G, p_i) = \begin{vmatrix} v_i & v_{i+1} & v_{i-1} & V \setminus \{v_{i-1}, v_i, v_{i+1}\} \\
\frac{r(p_i(v_i v_{i+1}))}{r(p_i(v_{i-1} v_i))} & -r(p_i(v_i v_{i+1})) & 0 & 0 \\
0 & 0 & -r(p_i(v_{i-1} v_i)) & 0 \\
0 & 0 & 0 & R(G, p_i; E \setminus \{v_{i-1} v_i, v_i v_{i+1}\}, V \setminus \{v_i\}) \end{vmatrix} \quad (6.79)$$

We now convert $R(G, p_i)$ to the matrix which contains $R(G_1, q_1)$ as its submatrix: perform the column operations which add the $j$-th column of $R(G, p_i; v_i)$ to that of $R(G, p_i; v_{i+1})$ for each $1 \leq j \leq D$ and then substitute all of (6.78) into the resulting matrix. Then, it is not difficult to see that $R(G, p_i)$ of (6.79) is changed to

$$R(G, p_i) = \begin{vmatrix} v_i & V \setminus \{v_i\} \\
\frac{r(L_i)}{r(q_i(v_i v_{i+1}))} & 0 \\
0 & R(G_1, q_1) \end{vmatrix} \quad (6.80)$$
with (6.81), we shall perform the fundamental row operations which particular, the row associated with \( \lambda \) where we used the following row correspondence between \( R(G, p_i) \) derived from (6.78):

\[
\begin{align*}
R(G, p_i) & \quad R(G_1, q_1) \\
\{ v_0 v_1 \} & \quad \{ v_0 v_2 \} \quad \text{for } e \in E \setminus \{ v_0 v_1, \ldots, v_{d-1} v_d \} \\
v_{j-1} v_j & \quad v_j v_{j+1} \quad \text{for } 2 \leq j \leq i \\
v_j v_{j+1} & \quad v_j' v_{j'+1} \quad \text{for } i + 1 \leq j' \leq d - 1
\end{align*}
\]

(and the column correspondence follows from the isomorphism \( \varphi_i \) defined in (6.73)). In particular, the row associated with \((v_0 v_2)_i^*\) in \( R(G_1, q_1) \) corresponds to the row associated with \((v_0 v_1)_i^*\) in \( R(G, p_i) \).

Recall that the row of \( R(G_1, q_1) \) associated with \((v_0 v_2)_i^*\) is redundant, i.e., removing the row associated with \((v_0 v_2)_i^*\) from \( R(G_1, q_1) \) preserves the rank as shown in (6.70). As before, we consider the row operations which convert \( R(G_1, q_1; (v_0 v_2)_i^*) \) to a zero vector and apply these row operations to the matrix \( R(G, p_i) \) of (6.80). More precisely, following the row correspondence described in (6.81), we shall perform the fundamental row operations which add the row vectors of \( R(G, p_i; E \setminus \{ v_{i-1} v_i, v_i v_{i+1} \}) \) to the row \( R(G, p_i; (v_0 v_1)_i^*) \) with the weight \( \lambda_{v_j} \). Namely, the row operations change the row associated with \((v_0 v_1)_i^*\) in (6.80) to the following row vector:

\[
\begin{equation}
\sum_{1 \leq j \leq d-1} \lambda_{(v_0 v_2)_i^*} R(G, p_i; (v_0 v_1)_i^*) + \sum_{2 \leq j' \leq i, 1 \leq j \leq d-1} \lambda_{(v_j v_{j+1})_i^*} R(G, p_i; (v_{j-1} v_{j'})_i^*) \\
+ \sum_{i+1 \leq j' \leq d-1 \atop 1 \leq j \leq d-1} \lambda_{(v_j v_{j+1})_i^*} R(G, p_i; (v_j v_{j+1})_i^*) + \sum_{e \in E \setminus \{ v_0 v_1, \ldots, v_{d-1} v_d \}} \lambda_{v_j} R(G, p_i; v_j).
\end{equation}
\]

where \( \lambda_{(v_0 v_2)_i^*} = 1 \).

By (6.71), all the entries of the part of the new row vector (6.82) associated with \( V \setminus \{ v_i \} \) become zero. Also, as for the rest part of the new row (6.82) associated with \( v_i \), since the entries of \( R(G, p_i; E \setminus \{ v_{i-1} v_i, v_i v_{i+1} \}, v_i) \) of (6.80) are all zero, the row operations will change \( R(G, p_i; (v_0 v_1)_i^*, v_i) \) to

\[
\sum_{1 \leq j \leq d-1} \lambda_{(v_0 v_{i+1})_i^*} r_j(p_i(v_{i-1} v_i))
\]

which is equal to

\[
\sum_{1 \leq j \leq d-1} \lambda_{(v_0 v_{i+1})_i^*} r_j(q_1(v_{i+1} v_i))
\]

since \( p_i(v_{i-1} v_i) = q_1(v_{i+1} v_i) \) by (6.78). Therefore, the fundamental row operations actually change \( R(G, p_i) \) of (6.80) to the following matrix (up to some row exchange of \((v_0 v_1)_i^*\)):

\[
R(G, p_i) = \begin{pmatrix}
    v_i & 0 \\
    r(L_i) & 0 \\
    \sum_j \lambda_{(v_0 v_{i+1})_i^*} r_j(q_1(v_{i+1} v_i)) & 0 \\
    \ast & R(G_1 \setminus (v_0 v_2)_i^*, q_1)
\end{pmatrix}
\]
Let us denote the top-left $D \times D$-submatrices of (6.69) and (6.72) by $M_0$ and $M_1$, and also that of (6.83) by $M_i$ for $2 \leq i \leq d - 1$, i.e.,

$$M_0 = \begin{pmatrix} r(L_0) \\ \sum_j \lambda_{(v_0,v_2)} r_j(q(v_0v_2)) \end{pmatrix}, \quad M_1 = \begin{pmatrix} r(L_1) \\ \sum_j \lambda_{(v_0,v_2)} r_j(q(v_0v_2)) \end{pmatrix}, \quad M_i = \begin{pmatrix} r(L_i) \\ \sum_j \lambda_{(v_i,v_{i+1})} r_j(q(v_i,v_{i+1})) \end{pmatrix} \text{ for } 2 \leq i \leq d - 1.$$ 

Recall that $L_0$ can be taken as any $(d-2)$-affine subspace satisfying $L_0 \subset \Pi_{G_1,q_1}(v_0)$ while for $1 \leq i \leq d - 1$ $L_i$ can be any $(d-2)$-affine subspace satisfying $L_i \subset \Pi_{G_1,q_1}(v_{i+1})$. Then, as in the proof of 3-dimensional case, the remaining task is to show the following fact:

At least one of $M_0, M_1, \ldots, M_{d-1}$ has full rank for an $L_i, 0 \leq i \leq d - 1$. \hspace{1cm} (6.84)

If this is true, then we obtain rank $R(G, p_i) \geq$ rank $M_i \oplus$ rank $R(G_1 \setminus \{(v_0v_2), q_1\}) = D + D(|V| - 2) = D(|V| - 1)$ by (6.70).

Let us show (6.84). Let $r = \sum_j \lambda_{(v_0v_2)} r_j(q(v_0v_2))$. Suppose $M_0$ (and $M_1$, resp.) does not have full rank. Then, $r$ is contained in the row space of $r(L_0)$ (and $r(L_1)$, resp.), that is, $r$ is contained in the orthogonal complement of the vector space spanned by a $(d - 1)$-extensor $C(L_0)$ (and that of $C(L_1)$, respectively). Also, due to the fact that $v_i$ is a vertex of degree two in $G_1$ for all $2 \leq i \leq d - 1$, we can easily show the following fact in a manner similar to the previous lemma (c.f. (6.63)):

$$\sum_{1 \leq j \leq d - 1} \lambda_{(v_i,v_{i+1})} r_j(q(v_i,v_{i+1})) = \pm r \quad \text{for } 2 \leq i \leq d - 1 \hspace{1cm} (6.85)$$

Notice that (6.85) implies that $M_i$ does not have full rank if and only if $r$ is contained in the orthogonal complement of the vector space spanned by $C(L_i)$. Therefore, none of $M_0, M_1, \ldots, M_{d-1}$ has full rank for any choice of $L_i, 0 \leq i \leq d - 1$ if and only if $r$ is contained in the orthogonal complement of the vector space spanned by

$$\bigcup_{0 \leq i \leq d - 1} \left( \bigcup_{L_i \subset \Pi_i} C(L_i) \right),$$

where $\Pi_0 = \Pi_{G_1,q_1}(v_0)$ and $\Pi_i = \Pi_{G_1,q_1}(v_{i+1})$ for each $i$ with $1 \leq i \leq d - 1$. Therefore, in order to show that at least one of $M_0, M_1, \ldots, M_{d-1}$ has full rank, it is enough to show that the dimension of the vector space spanned by (6.86) is equal to $D$.

Since the framework $(G_1, q_1)$ is a generic nonparallel framework, the set of the coefficients appearing in the equations expressing $\Pi_i, 0 \leq i \leq d - 1$ is algebraically independent over the rational field. Therefore, for any $j$ hyperplanes among them, their intersection forms a $(d - j)$-dimensional affine space. Hence we can take $d + 1$ distinct points in $\mathbb{R}^d$ as follows; for each $0 \leq i \leq d - 1$, $p_i \in \bigcap_{0 \leq j \leq d - 1, j \neq i} \Pi_j \setminus \Pi_i$, and $p_d = \bigcap_{0 \leq j \leq d - 1} \Pi_j$. Clearly, $\{p_0, p_1, \ldots, p_d\}$ is affinely independent, and also notice that any $(d-2)$-affine subspace spanned by $(d-1)$ points among them is contained in $\bigcup_{0 \leq j \leq d - 1} \Pi_j$. This implies that any $(d-1)$-extensor
obtained from \((d - 1)\) points belongs to the set of \((6.86)\). Thus the dimension of the vector space spanned by \((6.86)\) is equal to \(\binom{d+1}{d-1} = D\) by Lemma 3.21.

This completes the proof of Lemma 6.40 as well as Theorem 6.25 in general dimension. 

6.7 Conclusion

We have proved that any body-and-hinge rigid graph can be realized as an infinitesimally rigid panel-and-hinge framework based on the new inductive construction of minimally body-and-hinge rigid graphs. This settles the Molecular Conjecture affirmatively for general dimension. Also, as a corollary we found that a 3-dimensional bar-and-joint framework \((G^2, p)\) is infinitesimally rigid on a generic joint configuration if and only if \(5G\) contains six edge-disjoint spanning trees. This provides the mathematical validity of using the pebble game algorithms for computing the degree of freedom of molecules.

We should remark that the study of a bar-and-joint realization of the square of a graph has not been completed yet. Indeed, Corollary 6.27 tells us how to compute the rank of the whole graph only, and the explicit formula of the rank function has not been clarified yet. The research should be continued to understand general 3-dimensional bar-and-joint frameworks whose combinatorial characterization remains unexplained.
Chapter 7

Enumerating Non-crossing Minimally Rigid Frameworks

Apart from the theoretical part of rigidity theory, in this chapter we shall discuss how to efficiently generate all minimally rigid bar-and-joint frameworks (i.e. isostatic frameworks). By Laman’s theorem (Theorem 3.14), $G = (V, E)$ is the underlying graph of a minimally rigid bar-and-joint framework on a 2-dimensional generic joint configuration if and only if $|E| = 2|V| - 3$ and $|F| \leq 2|V(F)| - 3$ for any nonempty $F \subseteq E$. Therefore, the problem of enumerating generically minimally rigid bar-and-joint frameworks is equivalent to the enumeration of graphs satisfying Laman’s condition, which are called minimally rigid graphs.

In this chapter we present an algorithm for enumerating without repetitions all non-crossing minimally rigid graphs (under edge constraints) on a given set of $n$ points in the plane. Our algorithm is based on the reverse search paradigm of Avis and Fukuda [11]. It generates each output graph in $O(n^3)$ time and $O(n^2)$ space.

7.1 Introduction

Given a graph $G = (V, E)$ with $V = \{1, \ldots, n\}$, an embedding of the graph on a set of points $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^2$ is a mapping of the vertices to the points in the Euclidean plane $i \mapsto p_i$. A geometric graph is a graph embedded on $P$ such that each edge $ij$ of $G$ is mapped to a straight-line segment $(p_i, p_j)$, which is simply called an edge (on $P$) if no ambiguity arises. A set of edges on $P$ is called non-crossing if any pair of elements does not have a point in common except possibly for their endpoints, and a geometric graph is called non-crossing if its corresponding edge set is non-crossing.

Recall that a graph $G$ is minimally rigid if $|E| = 2|V| - 3$ and $|F| \leq 2|V(F)| - 3$ for any nonempty $F \subseteq E$ (see Section 3.1.4). An embedded minimally rigid graph on a planar point set can be regarded as a minimally rigid framework if a point set is generic.

Let $F$ be a set of non-crossing edges on $P$. A geometric graph containing $F$ is called $F$-constrained. In this paper we consider the following enumeration problem:

Input: A point set $P$ in the plane with $n$ points.
Output: The list of all the $F$-constrained non-crossing minimally rigid graphs on $P$.

Since the output of an enumeration problem may consist of exponentially many graphs in terms of the input size, the efficiency of an enumeration algorithm is measured customarily in both the input and output sizes. In particular, if the computational time can be bounded by a polynomial in the input size and by a linear function in the output size, the algorithm is said to work in polynomial time (on average).

Historical perspective. A pseudo-triangle is a simple polygon with exactly three convex vertices. A pseudo-triangulation is the partition of the convex hull of a planar point set $P$ into the interior disjoint pseudo-triangles, the vertices of which are points in $P$. It is known that a pseudo-triangulation with minimum number of edges, called a minimum pseudo-triangulation or a pointed pseudo-triangulation, is a non-crossing minimally rigid graph [110]. However a minimum pseudo-triangulation contains all the edges of the convex hull of the underlying point set, whereas a non-crossing minimally rigid graph need not as illustrated in Figure 7.1. It is known that in any minimum pseudo-triangulation the removal of any non-convex-hull edge leads to the choice of a unique other edge that can replace it, in order to maintain the pseudo-triangulation property (see, e.g., [16]). This leads to the definition of a graph whose vertices are minimum pseudo-triangulations and whose edges are simple flips. Bereg [16] showed that this graph is connected and showed how to apply the reverse search technique to generate all of its vertices (which correspond to minimum pseudo-triangulations). Bereg’s efficient algorithm makes use of specific properties of minimum pseudo-triangulations which do not extend to arbitrary non-crossing minimally rigid graphs. In particular, remove-add flips are not unique, relative to the removed edge, in the case of a non-crossing minimally rigid graph.

Unfortunately, it is not feasible to generate all minimally rigid graphs of practical interest due to the huge output size. However, certain engineering considerations allow us to limit ourselves to minimally rigid graphs containing a given set $F$ of non-crossing edges, which help us dramatically reduce the output size.

Contribution. We use both matroids and triangulations as important tools to solve our enumeration problem. Firstly we should note that, although minimally rigid graphs form the bases of a matroid as discussed in Section 3.1.3, this is not true in general for non-crossing minimally rigid graphs on a point set $P$. Indeed for two non-crossing minimally rigid graphs

---

Figure 7.1: (a) A minimum pseudo-triangulation. (b) A non-crossing minimally rigid graph.
Figure 7.2: Non-crossing minimally rigid graphs need not form the bases of a matroid.

$G_1$ and $G_2$ depicted in Figure 7.2 (a) and (b) there exists no edge to insert into $G_1 - \{e_1\}$ among $G_2 - G_1$ maintaining non-crossing property when removing the edge $ab$ from $G_1$. However, by choosing a non-crossing graph as a ground set, all subgraphs are automatically non-crossing, and hence the non-crossing minimally rigid graphs which are subgraphs of this graph do form the bases of a matroid. Triangulations are natural candidates for such base graphs.

The reverse search enumeration technique of Avis and Fukuda [10, 11] has been successfully applied to a variety of combinatorial and geometric enumeration problems. The necessary ingredients to use the method are an implicitly described connected graph on the objects to be generated, and an implicitly defined spanning tree in this graph. (See Section 7.2.2 for more detail.) In this paper we use triangulations and matroids to supply these ingredients for the problem of generating constrained minimally rigid graphs. After proving the correctness of our approach, we give an implementation based on reverse search that allows the enumeration without repetitions of all $F$-constrained non-crossing minimally rigid graphs on $n$ points in $O(n^4)$ time and $O(n)$ space per output framework. A slightly different implementation yields $O(n^3)$ time and $O(n^2)$ space per output framework.

## 7.2 Enumeration Techniques

Let us start with the introduction of two basic enumeration techniques.

### 7.2.1 Binary partition

The *branch-and-bound technique* (or sometimes called the binary-partition technique, see, e.g., [120, 121]) is a well-known framework for designing enumeration algorithms. Consider, for example, the problem of enumerating all spanning trees in a (multi)graph $G$ with $n$ vertices and $m$ edges. Then, we can easily design an algorithm that enumerates all spanning trees in $O(m^2)$ time per output graph as follows. The algorithm repeatedly divides the problem into two subproblems: one enumerates the spanning trees containing an edge $e$ of $G$, and the other enumerates those not containing $e$. In the first subproblem $e$ is contracted (and resulting loops are removed if there exists any), while in the second subproblem $e$ is removed. Then, the problem size is surely reduced in each subproblem. Moreover, since it can be checked in $O(m)$ time whether the resulting graph contains at least one spanning tree, the
algorithm can decide correctly whether it should continue the search or not. Therefore, by going down this branch-and-bound tree in $O(m)$ steps, the algorithm surely detects a new spanning tree. For a cleverer implementation with a sophisticated time complexity analysis, refer to [120, 121].

Uno [121] presented an efficient algorithm for enumerating bases of a matroid based on the binary partition technique. Given a matroid $\mathcal{M}$ on a ground set $E$ with rank $r$, his algorithm generates all bases of $\mathcal{M}$ in $t_C = O(t_{\text{cir}}/r)$ time per base with the preprocessing time $t_{C, \text{pre}}$, where $t_{\text{cir}}$ is the time to calculate the fundamental circuit (see Section 2.2.1) of $B \cup \{e\}$ for a base $B$ and $e \in E \setminus B$, and $t_{C, \text{pre}}$ is the time to compute the coloops of the matroid in $E$ (where $e \in E$ is called a coloop if all bases contain $e$). In the case of the generic rigidity matroid (or more generally the matroid obtained from the rigidity matroid by the contraction of elements), the algorithm by Berg and Jordán [17] can detect the circuit of $B \cup \{e\}$ in $t_{\text{cir}} = O(r^2)$ time. Moreover, they also developed an algorithm for detecting all coloops in $t_{C, \text{pre}} = O(r^2)$ time. Since the rank $r$ of the rigidity matroid is at most $2n - 3$, it thus enumerates all minimally rigid graphs in $G$ that contain a specified edge set in $t_C = O(t_{\text{cir}}/r) = O(n)$ time per output graph with $t_{C, \text{pre}} = O(n^2)$ preprocessing time.

**Theorem 7.1.** Let $G = (V, E)$ be a graph. Then, all the minimally rigid graphs contained in $G$ and containing a specified edge set $F \subseteq E$ can be enumerated in $O(n)$ time per output graph with $O(n^2)$ preprocessing time.

The branch-and-bound technique provides us with polynomial time enumeration algorithms for many graph classes because it just requires a polynomial time oracle that checks whether a given graph contains at least one subgraph belonging to a certain graph class. However, the problem of detecting a non-crossing subgraph in a given geometric graph, even detecting a non-crossing spanning tree or a non-crossing perfect matching, is known to be NP-complete [67]. The proof of Jansen and Woeginger [67] can be easily extended to the problem of detecting a non-crossing minimally rigid graph.

**Theorem 7.2.** The problem of deciding whether a given geometric graph contains a non-crossing minimally rigid graph or not is NP-complete.

For this reason, it seems difficult to apply binary partition technique to the enumeration of non-crossing minimally rigid graphs.

### 7.2.2 Reverse search

Reverse search is a memory efficient method for visiting all the vertices developed by Avis and Fukuda [10, 11].

Let $\mathcal{O}$ be a set of objects to be enumerated. Two objects are *connected* if and only if they can be transformed to each other by a pre-defined *local operation*, where local operation generates one object from the other by means of small changes. Especially, local operation is sometimes called *(1-)flip* if they have all but one edge (or element) in common. Define a *flip graph* $G_{\mathcal{O}}$ on $\mathcal{O}$ with a set of edges connecting between objects that can be transformed
to each other by a local operation, e.g., a base exchange between two bases in matroid (see Section 2.2.1) or a pivot operation between two dictionaries in simplex method. The reverse search generates all the elements in $\mathcal{O}$ by tracing the vertices of $\mathcal{G}_O$. To trace $\mathcal{G}_O$ efficiently, it defines a root on $\mathcal{G}_O$ and parent for each vertex except for the root. Define the parent-child relation satisfying the following conditions:

1. each non-root object has a unique parent, and
2. an ancestor of an object is not itself.

By this, iterating going up to the parent leads to the root from any other vertex in $\mathcal{G}_O$ if $\mathcal{G}_O$ is connected. The set of such paths defines a spanning tree, known as the search tree, and the algorithm traces it by depth-first search manner. So, the necessary ingredients to use the method are an implicitly described connected flip graph, and an implicitly defined spanning tree in this graph. We will supply these ingredients for the problem of generating all the edge-constrained non-crossing minimally rigid graphs.

### 7.3 Constrained Non-crossing Minimally Rigid Graphs

Let $T$ be a triangulation on a given set of $n$ points $P$ in the plane, containing $k$ triangles. The angle vector of $T$ is the vector of $3k$ interior angles sorted into non-decreasing order. We say that $T$ is an $F$-constrained triangulation, denoted $T(F)$, if it contains a given set of non-crossing edges $F$. Many facts about $F$-constrained triangulations are contained in the survey by Bern and Eppstein [19]. If $F$ is an independent set in the generic rigidity matroid on the complete graph $K_n$ with $n$ vertices, then a minimally rigid graph on $P$ containing $F$ is called $F$-constrained. The following lemma is a well known fact about the Laman matroid follows from rigidity considerations (see, e.g. [44]).

**Lemma 7.3.** Let $F$ be a non-crossing edge set on $P$ that is an independent set in the generic rigidity matroid on $K_n$. Every $F$-constrained triangulation on $P$ contains an $F$-constrained minimally rigid graph.

An $F$-constrained Delaunay triangulation plays an important role for developing our results, which we define by means of legal (illegal) edge and Delaunay-flip shown below. Let $T(F)$ be an $F$-constrained triangulation on a non-crossing edge set $F$. An edge $(p_i, p_j)$ incident to triangles $p_ip_jp_k$ and $p_ip_jp_l$ is said to be legal (with respect to $T(F)$) if the circumcircle of $p_ip_jp_k$ does not contain $p_l$ in its interior. An edge of $T$ is illegal if it is not legal. Let $(p_i, p_k)$ be an illegal edge (with respect to $T(F)$) that is not in $F$ and is the diagonal of a convex quadrilateral $p_ip_jp_kp_l$ whose edges are contained in $T(F)$. The replacement of $(p_i, p_k)$ by $(p_j, p_l)$ in $T(F)$ is called a Delaunay-flip.

An $F$-constrained triangulation $T(F)$ is an $F$-constrained Delaunay Triangulation, denoted by $DT(F)$, if it admits no Delaunay-flips. Equivalently, all edges in $T(F) \setminus F$ are legal.

In Figure 7.3 (a), (b), and (c), we illustrate examples of an $F$-constrained triangulation $T(F)$, an $F$-constrained Delaunay triangulation $DT(F)$, and a Delaunay triangulation $DT(\emptyset)$.
without the constraint, respectively, where \( F = \{13, 16, 23, 25, 27, 47, 57, 67\} \). We observe that \( T(F) \) and \( DT(F) \) have illegal edges \{15, 25\} and \{25\}, respectively, and \( DT(\emptyset) \) has no illegal edge.

**Proposition 7.4.** (see e.g. [19, Lemma 4]) An \( F \)-constrained triangulation \( T(F) \) can be converted to \( DT(F) \) by at most \( O(n^2) \) Delaunay-flips, taken in any order.

If \( P \) has four or more co-circular points, using a linear transformation as described in [14], we may transform \( P \) into a point set \( \mathcal{P} \) with a unique \( DT(F) \) so that \( P \) and \( \mathcal{P} \) have the same number of non-crossing minimally rigid graphs since the transformation does not change the relative order with respect to \( x \)- and \( y \)-coordinates among any three points. We will assume in what follows that \( P \) has a unique \( DT(F) \).

Two points \( p_i \) and \( p_j \) are visible (with respect to \( F \)) if no edge of \( F \) properly intersects the segment \( (p_i, p_j) \). \((p_i, p_j)\) is visible to point \( p_k \) (with respect to \( F \)) if the triangle \( p_ip_jp_k \) is not properly intersected by an edge of \( F \). Then the following fact gives another characterization of the \( F \)-constrained Delaunay triangulation.

**Proposition 7.5.** (see e.g. [19, Definition 1]) Let \( F \) be a non-crossing edge set on \( P \). An \( F \)-constrained Delaunay Triangulation contains the edge \((p_i, p_j)\) between points \( p_i \) and \( p_j \) in \( P \) if and only if \( p_i \) is visible to \( p_j \), and any circle through \( p_i \) and \( p_j \) contains no point of \( P \) visible to segment \((p_i, p_j)\). We call \((p_i, p_j)\) a Delaunay-edge (with respect to \( F \)).

Let \( H(F) \) be a non-crossing edge set on \( P \) containing \( F \), and let us consider \( H(F) \)-constrained Delaunay triangulation \( DT(H(F)) \). We observe that all illegal edges in \( DT(H(F)) \) are contained in \( H(F) \). It is because that all edges of \( DT(H(F)) \setminus H(F) \) are legal edge by the definition of the edge-constrained Delaunay triangulation. We also notice that if an edge \( e \in H(F) \) is a legal edge in \( DT(H(F)) \), we have \( DT(H(F)) = DT(H(F) - e) \) since all edges of \( DT(H(F)) \setminus (H(F) - e) \) are legal in \( DT(H(F)) \). On the other hand \( DT(H(F) - e) \) can not contain \( e \) if \( e \) is illegal. In fact \( DT(H(F) - e) \) can be obtained from \( DT(H(F)) \) after performing at least one Delaunay-flip. Note in addition that a Delaunay-flip increases the angle vector lexicographically. This can be used to prove the following.

**Proposition 7.6.** (see e.g. [19, Theorem 1]) \( DT(F) \) has the lexicographically maximum angle vector of all \( F \)-constrained triangulations on \( P \).
7.3. Constrained Non-crossing Minimally Rigid Graphs

Figure 7.4: An example of a flip described in the proof of Theorem 7.7. The bold lines represent $F = \{13, 16, 23, 25, 27, 47, 57, 67\}$, and the dotted lines represent additional edges for the constrained Delaunay triangulations.

Now let us consider non-crossing minimally rigid graphs. We say that an $F$-constrained minimally rigid graph is an $F$-constrained Delaunay rigid graph if it is a subset of $DT(F)$. Note that, unlike $DT(F)$, a $F$-constrained Delaunay rigid graph is not uniquely defined in general. A flip on a non-crossing minimally rigid graph $G$ is an edge insertion and deletion that takes $G$ to a new non-crossing minimally rigid graph $G'$. 

**Theorem 7.7.** Every $F$-constrained non-crossing minimally rigid graph $G$ can be transformed to an $F$-constrained Delaunay rigid graph by at most $O(n^2)$ flips.

**Proof.** Construct the $G$-constrained Delaunay triangulation $T = DT(G)$. If $T = DT(F)$, then $G$ is an $F$-constrained Delaunay rigid graph and we are done. Otherwise $T$ contains some illegal edge $e_1 \notin F$ from the definition. Now all edges in $T \setminus G$ are legal edges with respect to $T$, so $e_1$ must be an edge of $G \setminus F$. Consider now the constrained Delaunay triangulation $DT(G - e_1)$ which contains $G - e_1$. By Lemma 7.3, $DT(G - e_1)$ contains a $(G - e_1)$-constrained minimally rigid graph $G'$. Since $G'$ contains $G - e_1$ it must contain one additional edge, say $e_2$, and so $G' = G - e_1 + e_2$. In other words $G'$ is obtained from $G$ by a flip. Observe that $G'$ is $F$-constrained since $e_1 \notin F$.

Now we can construct $DT(G - e_1)$ from $DT(G)$ by a series of Delaunay-flips, starting by deleting the illegal edge $e_1$. Each of these Delaunay-flips lexicographically increases the angle vector. If $DT(G - e_1)$ is not $DT(F)$ then we repeat the above procedure. From Proposition 7.4, after $O(n^2)$ Delaunay-flips we will obtain $DT(F)$. The corresponding minimally rigid graph $G''$ is an $F$-constrained Delaunay rigid graph. Namely, we will obtain $G''$ by at most $O(n^2)$ flips from $G$. 

In Figure 7.4 we show an example of a flip described in the proof of Theorem 7.7 in which $G$ is not an $F$-constrained Delaunay rigid graph: deleting the illegal edge 15 in $DT(L(F)) \setminus F$, and updating the constrained Delaunay triangulation to $DT(G - 15)$ we find another non-crossing minimally rigid graph shown in the rightmost and upper corner of Figure 7.4.
For edges \( e = (p_i, p_j) \) with \( i < j \) and \( e' = (p_k, p_l) \) with \( k < l \), we use the notation \( e <_L e' \) or \( e' >_L e \) when \( e \) is lexicographically smaller than \( e' \) i.e., either \( i < k \) or \( i = k \) and \( j < l \), and \( e = e' \) when they coincide. For an edge set \( A \) we use the notations \( \max\{e : e \in A\} \) and \( \min\{e : e \in A\} \) to denote the lexicographically largest and smallest edges in \( A \), respectively.

Let \( E = \{e_1 <_L e_2 <_L \cdots <_L e_m\} \) and \( E' = \{e'_1 <_L e'_2 <_L \cdots <_L e'_m\} \) be the lexicographically ordered edge lists. Then \( E \) is lexicographically smaller than \( E' \) (with respect \( <_L \)) if \( e_i <_L e'_i \) for the smallest \( i \) such that \( e_i \neq e'_i \).

**Theorem 7.8.** Let \( G_1 \) and \( G_2 \) be two \( F \)-constrained non-crossing minimally rigid graphs on a point set \( P \). Then \( G_1 \) can be transformed to \( G_2 \) by at most \( O(n^2) \) flips.

**Proof.** By Theorem 7.7, starting from \( G_1 \) we can perform flips \( O(n^2) \) times to reach an \( F \)-constrained Delaunay rigid graph, say \( G \). Let \( G^* \) be the \( F \)-constrained Delaunay rigid graph with lexicographically smallest edge list. We show that we can do edge flips from \( G \) to \( G^* \), at most \( n - 3 \) times, maintaining the non-crossing property as well as Laman’s condition.

Consider the generic rigidity matroid on the ground set \( DT(F) \). Both \( G \) and \( G^* \) are bases in this matroid, i.e., they are subgraphs of \( DT(F) \) satisfying the Laman’s condition. Delete from \( G \) the lexicographically largest edge \( e \) among \( G \setminus G^* \). By the axiom (B3) of matroids (Section 2.2.1), there will always be an edge \( e' \) in \( G^* \setminus G \) such that \( G' = G - e + e' \) is an \( F \)-constrained minimally rigid graph. \( G' \) is non-crossing since it is a subgraph of the non-crossing graph \( DT(F) \). A triangulation on \( n \) points has at most \( 3n - 6 \) edges and a minimally rigid graph has \( 2n - 3 \) edges, so after at most \( n - 3 \) such flips we reach \( G^* \).

A similar argument shows that we can start with \( G_2 \) and reach \( G^* \) in at most \( O(n^2) \) flips, completing the proof of the theorem.

### 7.4 Algorithm

Let \( \mathcal{R}(F) \) be a set of \( F \)-constrained non-crossing minimally rigid graphs on \( P \), and \( \mathcal{DR}(F) \) be a set of \( F \)-constrained Delaunay rigid graphs on \( P \). Clearly \( \mathcal{DR}(F) \subseteq \mathcal{R}(F) \). Let \( G^* \) be the \( F \)-constrained Delaunay rigid graph with the lexicographically smallest edge list as denoted in the proof of Theorem 7.8. We define the following parent function \( f : \mathcal{R}(F) \setminus \{G^*\} \to \mathcal{R}(F) \) based on Theorems 7.7 and 7.8:

**Definition 7.9.** For \( G \in \mathcal{R}(F) \) with \( G \neq G^* \), \( G' = G - f_1 + f_2 \) is the parent of \( G \), where

- **Case 1:** If \( G \in \mathcal{DR}(F) \), then \( f_1 = \max\{e : e \in G \setminus G^*\} \) and \( f_2 = \min\{e : e \in G^* \setminus G : G - f_1 + e \in \mathcal{R}(F)\} \).

- **Case 2:** If \( G \in \mathcal{R}(F) \setminus \mathcal{DR}(F) \), then \( f_1 = \max\{e : e \in G \setminus F : e \text{ is illegal in } DT(G)\} \) and \( f_2 = \min\{e : e \in DT(G - f_1) \setminus G : G - f_1 + e \in \mathcal{R}(F)\} \).

To simplify the notations, we denote the parent function depending on Case 1 and Case 2 by \( f_1 : \mathcal{DR}(F) \setminus \{G^*\} \to \mathcal{DR}(F) \) and \( f_2 : \mathcal{R}(F) \setminus \mathcal{DR}(F) \to \mathcal{R}(F) \), respectively.

The reverse search algorithm can be considered on the underlying graph in which each vertex corresponds to a non-crossing minimally rigid graph and two frameworks are adjacent.
7.4. Algorithm 149

if and only if one can be obtained from the other by a flip. Then, for $G' \in R(F)$ the local search is given by an adjacency function, Adj, defined as follows:

$$\text{Adj}(G', e_1, e_2) := \begin{cases} G' - e_1 + e_2 & \text{if } G' - e_1 + e_2 \in R(F), \\ \text{null} & \text{otherwise}, \end{cases}$$

where $e_1 \in G' \setminus F$ and $e_2 \in K_n \setminus G'$. The number of candidate edge pairs $(e_1, e_2)$ is $O(n^3)$.

Let $\text{elist}_{G'}$ and $\text{elist}_{K_n}$ be lists of edges of $G'$ and $K_n$ ordered lexicographically, and let $\text{elist}_{G'}(i)$ and $\text{elist}_{K_n}(i)$ be the $i$-th elements of $\text{elist}_{G'}$ and $\text{elist}_{K_n}$, respectively. We also denote the above defined adjacency function by $\text{Adj}(G', i, j)$ for which $e_1 = \text{elist}_{G'}(i)$ with $e_1 \notin F$ and $e_2 = \text{elist}_{K_n}(j)$ with $e_2 \notin G'$. Then, based on the algorithm in [10, 11], we describe our algorithm in Figure 7.5. An example of the search tree on a set of $F$-constrained non-crossing minimally rigid graphs on seven points are illustrated in Figure 7.6.

**Algorithm** Enumerating $F$-constrained non-crossing minimally rigid graphs.

1: $G^* :=$ a $F$-constrained Delaunay rigid graph with lexicographically smallest edge list;
2: $G' := G^*$; $i, j := 0$; Output($G'$);
3: repeat
4: do $i := i + 1$; $e_1 := \text{elist}_{G'}(i)$; while ($e_1 \in F$);
5: while $j \leq |K_n|$ do
6: do $j := j + 1$; $e_2 := \text{elist}_{K_n}(j)$; while ($e_2 \in G'$);
7: if $\text{Adj}(G', i, j) \neq \text{null}$ then
8: $G := \text{Adj}(G', i, j)$;
9: if $f_1(G) = G'$ or $f_2(G) = G'$ then
10: $G' := G$; $i, j := 0$;
11: Output($G'$);
12: go to line 4;
13: end if
14: end if
15: end while
16: end while
17: repeat
18: if $G' \neq G^*$ then
19: $G := G'$;
20: if $G \in \mathcal{DR}(F)$ then $G' := f_1(G)$;
21: else $G' := f_2(G)$;
22: determine integers pair $(i, j)$ such that $\text{Adj}(G', i, j) = G$;
23: $i := i - 1$;
24: end if
25: until $G' = G^*$ and $i = |G'|$ and $j = |K_n|$;

Figure 7.5: Algorithm for enumerating $F$-constrained non-crossing minimally rigid graphs.

As we will show later, both the parent function and the adjacency function need $O(n^2)$ time for each process. Then, the while-loop from line 4 to 17 has $|G'| \cdot |K_n|$ iterations which require $O(n^3)$ time if simply checking the lines 8 and 10. In order to improve $O(n^5)$ time to $O(n^3)$ time we claim the following two lemmas:
Lemma 7.10. Let $G$ and $G'$ be two distinct $F$-constrained Delaunay rigid graphs for which $G = \text{Adj}(G', e_1, e_2)$ for $e_1 \in G' \setminus F$ and $e_2 \in K_n \setminus G'$. Then, $f_1(G) = G'$ holds if and only if $e_1$ and $e_2$ satisfy the following conditions:

(a) $e_1 \in G^*$,
(b) $e_2 \in DT(G^*) \setminus G^*$,
(c) $e_1 \prec_L \min\{e \in G^* \setminus G' : G' - e_1 + e \in R(F)\}$,
(d) $e_2 \succ_L \max\{e : e \in G' \setminus G^*\}$.

Lemma 7.11. Let $G$ and $G'$ be two distinct $F$-constrained non-crossing minimally rigid graphs for which $G = \text{Adj}(G', e_1, e_2)$ for edges $e_1 \in G' \setminus F$ and $e_2 \in K_n \setminus G'$ with $G \in R(F) \setminus DR(F)$. Then, $f_2(G) = G'$ holds if and only if $e_1$ and $e_2$ satisfy the following conditions:

(a) $e_1$ is a legal edge in $G'$,
(b) $e_2 \in K_n \setminus DT(G')$,
(c) $e_1 \prec_L \min\{e \in DT(G') \setminus G' : G' - e_1 + e \in R(F)\}$,
(d) $e_2 = \max\{e \in (G' - e_1 + e_2) \setminus F : e \text{ is illegal in } DT(G' - e_1 + e_2)\}$.

We will explain later (in the proof of Theorem 7.13) how Lemmas 7.10 and 7.11 are used to obtain $O(n^3)$ time for generating each output of our algorithm. Notice that for $G'$ and $G \in R(F)$ such that $G = G' - e_1 + e_2$, at most one of $f_1(G) = G'$ and $f_2(G) = G'$ holds from the conditions (b) of Lemmas 7.10 and 7.11.

Proof of Lemma 7.10. (“only if”-part.) Since $f_1(G) = G'$ holds, $e_1$ and $e_2$ must be chosen as $f_2$ and $f_1$ in Case 1 of Definition 7.9. Hence $e_2(= f_1) \in G \setminus G^*$ holds. Since $G \in DR$, $G \subseteq DT(G^*)$ holds, we have (b). Similarly since $e_1(= f_2) \in G^* \setminus G \subseteq G^*$, we have (a). From $e_1 = f_2$, we have

$$G' - e_1 = (G - f_1 + f_2) - e_1 = G - f_1.$$  

(7.1)

Let $e' = \min\{e \in G^* \setminus G' : G' - e_1 + e \in R(F)\}$. Suppose (c) does not hold, and $e' \prec_L e_1$ holds. (Note that the equality does not hold since $e_1 \notin G' \setminus F$.) Then from (7.1) and $e' \prec_L e_1 = f_2 \prec_L f_1$ (which follows from Definition 7.9),

$$e' = \min\{e \in G^* \setminus (G - f_1 + f_2) : G - f_1 + e \in R(F)\} \quad \text{(by (7.1))}$$

$$= \min\{e \in G^* \setminus G : G - f_1 + e \in R(F)\} \quad \text{(by $e' \prec_L f_2 \prec_L f_1$).}$$

Thus, $e'$ would have been selected instead of $e_1$ when the parent function $f_1$ is applied to $G$, which contradicts $e_1 = f_2$. Hence, (c) holds.

Let $e'' = \max\{e : e \in G' \setminus G^*\}$, and suppose that (d) does not hold. A similar argument leads a contradiction. Thus, (d) holds.

(“if”-part.) From (a) and (b), $G = G' - e_1 + e_2$ is a $F$-constrained Delaunay rigid graph.
Since $e_1 \in G^*$ from (a),

$$e_2 \succ_L \max\{e : e \in G' \setminus G^*\} \quad \text{(by (d))}$$

$$= \max\{e : e \in (G + e_1 - e_2) - G^*\} \quad \text{(by } G = G' - e_1 + e_2\text{)}$$

$$= \max\{e : e \in (G - e_2) \setminus G^*\} \quad \text{(by } e_1 \in G^*)$$

holds. Thus, $e_2 = \max\{e : e \in G \setminus G^*\}$ holds, and hence $f_1$ chooses $e_2$ for an edge $f_1$ to be deleted from $G$. From this we obtain $G - f_1 = G' - e_1 + e_2 - f_1 = G' - e_1$. Since $e_2 \notin G^*$ from (b),

$$e_1 \prec_L \min\{e \in G^* \setminus G' : G' - e_1 + e \in \mathcal{R}(F)\} \quad \text{(by (c))}$$

$$= \min\{e \in G^* \setminus (G + e_1 - e_2) : G - f_1 + e \in \mathcal{R}(F)\} \quad \text{(by } G - f_1 = G' - e_1\text{)}$$

$$= \min\{e \in G^* \setminus (G + e_1) : G - f_1 + e \in \mathcal{R}(F)\} \quad \text{(by } e_2 \notin G^*)$$

holds. Since $e_1 \in G^* \setminus G$, we obtain $e_1 = \min\{e \in G^* \setminus G : G - f_1 + e \in \mathcal{R}(F)\}$. Thus, $f_1$ chooses $e_1$ for an edge to be added, and $f_1(G)$ returns $G'$.

**Proof of Lemma 7.11.** (“only if”-part.) Since $f_2(G) = G'$ holds, $e_1$ and $e_2$ must be chosen as $f_2$ and $f_1$ in Case 2 of Definition 7.9. As in the proof of Lemma 7.10, we have (7.1) from $e_2 = f_1$. Since $f_2 \in DT(G' - f_1) \setminus G$ holds from Definition 7.9 and from the definition of the edge-constrained Delaunay triangulation, $f_2$ is a legal edge in $DT(G - f_1)$. Hence we have

$$DT(G - e_1) = DT(G - f_1) = DT(G - f_1 + e_2) = DT(G').$$

(7.2)

Let us consider $e_2$. Since $f_1$ is illegal in $DT(G)$ from Definition 7.9, we have $f_1 = e_2 \notin DT(G - f_1)$. Therefore $e_2 \notin DT(G')$ holds from (7.2). Thus (b) holds. Also (d) must hold since the parent function removes the lexicographically largest illegal edge in $DT(G) \setminus F$.

Finally let us consider $e_1$. Let $e' = \min\{e \in DT(G') \setminus G' : G' - e_1 + e \in \mathcal{R}(F)\}$. Suppose that (c) does not hold. Then $e' \prec_L e_1$ holds. (Note that the equality does not hold since $e_1 \in G' \setminus F$.) Therefore, we have

$$e' = \min\{e \in DT(G - f_1) \setminus (G - f_1 + e_1) : G - f_1 + e \in \mathcal{R}(F)\} \quad \text{(by (7.1) and (7.2))},$$

$$= \min\{e \in DT(G - f_1) \setminus (G - f_1) : G - f_1 + e \in \mathcal{R}(F)\} \quad \text{(by } e' \prec_L e_1\text{)}.$$

Then, $e'$ would have been selected when the parent function is applied to $G$, which contradicts $e_1 = f_2$. Hence (c) holds.

(“if”-part.) From (a), $e_1$ is legal in $DT(G')$. Then, we have $DT(G') = DT(G' - e_1)$. The condition (d) says that $e_2$ is the lexicographically largest illegal edge in $DT(G) \setminus F$. Thus, $f_2$ chooses $e_2$ for an edge $f_1$ to be deleted from $G$, and $G' - e_1 = G - f_1$ holds.

From $G' - e_1 = G - f_1$ and $DT(G') = DT(G' - e_1) = DT(G - f_1)$ the condition (c) implies $e_1 \prec_L \min\{e \in DT(G - f_1) \setminus (G - f_1 + e_1) : G - f_1 + e \in \mathcal{R}(F)\}$. Thus, $e_1 = \min\{e \in DT(G - f_1) \setminus G : G - f_1 + e \in \mathcal{R}(F)\}$ (by (7.1) and (7.2)). (Note that $e_1 \in DT(G - f_1)$ and $f_1 \notin DT(G - f_1)$, because $DT(G - f_1) = DT(G')$ hold and now we have $e_1 \in DT(G')$ and $f_1 = e_2 \notin DT(G')$ from (a) and (b), respectively.) Thus, $f_2$ chooses $e_1$ for an edge to be added, and $f_2(G)$ returns $G'$.

\[\square\]
Using Lemmas 7.10 and 7.11, we now describe an \(O(n^3)\) algorithm. Before it, we give a simple observation for checking the condition (d) in Lemma 7.11 efficiently:

**Observation 7.12.** Let \(DT(F)\) be an \(F\)-constrained Delaunay triangulation constrained by a non-crossing edge set \(F\), and let \(e_1 \in F\) be a legal edge in \(DT(F)\) and \(e_2 \in K_n \setminus DT(F)\) be an edge intersecting no edge of \(F\). Then \(DT(F - e_1 + e_2) = DT(F + e_2)\) holds.

**Proof.** Since \(e_1\) is the legal edge in \(DT(F)\), we have \(DT(F) = DT(F - e_1)\). Then there exists a circle through the endpoints of \(e_1\) containing no point visible to \(e_1\) with respect to \(F - e_1\) by Proposition 7.5. When inserting \(e_2\) into \(F - e_1\), this circle clearly does not contain any point visible to \(e_1\) with respect to \(F - e_1 + e_2\). Thus \(e_1\) remains the Delaunay edge with respect to \(F - e_1 + e_2\), and this implies that \(DT(F - e_1 + e_2)\) contains \(e_1\) by Proposition 7.5 and \(DT(F + e_2) = DT(F - e_1 + e_2)\) holds. \(\square\)

**Theorem 7.13.** The set of all \(F\)-constrained non-crossing minimally rigid graphs on a given point set can be reported in \(O(n^3)\) time per output using \(O(n^2)\) space, or \(O(n^4)\) time using \(O(n)\) space.

**Proof.** As described in Section 7.3, we use a linear transformation if necessary to get a unique \(DT(F)\). The complexity of testing the uniqueness of a \(DT(F)\) is \(O(n^2)\) by simply testing the circumcircle of each triangle in the \(DT(F)\) to see there is another point other than vertices of the triangle on the circumcircle.

Given a non-crossing minimally rigid graph \(G' \in \mathcal{R}(F)\) and \(G'\)-constrained Delaunay triangulation \(DT(G')\). The algorithm will check \(f_1(\text{Adj}(G', e_1, e_2)) = G'\) or \(f_2(\text{Adj}(G', e_1, e_2)) = G'\) at line 10 depending on the edge pair \((e_1, e_2)\). Here we will show that each condition in Lemmas 7.10 and 7.11 can be checked in \(O(1)\) time for each of the \(O(n^3)\) edge pairs \((e_1, e_2)\) by the following way.

First, for all edges \(e_2 \in \text{elist}_{K_n}\), we calculate the number of edges \(e_1 \in G'\) intersecting \(e_2\), which we denote by \#\text{cross}(e_2, G'). If \#\text{cross}(e_2, G') > 1, we delete \(e_2\) from \text{elist}_{K_n} since \(G' - e_1 + e_2\) is never non-crossing for any \(e_1 \in \text{elist}_{G'}\). If \#\text{cross}(e_2, G') = 1, we associate a pointer of the edge \(e_1\) intersecting \(e_2\) with \(e_2\), and we denote such \(e_1\) by \text{cross}(e_2, G').

Next, for each \(e_1 \in \text{elist}_{G'}\), we attach two flags to \(e_1\) which represent that \(e_1\) satisfies the conditions (a) of Lemmas 7.10 and 7.11, respectively. These help us to check the conditions (a) in Lemmas 7.10 and 7.11 in \(O(1)\) time. Similarly, we attach two flags to \(e_2 \in \text{elist}_{K_n}\) which represent that \(e_2\) satisfies the conditions (b) of Lemmas 7.10 and 7.11 to check them in \(O(1)\) time for each \(e_2\). Additionally we calculate the lexicographically largest edge in \(G' \setminus G^*\) in \(O(n)\) time to check the condition (d) in Lemma 7.10 in \(O(1)\) time for each \(e_2\). Apparently, these preprocessing can be done in \(O(n^2)\) time using the precomputed sorted edge list of \(G^*\) and \(DT(G)\).

Now let us consider how to identify a set of edges \(e_2 \in \text{elist}_{G'}\) satisfying the condition (d) in Lemma 7.11 in \(O(n^3)\) time with \(O(n^2)\) space. (In the case of \(O(n^4)\) time algorithm this process must be skipped, and the condition (d) in Lemma 7.11 will be checked simply by updating \(DT(G')\) to \(DT(G' - e_1 + e_2)\) for each pair \((e_1, e_2)\) using \(O(n)\) time and \(O(n)\) space by applying
the algorithm by Chin and Wang [25].) It can be done regardless of the removing edge $e_1$ when the condition (a) in Lemma 7.11 is satisfied. If $\#\text{cross}(e_2, G') = 0$, by Observation 7.12, we can say that the condition (d) holds if and only if $e_2$ is the lexicographically largest illegal edge among $DT(G + e_2) \setminus F$. On the other hand, if $\#\text{cross}(e_2, G') = 1$, then it is sufficient to check the condition (d) only in $DT(G - \text{cross}(e_2, G') + e_2)$. In both cases, we can check whether $e_2$ satisfies the condition (d) of Lemma 7.11 since updating the Delaunay triangulation takes $O(n)$ time (see [7, 25, 34] for a linear time update of the constrained Delaunay triangulation). Thus we can attach a flag to each $e_2 \in \text{elist}_{K_n}$ which represents whether $e_2$ satisfies (d) or not in $O(n^3)$ time.

By using the above mentioned data, we will show that for a fixed $e_1 \in \text{elist}_{G'}$, the inner while-loop from line 6 to 16 can be executed in $O(n^2)$. In order to efficiently test the condition (c) of Lemmas 7.10 and 7.11, we prepare the data structure for maintaining rigid components (i.e., maximal rigid subgraphs) of $G - e_1$ in order to determine whether two vertices are spanned by a common rigid component in $O(1)$ time using $O(n^2)$ preprocessing time with $O(n^2)$ space, or $O(n)$ time using $O(n^2)$ preprocessing time with $O(n)$ space (see Lee and Streinu [80]). From this, we can calculate Adj($G', e_1, e_2$) (i.e., determine whether $G' - e_1 + e_2 \in \mathcal{L}(F)$) in $O(1)$ time with $O(n^2)$ space, or $O(n)$ time with $O(n)$ space, for each edge $e_2 \in \text{elist}_{K_n}$. Also, we can compute $e' = \min\{e \in G'^* \setminus G' : G' - e_1 + e \in \mathcal{R}(F)\}$ and $e'' = \min\{e \in DT(G') \setminus G' : G' - e_1 + e \in \mathcal{R}(F)\}$ in $O(n)$ time with $O(n^2)$ space, or $O(n^2)$ time with $O(n)$ space. Using $e'$ and $e''$ we can check condition (c) of Lemmas 7.10 and 7.11 in $O(1)$ time.

Thus, we have confirmed that all conditions of Lemmas 7.10 and 7.11 can be checked in $O(1)$ time for each pair $(e_1, e_2)$ by taking $O(n^3)$ preprocessing time with $O(n^2)$ space, or $O(n)$ time for each $(e_1, e_2)$ with $O(n)$ space.

By using the above mentioned data structure for maintaining the rigid components, we can perform both parent function and adjacency function in $O(n^2)$ time with $O(n^2)$ space, or $O(n^3)$ time with $O(n)$ space. Thus, we obtain an $O(n^3)$ time algorithm using $O(n^2)$ space, or $O(n^4)$ time algorithm using $O(n)$ space. □

7.5 Generating Bistable Compliant Mechanisms

In this section we shall present a result of [91] which shows a computational result of our enumeration algorithm applied to the problem of finding bar-and-joint bistable compliant mechanisms (explained in Section 1.2.1).

We consider the problem instance of the node set shown in Figure 7.7, where $W = 0.2m$ and $H = 0.1m$. This is indeed a problem of finding bistable grippers; the gap between the nodes B and C is controlled to grip a workpiece by the forced displacement in $x$-direction at the input node A (in addition it is required to have two self-equilibrium states at both deformed and undeformed states).

Members 1–4 are considered as gripping arms and excluded from the design region, and
isostatic frameworks are enumerated within the convex design region (within the dotted lines) of Figure 7.8(a). The four thick dotted lines in Figure 7.8(b) are assumed to be necessary members. If we do not specify the four necessary members, the number of isostatic frameworks enumerated by our algorithm is 1027992, and the CPU time\(^1\) for enumeration is 34667s. If we specify the necessary members to reduce the candidate topologies as shown in Figure 7.8(b), the number of isostatic frameworks and CPU time are drastically reduced to 68072 and 2138s, respectively. (In fact, because of the singularity of the node set, the total number of isostatic frameworks is rather small. Although some frameworks enumerated here may not be infinitesimally rigid, such a framework surely becomes rigid by slightly perturbing its configuration. The perturbation is performed in the next nonlinear programming step.)

In the second step we carry out the nonlinear optimization to check whether each (enumerated) framework possesses a desired function. Refer to [91] for more detailed description on this step. Among the 68072 initial solutions, we found 156 different types of compliant mechanisms. The number of iterations in the nonlinear optimization for the converged solution is about 20, whereas it is more than 50 in the topology optimization approach in Ohsaki and Nishiwaki [92]. Furthermore, only three solutions were found from randomly generated 100 initial solutions in [92]. Therefore, the computational cost has been significantly reduced by the proposed method.

Figure 7.9 shows a typical solution, where the triangle “abc” realizes the so-called snap-through behavior for the dynamic deformation.

### 7.6 Conclusion

We have presented an algorithm for enumerating all the constrained non-crossing minimally rigid graphs. We note in passing that the techniques in this paper can also be used to generate all \(F\)-constrained non-crossing spanning trees of a point set since they also form bases of the graphic matroid on any triangulation of \(P\). The unconstrained case was considered in [2, 11].

An open problem, that is of considerable practical importance, is to generate efficiently all non-crossing minimally rigid graphs that do not contain any edge from a given set. This is equivalent to generating all non-crossing minimally rigid graphs that are subgraphs of a given geometric graph. An indication that this problem may be challenging, is Theorem 7.2 which says that determining if a geometric graph contains a non-crossing minimally rigid graphs is NP-complete.

\(^1\)We used a PC with Intel Xeon 3.4 GHz CPU and 2GB RAM.
Figure 7.6: An example of the search tree of our algorithm on seven points. The constraint edges $F$ are illustrated by the bold edges, and each edge of the search tree is distinguished by using the dotted or bold line according to whether it corresponds to Case 1 or Case 2 of Definition 7.9.

Figure 7.7: Initial set of nodes and supports for generating mechanisms.
Figure 7.8: Design region: (a) a set of nodes with no member specified, and (b) a set of nodes with predetermined members.

Figure 7.9: An obtained mechanism.
Chapter 8

Enumerating Non-crossing Geometric Graphs

In Chapter 7 we have proposed an efficient algorithm for enumerating (edge-constrained) non-crossing minimally rigid graphs with the help of edge-constrained triangulations. In this chapter, we shall generalize the idea of Chapter 7 to arbitrary non-crossing geometric graph classes.

Enumerating non-crossing geometric graphs on a given point set is a fundamental problem in computational geometry, and several algorithms have been proposed. Our new enumeration technique provides faster algorithms for various enumeration problems compared with existing ones, such as those for plane straight-line graphs, non-crossing connected graphs, non-crossing spanning trees, and non-crossing minimally rigid graphs. Moreover, since the idea is quite simple, it can be applied possibly to many other enumeration problems, for which enumeration algorithms have not been known to the best of our knowledge.

Also, combining our enumeration technique with Theorem 5.2, we obtain an efficient algorithm for enumerating all non-crossing $k$-dof mechanisms connecting given joints, some of which are connected with external environment by a set of sliders.

8.1 Introduction

Throughout this section we assume that a given point set $P$ is fixed in $\mathbb{R}^2$ and an embedding $V \to P$ is given. Since a graph class is defined in terms of the properties that all its members share, imposing the additional “non-crossing” requirement to an existing graph class, we can define a non-crossing geometric graph class on $P$, such as non-crossing spanning trees or non-crossing perfect matchings. Let us denote by $\mathcal{NGG}$ a specific non-crossing geometric graph class. We shall extensively study the following enumeration problem:

Input: A point set $P$ in the plane with $n$ points.

Output: The list of all the non-crossing geometric graphs belonging to $\mathcal{NGG}$ on $P$. 

157
**Contribution.** We present a new general technique for enumerating non-crossing geometric graphs. Our technique provides faster algorithms for various enumeration problems compared with existing ones, such as those for *plane straight-line graphs*, *non-crossing spanning connected graphs*, *non-crossing spanning trees*, and *non-crossing minimally rigid graphs*. Moreover, since the idea is quite simple, it can be applied to many enumeration problems, for which enumeration algorithms were not known to the best of our knowledge, such as *non-crossing matchings*, *non-crossing red-and-blue matchings*, *non-crossing k-vertex or k-edge connected graphs* or *non-crossing directed geometric graphs*. In Table 8.1 we list the time complexities of (a part of) new algorithms obtained in this chapter, where we use the following notation to denote the numbers of graphs on a point set $P$: $\text{pg}(P)$ for plane straight-line graphs, $\text{cg}(P)$ for non-crossing spanning connected graphs, $\text{st}(P)$ for non-crossing spanning trees, $\text{mrg}(P)$ for non-crossing minimally rigid graphs, $\text{tri}(P)$ for triangulations, and $\text{pm}(P)$ for non-crossing perfect matchings.

**General idea.** The key idea of our technique is to use triangulations as in Chapter 7. Let us consider enumerating all non-crossing spanning trees for example. Since every subgraph of a triangulation is non-crossing, enumerating all non-crossing spanning trees in a triangulation is easily done by applying algorithms such as [70, 109] developed for enumerating all spanning trees in a given (abstract) graph. Moreover, efficient enumeration algorithms for triangulations are already known [11, 20]. Therefore, by enumerating spanning trees in every triangulation, we will obtain all non-crossing spanning trees since every non-crossing spanning tree is a subgraph of some triangulation. However, some non-crossing spanning tree might be produced more than once since it could be a subgraph of more than one triangulation.

In order to avoid duplicate enumeration we introduce two key notions: *edge-constrained lexicographically largest triangulations* and *minimal representative sets*. Recall that a geometric graph containing a non-crossing edge set $F$ is called $F$-constrained. We show that, for each triangulation $T$, there exists the inclusionwise minimum non-crossing edge set $F^*$, called the minimal representative set, such that $T$ is the $F^*$-constrained lexicographically largest triangulation (that is the triangulation of the lexicographically largest edge list among all $F^*$-constrained triangulations with respect to a certain order on edges defined later). In the enumeration algorithm proposed in this chapter, every time a new triangulation $T$ is obtained, we will compute the minimal representative set $F^*$ of $T$ and then enumerate all spanning trees that are contained in $T$ and contain $F^*$. We will show that this algorithm correctly enumerates all non-crossing spanning trees without repetitions.

The overall idea of our techniques is described in two algorithms, Algorithm 1 and Algorithm 2, in Section 8.3 and Section 8.4, respectively. Let $\text{ngg}(P)$ be the total number of graphs of $\mathcal{NGG}$ to be enumerated. Then, Algorithm 1 enumerates all the non-crossing geometric graphs belonging to $\mathcal{NGG}$ without repetitions in $O(f(n) \cdot \text{tri}(P) + g(n) \cdot \text{ngg}(P))$ time, where $f$ is a polynomial function, and $g$ is a function of $n$ depending on $\mathcal{NGG}$. In the graph classes considered in this chapter, $g$ is also a polynomial function. By applying Algorithm 1 we obtain new algorithms for enumerating plane straight-line graphs, non-crossing spanning...
Table 8.1: Time complexities of new algorithms and previous ones for non-crossing geometric graphs.

<table>
<thead>
<tr>
<th></th>
<th>New results</th>
<th>Previous best</th>
</tr>
</thead>
<tbody>
<tr>
<td>plane straight-line graphs</td>
<td>$O(pg(P))$</td>
<td>$O(n \log n \cdot pg(P))$ [2]</td>
</tr>
<tr>
<td>connected graphs</td>
<td>$O(cg(P))$</td>
<td>$O(n \log n \cdot cg(P))$ [2]</td>
</tr>
<tr>
<td>spanning trees</td>
<td>$O(n \cdot tri(P) + st(P))$</td>
<td>$O(n \log n \cdot st(P))$ [2]</td>
</tr>
<tr>
<td>minimally rigid graphs</td>
<td>$O(n^2 \cdot mrg(P))$</td>
<td>$O(n^3 \cdot mrg(P))$ (Chap. 7)</td>
</tr>
<tr>
<td>perfect matchings</td>
<td>$O(n^{3/2} \cdot tri(P) + n^{5/2} pm(P))$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

connected graphs, non-crossing spanning trees, and non-crossing perfect matchings (see Table 8.1). In particular, for plane straight-line graphs or non-crossing spanning connected graphs, we show that $pg(P)$ and $cg(P)$ are exponentially larger than $tri(P)$ for every point set $P$, and thus the term of $f(n) \cdot tri(P)$ is dominated by $pg(P)$ or $cg(P)$. Consequently, our algorithms work in $g(n)$ time on average, which will be shown to be constant. These results improve the running time of the previous best ones by Aichholzer et al. [2].

Although Algorithm 1 enumerates all graphs of $NGG$ efficiently in terms of $tri(P)$ and $ngg(P)$, its time complexity cannot be bounded by $O(g(n) \cdot ngg(P))$ in general since its complexity is dominated by $tri(P)$ when $tri(P)$ is much larger than $ngg(P)$. The next proposed algorithm Algorithm 2 overcomes this drawback by avoiding the enumeration of the triangulations $T$ such that any $F^*$-constrained geometric graph does not belong to $NGG$ for the minimal representative set $F^*$ of $T$. Applying Algorithm 2, we obtain an enumeration algorithm for non-crossing minimally rigid graphs that works in $O(n^2)$ time on average. This result improves the one given in Chapter 7 by an $O(n)$ factor on average.

Related works. As for related work of our result, Razen and Welzl [98, 123] recently showed a relatively similar approach for counting the total number of planar straight-line graphs on a given point set, where he proposed the method of using the edge-constrained Delaunay triangulation and illegal edges (Lawson edges), which in our context correspond to the edge-constrained lexicographically largest triangulation and the minimal representative set, respectively. We remark that our work has been done independently from it. In addition, as the current fastest enumeration algorithm [20] for triangulations is based on the lexicographical ordering on the edge set, there are some advantages of using the lexicographically largest triangulation over the Delaunay triangulation especially in the time complexity analysis (e.g., a simple amortized analysis of the edge insertion algorithm given in Section 8.4.2). Also, the concept of the lexicographically largest triangulation enables us to prove a nontrivial lower bound on the number of non-crossing spanning connected graphs, which will be given in Theorem 8.16.

Recall the branch-and-bound enumeration technique explained in Section 7.2.1 which provides us with polynomial-time enumeration algorithms for many graph classes because it just requires a polynomial-time oracle that checks whether a given graph contains at least one subgraph belonging to a certain graph class. However, the problem of detecting a non-crossing
subgraph in a given geometric graph is known to be NP-hard for most graph classes [67]. For this reason, most of the enumeration problems for non-crossing geometric graphs become nontrivial and we need to introduce some new technique. In fact, all previous works for the enumeration of non-crossing geometric graphs are based not on the branch-and-bound technique but on sophisticated local transformations discussed below.

Two objects of \( \mathcal{NGG} \) are connected if they can be transformed into each other by a transformation which generates one graph from the other by a certain specified operation. Define a graph \( \mathcal{G}_{\mathcal{NGG}} \) on \( \mathcal{NGG} \) where the vertex set of \( \mathcal{G}_{\mathcal{NGG}} \) corresponds to the set of all objects of \( \mathcal{NGG} \) and two vertices are connected by an edge if the corresponding graphs of \( \mathcal{NGG} \) are connected. Then, the natural question is how we can design a transformation so that \( \mathcal{G}_{\mathcal{NGG}} \) is a connected graph. Moreover, from the viewpoint of the applications to enumeration problems, a transformation should be defined locally, i.e., the symmetric difference between two connected objects should be as small as possible. Developing such a nice transformation might be interesting in its own right, and there are many known results not only for local transformations [2, 5, 11, 50, 51, 54] but also for large transformations [1, 3, 53]. Almost all previous works for the enumerations of non-crossing geometric graphs discussed above are based on local transformations, and thereby they deeply rely on the property of a particular graph class. On the other hand, the proposed technique reveals that efficient enumeration of \( \mathcal{NGG} \) is possible without defining a local transformation of \( \mathcal{NGG} \) explicitly.

Organization. We introduce the \( F \)-constrained lexicographically largest triangulation in Section 8.2, and prove some new results on this special triangulation. In Section 8.3 we propose our general technique for enumerating all graphs of \( \mathcal{NGG} \), and then in Section 8.3.3 we show how to apply this technique to several specific problems. In Section 8.4 we present a further extended technique. Lastly, in Section 8.5, we briefly discuss the other applications.

8.2 Lexicographically Ordered Edge-constrained Triangulations

In this section we introduce the \( F \)-constrained lexicographically largest triangulation (F-CLLT) on \( P \), and then we show that every \( F \)-constrained triangulation can be transformed into the F-CLLT by \( O(n^2) \) flips. We remark again that F-CLLT is derived from the lexicographically ordered triangulation developed by Bespamyatnikh [20] although he did not extend his result to the edge-constrained case.

8.2.1 Notation

We assume that \( x \)-coordinates of all points of \( P \) are distinct and no three points of \( P \) are colinear. We label the points of \( P \) as \( p_1, \ldots, p_n \) in the increasing order of \( x \)-coordinates. For two vertices \( p_i, p_j \in P \), we denote \( p_i < p_j \) if \( i < j \) holds. Considering \( p_i \in P \), we often pay our attention only to its right point set, \( \{p_{i+1}, \ldots, p_n\} \subseteq P \), which is denote by \( P_{i+1} \).

As in Chapter 7, let \( K_n \) be the complete graph embedded on \( P \) (with the straight line segments), and the line segment between \( p_i \) and \( p_j \) with \( p_i < p_j \) is called an edge between
Let shaded. the following fact:

\[ x \text{ to the empty region } F \text{ respect to } p \]

defined as those from \( p_i \) and \( p_j \) moving along the line from \( p_i \) to \( p_j \) by \( \Delta(p_i, p_j, p_k) > 0 \) (or \( \Delta(p_i, p_j, p_k) < 0 \), respectively). A total ordering \( < \) on the set of edges is defined as follows: for \( e = (p_i, p_j) \) and \( e' = (p_k, p_l) \) (with \( p_i < p_j \) and \( p_k < p_l \), \( e \) is smaller than \( e' \), denoted by \( e \prec e' \)), if and only if \( p_i < p_k \), or \( p_i = p_k \) and \( \Delta(p_i, p_j, p_k) < 0 \). Note that, when \( p_i = p_k \), this ordering corresponds to the clockwise ordering around \( p_i \). Let \( E = \{e_1, \ldots, e_m\} \) and \( E' = \{e'_1, \ldots, e'_m\} \) be two sorted edge lists with \( e_1 \prec \cdots \prec e_m \) and \( e'_1 \prec \cdots \prec e'_m \). Then, \( E' \) is lexicographically larger (with respect to \( \prec \)) than \( E \) if \( e_i \prec e'_i \) for the smallest \( i \) such that \( e_i \neq e'_i \).

We say that two edges \((p_i, p_j)\) and \((p_k, p_l)\) properly intersect each other if \((p_i, p_j)\) and \((p_k, p_l)\) have a point in common except for their endpoints. Let \( F \) be a non-crossing edge set on \( P \). For two points \( p_i, p_j \in P, p_j \) is visible from \( p_i \) with respect to \( F \) when \((p_i, p_j)\) does not properly intersect any edge of \( F \). We assume that \( p_j \) is visible from \( p_i \) if \((p_i, p_j) \in F \).

The upper tangent \((p_i, p_i^{up})\) and the lower tangent \((p_i, p_i^{low})\) of \( p_i \) with respect to \( F \) are defined as those from \( p_i \) to the convex hull of the points of \( P_{i+1} \) that are visible from \( p_i \) with respect to \( F \) (see Figure 8.1). Notice that each of the upper and lower tangents defines an empty region in which no point of \( P \) exists as described below. Let \( l \) be the line perpendicular to the \( x \)-axis passing through \( p_i \), and let \( e_1 \) and \( e_2 \) be the closest edges from \( p_i \) among \( F \) intersecting with \( l \) in the upper and lower sides of \( p_i \), respectively (if such edge exists). Then there exists no point of \( P \) inside the region bounded by \( l, e_1 \) (resp. \( e_2 \)) and the line passing through \( p_i \) and \( p_i^{up} \) (resp. \( p_i^{low} \)). When \( e_1 \) (resp. \( e_2 \)) does not exist, the empty region is defined by the one bounded by \( l \) and the line through \( p_i \) and \( p_i^{up} \) (resp. \( p_i^{low} \)). Thus, we have the following fact:

**Lemma 8.1.** Let \( F \) be a non-crossing edge set on \( P \). Then, for any edge \( e \in K_\eta \) that properly intersects the upper or lower tangent of \( p_i \) with respect to \( F \), at least one of the following two facts holds: (1) the left endpoint of \( e \) is less than \( p_i \) or (2) \( e \) properly intersects some edge of \( F \).

![Figure 8.1: An example of the upper and lower tangents, denoted by \((p_i, p_i^{up})\) and \((p_i, p_i^{low})\), respectively. The bold edges represent \( F \). The empty regions of \((p_i, p_i^{up})\) and \((p_i, p_i^{low})\) are shaded.](image-url)
Enumerating Non-crossing Geometric Graphs

For \( p_i \in P \) and a geometric graph \( G \) on \( P \), let us denote by \( \delta_G(p_i) \) the set of edges of \( G \) which are incident to \( p_i \) with the left endpoints. Similarly, for an edge set \( F \) on \( P \), \( \delta_F(p_i) \) denotes the set of edges of \( F \) which are incident to \( p_i \) with the left endpoints. Let us consider the following construction of the \( F \)-constrained geometric graph on \( P \) for a non-crossing edge set \( F \):

**Construction 1.**

0. Repeat the following process for all \( p_i \in P \) in an arbitrary order.

1. Let \( (p_i, p_i^{up}) \) and \( (p_i, p_i^{low}) \) be the upper and lower tangents of \( p_i \in P \) with respect to \( F \), and denote the right endpoints of \( \delta_F(p_i) \cup \{(p_i, p_i^{up}), (p_i, p_i^{low})\} \) by \( p_{i0}, p_{i1}, \ldots, p_{im} \) arranged in clockwise order around \( p_i \) (where \( p_{i0} = p_i^{up} \) and \( p_{im} = p_i^{low} \) hold) (Figure 8.2(a)).

2. Consider the cone \( C_k \) with apex \( p_i \) bounded by two consecutive edges \( (p_i, p_{ik}) \) and \( (p_i, p_{ik+1}) \) for each \( k \) with \( 0 \leq k \leq m - 1 \), where \( C_k \) contains both \( p_{ik} \) and \( p_{ik+1} \), and construct the convex hull \( H_k \) of \( P_{i+1} \cap C_k \) inside each \( C_k \) (Figure 8.2(b)).

3. Draw an edge from \( p_i \) to every point \( p_j \in P_{i+1} \cap C_k \) if \( p_j = (p_i, p_j) \cap H_k \) holds for some \( k \) (Figure 8.2(c)).

We give an example of the graph obtained by Construction 1 in Figure 8.3. Notice that the graph obtained by Construction 1 always has the edges of \( \delta_F(p_i) \cup \{(p_i, p_i^{up}), (p_i, p_i^{low})\} \) for all \( p_i \in P \). In addition, the following property can be easily observed:
8.2. Lexicographically Ordered Edge-constrained Triangulations

Lemma 8.2. Let $F$ be a non-crossing edge set on a given point set $P$. Let $G$ be the graph obtained by Construction 1 for $F$, and let $(p_i, p_j)$ be an edge of $G$. Then, any edge of $K_n$ properly intersecting $(p_i, p_j)$ also properly intersects at least one edge of $\delta_F(p_i) \cup \{(p_i, p_i^{up}), (p_i, p_i^{low})\}$.

Proof. Let us consider Construction 1 around $p_i$. Then, there exists a convex hull $H_k$ for which $p_j = (p_i, p_j) \cap H_k$ from the definition of Construction 1. Notice that the two consecutive edges, $(p_i, p_{ik})$ and $(p_i, p_{ik+1})$ of $\delta_F(p_i) \cup \{(p_i, p_i^{up}), (p_i, p_i^{low})\}$ (bounding $C_k$ considered in Step 2), and the part of the boundary of $H_k$ from $p_{ik}$ to $p_{ik+1}$ (that is a convex chain) forms a simple polygon with exactly three convex vertices, $p_i, p_{ik}$ and $p_{ik+1}$, which is a so-called pseudo-triangle. Recall that $p_j$ is a vertex of the pseudo-triangle because $p_j = (p_i, p_j) \cap H_k$. Since there exists no point of $P$ inside of the pseudo-triangle, any edge properly intersecting $(p_i, p_j)$ must properly intersect at least one of $(p_i, p_{ik})$ and $(p_i, p_{ik+1})$.

The following lemmas describe the fundamental properties of the above defined construction:

Lemma 8.3. The graph $G$ obtained by Construction 1 is a triangulation on $P$.

Proof. We will prove, by induction on $i$ from $i = n$ to 1, that (1) the subgraph of $G$ induced by $P_i$, denoted by $G_i$, is non-crossing, and (2) all faces of $G_i$ are triangles except possibly for the outer face. This implies that $G$ is a triangulation since $G$ clearly contains the boundary edges of the convex hull of $P$ from the definition of Construction 1.

For the basis, $G_n$ has no edge, and hence the statement holds. Assume that (1) and (2) hold for $G_{i+1}$. We first show that (1) holds for $G_i$. Suppose there exists an edge $(p_a, p_b) \in G_{i+1}$ with $p_a < p_b$ that properly intersects some edge of $G_i \setminus G_{i+1}$. Then, from Lemma 8.2, $(p_a, p_b)$ properly intersects some edge of $\delta_F(p_i) \cup \{(p_i, p_i^{up}), (p_i, p_i^{low})\}$. By Construction 1 it is obvious that $(p_a, p_b)$ does not properly intersect any edge of $F$. Hence $(p_a, p_b)$ properly intersects $(p_i, p_i^{up})$ or $(p_i, p_i^{low})$. However, by Lemma 8.1, this implies $p_a < p_i$, which contradicts $p_a \in P_{i+1}$.

Let us prove (2). Let $(p_i, p_a)$ and $(p_i, p_b)$ be two consecutive edges of $G_i \setminus G_{i+1}$ in clockwise order around $p_i$. From the definition of Construction 1, there exists the convex hull $H_k$ such that $p_a$ and $p_b$ are consecutive vertices on the boundary of $H_k$. Hence, the edge between $p_a$ and $p_b$ is the upper or lower tangent of $p_a$ or $p_b$ with respect to $F$, and thus it is contained in $G_{i+1}$ by Construction 1. Moreover, from the definition of $H_k$, the triangle face $(p_i, p_a, p_b)$ contains no point of $P$, and thus (2) follows. As a result, $G$ is an $F$-constrained triangulation on $P$.

Lemma 8.4. The $F$-constrained triangulation $T^*(F)$ obtained by Construction 1 has the lexicographically largest edge list among all the $F$-constrained triangulations on $P$.

Proof. Let us denote the edges of $T^*(F)$ by $\{e_1^*, \ldots, e_m^*\}$ with $e_1^* \prec \cdots \prec e_m^*$. Suppose there exists an $F$-constrained triangulation $T$ whose edge set $\{e_1, \ldots, e_m\}$ with $e_1 \prec \cdots \prec e_m$ is lexicographically larger than that of $T^*(F)$. Then, there exists the smallest label $s$ with $e_s^* \neq e_s$ for which $e_s^* \notin T$ and $e_s^* \prec e_s$ hold.
Let \( e_i^* = (p_i, p_j) \in T^*(F) \setminus T \). Since \( s \) is the smallest label among the edges \( e_i \) for which \( e_i^* \neq e_i \), \( \delta_{T^*(F)}(p) = \delta_T(p) \) holds for every \( p \in \{p_1, \ldots, p_{i-1}\} \). Since \( T \) is a triangulation but does not contain \( e_i^* \), \( T \) must contain at least one edge \( e \notin T^*(F) \) that properly intersects \( e_i^* \).

By Lemma 8.2, \( e \) properly intersects some edge of \( \delta_F(p_i) \cup \{(p_i, p_i^\text{up}), (p_i, p_i^\text{low})\} \). In addition, since \( T \) is an \( F \)-constrained triangulation, \( e \) does not properly intersect any edge of \( \delta_F(p_i) \), and consequently \( e \) properly intersects at least \( (p_i, p_i^\text{up}) \) or \( (p_i, p_i^\text{low}) \). Lemma 8.1 hence implies that the left endpoint of \( e \) is on the left side of \( p_i \), which contradicts \( \delta_{T^*(F)}(p) = \delta_T(p) \) for \( p \in \{p_1, \ldots, p_{i-1}\} \).

Hence, we call the \( F \)-constrained triangulation obtained by the above construction the \textit{\( F \)-constrained lexicographically largest triangulation (\( F \)-CLLT)}. In fact we can show that the \( F \)-CLLT can be constructed by the greedily adding the edges to \( F \) in the descending edge ordering without violating the non-crossing property.

### 8.2.3 Improving Flips

Let \( T^*(F) \) denote the \( F \)-CLLT on \( P \). For any \( F \)-constrained triangulation \( T \) with \( T \neq T^*(F) \), the \textit{critical vertex} of \( T \) is the vertex having the smallest label among those incident to some edge in \( T \setminus T^*(F) \).

For an edge \( e \) with \( e \in T \setminus F \), \( e \) is called \textit{flippable} if two triangles incident to \( e \) in \( T \) form a \textit{convex} quadrilateral \( Q \). Flipping \( e \) in \( T \) generates a new \( F \)-constrained triangulation by replacing \( e \) with the other diagonal of \( Q \). Such an operation is called an \textit{improving flip} if the triangulation obtained by flipping \( e \) is lexicographically larger than the previous one, and \( e \) is called \textit{improving flippable}.

**Lemma 8.5.** Let \( T \) be an \( F \)-constrained triangulation with \( T \neq T^*(F) \) and \( p_c \) be the critical vertex of \( T \). Then, there exists at least one improving flippable edge incident to \( p_c \) in \( T \setminus T^*(F) \).

**Proof.** Let \( (p_c, p_c^\text{up}) \) and \( (p_c, p_c^\text{low}) \) be the upper and lower tangents of \( p_c \) with respect to \( F \). We shall first show \( \delta_{T^*(F)}(p_c) \subset T \).

It is obvious that \( T \) contains every edge of \( \delta_F(p_c) \) because \( T \) is an \( F \)-constrained triangulation. Let us show \( (p_c, p_c^\text{up}) \in T \). Suppose \( (p_c, p_c^\text{up}) \) is missing in \( T \). Then, \( T \) has some edge \( (p_a, p_b) \notin T^*(F) \) that properly intersects \( (p_c, p_c^\text{up}) \) since \( T \) is a triangulation. Moreover, by Lemma 8.1, \( p_a < p_c \) holds, implying that \( p_a \) is incident to an edge not in \( T^*(F) \) and contradicting that \( p_c \) is the critical vertex of \( T \). Thus, \( (p_c, p_c^\text{up}) \in T \). The same argument can apply to \( (p_c, p_c^\text{low}) \).

Suppose that an edge \( (p_c, p) \) of \( \delta_{T^*(F)}(p_c) \) is missing in \( T \). There exists some edge \( e \in T \setminus T^*(F) \) that properly intersects \( (p_c, p) \). Lemma 8.2 now implies that \( e \) also properly intersects some edge of \( \delta_F(p_c) \cup \{(p_c, p_c^\text{up}), (p_c, p_c^\text{low})\} \), contradicting that \( T \) contains all the edges of \( \delta_F(p_c) \cup \{(p_c, p_c^\text{up}), (p_c, p_c^\text{low})\} \). Therefore, we have \( \delta_{T^*(F)}(p_c) \subset T \).

Now let us show that there exists at least one improving flippable edge incident to \( p_c \). Since \( p_c \) is the critical vertex, there exists an edge \( e \in T \) incident to \( p_c \) with \( e \notin T^*(F) \). Let \( (p_c, p_{c+1}) \) and \( (p_c, p_{c+1}) \) be two consecutive edges of \( \delta_{T^*(F)}(p_c)(\subset T) \) around \( p_c \) such that...
Let constrained triangulation. This task can be done by using the algorithm for the unconstrained efficiency of finding the children of each object; in our case finding the children of each $F$ to the flips, the label of the critical vertex increases by at least one. Therefore, the improving flip never decreases the label of the critical vertex. Hence, after $O(T)$ such edge reduces the number of edges of $p$ incident to $p$, where $q_0 = p_{c_{k'}}$ and $q_l = p_{c_{k'+1}}$. Then, $(p_c, q_j) \in T \setminus T^*(F)$ holds for all $j = 1, \ldots, l - 1$, and moreover any of $q_1, q_2, \ldots, q_{l-1}$ is not inside of the triangle $p_c p_{c_k} p_{c_{k'+1}}$ since $T^*(F)$ has the empty triangle face $p_c p_{c_k} p_{c_{k'+1}}$. Therefore, every $(p_c, q_j)$ properly intersects the line segment between $p_{c_{k'}}$ and $p_{c_{k'+1}}$. Let $q_j^*$ be the vertex furthest from the line passing through $p_{c_k}$ and $p_{c_{k'+1}}$ among $q_j$. Then, the quadrilateral $p_c q_{j-1}^* q_j^* q_{j+1}^*$ is convex because $q_{j-1}^*$, $q_{j+1}^*$ and $q_{j+1}$ are not colinear, and flipping $e^* = (p_c, q_{j+1}^*)$ produces a lexicographically larger triangulation than $T$ because $p_c < q_{j-1}^*$ and $p_c < q_{j+1}^*$.

**Theorem 8.6.** Let $P$ be a set of $n$ points in the plane. Every $F$-constrained triangulation $T$ on $P$ can be transformed to the $F$-CLLT on $P$ by $O(n^2)$ improving flips.

*Proof.* From Lemma 8.5, $T(\neq T^*(F))$ always has an improving flippable edge, and flipping such edge reduces the number of edges of $T \setminus T^*(F)$ incident to the critical vertex $p_c$. Moreover, the improving flip never decreases the label of the critical vertex. Hence, after $O(n)$ improving flips, the label of the critical vertex increases by at least one. Therefore, $T$ can be transformed to the $F$-CLLT by $O(n^2)$ improving flips.

The rest of this section describes the enumeration of the $F$-constrained triangulations on $P$ based on the reverse search (see Section 7.2.2). As we have proved that the lexicographical order of the (unconstrained) triangulations can be naturally extended to the edge-constrained case, the algorithm for the unconstrained case by Bespamyatnikh [20] that is based on the lexicographical order of unconstrained triangulations can be also extended to the edge-constrained case. For every $F$-constrained triangulation $T$ with $T \neq T^*(F)$, let us define the parent of $T$ as the triangulation obtained by flipping the smallest improving flippable edge among $T \setminus T^*(F)$ with respect to the edge ordering $\prec$, which surely exists by Lemma 8.5. Then, due to the correctness of Theorem 8.6, these parent-child relations form the search tree of the $F$-constrained triangulations on $P$ whose root is $T^*(F)$.

As we have seen in Chapter 7, the time complexity of the reverse search relies on the efficiency of finding the children of each object; in our case finding the children of each $F$-constrained triangulation. This task can be done by using the algorithm for the unconstrained

![Figure 8.4: Existence of an improving flippable edge $e^* = (p_c, q_{j+1})$. The bold edges represent the edges of $F$ and the black vertices represent those incident to $p_c$ in $T^*(F)$.](image)
case by just ignoring the edges of $F$ in the algorithm by Bespamyatnikh [20], and thus we can obtain the algorithm that works in the same time complexity as that of the unconstrained case (see Section 4 of [20]).

**Theorem 8.7.** Let $P$ be a set of $n$ points in the plane. Then, all the $F$-constrained triangulations on $P$ can be reported in $O(\log \log n)$ time per output graph with linear space.

### 8.2.4 Deleting and inserting constrained edges

Let $\mathcal{F}$ be the collection of all non-crossing edge sets on a given point set $P$, and let $T$ be the collection of all triangulations on $P$. We will often treat a triangulation as an edge set in the subsequent discussion. We make use of the construction of the $F$-CLLT as a function $T^*: \mathcal{F} \rightarrow T$ that maps a non-crossing edge set $F$ to the corresponding $F$-CLLT $T^*(F)$.

**Lemma 8.8.** Let $F \in \mathcal{F}$. Then, for $E \subseteq F$, $T^*(F \setminus E) = T^*(F)$ holds if and only if every $e = (p_i, p_j) \in E$ is (i) the upper or lower tangent of $p_i$ with respect to $F$ or (ii) non-flippable in $T^*(F)$.

**Proof.** ("Only-if" part:) Assume, for a contradiction, that there exists $e = (p_i, p_j) \in E$ satisfying neither (i) nor (ii) of the statement when $T^*(F \setminus E) = T^*(F)$ holds. Notice that $T^*(F \setminus E) = T^*(F)$ implies that $T^*(F)$ is the $(F \setminus E)$-constrained lexicographically largest triangulation as well as the $F$-constrained lexicographically largest triangulation. Consider the two triangles of $T^*(F \setminus E)$ incident to $e$, and denote the two vertices appearing in these triangles other than $p_i$ and $p_j$ by $v$ and $w$. Since $e$ is flippable in $T^*(F \setminus E) (= T^*(F))$, the quadrilateral $p_i v p_j w$ is convex. In addition, since $e$ is neither upper nor lower tangent of $p_i$, both $v$ and $w$ lie on the right side of $p_i$, and hence $e \prec (v, w)$ holds. Therefore, flipping $e$ to $(v, w)$ produces an $(F \setminus E)$-constrained triangulation that is lexicographically larger than $T^*(F)$, which is a contradiction.

("If" part:) Let $e$ be the upper or lower tangent of some point $p_i$ with respect to $F$. Observe that flipping $e$ in $T^*(F)$ decreases the lexicographical ordering of the edge list.

Also observe that, if $e$ satisfies (i) and (ii), then every edge of $T^*(F) \setminus (F \setminus E)$ satisfies (i) and (ii). Hence $T^*(F)$ can be seen as a $(F \setminus E)$-constrained triangulation such that flipping any unconstrained edge (i.e., an edge of $T^*(F) \setminus (F \setminus E)$) does not increase the lexicographical ordering of the edge list. Theorem 8.6 says that any $(F \setminus E)$-constrained triangulation can be transformed to the $(F \setminus E)$-constrained lexicographically largest triangulation by improving flips, and hence $T^*(F)$ must be the $(F \setminus E)$-lexicographically largest triangulation. \qed

### 8.2.5 Maintaining the $F$-constrained lexicographically largest triangulation

Let us discuss how to maintain the $F$-CLLT when we newly insert one constrained edge $e$ to $F$. Developing the following efficient way to construct $T^*(F \cup \{e\})$ from $T^*(F)$ will be helpful for constructing the fast enumeration algorithm discussed in Section 8.4.1.
Lemma 8.9. Let \( T^*(F) \) be the F-CLLT on a given set of \( n \) points, and let \( e \) be an edge that does not properly intersect any edge of \( F \). Then, it takes \( O(n) \) time to construct \( T^*(F \cup \{e\}) \) from \( T^*(F) \).

Proof. Let \( e = (p_i, p_j) \), and let \( I \) be the set of edges of \( T^*(F) \) that properly intersect \( e \). Let us first verify the following fact: Every edge of \( T^*(F) \setminus I \), say \((p_k, p_l) \in T^*(F) \setminus I \), is still contained in \( T^*(F \cup \{e\}) \).

Let us consider how \( T^*(F) \) is determined by Construction 1 around \( p_k \). Since \((p_k, p_l) \in T^*(F) \), there exists the cone \( C_F \) with apex \( p_k \) considered in Step 2 of Construction 1 which contains \( p_l \), and the convex hull \( H_F \) of \( P_{k+1} \cap C_F \) with \( p_l = (p_k, p_l) \cap H_F \). Similarly, when constructing \( T^*(F \cup \{e\}) \), in Step 2 we shall consider the convex hull \( H_{F+e} \) inside some cone with apex \( p_k \) such that \( p_l \in H_{F+e} \).

When inserting \( e \), the vertices that are not visible from \( p_k \) with respect to \( F \) remain nonvisible from \( p_k \) with respect to \( F \cup \{e\} \). This implies \( H_{F+e} \subseteq H_F \). Hence, \( p_l = (p_k, p_l) \cap H_F \) implies \( p_l = (p_k, p_l) \cap H_{F+e} \), and \((p_k, p_l) \) remains in \( T^*(F \cup \{e\}) \) from Construction 1.

Therefore, the update occurs only inside the two polygons obtained by removing the edges of \( I \) and adding \( e \) (see Figures 8.5(a) and 8.5(b)). Without loss of generality, we assume that \( e \) is horizontal, and let us show an efficient algorithm to triangulate (the interior of) the polygon lying on the upper side of \( e = (p_i, p_j) \) (the lower side can be treated similarly). Consider the updated triangulation of the polygon by Construction 1. There exist two types of new edges: (1) lower tangent of each vertex of the polygon with respect to the boundary of the polygon, and (2) the others (see Figure 8.5(c)). We call them type (1) and type (2), respectively.

Let us consider how to find type (1) edges. Let \( v \) be a vertex of the polygon which misses the lower tangent in \( (T^*(F) \setminus I) \cup \{e\} \), that is, \( e \) properly intersects the lower tangent \( (v, v^\text{low}) \) of \( v \) with respect to \( F \) which existed in \( T^*(F) \). Consider a ray emanating from \( v \) to \( v^\text{low} \), which first hits \( e \) before reaching \( v^\text{low} \). Rotating the ray around \( v \) in counterclockwise order inside the polygon until it encounters a vertex of the polygon, we can find the new lower tangent \( (v, v^\text{\tilde{low}}) \) of \( v \), which is a type (1) edge (if \( (v, v^\text{\tilde{low}}) \) does not already exist in \( T^*(F) \)). We repeatedly continue the rotation of the ray around the newly encountered vertex \( v^\text{\tilde{low}} \) of the polygon until the ray encounters \( p_j \). Since we are rotating the ray in one direction, the sequence of vertices encountered in this process induces a convex chain connecting \( p_j \) and some vertices of the polygon. Consequently, the set of all type (1) edges is a subset of the convex chains connecting \( p_j \) and each vertex of the polygon as shown in Figure 8.5(c), which represent the shortest paths inside the polygon from \( p_j \). It is known [46] that the shortest paths from a single source to all vertices inside a simple polygon can be computed in \( O(n) \) time, although it requires an involved linear-time algorithm for triangulating a simple polygon [24]. Thus, we could obtain the desired time complexity through the shortest path algorithm.

Our problem, however, can be solved easily by performing a Graham scan (see, e.g., [28]) only once. Let us try to construct the lower part of the convex hull of the vertices of the polygon by performing a Graham scan algorithm from \( p_j \) to \( p_l \). We remark that the algorithm
scans all vertices not in the order of the coordinates as usual but in the vertex sequence order of the polygon from \( p_j \) to \( p_i \). When we encounter a new vertex \( p \) during the scan, we examine the top vertex \( q \) and the next one \( r \) on the stack. If the angle of the three points around \( q \) inside the polygon is convex (i.e. \( \Delta(r, q, p) > 0 \)), we draw the edge between \( p \) and \( r \) and then pop \( q \) from the stack. We continue this process until we obtain three vertices \( p, q', r' \) whose angle around \( q' \) inside the polygon is reflex, that is, \( \Delta(r', q', p) < 0 \). Then, we insert \( p \) into the stack and proceed to the next vertex. Repeating this process until \( p = p_i \) and the stack contains only \( p_i \) and \( p_j \), we can draw all of the required edges in linear time.

8.3 Enumerating Non-crossing Geometric Graphs

8.3.1 General idea

We define the relation \( \sim \) on \( F \) as follows: for two non-crossing edge sets \( F \) and \( F' \), \( F \sim F' \) if and only if \( T^*(F) = T^*(F') \). Let \( [T] = \{ F \in F \mid F \sim T \} \) for each \( T \in T \).

Lemma 8.10. The relation \( \sim \) is an equivalence relation on \( F \). The collection \( \{ [T] \mid T \in T \} \) of all equivalence classes forms a partition of \( F \).

Proof. Since \( \prec \) is a total ordering on \( E \), \( T^*(F) \) is uniquely determined for each \( F \in F \). Hence, \( \sim \) clearly is an equivalence relation, and \( \{ [T] \mid T \in T \} \) forms a partition of \( F \).

We say that an edge \( e \) of \( F \in F \) is the smallest or largest one among \( F \) if it is the smallest edge, or respectively the largest edge, among \( F \) with respect to the edge ordering \( \prec \). We remark that the upper tangent (and the lower tangent, resp.) of \( p_i \) with respect to \( F \) is the smallest edge (and the largest edge, resp.) among \( \{(p_i, q) \in T^*(F) \mid q \in \{p_{i+1}, \ldots, p_n\} = P_{i+1} \} \). This implies that, for any \( F \in [T] \) of a triangulation \( T \), the upper and lower tangents with respect to \( F \) are equivalent to the smallest and largest ones among \( \{(p_i, q) \in T \mid q \in P_{i+1} \} \). Using Lemma 8.8, a unique minimal representative set for each \( [T] \) is defined as follows.
8.3. Enumerating Non-crossing Geometric Graphs

**Lemma 8.11.** Let $T$ be a triangulation on a given point set $P$, and let $F^*$ be the set of all flippable edges in $T$ except for the smallest and largest edges of $\{(p_i, q) \in T \mid q \in P_{i+1}\}$ for every $p_i \in P$. Then, for any $F \in \mathcal{F}$, $F \in [T]$ if and only if $F^* \subseteq F \subseteq T$.

**Proof.** Let us first show $F^* \in [T]$. It is obvious that $T^*(T) = T$. Note that, by the definition of $F^*$, every edge $e = (p_i, p_j) \in T \setminus F^*$ is non-flippable in $T$, or the smallest or largest edge among $\{(p_i, q) \in T \mid q \in P_{i+1}\}$ (i.e., $e$ is the upper or lower tangent of $p_i$ with respect to $T$). Hence, by Lemma 8.8, removing $T \setminus F^*$ does not change the triangulation, that is, $T = T^*(T) = T^*(T \setminus (T \setminus F^*)) = T^*(F^*)$.

The “if-part” can be proved in the same way as above. In fact, removing the edges of $F \setminus F^*$, we obtain $T^*(F) = T^*(F \setminus (F \setminus F^*)) = T^*(F^*) = T$ by Lemma 8.8. Let us consider the “only-if” part. It is obvious that $F \subseteq T$ if $F \in [T]$. Suppose that $F$ (with $F \subseteq T$) is a counterexample, that is, $T^*(F) = T$ but $F \setminus F^* \neq \emptyset$. Then an edge $e = (p_i, p_j) \in F^* \setminus F$ is flippable in $T$ and neither the smallest nor largest edge among $\{(p_i, q) \in T \mid q \in P_{i+1}\}$ by the definition of $F^*$, which implies $T = T^*(F) = T^*(F^* \setminus (F^* \setminus F)) \neq T^*(F^*)$ by Lemma 8.8. This contradicts $T = T^*(F^*)$. \qed

Thus, we call $F^*$ defined in Lemma 8.11 the minimal representative set of $T$, denoted by $R(T)$. Our enumeration algorithm, which consists of two phases, can be easily described as follows.

**Algorithm 1: Enumeration of $\mathcal{NGG}$.**

**Phase 1:** Enumerate all triangulations for a given point set $P$ based on the fast enumeration algorithm by Bespamyatnikh [20].

**Phase 2:** Every time a new triangulation $T$ is found, enumerate all graphs $G$ contained in $T$ such that $G \in \mathcal{NGG}$ and $G$ contains the minimal representative set $R(T)$ as its subset, i.e., $G$ is an $R(T)$-constrained graph in $T$.

Let $\mathcal{C}$ be the graph class obtained by relaxing the non-crossing constraint from the non-crossing geometric graph class $\mathcal{NGG}$ (i.e., the collection of geometric graphs whose edge sets are not necessarily non-crossing but satisfy the combinatorial properties of $\mathcal{NGG}$). Notice that in Phase 2 the problem of enumerating all graphs of $\mathcal{NGG}$ is reduced to that of enumerating all elements of $\mathcal{C}$ containing $R(T)$ in a triangulation $T$ because $T$ is non-crossing. This implies that we may utilize an oracle for enumerating all graphs of $\mathcal{C}$ in a given (abstract) graph, and we can ignore “geometric” and “non-crossing”.

The algorithm needs $R(T)$ explicitly for every $T$ in Phase 2, and hence it is better to maintain and update $R(T)$ during the enumeration of triangulations rather than to compute it from scratch. The task of Phase 1 is in fact not only the enumeration of $T$ but also the generation of $R(T)$. This additional task can be handled by slightly modifying the triangulation enumeration, which will be discussed more formally in Section 8.3.2. Figure 8.6 shows an example of the enumeration of triangulations and the minimal representative sets.

**Theorem 8.12.** Algorithm 1 enumerates all graphs of $\mathcal{NGG}$ without repetitions.
Proof. Consider an arbitrary graph $G \in \mathcal{NGG}$. Then, $T = T^*(G)$ is uniquely determined. This implies $G \in [T]$ and $G \notin [T']$ for any triangulation $T'$ with $T' \neq T$ by Lemma 8.10. Since $G \in [T]$ implies $R(T) \subseteq G \subseteq T$ by Lemma 8.11, Phase 2 of Algorithm 1 for the triangulation $T$ enumerates $G$ by an (assumed) oracle. On the other hand, $G \notin [T']$ implies $R(T') \not\subseteq G$ or $G \not\subseteq T'$. Thus, any $G$ is enumerated exactly once in Phase 2 for $T = T^*(G)$. \qed

8.3.2 Time complexity of Algorithm 1

In order to analyze the time complexity of Algorithm 1, let us take a look at the enumeration algorithm of triangulations by Bespamyatnikh [20], which is based on the reverse search technique [11] (see Section 7.2.2 for more detail on the reverse search). The search graph of the algorithm by Bespamyatnikh is defined in such a way that two triangulations are connected if and only if they can be transformed to each other by a diagonal flip (see Figure 8.6). The following lemma states how to efficiently maintain the minimal representative set during the enumeration of triangulations.

Lemma 8.13. Let $T_1$ and $T_2$ be two triangulations for which $T_2$ is obtained from $T_1$ by a diagonal flip of the edge $f$. Then, the size of the symmetric difference between $R(T_1)$ and $R(T_2)$ is constant. More specifically, only the four edges of the two triangle faces incident to $f$ are involved in the symmetric difference.

Proof. Let us first characterize $e \in R(T_1) \setminus R(T_2)$ with $e \neq f$. There are two cases: (Case 1) $e = (p_i, p_j) \in R(T_1)$ becomes non-flippable in $T_2$, and (Case 2) $e = (p_i, p_j)$ becomes the smallest or largest edge among $\{(p_i, q) \in T_2 \mid q \in P_{i+1}\}$ in $T_2$. Notice that a diagonal flip
switches at most the four flippable edges in $T_1$ into non-flippable edges in $T_2$, and hence, if $e$ is an edge of Case 1, it must be one of the four edges of the triangles incident to $f$. Let us consider Case 2. Since $e$ is not the smallest (or largest, resp.) one in \(\{(p_i, q) \in T_1 \mid q \in P_{i+1}\}\), there exists an edge \(e' = (p_i, q')\) in $T_1$ with $e' \prec e$ (or $e \prec e'$, resp.) such that $e'$ and $e$ are incident to a common triangle face of $T_1$. We notice that $e'$ disappears in $T_2$ because $e$ becomes the smallest (or largest) one, and hence $e'$ is exactly $f$. So, $f$ and $e$ are incident to the same triangle face in $T_1$.

The analogous argument works for $e \in R(T_2) \setminus R(T_1)$.

By Lemma 8.13, during Algorithm 1, the symmetric difference of the minimal representative sets can be output in $O(1)$ time if the triangulation is maintained in a proper data structure, and a flag is attached to each edge to indicate whether it is in the minimal representative set or not. The algorithm by Bespamyatnikh [20] enumerates all triangulations in $O(\log \log n)$ time per output. We thus obtain the following theorem:

**Theorem 8.14.** Let $C$ be the graph class obtained by relaxing the non-crossing constraint from $NGG$. Suppose that there exists an algorithm for enumerating all $R(T)$-constrained graphs of $C$ in a triangulation without repetitions in time $t_C$ per output graph with preprocessing time $t_{C, \text{pre}}$. Then, all graphs of $NGG$ on a given set $P$ of $n$ points can be enumerated without repetitions in $O((\log \log n + t_{C, \text{pre}}) \cdot \text{tri}(P) + t_C \cdot \text{ngg}(P))$ time, where $\text{tri}(P)$ and $\text{ngg}(P)$ are the total numbers of triangulations and $NGG$ on $P$, respectively.

Most of the enumeration algorithms we will use as a subroutine in the applications take $t_{C, \text{pre}} = O(n)$ time in the preprocess phases (see Section 8.3.3).

### 8.3.3 Applications of Algorithm 1

**Enumerating non-crossing spanning trees**

We show here how to apply Algorithm 1 to the enumeration of non-crossing spanning trees on a given point set. What we have to consider here is just how to enumerate all spanning trees in a given triangulation $T$, each of which contains the minimal representative set $R(T)$. We remark that, in the above process, we do not have to care about whether an output spanning tree is non-crossing because $T$ is non-crossing. In Phase 2 of Algorithm 1, we use the algorithm for enumerating all spanning trees on a given undirected graph developed by Kapoor and Ramesh [70] or Shioura et al. [108, 109]. These algorithms can enumerate all spanning trees of a given graph in $O(1)$ time per output graph\footnote{The algorithm outputs each graph by the compact form, that is, the symmetric difference between the last found object and the current one otherwise it takes $O(n)$ time to output each graph. We remark that the symmetric difference between the last found object and the current one is not necessary of constant size. It can be shown that, if the symmetric difference between two consecutive objects in the search tree (or the branch-and-bound tree) is at most $k$, then it takes $O(k)$ time to output an object on average (see e.g. [70, 108, 109, 121] for more details).} with $O(n + m)$ preprocessing time, where $n$ and $m$ denote the numbers of vertices and edges of a given graph. The edge
constraint can be handled easily by contracting the constraint edges before calling these oracles. Hence, applying algorithms of [70, 108, 109] to the resulting (multi-)graph, we can enumerate all the $R(T)$-constrained spanning trees contained in $T$ in $t_C = O(1)$ time per output graph with $t_{C,\text{pre}} = O(n)$ preprocessing time (for contracting the edges of $R(T)$ and for the preprocessing of [70, 108, 109]). Thus, by Theorem 8.14 the following result is derived:

**Theorem 8.15.** Let $P$ be a set of $n$ points in the plane. Then the set of non-crossing spanning trees on $P$ can be enumerated in $O(n \cdot \text{tri}(P) + \text{st}(P))$ time.

**Remark.** Provided that there exists a constant $c (> 1)$ for which $c^n \cdot \text{tri}(P) \leq \text{st}(P)$ holds for every $P \subset \mathbb{R}^2$ of $n$ points, the above running time is dominated by $\text{st}(P)$. It is known that $\text{st}(P)$ becomes minimum when $P$ is in a convex position. On the other hand, $\text{tri}(P)$ is not always minimum for convex positions (see [4]). Furthermore, the number of $\text{st}(P)$ in the convex position is known to be $\Theta(6.75^n)$ [38] relative to the number of triangulations, which is $\Theta(4^n)$, where we ignore polynomial factors. Hence, we strongly conjecture that there exists such a constant $c > 1$.

**Enumerating non-crossing spanning connected graphs**

We show here how Algorithm 1 can be applied to the enumeration of non-crossing spanning connected graphs. To efficiently perform Phase 2 of Algorithm 1, we need an algorithm for enumerating all spanning connected subgraphs of a given graph. Although, to the best of our knowledge, previously there was no efficient enumeration algorithm for this graph class, we observe that they can be enumerated in $O(1)$ time per output with $O(n)$ preprocessing time with a slight modification of the algorithm by Uno [121], which was developed for the enumeration of all bases of a matroid (including spanning trees). Let us briefly explain how to modify this algorithm.

The algorithm by Uno is based on the branch-and-bound technique described in the introduction with a balancing operation and a sophisticated amortized analysis. Let $G = (V, E)$ be a given graph. Let us index the edges of $E$ by $e_i$ with $1 \leq i \leq |E|$ in an arbitrary order. Consider enumerating all the spanning trees (i.e., the bases of a graphic matroid) in $G$ by the branch-and-bound technique, starting with a given graph and recursively dividing the problem into two subproblems, where one subproblem is obtained by removing an edge $e_i$, and the other is obtained by contracting $e_i$ in the $i$th step. If the graph obtained by removing $e_i$ is disconnected, the algorithm does not proceed further, which is a bounding operation. This algorithm outputs each spanning tree when it reaches a leaf of the branch-and-bound tree. Due to the bounding operation, we easily observe that the algorithm keeps a spanning connected graph during the search. Hence, by outputting graphs not only at leaves but also at some internal nodes, we can enumerate all spanning connected subgraphs of $G$.

To make this more precise, let us take a look at this branch-and-bound tree in more detail. We can associate a spanning connected subgraph with each of its nodes as follows. The given graph $G$ is associated with the root node of the branch-and-bound tree. Suppose that a node $N$ at depth $i$ has the associated graph $G'$. Then, $N$ has two children, which have
the associated graphs $G' \setminus \{e_i\}$ and $G'$, respectively, if $G' \setminus \{e_i\}$ is connected. Otherwise $N$ has only one child $N'$ which has the associated graph $G'$. Note that $G'$ is associated with both $N$ and its one child $N'$. However, since the edge $e_i$ is never removed from the graph at depth greater than $i$, the edge $e_i$ in $G'$ may be considered as a contracted edge at $N'$.

In order to enumerate all the spanning connected subgraphs without repetitions, we initially output $G$ at the root node and then inductively output $G' \setminus \{e_i\}$ after the branching operation at depth $i$ (if $G' \setminus \{e_i\}$ is connected). To see the correctness, let us consider a spanning connected subgraph $G'' = (V, E'')$ of $G$. Let $j$ be the maximum index among $E \setminus E''$. Then, since $G''$ can be obtained by removing $E \setminus E''$ from $G$, there exists a node $N''$ at depth $j + 1$ whose associated graph is $G''$. It is not difficult to see that a node has the associated graph $G''$ only if it is either $N''$ or descendants of $N''$, and the algorithm outputs $G''$ only at $N''$.

To achieve $O(1)$ running time per graph, we just output the symmetric difference of two spanning connected subgraphs which are consecutively output during the enumeration, following the balancing technique proposed by Uno [121]. As a result, the collection of the spanning connected subgraphs can be enumerated in the same time bound as that of the spanning trees.

The edge constraint can be treated easily by edge contraction, and thus all the $R(T)$-constrained spanning connected subgraphs of $T$ can be enumerated in $t_{C} = O(1)$ time per output with $t_{C, pre} = O(n)$. Combined with Theorem 8.14, we found that Algorithm 1 enumerates all the non-crossing spanning connected graphs in $O(n \cdot \tri(P) + cg(P))$ time. Moreover, we obtain the following result.

**Theorem 8.16.** For every general point set $P$ in the plane with $n$ points, $1.52^{n-1} \tri(P) \leq cg(P)$ holds.

**Proof.** Let $T$ be a triangulation on $P$ with the minimal representative set $R(T)$. We show that, for every $T$, there exist at least $1.52^{n-1}$ non-crossing spanning connected subgraphs in $T$ that are not contained in the other triangulations.

Let us first show that, for every triangle face $p_ip_jp_k$ of $T$, $\{(p_i, p_j), (p_i, p_k), (p_j, p_k)\} \cap R(T) \leq 2$. Without loss of generality, assume that $p_i < p_j < p_k$. Then, notice that the edge $(p_j, p_k)$ is the largest or smallest edge among $\{(p_j, q) \in T \mid q \in P_{j+1}\}$. Hence, $(p_j, p_k)$ is not contained in $R(T)$ by definition of the minimal representative set given in Lemma 8.11.

Consider a subset $S$ of $T$ such that (i) $S$ forms a spanning connected graph on $P$, (ii) $S$ contains $R(T)$ as its subset, and (iii) $S$ has the minimum edge cardinality among the subsets of $T$ satisfying (i) and (ii). Then, from the above discussion, $S$ contains at most two edges for each face of $T$. Since $S$ is an $R(T)$-constrained non-crossing spanning connected graph on $P$, $S \cup F$ forms a distinct $R(T)$-constrained non-crossing spanning connected graph on $P$ for every $F \subseteq T \setminus S$. The number of bounded faces of a triangulation is known to be $2n - h - 2$, where $h$ is the number of vertices of the convex hull of $P$. Since at least one edge of each triangle is contained in $T \setminus S$ and each edge can belong to at most two triangles, by counting the elements of $T \setminus S$ for each triangle, we have $|T \setminus S| \geq (2n - h - 2)/2$. Therefore, there
exist at least $2^{n-h/2-1}$ subsets of $T \setminus S$, and $T$ contains at least $2^{n-h/2-1}$ $R(T)$-constrained non-crossing spanning connected graphs.

On the other hand (for a point set with large value of $h$), it can be shown that $T$ contains at least $2^{n-h/2-1}$ $R(T)$-constrained non-crossing spanning connected graphs as follows\(^2\). Notice that no edge of the convex hull of $P$ is contained in $R(T)$, and hence removing arbitrary edges of the convex hull results in an $R(T)$-constrained non-crossing spanning connected graph unless both edges incident to a degree-two vertex of $T$ are removed. If an edge of the convex hull is not incident to a degree-two vertex, there are two possibilities to obtain a connected subgraph of $T$ (i.e., remove it or not). For two edges incident to a degree-two vertex, at most one of them can be removed to obtain a connected graph. Hence there are three possibilities for these two edges (i.e., remove either one of the two edges or leave them). The number of ways to obtain connected subgraphs of $T$ in this manner is minimum if the number of degree-two vertices is maximum, i.e., $h/2$. Thus, $T$ contains at least $3^{h/2}$ $R(T)$-constrained non-crossing spanning connected graphs.

Finally, choosing one of two bounds according to whether $h < \frac{2(n-1)}{1+\log_3 3}$ or not, we obtain the claimed lower bound.

Theorem 8.16 implies that the running time of Algorithm 1, which is $O(n \cdot \tri(P) + \cg(P))$, is dominated by $\cg(P)$.

**Theorem 8.17.** Let $P$ be a set of $n$ points in the plane. Then, the set of non-crossing spanning connected graphs on $P$ can be enumerated in $O(\cg(P))$ time.

**Enumerating plane straight-line graphs**

For any $F \subseteq T \setminus R(T)$, $F \cup R(T)$ is a plane straight-line graph containing $R(T)$. Hence, by enumerating (the symmetric differences of) all subsets of $T \setminus R(T)$, we can obtain all $R(T)$-constrained plane straight-line graphs in $T$. Enumerating all subsets of $T \setminus R(T)$ is equivalent to generating all $|T \setminus R(T)|$-bit binary numbers with $O(n)$ preprocessing time, which can be done in constant time per output (see e.g., [103]). Algorithm 1 thus enumerates all the plane straight-line graphs in $O(n \cdot \tri(P) + \pg(P))$ time. Since a non-crossing spanning connected graph is also a plane straight-line graph, $1.52^{n-1}\tri(P) \leq \cg(P) \leq \pg(P)$ holds by Theorem 8.16. Thus, we obtain the following result.

**Theorem 8.18.** Let $P$ be a set of $n$ points in the plane. Then, the set of plane straight-line graphs on $P$ can be enumerated in $O(\pg(P))$ time.

**Enumerating non-crossing perfect matchings**

Given a point set $P$ of $2n$ points, a non-crossing perfect matching is a non-crossing geometric graph on $P$ such that every point of $P$ is incident to exactly one edge of the graph.

\(^2\)This lower bound for the case of large $h$ was pointed out by an anonymous referee of Discrete & Computational Geometry.
Let us consider how to design Phase 2 of Algorithm 1. Suppose that we have an algorithm for finding a perfect matching in a given (non-geometric) graph in \( t_{PM} \) time if it exists. Then, using this algorithm as an oracle, the naively implemented branch-and-bound algorithm can enumerate all the perfect matchings in \( O(nt_{PM}) \) time per output graph [87]. The edge constraint can be treated easily. If \( T \) has a vertex that is incident to more than one edge of \( R(T) \), we report that there is no \( R(T) \)-constrained perfect matching in \( T \). Otherwise we first remove all edges of \( R(T) \) together with the vertices incident to \( R(T) \) and then apply the above algorithm for enumerating perfect matchings to the resulting graph. By putting \( R(T) \) back to each solution, we obtain all the perfect matchings in \( T \) that contain \( R(T) \). Algorithm 1 hence enumerates all the non-crossing perfect matchings on \( P \) in \( O(t_{PM} \cdot \text{tri}(P) + nt_{PM} \cdot \text{pm}(P)) \) time.

### 8.4 \( \mathcal{I} \)-independent Minimal Representative Sets

We know that the algorithm by Bespamyatnikh [20] enumerates all triangulations efficiently, but its search tree is not nicely structured when we focus on the minimal representative sets (see Figure 8.6). Namely, for two triangulations \( T \) and \( T' \) for which \( T \) is a parent of \( T' \) in the search tree, \( T' \) may miss some representative edge that appears in \( T \). Consider, for example, the enumeration of non-crossing matchings. In Phase 2 of Algorithm 1 for a triangulation \( T \), the algorithm outputs no \( R(T) \)-constrained non-crossing matching if there is a vertex incident to more than one edge of \( R(T) \). However, since some descendant triangulation \( T' \) of \( T \) may not have a vertex which is incident to more than one edge of \( R(T') \), \( T' \) may contain an \( R(T') \)-constrained non-crossing matching, and thus we cannot skip the enumeration of \( T \) and its descendants. The next proposed algorithm avoids this inefficiency.

We first propose a new algorithm for enumerating triangulations whose search tree has a monotone structure with respect to the minimal representative sets such that \( R(T) \subset R(T') \) holds for any triangulation \( T \) and its descendant \( T' \) (see Figure 8.8). Using this monotonicity, we can efficiently enumerate only the minimal representative sets possessing the specified property, which allows us to skip the output of unnecessary triangulations. Let us explain this idea more formally. Recall that \( \mathcal{F} \) denotes the collection of all non-crossing edge sets on \( P \). Let \( \mathcal{I} \) be a subset of \( \mathcal{F} \) satisfying the following independent system:

\[ (\mathcal{I}'1) \quad \emptyset \in \mathcal{I}. \]

\[ (\mathcal{I}'2) \quad \text{If } F_2 \in \mathcal{I} \text{ and } F_1 \subseteq F_2, \text{ then } F_1 \in \mathcal{I}. \]

A non-crossing edge set \( F \in \mathcal{F} \) is called \( \mathcal{I} \)-independent edge set or \( \mathcal{I} \)-independent if \( F \in \mathcal{I} \). If \( \mathcal{I} \) satisfies the following condition,

\[ (\mathcal{I}'3) \quad \text{for every } G \in \mathcal{NGG}, \, G \in \mathcal{I} \text{ (where } G \text{ is considered as an edge set),} \]

then we can ensure that the minimal representative set of \( T^*(G) \) is \( \mathcal{I} \)-independent for every \( G \in \mathcal{NGG} \). This implies that it is sufficient to enumerate only the \( \mathcal{I} \)-independent minimal representative sets to enumerate all graphs of \( \mathcal{NGG} \).
8.4.1 Enumerating triangulations based on edge insertions

Our new enumeration algorithm for triangulations is also based on the reverse search [10, 11] whose search tree can be characterized by the root triangulation and the parent-child relation (see Section 8.3.2 for a brief explanation of the reverse search). Here, we define $T^*(\emptyset)$ as the root triangulation. Hence, the minimal representative set of the root triangulation is empty. For each non-root triangulation $T$, the parent of $T$ is defined as $T^*(R(T) \setminus \{e\})$ with the smallest edge $e$ among $R(T)$ with respect to the edge ordering $\prec$. The correctness of our parent-child relation follows from the next lemma.

**Lemma 8.19.** Let $T$ be a triangulation with $R(T) \neq \emptyset$. Then, for any $e \in R(T)$, the minimal representative set of $T^*(R(T) \setminus \{e\})$ is $R(T) \setminus \{e\}$.

**Proof.** Let $T' = T^*(R(T) \setminus \{e\})$. It is sufficient to show $R(T) \setminus \{e\} \subseteq R(T')$ because $R(T') \subseteq R(T) \setminus \{e\}$ by Lemma 8.11.

Consider any $(p_i, p_j) \in R(T) \setminus \{e\}$ with $p_i < p_j$. Let $p_ip_jv$ and $p_ip_jw$ be the two triangles incident to $(p_i, p_j)$ in $T$, and similarly let $p_ip_jv'$ and $p_ip_jw'$ be those in $T'$. Without loss of generality, we assume that $v$ and $v'$ (and $w$ and $w'$, resp.) lie on the right side (and the left side, resp.) of $(p_i, p_j)$. Note that $p_i < v$ and $p_i < w$ hold since $(p_i, p_j) \in R(T)$. If $v = v'$ and $w = w'$, then the triangle faces incident to $(p_i, p_j)$ do not differ between $T$ and $T'$. Hence $(p_i, p_j) \in R(T)$ implies $(p_i, p_j) \in R(T')$ by the definition of the minimal representative set given in Lemma 8.11.

Let us consider the case of $v \neq v'$. When generating $T = T^*(R(T))$ by Construction 1, there exists the cone $C$ with apex $p_i$ which is bounded by $(p_i, p_j)$ and the other consecutive edge among $\delta_{R(T)}(p_i) \cup \{(p_i, p_i^{up}), (p_i, p_i^{low})\}$ to $(p_i, p_j)$ and which contains both $p_j$ and $v$ since $p_i < v$. Let $H$ be the convex hull of $P_i \cap C$. Then, $v = (p_i, v) \cap H$ holds since $(p_i, v) \in T^*(R(T))$. Similarly, when constructing $T' = T^*(R(T) \setminus \{e\})$, there exists the convex hull $H'$ just below $(p_i, p_j) \in R(T) \setminus \{e\}$ for which $v' = (p_i, v') \cap H$ holds. Since every vertex visible from $p_i$ with respect to $R(T)$ is still visible from $p_i$ with respect to $R(T) \setminus \{e\}$, all the right endpoints of the edges of $\delta_{R(T)}(p_i) \cup \{(p_i, p_i^{up}), (p_i, p_i^{low})\}$ are still visible from $p_i$ with respect to $R(T) \setminus \{e\}$. Thus, $H \subseteq H'$ holds, and hence $H'$ contains $v$ (see Figure 8.7).

It is easily observed that, since $H \subseteq H'$, $(p_i, p_j)$ does not become the smallest one among $\{(p_i, q) \in T' \mid q \in P_{i+1}\}$ when removing $e$ (and it is not the largest one either). Hence, by the definition of the minimal representative set, $(p_i, p_j) \in R(T')$ if $(p_i, p_j)$ is flippable in $T'$. Since there exists no point of $P$ inside the triangle $p_ip_jv$ and no point inside $p_ip_jv'$, either one of the following two cases occurs depending on the position of $v'$: (i) $(p_i, v')$ intersects $(v, p_j)$, or (ii) $(v', p_j)$ intersects $(p_i, v)$. When (i) holds, $v'$ is properly contained in $H$. However, since $H \subseteq H'$, $v'$ is also properly contained in $H'$, which contradicts $v' = (p_i, v') \cap H'$. Thus, (ii) must hold. In this case the inner angles $\angle p_ip_jv$ and $\angle p_ip_jv'$ satisfy $\angle p_ip_jv' \leq \angle p_ip_jv$. Applying a similar argument to the pair of $w$ and $w'$, we have $\angle p_ip_jw' \leq \angle p_ip_jw$. (However, it is not difficult to see $w = w'$ from the fact that $e$ properly intersects $(v', p_j)$ but not $(p_i, p_j)$.) Hence, the inner angle of the quadrilateral $p_ip_jv'w'$ at $p_j$ is less than $\pi$ because $(p_i, p_j)$ is flippable in $T$. 

---

**Figure 8.7:** Illustration for Lemma 8.19.
Let us show that the opposite angle, that is, the inner angle of the quadrilateral $p_i v' p_j w'$ at $p_i$, is also less than $\pi$. This can be proved from the fact that both of $v'$ and $w'$ are on the right side of $p_i$ since $(p_i, p_j)$ is neither the smallest nor largest one among \{(p_i, q) \in T' \mid q \in P_i + 1\} with respect to $\prec$. Hence $(p_i, p_j)$ is flippable in $T'$ and $(p_i, p_j) \in R(T')$ follows. \hfill $\square$

By Lemma 8.19, $R(T) \subset R(T')$ holds for any triangulation $T$ and its descendant $T'$. Moreover, since the root triangulation has an empty minimal representative set, our definition of the parent-child relation correctly induces a rooted search tree on the collection of all triangulations. The algorithm traces this search tree in depth-first manner. We call this new algorithm edge-insertion algorithm for (enumerating) triangulations. An example of the new search tree is depicted in Figure 8.8.

Let us analyze the time complexity of the edge-insertion algorithm. In the reverse search the most time-consuming part is to find all children $T'$ of a triangulation $T$, i.e., to find all edges $e \in K_n$ for which $T' = T^*(R(T) \cup \{e\})$ is a child of $T$. Such $e$ can be characterized by the following lemma.

**Lemma 8.20.** Let $T$ and $T'$ be triangulations on $P$ for which $T' = T^*(R(T) \cup \{e\})$ holds for
some \( e \in K_n \), where \( e \) does not properly intersect any edge of \( R(T) \). Then \( T' \) is a child of \( T \) if and only if all of the following three conditions are satisfied:

(a) \( e \notin T \),
(b) \( e < e_1 \), where \( e_1 \) is the lexicographically smallest edge among \( R(T) \), and
(c) \( R(T) \subseteq R(T') \).

Proof. (“Only-if”-part) Let \( e' \) be the lexicographically smallest edge among \( R(T') \). Note that by Lemma 8.19 \( R(T) = R(T') \setminus \{e'\} \) whenever \( T \) is a parent of \( T' \) (i.e., \( T = T^*(R(T) \setminus \{e'\}) \)). Hence (c) holds. Also, since \( T' = T^*(R(T) \cup \{e\}) \), we have \( R(T') \subseteq R(T) \cup \{e\} \) by Lemma 8.11. Hence, combining \( R(T') \subseteq (R(T') \setminus \{e'\}) \cup \{e\} \) and \( e' \in R(T') \), we obtain \( e = e' \). Consequently, \( e \) is the lexicographically smallest edge among \( R(T') = R(T) \cup \{e\} \), implying (b).

Suppose, for a contradiction, that (a) does not hold. Then, we have \( R(T) \subseteq R(T) \cup \{e\} \subseteq T \) since \( e \in T \), and hence we obtain \( R(T) \cup \{e\} \in T \) by Lemma 8.11, which implies \( T' = T^*(R(T) \cup \{e\}) = T \). This contradicts that \( T' \) is a child of \( T \).

(“If”-part:) First let us show \( e \in R(T') \). Suppose otherwise; then, by the definition of \( R(T') \), \( e \) is non-flippable in \( T' \), or the smallest or largest edge among \( \{(p_i, q) \in T' \mid q \in P_{i+1}\} \) for the left endpoint \( p_i \) of \( e \). We hence have, by Lemma 8.8, \( e \in T'' = T^*(R(T) \cup \{e\}) = T^*(R(T) \cup \{e\}) = T^*(R(T)) = T \), which contradicts condition (a).

Combining \( e \in R(T') \) and condition (c), we obtain \( R(T) \cup \{e\} \subseteq R(T') \). On the other hand \( R(T') \subseteq R(T) \cup \{e\} \) is known from Lemma 8.11. Therefore, \( R(T') = R(T) \cup \{e\} \). Condition (b) says that \( e \) is the smallest edge among \( R(T) \cup \{e\} \), and hence, according to the definition of the parent, \( T^*(R(T') \setminus \{e\}) = T^*(R(T)) = T \) is the parent of \( T' \).

We now concentrate on how to find all edges that produce children of a given triangulation \( T \). We first show that all edges satisfying the conditions of Lemma 8.20 can be found in \( O(cn^2) \) time for each \( T \), where \( c \) is the subscription of the left endpoint \( p_i \) of the smallest edge among \( R(T) \) (and \( c \) is defined to be \( n \) if \( R(T) = \varnothing \)). Note that the number of edges satisfying condition (b) can be bounded from above by \( \sum_{i=1}^{c} (n - i) < cn \). The algorithm checks each of these edges one by one whether it satisfies the other conditions (a) and (c) in \( O(n) \) time (per edge). Clearly condition (a) can be checked in \( O(n) \) time. To check (c) the algorithm explicitly constructs \( T' = T^*(R(T) \cup \{e\}) \) in \( O(n) \) time based on the method of Lemma 8.9 for each edge \( e \) satisfying (a) and (b). Then it is enough to check whether all edges of \( R(T) \) are contained in \( R(T') \) in \( T' \). This can be done in \( O(1) \) time for each edge of \( R(T) \) (due to the definition of the minimal representative set given in Lemma 8.11), and thus (c) can be checked in \( O(n) \) time. As a result, we can find all edges that satisfy all the conditions of Lemma 8.20 in \( O(cn^2) \) time.

This \( O(cn^2) \) time is improved to \( O(n^2/c) \) time by a simple amortized analysis as follows. Consider the point set \( P' = \{p_1, \ldots, p_k\} \). We claim that, for any edge \( e \in K_n \setminus T \) whose both endpoints are contained in \( P' \), \( e \) always satisfies all the conditions of Lemma 8.20. Since such \( e \) clearly satisfies (a) and (b) from its definition, let us confirm that \( e \) also satisfies (c). Notice that every edge of \( R(T) \) lies completely to the right side of the right endpoint of \( e \) due to
the definition of \( p_e \). Hence, inserting \( e \) into \( R(T) \) does not affect the right side of \( p_e \) when constructing \( T^*(R(T) \cup \{e\}) \), i.e., every \( e' \in R(T) \) is incident to the same two triangles in \( T^*(R(T) \cup \{e\}) \) as in \( T = T^*(R(T)) \), and all edges of \( R(T) \) are still contained in the minimal representative set of \( T^*(R(T) \cup \{e\}) \). Thus, \( e \) satisfies (c).

The number of edges \( e \in K_n \setminus T \) whose both endpoints are contained in \( P' \) is at least \( c(c-1)/2 - (3c - 6) \). This implies that there exist \( \Omega(c^2) \) children of \( T \). Distributing the time \( O(cn^2) \) evenly to \( \Omega(c^2) \) children and \( T \) itself, we obtain the result.

**Theorem 8.21.** Let \( P \) be a set of \( n \) points. Then, the edge-insertion algorithm enumerates all the triangulations on \( P \) in \( O(n^2) \) time per output graph without repetition.

### 8.4.2 Enumerating \( I \)-independent minimal representative sets

Owing to the nicely structured search tree of the minimal representative sets, we can now perform the efficient enumeration of the \( I \)-independent minimal representative sets (defined at the beginning of Section 8.4) and the corresponding triangulations.

**Algorithm 2: Enumeration of \( N\mathcal{GG} \).**

**Phase 1:** Execute the edge-insertion algorithm starting from \( T^*(\emptyset) \) as described in Section 8.4.1 to enumerate triangulations.

**Phase 2:** Every time a new triangulation \( T \) is found, check whether \( R(T) \) is \( I \)-independent or not. If \( R(T) \) is dependent, skip the enumeration of all the descendants of \( T \).

**Phase 3:** Every time a new \( I \)-independent \( R(T) \) is found, enumerate all \( R(T) \)-constrained graphs of \( N\mathcal{GG} \) in \( T \).

The correctness of Algorithm 2 follows from the next lemma.

**Lemma 8.22.** Let \( I \) be the collection of \( I \)-independent edge sets of \( \mathcal{F} \). Then, Algorithm 2 correctly enumerates all graphs of \( N\mathcal{GG} \) without repetitions if \( I \) satisfies (I'1), (I'2), and (I'3).

**Proof.** We first note that all of the \( I \)-independent minimal representative sets are correctly enumerated in Algorithm 2. To verify this, let us imagine the search tree which is obtained by performing the edge-insertion algorithm for enumerating triangulations. The subgraph of this search tree induced by all \( T \) with \( R(T) \in I \) forms a rooted tree by (I'1) and (I'2), and hence the algorithm enumerates every \( I \)-independent \( R(T) \) correctly.

Let us show that every \( G \in N\mathcal{GG} \) is actually enumerated. Lemma 8.11 states \( R(T^*(G)) \subseteq G \subseteq T^*(G) \). Since \( G \in I \) holds by (I'3), \( R(T^*(G)) \in I \) follows from (I'2). Thus, \( G \) is enumerated in Phase 3 for \( T^*(G) \).

Let us analyze the time complexity of Algorithm 2 under the assumption that \( I \) satisfies (I'1), (I'2), and (I'3). Assume that there exists an oracle that checks in \( t_{\text{check}} \) time whether \( I \cup \{e\} \in I \) or not for an \( I \)-independent set \( I \) and an edge \( e \in K_n \). Let \( I_{\text{rep}} \subseteq I \) be the collection of the \( I \)-independent minimal representative sets on a given point set \( P \). We can
easily observe that the time to be spent in Phases 1 and 2 is $O(n^2 \cdot t_{\text{check}} \cdot |I_{\text{rep}}|)$ since there exist $O(n^2)$ children for each triangulation on the search tree and from Theorem 8.21. Hence, using the notation $C, t_C$ and $t_{C,\text{pre}}$ defined in Theorem 8.14, we obtain the following result:

**Theorem 8.23.** Algorithm 2 enumerates all the graphs of $NGG$ on a given point set $P$ without repetitions in $O((n^2 \cdot t_{\text{check}} + t_{C,\text{pre}}) \cdot |I_{\text{rep}}| + t_C \cdot \text{ngg}(P))$ time. Moreover, the time complexity is bounded by $O((n^2 \cdot t_{\text{check}} + t_{C,\text{pre}} + t_C) \cdot \text{ngg}(P))$, which is polynomial on average, if $|I_{\text{rep}}| \leq \text{ngg}(P)$.

**8.4.3 Application of Algorithm 2**

We show here how Algorithm 2 can apply to the enumeration of non-crossing minimally rigid graphs discussed in Chapter 7.

We define an $\mathcal{I}$-independence on $\mathcal{F}$ in such a way that $F \in \mathcal{F}$ is $\mathcal{I}$-independent if and only if $F$ is independent in the generic rigidity matroid on $K_n$. Then, since the edge set of each minimally rigid graph is a base of the rigidity matroid, the collection $\mathcal{I}$ of the $\mathcal{I}$-independent edge sets of $\mathcal{F}$ satisfies (I'1), (I'2) and (I'3). Lemma 7.3 says that for a non-crossing edge set $F$ on $P$ that is an independent set in the generic rigidity matroid every $F$-constrained triangulation on $P$ contains an $F$-constrained minimally rigid graph. This implies that a triangulation $T$ contains at least one $R(T)$-constrained non-crossing minimally rigid graph if $R(T)$ is $\mathcal{I}$-independent. Namely, $|I_{\text{rep}}| \leq \text{mrg}(P)$.

Let us consider the time complexity of Phase 2 of Algorithm 2. Recall that, for a graph $G = (V, I)$ with $n$ vertices and an independent set $I$ of the rigidity matroid, a maximal rigid subgraph $G' = (V', I')$ of $G$ is called a rigid component. Then $I \cup \{e\}$ is independent if and only if there is no rigid component containing both endpoints of $e$ in $G[I]$. It is known that all the rigid components of $G$ can be detected in $O(n^2)$ time [17, 80]. Moreover, using the data structure that maintains rigid components, it can be checked in $O(1)$ time whether two vertices belong to the same rigid component. Thus, the algorithm can check in $t_{\text{check}} = O(1)$ time whether the minimal representative set of a new child, that is, $R(T) \cup \{e\}$, is independent or not. If $R(T) \cup \{e\}$ is independent, the algorithm enters Phase 3 while updating the rigid components in $t_{\text{update}} = O(n)$ time for each edge insertion [17]. Algorithm 2 hence enumerates all the $\mathcal{I}$-independent minimal representative sets (and the corresponding triangulations) in $O(n^2 \cdot t_{\text{check}} + t_{\text{update}}) = O(n^2)$ time per output.

As for Phase 3, we can use the algorithm proposed in Theorem 7.1 which enumerates all the minimally rigid graphs in a graph $G$ that contain a specified edge set in $t_C = O(n)$ time per output graph with $t_{C,\text{pre}} = O(n^2)$ preprocessing time. Putting these facts and Theorem 8.23 together gives the following result:

**Theorem 8.24.** Let $P$ be a set of $n$ points in the plane. Then the set of non-crossing (generically) minimally rigid graphs on $P$ can be enumerated without repetitions in $O(n^2 \cdot \text{mrg}(P))$ time.

We note that Algorithm 1 enumerates all non-crossing minimally rigid graphs in $O(n^2 \cdot \text{tri}(P) + n \cdot \text{mrg}(P))$ time.
8.5 Other Applications

We briefly show below applications of the proposed technique to further graph classes. Algorithm 1 always works in time proportional to the number of triangulations and objects to be enumerated. Whereas, in some problems, Algorithm 2 works practically faster than Algorithm 1, although it seems a nontrivial task to evaluate its running time theoretically.

Non-crossing bar-and-slider frameworks. Combining the enumeration technique with Theorem 5.2 (that generalizes Laman’s theorem to bar-and-slider cases), we obtain an efficient algorithm for enumerating all non-crossing $k$-dof bar-and-slider mechanisms. The problem is formulated as follow:

Input: A generic joint configuration $p$ some of which are connected with external environment by a set of sliders.

Output: The list of the underlying graphs of all non-crossing bar-and-slider frameworks connecting these joints whose degree of freedom is equal to $k$.

Recall that $G$ is the underlying graph of a minimally rigid bar-and-slider framework if and only if its edge set is a base of the matroid $\mathcal{M}_{\mu_2'}$ induced by the submodular function

$$\mu_2'(F) = \begin{cases} 
\varrho_{2,3}(F) = 2|V(F)| - 3 & \text{if } L(F) = \emptyset \\
\mu_2(F) = 2|V(F)| - 2 + \min\{\chi(F), 2\} & \text{otherwise}.
\end{cases}$$

Therefore, if we consider a truncated matroid of $\mathcal{M}_{\mu_2'}$, the problem can be reduced to the enumeration of non-crossing edge sets that are bases of the matroid. In exactly the same way as in Section 8.4.3, Algorithm 2 can enumerate all such edge sets in $O(n^2)$ time per output.

Non-crossing red-and-blue matchings. For a given point set $P$, every point is assumed to have either red or blue color. A non-crossing red-and-blue matching is a non-crossing matching on $P$ each of whose edges is not allowed to connect points of the same color. The enumeration can be performed by using the algorithm for enumerating the matchings in a (non-geometric) bipartite graph [119] in Phase 2 of Algorithm 1 or in Phase 3 of Algorithm 2, which needs $t_C = O(n)$ time per output with $t_{\mathcal{C},\text{pre}} = O(n^{3/2})$ preprocessing time (if the edge cardinality of a given graph is $O(n)$). Hence, by Theorem 8.14, Algorithm 1 enumerates all non-crossing red-and-blue matchings in $O(n^{3/2} \cdot \text{tri}(P) + n \cdot \text{rbm})$ time, where \text{rbm} is the total number of non-crossing red-and-blue matchings on $P$, which depends not only on $P$ but also on the coloring of each point.

Algorithm 2 can enumerate all the red-and-blue-matchings efficiently if we define $\mathcal{I}$ as the collection of $F$ such that no two edges of $F$ are incident to a vertex and no edge of $F$ connects points of the same color. Notice that every $\mathcal{I}$-independent minimal representative set is also a non-crossing red-and-blue matching, which implies $|\mathcal{I}_{\text{rep}}| \leq \text{rbm}$. The $\mathcal{I}$-independence of each non-crossing edge set is trivially checked in $t_{\text{check}} = O(1)$ time, and thus Algorithm 2 works in $O(n^2 \cdot \text{rbm})$ time by Theorem 8.23.
Non-crossing $k$-vertex or $k$-edge connected graphs. A non-crossing $k$-vertex (or $k$-edge) connected graph is a non-crossing geometric graph spanning a given point set $P$ that remains connected after removing any $k-1$ vertices (or $k-1$ edges) from the graph. Since it can be checked in a polynomial time $Q_k$ whether a given (non-geometric) graph is $k$-vertex connected (or $k$-edge connected) or not, according to the branch-and-bound technique discussed in the introduction, we can enumerate $k$-vertex connected (or $k$-edge connected) subgraphs in $t_C = O(mQ_k)$ time per output with $t_{C,\text{pre}} = O(n + m + Q_k)$ preprocessing time, where $m$ denotes the number of edges in a subgraph. Thus, using this algorithm in Phase 2, Algorithm 1 enumerates all non-crossing $k$-vertex (or $k$-edge) connected graphs on $P$.

In particular, it is known that 2-vertex (or 2-edge) connectivity of a graph can be checked in linear time (see, e.g., [104, Chapter 15.2b]). Moreover, $\text{tri}(P) \leq c_{g_2}(P)$ for every point set $P$ since every triangulation is also a non-crossing 2-vertex (or 2-edge) connected graph on $P$. Algorithm 1 hence enumerates all the non-crossing 2-vertex (or 2-edge) connected graphs in $O(n^2c_{g_2}(P))$ time.

Non-crossing directed spanning trees. Each edge of the given geometric complete graph on $P$ is assumed to have an orientation. A non-crossing directed spanning tree (or non-crossing $r$-arborescence) is a non-crossing spanning tree on $P$ having a unique directed path from a rooted point $r$ to all points of $P \setminus \{r\}$. The enumeration can be performed by using the algorithm of [71, 120] in Phase 2 of Algorithm 1, or in Phase 3 of Algorithm 2. Given a digraph $D$ whose number of arcs is $O(n)$, this algorithm enumerates all the directed spanning trees in $D$ in $t_C = O(\log^2 n)$ time per graph with $t_{C,\text{pre}} = O(n \log n)$ preprocessing time. Hence, Algorithm 1 works in $O(n \log n \cdot \text{tri}(P) + \log^2 n \cdot \text{dst})$ time, where $\text{dst}$ denotes the total number of the non-crossing directed spanning trees which depends not only on $P$ but also on the orientation of $D$.

Algorithm 2 can enumerate all the non-crossing directed spanning trees if we define $\mathcal{I}$ as the collection of the non-crossing edge sets $F$ such that $F$ has no cycle and no vertex has indegree more than one in the directed graph induced by $F$, then clearly $t_{\text{check}} = O(1)$. Its running time becomes $O(n^2 \cdot |\mathcal{I}_{\text{rep}}| + \log^2 n \cdot \text{dst})$.

Edge-constrained non-crossing geometric graphs. The technique can be also applied to the enumeration of $S$-constrained non-crossing geometric graphs that are those containing a given specified edge set $S$ as their subsets, e.g., $S$-constrained non-crossing spanning trees or $S$-constrained non-crossing matchings. This is because both the algorithm by Bespamyatnikh [20] and the edge-insertion algorithm proposed in this paper for enumerating triangulations can be naturally extended to those for enumerating only the $S$-constrained triangulations by restricting the collection of non-crossing edge sets $F$ to those containing $S$ as their subsets.

For Algorithm 1, the $S$-constrained triangulations can be enumerated in $O(\log \log n)$ time.
per output (Theorem 8.7), while the edge-insertion algorithm for Algorithm 2 enumerates them in $O(n^3)$ time per output by setting the root as $T^*(S)$ instead of $T^*(\emptyset)$. (Note that we cannot use an amortized analysis as done in the proof of Theorem 8.21 to achieve an $O(n^2)$ bound.)

8.6 Conclusion

We proposed a new algorithmic framework for the efficient enumeration of non-crossing geometric graphs, and by applying our technique we obtained improved algorithms for several specific graph classes. Also, using a generalization of Laman’s theorem to bar-and-slider frameworks proposed in Chapter 5, we obtained an efficient algorithm for enumerating all non-crossing $k$-dof mechanisms connecting given joints some of which are supported by line-sliders.

An open problem related to non-crossing geometric graphs is whether one can enumerate all non-crossing perfect matchings on a given point set in polynomial time (on average). In Section 8.3.3 we showed an algorithm that works in $O(n^{3/2}\text{tri}(P) + n^{5/2}\text{pm}(P))$ time. However, pm$(P)$ may be smaller than tri$(P)$, and hence this algorithm is not efficient from the theoretical viewpoint.

An open problem related to the generation of discrete structures is to develop an efficient algorithm for enumerating circuits of a generic rigidity matroid. This is of considerable practical importance for the design of tensegrity structures. The detailed background will be discussed in the next chapter.
Chapter 9

Conclusion

In this dissertation, we have presented new combinatorial characterizations of rigidity of discrete structures, in particular bar-and-slider frameworks and panel-and-hinge frameworks, and presented efficient algorithms for generating bar-and-joint frameworks.

In Chapter 4, we newly considered a problem of partitioning a graph into edge-disjoint rooted-forests and presented a necessary and sufficient condition as a generalization of the Tutte-Nash-Williams tree-packing theorem.

In Chapter 5, we presented combinatorial characterizations of infinitesimal rigidity of 2-dimensional bar-and-slider frameworks based on a rooted-forest partition given in Chapter 4. In particular, we proved that, even though the directions of sliders are predetermined and degenerate (i.e., some sliders have the same direction), it is combinatorially decidable whether the framework is infinitesimally rigid or not. As a corollary, we obtained an $O(n^2)$-time algorithm for computing the degree of freedom of a bar-and-slider framework on a generic joint configuration.

In Chapter 6, we affirmatively proved the Molecular conjecture posed by Tay and Whiteley in 1984. In particular, we presented a combinatorial characterization of the infinitesimal rigidity of panel-and-hinge frameworks in terms of the number of edge-disjoint spanning trees that can be packed into the underlying graphs. Also, as a corollary, we obtained a characterization of the degree of freedom of a 3-dimensional bar-and-joint framework of the square of a graph.

In Chapter 7, we dealt with the problem of enumerating minimally rigid bar-and-joint frameworks connecting a given set of $n$ joints on the plane. Based on the well-known reverse search paradigm, we have presented an algorithm for enumerating non-crossing minimally rigid frameworks in $O(n^3)$ time per output. In particular, we have proved that any non-crossing minimally rigid graphs can be converted to each other by $O(n^2)$ remove-add flip operations.

In Chapter 8, extending the idea of using edge-constrained triangulations considered in Chapter 7, we proposed a general enumeration technique that can be applies to arbitrary non-crossing geometric graph classes. We showed that this technique provides not only faster algorithms for some enumeration problems compared with existing ones but also first algo-
rithms for various problems that had not been considered to the best of our knowledge. It is notable that, using a generalization of Laman’s theorem to bar-and-slider frameworks (proposed in Chapter 5), we obtained an efficient algorithm for generating all non-crossing $k$-dof mechanisms connecting given joints some of which are connected with external environment by a set of sliders. This broadens the possibility of applications (compared with the one considered in Chapter 7).

We conclude this chapter by remarking future directions of research concerning combinatorial rigidity. A big unsolved problem is of course to understand the infinitesimal rigidity of 3-dimensional bar-and-joint frameworks. Although some special cases including the square of a graph are well studied, there seems no convincing conjecture for general cases. See the survey [56] for more detail.

Another interesting problem related to this dissertation is the one posed by Schulz. In [105], he studied the rigidity/flexibility of bar-and-joint frameworks that possess non-trivial symmetries and proved extended-versions of Maxwell’s rule, Laman’s theorem, and Henneberg construction. He also conjectured symmetric extensions of other frameworks including panel-and-hinge frameworks and molecular frameworks (i.e., a generalization of the Molecular conjecture). This direction is particularly important in applications.

An interesting problem related to the generation of discrete structures is to enumerate tensegrity frameworks (see e.g., [69] for tensegrities). It is known [101] that the edge set of the underlying graph of a tensegrity framework is a redundantly rigid graph (i.e., removing any edge still remains rigid). A minimal edge set that forms a redundantly rigid graph in $\mathbb{R}^2$ is a circuit of the rigidity matroid (see e.g., [68]). However, there is no general technique that leads to an efficient algorithm for enumerating all circuits of a matroid (see [78]). It is known that all circuits of a graphic matroid (i.e., cycles of a graph) can be enumerated efficiently [99], but is not obvious in a rigidity matroid, even in a generic 2-dimensional rigidity matroid. Thus, enumerating all circuits of a generic 2-dimensional rigidity matroid must be an interesting problem from practical and theoretical viewpoints.
Bibliography


voronoi diagram of a simple polygon in linear time. SIAM Journal on Computing,

Available at http://www.math.cornell.edu/~connelly/.

563, 2005.


[33] W. Cunningham. Improved bounds for matroid partition and intersection algorithms.

[34] L. de Floriani and E. Puppo. An on-line algorithm for constrained Delaunay triangula-

[35] Z. Fekete. Source location with rigidity and tree packing requirements. Operations

of the 2nd International Workshop on Algorithmic Aspects of Wireless Sensor Networks,

[37] Z. Fekete and L. Szegő. A note on \([k, l]\)-sparse graphs. In Graph Theory in Paris; A

[38] P. Flajolet and M. Noy. Analytic combinatorics of non-crossing configurations. Discrete


[40] A. Frank and L. Szegő. Constructive characterizations for packing and covering with


